

Escola Tècnica Superior
d'Enginyeria



Universitat de València



Máster en Sistemas y Servicios de la Sociedad de la Información

PROYECTO FINAL DE MÁSTER

**Desarrollo de sitio web y aplicación de gestión de
contenidos para el Observatorio Beyond The Line de la
Universidad CEU Cardenal Herrera con .NET y SQL Server**

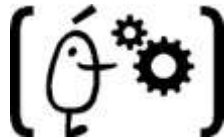
AUTOR:

José Antonio Martínez Gómez

TUTORA:

Inmaculada Coma Tatay

Escola Tècnica Superior
d'Enginyeria



Universitat de València



Máster en Sistemas y Servicios de la Sociedad de la Información

PROYECTO FINAL DE MÁSTER

**Desarrollo de sitio web y aplicación de gestión de
contenidos para el Observatorio Beyond The Line de la
Universidad CEU Cardenal Herrera con .NET y SQL Server**

AUTOR:

José Antonio Martínez Gómez

TUTORA:

Inmaculada Coma Tatay

A mi tutora por orientarme y apoyarme en
todas mis decisiones y a mi familia por
aguantarme y por llevarme en volandas
hasta aquí.

Resumen

La Universidad CEU Cardenal Herrera cuenta con el Servicio de Comunicación Digital para la elaboración de la página web corporativa de la universidad y de las aplicaciones web de gestión de contenidos tratados por los distintos departamentos, servicios y grupos de investigación pertenecientes a la misma. Para ello se hace uso de los servidores, plataformas de desarrollo y licencias (apostando generalmente por Microsoft) dispuestos por el departamento de informática, a cuyas limitaciones hay que adaptarse, pero de los que aún se puede obtener más rendimiento, ya que hasta hace poco se hacía uso de tecnologías de programación obsoletas.

Recientemente se ha procedido a la renovación completa de la página web corporativa (la antigua estaba programada con ASP clásico, pasando a utilizar el framework .NET como plataforma para el desarrollo de la nueva) y la migración o renovación de todas las aplicaciones de gestión y sitios web propios de los departamentos, servicios y grupos de investigación pertenecientes a la universidad, para conseguir la mayor integración posible con su sitio web corporativo.

Uno de esos grupos de investigación es el Observatorio Beyond The Line, que se dedica al estudio y divulgación de técnicas no convencionales de publicidad y comunicación empresarial. Para ello quiere servirse de su página web como herramienta principal de difusión de los contenidos, necesitando una web más moderna y que se actualice dinámicamente mediante una aplicación lo suficientemente amigable como para ser utilizada por cualquier miembro de su equipo, expertos en publicidad, pero no necesariamente expertos en programación.

Este proyecto relata el proceso de elección de la plataforma de programación que se ha considerado más adecuada, adaptándose al escenario dispuesto por el departamento de informática de la Universidad CEU Cardenal Herrera, la recopilación de requisitos obtenidos durante las entrevistas con miembros del equipo del observatorio, el proceso de diseño, desarrollo e implementación del sitio web y de su aplicación de gestión y las pruebas necesarias para su correcto funcionamiento y una eficiente utilización.

Resum

La Universitat CEU Cardenal Herrera compta amb el Servei de Comunicació Digital per a l'elaboració de la pàgina web corporativa de la universitat i de les aplicacions web de gestió de continguts tractats pels diferents departaments, serveis i grups de recerca pertanyents a la mateixa. Per a això es fa ús dels servidors, plataformes de desenvolupament i llicències (apostant generalment per Microsoft) disposats pel departament d'informàtica, a les limitacions dels quals cal adaptar-se, però dels que encara es pot obtenir més rendiment, ja que fins fa poc es feia ús de tecnologies de programació obsoletes.

Recentment s'ha procedit a la renovació completa de la pàgina web corporativa (l'antiga estava programada amb ASP clàssic, passant a utilitzar el framework .NET com a plataforma per al desenvolupament de la nova) i la migració o renovació de totes les aplicacions de gestió i pàgines web pròpies dels departaments, serveis i grups de recerca pertanyents a la universitat, per aconseguir la major integració possible amb la seua pàgina web corporativa.

Un d'aquests grups de recerca és l'Observatori Beyond The Line, que es dedica a l'estudi i divulgació de tècniques no convencionals de publicitat i comunicació empresarial. Per a això vol servir-se de la seua pàgina web com a eina principal de difusió dels continguts, necessitant una web més moderna i que s'actualitze dinàmicament mitjançant una aplicació prou amigable com per ser utilitzada per qualsevol membre del seu equip, experts en publicitat, però no necessàriament experts en programació.

Aquest projecte relata el procés d'elecció de la plataforma de programació que s'ha considerat més adequada, adaptant a l'escenari disposat pel departament d'informàtica de la Universitat CEU Cardenal Herrera, la recopilació de requisits obtinguts durant les entrevistes amb membres de l'equip de l'observatori, el procés de disseny, desenvolupament i implementació de la pàgina web i de la seua aplicació de gestió i les proves necessàries per al seu correcte funcionament i una eficient utilització.

Índice

1. Introducción	13
1.1. El Observatorio Beyond The Line	14
2. Motivación y Objetivos	15
2.1. Motivación	15
2.2. Objetivos	16
3. Estado del Arte	17
3.1. Plataformas de desarrollo web	17
3.1.1. Java EE	18
3.1.1.1. Breve historia	18
3.1.1.2. Requisitos y posibilidades	19
3.1.1.3. IDE Recomendado	19
3.1.1.4. Servidor de aplicaciones recomendado	20
3.1.1.5. SGBD recomendado	20
3.1.2. PHP	20
3.1.2.1. Breve historia	20
3.1.2.2. Requisitos y posibilidades	21
3.1.2.3. IDE Recomendado	21
3.1.2.4. Servidor web recomendado	22
3.1.2.5. SGBD recomendado	22
3.1.3. ASP.NET	22
3.1.3.1. .NET Framework	22
3.1.3.2. Breve historia	23
3.1.3.3. Requisitos y posibilidades	24
3.1.3.4. IDE Recomendado	25
3.1.3.5. Servidor web recomendado	25
3.1.3.6. SGBD recomendado	26
3.1.4. Resumen	26
3.2. Sistemas de Gestión de Contenidos (CMS)	27

3.2.1.	Plataformas basadas en ASP.NET: DotNetNuke	27
3.2.1.1.	Características básicas	28
3.2.1.2.	Módulos	29
3.2.1.3.	Descarga e instalación	31
3.2.1.4.	Limitaciones	31
3.2.2.	Umbraco	31
3.2.2.1.	Características básicas	32
3.2.2.2.	Limitaciones	33
3.2.3.	Otros CMS basados en .NET	33
3.2.4.	Plataformas basadas en Java: Liferay	34
3.2.4.1.	Características básicas	34
3.2.4.2.	Limitaciones	35
3.2.5.	dotCMS	36
3.2.5.1.	Características básicas	37
3.2.5.2.	Limitaciones	38
3.2.6.	Otros CMS basados en Java	38
3.2.7.	Plataformas basadas en PHP. Wordpress	38
3.2.7.1.	Características básicas	39
3.2.7.2.	Limitaciones	41
3.2.8.	Drupal	41
3.2.8.1.	Características básicas	42
3.2.8.2.	Limitaciones	44
3.2.9.	Otros CMS basados en PHP	45
3.3.	Tratamiento del contenido del sitio web. Sindicación, metadatos y posicionamiento	45
3.3.1.	Sindicación y metadatos en un solo marco: RDF	46
3.3.2.	Metadatos y posicionamiento SEO con HTML	47
3.3.3.	Sindicación mediante RSS	48
3.3.4.	Indexación y posicionamiento mediante SITEMAP de ASP.NET	50

4. Especificación	51
4.1. Especificación de requisitos	51
4.1.1. Funcionales	51
4.1.1.1. Requisitos funcionales para la página web del Observatorio.	51
4.1.1.2. Requisitos funcionales generales de la aplicación de gestión.	53
4.1.1.3. Requisitos funcionales para el rol de profesor en la aplicación de gestión.	54
4.1.1.4. Requisitos funcionales para el rol de administrador en la aplicación de gestión.	55
4.1.1.5. Requisitos funcionales para el rol de colaborador.	55
4.1.2. No funcionales	55
4.2. Planificación	56
4.2.1. Descomposición del proyecto en partes más asequibles	57
4.2.2. Identificación y definición de las actividades del proyecto	57
4.2.3. Diagrama de Gantt	58
4.2.4. Planificación de recursos	60
4.2.5. Prelaciones	61
4.3. Estimación de costes	61
4.3.1. Estimación en función de las líneas de código	61
4.3.2. Estimación mediante modelo COCOMO	61
4.3.3. Estimación de costes de personal	64
4.3.4. Estimación de recursos de Software y Hardware	64
4.3.4.1. Hardware	64
4.3.4.2. Software	65
4.3.4.3. Material fungible de oficina	65
4.3.4.4. Material bibliográfico	65
4.3.5. Resultado de la estimación de costes	66

4.3.6.	Revisión de los objetivos. Estudio de viabilidad	66
4.3.6.1.	Análisis de riesgos	66
4.3.6.2.	Estudio de la viabilidad económica/técnica/legal	69
5.	Desarrollo del proyecto	70
5.1.	Análisis del sistema	70
5.1.1.	Casos de uso	70
5.1.2.	Diagrama de actividad	81
5.1.3.	Diagrama de clases para el modelo de datos	83
5.2.	Diseño del sistema	85
5.2.1.	Aplicación de gestión del observatorio	85
5.2.1.1.	Diagramas de componentes	85
5.2.1.2.	Diagramas de clases	94
5.2.1.3.	Diagramas de secuencia	96
5.2.2.	Sitio web del Observatorio	100
5.2.2.1.	Diagramas de componentes	100
5.2.2.2.	Diagramas de clases	107
5.2.2.3.	Diagramas de secuencia	110
5.3.	Implementación del sistema	113
5.3.1.	Herramientas e interfaces usadas para la implementación	113
5.3.2.	SQL Server Management Studio Express	113
5.3.2.1.	Instalación	113
5.3.2.2.	Diseño de tablas SQL para aplicación web	115
5.3.2.3.	Diseño de procedimientos almacenados SSMSE 2005	117
5.3.3.	Visual Studio 2010	119
5.3.3.1.	Instalación	119
5.3.3.2.	Desarrollo de la aplicación y el sitio web con Visual Studio 2010	120
5.3.3.3.	Biblioteca de clases	122

5.3.3.4.	Otras clases implementadas	124
5.3.3.5.	Implementación de capas de presentación y negocio	126
5.3.4.	Despliegue de la aplicación	134
6.	Pruebas y resultados	140
6.1.	Descripción de experimentos	140
6.1.1.	Aplicación de gestión del Observatorio	140
6.1.2.	Sitio web del Observatorio	149
6.2.	Pruebas y resultados	152
6.2.1.	Accesibilidad	152
6.2.1.1.	Nivel de prioridad 1	152
6.2.1.2.	Nivel de prioridad 2	154
6.2.1.3.	Nivel de prioridad 3	157
6.2.2.	Test de visión vischeck	157
6.2.3.	Usabilidad. Evaluación Heurística	160
6.2.3.1.	Sitio web del Observatorio	160
6.2.3.2.	Aplicación de gestión del Observatorio	166
6.2.4.	Test de usuarios	172
6.2.5.	Minería y posicionamiento web	173
6.2.6.	Pruebas de carga y stress	177
6.3.	Evaluación presupuestaria	179
7.	Conclusiones y trabajo futuro	180
7.1.	Conclusiones	180
7.2.	Trabajo futuro	181
8.	Bibliografía	183
8.1.	Bibliografía impresa	183
8.2.	Enlaces web	183
9.	Anexos	184

1 – Introducción

La Universidad CEU Cardenal Herrera inauguró su Servicio de Comunicación Digital con el objetivo de difundir y promocionar su imagen corporativa y la de sus diferentes departamentos, servicios y facultades a través de las nuevas tecnologías de la información y la comunicación que emergían en ese momento. Para ello se adaptó la infraestructura informática, basada en las plataformas y los productos de Microsoft, a las tecnologías de desarrollo web que se comercializaban entonces como anexo al servidor Internet Information Server. La más estrechamente relacionada con el modelo tecnológico de Microsoft era la tecnología ASP (Active Server Pages), y con esta tecnología se implementó la web corporativa de la Universidad.

Aunque los sistemas informáticos de la universidad no se han actualizado a la velocidad de aparición de las nuevas tecnologías que se implantaban en el mercado, si que ha tratado de mantener el nivel suficiente para soportar e integrar las tecnologías de desarrollo que permitieran estar a la altura de las circunstancias, y a día de hoy, la Universidad CEU Cardenal Herrera dispone de servidores web equipados con Windows Server 2003 y servidor de aplicaciones IIS 6, además de poseer todas las licencias necesarias para la utilización de la plataforma .NET Framework para el desarrollo de aplicaciones.

Aunque esta infraestructura ha ido quedándose obsoleta con el paso de los años, la ausencia de actualización de la web corporativa, diseñada a principios de siglo con el lenguaje de programación ASP, no ayudaba a dar el empuje necesario para apoyar la actualización de los servidores y los sistemas operativos que funcionan en la universidad. Éste es el aspecto que ofrecía la página principal de la web corporativa de la Universidad CEU Cardenal Herrera el 1 de febrero de 2006.

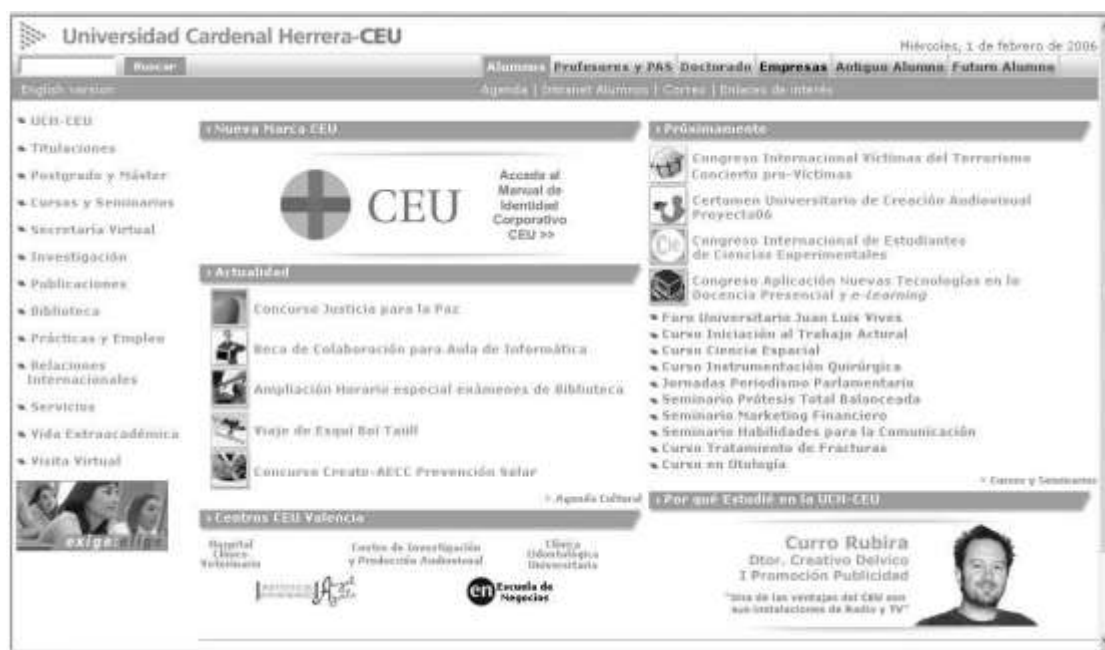


Figura 1.1. Página inicial de la web corporativa de la Universidad CEU Cardenal Herrera.

Desde entonces, aunque la página web mantenía su diseño basado en la tecnología clásica de Microsoft, ASP, no se puede decir que el Servicio de Comunicación Digital de la UCH se mantuviera al margen de las tecnologías emergentes, y las nuevas aplicaciones que se han ido desarrollando desde la adquisición de la plataforma .NET por parte de la universidad se han implementado con el lenguaje de programación ASP.NET, buscando una mayor dinamización de los contenidos y una mayor utilización de las bases de datos para almacenar y tratar la información.

En este sentido cabe destacar que la Universidad CEU Cardenal Herrera dispone de un servidor de bases de datos SQL Server, y que se ha puesto a disposición del Servicio de Comunicación Digital una base de datos, llamada “webceu.web”, donde se han ido creando las tablas y los procedimientos almacenados SQL a los que se accederá desde cualquier sitio o aplicación web dependiente del servicio que se alimente de una base de datos.

Así pues, ambas tecnologías, la clásica y la moderna, han convivido en el servidor web a la espera de la migración, tanto de la web corporativa, como de los sitios web que aún se basan en ASP.

1.1. El Observatorio Beyond The Line.

Entre los sitios web que aún mantienen su diseño basado en el lenguaje clásico de Microsoft, se encuentra la página web del *Observatorio Beyond The Line* de la Universidad CEU Cardenal Herrera. Los objetivos de este Observatorio son conocer la realidad del below a nivel nacional y establecer sus fundamentos teóricos y operativos.

Su página web debe ser la herramienta de difusión principal de las investigaciones de las distintas disciplinas below, como el marketing promocional, el patrocinio, el marketing directo y relacional, el marketing digital, las relaciones públicas, el mecenazgo, el merchandising y las nuevas técnicas de comunicación empresarial. Además también pretende publicar los seminarios y congresos relacionados con las técnicas de publicidad en medios no convencionales, entrevistas a expertos del sector, vídeos y noticias de interés...

En definitiva, a través de la difusión periódica de los resultados obtenidos, el Observatorio pretende convertirse en referente en el estudio y aplicación de estas disciplinas y contribuir a la formación y cualificación de los futuros profesionales de la publicidad en las nuevas técnicas de comunicación empresarial.

2 – Motivación y Objetivos

2.1 - Motivación

Como se ha introducido en el apartado anterior, el Servicio de Comunicación Digital de la Universidad Cardenal Herrera ya trabaja en el desarrollo de su nueva página web corporativa, y muchas de las aplicaciones alojadas en la web antigua van a ser migradas. Las páginas y aplicaciones diseñadas con ASP van a ser desarrolladas de nuevo, y aquellas aplicaciones programadas con la nueva plataforma, serán migradas a la nueva versión del .NET Framework para unificar tecnologías.

Como becario recién llegado a este servicio, se me ha propuesto iniciar mi colaboración en el proyecto de la universidad desarrollando el nuevo sitio web del *Observatorio Beyond The Line*, así como una aplicación de gestión dinámica de sus contenidos que irá dirigida a los miembros del equipo del Observatorio, profesionales de la comunicación publicitaria *pero desconocedores* de temas relacionados con el desarrollo web.

La web actual del Observatorio es uno de los sitios desarrollados en el antiguo ASP, y por eso va a ser rediseñada e integrada en la nueva web corporativa de la universidad. Además, se pretende desarrollar una aplicación de gestión de contenidos, con el objetivo de que la información pueda ser actualizada dinámicamente por el equipo del Observatorio, y no de forma estática por el Servicio de Comunicación Digital, como venía ocurriendo hasta ahora.



Figura 2.1. Antigua web del Observatorio Beyond The Line. Diseñada con ASP.

Según dijo el presidente del Grupo AGR Comunicación, Mateo Blay, "El Observatorio Below debe abarcar tanto las técnicas más asentadas en el ámbito del below the line como las que

surjan para los nuevos soportes tecnológicos, así como las tendencias internacionales que vayan a llegar pronto a España desde Estados Unidos o Japón”. Parece evidente que es necesaria de manera inminente la actualización y dinamización de su sitio web para estar en disposición de cumplir estos objetivos usando ésta como su herramienta principal.

Por otra parte, aludiendo a la infraestructura que pone a disposición la Universidad CEU Cardenal Herrera, con la premisa de unificar tecnologías y habiendo elegido el Framework .NET como plataforma de desarrollo de la nueva página web, parece bastante coherente que éstas sean las tecnologías elegidas para el diseño del sitio web del Observatorio y el desarrollo de la aplicación web para su gestión, aunque para tomar la decisión definitiva habrá que realizar un estudio detallado de las alternativas buscando la que mejor se adapte al escenario actual.

2.2 – Objetivos

Los objetivos marcados en este proyecto son los siguientes:

- Investigación de las tecnologías de desarrollo existentes en el mercado para la elección de la plataforma más adecuada para implementar la web del Observatorio Beyond The Line y la aplicación de gestión de sus contenidos atendiendo a criterios de rendimiento, potencia, escalabilidad y compatibilidad con el escenario que se presenta en la Universidad CEU Cardenal Herrera.
- Desarrollo e implementación del sitio web del Observatorio atendiendo a las especificaciones de los clientes, a las especificaciones de los estándares web, a los criterios de usabilidad y accesibilidad, a los parámetros de SEO actuales y al marco legal correspondiente.
- Desarrollo de la aplicación de gestión de contenidos atendiendo a las especificaciones de los clientes, a las especificaciones de los estándares web, a los criterios de usabilidad y accesibilidad, a los requisitos de seguridad recomendados y al marco legal correspondiente.
- Procurar que la plataforma escogida se integre lo mejor posible con la tecnología utilizada en el sitio web corporativo para homogeneizar lo máximo posible todos los futuros sitios y aplicaciones web de los distintos departamentos y servicios de la Universidad CEU Cardenal Herrera.

3 - Estado del Arte.

A continuación se va a realizar un repaso a los lenguajes de programación que compiten con ASP.NET en el ámbito del desarrollo web y la posibilidad de integración en el servidor de la universidad. También se estudiará la posibilidad de integrar en otro servidor un sistema gestor de bases de datos que complemente a SQL Server, valorando su integración con IIS 6, que es el único parámetro innegociable con el que se gestionan todos los servidores que proporciona el departamento de Nuevas Tecnologías en la Universidad.

Por otra parte, se valorará la opción de uso de un CMS (Sistemas Gestores de Contenidos) para las diversas plataformas existentes, efectuando un estudio de mercado previo, por si resultara rentable utilizar alguno para gestionar el contenido de la web. Hay que tener en cuenta, como se podrá comprobar en el apartado de Especificación de Requisitos, que se trata de una web con una gran cantidad de secciones y de contenido hechos a medida, por lo que la utilización o no de un CMS puede decidirse en función del grado de personalización del nuevo sitio web y de la aplicación encargada de su gestión.

Por último, se hará un repaso a las tecnologías y las técnicas de indexación, posicionamiento y redifusión de las páginas web para estudiar su posible implantación en este proyecto.

3.1 - Plataformas de desarrollo web.

Para escoger una plataforma de desarrollo web adecuada para el ámbito empresarial es fundamental atender a varios criterios, tales como su arquitectura, los servicios prestados y por supuesto sus fortalezas y sus debilidades.

En los comienzos de la web, cuando los sitios web estaban compuestos por páginas estáticas apenas se permitía interacción alguna con el usuario, a excepción de aquéllos que poseían los conocimientos de HTML necesarios. Se trataba de páginas con una tecnología primitiva, basada principalmente en formularios y buscadores básicos pero de difícil construcción. Posteriormente los servidores web fueron evolucionando para permitir diálogos entre capas a partir de la ejecución de bloques de código en el lado del servidor.

En primer lugar aparecieron aplicaciones para transferir datos entre un navegador web y el servidor. La tecnología CGI fue una de las pioneras en crear contenido dinámico para las páginas web. No obstante sus limitaciones, en concreto la ausencia de manejo de sesiones y de control de autorización obligaron a usar lenguajes de script débilmente tipados, como Javascript, ASP o PHP para solventarlas, a costa de perder velocidad, escalabilidad y seguridad frente a los lenguajes compilados.

Con el tiempo los lenguajes han ido evolucionando buscando la solución a estos problemas y han aparecido nuevas plataformas de programación orientadas a cubrir las especificaciones de las aplicaciones empresariales.

Se entiende por aplicación empresarial aquélla que se provee de un servidor de aplicaciones para servir a una gran cantidad de usuarios a través de la red. Debe ser capaz de enviar datos masivos sin descuidar la persistencia, de soportar accesos de manera concurrente, de dar funcionalidad a diversos usuarios mediante las correspondientes interfaces personalizadas, de ser escalable, es decir, que permita la separación entre la lógica de negocio y la presentación, y

también de ser fácilmente integrable en otros sistemas y permitir la integración de otras aplicaciones.

Existen varias tecnologías y lenguajes de programación orientados al desarrollo web pero, en la actualidad, hay tres tecnologías que han tomado la delantera en cuanto a los requisitos comentados arriba y que han absorbido el mercado a nivel empresarial: **Java EE, PHP y ASP.NET**.

3.1.1 - Java EE

Java EE (Java Platform, Enterprise Edition), es una plataforma de programación perteneciente a la plataforma Java, orientada al desarrollo y ejecución de aplicaciones escalables (con una arquitectura basada en N capas distribuidas) desarrolladas en el lenguaje de programación con su mismo nombre, que se sustenta de componentes modulares ejecutados sobre un servidor de aplicaciones.

3.1.1.1 - Breve historia

Java hace su aparición en el año 1995, introducida por la empresa Sun Microsystems®, con el propósito de conseguir que una aplicación desarrollada una única vez pueda ser ejecutada en cualquier ordenador, lo que supuso una revolución y un cambio de paradigma en el mundo de la programación.

La clave de su funcionamiento se basa en la existencia de un elemento llamado “Máquina Virtual de Java” (MVJ), cuya presencia hace posible que esta plataforma se ejecute, independientemente del hardware, en cualquier sistema operativo que dé soporte a la máquina virtual, ayudando así a la portabilidad de las aplicaciones.

Básicamente el compilador crea códigos de byte, llamados “Bytecodes”, que se envían al visualizador solicitado y se interpreta en la máquina virtual, que posee un intérprete de Java, ejecutando las órdenes guardadas en unos archivos con extensión “.class”.

Rendimiento

Como lenguaje de programación, Java reúne todos los requisitos que debe tener una aplicación empresarial. Orientado a objetos, sencillo, orientado también a aplicaciones distribuidas, robusto, fiable, seguro, portable, multi-hilo, dinámico y con gran rendimiento.

Su mayor potencial se encuentra en las API que especifica: tipos de datos, objetos, applets, networking, seguridad y componentes. Estos últimos constituyen la base de la plataforma Java EE.

Nacimiento de Java EE

Para explotar la capacidad del lenguaje orientado a las aplicaciones empresariales se diseñó una plataforma abierta y estándar de Java, conocida como Java 2 Enterprise Edition, que pasaría a llamarse Java EE a partir de la 5ª versión.

Además de proporcionar especificaciones técnicas de descripción del lenguaje, Java EE provee especificaciones únicas para componentes, como los Servlets, los Java Server Pages (JSP) o los Enterprise JavaBeans (EJB). Estos componentes, pensados para construir aplicaciones distribuidas, interaccionan entre sí para formar parte de una aplicación J2EE, desplegándose

de forma individual en contenedores (partes del servidor que ofrecen servicios de bajo nivel al componente).

3.1.1.2 - Requisitos y posibilidades

Portabilidad: Los componentes pueden funcionar en diversos entornos operativos, de trabajo, e incluso del sistema, y ser reubicados de unos a otros sin que se requiera cambios importantes. Su integración en un gran proyecto desarrollado en .NET, como es la página web corporativa de la Universidad Cardenal Herrera es una tarea más complicada.

Diversidad de Ambientes: Los esfuerzos por integrar esta aplicación podrían obligar a mantener más de un tipo de infraestructura de hardware/software que deberían poder aprovecharse de forma conjunta para el desarrollo de las aplicaciones web.

Java EE funciona correctamente sobre diversos servidores de aplicaciones sin necesidad de módulos. Los más utilizados con JBoss, Glassfish y Tomcat. Una fórmula para lograr una conexión a base de datos a través de IIS es instalando Tomcat en él. En cuanto al SGBD, el más eficiente a día de hoy es Oracle, aunque también funciona relativamente bien sobre MySQL. Para aprovechar el SGBD ya instalado en la universidad habría que crear una conexión JDBC con el SGBD de SQL Server.

En cuanto al entorno de desarrollo, las posibilidades de Java EE son enormes, desde IDE como JDeveloper (de Oracle), NetBeans y Eclipse hasta frameworks como Struts, Spring o los JavaServer Faces. El IDE Visual Studio no es compatible con el lenguaje de programación Java EE, pero sí que existe un plugin, llamado Grasshopper, que es capaz de compilar código intermedio de Microsoft a Java Bytecode, lo que permitiría integrar aplicaciones de la web de la universidad en un hipotético servidor con soporte para Java EE para este proyecto.

Oportunidad de aparición y reutilización: Los componentes deberían ser desarrollados pensando en su integración con otras aplicaciones en el momento en que son requeridos, para no afectar al proceso productivo. Sólo en caso de que las soluciones que generara Java EE supusieran una ventaja competitiva lo suficientemente atractiva como para seguir programando con esta plataforma, se le daría viabilidad a esta opción.

3.1.1.3 - IDE recomendado.

Eclipse: Eclipse empezó como un proyecto de IBM Canadá. Fue implementado por OTI, reemplazando a VisualAge. A finales de 2001 se constituyó un consorcio para tratar el desarrollo de Eclipse como código open-source. En 2003 se creó la fundación independiente de IBM.

En la página oficial www.eclipse.org, es definida como “un IDE para todo y particularmente para nada”. Se trata de una plataforma de trabajo sobre la que se instalan herramientas de desarrollo mediante el uso de plugins que permiten, además de integrar varios lenguajes sobre un mismo entorno de desarrollo, inyectar otras aplicaciones complementarias que pueden resultar de utilidad durante el proceso de desarrollo, a saber: utilidades UML, editores visuales de interfaces, ayuda online para librerías, etc.

El IDE Eclipse sólo es una utilidad más englobada bajo el llamado *Proyecto Eclipse*, que combina el desarrollo del IDE Eclipse como varios plugins de gran importancia (por ejemplo el

JDT, para Java, o el CDT, para C/C++). Eclipse puede descargarse desde la página oficial mientras que en la carpeta “plug-in catalogue” pueden encontrarse enlaces a varios plugins adicionales.

Al estar escrito en Java, necesita para ejecutarse de la existencia de un JRE (Java RuntimeEnvironment) que se instala previamente en el sistema.

3.1.1.4 - Servidor de aplicaciones recomendado:

Glassfish: Proyecto lanzado el 6 de junio de 2005, no es hasta el 4 de mayo de 2006 cuando aparece la primera versión apoyando la especificación Java EE 5. Se trata de un servidor de aplicaciones gratuito y open-source desarrollado por Sun, y posteriormente adquirido por Oracle. Basado en un código fuente donado por Sun y Oracle, que le proporcionó el módulo de persistencia TopLink. Posee como base al servidor Sun Java SystemApplication Server, un derivado de Apache Tomcat y usa un componente adicional llamado Grizzly que le dota de escalabilidad y velocidad. El 10 de noviembre de 2009 se lanza la 3ª versión junto a la especificación Java EE 6, que permite migrar fácilmente desde Tomcat, además de ofrecer plugins para NetBeans y Eclipse y preservación de sesiones a través de red despliegues.

3.1.1.5 - SGBD recomendado:

Oracle: Oracle nace en 1977 como Software Development Laboratories (SDL), motivado por un completo estudio sobre los SGBD de George Koch que usó por primera vez la teoría de las bases de datos relacionales. Hoy aún encabeza la lista, encontrándose en muchas industrias alrededor del mundo. A partir de la versión 10g Release 2 lanza varias ediciones, siendo Oracle Database Express Edition la única gratuita.

Se trata de un sistema de gestión de bases de datos muy completo, con soporte para transacciones, estabilidad, escalabilidad y soporte multiplataforma. Comparte su dominio en la actualidad con MS SQL Server y con otros RDBMS como MySQL.

3.1.2 - PHP

PHP no es una plataforma, sino un lenguaje de programación libre que funciona sobre una gran cantidad de plataformas, entornos de desarrollo, sistemas operativos, servidores de aplicaciones y sistemas gestores de bases de datos. Está diseñado también para la creación de páginas web dinámicas y se basa en la interpretación de scripts que generan el contenido dinámico y lo envían al servidor, para que éste lo envíe a su vez al cliente.

3.1.2.1 - Breve historia

PHP nace en 1994 y su padre es Rasmus Lerdorf. Entonces era un conjunto simple de CGIs escritos en el lenguaje de programación C. Un año después apareció un modelo capaz de interaccionar con bases de datos, incorporó interpretación automática de variables de formulario y sintaxis incrustada HTML. Deliberadamente fue asemejándose a C en estructura, además de mantener ciertas similitudes con Perl. Por entonces estaba limitado a sistemas Unix.

PHP 2.0 significó en 1996 una reforma completa del lenguaje, que incluyó soporte interno para DBM, mSQL, Postgres95, cookies y soporte para funciones definidas por el usuario. La aportación de una interfaz madura para múltiples bases de datos, protocolos y APIs, y su

extensión a sistemas Windows y Macintosh hicieron de PHP 3.0 la primera versión más parecida a la actual. PHP 4.0 añadió, en 1998, un nuevo motor para manejar mejor las aportaciones de PHP 3.0, además de sesiones HTTP, buffers de salida, formas más seguras de controlar las entradas de usuario y nuevas construcciones de lenguaje.

Hubo que esperar hasta 2004 para el lanzamiento de la última versión conocida hasta el momento. PHP 5.0. Las mejoras en el soporte para la programación orientada a objetos - la clonación de objetos, el uso de destructores y constructores, la aparición de niveles de acceso (private, public, protected) -, así como el soporte para MySQL y XML; y el manejo de excepciones (try, throw, catch), acercan a PHP al nivel de los grandes lenguajes de programación web.

3.1.2.2 - Requisitos y posibilidades.

Portabilidad: PHP está preparado para trabajar sobre cualquiera de los principales sistemas operativos y entornos de desarrollo del mercado (incluso sobre distintas arquitecturas de hardware) sin componentes adicionales, aunque donde mejor trabaja es en una plataforma LAMP (Linux + Apache + MySQL + PHP).

Diversidad de ambientes: No obstante no existe soporte nativo de PHP con Visual Studio 2010, aunque existe un IDE de Visual Studio que permite aprovechar esta herramienta para programar con PHP, llamado VS.PHP. En cuanto a Frameworks, el más usado es CodeIgniter con el que mejora considerablemente la velocidad de desarrollo. Por otra parte, se puede ejecutar sobre gran cantidad de servidores de aplicaciones, incluido IIS.

PHP tiene extensiones para soportar una gran diversidad de sistemas gestores de bases de datos, incluido SQL Server. No obstante, con quien mejor interactúa es con MySQL. Si en la universidad convivieran ambos SGBD y hubiera necesidad de migrar de una base de datos a otra, se pueden usar las llamadas “capas de abstracción para acceso a datos”, que permiten acceder a distintas bases de datos utilizando el mismo código, ya que proveen funciones comunes a todas. Las más conocidas son PEAR::DB y ADOdb.

Oportunidad de aparición y reutilización: PHP es uno de los lenguajes con mayor cantidad y variedad de aplicaciones desarrolladas y se recomienda reutilizar el código ya desarrollado para no realizar la misma tarea más de una vez. De esta forma, y debido al carácter libre del lenguaje los desarrolladores contribuyen a través de su experiencia a ampliar la biblioteca de código sin ambiciones comerciales de por medio, pues a ello está sujeta la licencia. Esto haría muy recomendable la elección si la biblioteca de clases y aplicaciones implementada en la Universidad no estuviera hecha con ASP.NET, con lo que la reutilización de código en este caso no es posible y, por tanto la rentabilidad del uso de PHP para este proyecto es cuestionable.

3.1.2.3 - IDE recomendado

No existe un entorno de desarrollo específico para PHP, aunque puede funcionar perfectamente con Eclipse o, como se ha dicho antes, con Visual Studio apoyándose en un plugin.

3.1.2.4 - Servidor web recomendado

Apache: Servidor web open-source, es el más popular desde 1996, penetrando actualmente en un 50% de los servidores web en todo el mundo, en competencia directa con IIS y Sun Java System Web Server. Está siendo implementado y mantenido por la comunidad de desarrolladores bajo el amparo de la Apache Software Foundation y es multiplataforma.

La primera versión fue implementada por Robert McCool mientras que la segunda (6 de abril de 2002) aportó mayor modularización, portabilidad, multitarea en UNIX, multiplataforma, seguridad SSL y TLS, nueva API, soporte para IPv6, etc.

Es el más utilizado para PHP junto con MySQL en los populares sistemas LAMP (con Linux) y WAMP (con Windows).

3.1.2.5 - SGBD recomendado

MySQL: SGBD relacional, creado por la empresa MySQL AB, que posee el copyright del código fuente del servidor SQL y de su marca. Es open-source, con licencia bajo la GPL de la GNU. Se estableció en Suecia en 1995, a raíz de las investigaciones y pruebas de Michael Widenis con mSQL hasta desarrollar una nueva interfaz SQL. Sus limitaciones iniciales, como la integridad referencial y las transacciones eran compensadas primero con su simplicidad de uso, y paliadas posteriormente por los desarrolladores.

En las últimas versiones destaca por su velocidad y robustez, la gran cantidad de tipos de dato que soporta, la portabilidad entre sistemas (multiplataforma), su implementación multihilo, su gestión de usuarios y su seguridad. Por el contrario, se echa en falta un poco más de documentación en sus utilidades.

3.1.3 - ASP.NET

ASP.NET es una tecnología multiplataforma ideada con el objetivo de desarrollar sitios web dinámicos, servicios web XML y aplicaciones web distribuidas que aprovechen al máximo las posibilidades ofrecidas por internet. Ha sido implementado y comercializado por Microsoft. En combinación con otros lenguajes, como C# o Visual Basic.Net ha reemplazado a las Active Server Pages (ASP), buscando mayor reusabilidad y portabilidad. Para ello dispone de una gran serie de herramientas y tecnologías orientadas a facilitar el desarrollo de aplicaciones empresariales.

3.1.3.1 - .NET Framework

.NET es un marco de trabajo multilenguaje que ofrece a los desarrolladores la posibilidad de crear Aplicaciones y Servicios Web de forma sencilla. Está construido sobre el "CommonLanguageRuntime" (CLR), un entorno de ejecución que se encarga de compilar el código fuente a código nativo que, a diferencia de lo que ocurre con el código compilado en una Máquina Virtual de Java, aquí corre nativamente sin que intervenga ninguna capa de abstracción por encima del hardware.

Por otra parte, se alimenta de la "Base Class Library", una librería de clases unificada que contiene todas las clases pertenecientes al entorno .NET en general, y a ASP.NET en particular.

3.1.3.2 - Breve historia.

La tecnología ASP nace en diciembre de 1996, coincidiendo con el lanzamiento del “Service Pack 3” de Windows NT 4.0, como parte del “Internet Information Server” (IIS) 3.0.

Ese fue el primer acercamiento de Microsoft hacia la tecnología scripting en internet, como ya hacía PHP, permitiendo diseñar la página y programar la lógica de negocio mezclando HTML con bloques de código escrito en algún lenguaje de programación de Microsoft, como VBScript o JScript. Microsoft definió los ASP contextualmente como “un ambiente de aplicación abierto y gratuito que permite combinar código HTML, scripts y componentes ActiveX del servidor para implementar soluciones dinámicas y potentes para la web”.

A pesar de su reciente implantación, las Active Server Pages lograron crear un estándar en el ámbito de las páginas web dinámicas, pues resultaba más sencillo que utilizar la tecnología CGI para intercambiar información con el servidor (como se venía haciendo hasta entonces), y además ofrecía mayor rendimiento y seguridad.

No obstante, ya en 1997, tras el lanzamiento del IIS 4.0, Microsoft empezó a investigar la posibilidad de diseñar un nuevo modelo de aplicaciones web que pudiera solventar las limitaciones más comunes de ASP, a saber:

El mantenimiento de las aplicaciones Cliente/Servidor, al mezclar el código ASP con la interfaz de usuario, no resultaba sencillo y cada actualización acababa suponiendo una pérdida de tiempo al no poder detectar fácilmente el núcleo del código.

A finales de año apareció el primer prototipo, llamado XSP y desarrollado inicialmente en Java. No obstante pronto se construyó una nueva plataforma implementada en C#, sobre el CLR, permitiendo así la separación de las capas de presentación y de negocio, además de brindar un ambiente orientado a objetos y recolección de basura, entre otras características. A esta nueva tecnología se le llamó ASP+ y aunque ha supuesto una reconstrucción del lenguaje, se ha intentado proporcionar un medio sencillo de migración para programadores ASP.

La primera demostración pública de la versión beta de ASP+ fue en el año 2000 y mostró el soporte que daba para una gran cantidad de lenguajes de programación, como Cobol, los nuevos lenguajes de Microsoft Visual Basic .NET y C#, y también para Python y Perl mediante la ayuda de algunas herramientas de interoperabilidad.

Durante la segunda mitad de año, ASP+ cambió su nombre a ASP.NET, para denotar su integración en la nueva plataforma .NET y finalmente el 5 de enero de 2002 se liberó ASP.NET 1.0 como parte de la versión .NET Framework 1.0. Las características que definieron esta primera versión fueron las siguientes:

- **Páginas o web forms:** Formularios web contenidos un archivo con extensión “.aspx”. Contienen generalmente etiquetas HTML o XHTML estático, además de etiquetas que definen *Controles Web* procesados en el lado del servidor y de *Controles de Usuario*, donde se sitúa el código estático y dinámico requerido por la página.
- **Modelo Code-File:** En las páginas se permite el uso de bloques de código scripting para incorporar contenido dinámico, pero desde esta versión se recomienda utilizar un

archivo separado con extensión “.aspx.cs”, en el cual se escribe el código asociado a diferentes eventos declarados en la página, como la carga de la misma o el clic en un control.

- **Controles de Usuario:** Permiten la creación de componentes reutilizables en todas las páginas, y tienen la misma estructura que éstas. Los formularios web se almacenan en archivos con extensión “.ascx” y la lógica de negocio en los archivos con extensión “.ascx.cs”
- **Ficheros de configuración:** Global.asax para declarar y manejar eventos a nivel de aplicación y de sesión, y web.config como fichero XML principal de configuración, que contiene información que controla la carga de módulos, configuraciones de seguridad, configuraciones del estado de la sesión, opciones de compilación y el lenguaje de la aplicación. También pueden especificar cadenas de conexión a la base de datos.
- **Servicios web:** Posibilidad de crear y consumir servicios web.

La segunda versión del marco de trabajo llegó en 2005 y su gran aporte fue **ADO.NET** con sus estructuras de datos: los *DataSets* (cachés de memoria interna de datos), los *DataTables* (tablas de datos en memoria) y los *DataSources* (origen de los datos listados). También aparecieron por primera vez las **páginas maestras** o plantillas, para centralizar el contenido común a todas las páginas y ahorrar líneas de código y los archivos de configuración **SiteMap**, que definen opciones de configuración para admitir la infraestructura de configuración, almacenamiento y representación de sitios.

La versión 3.0, lanzada en el año 2006 ofrece la posibilidad de crear archivos de servicio de “Windows Communication Foundation”, una nueva plataforma de mensajería basada en etiquetas orientadas a servicios (SOA),

Un año después apareció la versión 3.5, dando soporte al lenguaje de consultas de base de datos LINQ, que permite hacer consultas SQL directas desde ASP.NET.

Finalmente, a mediados de 2010 se liberó la versión del .NET Framework 4.0 de cuyas mejoras se puede destacar las relacionadas con el SEO, ya que añade dos nuevas propiedades de la clase Page: *Metakeywords* y *Metadescription*, que permiten establecer dinámicamente los valores de los metadatos “keywords” y “description”, algo que, bien utilizado, permitirá una mejor indexación del sitio en la web.

3.1.3.3 - Requisitos y posibilidades

Portabilidad: .Net está pensado para poder realizar migraciones de aplicaciones de forma eficiente. Para asegurar un buen rendimiento, el CLR puede compilar todos los códigos al ejecutar la aplicación o cuando se instala por primera vez. Se trata de código compilado “just-in-time”, dentro de un entorno “New Generation Windows Services” (NGWS), con soporte nativo y servicios de caché. El proceso de compilación utilizará todas las características del microprocesador sin necesidad de tener que cargar diferentes binarios para diferentes plataformas.

El soporte para distintos lenguajes, como Visual Basic.NET y C#, es una ventaja para los desarrolladores, que dejan de ver acotadas sus aplicaciones web sin que esto afecte a la portabilidad, gracias al proceso de compilación. El IDE por excelencia es Visual Studio, aunque también hay otros como Visual Web Developer (sucesor de WebMatrix), Visual Basic o Visual C#.

Otro aspecto fundamental es la compatibilidad con todos los navegadores, de forma que cuando se procesa un control, éste examina el tipo de navegador que lo ejecuta y genera una página que se adecúa al mismo.

Diversidad de ambientes: ASP.NET puede funcionar tanto con un servidor Apache (aunque es necesario un módulo llamado MONO que recibe y transforma las peticiones de ASP.NET a un lenguaje conocido por Apache), como con un gestor MySQL (junto a Apache o junto a IIS), pero donde ofrece un rendimiento y una integridad inmejorables es con IIS y SQL Server, para los que ofrece soporte de forma nativa.

Oportunidad de aparición y reutilización: Una de las características fundamentales de .NET es su librería de clases, común en toda la plataforma. Esta librería otorga al desarrollador una herramienta perfecta para programar aplicaciones multiplataforma, ahorrando líneas de código. La separación de las capas de presentación de las de código gracias a los controles “web forms” también facilita la reutilización del código.

3.1.3.4 - IDE recomendado

Visual Studio: Es un IDE para sistemas operativos Windows. Da soporte nativo a diversos lenguajes de programación como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET. Utilizando extensiones se puede ampliar su soporte a muchos otros. Permite crear aplicaciones, sitios y servicios web en cualquier entorno situado bajo la plataforma .NET, con lo que se puede intercomunicar entre estaciones de trabajo, páginas web y dispositivos móviles. Tiene una licencia comercial aunque ofrece desde 2005 de forma gratuita las ediciones básicas “Express Editions”.

Visual Studio 6 fue la última versión ejecutada sobre sistemas basados en MSDOS y la base para el desarrollo del .NET Framework, que aparecería 4 años después junto a Visual Studio .NET. Este framework es una plataforma de ejecución intermedia multilenguaje, lo que permite al lenguaje de programación ser independiente de la misma. Con esta versión se introdujo C#, basado en C++ y Java, además de remodelar el resto de lenguajes que soporta. Visual Studio 2005 vino de la mano .NET 2.0, incluyendo los tipos genéricos, un diseñador de implantación para validar los proyectos y soporte para arquitecturas de 64 bits. Visual Studio 2008 apareció junto a .NET 3.5, que incluye como mejora más destacada la biblioteca ASP.NET AJAX para implementar aplicaciones más eficientes, interactivas y personalizadas. La última versión es Visual Studio 2010, acompañada por .NET Framework 4.0 que incluye herramientas de desarrollo para Windows 7.

3.1.3.5 - Servidor web recomendado

IIS: Además de servidor web es un conjunto de servicios para Windows, tales como FTP, SMTP, NNTP y HTTP/HTTPS, convirtiendo al PC en un servidor web para internet y permitiendo que los ordenadores que dispongan del servicio publiquen páginas web

localmente o en remoto. Proporciona todas las herramientas para administrar de forma sencilla un servidor web seguro. Está basado en módulos que permiten procesar varios tipos de páginas, ofreciendo nativamente los de ASP y ASP.NET, aunque se puede incluir para otros.

Fue lanzado en sus inicios como un conjunto de servicios basados en internet para Windows NT 3.51. Hasta IIS 3.0 no se incluyó la tecnología de scripting dinámico con ASP. IIS 6 fue la versión que acompañó a Windows Server 2003, permitiendo un mayor número de conexiones simultáneas y soporte para IPv6. La versión actual es IIS 7.5, para Windows Server 2008 R2.

3.1.3.6 - SGBD recomendado

MS SQL Server: Sistema basado en el modelo relacional, hace uso de los lenguajes T-SQL (principal medio de programación y administración de SQL Server) y ANSI SQL. Entre sus características destaca su soporte para transacciones, su escalabilidad, estabilidad, seguridad, su soporte a procedimientos almacenados, su potente entorno gráfico y las posibilidades de trabajar en modo cliente/servidor y de administrar información de otros servidores de datos.

Nació en 1989 y desde entonces ya se han lanzado 11 versiones, incluido SQL Server 2008 R2. Desde 1999 SQL Server ofrece una versión gratuita llamada MSDE, con el mismo motor de base de datos. En sus versiones 2005 y 2008, esta versión gratuita pasa a llamarse SQL Express Edition. Para desarrollo de aplicaciones más complejas (tres o más capas), SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, como .NET.

También posee una biblioteca de acceso a datos para clientes de MS SQL Server, llamada Cliente Nativo de SQL que implementa soporte para bases de datos en espejo, ejecución de la secuencia de datos tabular, soporte completo a todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, notificaciones de consulta, soporte ara cifrado y también la posibilidad de recibir más de un conjunto de resultados en una única sesión de bases de datos.

3.1.4 - Resumen

La siguiente tabla indica la mejor de las opciones de cada uno de los aspectos tratados para cada plataforma.

	JavaEE	PHP	ASP.NET
Sintaxis	Sintaxis Java. Muy buena	Sintaxis por depurar.	Sintaxis C# y Visual Basic. Muy buena.
Velocidad de desarrollo	La más lenta, mejora con Frameworks.	Velocidad media. Aumenta con Frameworks	Muy rápido
Plataforma	Trabaja bien en gran variedad de plataformas.	Trabaja mejor en LAMP	.NET en Windows
Base de datos	La que más se adecúa es Oracle	La que más se adecúa es MySQL	La que más se adecúa es SQL Server
IDE	El favorito es Eclipse	Ninguno en especial.	Visual Studio
Soporte Orientado a Objetos	Completo.	Está por debajo, ha mejorado con PHP5	Completo
Servidor de Aplicaciones	Glassfish, JBoss, Tomcat	Apache	IIS

Frameworks	Struts, JSF, Spring	CodeIgniter	.NET
-------------------	---------------------	-------------	------

En definitiva, ASP.NET es una plataforma a la altura de las otras dos más populares en el mercado, con las máximas garantías de seguridad, escalabilidad y persistencia si se usa con el resto de herramientas para las que da soporte nativo, por lo que no parece ser necesario recurrir a otras tecnologías disponiendo ya de las licencias necesarias para el desarrollo de la aplicación con esta sistema.

No obstante, queda por delante la realización de un estudio sobre los CMS que hay en el mercado, pues encontrar un gestor de contenidos adecuado a la aplicación y de fácil integración con los sistemas de la Universidad CEU Cardenal Herrera podría suponer una reducción de trabajo considerable.

A continuación se presentarán los CMS más populares, sus características, la plataforma sobre la que están desarrollados y las posibilidades que ofrecen para facilitar el desarrollo del sitio web y la aplicación de gestión propuestos en este proyecto.

3.2 - Sistemas de Gestión de Contenidos (CMS)

Las plataformas de gestión de contenidos están ganando peso en el mundo de las empresas. Cada vez se usan con más frecuencia en el diseño de webs corporativas, blogs, portales de contenidos, intranets o incluso redes sociales, ya que permiten crear estructuras para la creación y administración de los contenidos con distintos privilegios en función del rol que se asigne al usuario. No es necesario implementarlos, porque ya están hechos y se encuentran a disposición de los usuarios.

Se trata interfaces que controlan una o varias bases de datos donde se aloja el contenido del sitio web, permitiendo manejar de manera independiente el contenido y el diseño. De esta forma es posible manejar el contenido y dar un diseño distinto al sitio web sin necesidad de dar formato al contenido de nuevo, además de permitir gestionar cada publicación en el sitio a varios editores.

Existen varias plataformas que sobresalen en popularidad y en funcionalidad. A continuación se van a presentar las características de los más reconocidos para posteriormente intentar hacer una valoración coherente de las ventajas e inconvenientes de utilizarlos y de implantarlos en los sistemas de la Universidad Cardenal Herrera.

3.2.1 - Plataformas basadas en ASP.NET: DotNetNuke

Parece coherente comenzar este repaso con una plataforma que haya sido desarrollada con la tecnología utilizada en la universidad, buscando así la mayor compatibilidad posible. En ese sentido, uno de los CMS más populares es DotNetNuke.

DotNetNuke es una CMS open-source y de distribución libre, desarrollado en Visual Basic.NET, con capacidad para administrar sitios, aplicaciones web e intranets. Es también un marco de trabajo potente para implementar aplicaciones web hechas en ASP.NET.

Está orientado principalmente al desarrollo de portales, compuesto por una parte pública, a la que accederán los visitantes y una parte privada para que los administradores puedan editar los contenidos del portal.

Se trata de un intento de extender el uso de software libre para esta plataforma. De hecho, ha sido implementado por una comunidad de desarrolladores con el propósito de popularizar las herramientas que ofrece.

3.2.1.1 - Características básicas

Cabe decir en primer lugar que el software es gratuito, por lo que se puede usar de forma libre para cualquier tipo de proyecto. Esto incluye la modificación, redistribución, venta o prestación de soporte técnico. El único requisito es hacer mención del origen del software en la propia página html. Otras características importantes son:

- Software open source: Su código fuente está disponible y posee una licencia estándar BSD. Gracias a esto, las empresas o personas que quieran utilizar el código de la aplicación con cualquier propósito pueden hacerlo con la condición de que indiquen que su código está implementado sobre un software desarrollado por la comunidad de proyecto original DotNetNuke.
- Intuitivo y amigable para el usuario: DNN ha sido diseñado para simplificar la gestión de todos los elementos del sitio. Posee asistentes, iconos de ayuda y una trabajada interfaz de usuario que permite administrar sus funciones con facilidad.
- Robusto y potente: DNN permite alojar una gran cantidad de portales en una única instalación. Presenta las opciones administrativas divididas en dos niveles, la primera para el alojamiento y la segunda individual a nivel de portal. Los administradores pueden gestionar varios sitios a la vez, cada uno con su propio diseño e identidad.
- Numerosas herramientas: DNN incorpora una serie de herramientas que le confieren capacidad para alojar páginas web, diseñarlas, generar y editar contenido, administrar las opciones de seguridad o gestionar los roles de los usuarios.
- Soporte técnico: Detrás de DotNetNuke se encuentra un elenco de programadores y una comunidad internacional dedicados al proyecto. Se puede acceder a este soporte a través de grupos de usuarios, foros de discusión, portales de recursos, y una red de empresas especializadas en DNN®.
- Instalación sencilla: En pocos minutos se puede llevar a cabo la instalación y puesta en funcionamiento. No hay más que descargar el ejecutable de la página web oficial de DotNetNuke y seguir el asistente de instalación. Muchas empresas de hosting ofrecen la instalación gratuita de DotNetNuke con sus ofertas.
- Función multi-lenguaje: DNN ofrece funciones de traducción a cualquier idioma.
- Innovador: DNN brinda al desarrollador con un buen mecanismo para aprender técnicas de implementación. Creación y distribución de módulos, métodos de depuración, etc. Para ello se utiliza la tecnología ASP.NET, Visual Web

Developer(VWD), Visual Studio y SQL Server correspondientes a la versión de DotNetNuke que se esté utilizando.

- Extensible: Además de crear sistemas de tanta complejidad como se desee para gestionar contenidos con sus características integradas, permite a los administradores de hosting añadir funcionalidad a la aplicación con programas de terceros, añadidos y herramientas personalizadas. Se pueden encontrar módulos o skins, tanto gratuitos como de pago, con relativa facilidad, y se pueden desarrollar otros nuevos.
- Reconocido: DNN tiene un cierto reconocimiento y respeto por parte de la comunidad de código abierto. Sus más de 500.000 usuarios registrados y un equipo de desarrolladores permiten la evolución del software a través de la participación, pruebas en el mundo real y sugerencias del usuario final.

De entre las muchas funcionalidades que se puede obtener de esta plataforma, cabe destacar las siguientes:

- Permite crear gran variedad de portales mediante un mismo código y una única base de datos, es decir, que posibilita la creación de varios sitios web usando un mismo espacio de hosting que soporte ASP.NET.
- Permite registrar usuarios y personalizar de forma completa el portal en función del tipo de usuario.
- Permite crear servicios o contenidos de acceso restringido, con acceso directo a métodos de pago para los usuarios que quieran utilizar dichos servicios.
- Permite acceso en modo administrador para gestionar contenidos y servicios, así como visualizar las estadísticas de uso del sitio.
- Posee un sistema de gestión de banners integrado.
- Permite acceso de superusuario que le otorga privilegios para la creación de nuevos portales.
- Permite la creación de skins por el administrador para personalizar el portal.
- Permite trabajar con distintas bases de datos. Para algunas bases de datos necesita un módulo adicional de soporte, pero para este proyecto no acarrearía problema ya que las bases de datos de la universidad es SQL Server.
- Permite gestionar diversos tipos de contenidos, como noticias, contacto, foros de discusión, documentos, eventos, etc. Puede incluso ampliarse con una serie de módulos disponibles.

3.2.1.2 - Módulos.

Los módulos son herramientas del programa que otorgan soporte a funcionalidades específicas para los usuarios del portal. La última versión disponible al escribir este informe disponía de 24 módulos para realizar tareas tan dispares como gestión de usuarios, calendarios

de eventos, banners, noticias, etc. Existen también módulos gratuitos, ofrecidos por portales relacionados con DotNetNuke y otros de pago ofrecidos por empresas y desarrolladores.

Los módulos se pueden instalar, desinstalar y configurar cuando se desee, dotando al portal con las funcionalidades que se puedan necesitar. Los módulos básicos disponibles en la versión actual son los siguientes:

- Administración de cuentas
- Noticias o novedades
- Administración de banners
- Contacto, que brinda la posibilidad de enviar mails a personas o grupos de personas
- Foros de discusión, posibilitando la configuración de los temas
- Administración de documentos, posibilitando su descarga y visualización
- Calendario de eventos
- FAQs
- Links
- Búsquedas
- Encuestas

Se pueden encontrar otros módulos más específicos, como un servicio de información del tiempo, un servicio WhoIs, para hacer búsquedas de registro de dominios, etc.

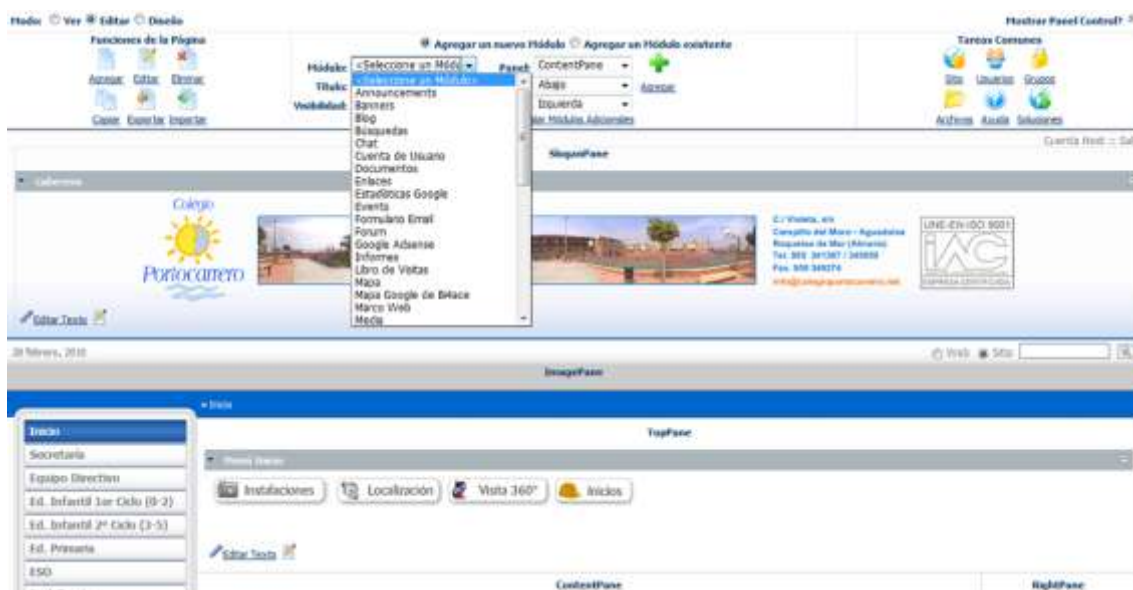


Figura 3.1. Panel de control de DNN

3.2.1.3 - Descarga e instalación

En <http://www.dotnetnuke.com> se puede encontrar información adicional y las instrucciones de instalación de DotNetNuke. Se puede encontrar a disposición del usuario una guía de instalación del software que explica paso a paso el proceso de instalación, así como guías de usuario y de administrador del portal, etc. Para efectuar la descarga hay que registrarse e iniciar sesión, encontrando a continuación ficheros de descarga para distintas versiones.

No es necesario tener IIS para instalar DotNetNuke, por lo que usuario con versiones Home del entorno Windows también podrían ejecutarlo.

3.2.1.4 - Limitaciones

A pesar de las múltiples ventajas que ofrece la plataforma, se han detectado también dos defectos importantes, uno de ámbito general y otra particular para este proyecto.

Una de las limitaciones está relacionada con el editor de contenidos que ofrece DotNetNuke. Pese a ser bastante completo, recoge múltiples caracteres extraños cuando se pega texto desde un fichero de MS Word, pdf o un simple mensaje de correo electrónico que ralentizan mucho la edición de texto, y la edición rápida y sencilla de contenidos es uno de los requisitos principales de la aplicación.

Pero la más importante es una que afecta al sistema particular en el que se va a alojar la aplicación. Uno de los requisitos que se han impuesto desde el departamento de informática es que las aplicaciones deben desplegarse pre-compiladas en el servidor. Al pre-compilar el proyecto, lo que se hace es eliminar los ficheros que contienen la lógica de negocio y almacenar el código en archivos “.dll” que son llamados desde la cabecera de las páginas web. Esto ayuda a minimizar el consumo y maximizar la seguridad de la aplicación. Desgraciadamente DotNetNuke no permite desplegar las aplicaciones desarrolladas de esta manera, siendo ésta una de las causas principales por las que no ha sido utilizado por la universidad para el desarrollo de aplicaciones en su sistema.

3.2.2 - Umbraco

Umbraco es un CMS de código abierto que es usado principalmente para la publicación de contenidos en la WWW e intranets. Ha sido implementado con C#, el otro lenguaje de Microsoft, sobre cuya infraestructura funciona este sistema. Esto no impide que se puedan usar otros lenguajes de la plataforma, como Visual Basic.NET o IronPython. También trabaja con XSLT, un potente lenguaje basado en XML y usado para la transformación de documentos XML. Umbraco está diseñado para ser utilizado como herramienta de implementación de software personalizado, integrando diversas tecnologías ya desarrolladas en vez de utilizar lenguajes implementados por ellos mismos. Por ejemplo, Umbraco da soporte a las páginas maestras de ASP.NET y al ya comentado XSLT, en lugar de diseñar sus propias plantillas.

El motor de código open-source es distribuido bajo MIT License y la interfaz de usuario bajo licencia Umbraco.

Fue implementado por Niels Hartvig en el año 2000 y se liberó como código abierto en 2004. Se trata del principal competidor de DotNetNuke dentro del ámbito de CMS open-source desarrollados con tecnología .NET. Funciona principalmente sobre Microsoft IIS y SQL Server,

aunque desde que el programador Ruben Verbough presentara la capa de abstracción de datos en 2008, Umbraco soporta otros SGBD como MySQL y VistaDB. En febrero de 2010 se lanzó la versión 4.1, que reestructuró el código fuente de varios componentes del framework, como el árbol de control en la interfaz de usuario para mejorar la productividad y la experiencia de usuario, además de partes de la capa de acceso a datos para reducir el número de llamadas.

A finales de 2011 apareció la primera versión beta de Umbraco 5, totalmente basada en el patrón Model View Controller (MVC), que viene acompañado del motor de vistas Razor. Este motor dota a la plataforma de la inteligencia necesaria para detectar en una vista (antiguo webform) cuándo se está haciendo uso de código de servidor sin necesidad de especificarlo mediante etiquetas.

Otras mejoras implementadas son la simplificación del proceso de instalación del software tanto en local como en remoto (uno de sus puntos débiles en versiones anteriores) y la relajación de los requisitos de acceso (anteriormente los permisos de acceso eran limitados) a bases de datos. Esta versión pretende corregir además muchos de los bugs que han presentado las versiones anteriores, la asignatura pendiente de este CMS.

Por otra parte, el equipo de soporte de Umbraco comunicó que va a dejar de dar soporte a XSLT en su última versión, supuestamente por la complejidad y obsolescencia de dicha tecnología, pese a las posibilidades que presenta y a la cada vez más notoria presencia de XML en el ámbito del desarrollo web. Por si acaso la versión 4 continuará desplegándose.

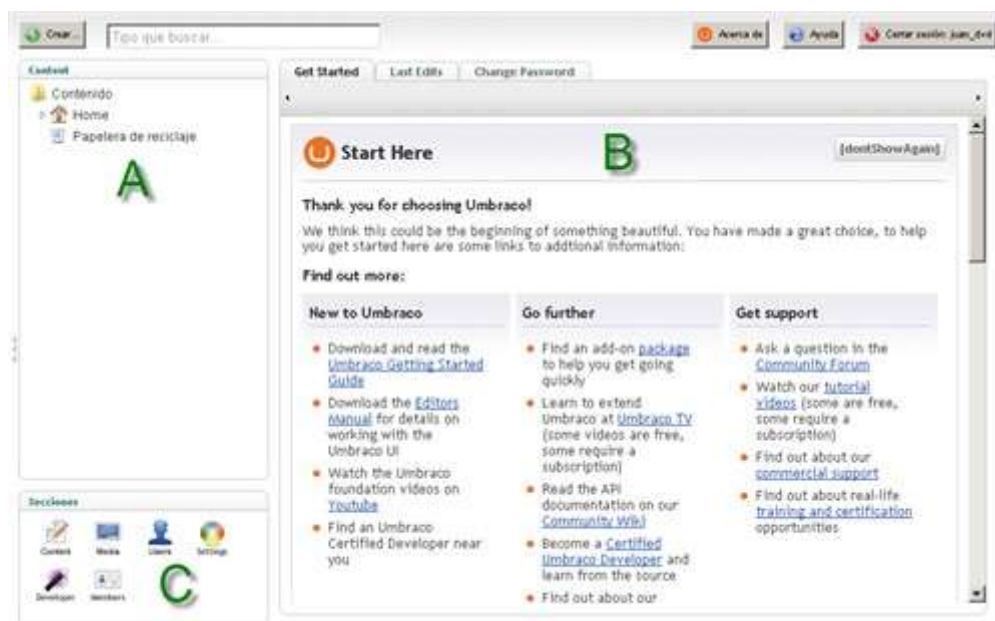


Figura 3.2 Panel principal de gestión de Umbraco.

3.2.2.1 - Características básicas

- Umbraco se integra por completo con el framework .NET. Se permite la utilización tanto de los controles web de .NET como de controles específicos de usuario, así como el uso del IDE Visual Studio para el diseño y desarrollo de las páginas.

- Ofrece soporte para Ajax, lo que permite el uso de frameworks y librerías como JQuery, Mootools, Prototype, YUI, así como otras plataformas Javascript, etc.
- Sistema basado en plantillas, como casi todos los CMS de popularidad, que permite elegir y gestionar los diseños proporcionados, así como editarlos o rediseñarlos a través de un editor de plantillas. Además se puede integrar con MS Word 2007 para la edición y creación de nuevas páginas e incluso para añadir contenido multimedia en las plantillas.
- Documentación y tutoriales, proporcionados desde la web oficial de la plataforma <http://www.umbraco.com>, donde se puede encontrar una cierta, aunque algo escasa, documentación de la API, así como descargas manuales y vídeo tutoriales divididos en dos categorías:
 - Creación de sitios, que explican cómo gestionar plantillas, gestionar usuarios y miembros, crear un sitemap, una lista o una sección de noticias, utilizar flash en el sitio o crear nuevas secciones.
 - Programación mediante controles web, menús contextuales, uso de Ajax, etc.

3.2.2.2 - Limitaciones

Navegando por diversos foros se puede detectar la frustración que Umbraco está generando en los usuarios que se embarcan en la aventura de programar con esta plataforma. El nivel de dificultad para utilizar esta herramienta es superior al de otros CMS, como el comentado DotNetNuke. Se echa en falta además una documentación más detallada y enfocada en la orientación al usuario menos experimentado.

Por otra parte, los bugs que se han ido detectando en cada una de las versiones lanzadas hasta la fecha, incluida esta versión beta de Umbraco 5 (<http://our.umbraco.org/forum/ourumb-dev-forum/bugs>) hacen minar la confianza de los usuarios en esta herramienta, de cara a al desarrollo de proyectos importantes.

Estos problemas empañan en parte el buen hacer de una herramienta que parecía hecha a medida para este proyecto, y que destaca por encima de otros CMS open-source gracias a su flexibilidad para programar y a su escalabilidad. Habrá que esperar al lanzamiento de la versión 5 oficial para ver si se gana lo suficiente en robustez y accesibilidad como para convertirse en el mejor complemento para ASP.NET.

3.2.3 - Otros CMS basados en .NET.

El tercer CMS en discordia de código abierto basado en la tecnología de Microsoft es WebMatrix que, al contrario que Umbraco, está orientado al desarrollador más inexperto destacando por la sencillez de su uso, pero que evidentemente no sobresale por su escalabilidad. Probablemente, para conseguir aunar robustez, accesibilidad y escalabilidad en los CMS basados en .NET, haya que recurrir a los sistemas comerciales, como Sitefinity, por el que se pagan licencias de hasta 20000 dólares por año. Así pues, de momento, con las licencias para ASP.NET nativo se antoja suficiente. No obstante continúa el repaso a los CMS basados en las otras grandes plataformas.

3.2.4 - Plataformas basadas en Java. Liferay

Liferay es un portal de gestión de contenidos open-source escrito en Java y basado en struts. Nació en el año 2000 en un intento por buscar soluciones para las organizaciones sin ánimo de lucro.

3.2.4.1 - Características básicas

Liferay portal es una fuente abierta rica en funciones que permite crear portales web de forma muy fácil y rápida, aliviando así tanto el tiempo de desarrollo como los costes de producción. Desde el punto de vista del usuario es un sistema muy completo ya que incluye más de 60 portlets que contienen ya la funcionalidad para la realización de las tareas típicas de un portal moderno a nivel empresarial. Un portlet es un componente modular que se visualiza en el portal web, produciendo fragmentos de código de marcado que se agregan en una página del portal.

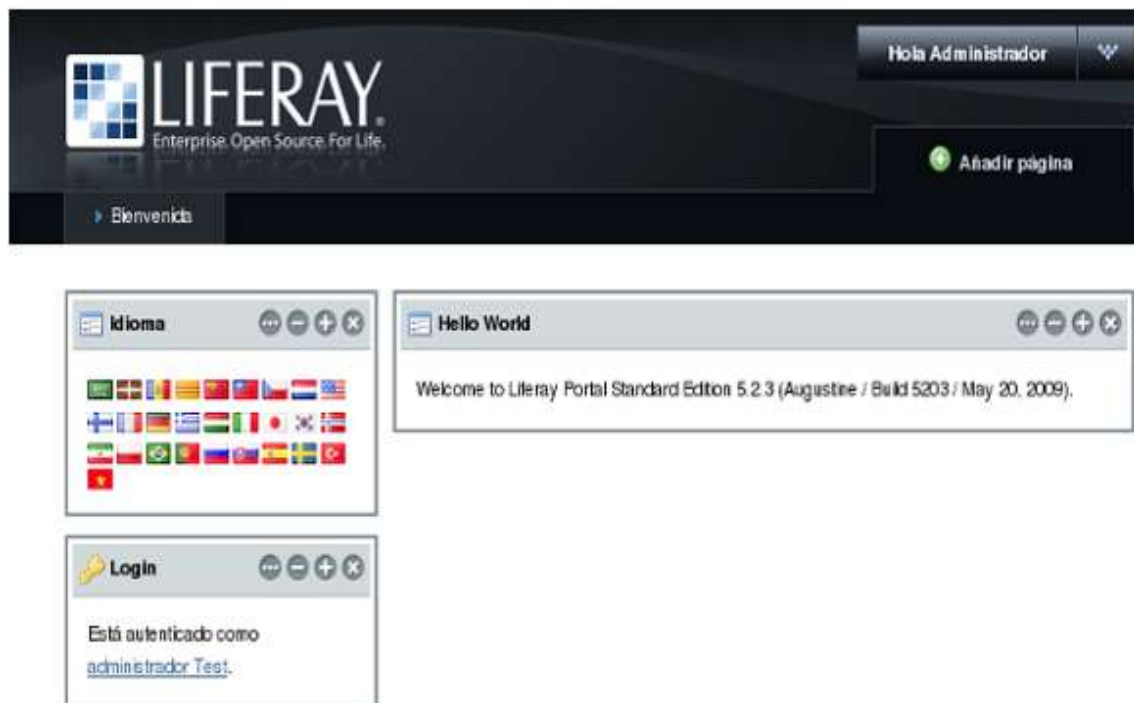


Figura 3.3. Administración de un portal con Liferay

Con Liferay es posible configurar un sitio web en muy poco tiempo sin necesidad de escribir código, debido a la gran cantidad de portlets disponibles para ser usados. Lo que produce mayor interés desde el punto de vista de la implementación de funciones personalizadas, es que Liferay además es una plataforma de desarrollo completa, un framework para el desarrollo del portal en Java. Esta característica, menos notoria para el usuario final, lo convierte en una buena herramienta para el desarrollo de portales y aplicaciones web de entre las disponibles actualmente en el ámbito del código abierto. Se trata de un sistema de gestión de contenidos, un agregado de aplicación y una herramienta de colaboración de gran alcance.

Así pues, Liferay es una plataforma para el desarrollo, la integración y la colaboración basada en los fundamentos de la Arquitectura Orientada a Servicios (SOA). Tiene todas las cualidades que se necesitan para crear un portal web, como publicación web, gestión de contenidos,

colaboración empresarial y creación de redes sociales, construido totalmente sobre una plataforma open-source en consonancia con los estándares abiertos y adecuado a la situación tecnológica actual.

Las características más destacadas en su versión 6 están resumidas a continuación:

- **Herramientas listas para ser usadas:** Se distribuye con gran cantidad de funcionalidades, cubriendo así las necesidades fundamentales de un portal web moderno de clase empresarial. Más de 60 portlets que ofrecen diversas funcionalidades para gestionar contenidos, redes sociales y páginas web en general.
- **Administración de usuarios y roles:** Tiene una gestión de usuarios, grupos, roles y permisos de gran potencia y flexibilidad, capaz de llegar a manipular la autorización a nivel de contenido o de elemento de portal.
- **Autenticación mediante SSO:** Puede integrarse con otros sistemas de gestión de identidad, a saber, OpenID, OpenSSO o CAS.
- **Soporte para varios idiomas:** Se puede traducir a más de veinte idiomas, proporcionando una traducción completa de cada elemento del portal.
- **Gestión de comunidades:** Permite administrar de forma flexible y organizada las distintas comunidades dentro de un mismo portal.
- **Arquitectura Orientada a Servicios (SOA):** Liferay es una auténtica plataforma orientada al desarrollo y la integración estructurada siguiendo las bases de un paradigma de arquitectura flexible y eficiente.
- **Gestión de contenidos:** Las herramientas que Liferay pone a disposición del usuario le permiten gestionar de forma legal y unificada cualquier tipo de contenido, sean artículos, documentos o imágenes. Asimismo, se permite tanto gestionar versiones para cada contenido como el flujo de trabajo para su redacción, aprobación y publicación. Además ofrece la posibilidad de administrar etiquetas y taxonomías así como de introducir buscadores de contenidos filtrados por palabras. Tiene también trabajada la indexación en la web y está optimizado para optimizar su posicionamiento en buscadores (SEO). En definitiva, una gran cantidad de herramientas pensadas para conseguir una gestión eficaz de todos los tipos de contenido.
- **Colaboración:** En Liferay también se han desarrollado todas las funcionalidades de colaboración y redes sociales, listas para ser usadas. Ofrece servicio de blogs, Wiki, foros, Chat, RSS, calendarios compartidos, e incluso encuestas, ayudando a que el portal sea una herramienta interesante en el ámbito de la web 2.0.

3.2.4.2 - Limitaciones:

El despliegue de Liferay sobre un servidor de aplicaciones Java y una conexión a una base de datos Oracle o MySQL está completamente probado y los resultados son muy buenos. Sin embargo, son ya conocidos los requisitos del sistema existente en la Universidad CEU Cardenal Herrera, por lo que hay que investigar el funcionamiento de esta plataforma sobre IIS 6 y SQL Server 2008.

Tras hacer las preguntas pertinentes en el foro oficial de discusión sobre Liferay <http://www.liferay.com/es/community/forums> la respuesta obtenida no fue la más optimista. Para conectar con SQL Server únicamente hay que modificar los parámetros de la cadena de conexión en el fichero de configuración, pero para hacerlo compatible con IIS 6, es necesario instalar y configurar previamente un contenedor de Servlets como Tomcat.

Aunque esta solución ha funcionado, los resultados están lejos de ser óptimos. Se han detectado problemas de rendimiento, el tiempo de carga de los portlets era alto y la navegación y uso de las aplicaciones se hacía inasumible cuando aumentaba el número de conexiones simultáneas. Así pues, parece que el uso de Liferay en este proyecto no parece la mejor solución posible.

3.2.5 - dotCMS

Se ha visto que Liferay proporciona muchas herramientas a nivel de desarrollo y está orientado a todo tipo de plataformas, sin embargo su especialidad es la creación de portales.

Una alternativa a tener en cuenta puede ser dotCMS, un sistema de gestión de contenidos web de código abierto para la creación de sitios web y de aplicaciones de gestión de sus contenidos que formó parte del desarrollo de Liferay hasta su separación en 2005. Este alejamiento se debió a la búsqueda de una mayor especialización en el ámbito de la web (se le conoce como un Web CMS).

dotCMS apareció inicialmente como una alternativa Java a los WCMS basados en PHP que dominaban el mercado y apuntó directamente a las aplicaciones empresariales. Es el resultado de alrededor de 5 años de desarrollo por parte de dotMarketing Inc. Que también desarrolló y lanzó la herramienta de gestión de proyectos open-source, llamada dotProject. El padre de ambas plataformas fue William Ezell, quien desarrolló dotCMS basándose en la tecnología de Liferay para cumplir con la especificación de portlets de Java JSR-168.

Sin embargo, a partir de la versión 3.2.2 se bifurcó de Liferay apareciendo en su versión 1.0 como una plataforma independiente de código abierto, y cambiando su contenido basado en portlets por un contenido basado en estructuras en su versión 1.2, lanzada en 2006. Hasta convertirse en un WCMS a la altura de los líderes ha tenido que ir incorporando algunas características básicas para este tipo de plataformas. Así, en 2007, con el lanzamiento de la versión 1.5 se mejora la interfaz de usuario, añadiendo el concepto de permisos. Con la versión 1.9 se introduce la arquitectura de plugins, ofreciendo una amplia gama de puntos de extensión y un instalador en Windows para acelerar el proceso de instalación, algo precario hasta octubre de 2009.

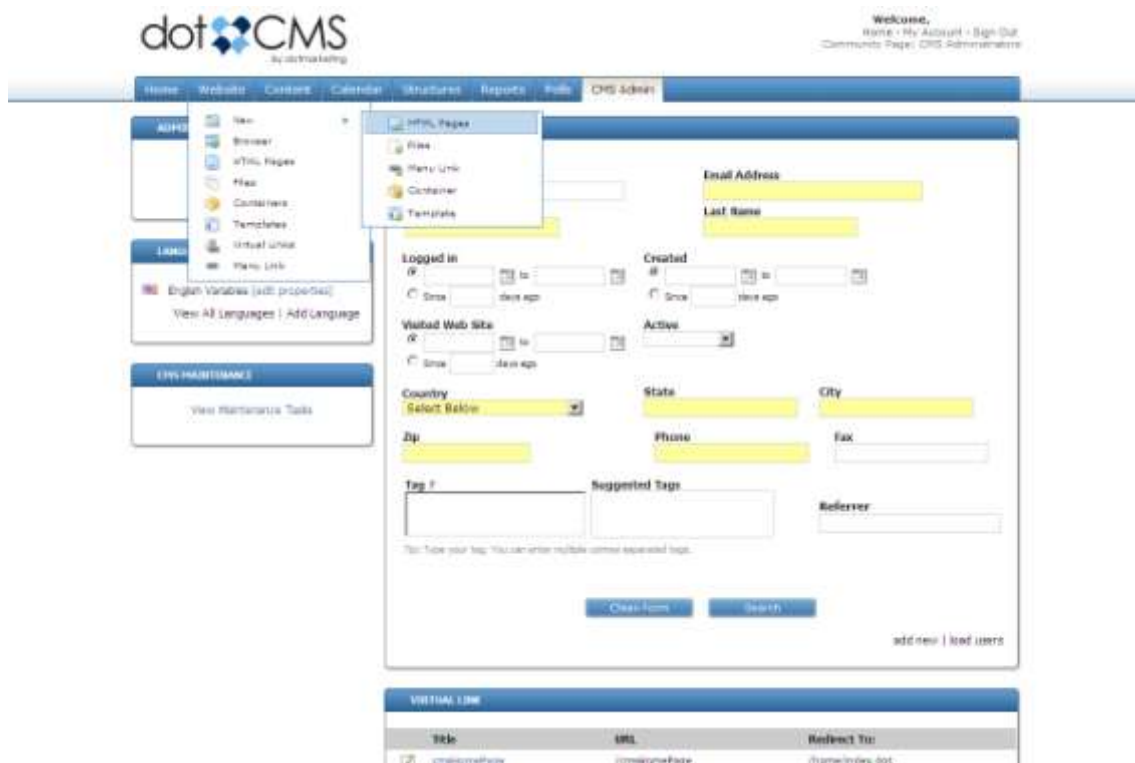


Figura 3.4. Panel de control de dotCMS.

3.2.5.1 - Características básicas

Se pueden enumerar otras características relativamente comunes entre los WCMS y presentes en dotCMS en las últimas versiones, como son el almacenamiento de páginas en caché durante un periodo de tiempo especificado por el administrador, el uso de plantillas o una API propia. Además, dotCMS ofrece una serie de módulos y funcionalidades, como un completo editor de contenidos que permite su edición en línea, visores de imágenes, reproductor de mp3, redifusión web mediante RSS, informes de estadísticas de uso del sitio y tendencias clave, descripciones SEO extendidas y uso de palabras clave para un control más fino de los metadatos relevantes para el SEO, calendarios con AJAX, listados de noticias, blogs, foros, seguimiento y etiquetado de usuarios, motor de búsqueda incorporado y traducción a otros lenguajes.

En cuanto a las características que dotCMS potencia más respecto a Liferay se presenta un resumen a continuación:

- Soporte para alojamiento virtual, permitiendo varios dominios sobre una misma dirección IP.
- WebDav, para crear, cambiar y mover documentos en el servidor web.
- Contenido estructurado: dotCMS define estructuras de datos indizadas, contenidas en portlets, con campos definidos por el usuario, lo que permite personalizar más el CMS
- Clustering, para potenciar el rendimiento y la escalabilidad.
- Posibilidad de ejecución en diversos SGBD, como PostgreSQL, MySQL, MSSQL y Oracle.

- Soporte y asistencia para servidores de aplicaciones JBoss, WebLogic, Glassfish o Websphere.

3.2.5.2 - Limitaciones

Aunque se trate de una plataforma que pretende abarcar un buen número de tecnologías para el desarrollo web, con dotCMS sigue sin ser posible la integración de las aplicaciones y de un sitio web desarrollado con ASP.NET, pues no hay que olvidar que está basado en Java. Por tanto, sólo queda plantearse la posibilidad de usar dotCMS de forma individual para este proyecto e integrarlo en el sistema.

En cuanto a las posibilidades de funcionamiento en armonía con el resto de aplicaciones en un mismo servidor, se puede afirmar que la instalación de dotCMS ha sido probada con IIS 6 y SQL Server en convivencia con .NET, pero desde el foro de dotCMS se desaconseja aludiendo al bajo rendimiento obtenido. <http://dotcms.com/forum/index.dot>

3.2.6 - Otros CMS basados en Java

En esta memoria se ha pretendido dar una visión de las posibilidades que ofrece Java como tecnología para el desarrollo de CMS a través de dos de sus plataformas más representativas. Existen otros, igual de válidos y tanto o más populares que los presentados aquí. Entre los orientados a aplicaciones empresariales funcionales y escalables destaca Alfresco, mientras que si lo que se busca es sencillez de edición y administración se puede elegir Magnolia, entre otros. En cualquier caso, las limitaciones de los CMS basados en Java para este caso concreto son evidentes, y obligan a seguir investigando a continuación en la plataforma más popular, PHP.

3.2.7 - Plataformas basadas en PHP. Wordpress

Wordpress es un CMS pensado para desarrollar blogs (sitios web con actualizaciones periódicas). Se ha implementado con PHP y MySQL, bajo licencia GPL, lo que permite su libre distribución y la modificación del código. Su padre es Matt Mullenweg, quien lo desarrolló a partir del extinto b2/cafeblog hasta convertirlo en uno de los CMS más populares en el ámbito de la creación de blogs. Las causas de su buena aceptación y crecimiento son la facilidad de uso y sus propiedades como gestor de contenidos.

La empresa responsable de Wordpress, llamada Automattic, posee un servicio de hosting de blogs gratuito que usa su software llamado Wordpress.com. Pese a que empezó como una plataforma básica de administración de blogs, su creciente funcionalidad para gestionar webs fuera del orden cronológico habitual de los blogs le ha permitido convertirse en un completo sistema de gestión de contenidos.

Aunque buena parte del proyecto ha sido implementado por su comunidad de desarrolladores, aún se asocia a Automattic, cuyos empleados son algunos de los mayores contribuyentes de Wordpress. Gracias a ellos y a los probadores, que dedican su tiempo para testear las versiones de prueba (beta) y las versiones candidatas (release), se puede examinar los errores y hacer un reporte al Sistema de seguimiento de errores.

3.2.7.1 - Características:

Desde su nacimiento, Wordpress se ha centrado en potenciar la personalización, la elegancia y la arquitectura de las publicaciones, prestando especial interés a la estética, los estándares web y la usabilidad.

Por defecto se ofrece instalar una única bitácora o weblog por instalación, pero también permite, sin necesidad de hacer uso de extensiones, crear una red de blogs que enlacen con una o más bases de datos. La versión de Wordpress más popular que permite la instalación de una Red Multisitio es Wordpress MU.

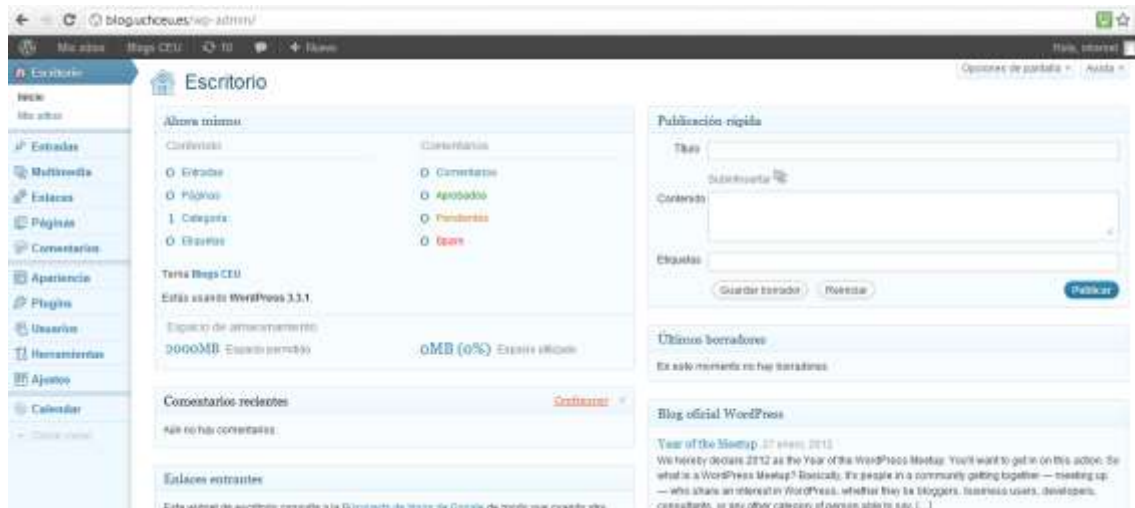


Figura 3.5. Gestión de una Red Multisitio con Wordpress MU.

Wordpress ofrece, por defecto, publicaciones ordenadas por fecha en una página dinámica, además de páginas estáticas. Su estructura y diseño visual depende de la plantilla escogida, cuyo diseño y contenido, separado en XHTML y CSS, respeta siempre las recomendaciones del W3C, aunque permite ser editado. La gestión y ejecución es tarea del sistema de administración con los plugins y los widgets que usan las plantillas, aunque un administrador también se puede editar a mano, tanto los ficheros de código PHP, como las tablas de la base de datos MySQL.

Funcionalidades:

- Sencillo de instalar, actualizar y personalizar.
- Permite la actualización automática del sistema desde la versión 2.7.
- Admite varios autores o usuarios, con diferentes roles y perfiles que definen los privilegios, desde la versión 2.0.
- Red multisite desde la versión 1.6.
- Posibilidad de creación de páginas estáticas a partir de la versión 1.5.
- Capacidad de ordenar publicaciones y páginas estáticas en categorías, subcategorías y etiquetas.

- Proporciona la posibilidad de elegir entre cinco estados diferentes para una entrada: Publicado, Borrador, Esperando Revisión (desde versión 2.3), Privado (para usuarios registrados), y protegido con contraseña.
- Proporciona un editor de contenidos WYSIWYG “What Yoy See Is What You Get” optimizado desde la versión 2.0.
- Permite publicar mediante mensajes de correo.
- Interesante sistema de importación de contenido desde Blogger, Blogware, Dotclear, Greymatter, Livejournal, Movable Type y Typepad, Textpattern y desde cualquier fuente con formato RSS. Se está estudiando la implementación de mejoras para importar desde pMachine y Nucleus, además de la importación a través de scripts o directamente desde base de datos.
- Guarda automáticamente (mediante temporizador) los artículos en estado Borrador, desde la versión 2.2.
- Registra comentarios y herramientas de comunicación entre blogs, mediante Trackback, pingBack, etc.
- Acepta enlaces permanentes de fácil memorización gracias a la función mod_rewrite.
- Ofrece sindicación o redifusión web de contenido de los artículos mediante RDF, RSS, RSS 2.0 y Atom 1.0.
- Ofrece los medios RSS 2.0 y Atom 1.0 para syndicar el contenido de las discusiones.
- Gestión y distribución de enlaces.
- Permite subir y gestionar archivos adjuntos y archivos multimedia.
- Admite plugins, desde la versión 1.6, plantillas y widgets para las plantillas.
- Ofrece búsqueda integrada, así como búsqueda de entradas y páginas estáticas y Widget de búsqueda integrada de Google desde la versión 2.5.
- Integra un sistema de foros llamado bbPress, de la misma compañía que se integra automáticamente y permite también la integración con el foro Vanilla de Lussumo factible hasta la versión 2.1.

Multiblogging

La posibilidad de utilizar Wordpress MU para crear varios weblogs en una única instalación resultaría atractiva para este proyecto pensando en la integración con los futuros sitios web de otros servicios, departamentos o facultades de la universidad. La cesión de un servidor que albergue una instalación de MySQL + PHP + Wordpress podría resultar rentable si se puede controlar una importante cantidad de weblogs desde un único entorno de administración.

Plantillas

Los “Themes” establecen el diseño y la estructura de los sitios web en Wordpress. Existe una importante comunidad de desarrolladores de plantillas, tanto profesional como de usuarios, que se dedica a diseñar plantillas y a comprobar y dar su aprobación a las mismas para permitir que sean listadas para poder utilizarlas. Aunque Wordpress apuesta por seguir las directrices del W3C y sus plantillas originales así las cumplen, se permite la completa personalización una web a partir de las mismas, por lo que se le podría dar el aspecto deseado a los weblogs, ya sea para integrarlos a nivel de diseño en la página web de la Universidad o para hacer un sitio web a medida de cada departamento.

Widgets

Para modificar de forma sencilla y funcional la estructura que una plantilla ha conferido a un blog, desde la versión 2.2 se puede hacer uso de los widgets, pequeños programas que confieren a la página información visual allí donde son situados. Tiene sus limitaciones ya que no todas las plantillas soportan cualquier widget.

Plugins

Hay una creciente cantidad de plugins que potencian las funcionalidades de la instalación de Wordpress haciéndola mucho más flexible hasta convertir un simple blog en una web de propósito general.

3.2.7.2 - Limitaciones

Cabe comentar en primer lugar las vulnerabilidades que ha sufrido Wordpress en los últimos años. Desde que en enero de 2007 muchos blogs que usaban esta plataforma fueran víctimas de ataques con exploits, los rumores sobre la ausencia de seguridad de los sitios creados con Wordpress han sido frecuentes (back doors, ataques por inyección sql, etc). Un estudio llegó a reflejar, cuando aún operaba la versión 2.1.1, que el 98% de los sitios hechos con Wordpress eran vulnerables a ataques. Con versiones posteriores ese alarmismo generado se ha relajado bastante.

Sin embargo la limitación principal para el uso de Wordpress en la Universidad CEU Cardenal Herrera, y por tanto también para este proyecto, es la versión del IIS que opera en los servidores del centro. Para trabajar con Wordpress en un servidor propio es necesario tener instalados los paquetes de MySQL y PHP acordes a la versión de Wordpress utilizada, operando sobre un servidor Apache. Sin embargo, en la universidad todos los servidores son Windows Server 2003, administrados por IIS 6. Desde que apareció IIS 7, las instalaciones de Wordpress son compatibles con el sistema, pero para versiones anteriores IIS no reconoce el fichero “.htaccess” de Apache. Al comienzo de este proyecto se valoró la posibilidad de comprar un módulo de Microsoft que actúa como filtro de URLs basado en expresiones regulares, es decir, un sustituto del fichero “.htaccess” de Apache para ser entendido por IIS 6, pero de momento la universidad no accede a pagar más dinero por software porque entiende que con el pago de las licencias actuales es más que suficiente. Por tanto la utilización de Wordpress, pese al interés que suscitó la posibilidad de crear redes multisite, no es posible en este escenario.

3.2.8 - Drupal

Drupal es otro sistema de administración de contenido por módulos con grandes posibilidades de configuración. Como Wordpress, es de código abierto, con licencia GNU/GPL y

está desarrollado con PHP por una comunidad de usuarios que se encargan también de su mantenimiento. También es destacable su respeto a los estándares del W3C y la atención prestada a la usabilidad y la estabilidad del sistema.

A priori, la especialidad de Drupal es la construcción y administración de comunidades en internet, aunque permite realizar bastantes tipos de sitios web. La dirección de su web a disposición de usuarios y desarrolladores es <http://www.drupal.org> que registra la participación de varios miles de usuarios.

Drupal nació en el año 2000, de la mano de Dries Buytaert y Hans Snijder, como un pequeño sitio online para publicar noticias, intereses o memorias, para ir cambiando de forma progresiva a un entorno de experimentación personal, en el que quedaran reflejados discusiones y flujos de ideas acerca de tecnologías web que poco a poco serían probadas en la misma plataforma. En 2001 el software fue liberado para extender el proyecto y ampliar las posibilidades de desarrollo.

3.2.8.1 - Características básicas

- Ayuda on-line: Un estable sistema en línea y sitios de ayuda para los módulos, tanto para administradores como para usuarios.
- Opciones de búsqueda: El contenido creado con Drupal se indexa de forma automática y eficiente pudiendo ser consultada al momento y ser encontrada con facilidad.
- Variedad de módulos: Hay una considerable cantidad de módulos desarrollados por la comunidad de Drupal proporcionando funcionalidades tales como paginación por categorías, autenticación con jabber, mensajes privados, bookmarks, etc.
- Open-source: El código fuente de Drupal es totalmente abierto, por las atribuciones que le otorga la licencia GNU/GPL.
- Personalización: El núcleo de Drupal cuenta con un entorno de personalización potente, tanto a nivel de contenido como de presentación, aunque con algunas limitaciones.
- URLs amigables: Drupal también hace uso del mod_rewrite de Apache para crear URL's que puedan ser manejadas más fácilmente tanto por usuarios como por buscadores.
- Autenticación de usuarios: Permite el registro e inicio de sesión localmente o usando sistemas de autenticación externos como Jabber, Blogger, LiveJournal, etc. Permite también integrar LDAP si se necesita desarrollar una intranet.
- Roles y privilegios: Los administradores de Drupal simplemente deben asignar un rol a un usuario para que queden definidos sus privilegios.
- Control de rendimiento: Drupal ofrece un mecanismo de control que posibilita habilitar o deshabilitar ciertos módulos según la carga del servidor. Este mecanismo puede ser ajustado y configurado.

- Mecanismo de Caché: Drupal incorpora un sistema de caché que permite reducir la carga del servidor, reduciendo el número de consultas a la base de datos y aumentando así el rendimiento.
- Sistema de taxonomías estable y potente: El contenido está organizado por categorías gracias a un módulo de taxonomías, que genera términos ordenados por jerarquías con capacidad para asociarse a un nodo particular si fuera necesario. Es posible configurar un nodo para clasificarlo bajo uno o múltiples términos de un vocabulario.

ICanLocalize Translation Dashboard

Show documents in: ☒ English ☐ German ☐ Spanish ☐ Language neutral

Translation status: ☐ All documents ☒ Not translated or needs updating

Filter further by: ☐ Status: published ☐ Type: Book page

[Update display](#)

Title	Type	Status	Translation
How translators work in ICanLocalize Translator	Page	Published	Not translated yet

Translation options:

☒ Translate to Spanish
☒ Translate to German
[Translate selected documents](#)

Your balance with ICanLocalize is \$475.81. Visit your [ICanLocalize finance page](#) to deposit additional funds.

Translation cost estimate

There are approximately 469 words in 1 nodes.
 The cost estimate to translate to any language is \$32.83.
 The details for each node type are:

Node type	Words	Cost estimate
page	469	\$32.83
Total	469	\$32.83

Figura. Panel de gestión de Drupal

Tipos de contenido:

En Drupal, cualquier tipo de contenido se conoce como nodo, pudiendo incluir en un nodo artículos, historias, comentarios, encuestas, imágenes, etc. De esta forma se estandariza el contenido asignando las mismas características a diferentes tipos de objetos y se mantiene la información centralizada y debidamente catalogada. Los tipos de nodo existentes son los siguientes:

- Página de libro, diseñadas para formar parte de un libro colaborativo, como puede ser la documentación de Drupal.
- Comentarios, que permiten a los usuarios comentar sobre cualquier nodo.
- Foros, que agrupan nodos a partir de un término de la taxonomía. Los foros a su vez pueden agruparse en contenedores de foros mediante el uso de estos términos, creando jerarquías.
- Páginas estáticas, que son los nodos más simples.
- Encuestas de selección múltiple, enfocadas a la participación de los usuarios.

- Historias, que se usan para aportar información relevante únicamente por un periodo breve de tiempo.

Módulos:

Si las características básicas de Drupal no satisfacen al desarrollador web, Drupal presenta una estructura basada en módulos que permite extender las funcionalidades y personalizar algo más el sitio web de forma sencilla. Algunos de los módulos más interesantes que ofrece la plataforma son los siguientes:

- Captcha, que provee al sistema de registro de usuarios de una validación más estricta para evitar la entrada de ciertos robots.
- Daily, que aporta tiras cómicas y frases diarias asociadas a la fecha en la que se muestran.
- E-Commerce, que provee de módulos para crear una herramienta de comercio digital.
- Font size, que otorga al usuario la posibilidad de cambiar el tamaño de fuente con CSS y Javascript.
- Live Discussion, que posibilita presentar un bloque con la lista de los últimos comentarios.
- Organic groups, que posibilita crear y administrar grupos de usuarios.
- Photo álbum, que posibilita mantener varios álbumes de fotos.
- Project, que permite el seguimiento de proyectos notificando por correo electrónico.
- Subscriptions, que posibilita a los usuarios suscribirse para que se les envíe notificaciones sobre nuevas entradas.
- Syndication, que posibilita la centralización en una única página todos los RSS generados.
- Video, que posibilita la inclusión de vídeos en formato QuickTime en la página.
- Workflow, que posibilita la definición de flujos de publicaciones personalizados. De esta manera se libera al autor original de tener que evaluar y modificar el estado de cada uno de los nodos.

3.2.8.2 - Limitaciones

Drupal exige utilizar una base de datos y un servidor web que soporte PHP. El SGBD recomendado es MySQL, aunque también se ha probado con PostgreSQL y funciona correctamente. El servidor web recomendado vuelve a ser Apache, y aunque es compatible con IIS desde hace un tiempo, vuelven a surgir problemas para hacerlo funcionar en versiones de IIS anteriores a la versión 7, y en concreto en este escenario se desechó su instalación tras varios intentos.

Por otra parte, Drupal no puede ser aprovechado para crear más de un blog, que es lo único que podría rentabilizar la cesión de un servidor dedicado, además, no es precisamente el CMS más intuitivo, por lo que el primer diseño con esta plataforma sería costoso y no proporcionaría excesivas ventajas sobre un desarrollo con la plataforma .NET.

3.2.9 - Otros CMS basados en PHP

No se puede finalizar un repaso a los CMS gratuitos con PHP sin hacer mención al más popular de todos, Joomla! Sin embargo, la razón del éxito de este CMS es la facilidad de uso, pensada para principiantes. Sin embargo, es inferior a Wordpress y Drupal tanto en soporte a otras tecnologías, APIs, variedad de módulos, sistemas de permisos, escalabilidad, etc.

Tras haber repasado la actualidad de CMS basados en las tres grandes plataformas de desarrollo web y no haber encontrado ninguna que se adecúe completamente a las rigurosas limitaciones impuestas por el sistema instalado en la Universidad CEU Cardenal Herrera, queda resuelto el enigma de la tecnología a utilizar para el desarrollo de este proyecto: ASP.NET.

A continuación, este estudio se va a centrar en el estudio básico de algunas de las tecnologías empleadas para redifusión y posicionamiento web, y en cómo hacer uso de ellas desde ASP.NET.

3.3 - Tratamiento del contenido del sitio web. Sindicación, metadatos y posicionamiento.

La **sindicación o redifusión web** consiste en reenviar contenidos desde el sitio web original hasta una página receptora que a partir de entonces se convierte en emisora de esa información. Esta reemisión de contenidos se realiza generalmente a través de un acuerdo mediante el cual el sitio web de origen pone sus contenidos a disposición del sitio web de destino. Sin embargo, este término también es válido cuando se habla de proporcionar una fuente o “feed” proveniente de un sitio web para proveer a otras personas de las actualizaciones del mismo.

Los feeds se codifican normalmente en XML, aunque también es válido cualquier otro formato que pueda ser transportado por HTTP, como son los casos de HTML o JavaScript. Los formatos más conocidos de redifusión web son RSS, RDF y ATOM.

Para recibir y leer las actualizaciones del feed normalmente se hace uso de un agregador, que es una aplicación, que puede estar basada en web o escritorio, que actualiza el feed cada vez que aparece una nueva publicación.

En este proyecto se plantea la posibilidad de obtener provecho de estas tecnologías de redifusión tanto para crear una fuente web de noticias que se actualice automáticamente y pueda ser leída mediante un agregador, como para acceder a los contenidos de un sitio web alojado en un servidor remoto a través de su fuente web y convertir una página de la web destino en una nueva fuente de dicha información.

Por otra parte, los **metadatos** son datos que describen otros datos, agrupados en recursos. Se utiliza como una herramienta básica para la indexación de los sitios web, ya que se trata de índices para ubicar los datos y localizarlos de forma más sencilla mediante las consultas a

buscadores. Con el uso de esta información adicional, los resultados son más exactos, ahorrando al usuario filtraciones manuales y mejorando así el posicionamiento del sitio web.

Concretando más sobre los beneficios que producen los metadatos, se puede decir que:

- Los metadatos incorporan contenido, contexto y estructura a los objetos de información, ayudando así al proceso de recuperación de datos desde conjuntos de objetos.
- Generan diferentes puntos de vista conceptuales a los usuarios y permiten liberar al sistema de recopilar información avanzada sobre los objetos descritos y sus características.
- Facilitan el intercambio de información sin que sea necesario enviar el propio recurso.
- Generan valor añadido a los recursos a medida que se crean metadatos para describir otros metadatos ya existentes.
- Posibilitan un acceso controlado a los recursos al conocer con exactitud el objeto que se describe.
- Mantienen almacenados dichos objetos de información estructural de cara a migraciones para su uso por generaciones venideras.

En definitiva, resultan esenciales para hacer sostenible el crecimiento de una web escalable para hacer viables las búsquedas, la integración y la recuperación del contenido desde cualquier otra fuente.

A continuación se realizará un breve resumen de los distintos formatos de fuente web, basados en lenguajes de marca y metadatos, antes de consumir la elección del feed para este proyecto.

3.3.1 - Sindicación y metadatos en un solo marco: RDF

El Marco de Descripción de Recursos (RDF) es una DTD o aplicación de metadatos basada en XML 1.0 que pretende otorgar un marco estándar para el intercambio y reutilización de los contenidos de la web. XML obliga a restringir la estructura de los documentos aplicando métodos de expresión semántica, y una de las infraestructuras creadas para ese uso es RDF, que permite codificar, reutilizar e intercambiar metadatos estructurados.

Sus inicios se remontan a agosto de 1997, a iniciativa del W3C, uniéndose a las investigaciones de la comunidad bibliotecaria en torno a Dublin Core (DC), una organización que trata de fomentar el desarrollo de vocabularios semánticos para la descripción de recursos de forma inteligente, y que ha sido la primera en basar su modelo en RDF.

El formato RDF se divide principalmente en un modelo de datos, una sintaxis y un esquema. El modelo de datos propuesto por RDF se basa en tres tipos de objetos:

- Los recursos, que se identifican por un identificador de recursos URI, pudiendo ser desde un simple documento HTML hasta un sitio web completo.

- Las propiedades, que imponen las restricciones para definir esos recursos: valores permitidos, tipos de recursos que puede describir y relaciones entre propiedades.
- Las descripciones, que conjuntan el recurso, un nombre de propiedad y su valor.

Para definir las propiedades y sus relaciones con otros recursos, RDF se apoya en otra especificación llamada esquema RDF. Se trata de una serie de informaciones relativas a las clases de recursos que indica las relaciones jerárquicas establecidas entre ellos y otros matices. Además, para evitar disparidades de términos provocadas por la diversidad de intereses de los usuarios de la web se utilizan los namespaces (xml:ns), que proporcionan espacios inequívocos para cada recurso.

Pero aún es posible añadir más vocabulario para describir propiedades y clases, tales como relaciones entre clases (como puede ser la disyunción), cardinalidad, igualdad, tipologías de propiedades de mayor complejidad, enumeración de clases, caracterización de propiedades...y para ello RDF se apoya en **OWL**, otro lenguaje de marcado que hace uso de rigurosos esquemas conceptuales, llamados ontologías.

```
<!-- http://data.ordnancesurvey.co.uk/ontology/spatialrelations/touches -->

<owl:ObjectProperty rdf:about="touches">
  <rdfs:isDefinedBy rdf:resource="http://www.ordnancesurvey.co.uk/ontology/spatialrelations.owl"/>
  <rdf:type rdf:resource="&owl;SymmetricProperty"/>
  <rdfs:label xml:lang="en">touches</rdfs:label>
  <rdfs:comment rdf:datatype="&xsd:string">
    >The boundaries of the two objects intersect but their interiors do not.</rdfs:comment>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
  <owl:propertyChainAxiom rdf:parseType="Collection">
    <rdf:Description rdf:about="equals"/>
    <rdf:Description rdf:about="touches"/>
  </owl:propertyChainAxiom>
  <owl:propertyChainAxiom rdf:parseType="Collection">
    <rdf:Description rdf:about="touches"/>
    <rdf:Description rdf:about="equals"/>
  </owl:propertyChainAxiom>
</owl:ObjectProperty>
```

Figura 3.6. Fragmento de la ontología de relaciones espaciales (en el lenguaje OWL y en sintaxis XML/RDF) creada por Ordnance Survey, la agencia gubernamental británica que elabora los mapas del Reino Unido. Se muestra la definición de la relación espacial 'touches' como relación simétrica, entre otras propiedades.

El gran problema que limita la recuperación y organización de la información basado en la semántica es que en la actualidad no existen motores de recuperación de carácter general para permitir búsquedas que se basen en documentos con formato RDF-XML por toda la web.

3.3.2 - Metadatos y posicionamiento SEO con HTML

En la actualidad están más extendidos otros medios para la sindicación de los contenidos buscando mayor simplicidad para la redifusión a costa de perder potencia, eliminando los vocabularios de web semántica que conjuntaba RDF.

Para trabajar la indexación, localización y clasificación de los contenidos por los buscadores, se recurre a los metadatos que aparecen en la cabecera del documento HTML. Los tres campos META más importantes por orden de relevancia son title, description y keywords.

El campo <META NAME='title' CONTENT='título de la página'> es el primero en el que reparan los robots y se refiere al título que aparece en los buscadores. Debe ser el título de cada

página, ya que de esta forma se indexará el sitio web tantas veces como 'titles' distintos haya. Se recomienda que no supere las 10 palabras o 60 caracteres.

El campo META DESCRIPTION contiene la descripción de la página, limitado orientativamente a unas 30 palabras o 160 caracteres y es el contenido que se muestra en los buscadores a continuación del título de la página, por lo que es importante que contenga riqueza de palabras clave de relevancia.

El campo META KEYWORDS se usaba para ayudar a indexar más fácilmente los documentos a robots y buscadores, pero ha perdido relevancia frente a los dos metadatos anteriores. De hecho, Google la ha retirado de su algoritmo debido al uso abusivo que se le daba, repitiendo palabras clave e incluso colocando palabras de otro ámbito distinto al del sitio web concreto buscando mejor posicionamiento a través del engaño.

Así pues, uno de los métodos que se pretende implementar para lograr un mejor posicionamiento SEO en la web consiste en rellenar de forma dinámica, mediante consultas a los campos adecuados de la base de datos, los campos 'title' y 'description' de cada una de las páginas del sitio web.

En cuanto a la redifusión web, actualmente conviven dos métodos con bastante popularidad, y aunque el RSS es el formato más extendido en la web, el formato de redifusión Atom ofrece algunas mejoras al primero y se postula como una alternativa válida de cara al futuro.

3.3.3 - Sindicación mediante RSS

RSS es un estándar para la redifusión web de contenidos cuyas siglas significan "Really Simple Syndication", y esta simplicidad parece ser la causa de su aceptación y soporte en la mayoría de los navegadores, sin necesidad de software adicional. Nació en 2002 y se refiere realmente al estándar RSS 2.0, ya que los estándares anteriores, RSS 0.9 y RSS 1.0, se correspondían con el formato RDF.

A continuación se muestra un ejemplo básico para poner de manifiesto la afirmación sobre la simplicidad de estos documentos XML frente a la versión anterior del sindicador.

Estructura de un RSS

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rss version="2.0">
<channel>
<title>WebNet Architect</title>
<link>http://webnetarchitect.wordpress.com</link>
<description>Tutorial para importar RSS mediante ASP.NET</description>
<item>
<title>Estructura de un RSS</title>
<link>http://wp.me/p1w1Ur-6G</link>
<description>Ejemplo de la estructura de un RSS básico</description>
</item>
</channel>
</rss>
```

La primera línea del documento es la declaración de XML que define la versión XML y la codificación de caracteres utilizados en el documento. En este caso, el documento se ajusta a la 1.0 especificación de XML y utiliza el conjunto de caracteres utf-8.

La siguiente línea es la declaración RSS, que identifica que trata, de hecho, un documento RSS (más concretamente, RSS versión 2.0), ya que las versiones anteriores ya se ha visto que presentan un formato RDF, completamente distinto al RSS 2.0.

La siguiente línea contiene el elemento `<channel>`. Este elemento se utiliza para describir la fuente RSS. El elemento `<channel>` define tres elementos requeridos:

- `<title>`: define el título del canal (por ejemplo, el nombre de su sitio Web)
- `<link>`: define el hipervínculo al canal (por ejemplo, <http://webnetarchitect.wordpress.com>)
- `<description>`: describe el canal (por ejemplo Tutorial para importar...)

Cada elemento de `<channel>` puede tener uno o más elementos de `<item>`. Cada elemento de `<item>` define un artículo dentro de la RSS feed. El elemento `<item>` requiere tres elementos secundarios:

- `<title>`: define el título del tema (por ejemplo, primer artículo)
- `<link>`: define el hipervínculo al elemento (por ejemplo, <http://wp.me/p1w1Ur-6G>)
- `<description>`: describe el elemento (por ejemplo, la descripción del primer artículo).

La sindicación de contenidos mediante RSS es también una buena herramienta para SEO, ya que ayuda a conseguir enlaces en pocos segundos, aumenta el valor de la página en buscadores, mejora la relevancia de las palabras clave y ayuda a entrar a formar parte de una red que comparte características parecidas.

Como alternativa al RSS se ha desarrollado el formato de sindicación **ATOM**, que presenta soluciones frente a algunos problemas de compatibilidad entre algunas versiones del protocolo RSS, debido a fallos de interoperabilidad de los protocolos de publicación basados en XML-RPC. Por otra parte, parece volver por la senda de la complejidad que se abandonó cuando se estandarizó RSS 2.0. De momento no es el formato de sindicación más rentable, aunque Google ya ha apostado por él para Blogger.

No obstante, parece más conveniente la adopción del formato RSS 2.0 en el sitio web que se va a desarrollar en este proyecto. En primer lugar se pretende generar un feed RSS mediante consultas a la base de datos que se realizarán cada vez que un usuario enlace con la página que contiene el RSS. Este RSS contendrá las noticias publicadas por el observatorio en la sección de noticias y se podrá leer directamente desde el navegador (todos los navegadores, excepto Google Chrome construyen un formato de lectura para los feed RSS), o mediante agregadores de contenidos.

Otra interesante aplicación es la de extraer la información contenida en formato RSS de otros sitios web a cuya base de datos no se tiene acceso o con bases de datos incompatibles, para plasmar este contenido maquetado en una página de este sitio web y así utilizarlo como nueva fuente. Este proceso se realizaría mediante la correspondiente licencia, y resulta ventajoso frente a otros métodos, como la creación de un **servicio web** para poder realizar consultas a dicho contenido.

3.3.4 - Indexación y posicionamiento mediante SITEMAP de ASP.NET

Otra estrategia para ayudar a posicionar mejor el sitio web es otorgar a los buscadores un “listado” de las páginas que se desea indexar, acompañado de información adjunta que haga la visita más eficaz. Para ello hay que recurrir a la utilización de Sitemaps.

Un sitemap o mapa del sitio es un fichero XML que contiene ese listado de páginas junto a algún contenido adicional como el título, la descripción o la URL de esa página. Al publicar un sitemap en la web se avisa a los buscadores cuando las páginas cambian, proporcionado su nueva ubicación.

La forma más simple de crear un mapa del sitio es crear un archivo XML llamado web.sitemap que organizará de forma jerárquica las páginas del sitio. El proveedor del mapa del sitio lo cargará de forma predeterminada en ASP.NET. El fichero web.sitemap tiene que estar ubicado en el directorio raíz del sitio web, pero puede hacer referencia a otros proveedores o archivos de sitemap que se encuentren en otras carpetas siempre y cuando esos ficheros se encuentren en la misma aplicación. A continuación se puede ver un ejemplo de web.sitemap:

```
<siteMap>
  <siteMapNode title="Home" description="Home" url="~/default.aspx">
    <siteMapNode title="Products" description="Our products"
      url="~/Products.aspx">
      <siteMapNode title="Hardware" description="Hardware choices"
        url="~/Hardware.aspx" />
      <siteMapNode title="Software" description="Software choices"
        url="~/Software.aspx" />
    </siteMapNode>
    <siteMapNode title="Services" description="Services we offer"
      url="~/Services.aspx">
      <siteMapNode title="Training" description="Training classes"
        url="~/Training.aspx" />
      <siteMapNode title="Consulting" description="Consulting services"
        url="~/Consulting.aspx" />
      <siteMapNode title="Support" description="Supports plans"
        url="~/Support.aspx" />
    </siteMapNode>
  </siteMapNode>
</siteMap>
```

Como se puede observar, hay que agregar un elemento siteMapNode a cada página del sitio web, para crear posteriormente la jerarquía anidando los elementos siteMapNode. En el ejemplo, las páginas de Hardware y Software son elementos secundarios del elemento siteMapNode de Products. El atributo “title” indica el texto que se usa generalmente como texto del vínculo y el atributo “description” se utiliza como documentación y como información de las herramientas en el control SiteMapPath, que representa los datos del mapa del sitio que el proveedor mediante la propiedad siteMapProvider.

Cuando el sitio web que se desea indexar está compuesto de páginas web que se actualizan dinámica y periódicamente, la generación del fichero web.sitemap puede realizarse desde un control que se activará periódicamente. Este control generaría el fichero escribiendo los nombres de las etiquetas y completando la información dinámica de los nodos mediante consultas a la base de datos de la aplicación.

4 - Especificación

4.1 - Especificación de requisitos

4.1.1 - Funcionales:

La página web del Observatorio pretende dar a conocer la realidad del below a nivel nacional y establecer sus fundamentos teóricos y operativos.

A través de la difusión periódica de los resultados obtenidos, el Observatorio pretende convertirse en referente en el estudio y aplicación de estas disciplinas y contribuir a la formación y cualificación de los futuros profesionales de la publicidad en las nuevas técnicas de comunicación empresarial.

Para el presidente del Grupo AGR Comunicación, Mateo Blay, el Observatorio Below abarcará tanto las técnicas más asentadas en el ámbito del below the line como las que surjan para los nuevos soportes tecnológicos, así como las tendencias internacionales que vayan a llegar pronto a España desde Estados Unidos o Japón.

Tras mantener una reunión con M^a Isabel de Salas Nestares, directora del Observatorio Below, obtuvimos los requisitos funcionales de la página web con la que se pretende comunicar los resultados de dichos estudios. Son los siguientes:

4.1.1.1 - Requisitos funcionales para la página web del Observatorio Below.

ID	REQUISITO
RF – 1	A nivel de diseño, se desea que el ancho de la página web se adapte al ancho de las pantallas de los ordenadores, ocupando todo el espacio posible. En la parte superior habrá un menú horizontal con pestañas mediante el cual se accederá a los apartados más importantes de la web. Habrá dos bandas verticales en los laterales. La banda izquierda contendrá el logo corporativo, a continuación los sub-apartados correspondientes a cada uno de los apartados del menú superior y otros apartados a especificar. La banda derecha contendrá un buscador, enlaces a redes sociales y otros apartados a especificar. Este diseño se mantendrá para todos los apartados principales de la web.

ID	REQUISITO
RF – 2	En la parte central se dispondrá de un espacio amplio para poder introducir los contenidos – que se actualizarán dinámicamente – correspondientes a cada apartado de la web. En la parte inferior habrá una lista con un histórico de todos los contenidos anteriores a los que se podrá acceder haciendo click en cada uno de los enlaces.

ID	REQUISITO
RF - 3	Los apartados del menú superior serán los siguientes: - Nosotros - Expertos - Estudios - Seminarios, congresos y jornadas

	- Experiencias creativas - El observatorio estuvo en... Los contenidos de cada apartado deben ser proporcionados por los usuarios de la web de forma dinámica.
--	--

ID	REQUISITO
RF – 4	En la banda lateral izquierda, los sub-apartados del apartado “Nosotros” son los siguientes: - Objetivos - Memorias - Equipo - Premios obtenidos - Observador admirado - Partners - Contacta con nosotros. El apartado “Observador admirado” debe mostrar los contenidos de un blog externo, que también se actualiza periódicamente. El apartado “partners” contendrá un listado de los logos de los partners del observatorio con un enlace a la página web de cada uno de ellos. De momento se introducirán de forma estática. Los sub-apartados del resto de apartados del menú podrán ser introducidos por los usuarios de forma dinámica. En la parte inferior de dicha banda, encontraremos también el apartado: “Observamos el sector”, con un espacio dedicado a la introducción de un vídeo embebido de producción externa y un enlace a una biblioteca de vídeos que se introducirán dinámicamente. Finalmente, se dispondrá de un enlace a un RSS de la página web para poder suscribirse.

ID	REQUISITO
RF – 5	En la banda lateral derecha, además del buscador y de los enlaces a las redes sociales, encontraremos los siguientes apartados: - Enlaces y directorios - Últimas noticias - Nuestros vídeos El apartado “Últimas noticias” mostrará un avance de las tres noticias más recientes y un enlace al resto de las noticias. El apartado “Nuestros vídeos” mostrará un vídeo embebido, de producción interna y un enlace a una biblioteca de vídeos propios que se introducirán dinámicamente.

Para que la web del Observatorio se convierta en un medio de difusión con una eficacia acorde al propio observatorio, es necesario contar con una aplicación de gestión que permita a sus usuarios introducir contenido personalizado para cada uno de los apartados de la web de forma rápida y sencilla. El acceso a los distintos apartados de esta aplicación dependerán de los privilegios que se le otorguen a cada tipo de usuario.

Por esta razón, los requisitos funcionales de la aplicación de gestión se han dividido en cuatro categorías: generales, la de profesor, la de becario y la de administrador.

4.1.1.2 - Requisitos funcionales generales de la aplicación de gestión.

Los requisitos funcionales generales que se definen para la aplicación son los siguientes:

ID	REQUISITO
RF – 1	La aplicación debe incluir el logo corporativo y sus colores, el rojo y el negro, pero no es necesario que esté integrada en la página web ni que mantenga su diseño. Se pide que sea lo más funcional posible.

ID	REQUISITO
RF – 2	Es imprescindible autenticarse como usuario para poder acceder a la aplicación, obteniendo privilegios en función al rol definido desde el gestor de usuarios. El control de acceso se realizará a través de un mecanismo de login/password tradicional. La información de las cuentas de usuario estará almacenada en la base de datos de la aplicación.

ID	REQUISITO
RF – 3	Debe existir una pantalla de gestión de contenido para cada uno de los apartados de la web. Desde el gestor se podrá acceder a los siguientes enlaces: - Página principal - Noticias - Seminarios - Estudios - Conferencias - Experiencias Creativas - Expertos - Vídeos - Secciones de las Noticias - Secciones de los Expertos - Secciones de los Estudios - Enlaces y Directorios - El observatorio estuvo en - Memorias - Premios - Gestión de los usuarios Siendo cada uno de estos apartados independiente de los demás a excepción de los apartados “Conferencias” y “El observatorio estuvo en”, que estarán ligados a un seminario.

4.1.1.3 - Requisitos funcionales para el rol de profesor en la aplicación de gestión.

Los requisitos funcionales que se definen para el rol de profesor son los siguientes:

ID	REQUISITO
RF – 4	Un profesor del observatorio tendrá acceso a todos los apartados de la aplicación a excepción del apartado “Gestión de los usuarios”.

ID	REQUISITO
RF – 5	En los apartados “Página principal”, “Noticias”, “Seminarios”, “Estudios”, “Conferencias”, “Experiencias Creativas”, “Expertos”, “Vídeos”, “Enlaces y directorios”, “El observatorio estuvo en”, “Memorias” y “Premios” se incluirá un cuadro de texto para el título, un control WYSIWYG para incluir el contenido deseado de forma personalizada, un calendario para establecer el plazo de vigencia de cada entrada, un control para subir al servidor un fichero de texto y mostrar un enlace al mismo, un control para subir una imagen al servidor, redimensionarla a elección del usuario y mostrarla en el control WYSIWYG y una tabla con un registro de todas las entradas ya almacenadas para poder editarlas y borrarlas a elección del usuario. Todos estos contenidos se almacenarán en la base de datos.

ID	REQUISITO
RF – 6	Los apartados “Vídeos” y “Enlaces” incorporarán un cuadro de texto para incluir la url de los mismos y una lista desplegable para elegir la categoría del vídeo o el enlace correspondiente. En el caso de los vídeos, el apartado url servirá para incluir el enlace de un vídeo - subido previamente a youtube - que será embebido en el apartado correspondiente de la página web, en función de la categoría del vídeo. Las categorías serán las mencionadas en los requisitos de la web: “Nuestros Vídeos” y “Observamos el sector”. El apartado “Enlaces”, además, incluirá cuadros de texto para incluir la siguiente información: “contacto”, “email”, “teléfono”, “fax” y “lugar”.

ID	REQUISITO
RF – 7	Los apartados “Premios” y “Memorias” incluirán un desplegable para elegir el curso académico al que se enlazará el contenido de dichos apartados.

ID	REQUISITO
RF – 8	Los apartados “Secciones de las Noticias”, “Secciones de los Expertos” y “Secciones de los Estudios” ofrecerán la posibilidad de incluir nuevas secciones que se almacenarán en la base de datos y serán incluidos en un desplegable en los apartados “Noticias”, “Expertos” y “Estudios” respectivamente. Todas las secciones se visualizarán en la banda lateral izquierda de la página web y los contenidos podrán ser filtrados en función de las mismas.

4.1.1.4 - Requisitos funcionales para el rol de administrador en la aplicación de gestión.

Los requisitos funcionales que se definen para el rol de administrador son los especificados para un profesor con el añadido de los siguientes:

ID	REQUISITO
RF – 9	Un administrador tendrá acceso a todos los apartados comentados anteriormente, y además, tendrá visible el apartado “Gestión de los usuarios” para poder introducir nuevos usuarios, borrarlos y editarlos.

ID	REQUISITO
RF – 10	El apartado “Gestión de los usuarios” dispondrá de cuadros de texto para incluir los datos de “Login”, “Nombre”, “Contraseña”, “Confirmación de la contraseña” y “Email”. También dispondrá de un calendario para fijar la caducidad de la cuenta de usuario, un control WYSIWYG para introducir más información personalizada, un desplegable para seleccionar el rol, un checkbox para elegir los apartados a los que tendrá acceso en caso de que el rol escogido sea el de “colaborador”, otro si se desea que el usuario se visualice en el apartado “Equipo” de la página web y, finalmente, una tabla con el listado de todos los usuarios registrados para editarlos o borrarlos a elección del administrador.

4.1.1.5 - Requisitos funcionales para el rol de colaborador.

Los requisitos funcionales que se definen para el rol de becario se definen a continuación:

ID	REQUISITO
RF – 11	El colaborador sólo tendrá acceso a los apartados que los administradores elijan desde el panel de “gestión de usuarios”, y tendrán plenos poderes sobre dichos apartados.

4.1.2 - No funcionales.

Además de los requisitos funcionales, tanto la página web como la aplicación que se pretenden desarrollar deben cumplir con una serie de requisitos técnicos y de desarrollo que vienen dados por la tecnología de la que se dispone para el desarrollo web en el servicio de Comunicación Digital de la Universidad CEU Cardenal Herrera.

ID	REQUISITO
RNF – 1	Se debe implementar una aplicación web desarrollada completamente utilizando tecnología .NET. Se utilizará el entorno de desarrollo Visual Studio 2010.

ID	REQUISITO
RNF – 2	La información que gestione la aplicación será almacenada en una base de datos SQL Server que será accesible desde la aplicación SQL Server Management Studio Express 2008.

ID	REQUISITO
RNF – 3	El servidor de aplicaciones sobre el que se desplegará la aplicación será el IIS 6 de Microsoft para el sistema operativo Windows Server 2003.

ID	REQUISITO
RNF – 4	La interactividad en el lado del cliente se realizará en ASP.NET, aunque se permitirá el uso de JavaScript si se considera conveniente. Para la programación en el lado del servidor se utilizará el lenguaje C#.

4.2 - Planificación

En este momento del proyecto se pretende combinar de la mejor forma posible los recursos disponibles, así como sus costes y su rigidez, y predecir todas las variables que pueden influir en el comportamiento futuro del mismo para tomar de antemano todas las medidas necesarias para lograr la máxima probabilidad de éxito y alcanzar los objetivos marcados en el proyecto.

El esquema seguido para llevar a cabo el proceso de planificación ha sido el siguiente:

- Definición y especificación del alcance: descomposición del proyecto en partes más sencillas de abordar para hacer más asequible el proceso de gestión y control.
- Identificación, definición y relación de las actividades de las que se va a componer el proyecto.
- Planificación de los recursos: El vínculo entre las actividades debe planificarse de forma rigurosa, haciendo una estimación de su duración que resulte viable, y especificando los recursos que se emplearán para cada una, teniendo en cuenta el tipo, la cantidad y la previsión del tiempo.
- Prelaciones: Después de definir las tareas y su duración, hay que determinar una secuenciación de las actividades para solucionar de forma adecuada las necesidades técnicas del proyecto y para conseguir una composición adecuada de costes y plazos.
- Presupuestación y programación de actividades: La estimación de los costes se consigue inmediatamente obteniendo la cantidad y el tiempo de asignación de cada recurso. Los plazos se delimitan tomando como referencia el camino crítico definido por las actividades y su independencia.
- Revisión de los objetivos: De la planificación global se puede obtener una variación de plazos y costes respecto de los previstos inicialmente. En este caso se debe volver a considerar y definir las especificaciones o la planificación realizada.

Cada uno de estos pasos se va a detallar a continuación:

4.2.1 - Descomposición del proyecto en partes más asequibles

A continuación se procede a la descomposición del proyecto en fragmentos más abordables de forma que cada una tendrá un papel fundamental en una fase determinada de cara al resultado final.

- Fase de investigación y recopilación de información
- Fase de análisis del sistema
- Fase de diseño del sistema
- Fase de implementación del sistema
- Fase de pruebas y resultados

- Fase de documentación

4.2.2 - Identificación y definición de las actividades del proyecto

Para especificar y describir las tareas se ha apostado por un modelo de tablas encabezadas por las fases y organizadas a su vez por tareas.

Fase 1. Investigación y recopilación de información	15 días
Tarea 1.1. Investigación sobre las técnicas y tecnologías que ofrece el mercado en la actualidad, realizando para ello una búsqueda bibliográfica.	5 días
Tarea 1.2. Estudio detallado sobre las técnicas y tecnologías investigadas mediante esa búsqueda y su viabilidad de uso en el contexto actual en la universidad.	10 días

Fase 2. Análisis del sistema	20 días
Tarea 2.1. Recopilación de requisitos. Anotación y análisis de los requisitos apuntados por el cliente.	3 días
Tarea 2.2. Análisis de casos de uso. Análisis de todos los casos de uso para cada uno de los actores que pueden hacer uso de la aplicación y del sitio web.	10 días
Tarea 2.3. Realización del diagrama de actividad. Estudio del modelo de negocio mediante el análisis de las acciones simultáneas que puede realizar cada uno de los actores.	5 días
Tarea 2.4. Realización del diagrama de clases para el modelo de datos.	2 días

Fase 3. Diseño del sistema	35 días
Tarea 3.1. Diseño arquitectónico del sistema. Se realiza un diseño de la arquitectura del sistema para organizar después cada uno de sus componentes y tener un patrón a seguir	5 días
Tarea 3.2. Diseño de los componentes del sistema. Se diseñan los componentes que serán necesarios tanto en la aplicación como en el sitio web.	15 días
Tarea 3.3. Diseño de clases del sistema. Se diseñan las clases que formarán parte de la aplicación y la web y los métodos que implementará cada una.	5 días
Tarea 3.4. Diseño de los diagramas de secuencia. Se hace un diseño mediante diagramas de las secuencias más representativas que pueden darse tanto en la aplicación como en la web.	5 días
Tarea 3.5. Diseño de la interfaz del sistema. Se diseña la interfaz de usuario de la aplicación y la interfaz del sitio web.	5 días

Fase 4. Implementación del Sistema	60 días
Tarea 4.1. Implementación de los componentes del sistema.	50 días
Tarea 4.2. Ensamblaje de los componentes.	10 días

Fase 5. Pruebas	10 días
Tareas 5.1. Pruebas.	5 días
Tarea 5.2. Depuración.	5 días

Fase 6. Documentación	17 días
Tareas 6.1. Redacción del manual de usuario y del administrador.	2 días
Tareas 6.2. Redacción de la memoria.	15 días

Total: 157 días.

Se observa que el proyecto tiene una duración aproximada de 5 meses y una semana sin tener en cuenta los días festivos. A esto hay que sumarle los días festivos que marca el calendario académico de la Universidad CEU Cardenal Herrera. De este modo se obtendrá la relación de días hábiles para la realización de este proyecto.

La primera reunión con el cliente para la obtención de los requisitos se espera producir a finales de enero de 2011 por lo que el estudio del estado del arte comenzará a la vuelta de las vacaciones de navidad. Se estima por tanto que los meses que quedarán hasta la finalización del curso académico 2010-2011 serán insuficientes, y habrá que continuar a la vuelta de las vacaciones de verano. Hay que tener en cuenta que el proyecto se realizará gracias a una beca de colaboración de 25 horas semanales y que puede ser renovada de cara al curso académico 2011-2012, periodo en el cual se puede finalizar el proyecto.

	TOTAL	DÍAS FESTIVOS	DÍAS
ENERO	31 días	15 días	16 días
FEBRERO	28 días	8 días	20 días
MARZO	31 días	13 días	18 días
ABRIL	30 días	16 días	14 días
MAYO	31 días	10 días	21 días
JUNIO	30 días	8 días	22 días
OCTUBRE	31 días	12 días	19 días
NOVIEMBRE	30 días	9 días	21 días
DICIEMBRE	31 días	20 días	9 días
	273 días	111 días	160 días

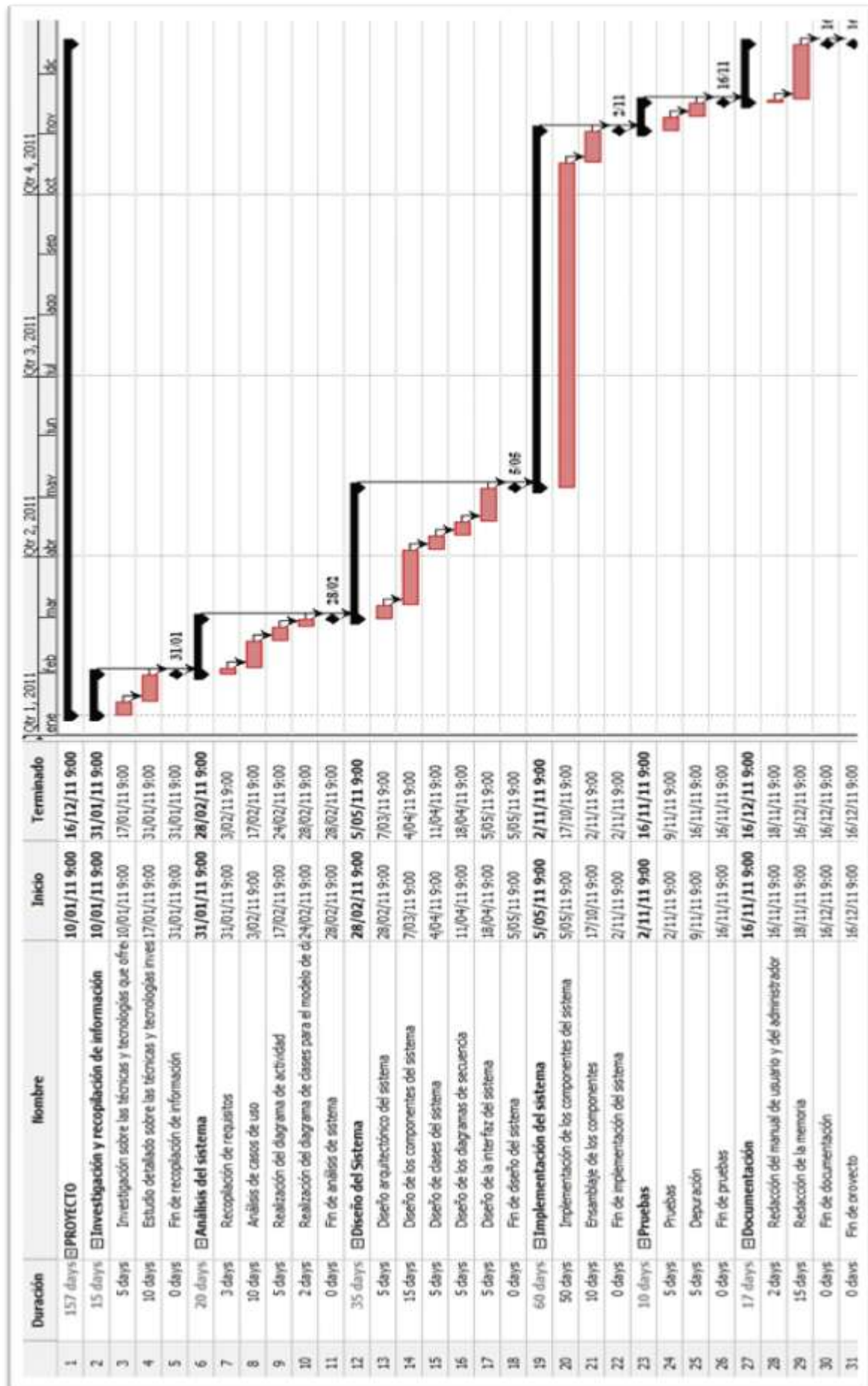
En base a estos cálculos sobran únicamente tres días para la resolución de posibles contratiempos, pero así el proyecto podría ser entregado en la convocatoria de enero de 2012.

4.2.3 - Diagrama de Gantt

Ha llegado el momento de detallar y precisar los tiempos de dedicación previstos de cada una de las tareas o actividades a lo largo del tiempo estimado, y para ello se va a hacer uso de un diagrama de Gantt.

En gestión de proyectos, el diagrama de Gantt muestra el principio y el fin de las distintas unidades mínimas de trabajo y los grupos de tareas o sus dependencias, representando las distintas fases, tareas y actividades proyectadas como parte de un proyecto. En el eje vertical se representan las actividades que componen el trabajo que se va a realizar, mientras que en el eje horizontal se muestra la duración de las mismas.

En la figura X se representa el diagrama de Gantt para las actividades que conforman este proyecto.



Para la planificación del desarrollo de proyectos de envergadura (que superen 25-30 actividades) se recomienda el uso de técnicas que complementen al diagrama de Gantt, como son el método de la ruta crítica (CPM) o los grafos PERT, basados en redes de precedencia. De esta forma se relacionan las actividades para poder visualizar el camino crítico del proyecto y posibilitan reflejar una escala de tiempos para hacer más sencilla la asignación de recursos y la especificación del presupuesto. Sin embargo, en este caso el método PERT no va a aportar información de utilidad pues todas las tareas se van a desarrollar de forma secuencial, ya que hay una única persona asignada para la realización de este proyecto y el camino crítico afecta a todas las tareas de igual manera.

4.2.4 - Planificación de recursos

Los recursos, tanto de Hardware como de Software de que dispone la universidad para la realización de este proyecto son los siguientes:

Componentes de Hardware:

- Intel® Pentium® Dual CPU E2220 @ 2.40 GHz 2.4 GHz, con 4 GB de RAM, 300GB de disco duro y conexión a internet para el desarrollo del proyecto en servidor local.
- Intel® Xeon® CPU E554 @2.83 GHz 2.83 GHz, con 4 GB de RAM, 30 GB dedicados de disco duro, conexión a internet y soporte para bases de datos SQL Server para el despliegue de la aplicación.

Componentes de Software:

- Sistema operativo Microsoft Windows 7 Professional para equipo local.
- Sistema operativo Microsoft Windows Server 2003 para servidor.
- Servidor Web IIS versión 6.
- IDE Visual Studio 2010 Ultimate para el entorno de desarrollo de aplicaciones Web con ASP.NET, NET Framework 4.0 y los lenguajes de programación Visual C#, Visual Basic, Visual C++ y Visual F#.
- Gestor de bases de datos SQL Server Management Studio Express 2005.
- SQL Server 2008 para administración de bases de datos.
- Procesador de textos Microsoft Office Word 2007 para la redacción de la memoria y los manuales con plugin para conversión a PDF.
- Aplicación Microsoft Office Project 2007 para la realización de la planificación del proyecto y el diagrama de Gantt.
- Navegadores Microsoft Internet Explorer 8, Mozilla Firefox 9 y Google Chrome.

4.2.5 - Prelaciones

Después de definir las tareas y su duración hay que instaurar una conexión lógica entre las actividades a ejecutar, para solucionar de forma adecuada los requisitos técnicos del proyecto y para conseguir la combinación óptima de costes y plazos.

Como se ha podido observar en el diagrama de Gantt de este proyecto, las actividades a realizar son acumulativas, por lo que no es posible comenzar una nueva actividad hasta que no finalice la actividad anterior pues únicamente hay una persona trabajando con una jornada de 5 horas desarrollando el proyecto sin que se lleven a cabo actividades al mismo tiempo. Así pues, este apartado no resulta de mayor interés para el desarrollo de este proyecto.

4.3 - Estimación de Costes

A continuación se va a hacer uso de las métricas propuestas para los proyectos software para determinar el alcance del proyecto, su esfuerzo y la productividad de los programadores.

4.3.1 - Estimación en función de las líneas de código

La evaluación de la magnitud de una aplicación web puede basarse en la medición del número de las líneas de código que contiene el proyecto. Este es el caso de la métrica LDC en la que se entiende por línea una sentencia del lenguaje en el que esté programado, sin contar comentarios ni líneas en blanco. A continuación se muestran distintos ejemplos de categorías de proyectos.

Categoría	Programadores	Duración	Líneas de código
Trivial	1	0 – 4 semanas	< 1K
Pequeño	1	1 – 6 meses	1K – 3K
Media	2 – 5	0,5 – 2 años	3K – 50K
Grande	5 – 20	2 – 3 años	50K – 100K
Muy Grande	100 – 1000	4 – 5 años	100K – 1M
Gigante	1000 – 5000	5 – 10 años	> 1M

Tabla. Categoría de un proyecto en función de las líneas de código.

Se prevé que el código generado entre el sitio web, la aplicación de gestión, la biblioteca de clases y la base de datos se sitúe en torno a las 10K líneas de código, lo que lo sitúa en la categoría de proyecto de complejidad media, aunque se pretende que en buena parte de los casos reutilizar código. El número de programadores que suelen hacer frente a proyectos de esta magnitud oscila entre 2 y 5 (en este caso sólo lo abordará uno) y la duración aproximada en la implementación de un proyecto como éste varía entre 6 meses y 2 años (en concreto se prevé que dure 9 meses).

4.3.2 - Estimación mediante modelo COCOMO

El término COCOMO es una construcción formada por las siglas del nombre “COConstructive COSt MOdel”, o modelo de coste de construcción, de una jerarquía de modelo de cálculo para el software, y está construida por

- Modelo COCOMO básico: Se trata de un modelo estático mediante el cual se puede calcular el esfuerzo del desarrollo del software en función de la magnitud del programa, expresado en líneas de código estimado.
- Modelo COCOMO intermedio: permite calcular el esfuerzo del desarrollo del software en función de la magnitud del programa y una adaptación de complejidad, en el que se incluye la evaluación subjetiva del sistema, del hardware, del personal y de las singularidades del proyecto.
- Modelo COCOMO avanzado: Se trata de un modelo que integra todas las funcionalidades del modelo intermedio y evalúa el impacto de los conductores de coste en todas y cada una de las fases del proyecto.

Estos tres tipos de modelos COCOMO están a su vez especificados para tres tipos de proyectos de software:

- Orgánico: Es el tipo de proyecto de software de pequeñas magnitud y complejidad, de los que se ocupan pequeños equipos (KLDC < 50).
- Semi-acoplados: Es el tipo de proyecto de software de magnitud y complejidad intermedias de los que se ocupan varios equipos, con distintos niveles de experiencia, que han de satisfacer requisitos generalmente de poca rigidez (KLDC < 300).
- Empotrado: Es el tipo de proyecto de software que tiene que ser implementado en un conjunto de hardware, software y restricciones operativas muy rígido.

En este caso únicamente se va a utilizar una variable de estimación que será el número de líneas de código, utilizando por tanto el COCOMO básico, ya que permite la obtención de una evaluación objetiva del esfuerzo que se ha realizado mediante muy poca información. Se puede considerar este proyecto como un proyecto de tipo orgánico, ya que presenta menos de 50K líneas de código.

La ecuación del esfuerzo de COCOMO básico es la siguiente:

$$E = \text{Esfuerzo} = a * KLDC^b \text{ (persona x mes)}$$

Donde KLDC es el número de líneas de código, distribuidas en miles, para el proyecto.

Por otra parte, la ecuación del tiempo de desarrollo es:

$$T = c * \text{Esfuerzo}^d \text{ (meses)}$$

Los coeficientes a y b y los exponentes c y d que aparecen en las fórmulas anteriores se obtienen de la siguiente tabla:

Tipo de proyecto	a	b	c	d
Orgánico	2,4	1,05	2,5	0,38
Semiacoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Así pues, a partir del esfuerzo y del tiempo de desarrollo se puede obtener una estimación del número de personas recomendadas para abordar este proyecto.

$$N = E / T$$

En el desarrollo de este proyecto se prevé la codificación de unas 10000 líneas de código (10 KLDC), por tanto, los resultados obtenidos al aplicar el modelo COCOMO básico para proyectos software de tipo orgánico son:

$$E = \text{Esfuerzo} = a * \text{KLDC}^b = 2,4 * (10)^{1,05} \approx 27 \text{ personas} * \text{mes}$$

$$T = \text{Tiempo desarrollado en meses naturales} = c * E^d = 2.5 * (27)^{0,38} = 8.75 \text{ meses}$$

(≈ 9 meses).

Número de personas recomendadas para desarrollar el proyecto:

$$N = E / T = 27 / 8.75 \approx 3 \text{ personas.}$$

4.3.3 - Estimación costes de personal

Ha llegado el momento de definir el personal que interviene en el proyecto, especificando su posición dentro de la organización y su especialidad.

En este caso se va a estimar los costes en función de la media de sueldo mensual de los programadores, situado en torno a los 1700 euros. Aplicando el método COCOMO de estimación de costes se obtiene que el coste de tres ingenieros durante 9 meses de trabajo asciende a 45900 euros (15300 por ingeniero).

4.3.4 - Estimación de recursos de Software y Hardware

Los costes del material se van a dividir en hardware y software, además de la posible utilización de material fungible y bibliográfico.

4.3.4.1 - Hardware:

Para el desarrollo del proyecto se han utilizado dos equipos, un ordenador local, que se va a suponer de nueva adquisición:

- Intel® Pentium® Dual CPU E2220 @ 2.40 GHz 2.4 GHz, con 4 GB de RAM y 300GB de disco duro → 800 euros. El ordenador se compró el año anterior y se prevé obtener rendimiento durante al menos este año y el próximo, por lo que la amortización anual es del 33,3%. Si el uso del PC para este proyecto se extiende durante 9 meses, su coste de amortización asciende a **200 euros**.
- Intel® Xeon® CPU E554 @2.83 GHz 2.83 GHz, con 4 GB de RAM, 30 GB dedicados de disco duro → 1200 euros. Se prevé una amortización similar a la del PC, por lo que cada año se amortizará un 33,3% y su coste a los 9 meses será de **300 euros**.
- Impresora multifunción Kyocera FS-C2126 Color → 1285 euros. Se prevé la utilización de esta impresora (compartida por todo el personal del edificio de Rectorado de la Universidad) para la impresión de las tres copias de la memoria y de los manuales para

los usuarios de la aplicación de gestión del observatorio. Por ello se considera que la amortización de este elemento frente al uso que se le va a dar es casi nulo, y su coste se va a estimar en **0 euros**.

En total, el coste de hardware asciende a **500 euros**.

4.3.4.2 - Software:

Cabe destacar en este apartado que la Universidad CEU Cardenal Herrera se beneficia de un 70% de descuento en la compra de licencias de software, por lo que los precios de las licencias utilizadas son los siguientes:

- Sistema operativo Microsoft Windows 7 Professional para equipo local → **45 euros**.
- Sistema operativo Microsoft Windows Server 2003 para servidor → **135 euros**.
- Servidor Web IIS versión 6 → Incluido en el precio de Windows Server 2003.
- IDE Visual Studio 2010 Ultimate para el entorno de desarrollo de aplicaciones Web con ASP.NET, NET Framework 4.0 y los lenguajes de programación Visual C#, Visual Basic, Visual C++ y Visual F#. → **3800 euros**.
- Gestor de bases de datos SQL Server Management Studio Express 2005. → **0 euros**.
- SQL Server 2008 para administración de bases de datos. → **7500 euros**.
- Procesador de textos Microsoft Office Word 2007 para la redacción de la memoria y los manuales con plugin para conversión a PDF. → **60 euros**.
- Aplicación Microsoft Office Project 2007 para la realización de la planificación del proyecto y el diagrama de Gantt. → **60 euros**.
- Navegadores Microsoft Internet Explorer 8, Mozilla Firefox 9 y Google Chrome. → **0 euros**.

En total, el coste de software asciende a **11600 euros**.

4.3.4.3 - Material fungible de oficina

- Folios para impresión de la memoria y manuales → **5 euros**.
- Kit de mantenimiento (toner) de impresora multifunción Kyocera FS-C2126 200000 páginas → **450 euros**. Se prevé una amortización en base al gasto de unos 600 folios, con lo cual el gasto de amortización es de aproximadamente **1,35 euros**.
- CD's para memoria y otros datos del proyecto → El pack de 50 CD Rom cuesta 20 euros. Sólo se va a hacer uso de 5 CD, por lo que el gasto va a ser de **2 euros**.

En total, el material fungible ha costado **8,35 euros**.

4.3.4.4 - Material bibliográfico

- Visual Studio 2010, .NET 4.0 y ALM → **32 euros**.

- La Biblia de C# (Anaya Multimedia) → **65 euros**.
- Tecnologías ASP.NET 4.0 (saltando desde la versión 2.0), José Manuel Alarcón Aguirre → **0 euros** (online).

En total, el material bibliográfico ha costado **97 euros**.

Sumando el coste de todos los recursos el resultado es de **12244 euros**.

4.3.5 - Resultado de la estimación de costes

A continuación se obtiene el resultado final de la estimación de costes, que se obtiene al realizar la suma de la estimación de costes de personal y la estimación de costes de los recursos de software, hardware y material fungible y bibliográfico que se necesitan para desarrollar este proyecto.

Elementos económicos del proyecto	PRECIO
Coste personal	45.900 euros
Recursos SW	11600 euros
Recursos HW	500 euros
Material Fungible	8,35 euros
Material bibliográfico	97 euros
TOTAL:	58.105,35 €

4.3.6 - Revisión de los objetivos. Estudio de viabilidad

El estudio de viabilidad se realiza para tomar la decisión final acerca de proseguir con el desarrollo del proyecto o abandonarlo. Para ello hay que valorar los posibles riesgos que pueden dar lugar al fracaso del mismo.

El concepto de riesgo hace referencia a los efectos imprevistos y a las contingencias que ponen en peligro la consecución de los objetivos que se persiguen.

La identificación y el análisis posterior de los posibles riesgos aportan la capacidad necesaria para tomar decisiones rápidas y efectivas para poder evitar los efectos negativos de los mismos a la hora de desarrollar el proyecto.

4.3.6.1 - Análisis de riesgos

Pese a que no hay una definición clara para los riesgos de software, existen dos características que se repiten en todas las definiciones de riesgo de software que existen:

- Incertidumbre: el acontecimiento que caracteriza al riesgo no ocurre con total probabilidad.
- Pérdida: si el riesgo se convierte una realidad, habrá consecuencias indeseadas.

Al analizar los riesgos es fundamental cuantificar en primer lugar el nivel de incertidumbre, y en segundo, el grado de pérdidas asociado a cada riesgo. Para ello, hay que asociar distintas categorías de riesgos:

- Riesgos de proyecto: son los que suponen una amenaza para el correcto desarrollo del proyecto y que afectan a la planificación temporal causando retrasos y aumentando los costes. Los riesgos del proyecto identifican los posibles problemas de presupuesto, planificación temporal, personal (asignación y organización), recursos, cliente y requisitos y su impacto en un proyecto de software. En este caso se identifican riesgos asociados a la planificación temporal y de personal, ya que hay un único programador en prácticas (becario de colaboración) asignado al proyecto con un horario de 25 horas semanales y existe incertidumbre acerca la continuidad del proyecto después del periodo estival, al suponer el fin del curso académico.
- Riesgos técnicos: son los que ponen en peligro la calidad y la planificación temporal del software. Si un riesgo técnico se hace realidad, la implementación se complica hasta volverse imposible. Los riesgos técnicos ponen de manifiesto problemas potenciales de diseño, implementación, de interfaz, verificación y de mantenimiento. En este sentido, el Servicio de Comunicación Digital de la Universidad CEU Cardenal Herrera posee las herramientas necesarias para el desarrollo de la aplicación y cuenta con un programador experto que ejercerá de tutor y asesor del programador asignado al mismo.
- Riesgos de negocio: son los que amenazan a la viabilidad del software que se pretende desarrollar. Frecuentemente dificultan la construcción del producto o proyecto.

Principalmente se puede distinguir entre cinco riesgos de negocio:

- *Riesgo de mercado*: buen funcionamiento del producto o sistema pero sin demanda. En este caso se trata de una aplicación hecha a medida, por lo que se es plenamente consciente del uso que se va a hacer de la misma.
- *Riesgo estratégico*: el producto no entra en el perfil de la estrategia comercial general de la empresa. Como ya se ha dicho, entre los objetivos del servicio se encuentra el desarrollo de aplicaciones de gestión para todos los departamentos y servicios de la universidad, por lo que este proyecto está plenamente justificado.
- *Riesgo de marketing*: el producto desarrollado no se sabe vender. En este caso el producto es un sitio web y una aplicación de gestión personalizados para una plataforma de investigación de distintas disciplinas de publicidad, por lo que es de suponer que sepa venderse. Aquí simplemente se le proporciona una herramienta.
- *Riesgos de presupuesto*: causa pérdidas en el presupuesto o personal asignado. Los riesgos de proyecto comentados en primer lugar desembocan en un riesgo de presupuesto ante la posibilidad de cambios en el personal y p'or tanto en la planificación.
- *Riesgos de dirección*: causa la pérdida del apoyo de una gestión experimentada a causa de cambios de perspectiva o a cambios de personal. En

este caso depende de la continuidad del programador jefe del servicio, actualmente con contrato indefinido.

Existe otra categorización general de los riesgos del software consistente en clasificarlos en función de su grado de predicción o conocimiento. Se puede observar a continuación:

- Los riesgos conocidos: son los que son descubiertos a raíz de una meticulosa evaluación del plan del proyecto, del entorno técnico y comercial en el que se ha implementado el proyecto, así como otras fuentes de información fiables, como pueden ser fechas de entrega demasiado optimistas, etc.
- Los riesgos predecibles: son lo que se pueden extrapolar de la experiencia en otros proyectos, como el cambio de personal o problemas de comunicación con el cliente.
- Los riesgos impredecibles: son los que pueden ocurrir pero son muy difíciles de identificar de antemano.

Identificar el riesgo es intentar sistemáticamente una especificación de las dificultades que pueden evitar la consumación del plan del proyecto (estimaciones, planificación temporal, carga de recursos, etc). Al identificar los riesgos conocidos y predecibles, se puede dar un paso al frente para eludirlos cuando sea posible y afrontarlos cuando sea menester.

Una forma de identificar riesgos es generar una lista de comprobación de elementos de riesgo. Así se puede centrar en un subconjunto de riesgos conocidos y predecibles en las siguientes subcategorías genéricas:

- Tamaño de producto: riesgos asociados con el tamaño general del software a desarrollar. Como ya se ha comentado, se trata de un proyecto de tamaño medio, de alrededor de 10K líneas de código. Este riesgo por sí solo no es representativo, pero sí si se relaciona con la planificación de tiempos al prever su comienzo para el mes de enero y finalizando el curso académico en julio.
- Impacto en el negocio: riesgos asociados con las limitaciones que se han impuesto a través de la gestión o debido a la situación del mercado. No se detectan riesgos importantes en esta categoría.
- Características del cliente: riesgos asociados con la sofisticación del cliente y la capacidad del programador de comunicarse con él en los momentos oportunos. Se van a producir reuniones cuando sea necesario y se espera que la comunicación sea frecuente, no obstante, hay que tener en cuenta el poco grado de conocimiento en materia de aplicaciones web por parte del cliente, lo que puede repercutir a la hora de explicar sus necesidades y de medir sus pretensiones.
- Definición del proceso: riesgos asociados con el grado de definición del proceso de software y su seguimiento por la organización de desarrollo. En este paso no se considera que se estén tomando grandes riesgos, al menos hasta el fin del curso académico, cuando la situación del personal implicado puede resultar incierta.

- Entorno de desarrollo: riesgos asociados con la disponibilidad y calidad de las herramientas que se van a emplear en el desarrollo de la aplicación. No se identifican riesgos en este punto.
- Tecnologías a construir: riesgos asociados con la complejidad de la aplicación a implementar y la tecnología de la que se va a hacer uso. La tecnología es la adecuada para el desarrollo de las aplicaciones, que pese a su extensión, no se prevé que sea excesivamente compleja.
- Tamaño y experiencia de la plantilla: riesgos asociados con la experiencia técnica y de proyectos de los programadores que van a desarrollar la aplicación. En este caso hay un tutor experimentado que cumplirá funciones de asesor y un único programador asignado al mismo, que es novato en el uso de la tecnología a utilizar, por lo que hay un riesgo considerable.

Esta lista proporciona una visión útil de riesgos genéricos y debe utilizarse cada vez que se realice el análisis y gestión de riesgos de un proyecto de software.

Una vez que se han identificado los riesgos potenciales que se pueden dar durante el proceso de implementación del proyecto, hay que proceder a *proyectar o estimar el riesgo*, es decir, medir cada riesgo de dos formas: en primer lugar efectuar el cálculo de la probabilidad de que un riesgo sea real y acabe ocurriendo, y en segundo lugar prever las consecuencias de los problemas asociados con el riesgo si finalmente ocurre. Así se podrá clasificar los riesgos detectados en función de su probabilidad y sus consecuencias para así tomar las precauciones con la anticipación necesaria.

Así pues, los pasos que se deben seguir al analizar los riesgos son:

IDENTIFICACIÓN		PROBABILIDAD		IMPACTO
DEL	→	DEL	→	DEL
RIESGO		RIESGO		RIESGO

Con los riesgos ya identificados y expuestos, cabe centrarse fundamentalmente en uno de ellos, la planificación temporal, ya que el fin del curso académico antes de la fecha prevista de fin de proyecto provoca incertidumbre sobre la continuidad del programador asignado. Esto haría inviable la presentación de este trabajo como Proyecto Fin de Master, por lo que en ese sentido el impacto de riesgo sería muy alto, pero la probabilidad del mismo no es tan elevada, pues existe compromiso firme tanto por parte del programador para llevar a buen término el proyecto, como por parte de la universidad para prorrogar la beca hasta entonces.

4.3.6.2 - Estudio de viabilidad económica/técnica/legal

Tras realizar el análisis de la planificación y los riesgos asociados al desarrollo del proyecto, es el momento de tratar de obtener conclusiones acerca de la viabilidad del sistema en función de tres parámetros muy claros: la viabilidad económica, técnica y legal.

4.3.6.2.1 - Viabilidad económica

La estimación de costes resulta un tanto alta para un tipo de proyecto que puede realizarse con licencias de software gratuitas. No obstante, cabe recordar que la elección de este software se debe a que se ha venido usando en todas las aplicaciones desarrolladas en la universidad. Por tanto, los recursos de software y hardware necesario han sido dispuestos de antemano y no hay que preocuparse por los costes de adquisición. Así pues, la oportunidad de amortización de los mismos hace altamente recomendable su uso. Por otra parte, el precio de un programador en prácticas es sensiblemente menor que el dispuesto en los costes de personal recomendado, por lo que, asumiendo los riesgos expuestos anteriormente al respecto, el coste previsto resulta menor que el estimado mediante el método COCOMO.

4.3.6.2.2 - Viabilidad técnica

No existe ninguna exigencia por parte del cliente acerca de las tecnologías a utilizar por lo que hay vía libre para elegir si es conveniente ajustarse a los recursos de software que ya está amortizando la universidad o escoger entre la variedad de licencias de software de código abierto que se han investigado en el apartado del estudio del arte. Por esta razón, la viabilidad técnica está completamente garantizada.

4.3.6.2.3 - Viabilidad legal

La viabilidad legal es un aspecto importante a tratar en este proyecto atendiendo a tres factores fundamentales.

- En primer lugar atendiendo a las licencias de software, ya que se está haciendo uso de licencias pago de Microsoft. Evidentemente han sido adquiridos de forma legal aprovechando los descuentos de los que se aprovecha por tratarse de una universidad.
- En segundo lugar hay que recurrir a la Ley Orgánica 15/1999 de Protección de Datos (LOPD), tanto por la inclusión de datos personales en el panel de gestión de usuarios (en este caso nombre y dirección de correo, que son datos de nivel de seguridad bajo), como por la posibilidad de que los usuarios introduzcan imágenes o ficheros adjuntos que puedan incluir datos de otras personas y publicarlos en su sitio web.

Para ello es conveniente dejar constancia en el contrato que los desarrolladores de la aplicación no se hacen responsables del contenido que los usuarios de la aplicación decidan adjuntar en la web.

- En tercer lugar hay que recurrir también a la Ley 34/2002 de Servicios de la Sociedad de la Información (LSSI) por la inclusión de formularios de contacto en el sitio web, donde el usuario introduce su dirección de correo electrónico.

Para cumplir con la normativa hay que informar conveniente al usuario de la web de los usos que se le puede dar a su dirección de correo y de sus derechos de acceso, rectificación, cancelación y oposición al uso de sus datos en caso de que desde la Universidad o desde el Observatorio se vaya a hacer uso de los mismos. Para ello hay que incluir la información de las condiciones de uso legal del sitio público en un lugar visible y cuyo contenido no dé lugar a la confusión.

En este caso, el Observatorio se somete a las condiciones legales del sitio web corporativo, del que es una rama, por lo que el aviso legal se encuentra en un enlace situado en la parte inferior de la página principal de la Universidad CEU Cardenal Herrera.

http://www.uchceu.es/aviso_legal.aspx

Por otra parte, aunque no se pretenda hacer uso de los datos con motivos publicitarios, es conveniente incluir en el formulario de contacto un enlace a un mensaje con la política de privacidad de la universidad en el uso de los datos que introduce el usuario. En este caso el mensaje que se ha escrito, asesorado por un experto en protección de datos de la Universidad es el siguiente:

“POLÍTICA DE PRIVACIDAD

En cumplimiento de lo dispuesto en la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal, la Universidad de CEU Cardenal Herrera, le informa que sus datos personales obtenidos mediante la cumplimentación de este formulario, van a ser incorporados al fichero automatizado de alumnos de la Universidad.

Asimismo, se le informa que la recogida y tratamiento de dichos datos tienen como finalidad facilitar y posibilitar la gestión de procesos administrativos, académicos, docentes y servicios que se efectúan la propia Universidad.

Deberá garantizar y responder, en cualquier caso, de la veracidad, exactitud, vigencia y autenticidad de los Datos Personales facilitados, y se comprometen a mantenerlos debidamente actualizados.

Si lo desea, puede ejercitar los derechos de acceso, rectificación, cancelación y oposición, previstos por la Ley, dirigiendo un correo electrónico a datos@uch.ceu.es comunicando el derecho a ejercitar, o enviando un escrito a Universidad CEU Cardenal Herrera, Secretaría General”.

5. Desarrollo del Proyecto

5.1 - Análisis del sistema.

Una vez expresados los requisitos funcionales y de software de la aplicación que se está diseñando, se procede en esta sección a realizar la tarea de Análisis de los Sistemas de Información, de la cual deben obtenerse los diagramas específicos (UML) que definirán de manera concreta el comportamiento que debe seguir el sistema que se está diseñando.

5.1.1 - Casos de uso.

La definición particular de cada caso de uso se realiza a continuación. En primer lugar se definen los actores:

ID	ACTOR
ACT - 1	ADMINISTRADOR
	DESCRIPCIÓN: Es el usuario encargado de gestionar la inserción, edición y eliminación de los usuarios de la aplicación, y se debe autenticar en el sistema como tal. También tiene privilegios para gestionar los contenidos. TIPO: Primario. CASOS DE USO: Insertar contenido nuevo, Consultar entradas antiguas, Editar entradas antiguas, Borrar entradas, Asignar estado, Subir archivos a servidor, Subir imágenes a servidor, Insertar nuevo usuario, Asignar rol a usuario, Asignar caducidad a usuario, Editar usuario, Eliminar usuario.

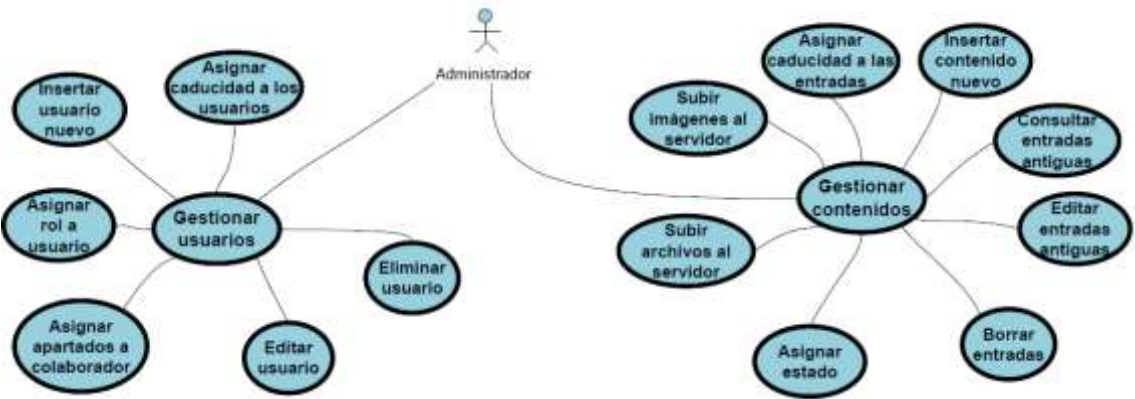


Figura 5.1. Casos de uso para Administrador

ID	ACTOR
ACT - 2	PROFESOR
	DESCRIPCIÓN: Usuario encargado de gestionar los contenidos de la aplicación.
	TIPO: Primario.
	CASOS DE USO: Insertar contenido nuevo, Consultar entradas antiguas, Editar entradas antiguas, Borrar entradas, Asignar estado, Subir archivos a servidor, Subir imágenes a servidor.



Figura 5.2 Casos de uso para Profesor

ID	ACTOR
ACT - 3	COLABORADOR
	DESCRIPCIÓN: Es el usuario encargado de gestionar, únicamente, los contenidos de la aplicación para los apartados a los que se le concede acceso.
	TIPO: Primario.
	CASOS DE USO: Insertar contenido nuevo, Consultar entradas antiguas, Editar entradas antiguas, Borrar entradas, Asignar estado, Subir archivos a servidor, Subir imágenes a servidor.

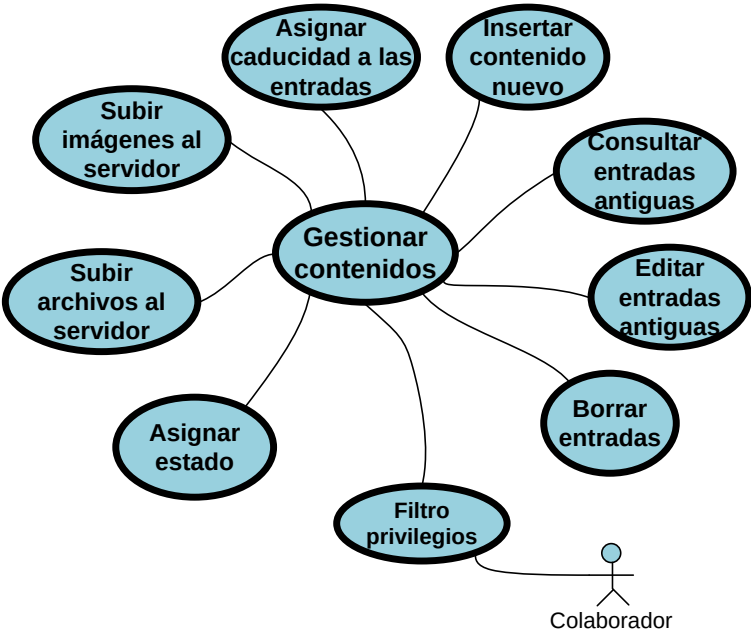


Figura 5.3. Casos de uso para Colaborador

ID	ACTOR
ACT - 4	ANÓNIMO
DESCRIPCIÓN: Persona que accede a la página web para visualizar los contenidos publicados e interactuar mediante los medios disponibles.	
TIPO: Primario.	
CASOS DE USO: Visualizar contenidos, Enviar formulario de contacto, Filtrar contenido con buscador, Compartir en redes sociales, Suscribirse al feed, Acceder a la zona privada, Recuperar datos de usuario.	

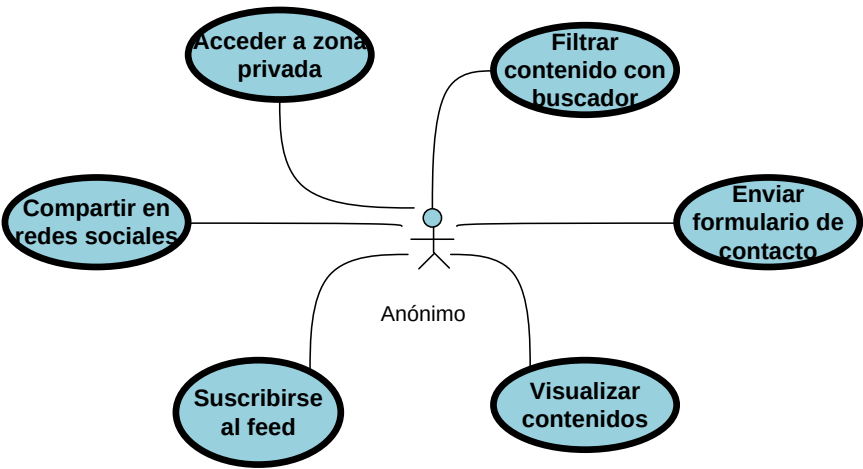


Figura 5.4. Casos de uso, usuario anónimo

A continuación se definen, acompañados de una breve descripción, cada uno de los casos de uso nombrados:

ID	CASO DE USO
CU - 1	VISUALIZAR CONTENIDOS WEB DESCRIPCIÓN: Un usuario anónimo puede acceder a la web pública y visualizar cualquier contenido de la misma. ACTORES: Anónimo. TIPO: Básica.

ID	CASO DE USO
CU - 2	FILTRAR CONTENIDOS MEDIANTE BUSCADOR DESCRIPCIÓN: El usuario anónimo puede realizar consultas de contenido introduciendo las palabras por las que filtrarlo. ACTORES: Anónimo. TIPO: Básica.

ID	CASO DE USO
CU - 3	ENVIAR FORMULARIO DE CONTACTO DESCRIPCIÓN: El usuario anónimo puede realizar consultas a través de un formulario cuyo contenido será enviado por email al equipo del Observatorio. ACTORES: Anónimo. TIPO: Básica.

ID	CASO DE USO
CU - 4	COMPARTIR EN REDES SOCIALES DESCRIPCIÓN: El usuario anónimo puede compartir cada contenido que visualiza en Facebook y Twitter - si es usuario - haciendo clic en el botón correspondiente. ACTORES: Anónimo. TIPO: Básica.

ID	CASO DE USO
CU - 5	SUSCRIBIRSE AL FEED DESCRIPCIÓN: El usuario anónimo puede acceder a un feed de las noticias del observatorio y suscribirse al mismo para recibir las actualizaciones. ACTORES: Anónimo. TIPO: Básica.

ID	CASO DE USO
CU - 6	ACCEDER A ZONA PRIVADA DESCRIPCIÓN: El usuario anónimo puede acceder al panel de identificación del gestor de contenidos indicando la URL. ACTORES: Anónimo. TIPO: Básica.

ID	CASO DE USO
CU - 7	RECUPERAR DATOS DE USUARIO DESCRIPCIÓN: Es posible que un usuario registrado haya olvidado su contraseña y desee recuperarla. Para ello tiene un enlace a su disposición que da acceso a un panel en el que puede introducir su dirección de correo registrada en la aplicación para que se le devuelva la contraseña. ACTORES: Anónimo. TIPO: Básica, Inclusión.

ID	CASO DE USO
CU - 8	VALIDAR USUARIO DESCRIPCIÓN: Cuando un usuario quiera acceder a la sección privada, se le solicitará que se autentifique a través del login y del password que el administrador le haya asignado en el panel de gestión de usuarios. ACTORES: Administrador, profesor, colaborador. TIPO: Básica, Inclusión.

ID	CASO DE USO
CU - 9	GESTIONAR CONTENIDOS DESCRIPCIÓN: Cuando el usuario se haya identificado, accederá al panel de gestión de contenidos. Si el usuario se ha identificado con rol de colaborador, tendrá acceso únicamente al apartado o apartados a los que le haya asignado un administrador desde el panel de gestión de usuario, con permisos de lectura, inserción, edición y eliminación. Si se ha identificado con rol de profesor, tendrá acceso a todos los apartados de la web, con los permisos citados anteriormente, a excepción del apartado de gestión de usuarios, que será visible únicamente para un usuario con rol de administrador. ACTORES: Administrador, profesor, colaborador (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 10	INSERTAR CONTENIDO NUEVO DESCRIPCIÓN: Cuando el usuario haya accedido a cualquiera de los apartados, podrá insertar contenido en cualquiera de los cuadros de texto habilitados para ello. Además de los cuadros de texto convencionales, el usuario tendrá a su disposición un gestor de contenido WYSIWYG que le permitirá maquetar dicho contenido tal y como desee que aparezca en la página web. La inserción de contenido se llevará a cabo tras pulsar el botón guardar. ACTORES: Administrador, profesor, colaborador (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 11	CONSULTAR ENTRADAS ANTIGUAS DESCRIPCIÓN: Cualquier usuario que haya accedido al panel de gestión de un apartado específico de la web tendrá acceso a un listado de las entradas almacenadas en la base de datos, y podrá visualizarlas. ACTORES: Administrador, profesor, colaborador (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 12	EDITAR ENTRADAS ANTIGUAS DESCRIPCIÓN: Cualquier usuario que haya accedido al panel de gestión de un apartado específico de la web tendrá acceso a un listado de las entradas almacenadas en la base de datos, y podrá acceder a las mismas para editar cualquier contenido, pulsando el botón guardar para actualizar el contenido editado. ACTORES: Administrador, profesor, colaborador (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 13	BORRAR ENTRADAS DESCRIPCIÓN: Cualquier usuario que haya accedido al panel de gestión de un apartado específico de la web tendrá acceso a un listado de las entradas almacenadas en la base de datos, y podrá borrarlas tras pulsar el enlace “borrar” y pulsar posteriormente el botón “aceptar” en la ventana que aparecerá solicitando la confirmación de la acción. ACTORES: Administrador, profesor, colaborador (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 14	ASIGNAR ESTADO DESCRIPCIÓN: Cualquier usuario que haya accedido al panel de gestión de un apartado específico de la web podrá elegir – seleccionando la opción “sí” o “no” - si desea que el contenido que se va a guardar en la base de datos sea visible en la página web. ACTORES: Administrador, profesor, colaborador (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 15	SUBIR ARCHIVOS AL SERVIDOR DESCRIPCIÓN: Cualquier usuario que haya accedido al panel de gestión de un apartado específico de la web podrá seleccionar un fichero (con extensión .doc .docx o .pdf) que se encuentre en cualquier dispositivo físico accesible desde el ordenador y subirlo al servidor en el que está alojada la aplicación web. ACTORES: Administrador, profesor, becario (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 16	SUBIR IMÁGENES AL SERVIDOR DESCRIPCIÓN: Cualquier usuario que haya accedido al panel de gestión de un apartado específico de la web podrá seleccionar una imagen que se encuentre en cualquier dispositivo físico accesible desde el ordenador y subirla al servidor en el que está alojada la aplicación web. Esta imagen se insertará automáticamente en el gestor de contenido WYSIWYG tras seleccionar el tamaño con la que desea que sea redimensionada. ACTORES: Administrador, profesor, colaborador (si tiene acceso a dicho apartado). TIPO: Básica.

ID	CASO DE USO
CU - 17	ASIGNAR CADUCIDAD A LAS ENTRADAS DESCRIPCIÓN: Cualquier usuario que haya accedido al panel de gestión de un apartado específico de la web podrá elegir la fecha en la que desea que caduque la entrada, seleccionándola en un calendario. Cuando esto ocurra la entrada pasará a tener estado “no visible”. ACTORES: Administrador. TIPO: Básica.

ID	CASO DE USO
CU - 18	GESTIONAR USUARIOS DESCRIPCIÓN: Cuando el administrador se haya identificado, tendrá acceso al apartado gestionar usuarios, desde el cual podrá consultar, insertar, editar, asignar rol y estado, y borrar cualquier usuario del sistema. ACTORES: Administrador. TIPO: Básica.

ID	CASO DE USO
CU - 19	INSERTAR NUEVO USUARIO DESCRIPCIÓN: El administrador podrá insertar los parámetros “login”, “nombre”, “password”, “rol”, “estado: visible/no visible”, y “descripción” del usuario a insertar. La inserción en la base de datos se realizará cuando se pulse el botón “guardar”. ACTORES: Administrador. TIPO: Básica.

ID	CASO DE USO
CU - 20	EDITAR USUARIO DESCRIPCIÓN: El administrador tendrá a su disposición un listado con todos los usuarios del sistema y podrá acceder y editar cualquier contenido del usuario seleccionado a través del enlace “editar”. Para que se produzca la actualización en la base de datos hay que pulsar el botón “guardar”. ACTORES: Administrador. TIPO: Básica.

ID	CASO DE USO
CU - 21	ELIMINAR USUARIO DESCRIPCIÓN: El administrador tendrá a su disposición un listado con todos los usuarios del sistema y podrá borrar el usuario seleccionado a través del enlace “editar”. Para que se produzca el borrado de la base de datos hay que pulsar el botón “aceptar” del cuadro de confirmación. ACTORES: Administrador. TIPO: Básica.

ID	CASO DE USO
CU - 22	ASIGNAR ROL A USUARIO DESCRIPCIÓN: El administrador podrá seleccionar en un desplegable el rol que le asignará a cada usuario que inserte, o a cada usuario que modifique en el panel de gestión de usuarios. ACTORES: Administrador. TIPO: Básica.

ID	CASO DE USO
CU - 23	SELECCIONAR COMPETENCIAS AL USUARIO COLABORADOR DESCRIPCIÓN: Cuando el administrador seleccione el rol de colaborador, podrá elegir los apartados para los cuales dicho usuario tendrá competencias de inserción, edición y borrado. ACTORES: Administrador. TIPO: Básica.

ID	CASO DE USO
CU - 24	ASIGNAR CADUCIDAD AL USUARIO DESCRIPCIÓN: El administrador podrá elegir la fecha de caducidad de los usuarios del sistema durante la inserción o edición de los mismos, marcándola en un calendario. Cuando caduque la cuenta el usuario será borrado. ACTORES: Administrador. TIPO: Básica.

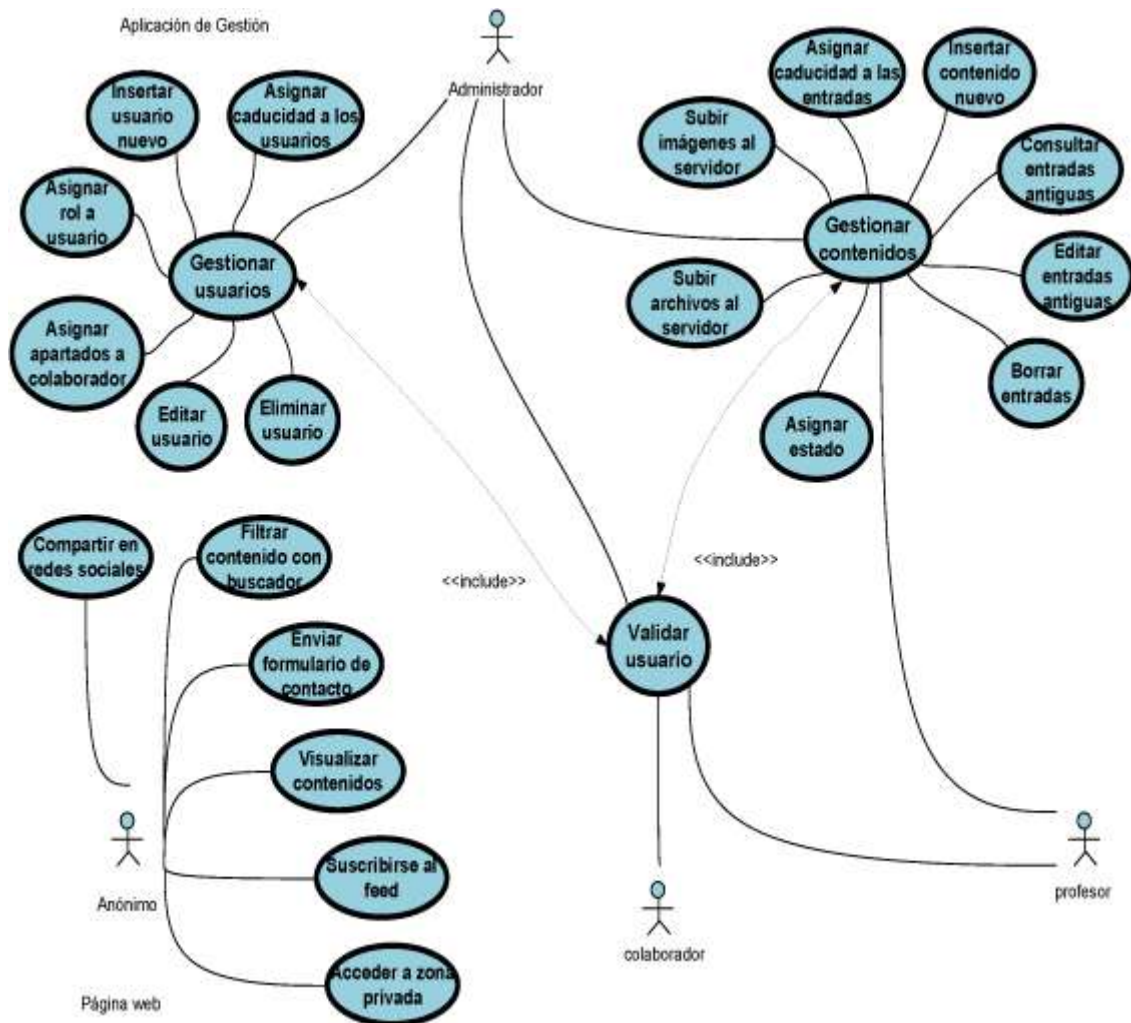


Figura 5.5. Diagrama completo de casos de uso.

5.1.2 - Diagrama de actividad

El modelo de negocio que se ha diseñado para la aplicación es el que se muestra en el siguiente diagrama de actividad, que es el único que se considera para analizar el sistema. En él se reflejan de forma simultánea las acciones que pueden realizar en el sistema cada uno de los actores considerados.

Más allá de pretender obtener un detalle completo del modelo de negocio, se pretende representar la secuencia de acciones que se pueden llevar a cabo para completar las actividades que se plantean en los casos de uso.

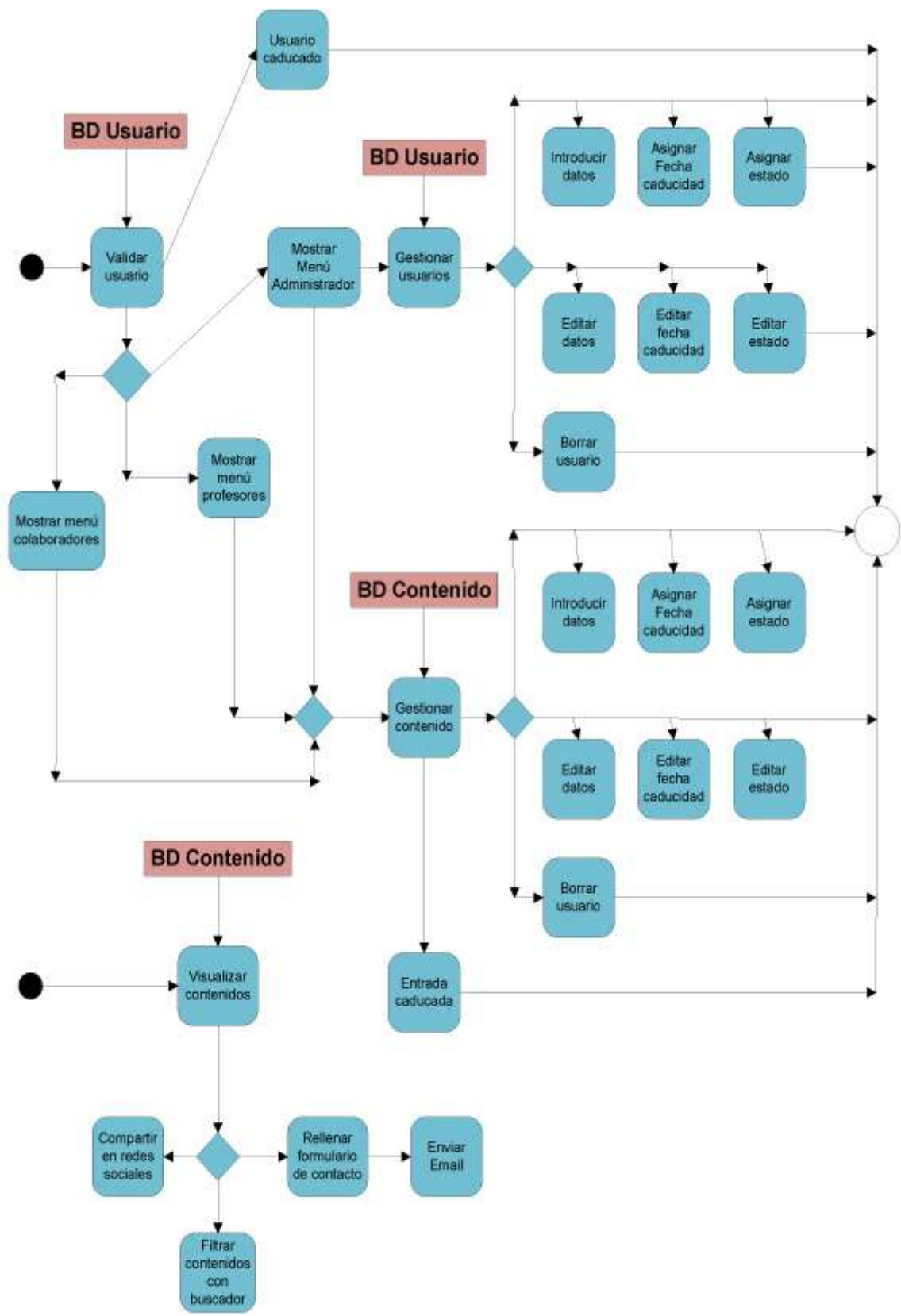


Figura 5.6. Diagrama de actividad

5.1.3 - Diagrama de clases para el modelo de datos:

Como se puede prever, el objetivo de este proyecto no requiere de un modelo de almacenamiento que presente una gran integridad, pues la mayoría de las entidades carecerán de relaciones. No obstante, a continuación se presenta el modelo conceptual de la base de datos que almacenará la información.

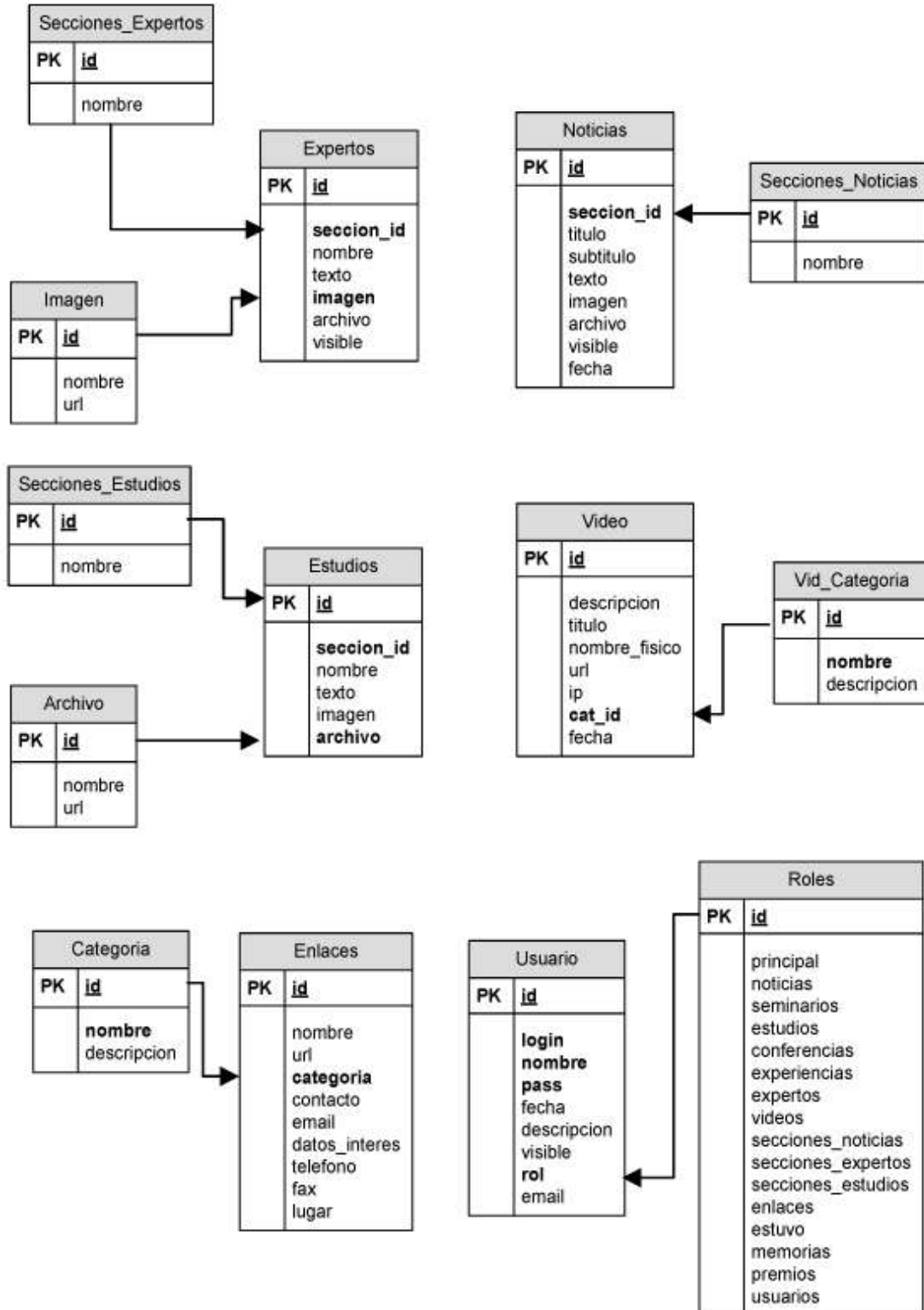


Figura 5.7. Tabla Modelo de datos 1.

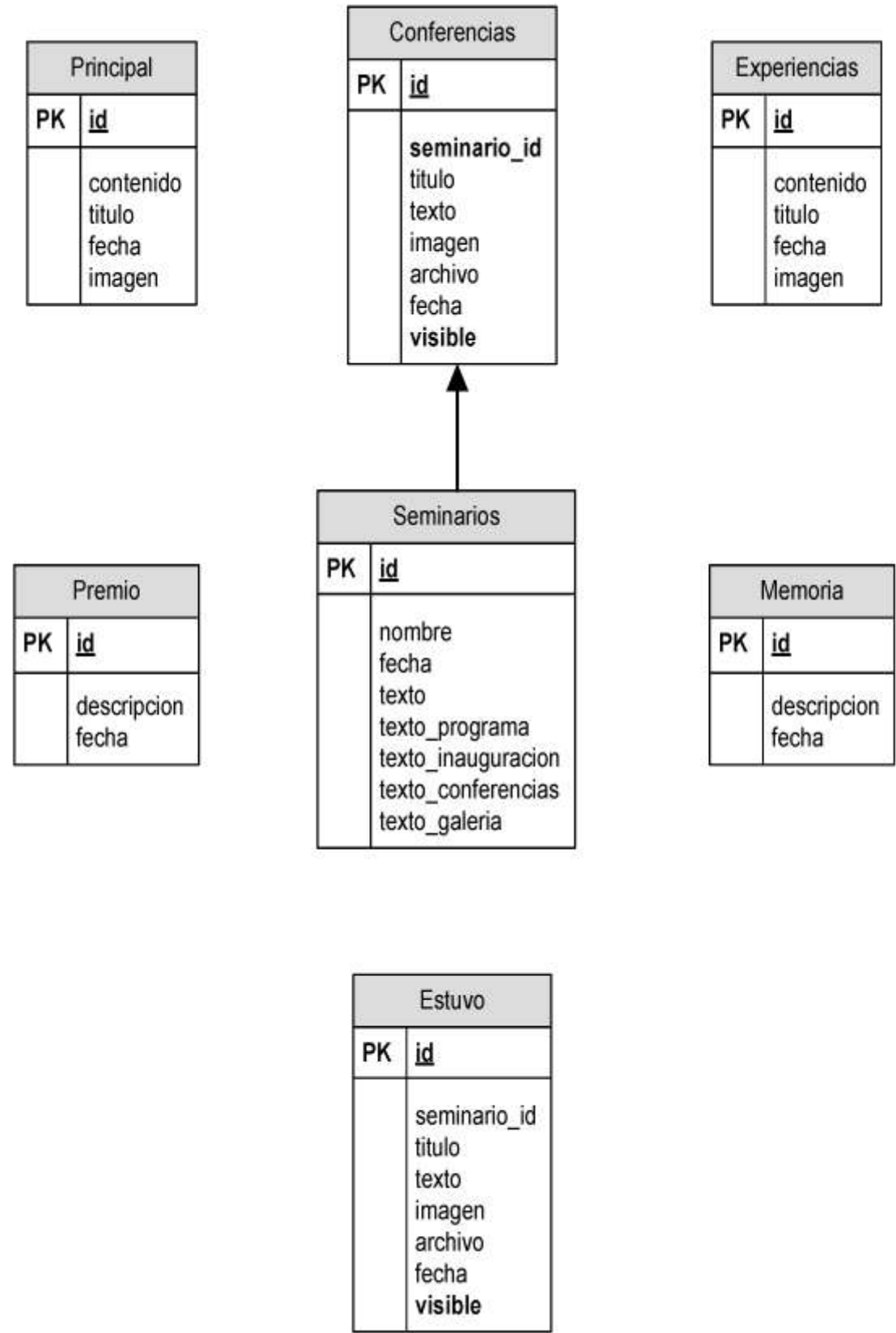


Figura 5.8. Tabla modelo de datos 2.

5.2 - Diseño del sistema

A continuación se procederá a realizar el diseño del sistema, diseñando en primer lugar la aplicación de gestión, que es la herramienta para generar los contenidos, y posteriormente la página web, que es el medio de visualización.

Para ello se va a hacer uso del modelo UML. En primer lugar se realizará una representación mediante diagramas de componentes para mostrar las dependencias entre ellos, posteriormente se hará uso de diagramas de clases para mostrar atributos, métodos y relaciones, y finalmente se utilizarán diagramas de algunas de las secuencias para mostrar la interacción entre clases y objetos del escenario.

5.2.1 - Aplicación de gestión del Observatorio.

5.2.1.1 - Diagramas de componentes.

Contenedor Web.

Se va a dividir la aplicación de gestión en tres grandes bloques: la capa de presentación, la capa de negocio – que se encuentran en el llamado contenedor web - y la capa de acceso a datos. Los componentes del sistema diseñado son los siguientes:

Capa de presentación

Active Server Page Extended <<aspx>>

frmLogin.aspx – Muestra el formulario para la autenticación del usuario y un enlace para acceder a frmEmail.aspx si el usuario ha olvidado su contraseña.

frmEmail.aspx – Muestra el formulario para la introducción de la dirección de email del usuario, que está almacenada en la base de datos, con el objetivo de recibir en su cuenta de correo la clave olvidada.

noAutorizado.aspx - Muestra el mensaje “no está autorizado a ver esta página” y proporciona el enlace para identificarse de nuevo.

Plantilla.Master – Como su nombre indica se trata de la plantilla que comparten todas las secciones de la aplicación y contiene los enlaces a las mismas. Contiene también el código que permite la inserción de un gestor de contenidos WYSIWYG que permitirá a los usuarios maquetar el contenido a insertar a su gusto, con las limitaciones impuestas en cuanto a tipografía de letra y tamaño de las imágenes.

frmIndex.aspx – Muestra una presentación general al usuario y el índice con las secciones de la web.

frmPrincipal.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado del contenido que se mostrará en la página principal de la web del Observatorio Below, así como la visualización de un histórico de todas las páginas principales anteriores – ordenadas por fecha de inserción – que hay almacenadas en la base de datos.

frmNoticia.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de las noticias, incluidas las secciones de las noticias en las que podemos clasificarlas, así como la visualización de un listado de las noticias que hay almacenadas en la base de datos.

frmDepartamentoNoticias.aspx – Contiene los elementos visibles del formulario para la inserción o eliminación de departamentos para la sección de noticias.

frmMemorias.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de las memorias, así como la visualización de un listado de las memorias de cada curso académico, que están almacenadas en la base de datos.

frmExpertos.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de expertos, incluidas las especialidades en las que podemos incluirlos, así como la

visualización de un listado de entrevistas que se han hecho a expertos en el below the line.

frmDepartamentoExpertos.aspx – Contiene los elementos visibles del formulario para la inserción o eliminación de las distintas especialidades en las que se puede englobar a los expertos.

frmEstudios.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de estudios, incluidas las secciones de los estudios en las que podemos clasificarlos, así como la visualización de un listado de estudios anteriores sobre la materia.

frmDepartamentoEstudios.aspx – Contiene los elementos visibles del formulario para la inserción o eliminación de las distintas materias en las que se pueden englobar los estudios.

frmEnlaces.aspx – Contiene los elementos visibles del formulario para la inserción o eliminación de enlaces de interés.

frmUsuarios.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de usuarios, así como la visualización de un listado de los usuarios que tendrán acceso a la aplicación y sus roles.

frmVideos.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de enlaces de vídeos obtenidos de youtube, así como la visualización de un listado de los vídeos almacenados en función de su categoría.

frmPremios.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de los premios, así como la visualización de un listado de los premios otorgados durante cada curso académico, que están almacenados en la base de datos.

frmExperiencias.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado del contenido del apartado “Experiencias Creativas”, así como la visualización de un histórico de todas las experiencias anteriores – ordenadas por fecha de inserción – que hay almacenadas en la base de datos.

frmSeminarios.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de los seminarios, así como la visualización de un listado de los seminarios realizados durante cada curso académico, que están almacenados en la base de datos.

frmConferencias.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de los seminarios, así como la asociación de la conferencia a uno de los seminarios almacenados, y la visualización de un listado de las conferencias realizadas durante cada curso académico, que están almacenadas en la base de datos.

frmEstuvo.aspx – Contiene los elementos visibles del formulario para la inserción, edición y borrado de los actos en los que ha participado el observatorio, así como la asociación de la conferencia a uno de los seminarios almacenados, y la visualización de un listado de las conferencias realizadas durante cada curso académico, que están almacenadas en la base de datos.

Active Server Control Extended <<ascx>>

ctrlArchivo.ascx – Contiene el control que permite la recepción de los ficheros que el usuario desea subir al servidor y el botón que activará el evento. Este control es llamado desde la mayoría de las páginas aspx, pero se asocia de forma obligatoria únicamente a estudios.

ctrlImagen.ascx – Contiene el control que permite la recepción de las imágenes que el usuario desea subir al servidor y el botón que activará el evento, así como un listado de opciones para redimensionar dicha imagen a tres tamaños posibles: pequeño (160px), mediano (320px) y grande (640px). Este control es llamado desde la mayoría de las páginas aspx, pero se asocia de forma obligatoria únicamente a expertos.

Capa de negocio

Active Server Page Extended C# <<aspx.cs>>

frmLogin.aspx.cs – Recoge los datos del formulario de autenticación y los envía a la capa de acceso a datos, redireccionando a la página noAutorizado.aspx si los datos no se encuentran en la base de datos y a frmIndex.aspx en caso de que sean válidos.

frmEmail.aspx.cs – Envía un email a la dirección de correo electrónico solicitada por el usuario si dicha dirección se encuentra en la base de datos. En su caso, escribirá en dicho correo el nombre de usuario y la contraseña.

frmPrincipal.aspx.cs – Recoge los contenidos introducidos a través del formulario que serán visualizados en la página principal del observatorio. Además envía las peticiones de inserción de cada nuevo contenido, visualización y edición de cada uno de los contenidos que figuran en el histórico, y eliminación de los mismos en función de las decisiones del usuario.

frmNoticias.aspx.cs – Recoge los datos del formulario de gestión de noticias y envía las peticiones de inserción, edición, obtención y eliminación de datos a la capa de acceso a los datos. También llama al componente Seccion_Noticias para obtener el listado de secciones en las que clasificar la noticia.

frmDepartamentoNoticias.aspx.cs – Recoge los datos del formulario de departamentos de la sección de noticias y realiza las peticiones de obtención de los departamentos existentes, así como las peticiones de inserción y borrado en función de la interacción con el usuario.

frmMemorias.aspx.cs – Recoge los datos del formulario de gestión de memorias y envía las peticiones de inserción, edición, obtención y eliminación de memorias a la capa de acceso a los datos.

frmExpertos.aspx.cs – Recoge los datos del formulario de gestión de expertos y envía las peticiones de inserción, edición, obtención y eliminación de las entrevistas a expertos que se van publicando a la capa de acceso a los datos. También llama al componente Seccion_Expertos para obtener el listado de secciones a las que se puede asociar al experto.

frmDepartamentoExpertos.aspx.cs – Recoge los datos del formulario de especialidades de la sección de expertos y realiza las peticiones de obtención de las especialidades ya registradas, así como las peticiones de inserción y borrado en función de la interacción con el usuario.

frmEstudios.aspx.cs – Recoge los datos del formulario de gestión de estudios y envía las peticiones de inserción, edición, obtención y eliminación de los estudios publicados a la capa de acceso a los datos. También llama al componente Seccion_Estudios para obtener el listado de secciones en las que clasificar el estudio.

frmDepartamentoEstudios.aspx.cs – Recoge los datos del formulario de especialidades de la sección de estudios y realiza las peticiones de obtención de los estudios ya realizados y almacenados, así como las peticiones de inserción y borrado en función de la interacción con el usuario.

frmEnlaces.aspx.cs – Recoge los datos del formulario de gestión de la sección de enlaces y envía las peticiones de inserción, edición, obtención y eliminación de los enlaces publicados a la capa de acceso a los datos.

frmUsuarios.aspx.cs – Recoge los datos del formulario relativos a la gestión de los usuarios de la aplicación y envía las peticiones de inserción, obtención y eliminación de usuarios del sistema a la capa de acceso a los datos.

frmVideos.aspx.cs – Recoge los datos del formulario relativos a la gestión de los vídeos enlazados de la aplicación y envía las peticiones de inserción, obtención y eliminación de url's de los vídeos a la capa de acceso a los datos.

frmPremios.aspx.cs – Recoge los datos del formulario de gestión de premios y envía las peticiones de inserción, edición, obtención y eliminación de premios obtenidos a la capa de acceso a los datos.

frmExperiencias.aspx.cs – Recoge los contenidos introducidos a través del formulario que serán visualizados en la sección de experiencias creativas. Además envía las peticiones de inserción de cada nuevo contenido, visualización y edición de cada uno de los contenidos que

figuran en el histórico de experiencias, y eliminación de estos contenidos en función de las decisiones del usuario.

frmSeminarios.aspx.cs – Recoge los datos del formulario de gestión de seminarios y envía las peticiones de inserción, edición, obtención y eliminación de los seminarios celebrados en cada curso a la capa de acceso a los datos.

frmConferencias.aspx.cs – Recoge los datos del formulario de gestión de seminarios y envía las peticiones de inserción, edición, obtención y eliminación de las conferencias celebradas en cada curso a la capa de acceso a los datos. Además envía el id del seminario que el usuario ha decidido asociar a dicha conferencia.

frmEstuvo.aspx.cs – Recoge los datos del formulario de gestión de contenidos relativos a la sección “El observatorio estuvo en...” y envía las peticiones de inserción, edición, obtención y eliminación de estos contenidos a la capa de acceso a los datos. Además envía el id del seminario que el usuario ha decidido asociar con esta sección.

Active Server Control Extended <<ascx>>

ctrlArchivo.ascx.cs – Recoge la ruta del archivo local que el usuario ha decidido almacenar y lo sube al servidor en el que se encuentra alojada la aplicación.

ctrlImagen.ascx.cs – Recoge la ruta de la imagen local que el usuario ha decidido almacenar, así como el tamaño de redimensión de la imagen. En primer lugar efectúa la llamada al método de la biblioteca “Utilidades” para redimensionar la imagen tomando como referencia el ancho elegido, en segundo lugar la sube al servidor, y en tercer lugar coloca la imagen en el cuadro de gestión de contenidos para que el usuario pueda colocarla donde le plazca.

Diagrama de componentes de la Aplicación de gestión.

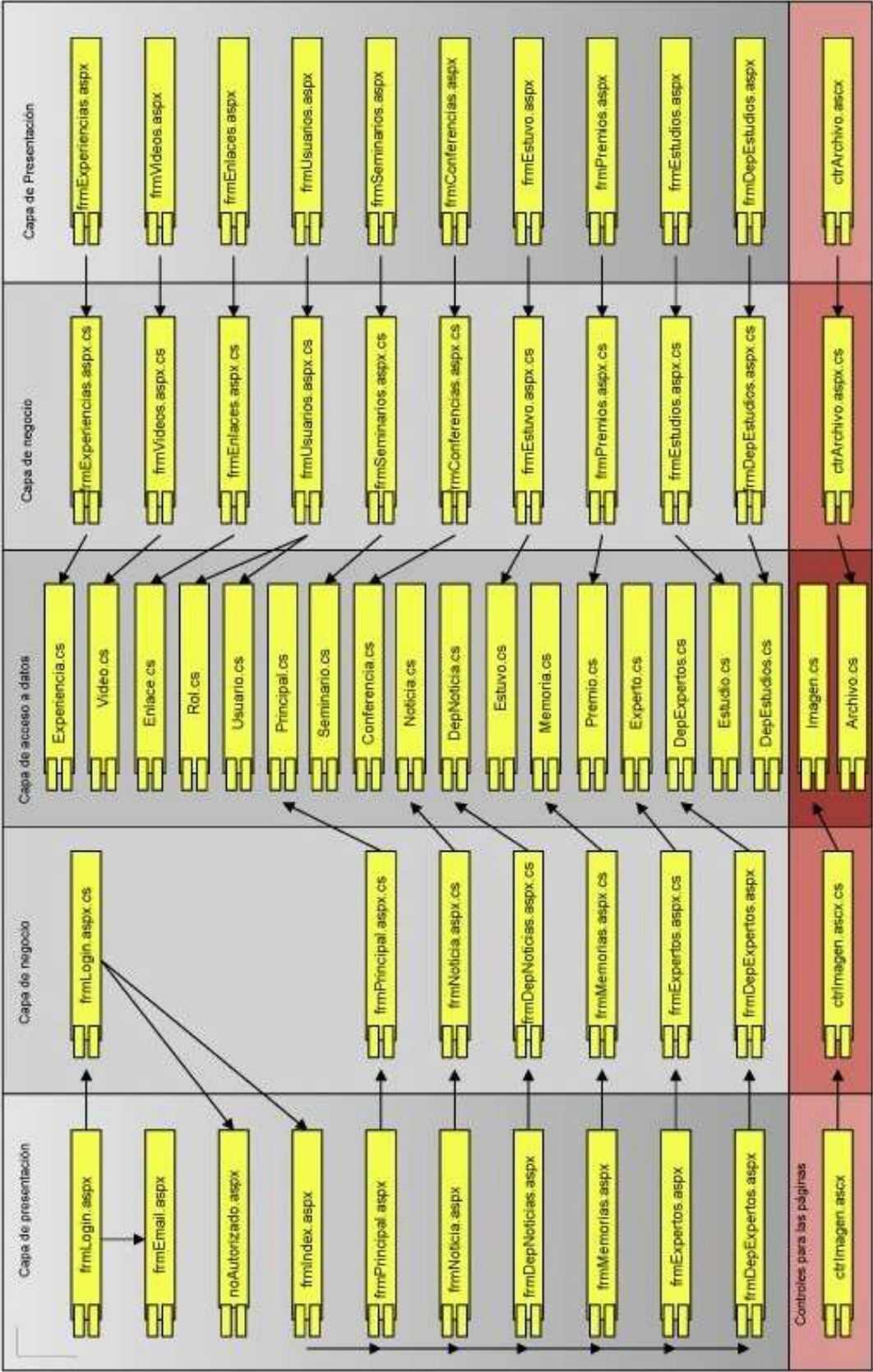


Figura 5.9 Diagrama de componentes 1 de la aplicación de gestión.

Contenedor de componentes de acceso a la base de datos.

Tanto la página web del observatorio como la aplicación de gestión de la misma comparten los mismos componentes de la capa de acceso a datos, ya que ambos acceden a las mismas tablas. Sin embargo, cada componente utiliza un procedimiento almacenado sql distinto en función de si es llamado desde la aplicación o desde la visualización. Por esta razón vamos a enumerar los componentes de la capa de acceso a datos en ambos casos, explicando a qué procedimiento se llama en cada caso.

Capa de acceso a datos

Clases <<.cs>> y procedimientos almacenados a los que llaman en la aplicación de gestión.

Principal.cs – Obtiene el listado de históricos de páginas principales a través de la llamada al procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Principal_todos]” así como la entrada filtrada por su identificador mediante el procedimiento

“dbo.[sp_obtener_WEB_Observatorio_Principal]”, inserta una nueva entrada mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Principal]”, que recibe todos los campos a insertar, actualiza cada entrada a editar mediante la llamada a

“dbo.[sp_actualizar_WEB_Observatorio_Principal]”, que recibe también todos los parámetros y elimina la entrada mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Principal]” a partir del Identificador único de cada entrada.

Noticia.cs – Obtiene el listado de noticias publicadas a través de la llamada al procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Noticias_todos]” así como la entrada filtrada por su identificador mediante el procedimiento

“dbo.[sp_obtener_WEB_Observatorio_Noticia]”, inserta una nueva noticia mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Noticia]”, actualiza los campos de cada noticia que se desea editar mediante la llamada a

“dbo.[sp_actualizar_WEB_Observatorio_Noticia]” y elimina la noticia mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Noticia]” a partir del identificador de la noticia.

Seccion_Noticias.cs – Proporciona el listado de secciones de noticias mediante el procedimiento “dbo.[sp_obtener_WEB_Observatorio_Secciones_Noticias_todos]”, inserta una nueva sección mediante la llamada a

“dbo.[sp_insertar_WEB_Observatorio_Secciones_Noticias]”, modifica la sección mediante la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Secciones_Noticias]” y la elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Secciones_Noticias]” a partir del ID. De la sección.

Memoria.cs – Obtiene el listado de memorias a través de la llamada al procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Memorias_todos]” así como la entrada filtrada por su identificador mediante el procedimiento

“dbo.[sp_obtener_WEB_Observatorio_Memoria]”, inserta una nueva memoria mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Memoria]”, actualiza los campos de cada memoria que se desea editar mediante la llamada a

“dbo.[sp_actualizar_WEB_Observatorio_Memoria]” y elimina la memoria mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Memoria]” a partir del ID. De la memoria.

Experto.cs – Obtiene el listado de expertos a través de la llamada al procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Expertos_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado

“dbo.[sp_obtener_WEB_Observatorio_Experto]”, inserta una nueva entrevista mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Experto]”, actualiza los campos de cada contenido que se desea editar mediante la llamada a

“dbo.[sp_actualizar_WEB_Observatorio_Experto]” y elimina la entrevista mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Experto]” a partir del ID. De la entrevista.

Seccion_Expertos.cs – Proporciona el listado de secciones de expertos mediante el procedimiento “dbo.[sp_obtener_WEB_Observatorio_Secciones_Expertos_todos]”, inserta una nueva sección mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Secciones_Expertos]”, modifica la sección mediante la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Secciones_Expertos]” y la elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Secciones_Expertos]” a partir del ID. De la sección.

Estudio.cs – Obtiene el listado de estudios a través de la llamada al procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Estudios_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Estudio]”, inserta un nuevo estudio mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Estudio]”, actualiza los campos de cada contenido que se desea editar mediante la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Estudio]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Estudio]” a partir de su ID.

Seccion_Estudios.cs – Proporciona el listado de secciones de estudios mediante el procedimiento “dbo.[sp_obtener_WEB_Observatorio_Secciones_Estudios_todos]”, inserta una nueva sección mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Secciones_Estudios]”, modifica la sección mediante la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Secciones_Estudios]” y la elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Secciones_Estudios]” a partir del ID. De la sección.

Enlace.cs – Obtiene el listado de enlaces a través de la llamada al procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Enlaces_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Enlace]”, inserta un enlace mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Enlace]”, actualiza los campos de cada enlace que se desea modificar a través de la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Enlace]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Enlace]”.

Usuario.cs – Obtiene el listado de usuarios registrados a través de la llamada al procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Usuarios_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Usuario]”. Inserta un nuevo enlace mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Usuario]”, actualiza los campos de cada usuario que se desea modificar a través de la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Usuario]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Usuario]” a partir de su ID. Este componente también se encarga de obtener el email del usuario – en caso de que haya perdido la contraseña – mediante la llamada al procedimiento “dbo.[sp_obtener_WEB_OBSERVATORIO_Email]”.

Rol.cs – Este componente se encarga de obtener los roles del usuario a partir del procedimiento “dbo.[sp_obtener_WEB_OBSERVATORIO_Roles]”.

Video.cs – Obtiene el listado de vídeos a través de la llamada al procedimiento “dbo.[sp_obtener_WEB_Observatorio_Videos_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Video]”, inserta un nuevo vídeo mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Video]”, actualiza los campos de cada vídeo que se desea modificar a través de la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Video]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Video]” a partir de su ID.

Premios.cs – Obtiene el listado de premios a través de la llamada al procedimiento “dbo.[sp_obtener_WEB_Observatorio_Premios_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado

“dbo.[sp_obtener_WEB_Observatorio_Premio]”, inserta un nuevo premio mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Premio]”, actualiza los campos de cada premio que se desea modificar a través de la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Premio]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Premio]” a partir de su ID.

Experiencias.cs – Obtiene el listado de experiencias creativas a través de la llamada al procedimiento “dbo.[sp_obtener_WEB_Observatorio_Experiencias_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Experiencia]”, inserta una nueva experiencia mediante la llamada al procedimiento “dbo.[sp_insertar_WEB_Observatorio_Experiencia]”, actualiza los campos de cada experiencia que se desea modificar a través de la llamada al procedimiento “dbo.[sp_actualizar_WEB_Observatorio_Experiencia]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Experiencia]” a partir de su ID.

Seminario.cs – Obtiene el listado de seminarios a través de la llamada al procedimiento “dbo.[sp_obtener_WEB_Observatorio_Seminarios_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Seminario]”, inserta un nuevo seminario mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Seminarios]”, actualiza los campos de cada seminario que se desea modificar a través de la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Seminario]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Seminario]” a partir de su ID.

Conferencia.cs – Obtiene el listado de conferencias a través de la llamada al procedimiento “dbo.[sp_obtener_WEB_Observatorio_Conferencias_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Conferencia]”, inserta un nuevo vídeo mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Conferencias]”, actualiza los campos de cada conferencia que se desea modificar a través de la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Conferencia]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Conferencia]” a partir de su ID.

Estuvo.cs – Obtiene el listado de contenidos a través de la llamada al procedimiento “dbo.[sp_obtener_WEB_Observatorio_Estuvo_todos]” así como la entrada dada a través de su identificador mediante el procedimiento almacenado “dbo.[sp_obtener_WEB_Observatorio_Estuvo]”, inserta un nuevo vídeo mediante la llamada a “dbo.[sp_insertar_WEB_Observatorio_Estuvo]”, actualiza los campos de cada contenido que se desea modificar a través de la llamada a “dbo.[sp_actualizar_WEB_Observatorio_Estuvo]” y lo elimina mediante la llamada a “dbo.[sp_borrar_WEB_OBSERVATORIO_Estuvo]” a partir de su ID.

Archivo.cs – Este componente se encarga en este caso únicamente de almacenar en la base de datos la ruta del archivo subido al servidor mediante la llamada al procedimiento “dbo.[sp_insertar_WEB_Observatorio_Archivos]”.

Imagen.cs – Este componente se encarga en este caso únicamente de almacenar en la base de datos la ruta de la imagen subida al servidor mediante la llamada al procedimiento “dbo.[sp_insertar_WEB_Observatorio_Imagenes]”.

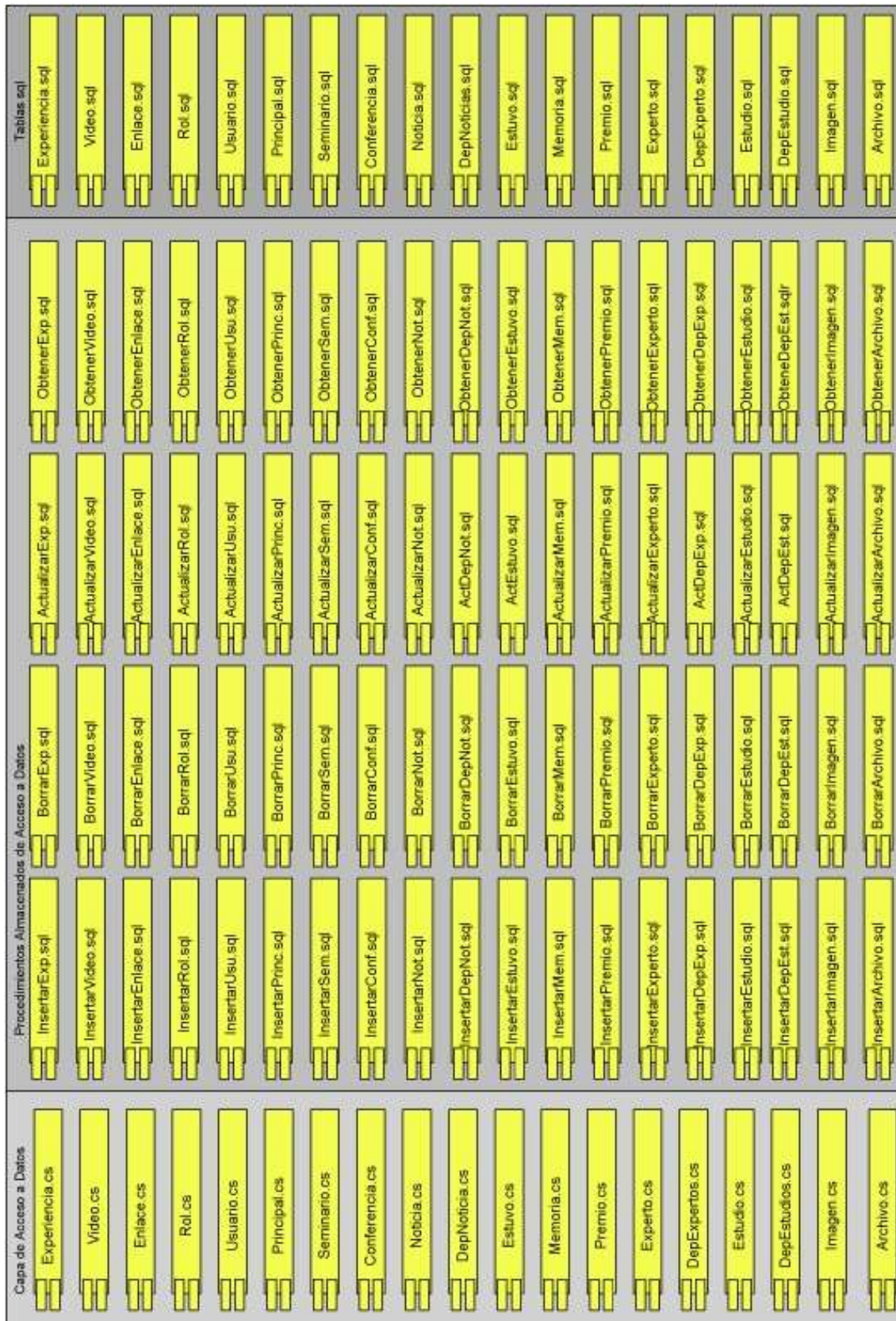


Figura 5.10. Diagrama de componentes 2 de la aplicación de gestión.

5.2.1.2 - Diagramas de clases.

En el diagrama de clases vamos a representar los métodos implementados por cada una de ellas, que en conjunto forman la lógica de negocio de la aplicación, y la relación de los mismos con la clase Utilidades, que proporciona otros métodos a los que acceden las demás clases.

Los métodos implementados por la clase Utilidades que usan el resto de las clases que componen la aplicación son los siguientes:

Métodos
Métodos de la clase Utilidades.cs
mandarMail(string destino, string origen, asunto, cuerpo del mensaje) – Envía un mensaje de correo electrónico con los parámetros de entrada obtenidos.
Conectar(): SqlConnection - Contiene la cadena de conexión a la base de datos de la Universidad CEU Cardenal Herrera.
GuardarError(string ruta, Exception ex) – Si se produce algún error de conexión con la base de datos o con los procedimientos almacenados SQL durante la ejecución de la aplicación, este método escribe dicho error en un fichero de log almacenado en el servidor.
ValidarSesion(Usuario usuario) : bool – Compara las credenciales buscando coincidencias en la tabla Usuario. En caso afirmativo devuelve true.
Métodos del resto de clases
RellenarCampos(Objeto) – Rellena los campos del formulario una vez se ha accedido a un objeto para su visualización o edición.
LimpiarCampos(Objeto) – Limpia los campos del formulario una vez se ha borrado un objeto de la base de datos.
GuardarObjeto(Objeto) – Accede a la capa de acceso a los datos para insertar el nuevo objeto en la tabla correspondiente.
ModificarObjeto(Objeto) – Accede a la capa de acceso a los datos para actualizar un objeto existente.
BorrarObjeto(Objeto) – Accede a la capa de acceso a los datos para borrar un objeto de su tabla correspondiente.
BuscarObjeto(Objeto, String filtro) – Escribe una consulta, filtrando por la palabra que recibe como parámetro de entrada, y accede a la capa de acceso a los datos para realizar dicha consulta y obtener el objeto precisado.
RellenarDesplegable() – Accede a la capa de acceso a los datos para listar los elementos a mostrar en un desplegable.
RellenarHistorico() - Accede a la capa de acceso a los datos para realizar una consulta de todas las entradas de la tabla correspondiente y mostrarlas al usuario.

Diagrama de Clases - Aplicación de gestión del Observatorio.

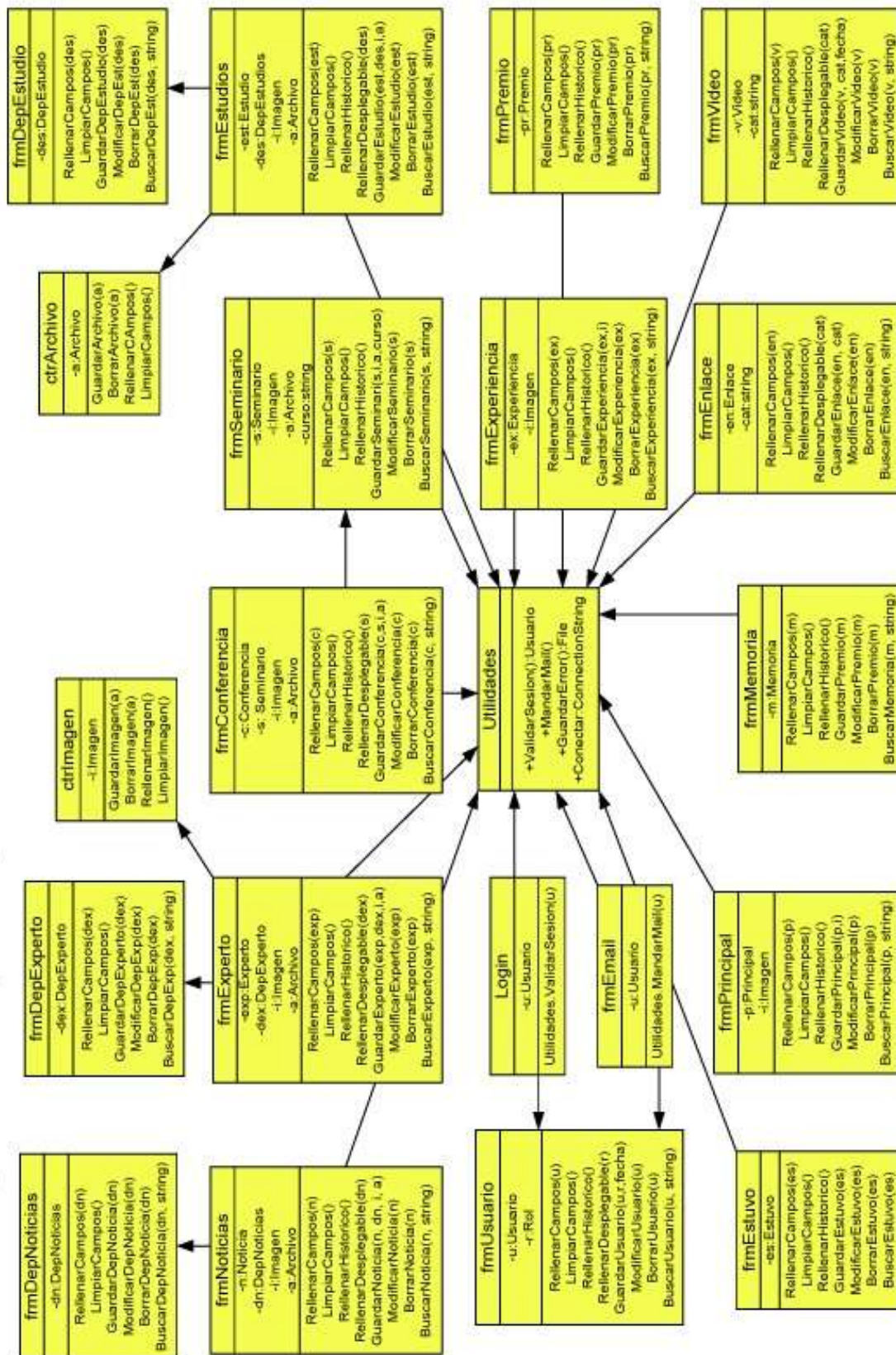


Figura 5.11. Diagrama de clases de la aplicación de gestión.

5.2.1.3 - Diagramas de secuencia.

A continuación vamos a ver las secuencias más representativas de la aplicación de gestión. No vamos a mostrar todas ya que esta aplicación web contiene muchos apartados independientes y hay gran paralelismo entre las operaciones que se pueden realizar en cada uno.

En primer lugar, representamos la secuencia de recuperación de las credenciales de un usuario no identificado que ha olvidado su contraseña.

Este usuario accederá a un formulario que solicitará su dirección de correo electrónico, y tras hacer una petición de lectura a la base de datos para confirmar la existencia de dicha dirección, enviará un email al mismo con el nombre de usuario y la contraseña para identificarse.

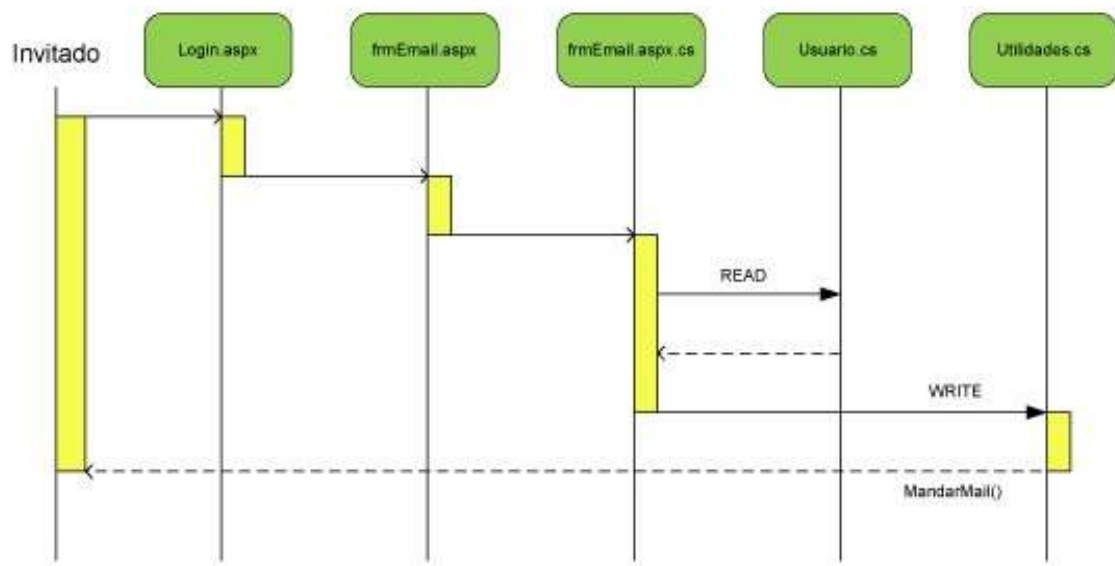


Figura 5.12. Diagrama de secuencia. Recuperación de credenciales.

En 2º lugar representamos la secuencia de introducción de un nuevo usuario con privilegios para gestionar la aplicación. Para ello accedemos al panel de gestión de usuarios, rellenamos los campos con sus datos, incluido el rol, a seleccionar de un desplegable estático, y la especificación de los apartados a los que tendrá acceso si es "colaborador". Finalmente guardamos, insertando los datos correspondientes a los privilegios en la tabla de Roles y el resto de datos personales en la de Usuarios.

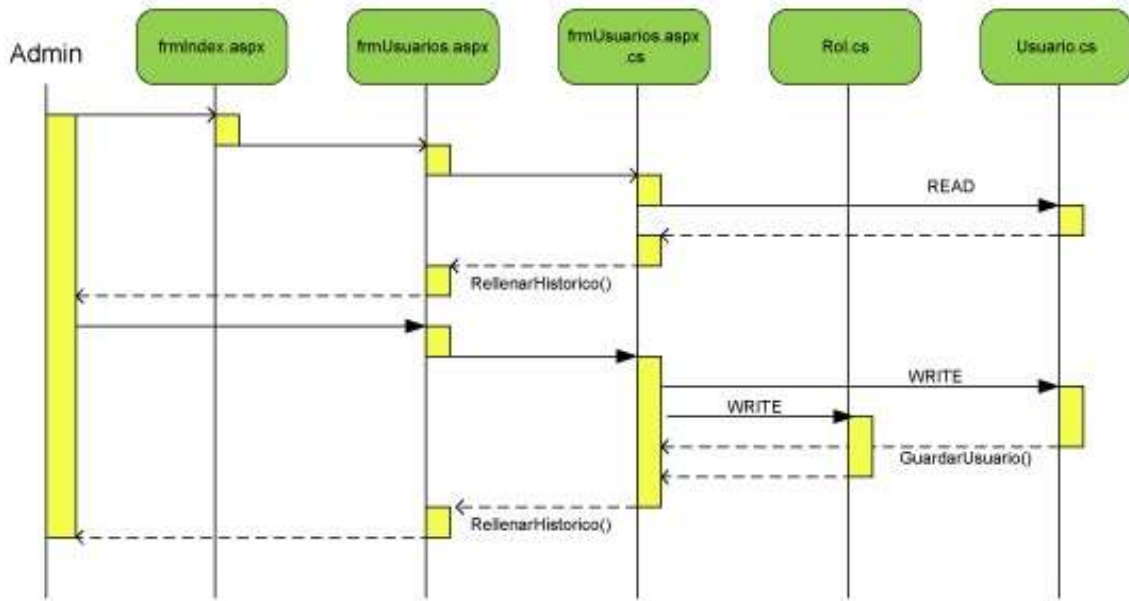


Figura 5.13 Diagrama de secuencia. Introducción de nuevo usuario.

En tercer lugar representamos la secuencia de creación de una conferencia, asignándole un seminario ya existente. Para ello accedemos al panel de conferencias, rellenamos los campos del formulario y seleccionamos un seminario en el desplegable (el contenido del desplegable se lee dinámicamente desde la tabla de seminarios). Finalmente guardamos la conferencia en la base de datos.

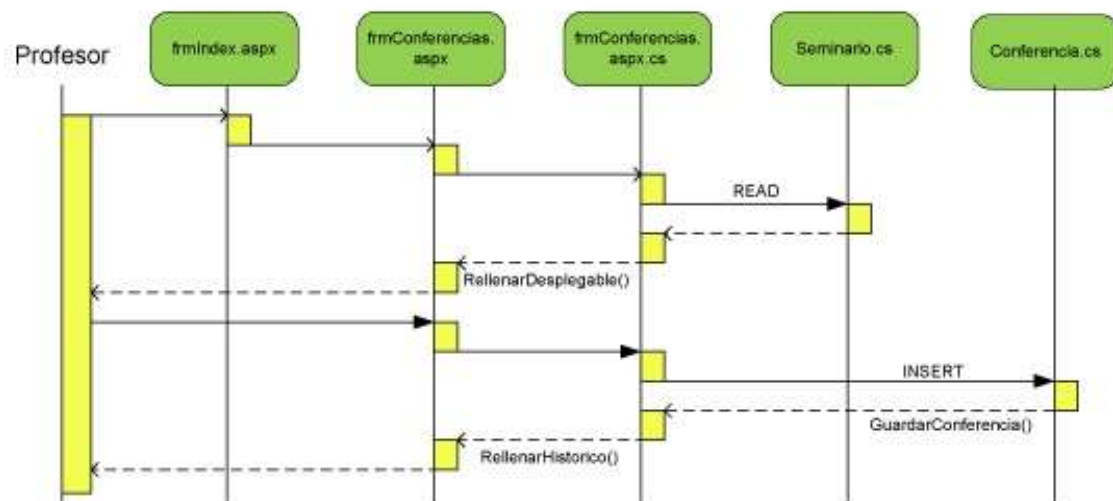


Figura 5.14 Diagrama de secuencia. Creación de conferencia con seminario asignado.

En 4º lugar representamos cómo añadir un nuevo Estudio al que le incluiremos un Archivo adjunto. Para ello accedemos al panel de Estudios y rellenamos los datos. La sección de estudios a la que asociaremos este estudio la seleccionamos a través de un desplegable cuyo contenido se obtiene dinámicamente desde la base de datos. Posteriormente seleccionamos un archivo a subir desde nuestro ordenador y finalmente guardamos el Estudio en la base de datos.

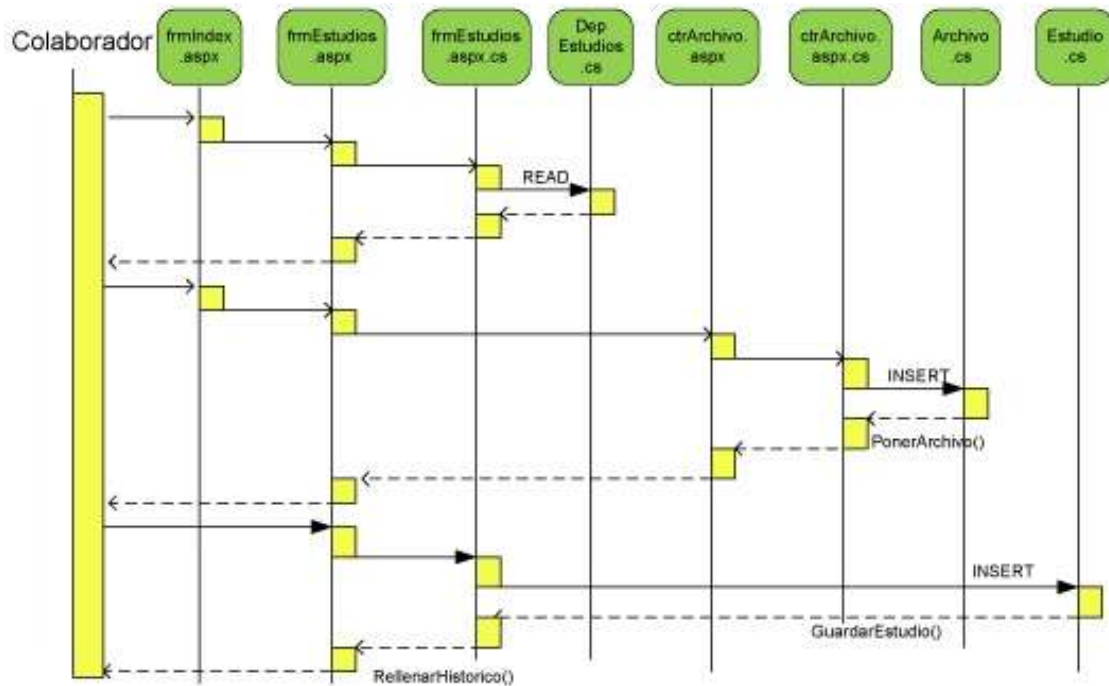


Figura 5.15 Diagrama de secuencia. Crear un estudio con un archivo adjunto.

En 5º lugar representamos la secuencia “borrar un Experto y su imagen”. Para ello entramos en el panel de Expertos, seleccionamos el experto a borrar del listado de históricos y hacemos clic en el botón borrar. Automáticamente se procederá al borrado del experto de la tabla de expertos y de la imagen de la tabla de imágenes.

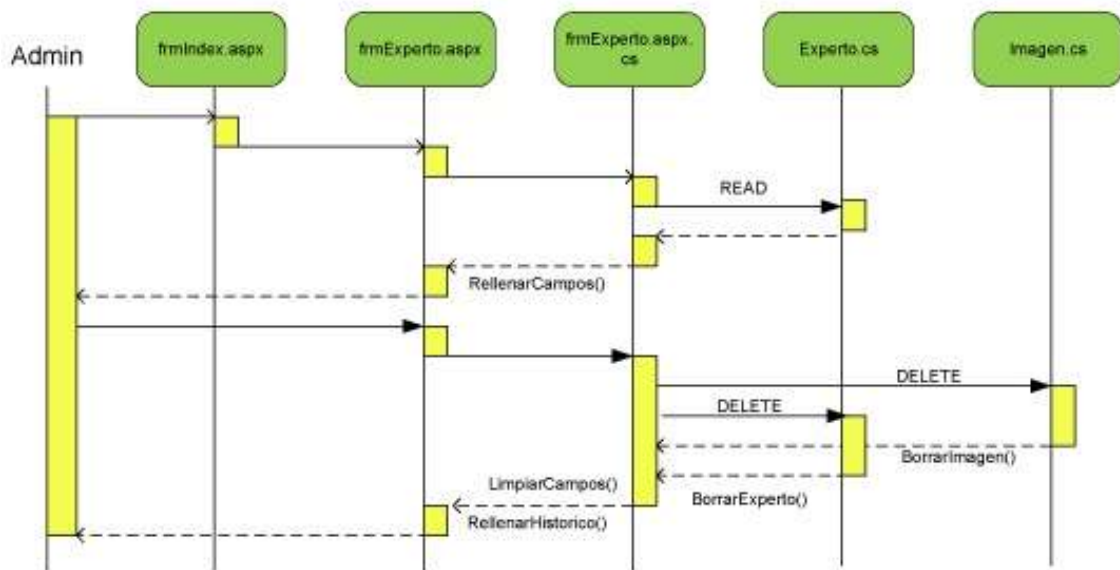


Figura 5.16 Diagrama de secuencia. Borrar un experto y su imagen.

Y por último representamos la secuencia “buscar noticia mediante buscador y cambiar estado”. Para ellos accedemos al panel de Noticias y escribimos una palabra o frase por la que filtrar su título. A continuación se hará una consulta a la base de datos con ese filtro y si encuentra una o más noticias que cumplan esos requisitos, aparecerán en el listado de históricos. Hacemos clic en el botón editar y automáticamente se rellenarán los campos con los datos de la noticia. Entonces podemos modificar el campo estado para hacer la Noticia “no visible”. Finalmente guardamos la Noticia, produciéndose la actualización en la base de datos.

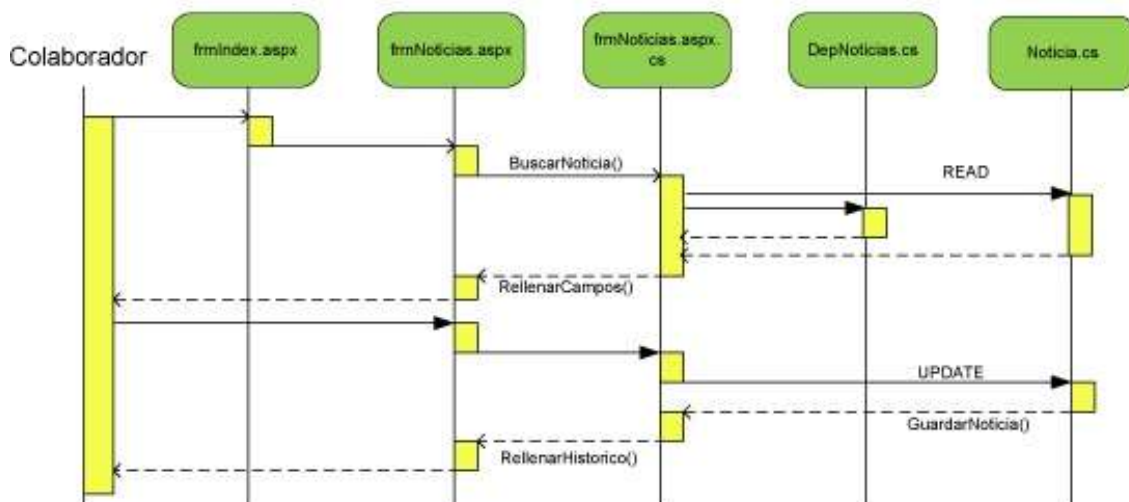


Figura 5.17 Diagrama de secuencia. Buscar noticia y cambiar estado.

5.2.2 - Sitio web del Observatorio.

5.2.2.1 - Diagramas de componentes.

Contenedor Web.

En la visualización se va a diseñar una estructura un tanto distinta. El planteamiento es el siguiente: cada página <<aspx>> está compuesta por un buen número de controles <<ascx>> “embebidos” que proporcionan una funcionalidad en concreto. Muchos de esos controles deben ser accesibles desde la mayoría de las páginas, pero todas las páginas incorporan un control específico que aportan esa funcionalidad que la diferencia de las demás.

A continuación se explicará únicamente la funcionalidad que diferencia una página <<aspx>> de las demás, indicando el control o controles que le dan dicha funcionalidad y posteriormente se expondrá la utilidad del resto de controles <<ascx>>.

Capa de presentación

Active Server Pages <<aspx>> y sus Active Server Control <<ascx>> asociados.

Index.aspx - Visualiza el contenido actual de la página principal y un histórico de los títulos de las páginas principales anteriores. Cada título enlaza con la visualización de la página principal del histórico correspondiente. Muestra la página principal actual a través de ctrPrincipal.ascx las anteriores a través de ctrPrincipalAnterior.ascx.

Expertos.aspx – Visualiza un listado de los títulos del apartado de expertos, filtrado por distintas categorías, y enlaza cada uno con su contenido. Muestra el listado de especialidades a través de ctrMenuExpertos.ascx, y el listado de expertos por especialidad a través de ctrExpertos.ascx. Este mismo control se encarga de visualizar el contenido de la entrevista a dicho experto una vez se hace click en el enlace a la misma.

Estudios.aspx - Visualiza un listado de los títulos de la sección de estudios, filtrado por categoría, y enlaza cada uno con su contenido. Muestra el listado de las secciones a través de ctrMenuEstudios.ascx, y el listado de expertos por especialidad a través de ctrEstudios.ascx. Este mismo control se encarga de visualizar el contenido del estudio a través de su enlace.

Seminarios.aspx – Visualiza un listado de seminarios, jornadas y congresos, filtrado por curso académico, y enlaza cada uno con su contenido. Muestra el listado de los cursos académicos a través del control ctrMenuSeminarios.ascx, el listado de seminarios por cada curso a través de ctrSeminarios.ascx y el contenido de cada seminario a través de ctrSeminarioInfo.ascx.

Experiencias.aspx - Visualiza el contenido actual de la sección de experiencias creativas y un histórico de los títulos de las experiencias anteriores. Cada título enlaza con la visualización del contenido del histórico correspondiente. Cada título enlaza con la visualización de la página de experiencias del histórico correspondiente. Muestra la experiencia creativa actual a través de ctrExperiencias.ascx las anteriores a través de ctrExperienciaAnterior.ascx.

Estuvo.aspx - Visualiza un listado de los títulos del apartado “El observatorio estuvo en”, y enlaza cada uno con su contenido. Muestra el listado de títulos a través del control ctrlEstuvo.ascx, y al hacer click sobre el enlace, muestra el contenido de este apartado a través del control ctrEstuvoInfo.ascx.

Objetivos.aspx – Texto estático que explica los objetivos. Lo muestra a través de ctrObjetivos.ascx.

Memorias.aspx - Visualiza un listado de las memorias de cada curso académico, enlazando cada una con su contenido. Muestra el listado de cursos académicos a través de ctrlListaMemorias.ascx y el listado de las memorias a través de ctrMemorias.ascx.

Equipo.aspx – Muestra un enlace de los nombres de usuarios registrados desde la aplicación y

marcados como visibles. Los nombres enlazan con la información almacenada del usuario correspondiente. Lo muestra a través de `ctrListaEquipo.ascx`, mientras que a través de `ctrEquipo.ascx`, se muestra información de cada usuario enlazado.

Premios.aspx - Visualiza un listado de los premios de cada curso académico, enlazando cada uno con su contenido. Muestra el listado de cursos académicos a través de `ctrListaPremios.ascx` y el listado de las memorias a través de `ctrPremios.ascx`.

mediosCEU.aspx – Visualiza un listado de los vídeos – nombre, url, descripción, y vídeo embebido desde youtube - cuya categoría es “Observamos el sector”. Para acceder, encontramos un control en todas las páginas, llamado `ctrMedios.ascx` que muestra un vídeo embebido y un enlace al listado de vídeos. El control a través del cual se muestran estos contenidos es `ctrMedios_todos.ascx`.

NuestrosVideos.aspx – Visualiza un listado de los vídeos – nombre, url, descripción, y vídeo embebido desde youtube - cuya categoría es “Nuestros vídeos”. Para acceder, encontramos un control en todas las páginas, llamado `ctrNuestrosVideos.ascx` que muestra un vídeo embebido y un enlace al listado de vídeos. El control a través del cual se muestran estos contenidos es `ctrNuestrosVideos_todos.ascx`.

Contacta.aspx – Muestra un formulario de contacto a través del control `ctrContactar.ascx`.

Enlaces.aspx - Visualiza un listado de los enlaces y directorios, filtrados por categoría, y enlaza cada uno con la página o directorio destino. Tanto el menú de secciones para enlaces como los propios enlaces los proporciona el control `ctrEnlaces.ascx`.

Noticias.aspx - Visualiza un listado de los títulos del apartado de noticias, filtrado por distintas secciones, y enlaza cada una con su contenido. Las secciones las proporciona el control `ctrListaNoticias.ascx`, el listado de títulos de las noticias lo muestra el control `ctrNoticias.ascx` y el contenido de cada noticia es mostrado por `ctrNoticiaInfo.ascx`.

Partners.aspx – Imágenes estáticas de los logos de los partners y enlaces a las páginas web de los mismos.

Observador_admirado.aspx - Visualiza, a partir de su RSS, las últimas entradas de la categoría Observador Admirado del blog de publicidad y relaciones públicas de la Universidad CEU Cardenal Herrera.

RSS.aspx – Visualiza una página con formato XML – RSS estándar que contiene la redifusión de las últimas noticias sobre el observatorio para quien desee realizar la subscripción.

Buscar.aspx – Visualiza un listado de ítems que contengan el término de búsqueda por el cual el usuario ha decidido filtrar los contenidos.

A través de `ctrBuscar.ascx` se escriben las palabras que servirán de filtrado y los resultados se muestran gracias al control `ctrBuscarInfo.ascx`.

Mapasitio.aspx – Muestra un listado jerarquizado o mapa del sitio con todos los ítems que componen el sitio web. Este contenido se visualiza a través del control `ctrMapaSitio.ascx`.

Otros Active Server Control <<ascx>>

CtrCabecera.ascx y ctrMenuNosotros.ascx – Visualizan los menús principales superior y de la parte derecha de la página, enlazando a las distintas secciones del sitio web.

La lógica de negocio de la web del observatorio se encuentra principalmente en los Active Server Control C#, ya que, como se ha comentado antes, se ha delegado prácticamente toda la funcionalidad del sitio sobre estos controles, que se van a encargar de efectuar las llamadas a la base de datos para obtener la información insertada desde la aplicación de gestión.

Capa de negocio

Active Server Control Extended C# <<ascx.cs>>

ctrPrincipal.ascx.cs – Obtiene en primer lugar la última página principal desde la capa de acceso a los datos. Obtiene también el histórico de páginas principales, listando los títulos y los enlaces a sus contenidos. Crea las url's para visualizar las páginas principales anteriores añadiendo a la URL inicial el ID de cada entrada de la tabla.

ctrPrincipalAnterior.ascx.cs – Obtiene la página principal desde la capa de acceso a los datos en función del ID que lee en la URL creada en ctrPrincipal.ascx.cs.

ctrMenuExpertos.ascx.cs – Obtiene todas las secciones de expertos desde la capa de acceso a los datos y lista en una tabla los títulos y sus enlaces, creando la url para cada enlace a partir de la url original y añadiendo la cadena “seccion” y el ID de la sección.

ctrExpertos.ascx.cs – Si la URL contiene el ID del experto, lo extrae. A partir del id obtiene el título, el archivo, la imagen y el texto de la entrevista a dicho experto desde la capa de acceso a los datos. Se hace uso del título también para insertarlo dinámicamente en la etiqueta “title” del html por razones de SEO, ya que los “titles” de las páginas web son muchas veces un factor clave para que los motores de búsqueda puedan indexar una web y de esa forma escalar en el ranking de búsqueda.

Si la URL no contiene el ID del experto, pero sí el de la sección, obtiene todos los expertos englobados en esa sección, a partir del ID de la sección y a través de la capa de acceso a los datos. Muestra una lista con los títulos de cada entrevista al experto y un enlace al contenido, creando la URL añadiendo a la original la cadena “&” y el ID del experto.

Si la URL no contiene ni el ID del experto ni el de la sección, se muestra el listado de secciones.

ctrMenuEstudios.ascx.cs - Obtiene las secciones de estudios y lista en una tabla los títulos y sus enlaces, añadiendo a la url original la cadena “seccion” y el ID de la sección.

ctrEstudios.ascx.cs – Obtiene todos los estudios y todas las secciones y las muestra en función de los parámetros leídos en la URL, como en ctrExpertos.ascx.cs.

ctrMenuSeminario.ascx.cs – Obtiene un listado de los cursos académicos desde la capa de acceso a los datos.

ctrSeminarios.ascx.cs - Obtiene todos los seminarios y los cursos académicos y los lista en función de los parámetros leídos en la URL, como en ctrExpertos.ascx.cs. Obtiene también, desde la capa de acceso a los datos, cada conferencia asociada a un seminario a partir del ID del seminario, generando la URL que enlazará a dicha conferencia añadiendo a la URL original la cadena “&menu=conferencias&conferencia=” y el ID de la conferencia.

ctrSeminarioInfo.ascx.cs – Obtiene el seminario y su conferencia asociada desde la capa de acceso a los datos en función del ID que lee en la URL creada en ctrSeminario.ascx.cs.

ctrExperiencias.ascx.cs – Obtiene la última experiencia y el histórico de experiencias creativas, listando los títulos y los enlaces a sus contenidos. Crea las url's para visualizar las experiencias anteriores añadiendo a la URL inicial el ID de cada entrada de la tabla.

ctrExperienciaAnterior.ascx.cs - Obtiene la experiencia creativa desde la capa de acceso a los datos en función del ID que lee en la URL creada en ctrExperiencias.ascx.cs.

ctrEstuvo.ascx.cs – Obtiene todas las entradas de esta sección desde la capa de acceso a datos y lista sus títulos y los enlaces a su contenido.

Crea las URL's de estos enlaces añadiendo la ID de las entradas correspondientes y genera también los “titles” dinámicamente.

ctrEstuvoInfo.ascx.cs – Obtiene título, texto, archivo e imagen de la entrada a partir de la ID

leía en la URL que se ha creado en `ctrEstuvo.ascx.cs` y el método llamado desde la capa de acceso a datos.

ctrListaMemorias.ascx.cs - Obtiene un listado de las memorias y los enlaces a su contenido.

ctrMemorias.ascx.cs – Obtiene el contenido de las memorias a partir del ID del enlace correspondiente. Genera “titles” dinámicos con cada memoria.

ctrListaEquipo.ascx.cs – Obtiene un listado de los usuarios y los enlaces a sus datos.

ctrEquipo.ascx.cs – Obtiene los datos del usuario a partir del ID del enlace. Genera “titles” dinámicos con cada usuario.

ctrListaPremios.ascx.cs - Obtiene un listado de los premios y los enlaces a su contenido.

ctrPremios.ascx.cs - Obtiene la descripción de los premios a partir del ID del enlace correspondiente. Genera “titles” dinámicos con cada premio.

ctrMedios.ascx.cs – Obtiene todos los vídeos filtrados por la categoría “Observamos el sector” desde la capa de acceso a datos, selecciona el primero de la misma y lo devuelve embebido en código html.

ctrMediosTodos.ascx.cs – Obtiene un listado de todos los vídeos filtrados por la categoría “observamos el sector”, embebidos en código html junto a su título y a una descripción de cada uno.

ctrNuestrosVideos.ascx.cs – Obtiene todos los vídeos filtrados por la categoría “Nuestros Vídeos” desde la capa de acceso a datos, selecciona el primero de la misma y lo devuelve embebido en código html.

ctrNuestrosVideosTodos.ascx.cs – Obtiene un listado de todos los vídeos filtrados por la categoría “Nuestros Vídeos”, embebidos en código html junto a su título y a una descripción de cada uno.

ctrContactar.ascx.cs – Recoge los datos escritos en el formulario y envía un mail a las direcciones de correo indicadas.

ctrEnlaces.ascx.cs – Obtiene las diferentes categorías de los enlaces y los devuelve en forma de menú. Además concatena el nombre de la sección enlazada a la url original y obtiene los enlaces en función de dicha categoría.

ctrListaNoticias.ascx.cs - Obtiene un listado de las noticias y los enlaces a su contenido, concatenando el ID de cada noticia enlazada a la url.

ctrNoticias.ascx.cs – Obtiene las tres últimas noticias desde la capa de acceso a datos y devuelve una subcadena de máximo 50 caracteres para ser mostrada en la página principal.

ctrNoticialInfo.ascx.cs – Recoge el ID de la noticia a través de la url y muestra los contenidos de cada noticia: título, subtítulo, texto, archivo adjunto e imagen.

ctrBuscar.ascx.cs – Concatena el término insertado en el cuadro de texto con la url inicial y redirecciona a dicha url.

ctrBuscarInfo.ascx.cs - Realiza búsquedas, filtradas por el término dado por el usuario, de todas las entradas agrupadas por cada uno de los apartados, mostrando su título y un enlace a dicha entrada. Muestra además un enlace al contenido de cada una.

ctrEnlaces.ascx.cs – Obtiene un menú de categorías en los que agrupar los distintos enlaces. Cada categoría enlaza con un listado de enlaces agrupados por categoría, y cada uno de esos enlaces redirige a una página externa.

ctrContactar.ascx.cs – Recoge los datos del formulario de contacto mostrado en la web y los envía por correo electrónico a los responsables del Observatorio.

ctrObservadorAdmirado.ascx.cs – Lee un feed RSS de la categoría Observador Admirado de uno de los blogs de la Universidad CEU Cardenal Herrera. Extrae título, link, un pequeño extracto de la información, y finalmente la entrada completa a través de una clase que serializa la información y finalmente la visualiza en la página web, sin necesidad de haber accedido a la base de datos relacional.

ctrMapaSitio.ascx.cs – Genera un fichero XML, específico para ASP.NET que contendrá el mapa del sitio de toda la web. Cada nodo “<siteMapNode>” tiene un atributo “url” que contiene el enlace a cada una de las entradas, un atributo “title” y un atributo “description”.

Estos atributos se rellenarán con el contenido obtenido al acceder a los campos correspondientes de la base de datos del Observatorio. Este fichero permite obtener una estructura jerarquizada de todo el sitio web que puede facilitar el acceso a cualquier parte del mapa, y además mejora la indexación y el posicionamiento del sitio.

Contenedor de componentes de acceso a la base de datos.

Como se ha comentado antes, el sitio web y la aplicación de gestión comparten los mismos componentes de acceso a la base de datos. Sin embargo, en este caso sólo se llama a procedimientos almacenados de consulta específicos para cada caso, que se enumeran a continuación. En este apartado, además, se habla de dos componentes nuevos, que son los que se encargan de usar como fuente de datos ficheros con formato RSS-XML en lugar la base de datos relacional.

Capa de acceso a datos

Clases <<.cs>> y procedimientos almacenados a los que llaman en la aplicación de gestión.

RSS.cs – Hace uso de la librería XML de ASP.NET para crear objetos de tipo XML en el que definimos los nodos del formato RSS para almacenar y extraer la información como si se tratase de una base de datos.

RSSBLL.cs – Contiene dos métodos. Por una parte presenta un método de escritura que recibe como parámetro de entrada un objeto RSS, serializa la información y la escribe en un fichero de texto con formato RSS al que le añade la cabecera correspondiente. Por otra tiene un método de lectura que recibe la url del feed que queremos leer, extrae la información, la deserializa y la almacena en un objeto RSS.

Principal.cs – Obtiene el listado de históricos de páginas principales a través de la llamada al procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Principal_todos]". Además accede a la entrada más reciente mediante el procedimiento "dbo.[sp_obtener_WEB_OBSERVATORIO_Principal_Ultimo] y a cualquier entrada anterior filtrada por su identificador mediante "dbo.[sp_obtener_WEB_Observatorio_Principal]".

Noticia.cs – Obtiene el listado de noticias publicadas a través de la llamada al procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Noticias_todos]". También contiene un procedimiento que obtiene únicamente las 3 últimas noticias, "dbo.[sp_obtener_WEB_Observatorio_Noticias_3primeros]" y un procedimiento "dbo.[sp_obtener_WEB_Observatorio_Noticia]" que obtiene una noticia concreta a través de su ID. Por otra parte, este componente también contiene un método que recoge los datos necesarios del procedimiento dbo.[sp_obtener_WEB_Observatorio_Noticias_todosRSS] y genera un fichero con formato RSS.

Seccion_Noticias.cs – Proporciona el listado de secciones de noticias mediante el procedimiento "dbo.[sp_obtener_WEB_Observatorio_Secciones_Noticias_todos]" y una sección concreta a través de su id mediante "dbo.[sp_obtener_WEB_Observatorio_Seccion_Noticias]".

Memoria.cs – Obtiene el listado de memorias a través de la llamada al procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Memorias_todos]" así como la entrada filtrada por su identificador mediante el procedimiento "dbo.[sp_obtener_WEB_Observatorio_Memoria]".

Experto.cs – Obtiene el listado de expertos a través de la llamada al procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Expertos_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Experto]". Se pueden obtener también los expertos por cada sección mediante "dbo.[sp_obtener_WEB_Observatorio_Expertos_porSeccion]".

Seccion_Expertos.cs – Proporciona el listado de secciones de expertos mediante el

procedimiento "dbo.[sp_obtener_WEB_Observatorio_Secciones_Expertos_todos]" y una sección concreta a través de su id mediante el procedimiento "dbo.[sp_obtener_WEB_Observatorio_Seccion_Expertos]".

Estudio.cs – Obtiene el listado de estudios a través de la llamada al procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Estudios_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Estudio]". Se pueden obtener también los expertos por cada sección mediante "dbo.[sp_obtener_WEB_Observatorio_Estudios_porSeccion]".

Seccion_Estudios.cs – Proporciona el listado de secciones de estudios mediante el procedimiento "dbo.[sp_obtener_WEB_Observatorio_Secciones_Estudios_todos]" y una sección concreta a través de su id mediante el procedimiento "dbo.[sp_obtener_WEB_Observatorio_Seccion_Estudios]".

Enlace.cs – Obtiene el listado de enlaces a través de la llamada al procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Enlaces_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Enlace]".

Usuario.cs – Obtiene el listado de usuarios registrados a través de la llamada al procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Usuarios_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Usuario]".

Video.cs – Obtiene el listado de vídeos a través de la llamada al procedimiento "dbo.[sp_obtener_WEB_Observatorio_Videos_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Video]".

Premios.cs – Obtiene el listado de premios a través de la llamada al procedimiento "dbo.[sp_obtener_WEB_Observatorio_Premios_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Premio]".

Experiencias.cs – Obtiene el listado de experiencias creativas a través de la llamada al procedimiento "dbo.[sp_obtener_WEB_Observatorio_Experiencias_todos]". Además accede a la entrada más reciente mediante el procedimiento "dbo.[sp_obtener_WEB_Observatorio_Experiencia_Ultima]" y a cualquier entrada anterior filtrada por su identificador mediante "dbo.[sp_obtener_WEB_Observatorio_Experiencia]".

Seminario.cs – Obtiene el listado de seminarios a través de la llamada al procedimiento "dbo.[sp_obtener_WEB_Observatorio_Seminarios_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Seminario]". También puede listar los seminarios por fecha mediante "dbo.[sp_obtener_WEB_Observatorio_Seminarios_porFecha]".

Conferencia.cs – Obtiene la conferencia correspondiente a un seminario a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Conferencia_Seminario]".

Estuvo.cs – Obtiene el listado de contenidos a través de la llamada al procedimiento "dbo.[sp_obtener_WEB_Observatorio_Estuvo_todos]" así como la entrada dada a través de su identificador mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Estuvo]".

Archivo.cs – Este componente obtiene los datos de los archivos correspondientes a una entrada a través del ID del archivo mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Archivos]".

Imagen.cs - Este componente obtiene los datos de las imágenes correspondientes a una entrada a través del ID de la imagen mediante el procedimiento almacenado "dbo.[sp_obtener_WEB_Observatorio_Imagenes]".

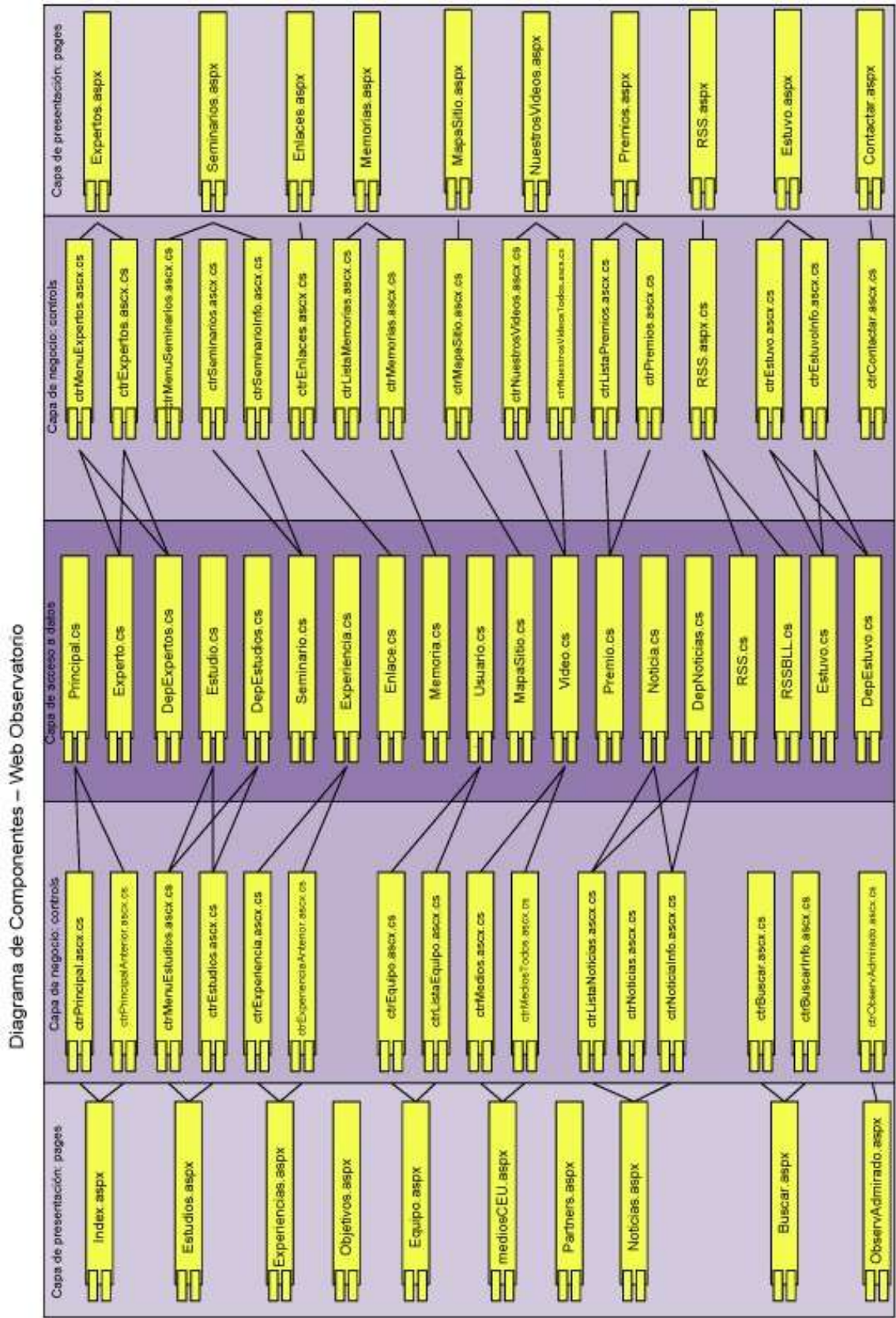


Figura 5.18 Diagrama de componentes 1 del sitio web.

5.2.2.2 - Diagramas de clases.

De igual forma que se ha hecho en la aplicación de gestión, en el diagrama de clases se representan aquellos elementos que contienen los métodos que gestionan la lógica de negocio.

En la web del observatorio se encuentran la mayoría de los métodos dentro de los controles que se han definido en la capa de negocio del diagrama de componentes. De forma similar a lo ocurrido en la aplicación de gestión, varios de los métodos se repiten en buena parte de las clases, ya que acceden a tablas independientes de la base de datos. Por esa razón se procede a explicar los métodos de forma genérica y luego a mostrar el diagrama completo.

Métodos
ObtenerTodos() – Lista todas las entradas de la tabla de la base de datos a la que se pide acceso desde la capa de acceso a datos. Esta lista se guarda en un DataView para poder acceder a todos los elementos y a sus campos.
ObtenerÚltimo() – Obtiene el último elemento insertado en la tabla a la que se pide acceso y lo almacena en un objeto del tipo correspondiente.
Obtener(idObjeto) – Obtiene todos los campos del elemento que se corresponde con la id obtenida como parámetro de entrada. Dicho id se encuentra concatenado a la Url de la página.
ObtenerPorSeccion(idSeccion) – Lista un conjunto de entradas si su campo “sección” de la tabla coincide con la sección o departamento especificado en la consulta.
ObtenerSecciones() – Lista todas las entradas de la tabla secciones correspondiente.
ObtenerPorCurso(curso) – De igual forma se listan un conjunto de entradas si su campo “curso” de la tabla coincide con el curso académico que se especifica en la consulta.
ObtenerCursos() – Realiza una consulta al campo “curso” de la tabla de la base de datos correspondiente buscando y listando todos los cursos académicos no repetidos que encuentra. De esta forma no se listan cursos académicos vacíos.
ponerNombre() – Aquellas clases que contienen una lista de elementos obtenida con el método ObtenerTodos(), muestran el campo “nombre” de ese objeto llamando a este método.
ponerUrl() – Esas mismas clases llaman a este método para asociar una Url a cada nombre. Esta Url se genera concatenando a la Url de la página actual, la cadena “&id=” y el id de dicha entrada. De esta forma se puede tomar este id para mostrar todos los campos del objeto mediante el método Obtener(idObjeto).
ponerTitleDinamico(Objeto) – Este método obtiene el campo nombre de cada entrada de la base de datos y lo escribe en la etiqueta “title” de la página html. Al escribir los title de forma dinámica conseguiremos una mayor indexación del sitio en la web y por tanto un mejor posicionamiento.
LeerRSS() – Lee de un fichero de texto con formato XML – RSS y guarda el contenido serializado en un objeto RSS, para posteriormente mostrar dicha información en la página web correspondiente.
EscribirRSS() – Genera un fichero de texto con formato XML – RSS a partir de los datos obtenidos mediante una consulta específica a la base de datos correspondiente. Ese fichero será utilizado posteriormente como sindicador de contenidos mediante el lector de feeds RSS de la web.
GenerarCadena(MapaSitio) – Genera dinámicamente un fichero con formato XML SiteMap específico para ASP.NET con todas las entradas de todas las tablas de la base de datos de la aplicación. Mediante este fichero tenemos un mapa completo del sitio, mediante el cual podemos: mostrar el mapa completo de la web, mostrar la ruta de acceso a cada uno de los apartados y mejorar aún más la indexación, y por tanto el posicionamiento, de nuestro sitio

en la web.

Redireccionar(EscribirUrlConFiltro) – Concatena la cadena indicada en el buscador a la URL básica y nos redirecciona a dicha URL. A través de este parámetro se ejecutará la consulta a la base de datos para obtener todos los resultados que pasen el filtro indicado por el usuario.

EnviarMail() – Envía un mail a la dirección de correo indicada en el método con el contenido escrito por el usuario en el formulario de contacto facilitado.

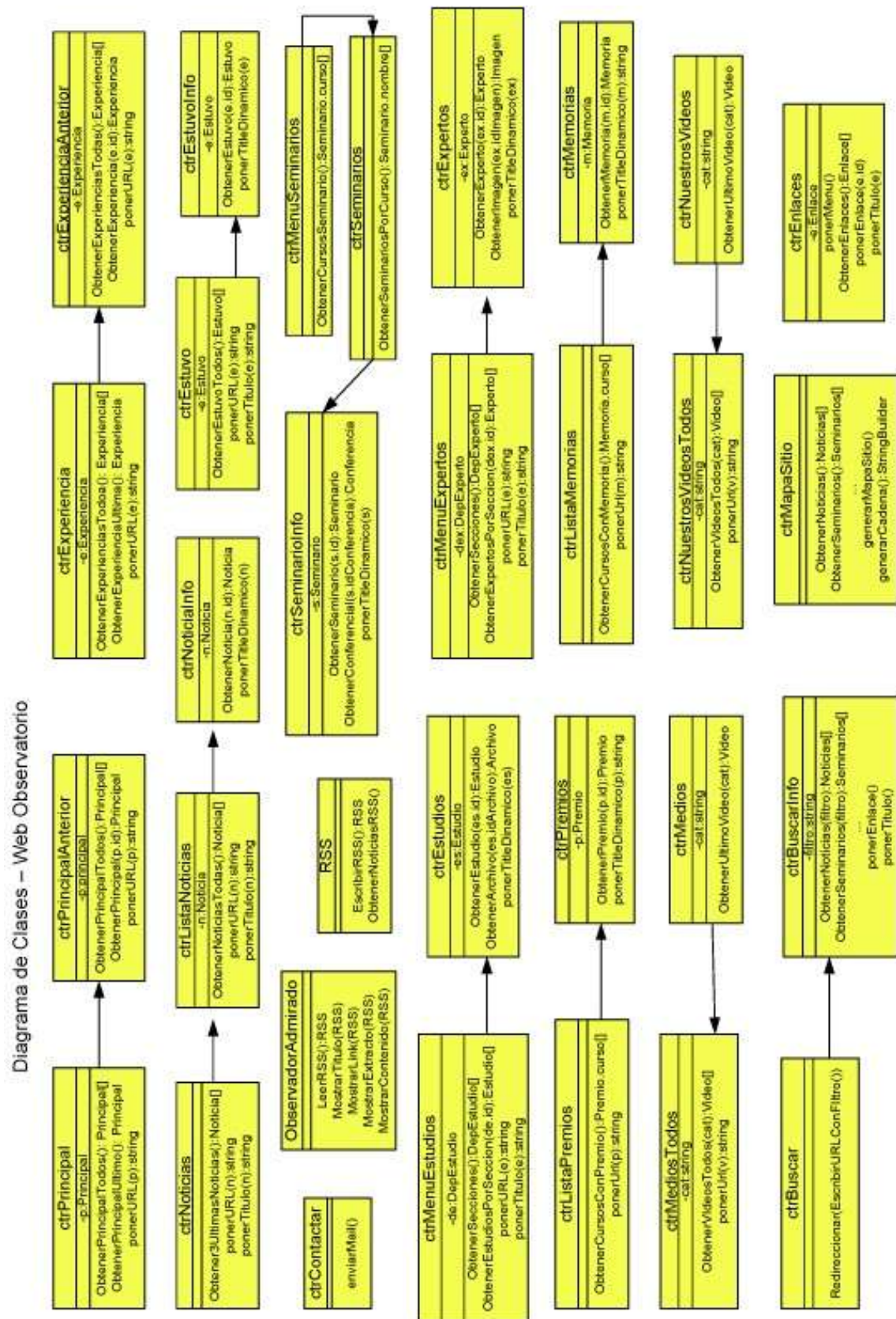


Figura 5.19 Diagrama de clases del sitio web.

5.2.2.3 - Diagramas de secuencia.

Tal y como ocurría en la aplicación de gestión, hay un gran paralelismo entre muchas de las secuencias que se pueden realizar en el sitio web, debido al número de secciones independientes que contiene, por esa razón, se han seleccionado solamente las cuatro más representativas.

En primer lugar se procede a enseñar cómo descargar al archivo adjunto a un estudio. Para ello se accede a la página principal, que carga la última entrada contenida en la base de datos. Se enlaza con la sección de Estudios, donde se puede ver un listado con todas las secciones de los estudios disponibles, obtenidas también desde la tabla de la base de datos correspondiente, ya que las secciones pueden ser modificadas desde la aplicación de gestión. Cada sección enlazará con el listado de Estudios que pertenezcan a la misma. Los títulos de cada estudio contendrán un enlace al contenido de dicho estudio, y como parte de ese contenido se podrá encontrar un enlace a un archivo adjunto, que podrá ser descargado al hacer clic sobre el mismo.

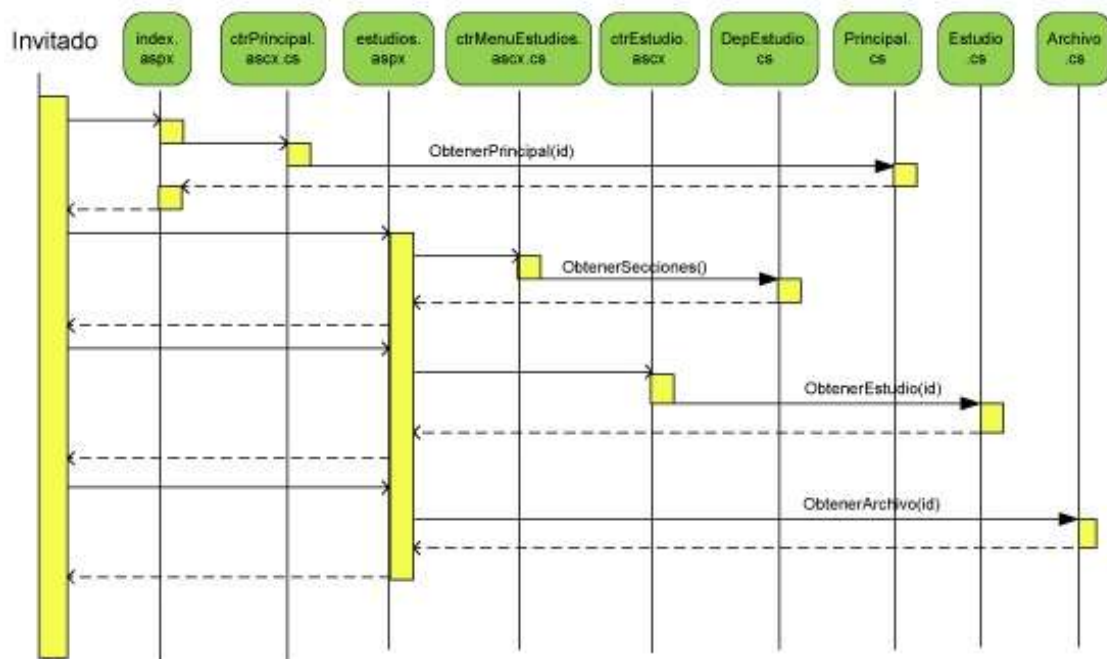


Figura 5.20. Diagrama de secuencia. Descargar archivo adjunto a un estudio.

En segundo lugar se descubrirá el funcionamiento del buscador del sitio web. En la parte superior derecha de la página principal, contenido en el control ctrBuscar.ascx se podrá encontrar un cuadro de texto que permitirá filtrar búsquedas por el texto introducido. Ese texto se incluirá como filtro en las consultas sql que se realizarán a las tablas Noticia, Estudio, Experto, Seminario y Conferencia, que devolverán todos los resultados en cuyo contenido figure dicho texto.

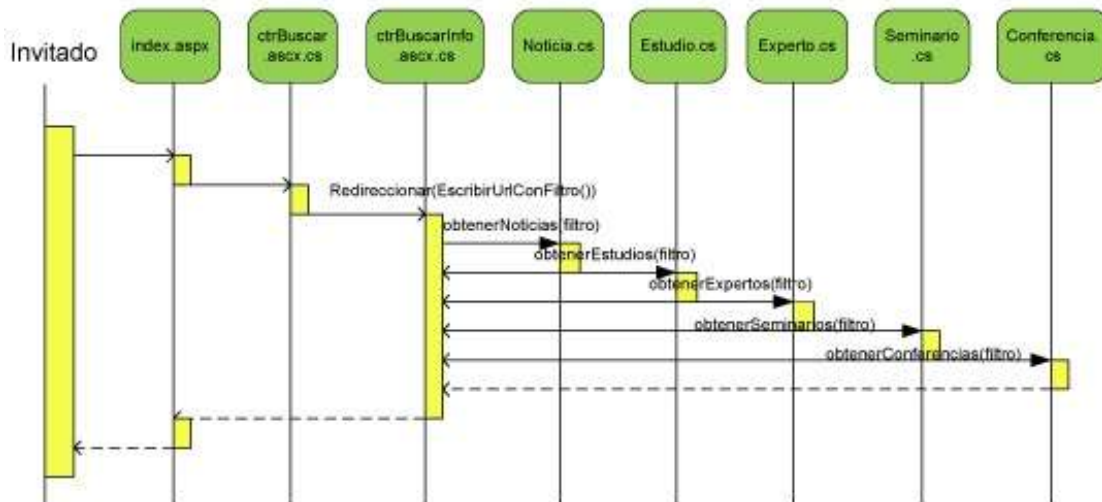


Figura 5.21 Diagrama de secuencia. Buscador del sitio web.

En tercer lugar se invita a descubrir lo sencillo que resulta acceder al mapa del sitio. El enlace para acceder se encontrará en el mismo control que el buscador, y permitirá acceder a un mapa con todos los títulos de todas las entradas de la web, clasificadas por secciones, mediante consultas a todas las tablas de la base de datos. Este contenido se maquetará en formato XML SiteMap de ASP.NET, permitiendo además hacer un seguimiento de las rutas a cada página del sitio y ayudando a una mejor indexación en la web.

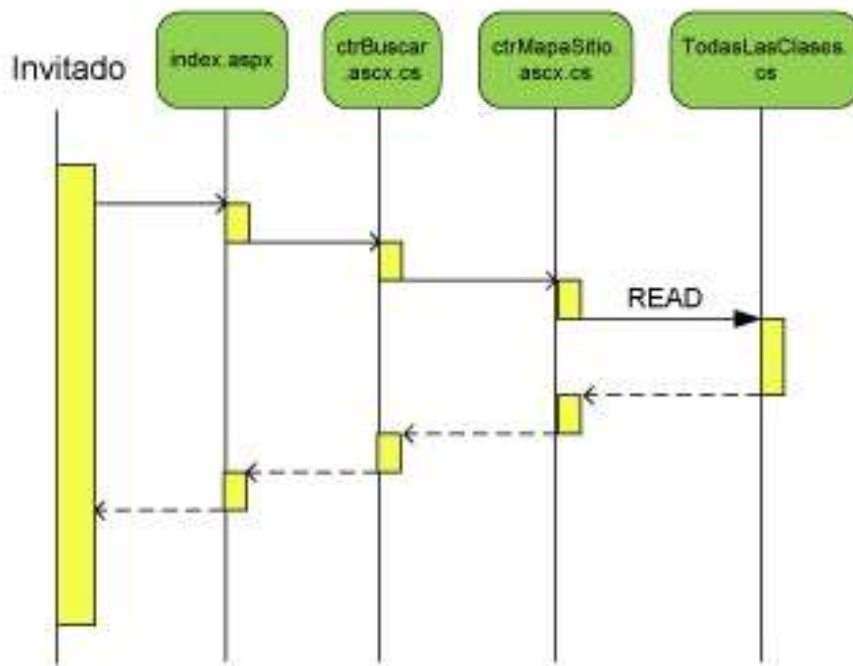


Figura 5.22. Diagrama de secuencia. Acceder a Mapa del Sitio.

Finalmente se podrá acceder con un simple clic al apartado Observador Admirado. Al contenido de esta sección no se accede a través de la base de datos del Observatorio, sino que se realizará una consulta al feed RSS de la categoría “Observador Admirado” del blog de Comunicación Audiovisual de la Universidad CEU Cardenal Herrera, mediante la clase RSS.cs, y se volcará el contenido seleccionado a través de los métodos sobre la página del sitio web.

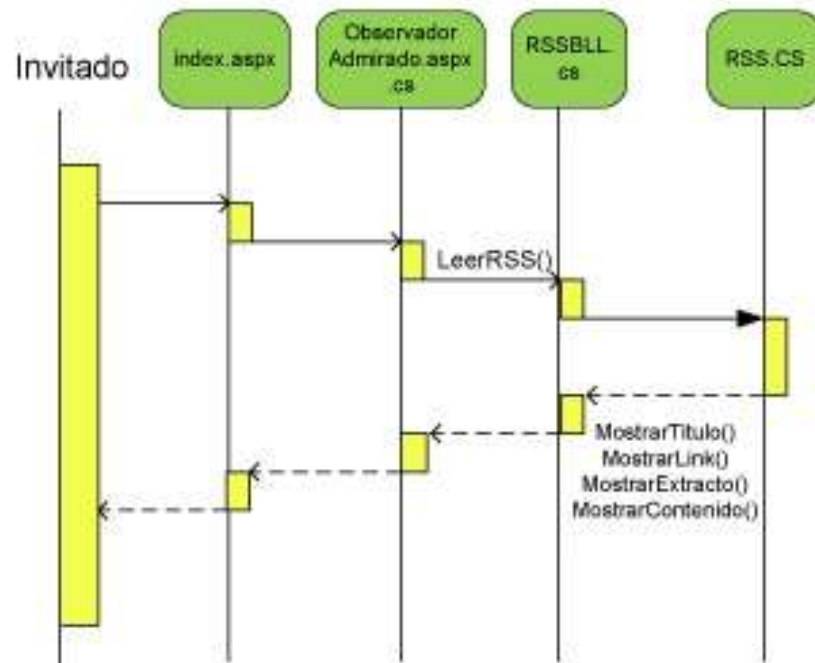


Figura 5.23 Diagrama de secuencia. Visualizar contenidos a partir de RSS de otro sitio web.

5.3 - Implementación del Sistema.

5.3.1 - Herramientas e interfaces usadas para la implementación.

En el Servicio de Comunicación Digital de la Universidad CEU Cardenal Herrera se dispone de equipos con acceso a un servidor web que funciona sobre IIS 6 y tienen soporte para ASP.NET. En este servidor, llamado “Webceu” se aloja la página de la Universidad y todas las aplicaciones web implementadas por este servicio.

Por otra parte, se dispone también de un servidor de bases de datos SQL Server en el que se aloja la base de datos a la que se realizan los accesos tanto desde la web como desde las aplicaciones. En esa base de datos se van a crear las tablas y los procedimientos almacenados necesarios para llevar a cabo todos los casos de uso, tanto del sitio web del Observatorio Beyond, como de su aplicación de gestión.

Existen varias herramientas para acceder y manejar una base de datos SQL Server de forma sencilla. Una herramienta gratuita y bastante intuitiva es Microsoft SQL Server Management Studio Express. La versión que se ha instalado y utilizado para la realización de este proyecto es exactamente SQL Server Management Studio Express 2005. Existen varias versiones, tanto anteriores como posteriores, ya que es bastante versátil, pero con esta versión ha sido suficiente para llevar a cabo la gestión de los datos. A continuación se procede a explicar su instalación y uso.

5.3.2. SQL Server Management Studio Express 2005.

5.3.2.1 - Instalación.

Esta herramienta se puede descargar desde la página de Microsoft y su instalación es realmente sencilla, ya que ofrece un asistente totalmente mecánico, como es costumbre con cualquier software de la compañía. Si la instalación se realiza a partir del archivo que se descarga desde la web, ésta se empaqueta como un ejecutable mediante una tecnología de instalación llamada SFXCab. Al hacer doble clic en el .exe se inicia automáticamente el proceso de instalación.



Figura 5.24. Asistente de instalación SSMSE 2005 (1).

Tras hacer clic en “siguiente” en esta ventana y en la posterior, que es la de aceptación de los términos del contrato de licencia, hay que elegir la ubicación del disco en la que se pretende realizar la instalación. Por defecto se alojará en el directorio “Program Files”, creando la carpeta “Microsoft SQL Server”.

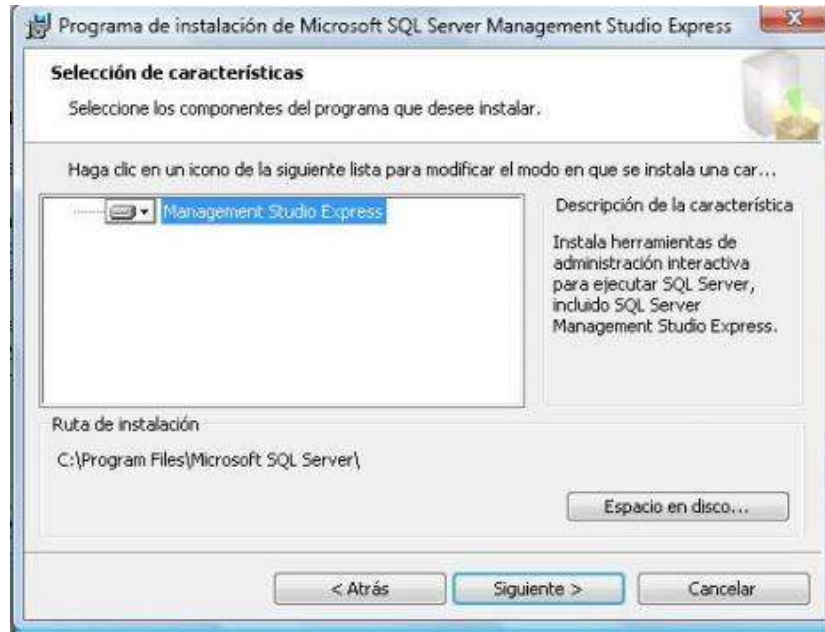


Figura 5.25. Asistente de instalación SSMSE 2005 (2).

En la siguiente ventana simplemente hay que presionar “instalar” y esperar a que se lleve a cabo la instalación de forma automática. Tras pulsar en finalizar, se puede iniciar el servicio desde la barra de inicio buscando el nombre del programa en el listado y pulsándolo.



Figura 5.26 Iniciar SSMSE 2005.

Entonces aparecerá una ventana en la que hay que elegir el nombre del servidor y los parámetros de autenticación para poder establecer la conexión con el servidor de bases de datos de la Universidad.



Figura 5.27. Autenticación en SSMSE 2005.

Una vez introducidos los parámetros hay que pulsar el botón conectar y se accederá a la pantalla de inicio de SSMSE 2005.

5.3.2.2 - Diseño de tablas SQL para la aplicación web

Una vez en la pantalla principal se puede observar en la parte izquierda el explorador de objetos, donde se ve una lista cuyo primer elemento es el nombre del servidor de bases de datos y el nombre de usuario. Al desplegarlo encontramos como primer elemento hijo las bases de datos contenidas en ese servidor.

Las tablas diseñadas para este proyecto se han alojado en la base de datos “web”, que se ve desplegada en el último lugar. Después hay que pulsar el elemento “tables” con el botón derecho y seleccionar “New table”. A la derecha aparecerá una estructura de datos con tres columnas a la que se pueden añadir tantas filas como campos se necesiten. En la imagen se puede ver cómo se ha diseñado una de las tablas, en concreto la de conferencias. El modo de diseño es muy sencillo e intuitivo, y en este caso ha bastado con indicar nombre, tipo de dato y si este puede ser nulo o no.

Evidentemente el sistema no permite que la clave primaria (en este caso el campo id) sea nulo, y se puede especificar en el recuadro de abajo, “column properties”, otras características, como por ejemplo indicar que el campo “seminario_id” es una clave ajena que referencia a la clave primaria de la tabla Seminarios.

También es posible crear la tabla a partir de un script. Simplemente hay que acceder al menú archivo y hacer clic sobre “New query”. En la nueva pestaña que se abre se puede escribir el script para crear la nueva tabla.

En el caso ejemplificado anteriormente el script es el siguiente:

```
USE [web]

GO

/***** Object:  Table [dbo].[WEB_OBSERVATORIO_Conferencias]
Script Date: 11/11/2011 13:38:44 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[WEB_OBSERVATORIO_Conferencias] (

    [id] [int] IDENTITY(1,1) NOT NULL,

    [seminario_id] [int] NOT NULL,

    [titulo] [nvarchar](100) NULL,

    [texto] [ntext] NULL,

    [imagen] [nvarchar](100) NULL,

    [archivo] [nvarchar](100) NULL,

    [fecha] [datetime] NULL,

    [visible] [bit] NULL,

    CONSTRAINT [PK_WEB_OBSERVATORIO_Conferencia] PRIMARY KEY
    CLUSTERED

(

    [id] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

ALTER TABLE [dbo].[WEB_OBSERVATORIO_Conferencias] WITH CHECK
ADD CONSTRAINT
[FK_WEB_OBSERVATORIO_Conferencias_WEB_OBSERVATORIO_Seminarios]
FOREIGN KEY([seminario_id])
```

```

REFERENCES [dbo].[WEB_OBSERVATORIO_Seminarios] ([id])

GO

ALTER TABLE [dbo].[WEB_OBSERVATORIO_Conferencias] CHECK
CONSTRAINT
[FK_WEB_OBSERVATORIO_Conferencias_WEB_OBSERVATORIO_Seminarios]

```

Se pueden observar algunas diferencias respecto a la sentencia que se ejecutaría en MySQL. Por ejemplo el tipo de dato “ntext”, que almacenan cadenas superiores a 4000 caracteres. También es diferente la forma de especificar los índices y su agrupación, en este caso bajo la palabra “CLUSTERED”, especificando la configuración de las propiedades entre paréntesis.

Los scripts de todas las tablas del proyecto se encuentran en el CD entregado junto a la memoria, en una carpeta con el nombre “Tablas Observatorio”.

A continuación se va a explicar cómo se han diseñado los procedimientos almacenados SQL para la inserción, borrado, actualización y consulta a las tablas diseñadas para la aplicación.

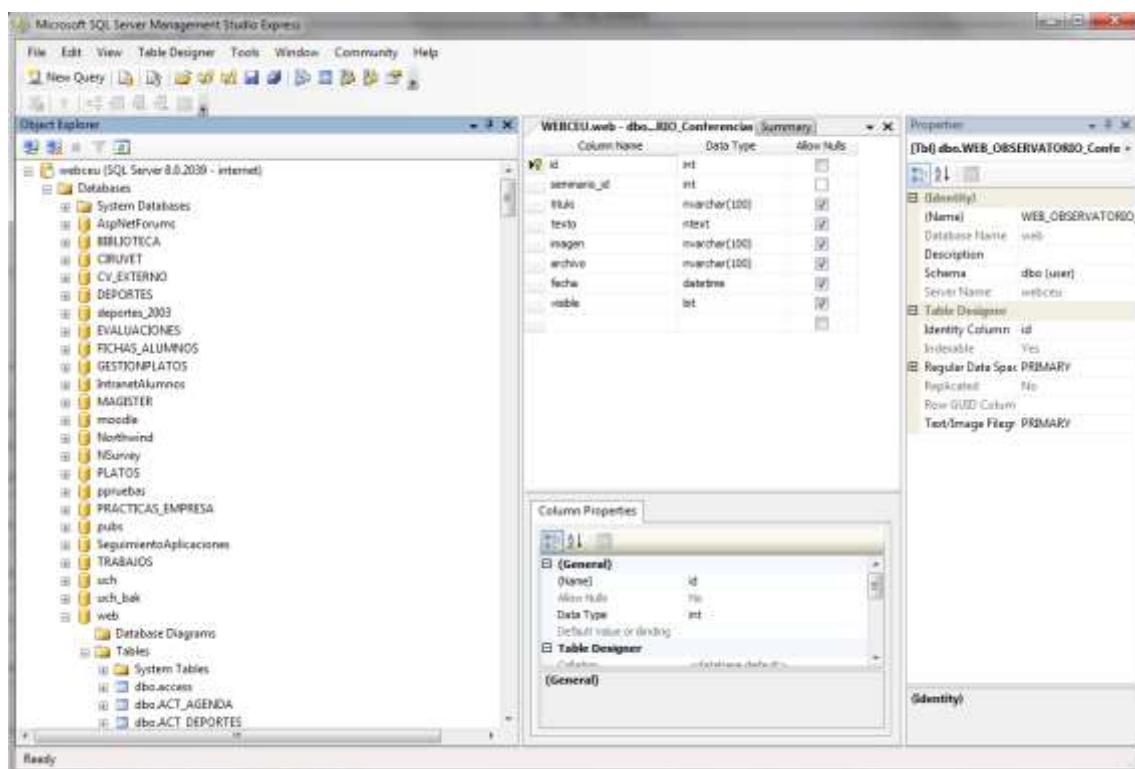


Figura 5.28 Diseño de tabla dbo.[Web_Observatorio_Conferencias] con SSMSE 2005.

5.3.2.3 - Diseño de procedimientos almacenados SSMSE 2005

Para implementar los procedimientos almacenados hay que volver a buscar en la columna “Object Explorer”. En el listado, al mismo nivel del elemento “Tables”, se encuentra

otro elemento llamado “Programmability”. Al desplegarlo aparece el elemento “Stored Procedures” en primer lugar.

Si se pulsa con el botón derecho del ratón sobre ese elemento, la primera opción que aparece es “New stored procedure”, que abrirá una nueva pestaña en el recuadro del centro en el que se podrá escribir la sentencia SQL. También se puede abrir una nueva consulta desde el menú archivo, haciendo clic en “New query”, pero en ese caso hay que especificar manualmente la base de datos en la que se quiere incluir el procedimiento almacenado.

En la imagen se muestra la sentencia SQL para actualizar la tabla “Conferencias”, diseñada anteriormente.

Evidentemente no es necesario crear un procedimiento almacenado para cada consulta que se quiere realizar a la base de datos, pero sí es recomendable tener definidos estos procedimientos para no tener que especificar las consultas desde ASP.NET cada vez que sea necesario. Una buena arquitectura de capas favorecerá la programación y la depuración de código posterior.

Por esa razón en este proyecto los procedimientos almacenados se han creado de esta manera y se pueden encontrar en el CD adjunto al informe, en la carpeta “Procedimientos Almacenados Observatorio”.

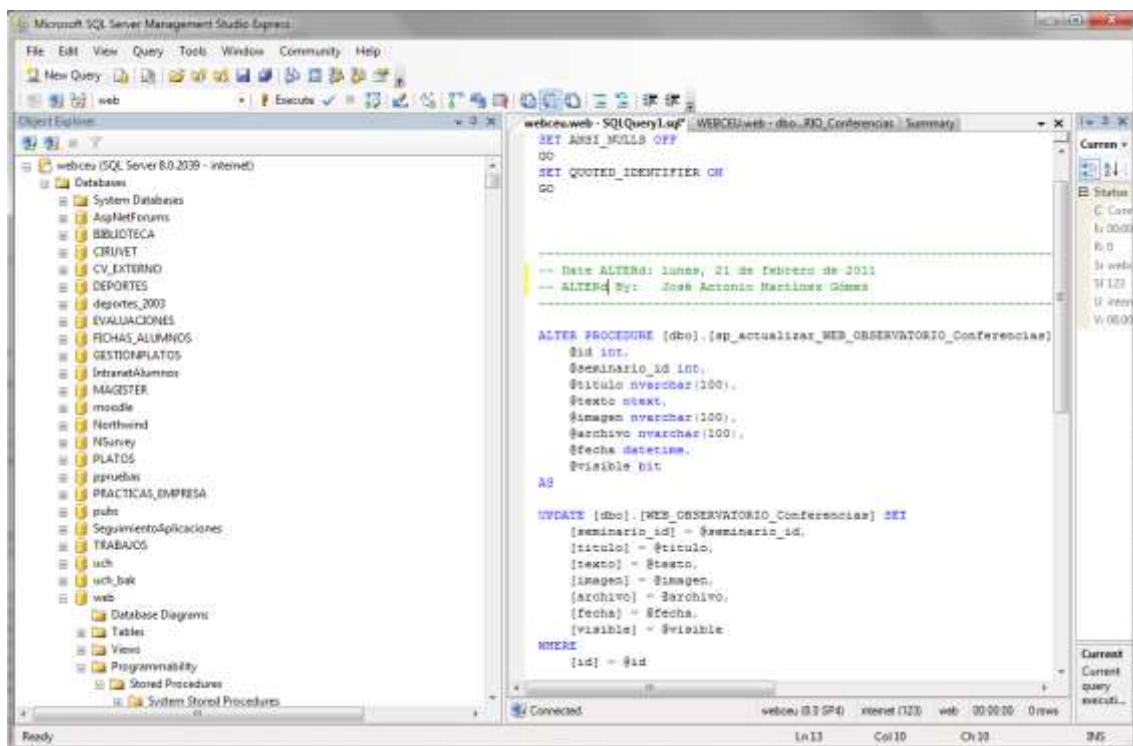


Figura 5.29 .Diseño de procedimiento `dbo.[sp_actualizar_Web_Observatorio_Conferencias]` con SSMSE 2005.

Aquí finaliza la implementación con SQL, para implementar el resto de la aplicación es necesario contar con otra herramienta. Para proyectos anteriores en la Universidad se ha utilizado el entorno de desarrollo Visual Studio en sus versiones 2003, 2005 y 2008. Al

comienzo de este proyecto ya estaba disponible Visual Studio 2010, y en este caso sí que es importante estar actualizado, ya que la última versión utiliza el .NET Framework 4.0, y los servidores de la universidad se disponen a actualizar el soporte a los nuevos frameworks a través del IIS.

Así pues, a continuación se procederá a explicar la instalación y uso del Visual Studio 2010 para este proyecto.

5.3.3 - Visual Studio 2010.

5.3.3.1 - Instalación.

Recientemente la Universidad ha adquirido una licencia de Visual Studio 2010 Ultimate, la más completa de las versiones de esta herramienta, dedicada a aplicaciones escalables tanto de escritorio como web. Este último caso se ajusta a las necesidades de la web de esta entidad.

Aunque la aplicación desarrollada en este proyecto está pensada para ser alojada en el servidor “Webceu” para acabar constituyendo una de las secciones del sitio web de la Universidad, la instalación de Visual Studio se va a realizar en el equipo local. En primer lugar porque no es recomendable instalar software de más en un servidor, y en segundo lugar porque trabajar en un host local permitirá compilar y depurar la aplicación antes de subirla.

Así pues, se va a proceder a la instalación de Visual Studio 2010 en un equipo con Windows 7. El proceso es realmente sencillo.

En primer lugar aparecen una serie de pantallas donde el usuario únicamente tiene que pulsar el botón “siguiente”, tras aceptar el contrato de licencia y decidir si se desea enviar información de la experiencia a Microsoft.

A continuación se puede elegir seleccionar la instalación de todos los componentes posibles o una instalación personalizada en la que se puede decidir para qué lenguajes de programación se requiere soporte y qué herramientas se pretende instalar.

Los lenguajes de programación disponibles son:

- Visual Basic.
- Visual C++.
- Visual C#.
- Visual F#.

Las herramientas que permite seleccionar en la instalación:

- Microsoft Office Developer Tools.
- Dotfuscator Software Services.
- Microsoft SQL Server 2008 Express.

- Microsoft SharePoint Developer Tools.

Aunque el lenguaje que se viene utilizando en todas las aplicaciones y que se va a utilizar en este proyecto es C#, se ha decidido instalarlo todo. En el caso de las herramientas, aunque ya está instalado el SQL Server 2008 no hay ningún problema, porque el instalador detectará este software y se saltará esta parte del proceso.

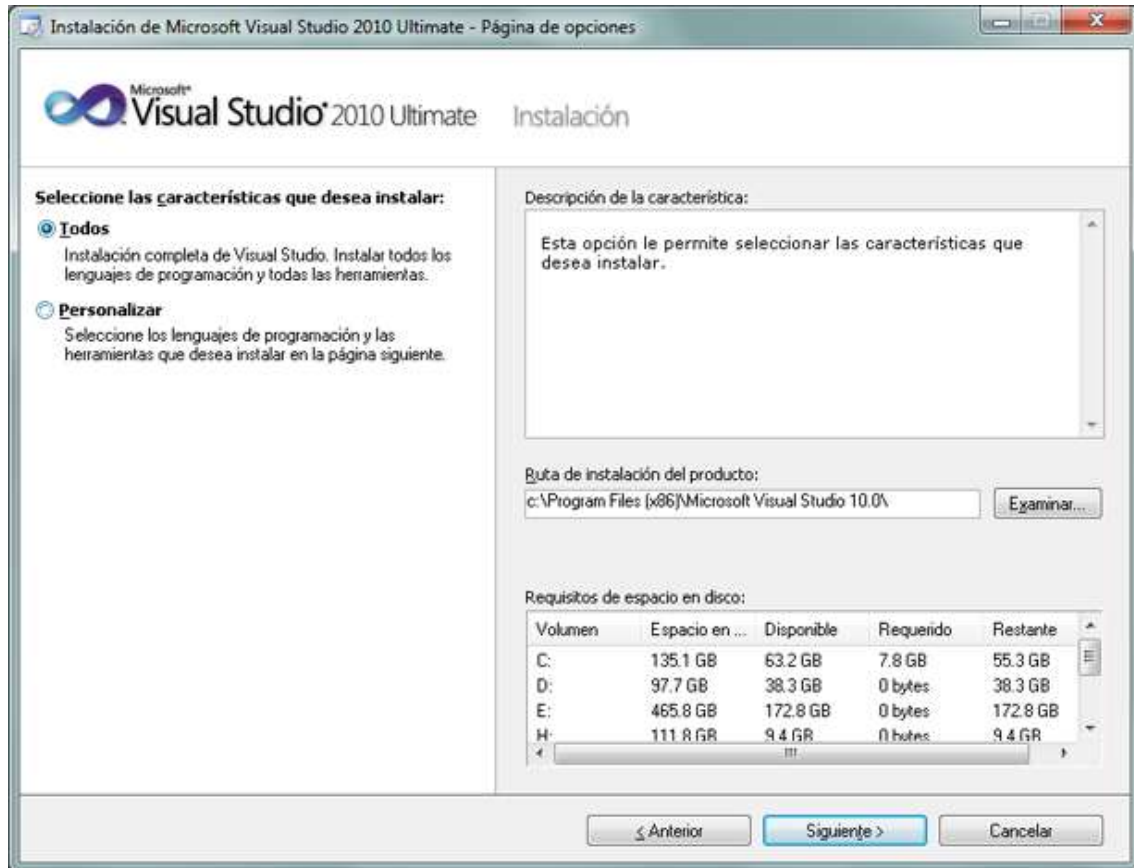


Figura 5.30 Instalación de Microsoft Visual Studio 2010 Ultimate.

Tras la instalación, el asistente para instalar Visual Studio mostrará una nueva ventana, con el texto "Correcto. Se instaló Visual Studio 2010 y finalizó el programa de instalación". Hay que pulsar entonces el botón "Finalizar", pudiendo darse el caso de que el programa pida reiniciar el equipo.

5.3.3.2 - Desarrollo de la aplicación y el sitio web con Visual Studio 2010.

Una vez instalado el entorno de desarrollo se procede a empezar la implementación de la aplicación web. En primer lugar se inicia Visual Studio desde la barra de inicio, en "programas".

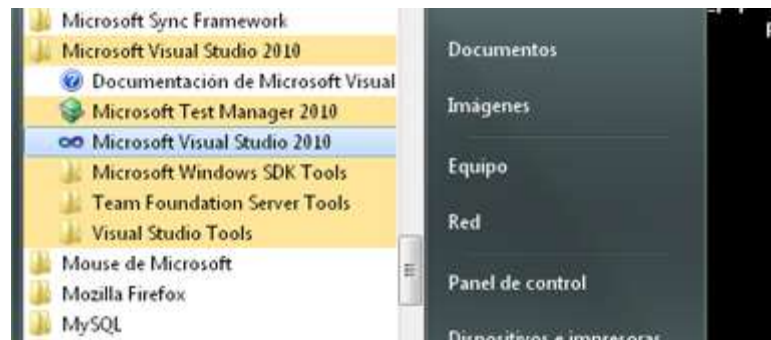


Figura 5.31 Iniciar Visual Studio 2010.

Al iniciar Visual Studio por primera vez, pedirá qué lenguaje de programación se va a utilizar como opción predeterminada para desarrollar las aplicaciones. La elección, como se dijo anteriormente, es C#. Entonces aparecerá la pantalla principal.

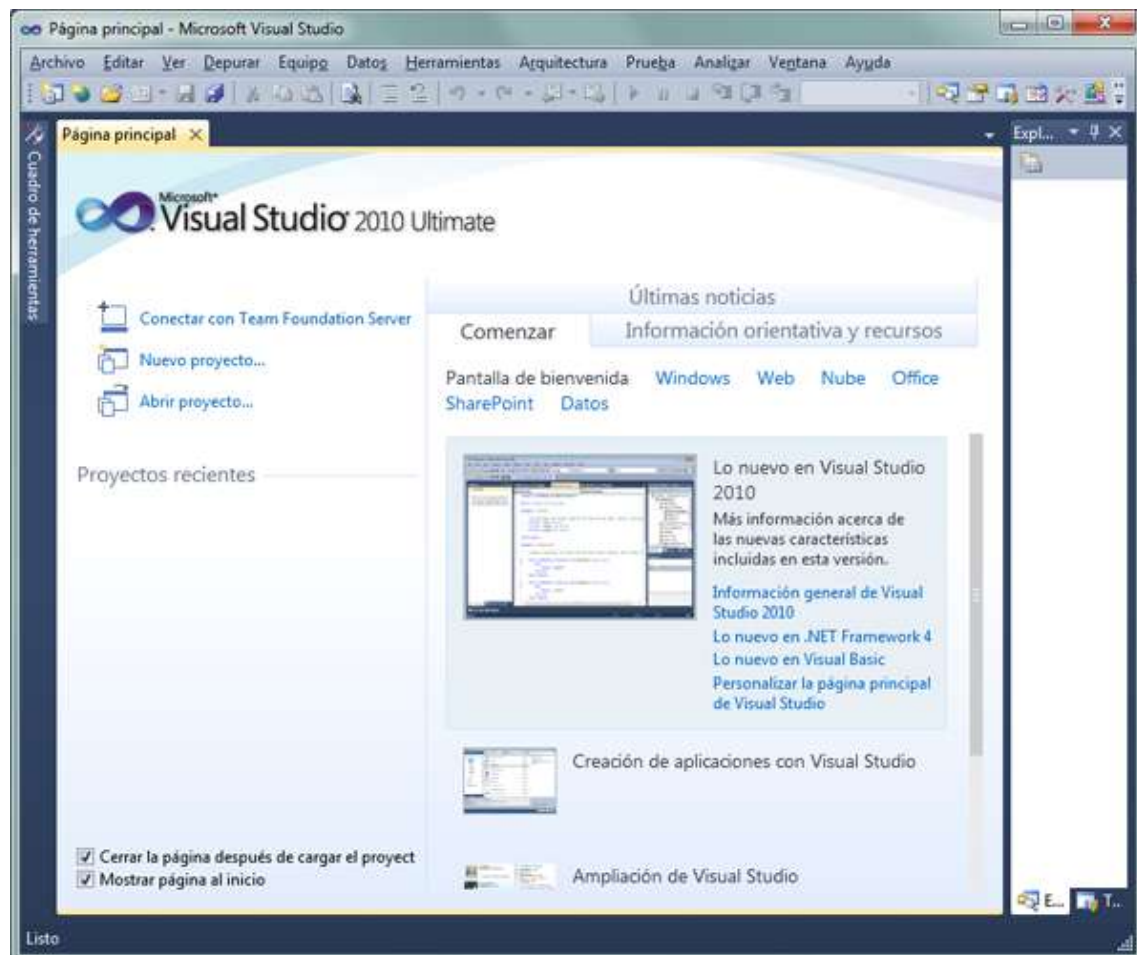


Figura 5.32 Ventana principal de Visual Studio 2010.

5.3.3.3 - Biblioteca de clases.

En primer lugar se procederá a la implementación de la capa de acceso a los datos, es decir, las clases que se comunicarán directamente con los procedimientos almacenados implementados con SQL Server.

Para ello hay que dirigirse al menú “Archivo”, situado en la parte superior izquierda de la ventana y hacer clic en la opción “Nuevo proyecto”, abriéndose a continuación una nueva ventana. En esa ventana aparecerán en primer plano los diferentes tipos de proyecto que se pueden realizar, adaptados al lenguaje C#, que se eligió como predeterminado, pero al haber realizado la instalación completa se permite seleccionar proyectos para el resto de lenguajes. En la parte superior también se puede elegir el Framework a utilizar en el desarrollo, apareciendo por defecto el más actual, el .NET Framework 4. Como el servidor de la universidad va a ser actualizado para dar soporte a esta versión, se va a mantener así.

El tipo de proyecto que se va a seleccionar en esta fase es “Biblioteca de Clases”. El nombre de esta biblioteca será “ClasesObservatorio” y se guardará en la carpeta “Projects” del directorio por defecto para archivos de Visual Studio 2010.

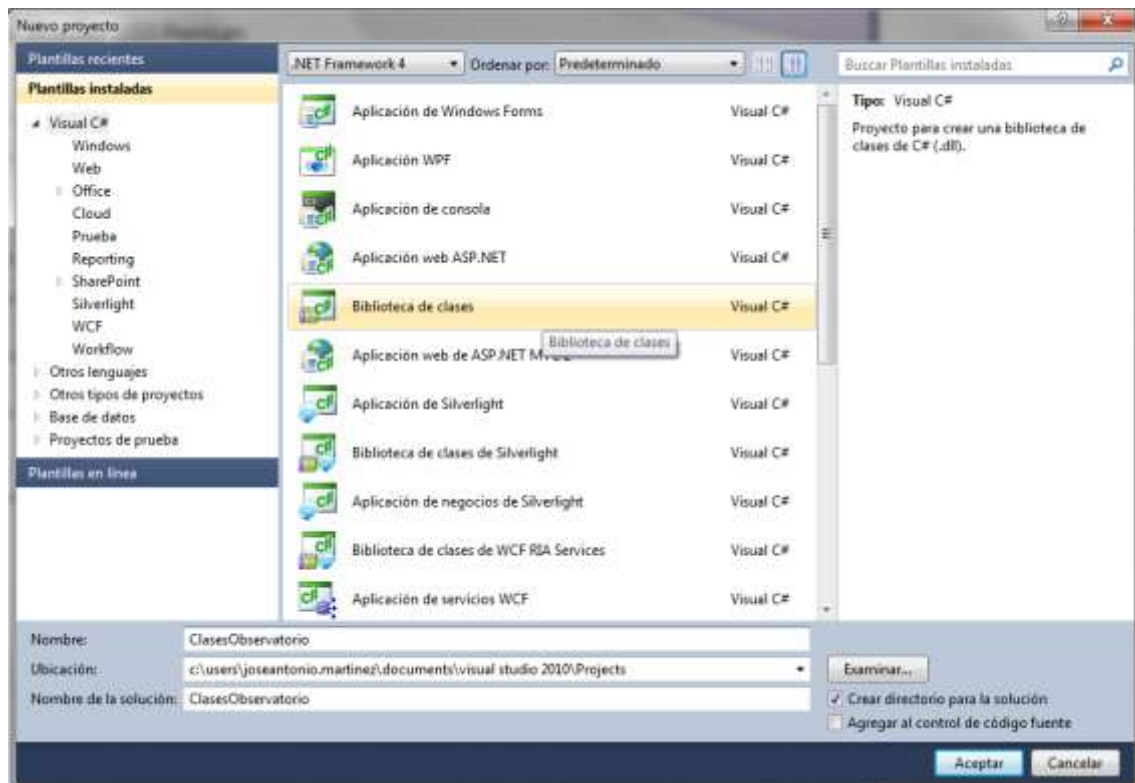


Figura 5.33. Nuevo Proyecto. Biblioteca de clases.

Tras hacer la selección se volverá a la pantalla principal y aparecerá, en la ventana “Explorador de soluciones”, situado en la parte derecha, el nombre del proyecto y una lista desplegable que se irá extendiendo en la medida en la que se incluyan las clases.

Para añadir una clase hay que hacer clic con el botón derecho sobre el nombre del proyecto en el “Explorador de soluciones”, y seleccionar la opción “Agregar nuevo elemento”. Todos los elementos que se van a crear en esta biblioteca van a ser clases. Hay que seleccionar por tanto esa opción, escribiendo en el cuadro inferior junto a la etiqueta “Nombre:” el nombre de la clase a implementar. Seguidamente hay que pulsar “Aceptar”.

Automáticamente aparecerá una pestaña en el panel principal con el nombre de la clase elegida y un espacio de texto habilitado para su implementación.

En primer lugar se escriben las directivas “using” seguidas de los nombres de las bibliotecas, pertenecientes a .NET o externas a la plataforma, de las que se va a hacer uso. Son comunes y se utilizan en este proyecto las bibliotecas del sistema “System.Text” para representar codificaciones de caracteres habituales, “System.Data” para poder hacer uso de las estructuras de datos, “System.web” para poder comunicar el explorador con el servidor web, y “System.Data.SqlClient” para poder comunicar la aplicación con el servidor SQL.

A continuación se especifica el espacio de nombres de esta clase, para poder ser llamada desde otras capas de la aplicación, y se declara el nombre de la clase.

Posteriormente se guarda la cadena de conexión con la base de datos en un objeto de la clase “SqlConnection”, definida en la librería “SqlClient”. El nombre y los parámetros de la cadena de conexión pueden especificarse directamente en cada clase o en el fichero de configuración del proyecto, como se verá después.

Entonces hay que definir los atributos de la clase, que se corresponderán con los campos de la tabla de la base de datos a la que se desee acceder. En el ejemplo posterior se puede ver que el tipo de dato “int” de C# es el equivalente al tipo “int” de SQL, y que el tipo “string” se corresponde con los tipos “varchar” y “ntext”.

Después se define el constructor de la clase, que recibe como parámetro el identificador de la clase para hacer una consulta a la base de datos y recibir los datos de todos los campos de la entrada correspondiente en la tabla requerida. Para ello se declara un objeto de la clase “SqlCommand” de la biblioteca “SqlClient”, cuyo método “CommandText” permite hacer una llamada a un procedimiento almacenado creado con Sql Server.

Esos datos se guardan en un objeto de la clase “SqlDataReader”, que nos permite acceder a cada uno de los campos declarados en el procedimiento SQL, y así poder almacenarlos en los atributos de la clase.

Además del constructor, se definen de manera similar los métodos que llaman a los diferentes procedimientos almacenados de consulta, inserción, actualización y borrado de elementos de la base de datos. Cabe destacar que los métodos de consulta devuelven un objeto de la clase “DataView”, de la librería “Data”, que permite almacenar listas de elementos para visualizarlas en la web.

Al compilar esta biblioteca de clases se generará una biblioteca de enlace dinámico (archivo .dll) que se alojará en el directorio “bin” tanto de la aplicación de gestión como del sitio web del Observatorio Beyond.

Todas las clases de esta biblioteca se han definido de manera similar y su código se encuentra en el CD adjunto al informe, en la carpeta “ClasesObservatorio”.

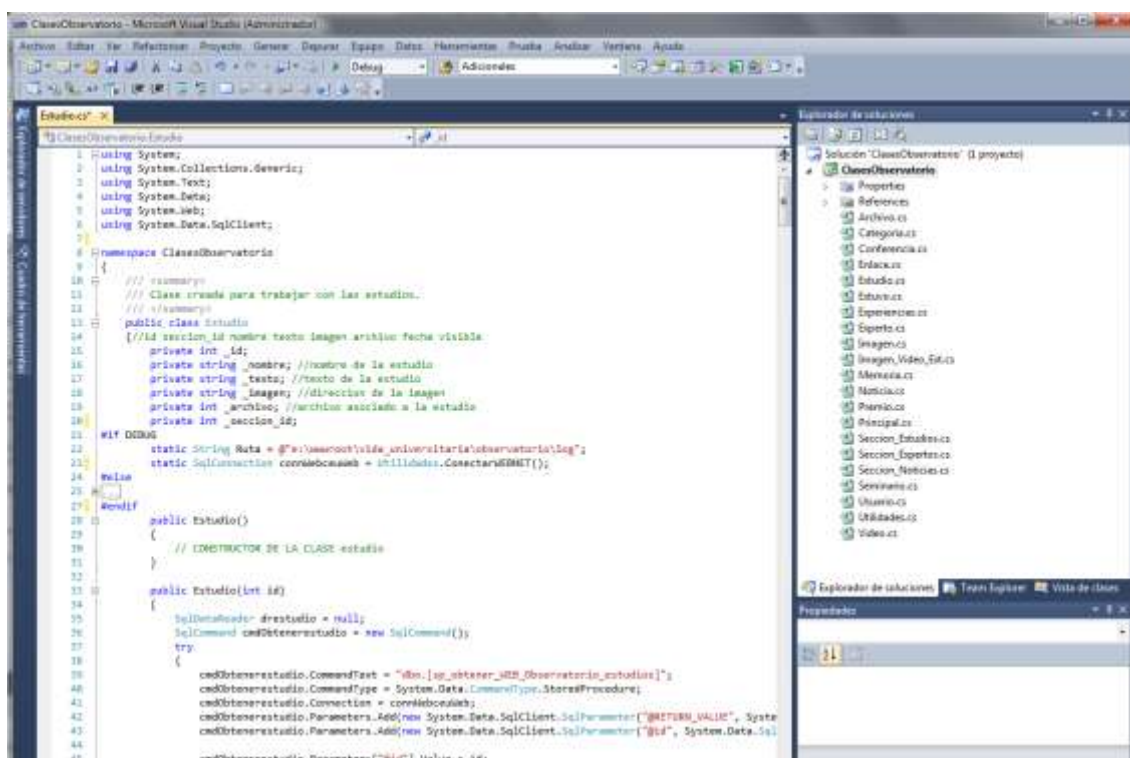


Figura 5.34 Implementación de una Clase del proyecto.

5.3.3.4 - Otras clases implementadas.

Además de la biblioteca “ClasesObservatorio”, en la que cada clase se corresponde con una entidad de la base de datos, se han implementado durante el desarrollo de esta aplicación tres clases más que contienen métodos genéricos de gran utilidad.

En primer lugar está la clase “**Utilidades.cs**”, que contiene los siguientes métodos.

“GuardarError(string Ruta, Exception ex, string Aplicacion)”, que recoge como parámetros el nombre de la aplicación, el nombre de la excepción que se haya generado en la aplicación y la ruta en la que se desee guardar un fichero de log que registre los errores (generado mediante el método “StreamWriter”).

“MandarMail(string destino, string origen, string asunto, string mensaje)”, que envía un mensaje de correo electrónico al destino especificado y con el resto parámetros de entrada obtenidos mediante el método “MailMessage” de la librería “System.Web.Mail”.

“System.Data.SqlClient.SqlConnection Conectar()”, es una manera alternativa a la que se ha descrito en el fichero de comunicación para iniciar una conexión mediante la definición la cadena mediante el método “ConnectionString()”. Las clases de la biblioteca se conectarán mediante este método, para no tener que depender de un fichero de configuración únicamente para esto.

Por otra parte, se han definido dos clases para generar y leer contenido RSS.

En primer lugar, se ha implementado la clase **"RSS.cs"**, que utiliza la clase **"XmlSerializer"**, ubicada dentro del espacio de nombres **"System.Xml.Serialization"**, para leer y mostrar el fichero RSS.

"XmlSerializer" es un método muy potente. A partir del tipo de una clase puede mostrar su equivalente XML, y a su vez rellenar una instancia de la clase a partir de un XML.

En este caso concreto, a partir de la clase RSS, **XmlSerializer** lo transforma en un fichero RSS mediante su método **"Serialize"**; y además, a partir de cualquier fichero RSS, rellena una instancia a un objeto RSS mediante su método **"Deserialize"**.

Esto es posible porque lee los metadatos para saber el nombre de las propiedades de una clase, su tipo, etc. Por otra parte dispone de unos atributos XML (**XmlElement**, **XmlRoot** y **XmlIgnore**), que sirven para indicar a **XmlSerializer**, qué propiedades no se muestran por XML o cuál es la raíz del XML, (en este caso **"rss"**).

En segundo lugar se ha implementado la clase **"RSSBLL.cs"**, que contiene principalmente dos métodos:

ReadRSS(), que deserializa la información y lee el contenido un fichero con formato RSS para almacenarla en el objeto de tipo RSS definido antes. Para ello se hace una petición a una URL que contenga el código XML. Los métodos **"WebRequest"** y **"WebResponse"** de la librería **"System.Net"** realizan la solicitud y la respuesta respectivamente.

WriteRSS(), que se ha implementado recogiendo el contenido del objeto RSS y serializando la información con el método **"XmlTextWriter"** para posteriormente encapsularla con HTTP y mostrarla a través del método **"HttpResponse"**.

Finalmente se ha implementado también la clase **"MapaSito.cs"**, que se encarga de generar el mapa de la web del Observatorio.

En este caso se generan ficheros de texto mediante los métodos **"StringBuilder"** y **"StringWriter"**. Mediante estos métodos se va escribiendo línea a línea hasta terminar la estructura del fichero XML, y se le añade al nombre del fichero la extensión **".sitemap"**. Estos ficheros se rellenan dinámicamente con cada una de las entradas de cada sección de la web mediante consultas a la base de datos, y se actualiza con cada nueva entrada que se inserta, se borra o se actualiza.

Estas tres clases adicionales también se han incluido en el CD adjunto al proyecto y se encuentran en la carpeta **AppCode**.

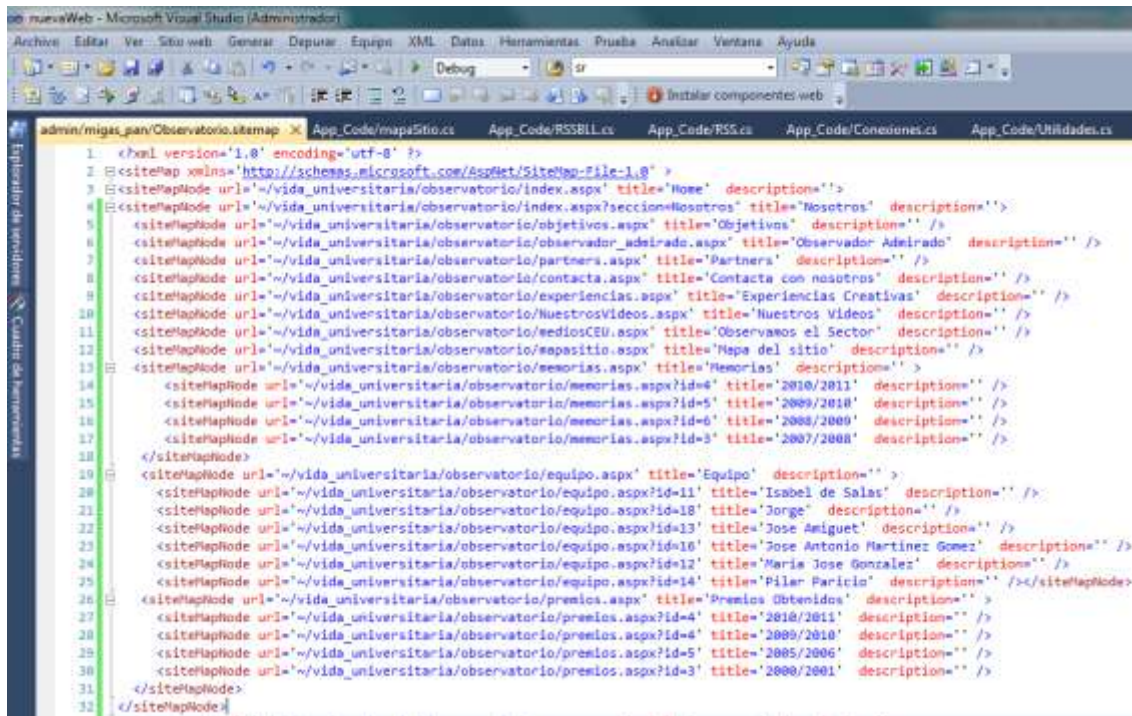


Figura 5.35 Fichero “Observatorio.sitemap” generado por la clase MapaSitio.cs

5.3.3.5 - Implementación de las capas de presentación y de negocio.

Una vez terminada la biblioteca de clases, ha llegado el momento de desarrollar la parte de la aplicación web que comprende las capas de negocio y de presentación. Para ello hay que abrir una nueva ventana de Visual Studio, dirigirse al menú Archivo, y seleccionar la opción “Nuevo sitio web”. Entonces aparecerá una ventana en la que se podrá elegir varias plantillas con C# y con otros lenguajes, así como el framework utilizado, tal y como sucedía en el caso anterior.

Esta vez los elementos a seleccionar son “Sitio Web ASP.NET”, “Visual C#” y “.NET Framework”. A continuación hay que elegir la ruta en la que se va a guardar y escribir el nombre de la aplicación, en este caso “AplicaciónObservatorio”. Finalmente se pulsa en aceptar, para volver de nuevo a la ventana principal.

(Este proceso es exactamente igual tanto para la creación de la Aplicación de Gestión como para la creación de la Página Web del Observatorio. En este último caso, el nombre elegido es “Observatorio”)

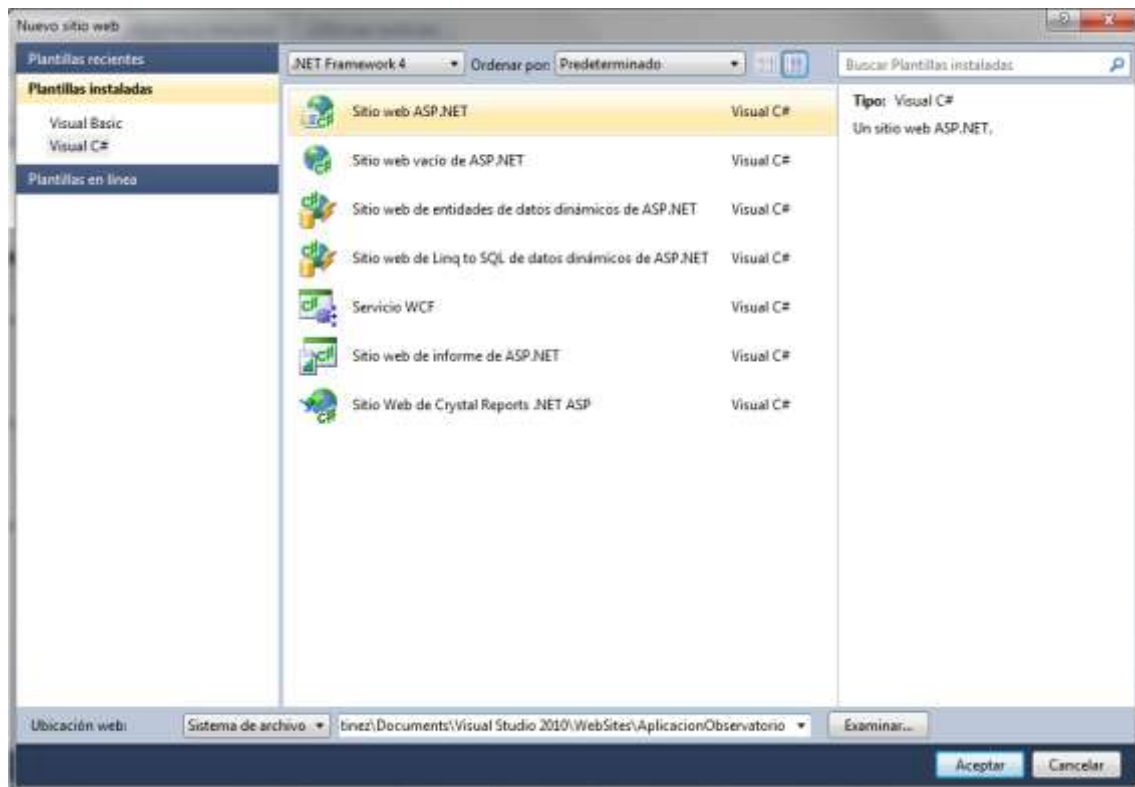


Figura 5.36 Creación de un nuevo sitio web con Visual Studio.

Tanto en la aplicación web como en el sitio web del Observatorio se pueden destacar una serie de archivos de gran relevancia.

“**Web.config**” es el archivo principal de opciones de configuración para una aplicación web en ASP.NET. El archivo es un documento XML que define información de configuración concerniente a la aplicación web. El archivo “web.config” contiene información que controla la carga de módulos (mediante la etiqueta “<appSettings>”), configuraciones de seguridad, configuraciones del estado de la sesión (etiqueta “<sessionState>”), opciones de compilación y el lenguaje de la aplicación. Los archivos “web.config” pueden contener también objetos específicos tales como cadenas de conexión a la base de datos (mediante la etiqueta “<connectionStrings>”), aunque ya se ha visto que se puede definir también en otras partes del proyecto.

Otros parámetros que permite especificar son, la versión del .NET framework que utiliza, el sistema de codificación, el modo de autenticación, el tiempo de cacheo, el modo depuración de errores - si se requiere -, etc.

Además de la información que define el fichero que se encuentra en la aplicación de gestión, el archivo “web.config” de la web del Observatorio, desde el directorio raíz del proyecto web, define la ruta (mediante la etiqueta “<siteMap>”) en la que se ubica el archivo que contiene el mapa del sitio web.

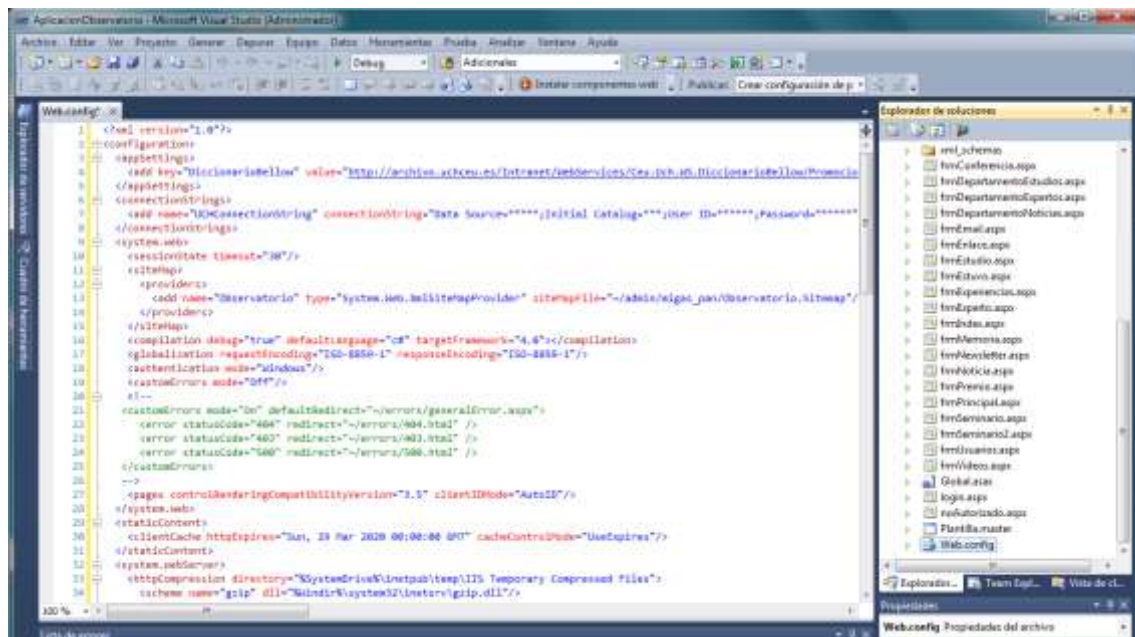


Figura 5.37. Fichero “web.config” compartido por la aplicación y por el sitio web.

El siguiente elemento a destacar es “Plantilla.Master”, una página maestra de ASP.NET que permite crear un diseño coherente para el resto de páginas de la aplicación. Puede definir el aspecto, el diseño y el comportamiento estándar que se desea que tengan las páginas de la aplicación en una sola página maestra. A continuación se crearán páginas de contenido individuales que añadirán el contenido que desea mostrar. De esta forma, cuando un usuario solicite una página de contenido, ésta se combinará con la página maestra para generar una salida que sume el diseño de la página maestra al de la página de contenido.

En este caso la plantilla, que comienza con la directiva “@Master”, contiene en primer lugar la declaración de los controles de usuario (ficheros con extensión .ascx) para que puedan ser ubicados en cualquier página de la web. En esta aplicación se han creado los controles `ctrArchivo` y `ctrImagen`. También contiene las cabeceras HTML, completas y la declaración de los CSS y los scripts que se usan en toda la aplicación.

También contiene parte del cuerpo de todas las páginas de la aplicación, entre las etiquetas HTML `<body>`, en concreto el listado de títulos y su enlace a todos los apartados de la aplicación.

Para especificar el panel en el cual se encuadran todas las páginas dentro de la plantilla se usa una etiqueta de ASP.NET que se llama “`<asp:ContentPlaceHolder>`” y que tiene un atributo “`id`” que lo identifica. Los ficheros de todas las páginas que compartan esta plantilla comenzarán con la etiqueta “`<asp:content>`” y tendrán un atributo “`ContentPlaceHolderID`” cuyo valor será el “`id`” de la plantilla maestra.

Finalmente, en esta plantilla se incluye también la información de pie de página.

A diferencia de lo que ocurre en la aplicación de gestión, en la página web del Observatorio no se ha creado una página maestra. En lugar de eso se ha optado por crear una gran cantidad de

controles de usuario independientes, ya que también son plantillas que se pueden ubicar en tantos sitios de la aplicación como se desee.

De esa forma también se consigue que todo quede centralizado, y así cualquier cambio en el fichero “.ascx” queda reflejado en cada página en la que se ha ubicado.

Ambos métodos son interesantes para evitar repetir código innecesario, tanto en la aplicación de gestión como en el sitio web.

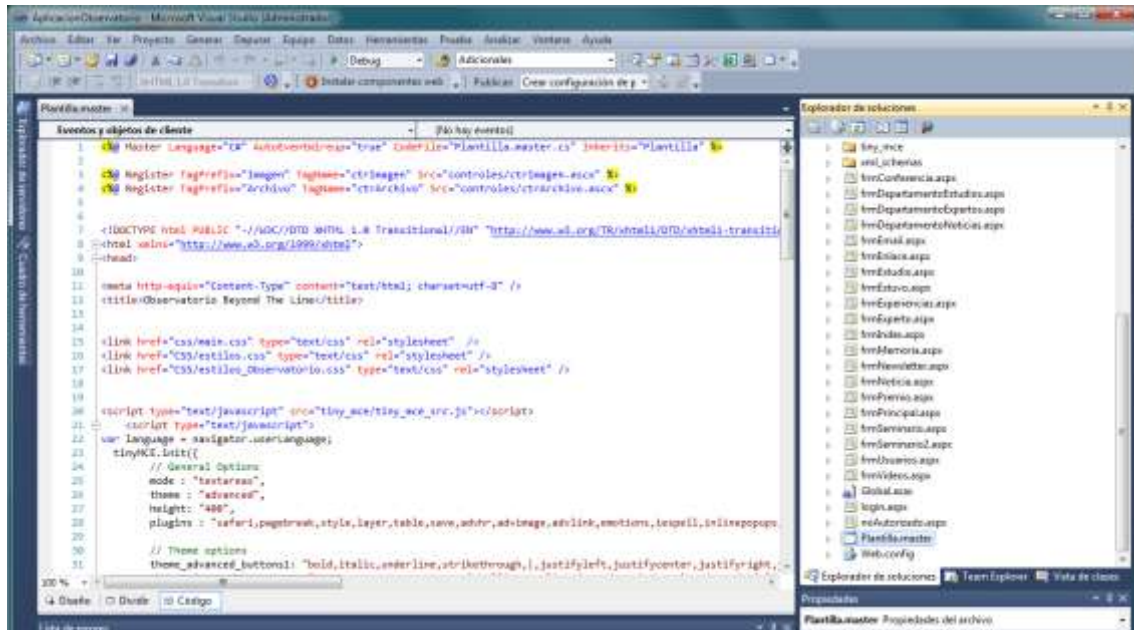


Figura 5.38. Fichero “Plantilla.master” de la Aplicación de Gestión.

A continuación se van a definir los controles de usuario mencionados anteriormente.

La estructura de los controles de usuario es muy similar al de las páginas “.aspx”, de las que se mostrará un ejemplo después.

Se dividen en dos partes: Una parte con extensión “.ascx” que se corresponde con la capa de presentación, incluyendo el código HTML y ASP que implementa la visualización por parte del usuario. Este fichero puede incorporar también la declaración de CSS específicos que no se han declarado en la plantilla porque no se quiere aplicar al resto de elementos.

La primera línea contiene la directiva “@Control” que indica que es un control de usuario y que enlaza con el archivo que implementa la capa de negocio mediante los atributos “codeFile”, que indica el nombre del archivo, e “inherits”, que indica el nombre de la clase que implementa.

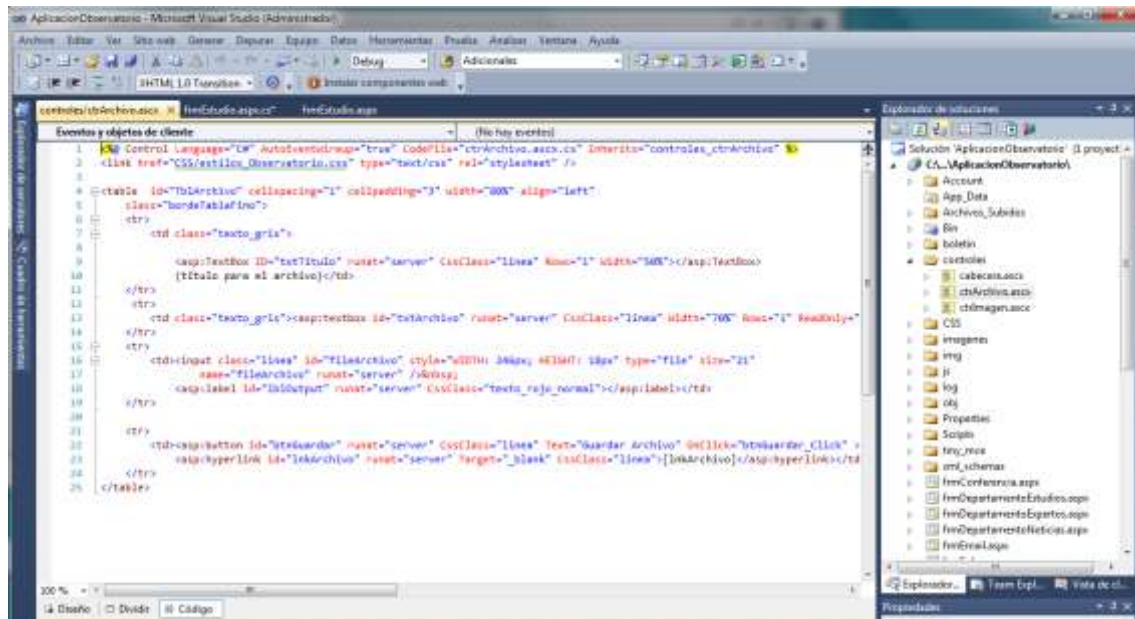


Figura 5.39 Control de usuario “.ascx”. Capa de presentación.

La lógica de negocio correspondiente a cada control de usuario se encuentra en unos archivos con el mismo nombre y con una extensión adicional “.cs”. Están programados con el lenguaje de programación C#, y contienen en primer lugar las directivas “using” que declaran las librerías de las que se van a hacer uso.

Cabe destacar aquí la declaración de la librería de clases “ClasesObservatorio”, definida anteriormente, que permite la declaración de instancias a los objetos definidos en estas clases.

En el caso concreto de los controles definidos en esta aplicación, estos ficheros implementan los métodos para almacenar en la ruta especificada del servidor, el archivo o imagen que el usuario ha decidido subir a través del formulario proporcionado en la parte visible del control.

La subida de ficheros e imágenes es posible gracias a un método llamado “FileStream(Ruta + NombreFichero, System.IO.FileMode.Create)” de la librería “System.IO”, que genera el archivo nuevo recibiendo como parámetros la ruta, el nombre del fichero, y la orden de creación de un nuevo fichero, pero hay que tener cuidado con el nombre de los ficheros, ya que tiene problemas con acentos y signos extraños. Con la función “Replace (string nuevo, string antiguo)” se pueden reemplazar estos caracteres.

Además, antes de subir una imagen, ésta se puede redimensionar a tres tamaños distintos: pequeña, mediana y grande, generando la nueva imagen con el método “DrawImage(original, coordenada x, coordenada y, nuevo ancho, nuevo alto)” de la librería “System.Drawing.Graphics”.

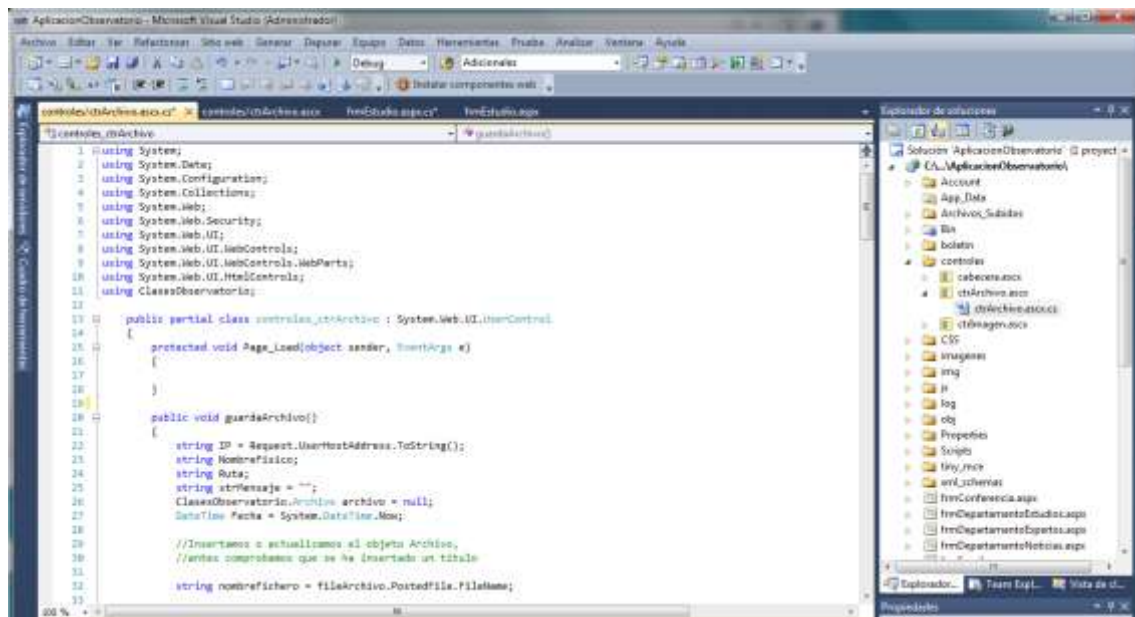


Figura 5.40. Control de usuario “.ascx”. Capa de negocio.

Y finalmente se definen las diferentes páginas de la aplicación web, que también se dividen en dos partes. Los ficheros con extensión “.aspx” se corresponden con la capa de presentación de cada uno de los apartados de la web.

En estos ficheros también se pueden declarar controles de usuario, plantillas CSS o Scripts específicos para esta página y que por tanto no hayan sido definidos en la plantilla maestra. El documento comienza con la directiva “@Page” que enlaza con la capa de negocio, indicando el nombre del fichero y de la clase correspondiente mediante los atributos “CodeFile” e “Inherits”, y abre el contenido del fichero con la etiqueta “<asp:Content>”, en la que se indica el ID de la plantilla maestra a través del atributo “ContentPlaceHolderID”.

Dentro de esta etiqueta se escribe todo el contenido estático mediante etiquetas HTML y ASP y el contenido dinámico que se ejecuta del lado del cliente, ya sea mediante Javascript o mediante las etiquetas de validación en el cliente que proporciona ASP.NET, por ejemplo la etiqueta mostrada en el ejemplo, “<asp:RequiredFieldValidator>”, que mostrará un mensaje de error si se deja vacío el campo de texto “<textBox>” asociado.

Por otra parte, estos ficheros también contienen la declaración de los métodos asociados a eventos, que se implementarán en la capa de negocio. Estos eventos se pueden activar al pulsar un botón <asp:Button>, mediante el atributo “onclick”, al seleccionar un elemento de un desplegable <asp:DropDownList>, mediante el atributo “onSelectedIndexChanged”, etc.

Pese a su paralelismo con los atributos que declaran métodos asociados a eventos en HTML, hay que diferenciarlos de éstos, ya que se implementan exclusivamente en la capa de negocio, con C#. Para poder hacer uso, por ejemplo, de un evento de pulsar botón programado con JavaScript en una etiqueta de ASP.NET, hay que hacer uso del atributo exclusivo de ASP.NET “OnClick”.

Otras herramientas que se han utilizado en las páginas de la aplicación de gestión del Observatorio son las que proporciona la librería de Ajax para ASP.NET. Con Ajax se han implementado los controles para la gestión de contenidos WYSIWYG (“What You See Is What You Get”), que permite integrar texto, vínculos e imágenes eligiendo la maquetación y los estilos (con las limitaciones que haya impuesto el desarrollador, para no desentonar en exceso con los estilos del sitio web estático). También se han implementado con Ajax los calendarios y los desplegables dinámicos, para que éstos únicamente realicen consultas a la base de datos cuando sean activados.

Cabe destacar también que en muchas de las páginas, tanto de la aplicación de gestión como del sitio web, se va a hacer uso de estructuras de datos llamadas “GridView” para representar el histórico de entradas de cada sección mediante una tabla donde cada columna representa un campo y cada fila representa un registro. El control “GridView” permite seleccionar, ordenar y editar estos elementos.

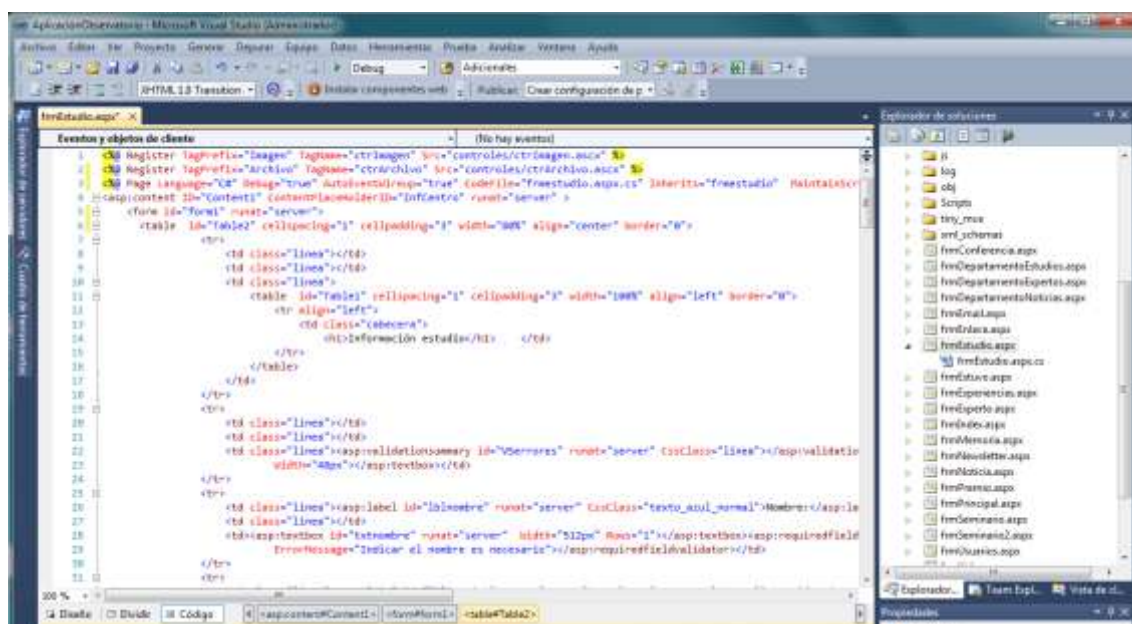


Figura 5.41 Página de ASP.NET “.aspx”. Capa de presentación.

Finalmente los ficheros con extensión “.aspx.cs” contienen la lógica de negocio de cada una de las páginas con extensión “.aspx” con su mismo nombre. De igual manera que ocurre con los ficheros de código “.cs” de los controles de usuario, aquí se declaran también las librerías que se van a utilizar, a destacar de nuevo la librería de clases “ClasesObservatorio”, definida para este proyecto.

A continuación se puede ver en la imagen el método “Page_Load”, que responde a un evento que se inicia cada vez que se carga la página. En este método se define qué valores deben mostrar los campos del formulario cuando carga la página. Por ejemplo, si se usa la condición “if (!Page.IsPostBack)” se está indicando que sólo se ejecute lo que hay dentro de la sentencia

cuando la página se carga por primera vez, y en este caso habría que indicarle que los campos deben ser vaciados de contenido.

Si no se accede por vez primera y por el contrario, se está editando algún registro, desde el método "Page-Load" se realizan también las consultas a la capa de acceso a datos a mediante el uso de los objetos proporcionados por la librería de clases "ClasesObservatorio", para poder rellenar los campos del formulario.

El método Page-Load responde al evento de carga de página, pero existen otros métodos que responden a eventos provocados por la interacción con el usuario, ya sea al pulsar un botón "<asp:Button onclick=...", al seleccionar una opción en un desplegable "<asp:DropDownList onselectedIndexChanged=...", o al editar uno de los registros mostrados en el "GridView", como el método "OnItemDataBound", mediante el cual se puede hacer aparecer popups con un mensaje de confirmación si se ha pulsado el evento Borrar.

Algunas otras funciones delegadas en estos archivos son el manejo de sesiones en la aplicación, almacenadas en objetos de tipo "Session", que se inician en el momento de la autenticación y se comprueban también en el "Page-Load" de cada página. Si la sesión se cierra o expira se produce la redirección (a través del método "Response.Redirect") a la página "NoAutorizado.aspx".

En el sitio web también se han realizado desde aquí labores de indexación y posicionamiento, generando "titles" dinámicos en las cabeceras de las páginas con los títulos de las entradas así como "keywords" y otros metadatos en función de los registros que se cargan. El acceso a los elementos ASP, para poder leer su contenido o editarlo desde la capa de negocio, se realiza con el método "FindControl". Este método lo implementan las clases "Page", "Control" y/o "Master", en función del tipo de archivo en el que se quiera buscar dicho elemento. También se han introducido, aunque de forma estática, etiquetas con contenido semántico Dublin Core para aportar otra manera válida y reconocida de proporcionar información descriptiva sobre el recurso.

En cuanto a minería web, se ha introducido un fragmento de JavaScript básico de Google Analytics en la página inicial con la intención de recopilar, ver y analizar datos sobre el tráfico del sitio web. De esta forma se puede ver qué contenido es el más visitado, cuál es el promedio de visitas de la página, la hora local de las visitas, etc.

En definitiva, en estos documentos se efectúan las llamadas a los métodos de la capa de acceso a los datos y se implementan todas las funcionalidades de la página explicadas en el apartado de diseño.

Si se desea ver el código implementado en las páginas se puede encontrar también en el CD adjunto al proyecto, en las carpetas "Observatorio" y "AplicacionObservatorio".

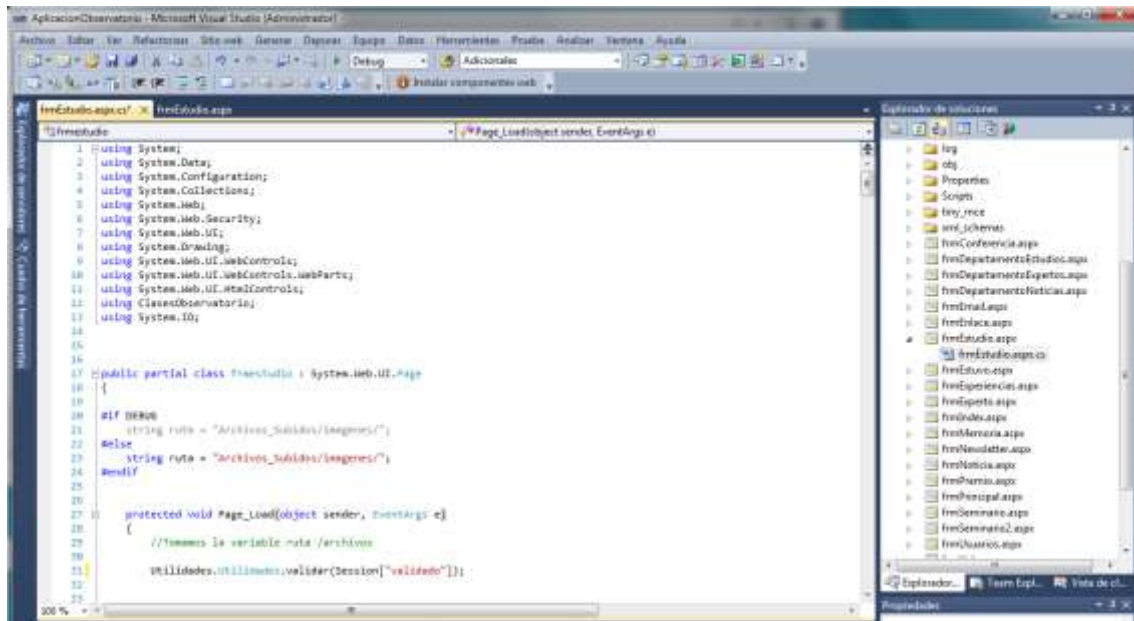


Figura 5.42 Página de ASP.NET “.aspx.cs”. Capa de negocio.

5.3.4 – Despliegue de la Aplicación

Una vez está comprobado el buen funcionamiento de la aplicación y del sitio web en local, ha llegado el momento de desplegarlo en el servidor en el que está alojada la web corporativa de la universidad.

Se ha decidido que la web del Observatorio quede integrada en el directorio “Vida Universitaria” de la web de la Universidad CEU Cardenal Herrera y se proporcionará un vínculo en la Home para acceder.

Así pues, el primer paso es simple, conectarse desde un ordenador perteneciente a la red de la universidad, a la unidad de red en la que se encuentra la web corporativa, acceder al directorio “Vida Universitaria” y dejar caer la carpeta con todos los archivos implementados del Observatorio.

Para desplegar la aplicación de gestión el paso es el mismo, pero éste se alojará dentro de la estructura de directorios “/admin/aplicaciones/”, donde se alojan todas las aplicaciones implementadas por el servicio de Comunicación Digital relacionadas con otros departamentos o servicios de la web.

El siguiente paso es configurar el IIS para que interprete las páginas de las nuevas aplicaciones. Para ello, hay que efectuar una conexión a escritorio remoto a la dirección IP del servidor. Se accederá entonces a un escritorio de Windows Server 2003. Una forma de acceder al administrador de IIS es la siguiente:

Pulsando con el botón derecho en MI PC y seleccionando la opción que pone "Administrar". Esto abre la "Administración de equipos". En la lista de la izquierda, en la parte de abajo aparece "Servicios y aplicaciones", entre los que se encuentra la opción buscada: "Servicios de Internet Information Server".

Una vez aparece la ventana de administración de IIS, se busca en el desplegable de la izquierda el directorio en el que está alojada la aplicación implementada. Se pincha con el botón derecho y se hace clic en propiedades. Se abrirá una nueva ventana en la que se podrán configurar varios detalles.

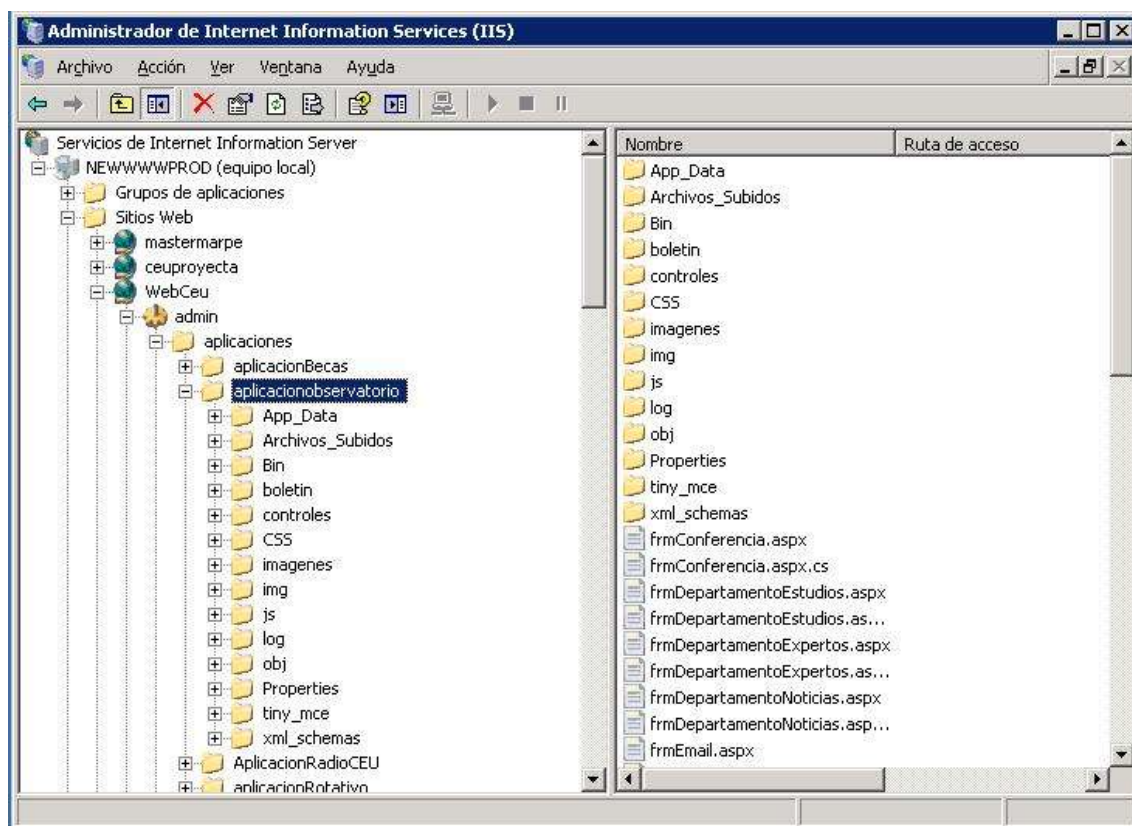


Figura 5.43 Administrador de IIS

En primer lugar las propiedades relativas al directorio: ruta de acceso, posible redirección, permisos de lectura, escritura, ejecución, acceso al código fuente de secuencia de comandos, examen de directorios, registro de visitas y permisos de indexación de recursos.



Figura 5.44. Propiedades de la aplicación del Observatorio

A continuación las propiedades relativas a los encabezados HTTP, donde se establece el tiempo de caché del sitio. Esto sirve para que no haya un desbordamiento de la pila de conexiones por consultas a la base de datos, por ejemplo, cuando al acceder a la página de inicio se muestran las últimas noticias, o cuando se accede al mapa del sitio o al rss, que ejecutarían automáticamente una consulta por cada usuario que accediera de no haber establecido un tiempo de caché.

Se trata de aumentar rendimiento y mantener la estabilidad de la aplicación aunque las consultas no sean completamente a tiempo real, ya que al abrir muchas conexiones desde un DataView sin que éstas se cierren provoca desbordamiento y en consecuencia error 500 en el sistema.



Figura 5.45. Propiedades de IIS. Encabezados HTTP

También hay que establecer la página de inicio de la aplicación.



Figura 5.46. Configurar página de inicio

Indicar la versión de ASP.NET, aunque en este caso está condicionada por la versión establecida desde el directorio raíz de la web corporativa y establecer la ruta de acceso virtual, que no hay que confundir con la ruta física de los archivos.



Figura 5.47 Ruta de acceso virtual

Y finalmente, establecer la política de seguridad de los directorios, en el que se indica si se desea habilitar el acceso anónimo y modificar los métodos de autenticación para el recurso, conceder o denegar acceso al recurso usando las direcciones IP o los nombres de dominio de internet y habilitar certificados cliente para cuando tenga acceso al recurso.



Figura 5.48. Política de seguridad del recurso

6 - Pruebas y resultados

6.1 - Descripción de experimentos

Ha llegado el momento de realizar las pruebas de implementación del sitio web y de su aplicación de gestión. En primer lugar se van a probar los distintos diagramas de secuencia diseñados, mostrando el proceso de navegación mediante capturas de pantalla.

6.1.1 - Aplicación de gestión

El primer diagrama representaba la secuencia de recuperación de credenciales de un usuario no identificado que había olvidado o perdido la contraseña.

En primer lugar, el usuario debe escribir la URL de la página de identificación de la aplicación de gestión, que se encuentra dentro del directorio asignado a las aplicaciones en el dominio de la web corporativa de la Universidad CEU Cardenal Herrera. La página de inicio de la aplicación se llama “login.aspx”, pero al establecerla como página de inicio desde el IIS es suficiente con escribir el nombre del directorio en el que está alojado la aplicación para ser redireccionado a esta página.

<http://www.uchceu.es/admin/aplicaciones/aplicacionobservatorio>

login.aspx



Figura 6.1. Página de inicio de la aplicación. Identificación de usuario.

Para acceder a la página de recuperación de credenciales hay que hacer clic en el texto con hipervínculo “¿Has olvidado la contraseña?”. Esta página se muestra a continuación.

frmEmail.aspx



Figura 6.2 Página recuperación de contraseña de usuario.

En esta página se pide la dirección de correo electrónico del usuario para comprobar si está registrado en la base de datos. En caso afirmativo, al hacer clic en “Recuperar Clave”, se consultará la contraseña del usuario y se le adjuntará en un e-mail enviado a dicha dirección de correo. En caso negativo aparecerá el mensaje “Los datos no existen en la base de datos”.

El siguiente diagrama mostraba la secuencia de introducción de un nuevo usuario en la base de datos. Para ello es necesario identificarse como administrador en la página “login.aspx”. En caso de que el proceso de identificación sea exitoso se mostrará la página inicial.

frmIndex.aspx

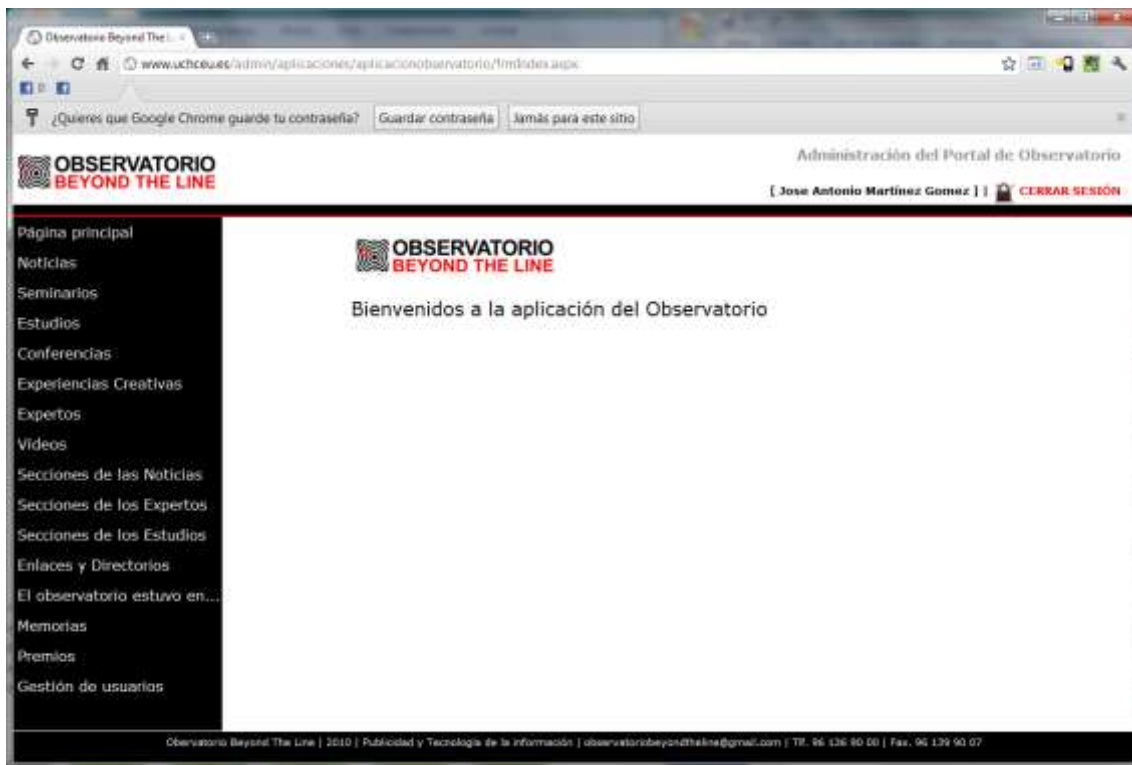


Figura 6.3. Página inicio para usuario administrador.

En la parte izquierda se puede acceder a todos los apartados de la aplicación en este caso, ya que el usuario está identificado como administrador. En la parte superior derecha aparece el nombre de usuario, recogido en una variable de sesión, y junto al mismo se encuentra el vínculo para cerrar la sesión.

Cabe decir que, tanto en la aplicación como en el sitio web, el tamaño de pantalla es variable, y se expandiéndose cuando se maximiza la pantalla y contrayéndose al minimizar hasta un mínimo horizontal de 800 píxeles.

El vínculo para acceder a la gestión de usuarios es el último del listado de la izquierda.

frmUsuario.aspx

Observatorio Beyond The Line

www.uchc.edu.es/admin/aplicaciones/aplicacionobservatorio/frmUsuario.aspx

Administración del Portal de O

[Jose Antonio Martínez Gomez]

OBSERVATORIO
BEYOND THE LINE

Página principal
Noticias
Seminarios
Estudios
Conferencias
Experiencias Creativas
Expertos
Videos
Secciones de las Noticias
Secciones de los Expertos
Secciones de los Estudios
Enlaces y Directorios
El observatorio estuvo en...
Memorias
Premios
Gestión de usuarios

Gestión de Usuarios

Login:

Nombre:

Contraseña:

Confirmar Contraseña:

Email:

Rol:

Fecha de caducidad:

febrero de 2012						
lun	mar	mié	jue	vie	sáb	dom
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	1	2	3	4
5	6	7	8	9	10	11

Observatorio Beyond The Line | 2010 | Publicidad y Tecnología de la información | observatoribeyondtheline@gmail.com | Tlf. 96 136 90 00 | Fax. 96 139 90 07

Figura 6.4. Gestión de usuarios scroll 1

Rol:

Elige las secciones accesibles para el colaborador

- ☐ Página Principal
- ☐ Noticias
- ☐ Seminarios
- ☐ Estudios
- ☐ Conferencias
- ☐ Experiencias Creativas
- ☐ Expertos
- ☐ Videos
- ☐ Secciones de Noticias
- ☐ Secciones de Expertos
- ☐ Secciones de Estudios
- ☐ Enlaces
- ☐ Estuvo
- ☐ Memorias
- ☐ Premios

Como puede verse, la página necesita de un scroll vertical para verse completa. En la parte superior se facilitan los campos de texto del formulario para la inserción del nombre de usuario, el nombre real, la contraseña, la confirmación de la misma, la dirección de correo, la fecha de caducidad del usuario (que dejará de permitírsele el acceso cuando pase dicha fecha) y el rol.

En caso de elegir rol "Colaborador" aparecerá el listado que se observa a la izquierda en el que se podrá elegir los apartados de gestión del sitio web a los que el futuro usuario tendrá acceso, seleccionando cada uno de los checkbox correspondientes.

Figura 6.5. Listado de secciones accesibles
para el colaborador

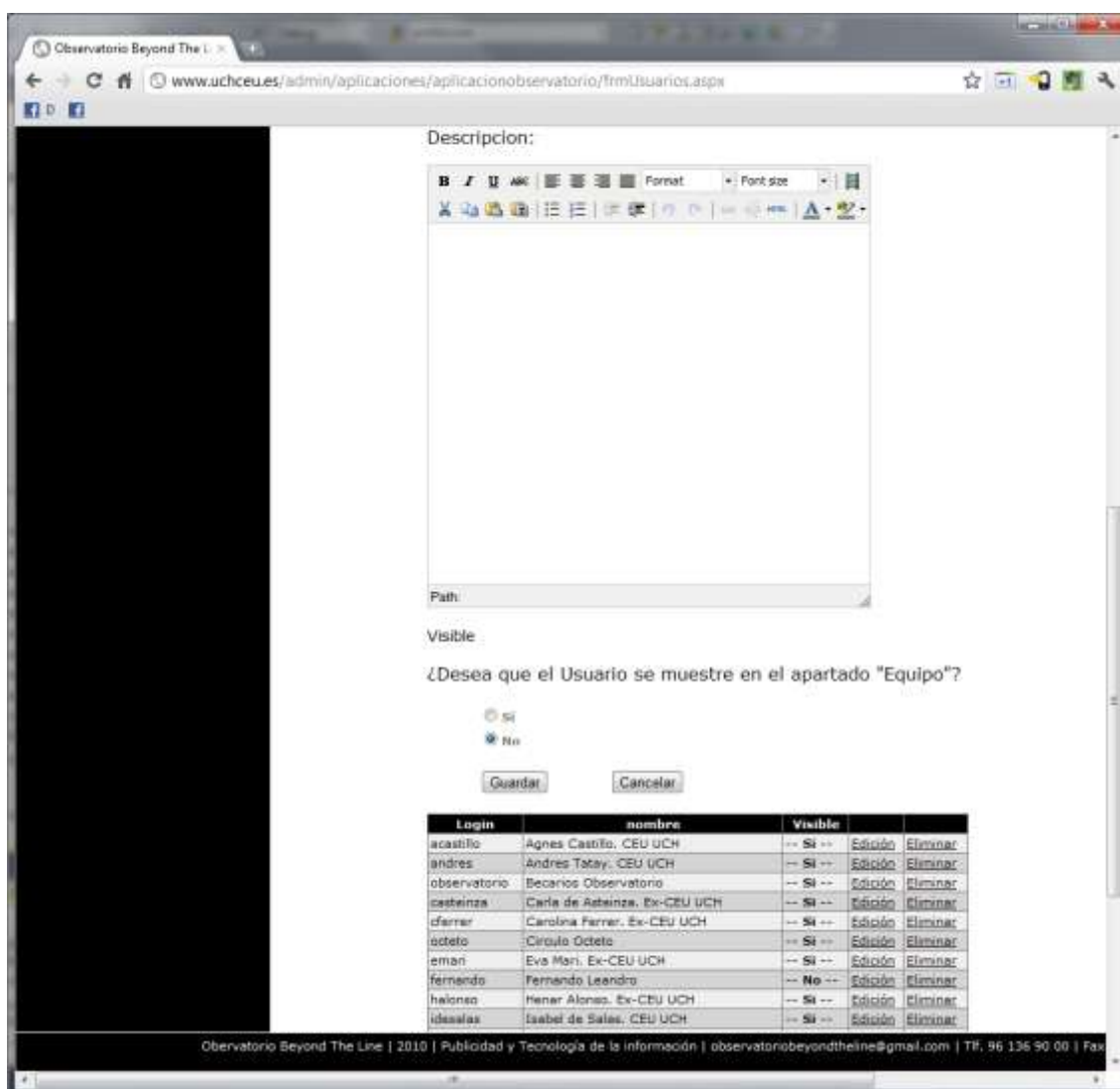


Figura 6.6 Gestión de usuarios scroll 2

Al hacer uso del scroll para acceder a la parte inferior de la página se puede observar un editor de contenidos WYSIWYG (What You See Is What You Get), habilitado para realizar una descripción del usuario que será mostrada en el apartado “Equipo” de la web en caso de que el usuario haya sido seleccionado como “Visible”.

Además de texto, el editor permite la inclusión de estilos, en modo diseño o con HTML, así como la inclusión de vínculos a ficheros o contenido multimedia.

Una vez rellenados todos los campos hay que hacer clic en el botón guardar para que se almacenen en la base de datos. Si algún campo no ha sido rellenado o las contraseñas no coinciden la operación no se llevará a cabo y se emitirá a cambio un mensaje de alerta.

Finalmente, en la parte inferior se muestra una tabla con un listado de los usuarios que tienen acceso a la aplicación, desde la cual se pueden editar o borrar. Antes de borrar aparecerá un mensaje de confirmación.

El posterior diagrama explicaba la secuencia que se producía al crear una conferencia y asignarle un seminario ya existente. Una vez implementado se muestra a continuación.



Figura 6.7. Pantalla de inicio identificado como profesor

Así se muestra la página inicial cuando se identifica un usuario con privilegios de profesor. A la izquierda aparecen todas las secciones excepto la sección de gestión de usuarios. En esta ocasión el usuario va a acceder a la sección de Conferencias con la intención de crear una nueva.

frmConferencias.aspx

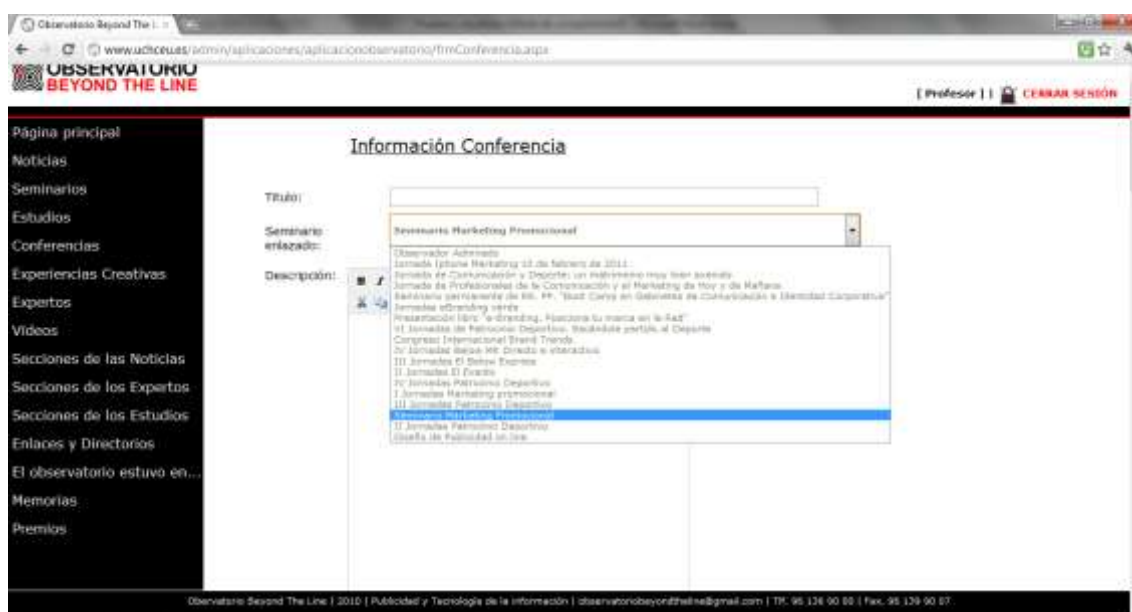


Figura 6.6. Pantalla Conferencia identificado como profesor (1)

De nuevo se accede a una pantalla que necesita de un scroll vertical para navegar. En la parte superior se dispone de un cuadro de texto para indicar el título de la nueva conferencia y posteriormente un desplegable que contiene los nombres de todos los seminarios almacenados en la base de datos, obtenidos mediante consulta. Justo debajo se encuentra el editor de contenidos habilitado para describir la conferencia.

Observatorio Beyond The Line

www.uchceu.es/admin/aplicaciones/aplicacionobservatorio/frmConferencia.aspx

Archivo: (título para el archivo)
 (ruta)
 Seleccionar archivo No se ha ... archivo
 Guardar Archivo Eliminar Archivo

Imagen: (ruta)
 Seleccionar archivo No se ha ... archivo
 Pequeña Mediana Grande
 Guardar Imagen Eliminar Imagen

Visible: ☒ Si ☐ No

Guardar Conferencia Cancelar

Buscar por Título Mostrar Todos

Título	Visible	Archivo	Seminario	
Conferencia de Marçal Melins en la entrega de la placa "Observador Admirado"	Si		(en asignar)	Edición Eliminar
El profesor José Amiguet habló sobre la Reputación	Si		Presentación libro "e-Branding"	Edición Eliminar

Observatorio Beyond The Line | 2010 | Publicidad y Tecnología de la información | observatoribeyondtheline@gmail.com | Tlf. 96 136 90 00 | Fax. 1

Figura 6.7. Pantalla Conferencia identificado como profesor (2)

Al descender aparece la opción de adjuntar un fichero de texto desde el propio ordenador, en formato .pdf, .docx o .docx, que será almacenado en el servidor de la universidad, incluyendo una referencia a su localización en la base de datos. También se puede adjuntar de igual forma una imagen (a escoger entre tres tamaños predefinidos), elegir si se desea que la conferencia almacenada sea visible o no (puede almacenarse como borrador en primera instancia a la expensas de modificaciones posteriores) y filtrar las consultas a la tabla de conferencias de la base de datos a través de un buscador con filtro por título. El resultado de la búsqueda se mostrará en la tabla inferior. Para guardar la conferencia, basta con un clic en el botón habilitado para ello.

El siguiente diagrama explicaba un ejemplo de secuencia para un usuario colaborador, y se centraba en el procedimiento de subir un archivo al servidor y adjuntarlo a una entrada en el apartado de expertos.

A continuación se va a mostrar una captura del panel de inicio para un usuario colaborador que sólo tiene acceso a la inserción, modificación y borrado de entradas en la sección "Estudios".



Figura 6.8. Panel de inicio para un usuario colaborador con privilegios para el apartado Estudios.

Para adjuntar un archivo a la sección expertos sólo hay que hacer clic en el botón “Seleccionar archivo”, elegir el archivo del ordenador local a través de la ventana nueva que aparece y hacer clic en el botón “Guardar Archivo”. El archivo se guardará en la carpeta del servidor especificada mediante programación y la URL de referencia se guardará en la tabla Archivos a la espera de que su “id” sea asociado a la clave ajena de la tabla “Estudios”.

frmEstudios.aspx

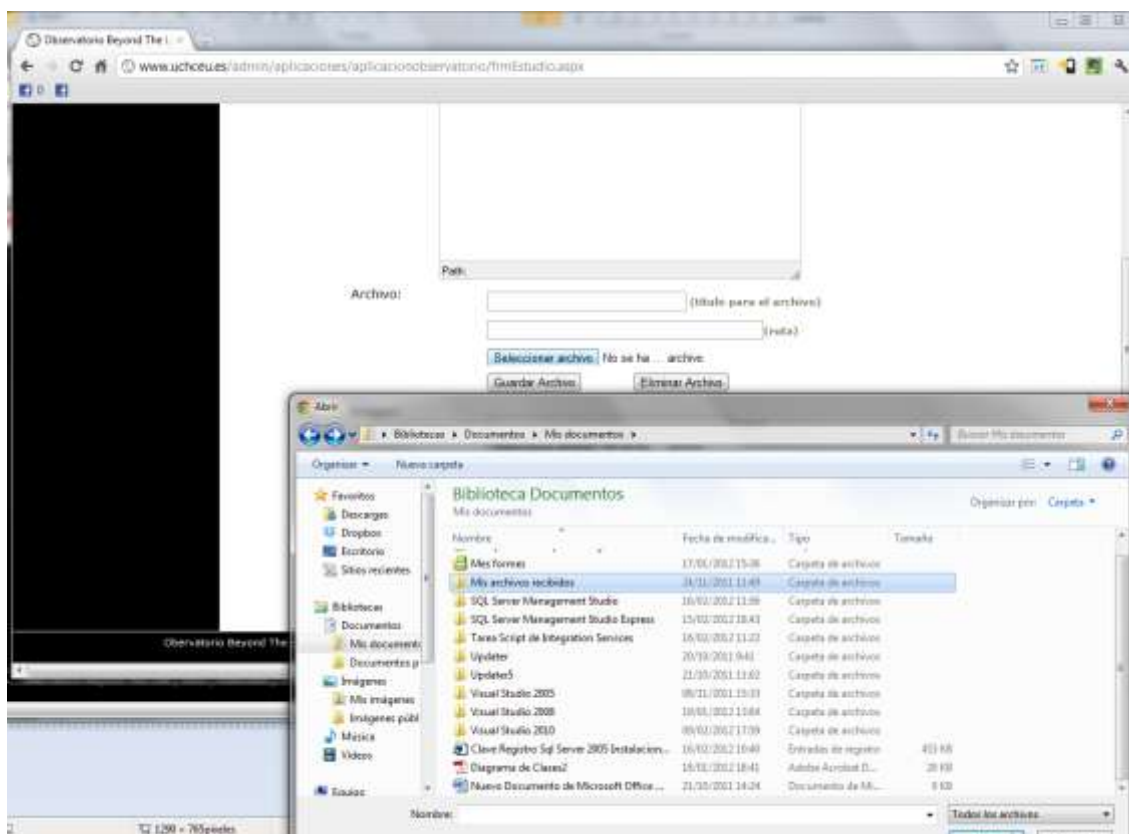


Figura 6.9. Guardar un archivo en la sección Estudios.

El penúltimo diagrama explicaba la secuencia “borrar un experto y su imagen”. El proceso es muy sencillo. Únicamente hay que buscar el experto en la tabla inferior en la que están listados, y hacer clic en la última columna, en la que pone “eliminar”. Entonces aparecerá un mensaje en el que se pedirá la confirmación. Al pulsar en “aceptar” se borrará automáticamente tanto el experto como su imagen asociada.

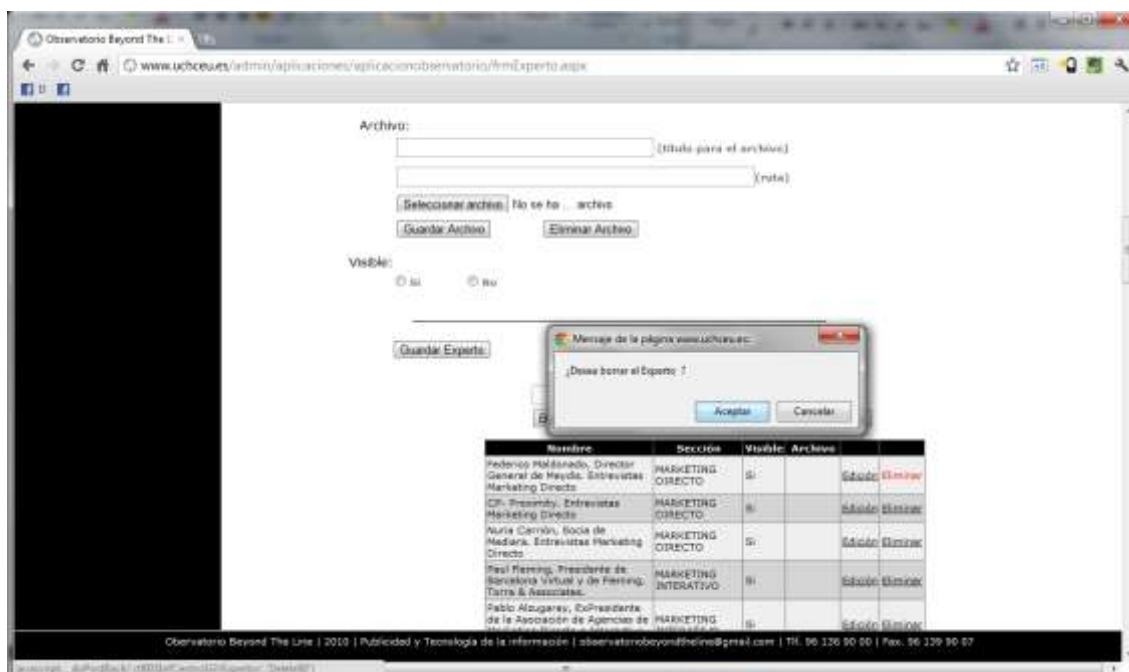


Figura 6.10. Guardar un archivo en la sección Estudios.

Finalmente se explicó la secuencia de edición de una noticia encontrada previamente mediante el buscador y su cambio de estado. Este proceso también es bastante intuitivo, ya que sólo hay que escribir una palabra del título de la noticia por el cual queramos filtrar y elegir la noticia de la tabla haciendo clic en la celada “Edición”.

frmNoticia.aspx

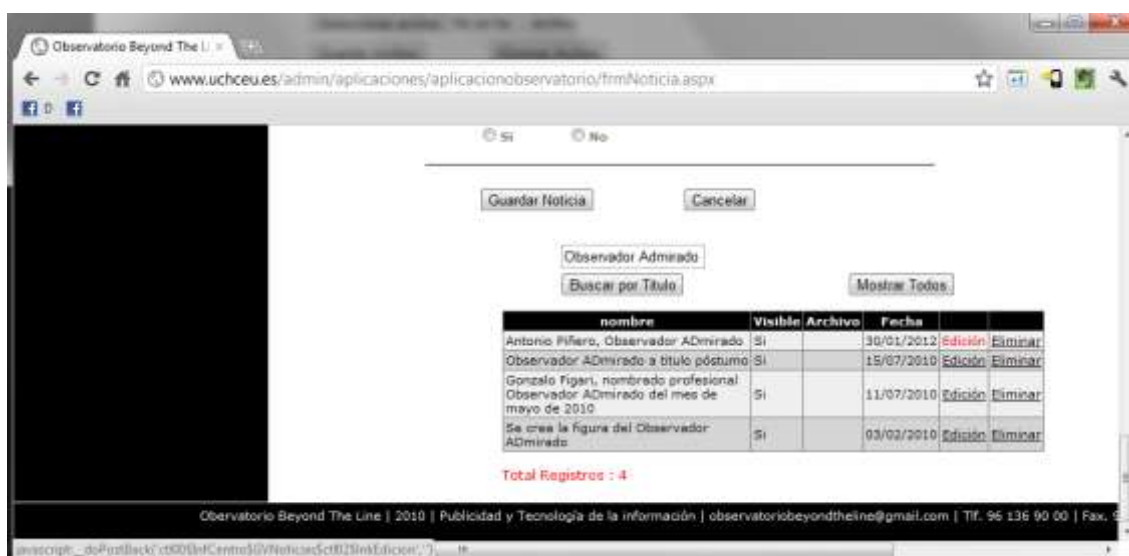


Figura 6.11. En este ejemplo se ha filtrado por las palabras “Observador Admirado” para buscar la noticia y editarla.

Una vez pulsada la opción “Edición” se rellenarán los campos con los datos de la noticia seleccionada y podrán editarse pulsando el botón “Guardar Noticia”.

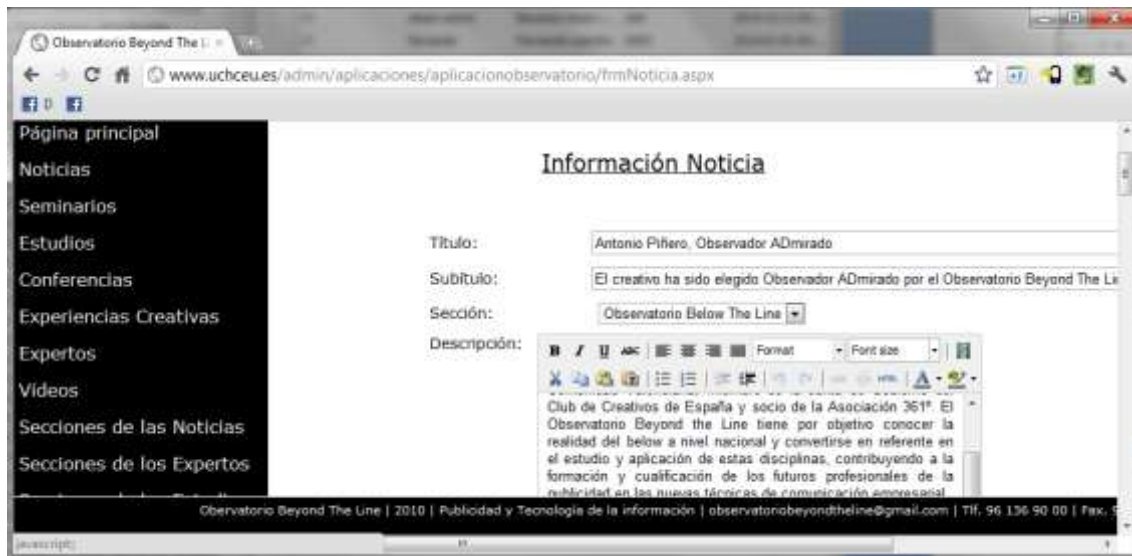


Figura 6.12. Noticia recuperada con la intención de editarla.

6.1.2 - Sitio web del Observatorio

El primer diagrama de la página web mostraba la secuencia de descarga de un archivo que se había adjuntado a un Estudio. A continuación se va a mostrar las capturas del proceso de navegación hasta obtener dicho fichero.

La URL de acceso directo a la web del Observatorio Bellow es la siguiente:

http://www.uchceu.es/vida_universitaria/observatorio

Sin embargo, también se puede acceder desde la página inicial de la web corporativa de la Universidad CEU Cardenal Herrera: <http://www.uchceu.es>, desde la página principal o navegando a través de la sección Vida Universitaria.

La página inicial del Observatorio presenta una gran cantidad de contenido que se ha distribuido atendiendo a las especificaciones de los clientes. Horizontalmente tiene un tamaño mínimo de 800 píxeles y se extiende hasta el tamaño máximo de resolución de cada pantalla. La presencia de Scroll vertical ha sido inevitable para abordar la presentación de toda la información requerida y por eso se presenta en dos capturas de pantalla.

Para acceder a la sección estudios simplemente hay que acceder al menú superior y hacer clic en la pestaña correspondiente.

Index.aspx



Figura 6.13. Página de inicio del Observatorio Beyond The Line, parte superior



Figura 6.14. Página de inicio del Observatorio Beyond The Line, parte inferior

Una vez se accede a la página de Estudios, aparece el listado de Secciones de Estudios por los que están agrupados los estudios almacenados. Hay que hacer clic la sección que contenga el estudio elegido y pinchar posteriormente en el estudio elegido. En la misma página aparecerá la información relativa a ese estudio y un enlace al archivo que se haya adjuntado al contenido.

Sólo hay que hacer clic sobre el enlace a ese archivo y si se trata de un fichero .doc o .docx se descargará, mientras que si se trata de un pdf se abrirá en una pestaña nueva del navegador.

Estudios.aspx

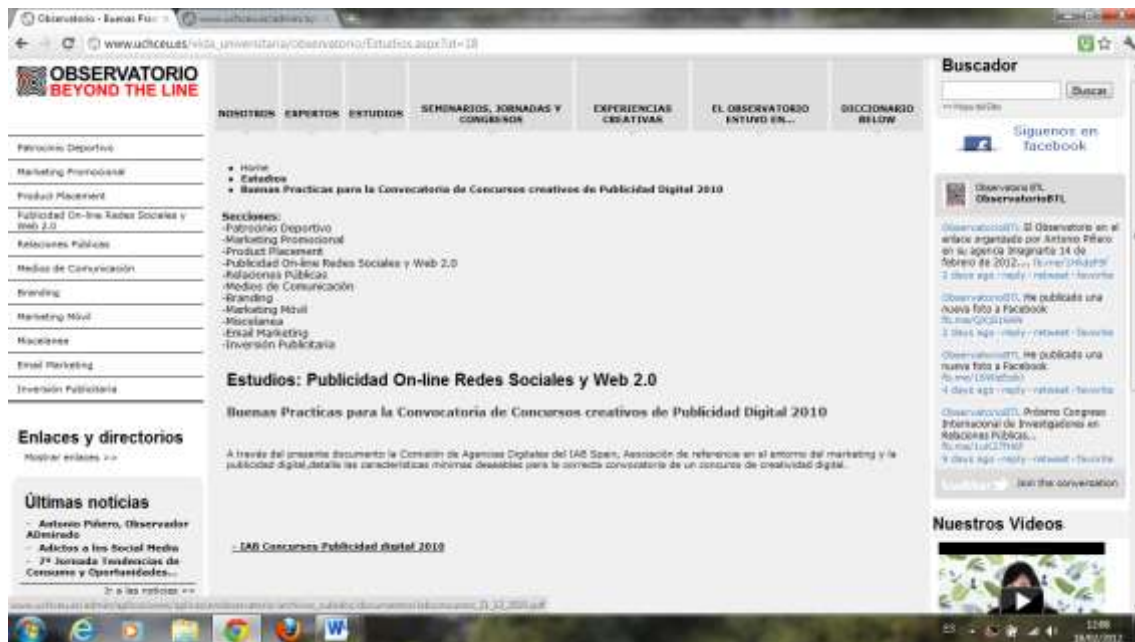


Figura 6.15. Sección Estudios con enlace a fichero en la parte inferior y fichero abierto en nueva pestaña

En el diagrama siguiente del apartado de diseño se representaba la secuencia de tareas que se ejecutan al acceder al mapa del sitio de la web. En realidad la clase que muestra el mapa del sitio se ejecuta en todas las páginas de la web. Como puede verse en la captura de imagen anterior, en la parte superior, justo debajo del menú se muestra la ruta de acceso a la página visitada. Pero también se puede acceder al mapa del sitio completo haciendo clic en el enlace situado debajo del buscador, en la parte superior derecha de la página. De esta forma se puede ver el mapa del sitio completo y acceder por tanto a cualquier apartado de la web desde ahí:

MapaSitio.aspx

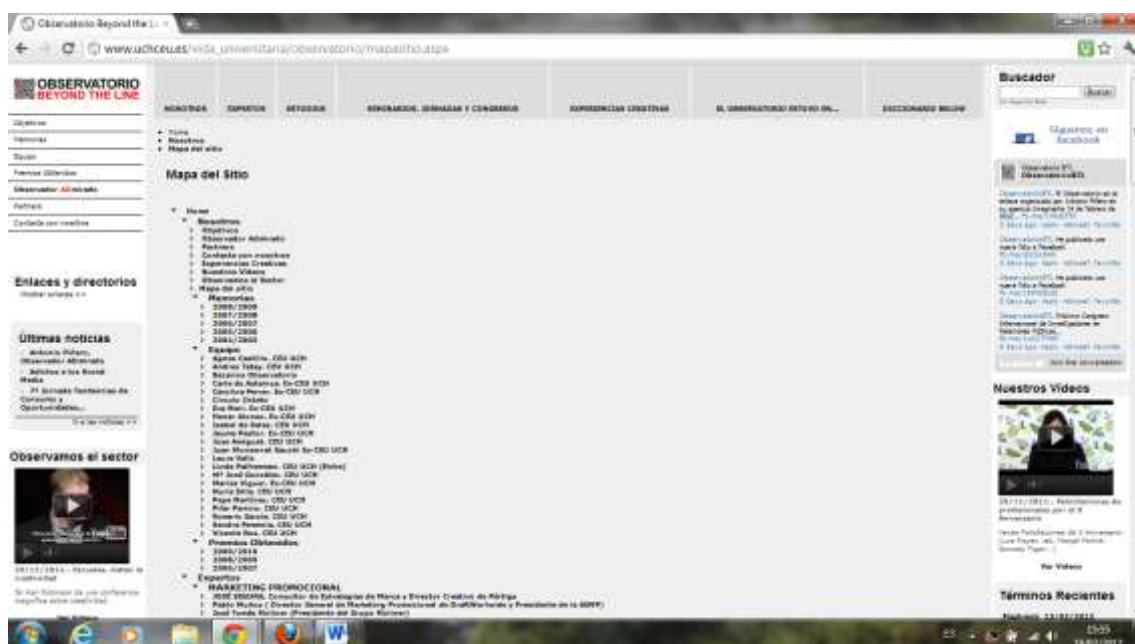


Figura 6.16. Mapa del sitio de la Web del Observatorio

Por último se presenta el contenido de la sección Observador Admirado, que a diferencia del contenido del resto de secciones de la web, no se obtiene de la base de datos sino desde el RSS del blog del observatorio situado en un servidor de Wordpress.

Así pues, con este método es posible acceder al contenido y mostrarlo maquetado y en tiempo real sin necesidad de hacer uso de un servicio web.

ObservadorAdmirado.aspx

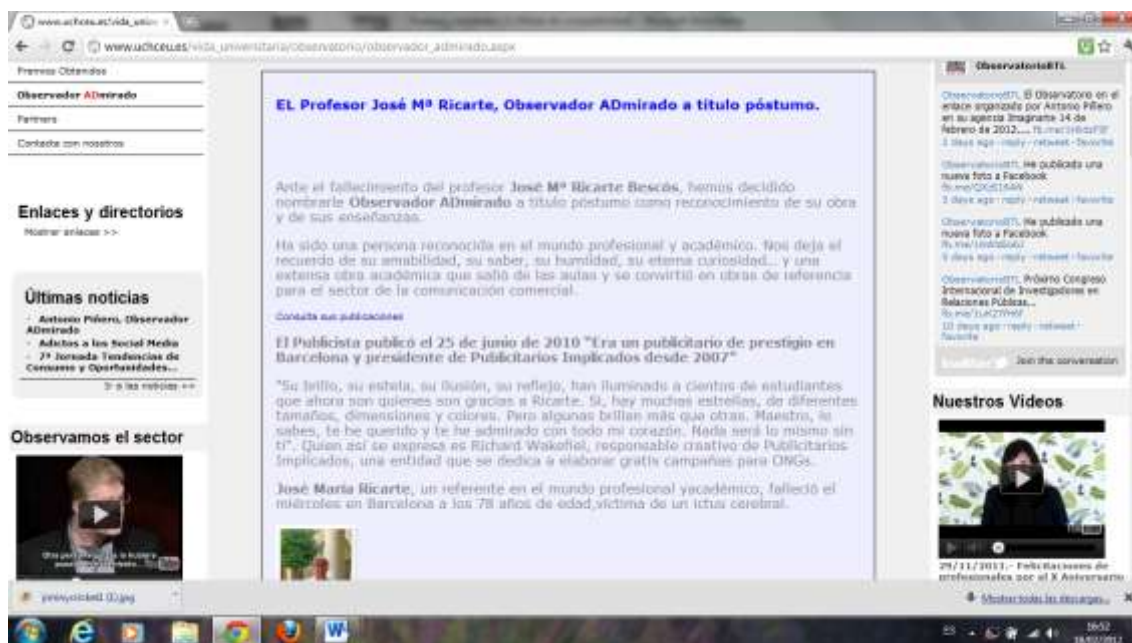


Figura 6.17. Sección Observador Admirado, cuyo contenido se obtiene a través del RSS del blog del Observatorio.

6.2 - Pruebas y resultados

6.2.1 - Accesibilidad

A continuación se va a analizar el sitio web en términos de accesibilidad mediante el Test de Accesibilidad Web (TAW). La aplicación de gestión no puede ser analizada en estos términos por motivos de seguridad, ya que las herramientas de que se disponen no tienen acceso a sitios que requieren autenticación de usuario.

El análisis de la web se hizo concretamente mediante TAW 3.0, el 18/2/12 a las 11:42 AM, utilizando las normas WAI del 5 de mayo de 1999, y en base a un nivel de conformidad "AAA", por el cual se pretende que todos los puntos de prioridad 1, 2 y 3 se satisfagan.

El análisis del test está dividido, por tanto, en posibles problemas de accesibilidad localizados de cada uno de estos tres niveles de prioridad. La mayoría de los problemas detectados por el test necesitarán de comprobación manual.

Junto a cada problema detectado se ofrece una alternativa para solucionarlo.

6.2.1.1 - Nivel de prioridad 1.

Un desarrollador de contenidos de páginas Web tiene que satisfacer este punto de verificación. De otra forma, uno o más grupos de usuarios encontrarán imposible acceder a la

información del documento. Satisfaciendo este punto de verificación es un requerimiento básico para que algunos grupos puedan usar estos documentos Web.

1. Hay que asegurarse de que este documento puede ser leído sin necesidad de utilizar hojas de estilo.

- Línea 8: `<link rel="stylesheet" type="text/css" href="estilos/estilos.css" />`

Tras eliminar dicha línea del fichero.html, se puede comprobar un cambio en el formato de la página, en el que sus elementos cambian de posición y se superponen, y en el que se observa que se pierde información relativa a las noticias de portada, que se presenta de forma dinámica, así como una gran cantidad de submenús y de imágenes de los que debería mostrarse información alternativa sin necesidad de recurrir a la hoja de estilo, a la que debería delegarse únicamente la presentación del contenido, y no el contenido mismo. Algunos ejemplos de líneas de código en las que se muestra esta dependencia de las hojas de estilo en cuanto al contenido son las siguientes:

- Línea 118: `<span style="font-weight:bold;">`
- Línea 365: `<p style="margin-bottom:-10px;font-family: Arial, Helvetica, sans-serif;margin-bottom: -10px; text-align: center;">`
- Línea 365: `<span style="font-size: large;">`

Estas irregularidades no tienen solución trivial, ya que provienen del código introducido dinámicamente por los usuarios a través de la aplicación de gestión. Hay que limitar desde la aplicación de gestión la posibilidad de introducir código html o estilos en el editor de contenidos.

2. Hay que asegurarse de que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, applets u otros objetos programados. Si esto no es posible, hay que proporcionar información equivalente en una página alternativa accesible. Además, los equivalentes de un contenido dinámico deben ser actualizados cuando cambia el contenido dinámico.

- Línea 266: `<embed src='http://www.youtube.com/v/nPB-41q97zg?fs=1&hl=es_ES' type='application/x-shockwave-flash' width='200' height='167'>`
- Línea 620: `<embed src='http://www.youtube.com/v/RJoETghGiZI?fs=1&hl=es_ES' type='application/x-shockwave-flash' width='200' height='167'>`

Estos vídeos contenidos en Youtube han sido embebidos por los usuarios desde la aplicación de gestión. La recomendación de este punto es programar en la aplicación una imagen alternativa para cada marco que sea siempre una página Web. Si se especifica una imagen u otro tipo de contenido para un marco es difícil proporcionar un texto alternativo para ese contenido.

3. También avisa sobre el contenido alternativo que deberían proporcionar las imágenes.

- *Línea 382: *

Por ejemplo esta imagen, subida también dinámicamente a través de la aplicación. En este caso habría que programar desde la aplicación de gestión que se presente el texto alternativo introduciendo en la etiqueta alt, por ejemplo, el título que los usuarios han puesto en la entrada.

6.2.1.2 - Nivel de prioridad 2.

Un desarrollador de contenidos de páginas Web debería satisfacer este punto de verificación. De otra forma, uno o más grupos encontrarán dificultades en el acceso a la información del documento. Satisfaciendo este punto de verificación se eliminarán importantes barreras de acceso a los documentos Web.

1. Hay que utilizar unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo. El test ha comprobado de forma automática que los siguientes elementos utilizan unidades de medida absolutas en lugar de unidades de medida relativas.

- *Línea 516: <table width="230" border="0" align="center" cellpadding="0" cellspacing="0" class="tabla_gris">*
- *Línea 531: <input name="ctrBuscar1\$txtBuscar" type="text" id="ctrBuscar1_txtBuscar" class="Verdana" style="width:139px;" />*
- *Línea 672: <td class="verdanapie" style="height: 19px">*

En lugar de poner estas medidas se puede utilizar la unidad "em" para fijar el tamaño de letra. CSS permite utilizar unidades relativas incluso en posiciones absolutas. Por tanto, se puede colocar una imagen que será alejada por "3em" desde el comienzo del elemento que contenga. Es una distancia fija, pero relativa al tamaño de letra actual, con lo cual se escalará adecuadamente. También se pueden utilizar porcentajes.

2. Hay que Evitar características desaconsejadas por las tecnologías W3C. El test ha detectado el uso de elementos desaconsejados u obsoletos en la especificación html 4.

- Línea 57: ``
- Línea 64: `<td height="200" valign="top">`
- Línea 69: `<td width="7%" height="1" bgcolor="#000000" >`
- Línea 177: `<table width="230" border="0" align="center" cellpadding="0" cellspacing="0" class="tabla_gris">`

Hay que evitar en la medida de lo posible el uso de atributos relativos al estilo, delegándolos en las hojas de estilo.

3. Hay que asegurarse de que los colores de fondo y primer plano en imágenes tengan suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o en pantallas en blanco y negro.

- Línea 182: ``
- Línea 184: ``
- Línea 234: ``
- Línea 236: ``

El uso de tonalidades grises para diferenciar los elementos del fondo blanco puede no ser la decisión más adecuada, y aunque viene impuesta por el cliente, hay que comprobar que se cumple la condición recién expuesta. En este caso se utilizará el test “vischeck” para comprobar las posibles dificultades de percepción de las personas con deficiencias de visión.

4. Para script y applets, hay que asegurarse de que los manejadores del evento son entradas independientes del dispositivo.

Para los siguientes casos:

- Línea 74: `<tr onmouseover="this.bgColor= '#efefef';this.style.cursor = 'hand';" onmouseout="this.bgColor='white'">`
- Línea 545: `<a href="http://www.facebook.com/group.php?gid=153637982020" target="_blank" title="Síguenos en FACEBOOK" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Síguenos en Facebook','', 'imagenes/face2.gif',1)">`

Es recomendable duplicar los manejadores del evento para proporcionar independencia del dispositivo, ya que los efectos no son exclusivamente visuales. Además, para scripts, hay que especificar controladores de evento lógicos (de aplicación) en lugar de controladores de evento dependientes de dispositivo. Por último, los elementos con script no son accesibles para usuarios que no poseen javascript instalado.

- Línea 10: `<script src="js/flash_activex.js" type="text/javascript">`
- Línea 11: `<script type="text/javascript" src="http://w.sharethis.com/button/buttons.js">`
- Línea 266: `<embed src='http://www.youtube.com/v/nPB-41q97zg?fs=1&hl=es_ES' type='application/x-shockwave-flash' width='200' height='167'>`
- Línea 555: `<script type="text/javascript" src="http://widgets.twimg.com/j/2/widget.js">`

Hay que proporcionar elementos alternativos para poder acceder a los contenidos en caso de no tener javascript.

Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no hay que provocar apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario.

Los siguientes enlaces abren nuevas ventanas que pueden desorientar a algunos usuarios. Se recomienda informar al usuario.

- Línea 273: `<a id="Medios_hlnkTexto1" href="http://www.youtube.com/watch?v=nPB-41q97zg" target="_blank">`
- Línea 306: `<a class="verdanapeq" href="mailto:observatoriobeyondtheline@gmail.com" target="_blank">`
- Línea 313: `<a href="RSS.aspx" target="_blank">`
- Línea 339: `<a href="DiccionarioBelow/Default.aspx" target="_blank">`
- Línea 545: `<a href="http://www.facebook.com/group.php?gid=153637982020" target="_blank" title="Síguenos en FACEBOOK" onmouseout="MM_swapImgRestore()" onmouseover="MM_swapImage('Síguenos en Facebook','imagenes/face2.gif',1)">`

Hay que identificar claramente el objetivo de cada vínculo. En este caso el vínculo es muy largo.

- Línea 627: *29/11/2011.- Felicitaciones de profesionales por el X Aniversario*

En estos otros enlaces se comparte el mismo texto pero están vinculados a recursos diferentes.

- Línea 284: `<a id="Medios_hlnkExpertos" href="mediosCEU.aspx">Ver Videos</a>`
- Línea 638: `<a id="ctrVideo_hlnkExpertos" href="NuestrosVideos.aspx">Ver Videos</a>`

Para finalizar este nivel de prioridad, habría que confirmar el uso de metadatos para incluir información semántica así como la inclusión de un mapa del sitio. Ambas características han sido tratadas en la web y explicadas en esta memoria.

6.2.1.3 - Nivel de Prioridad 3.

Un desarrollador de contenidos de páginas Web puede satisfacer este punto de verificación. De otra forma, uno o más grupos encontrarán dificultades en el acceso a la información del documento. Satisfaciendo este punto de verificación mejorará la accesibilidad de los documentos Web.

1. Hay que indicar el idioma principal del documento, y así se ha hecho, mediante la utilización de la etiqueta `xml:lang`
2. Hay que proporcionar resúmenes de las tablas, así como un título. Para proporcionar un resumen, hay que incluir el atributo "summary". Para la inclusión de un título, se puede recurrir al elemento "caption" o al atributo "title". Se ha detectado de forma automática su ausencia en los elementos:
 - Línea 51: `<table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0">`
 - Línea 53: `<table width="100%" border="0" cellspacing="0" cellpadding="0">`
 - Línea 67: `<table width="98%" border="0" cellspacing="0" cellpadding="0">`
3. Para crear un orden más lógico para navegar sería interesante el uso del atributo "tabindex". Así se navega con el tabulador a través de vínculos, controles de formulario y objetos. Esta web no hace uso de dicho atributo.
4. Hay que proporcionar atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario. Esta web no usa esta opción. Para ello se puede usar el atributo "accesskey". La web analizada no hace uso de esta característica y, dado el número de opciones de navegación que presenta, sería bastante adecuado.

6.2.2 - Test de visión vischeck

A continuación se va a aprovechar una herramienta online que permite visualizar la web de la misma forma que los haría una persona con alteraciones de la visión tales como la deuteranopía, la protanopía y la tritanopía.

En primer lugar se visualiza la página tal y como lo haría una persona afectada de deuteranopía, es decir, con una carencia de sensibilidad para la percepción del color verde. Tan sólo se percibe sensiblemente el cambio en el color rojo de las letras “Beyond The Line” del Logo. El resto de los colores apenas presentan degradación alguna y se distinguen los contenidos a la perfección.



Figura 6.18. Web del Observatorio desde la perspectiva de un usuario con deuteranopía.

Para usuarios con protanopía, es decir, con carencia de sensibilidad al color rojo, las consecuencias son exactamente las mismas, degradación del subtítulo del logo, que por otra parte sigue siendo perfectamente visible.

No hay dificultad alguna para distinguir la información mínimamente relevante, ya que se trata de contenidos en fuentes de color negro sobre fondo blanco en la mayoría de los casos.



Figura 6.19. Web del Observatorio desde la perspectiva de un usuario con protanopía.

Las percepciones son, a nivel general, muy parecidas a las que tendría un usuario con tritanopía, es decir, con carencia de sensibilidad hacia el color azul. En este caso el subtítulo del logo del Observatorio se aprecia en tonos rosados. No afecta en ningún momento a la escala de grises de las pestañas y menos todavía a los contrastes entre fondo y texto, que siguen siendo perfectamente visibles.



Figura 6.20. Web del Observatorio desde la perspectiva de un usuario con tritanopía.

6.2.3 - Usabilidad. Evaluación Heurística.

La heurística es una forma de revisión de sitios web basada en la experiencia de quien realiza la evaluación. Consiste en comprobar diferentes aspectos, desde diseño y navegación hasta operación y utilidad. A continuación se va a realizar una evaluación heurística, tanto de la página web como de la aplicación de gestión de contenidos.

Datos del evaluador

Nombre	José Antonio Martínez Gómez
Perfil	Estudiante de Máster y desarrollador de aplicaciones ASP.NET
Estudios	Máster en Sistemas y Servicios de la Sociedad de la Información

LEYENDA

Impacto: Cuando se produce el problema, es fácil o difícil superar el problema para los usuarios.

Frecuencia: con la que se produce el problema, es frecuente o poco frecuente.

Valoración:

0: No es un problema

1: Problema que no necesita dedicarle tiempo

2: Problema mínimo de usabilidad. La solución es de baja prioridad

3: Problema grave de usabilidad. La solución es de alta prioridad

4: Problema crítico de usabilidad. La solución debe ser inmediata

5: No se aplica

6.2.3.1 – Sitio web del Observatorio

Cabe recordar que los apartados y secciones del sitio web han sido implementados bajo demanda del cliente y que las recomendaciones aquí especificadas se han puesto en común con él a modo de orientación y sugerencia.

Objetivos del sitio web

Conocer la realidad del below a nivel nacional y establecer sus fundamentos teóricos y operativos. A través de la difusión periódica de los resultados obtenidos, el Observatorio pretende convertirse en referente en el estudio y aplicación de estas disciplinas y contribuir a la formación y cualificación de los futuros profesionales de la publicidad en las nuevas técnicas de comunicación empresarial.

Claridad de los Objetivos

La interfaz debe comunicar de manera inmediata su propósito, objetivo y funciones.

Subheurísticos	Impacto	Frecuencia	Observaciones
1. El propósito u objetivo que la interfaz nos transmite es claro y obvio. Con una simple ojeada podemos deducir cuál es su propósito y cuál es su finalidad	1	1	El exceso de información de la página principal puede dificultar la rápida comprensión del contenido, pero el propósito y finalidad del sitio permanece claro.

2. En el caso de existir más de un objetivo, éstos están relacionados con el objetivo o función global y establecen coherencia entre ellos.	2	2	Hay muchas secciones, todas relacionadas y coherentes pero tal vez algunas innecesarias.
3. En caso de existir más de un objetivo, éstos son claros y separados.	3	3	Redundancia de secciones que dificultan la búsqueda.

Adecuación al mundo y a los objetos mentales del usuario / lógica de información.

El sitio está adaptado al mundo real de los usuarios, su lenguaje, conocimientos, etc.

<i>Lógica de la información</i>	Impacto	Frecuencia	Observaciones
1. La presentación de los contenidos es familiar o comprensible para el usuario	2	2	Exceso de contenidos en la página inicial, aunque de fácil acceso a todos.
2. Cuando existen opciones están ordenadas de manera lógica para la forma de pensar del usuario	2	2	Exceso de subcategorías, ya que se pueden añadir desde la aplicación.
3. Las metáforas e iconos que utilizan son entendibles para el usuario y facilitan la interacción con la interfaz.	1	1	Contenido basado casi íntegramente en texto e imágenes, Totalmente entendibles pero el uso de iconos mejoraría la usabilidad.
<i>Formato de la información</i>			
4. La interfaz usa el lenguaje del usuario con palabras, frases y conceptos que le son familiares. El lenguaje debe ser claro, simple y con una sola idea por párrafo	0	0	El contenido está escrito por especialistas en la materia. En cualquier caso es también entendible para usuarios no familiarizados con los conceptos.
5. La información esta estructurada con títulos, negritas y viñetas	2	2	La información introducida dinámicamente a veces no cumple este requisito.
6. El lenguaje y la disposición de la información es asequible y de lectura rápida para el usuario	2	2	El lenguaje sí, la disposición presenta alguna dificultad de navegación.
7. La estructura y presentación de la información no necesita explicaciones o información adicional para su comprensión	2	2	Pese a la densidad de contenidos y secciones, están estructurados de forma comprensible. Sería interesante una reducción más que una reorganización.
8. Los textos y enunciados de los campos están redactados de forma afirmativa	0	0	
9. Se utiliza el lenguaje en forma directa, no impersonal	0	0	

Visibilidad del estado del sistema

El sistema debe tener siempre a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.

<i>Situación</i>	Impacto	Frecuencia	Observaciones
1. La interfaz (en el caso de tratarse de un sitio web) incluye de forma visible el título del sitio, de la sección o el título de la página	1	1	
2. Sabes en todo momento dónde estás posicionado	0	0	El mapa del sitio informa en todo momento.
<i>Acciones posibles</i>			
3. Los vínculos están claramente diferenciados	1	1	Cambios en el puntero del ratón, subrayados y cambios de color
4. No existe información o acciones que necesiten de una acción para su visualización	1	2	Tan sólo hay que hacer clic en los videos para su reproducción

Control y libertad para el usuario

Los usuarios eligen a veces funciones del sistema por error y necesitan a veces una salida de emergencia claramente marcada, eso es salir del estado indeseado sin tener que pasar por un diálogo extendido. Es importante disponer de deshacer y rehacer.

<i>Subheurísticos</i>	Impacto	Frecuencia	Observaciones
1. Se utilizan animaciones no controladas por el usuario	0	0	
2. El scroll no es más grande que dos pantallas	2	2	Cuando una entrada tiene mucho contenido el scroll vertical aumenta.
3. Es posible guardar información de la página Web/sistema interactivo	0	0	
4. Es posible imprimir la información de la página Web/sistema interactivo sin perder información	1	1	
5. Es posible regresar a la página principal desde cualquier sección	0	0	
6. Es posible aumentar y disminuir el tamaño de la letra	0	0	
7. La interfaz se visualiza perfectamente con diferentes resoluciones	1	1	Se expende hasta cualquier resolución a partir de 800 pixeles.
8. La interfaz no introduce tecnologías que requieren versiones actualizadas de navegadores o plugs-ins externos.	2	2	Flash para vídeos de youtube, javascript para pequeñas funcionalidades de presentación.

Consistencia

Consistencia	Impacto	Frecuencia	
1. Las etiquetas de los vínculos tienen los mismos nombres que los títulos de las páginas a las que se dirigen	0	0	
2. Las mismas acciones llevan a los mismos resultados	0	0	

3. Los mismos elementos son iguales en todo el sitio	0	0	
4. La misma información (texto) se expresa de la misma forma en toda la página	2	2	El texto introducido dinámicamente puede cambiar de estilo si el usuario lo estima oportuno.
5. La información esta organizada y es mostrada de manera similar en cada página	2	2	Hay diferencias entre grupos de secciones, pero la organización dificulta el acceso.
<i>Estándares</i>			
6. Se utilizan los colores estándares para los vínculos visitado y no visitados.	2	2	Se usan subrayados y cambio de forma de puntero principalmente. Algún cambio de color, de negro a azul puntualmente.
7. Utiliza de manera diferente a la norma, convenciones o etiquetas universales	1	1	Algún estilo se ha colocado puntualmente en el código HTML.
8. Las áreas de navegación superior, laterales, herramientas de búsqueda y controles (botones, radio buttons...) siguen los estándares comunes de mercado.	1	1	Algunos elementos presentan vínculos a través de imágenes o iconos personalizados.

Prevenir errores

Es más importante prevenir la aparición de errores que generar buenos mensajes de error.

	Impacto	Frecuencia	Observaciones
1. El motor de búsqueda tolera errores tipográficos (mayúsculas), ortográficos (acentos) y acepta palabras similares	4	4	Es necesario escribir una palabra o parte de palabra en el orden exacto y sin errores.

Reconocimiento más que memoria

La página Web/sistema interactivo se basa en el reconocimiento más que en el recuerdo, que permite al usuario interactuar con el sitio de manera fácil y productiva.

	Impacto	Frecuencia	Observaciones
1. EL uso de la interfaz no requiere recordar información de interfaces previas para interactuar con ella. Toda la información necesaria para la interacción se encuentra en la interfaz actual.	0	0	Al ser accesible todo desde el menú de la página de inicio, en un solo click se puede acceder, aunque el menú es un poco largo.
2. Es fácil localizar información previamente encontrada.	2	2	Este punto depende de cómo se estructure el contenido que se introduce dinámicamente a través del gestor de contenidos.

3. La interfaz permite ver y seleccionar, más que recordar y escribir	0	0	
4. La información esta organizada según una lógica reconocida y familiar para el usuario	2	2	El exceso de apartados obliga a poner demasiados menús.
5. Se utilizan iconos relacionados con los contenidos a los que se asocian	2	2	A excepción de unas pocas imágenes, todos los enlaces son puramente texto.
6. La estructura, orden y lógica es familiar e intuitiva para los usuarios.	2	2	La estructura de los menús es todo lo lógica que permite el exceso.

Diálogos estéticos y diseño minimalista

La página Web/sistema interactivo evita toda información o gráfico irrelevante y sólo incluye la información necesaria.

	Impacto	Frecuencia	Observaciones
1. La información visible es la única esencial para realizar la acción. La página no contiene información que es irrelevante o raramente necesaria	0	0	Todo el contenido está relacionado con el objetivo principal de la web.
2. No existe redundancia de información en la página	1	1	Hay redundancia de apartados, pero no de información.
3. La información es corta, concisa y precisa.	2	2	Hay artículos largos, pero el objetivo de los mismos es mostrar esa cantidad de información.
4. Cada elemento de información se distingue del resto y no se confunde con otros	3	3	Algunos artículos podrían ir dentro de varios apartados, lo que puede demostrar que una reducción de secciones puede ser interesante.
5. El texto es fácil de hojear, esta bien organizado y las frases no son muy largas	2	2	Hay artículos largos, pero bien escritos y organizados.
6. Las fuentes son legibles y tienen un tamaño adecuado	2	2	La densidad de contenido obliga a utilizar fuentes algo pequeñas.
7. Las fuentes utilizan colores con suficiente contraste con el fondo	0	0	Fuentes negras sobre fondo blanco o gris blanco.

Flexibilidad y eficiencia de uso

La interfaz facilita y optimiza el acceso a los usuarios independientemente de cuales sean sus características.

	Impacto	Frecuencia	Observaciones
1. Existen aceleradores de teclado para realizar operaciones frecuentes	5	5	Ninguno más allá de los facilitados por los navegadores
2. Si existen, quedan claros cuales son estos aceleradores	5	5	
3. Es posible repetir una acción ya realizada	0	0	

anteriormente de manera sencilla			
4. Se utiliza un diseño líquido* para que la página se adapte a las diferentes resoluciones posibles que pueda tener un usuario.	0	0	A partir de 800 píxeles, la interfaz se expone adaptándose a cualquier resolución de pantalla.

Ayuda y documentación

Aunque es mejor si el sistema puede utilizarse sin documentación, puede ser necesario disponer de ayuda y documentación. Esta debe ser fácil de buscar, centrada con las tareas del usuario, tener información de las etapas a realizar y que no sean muy extensas.

	Impacto	Frecuencia	Observaciones
1. En caso de existir ayuda, es visible y fácil de encontrar	5	5	Existe un manual de uso para la aplicación de gestión, pero no hay ayuda para la web.
2. La documentación de ayuda es sensible al contexto, se refiere a la sección donde se encuentra el usuario	5	5	
3. La documentación de ayuda sobre accesibilidad esta adaptada a las necesidades del usuario	5	5	
4. La ayuda esta orientada a los objetivos del usuario (generalmente a la resolución de problemas)	5	5	
5. La página dispone de un apartado de preguntas frecuentes	5	5	
6. La documentación de ayuda utiliza ejemplos	5	4	

Búsqueda

Este heurístico se aplica si existe un buscador en la interfaz (normalmente para interfaces web).

	Frecuencia	Impacto	Observaciones
1. En la página de inicio existe un cuadro de texto para introducir palabras a buscar en el sitio	0	0	
2. El cuadro de entrada de texto para buscar ocupa entre 25 y 30 caracteres visibles	3	3	Ocupa 35.
3. El área de búsqueda esta identificada con una cabecera que titula la opción de búsqueda	0	0	

Noticias

Este heurístico se aplica para aquellas interfaces que incluyen un apartado específico para las noticias.

	Impacto	Frecuencia	Observaciones
1. Los titulares de las noticias son breves y descriptivos	1	1	Salvo alguna excepción, son titulares cortos y claros.
2. Existen resúmenes específicos en las noticias (que no incluyen solamente las primeras líneas o el primer párrafo de la noticia)	3	3	Hay título y subtítulo, escritos dinámicamente desde la aplicación de gestión.
3. El vínculo principal a la noticia ampliada se encuentra en los titulares de las noticias	0	0	

Arquitectura de la información

Organización la información del sitio Web/sistema interactivo.

	Frecuencia	Impacto	Observaciones
1. El número de pestañas es adecuado	2	2	Algo excesivo en la página de inicio.
2. Los niveles de profundidad de información* existentes son suficientes	1	1	Muchos iconos en el inicio, pero a lo sumo hay dos niveles de profundidad. Tres si hay que descargar algún documento.

Varios

1. Se muestra la fecha de la última actualización	0	0
2. Las fechas se muestran en formato internacional	0	0

6.2.3.2 - Aplicación de gestión del Observatorio

Objetivos de la aplicación

Proporcionar una interfaz amigable y sencilla para la inserción, modificación o borrado de contenidos para cada una de las secciones de la página web del Observatorio.

Claridad de los Objetivos

La interfaz debe comunicar de manera inmediata su propósito, objetivo y funciones.

Subheurísticos	Impacto	Frecuencia	Observaciones
1. El propósito u objetivo que la interfaz nos transmite es claro y obvio. Con una simple ojeada podemos deducir cuál es su propósito y cuál es su finalidad	0	0	La interfaz es directa y clara desde la página de inicio.
2. En el caso de existir más de un objetivo, éstos están	0	0	

relacionados con el objetivo o función global y establecen coherencia entre ellos.			
3. En caso de existir más de un objetivo, éstos son claros y separados.	0	0	Posee las mismas secciones que el sitio web y su único objetivo es la gestión de esos contenidos.

Adecuación al mundo y a los objetos mentales del usuario / lógica de información.

El sitio está adaptado al mundo real de los usuarios, su lenguaje, conocimientos, etc.

<i>Lógica de la información</i>	Impacto	Frecuencia	Observaciones
1. La presentación de los contenidos es familiar o comprensible para el usuario	0	0	Apartados en la parte izquierda y campos a rellenar de arriba abajo.
2. Cuando existen opciones están ordenadas de manera lógica para la forma de pensar del usuario	0	0	
3. Las metáforas e iconos que utilizan son entendibles para el usuario y facilitan la interacción con la interfaz.	1	1	Texto, botones, y gestor de contenidos con iconos estándar.
<i>Formato de la información</i>			
4. La interfaz usa el lenguaje del usuario con palabras, frases y conceptos que le son familiares. El lenguaje debe ser claro, simple y con una sola idea por párrafo	0	0	Cada campo de texto tiene su título. Lenguaje extremadamente sencillo.
5. La información esta estructurada con títulos, negritas y viñetas	0	0	
6. El lenguaje y la disposición de la información es asequible y de lectura rápida para el usuario	0	0	
7. La estructura y presentación de la información no necesita explicaciones o información adicional para su comprensión	3	3	Algunos campos están presentados de forma poco amigable y aunque los campos son claros, tal vez no sobre alguna explicación.
8. Los textos y enunciados de los campos están redactados de forma afirmativa	0	0	
9. Se utiliza el lenguaje en forma directa, no impersonal	0	0	

Visibilidad del estado del sistema

El sistema debe tener siempre a los usuarios informados del estado del sistema, con una realimentación apropiada y en un tiempo razonable.

<i>Situación</i>	Impacto	Frecuencia	Observaciones
------------------	----------------	-------------------	----------------------

1. La interfaz (en el caso de tratarse de un sitio web) incluye de forma visible el título del sitio, de la sección o el título de la página	0	0	
2. Sabes en todo momento dónde estás posicionado	0	0	
<i>Acciones posibles</i>			
3. Los vínculos están claramente diferenciados	0	0	
4. No existe información o acciones que necesiten de una acción para su visualización	0	0	

Control y libertad para el usuario

Los usuarios eligen a veces funciones del sistema por error y necesitan a veces una salida de emergencia claramente marcada, eso es salir del estado indeseado sin tener que pasar por un diálogo extendido. Es importante disponer de deshacer y rehacer.

Subheurísticos	Impacto	Frecuencia	Observaciones
1. Se utilizan animaciones no controladas por el usuario	0	0	
2. El scroll no es más grande que dos pantallas	2	2	El apartado de Seminarios tiene un scroll sensiblemente más grande, debido a la cantidad de secciones a rellenar. Una disposición por pestañas podría ser una solución.
3. Es posible guardar información de la página Web/sistema interactivo	0	0	Precisamente la aplicación está preparada para guardar contenido.
4. Es posible imprimir la información de la página Web/sistema interactivo sin perder información	3	3	Se pierde la información de los campos de texto, pero ésta es visible e imprimible desde el sitio web, que es el objetivo principal.
5. Es posible regresar a la página principal desde cualquier sección	0	0	
6. Es posible aumentar y disminuir el tamaño de la letra	0	0	
7. La interfaz se visualiza perfectamente con diferentes resoluciones	1	1	Se expende hasta cualquier resolución a partir de 800 pixeles.
8. La interfaz no introduce tecnologías que requieren versiones actualizadas de navegadores o plugs-ins externos.	2	2	Tiene un apartado para embeber vídeos de Youtube, pero sólo son visibles desde el sitio web.

Consistencia

Consistencia	Impacto	Frecuencia	
1. Las etiquetas de los vínculos tienen los mismos nombres que los títulos de las páginas a las que se dirigen	0	0	

2. Las mismas acciones llevan a los mismos resultados	0	0	
3. Los mismos elementos son iguales en todo el sitio	0	0	
4. La misma información (texto) se expresa de la misma forma en toda la página	0	0	
5. La información esta organizada y es mostrada de manera similar en cada página	2	2	Está mostrada de manera similar y ordenada, pero tal vez podría presentarse de forma más amigable.
<i>Estándares</i>			
6. Se utilizan los colores estándares para los vínculos visitado y no visitados.	1	1	Se usan subrayados y cambio de forma de puntero principalmente.
7. Utiliza de manera diferente a la norma, convenciones o etiquetas universales	0	0	Interfaz muy sencilla y adaptada a la norma.
8. Las áreas de navegación superior, laterales, herramientas de búsqueda y controles (botones, radio buttons...) siguen los estándares comunes de mercado.	0	0	Se utilizan las áreas de navegación estándares de ASP.NET.

Prevenir errores

Es más importante prevenir la aparición de errores que generar buenos mensajes de error.

	Impacto	Frecuencia	Observaciones
1. El motor de búsqueda tolera errores tipográficos (mayúsculas), ortográficos (acentos) y acepta palabras similares	4	4	Es necesario escribir una palabra o parte de palabra en el orden exacto y sin errores.

Reconocimiento más que memoria

La página Web/sistema interactivo se basa en el reconocimiento más que en el recuerdo, que permite al usuario interactuar con el sitio de manera fácil y productiva.

	Impacto	Frecuencia	Observaciones
1. El uso de la interfaz no requiere recordar información de interfaces previas para interactuar con ella. Toda la información necesaria para la interacción se encuentra en la interfaz actual.	0	0	Al ser accesible todo desde el menú de la página de inicio, en un solo click se puede acceder.
2. Es fácil localizar información previamente encontrada.	0	0	Las entradas insertadas se pueden encontrar en una tabla paginada o a través del buscador.
3. La interfaz permite ver y	2	2	Es un gestor de contenidos, está

seleccionar, más que recordar y escribir			pensado para escribir.
4. La información esta organizada según una lógica reconocida y familiar para el usuario	0	0	Menú en la parte izquierda y contenido en el centro.
5. Se utilizan iconos relacionados con los contenidos a los que se asocian	2	2	Todos los enlaces son puramente texto.
6. La estructura, orden y lógica es familiar e intuitiva para los usuarios.	2	2	Estructura y orden lógicos e intuitivos. Presentación mejorable.

Diálogos estéticos y diseño minimalista

La página Web/sistema interactivo evita toda información o gráfico irrelevante y sólo incluye la información necesaria.

	Impacto	Frecuencia	Observaciones
1. La información visible es la única esencial para realizar la acción. La página no contiene información que es irrelevante o raramente necesaria	0	0	Todo el contenido está relacionado con el objetivo principal de la web.
2. No existe redundancia de información en la página	1	1	Hay redundancia de apartados, pero no de información.
3. La información es corta, concisa y precisa.	0	0	
4. Cada elemento de información se distingue del resto y no se confunde con otros	3	3	Algunos artículos podrían ir dentro de varios apartados, lo que puede demostrar que una reducción de secciones puede ser interesante.
5. El texto es fácil de hojear, esta bien organizado y las frases no son muy largas	0	0	
6. Las fuentes son legibles y tienen un tamaño adecuado	0	0	
7. Las fuentes utilizan colores con suficiente contraste con el fondo	0	0	Fuentes negras sobre fondo blanco.

Flexibilidad y eficiencia de uso

La interfaz facilita y optimiza el acceso a los usuarios independientemente de cuales sean sus características.

	Impacto	Frecuencia	Observaciones
1. Existen aceleradores de teclado para realizar operaciones frecuentes	5	5	Ninguno más allá de los facilitados por los navegadores
2. Si existen, quedan claros cuales son estos aceleradores	5	5	
3. Es posible repetir una acción ya realizada anteriormente de manera sencilla	0	0	

4. Se utiliza un diseño líquido* para que la página se adapte a las diferentes resoluciones posibles que pueda tener un usuario.	0	0	A partir de 800 píxeles, la interfaz se expone adaptándose a cualquier resolución de pantalla.
--	---	---	--

Ayuda y documentación

Aunque es mejor si el sistema puede utilizarse sin documentación, puede ser necesario disponer de ayuda y documentación. Esta debe ser fácil de buscar, centrada con las tareas del usuario, tener información de las etapas a realizar y que no sean muy extensas.

	Impacto	Frecuencia	Observaciones
1. En caso de existir ayuda, es visible y fácil de encontrar	5	5	Existe un manual de uso para la aplicación de gestión pero no se muestra en la web.
2. La documentación de ayuda es sensible al contexto, se refiere a la sección donde se encuentra el usuario	5	5	
3. La documentación de ayuda sobre accesibilidad esta adaptada a las necesidades del usuario	5	5	
4. La ayuda esta orientada a los objetivos del usuario (generalmente a la resolución de problemas)	5	5	
5. La página dispone de un apartado de preguntas frecuentes	5	5	
6. La documentación de ayuda utiliza ejemplos	5	5	

Búsqueda

Este heurístico se aplica si existe un buscador en la interfaz (normalmente para interfaces web).

	Frecuencia	Impacto	Observaciones
1. En la página de inicio existe un cuadro de texto para introducir palabras a buscar en el sitio	0	0	
2. El cuadro de entrada de texto para buscar ocupa entre 25 y 30 caracteres visibles	5	5	Ocupa 58
3. El área de búsqueda esta identificada con una cabecera que titula la opción de búsqueda	0	0	

Arquitectura de la información

Organización la información del sitio Web/sistema interactivo.

	Frecuencia	Impacto	Observaciones
1. El número de pestañas es adecuado	2	2	El que exige el sitio web.
2. Los niveles de profundidad de información* existentes son suficientes	0	0	No hay más de un nivel de profundidad.

Varios

1. Se muestra la fecha de la última actualización	0	0
2. Las fechas se muestran en formato internacional	0	0

6.2.4 - Test de usuarios.

Se ha confeccionado un test de usuarios para la página web del Observatorio y para su aplicación de gestión. Las tareas a realizar por parte del usuario son las representadas tanto en el apartado de diseño mediante los diagramas de secuencia como al comienzo de este apartado de pruebas y resultados. Se vuelven a enumerar a continuación.

Tareas en el Sitio Web del Observatorio.

Este test se ha realizado con 3 alumnos de la universidad CEU Cardenal Herrera, los cuales han realizado las pruebas bajo supervisión y cuyos resultados se pueden encontrar en el anexo.

- Descargar archivo adjunto al estudio “Libro blanco del marketing de afiliación 2010”.
- Encontrar el mismo estudio a través del buscador.
- Encontrar el mismo estudio a través del mapa del sitio.

Valoración

La primera tarea es la que más ha costado a los usuarios. El caso mejor se realizaba con 4 clics (Menú “Estudios” → Sección “Publicidad On-line, Redes Sociales y Web 2.0” → Estudio “Libro blanco del marketing de afiliación 2010” → Archivo “Guía Marketing de Afiliación”). Sin embargo, los usuarios han tenido que buscar la sección adecuada de entre más de 10, y ha habido un alumno que ha necesitado de 11 clics y más de 40 segundos.

La segunda tarea ha sido la más rápida. Los tres alumnos la han finalizado con 3 clics (poco más de 15 segundos) escribiendo las palabras clave en el buscador.

La tercera tarea también se ha solucionado con tres clics, pero el tiempo ha sido más variable, ya que la búsqueda a través del mapa del sitio no es tan cómoda como a través del buscador, pues hay mucho contenido.

Tareas en la aplicación de gestión.

Este test se ha realizado con 3 miembros del equipo del Observatorio Below, los cuales han realizado las pruebas bajo supervisión y cuyos resultados se pueden encontrar también en el anexo.

- Crear un nuevo usuario.
- Crear una conferencia y asignarle un seminario.
- Crear un estudio y adjuntarle un archivo.
- Borrar un experto.
- Editar la noticia “Las Técnicas "Below the Line" superan a la publicidad” para cambiarle el estado a “no visible”.

Valoración.

La primera tarea ha llevado a los usuarios alrededor de un minuto de tiempo y 3 clics. Uno para identificarse, otro para acceder al panel de gestión de usuarios y otro para guardar el usuario nuevo.

Para la segunda tarea el tiempo empleado por los usuarios ha sido similar, también inferior a un minuto, ya que el contenido lo han podido importar de un fichero ya escrito. Apenas 4 clics de media también. Uno para acceder a la sección de conferencias, otro para seleccionar el seminario enlazado desde el listado, otro para pegar el contenido en el editor de texto y otro para guardar.

El tiempo de media empleado para la tercera tarea ha sido algo mayor, debido al tiempo usado para seleccionar el archivo a adjuntar desde el PC. Por lo demás ha sido rápido y se ha solucionado con 4 clics de media y poco más de un minuto.

El tiempo empleado en borrar un experto ha sido inferior a 30 segundos de media. Sólo tres o cuatro clics. Acceder a la sección de expertos, buscar el experto en el listado (haciendo un clic de más si estaba en otra página), pinchar en “borrar” y finalmente en “confirmar”.

Finalmente, el tiempo empleado en buscar la noticia ha sido el más variable (de poco más de 30 segundos a más de 1 minuto y medio), ya que ha habido usuarios que buscado la noticia directamente en la tabla (la noticia estaba en una de las últimas páginas y han necesitado más de 5 clics para encontrarla), y otros que han hecho uso del buscador (con un único clic bastaba). Posteriormente, cambiar el estado a “no visible” y guardar ha sido rápido en todos los casos.

6.2.5 - Minería y posicionamiento web

Al escribir en el buscador de Google las palabras observatorio beyond, cuatro de los primeros cinco resultados pertenecen al Observatorio al que hace referencia esta web, aunque las fuentes son diversas.

El primer resultado sí que se ha conseguido gracias al código insertado en el sitio web. Se trata de la página principal y se muestra el contenido introducido en las etiquetas de metadatos “title” y “description” de la cabecera del fichero “index.aspx”.

Los siguientes resultados pertenecientes al sitio tienen como fuente el grupo de Facebook, enlazado también desde la web, un artículo referente al observatorio, publicado en el blog del

Colegio de Publicitarios y Relaciones Públicas de la Comunidad Valenciana y dos vídeos del Observatorio publicados en Youtube y embebidos en el sitio web.

Esto demuestra que usar Youtube para embeber los vídeos que se muestran en la web es una buena idea en términos de SEO, o al menos hablando de posicionamiento en Google.

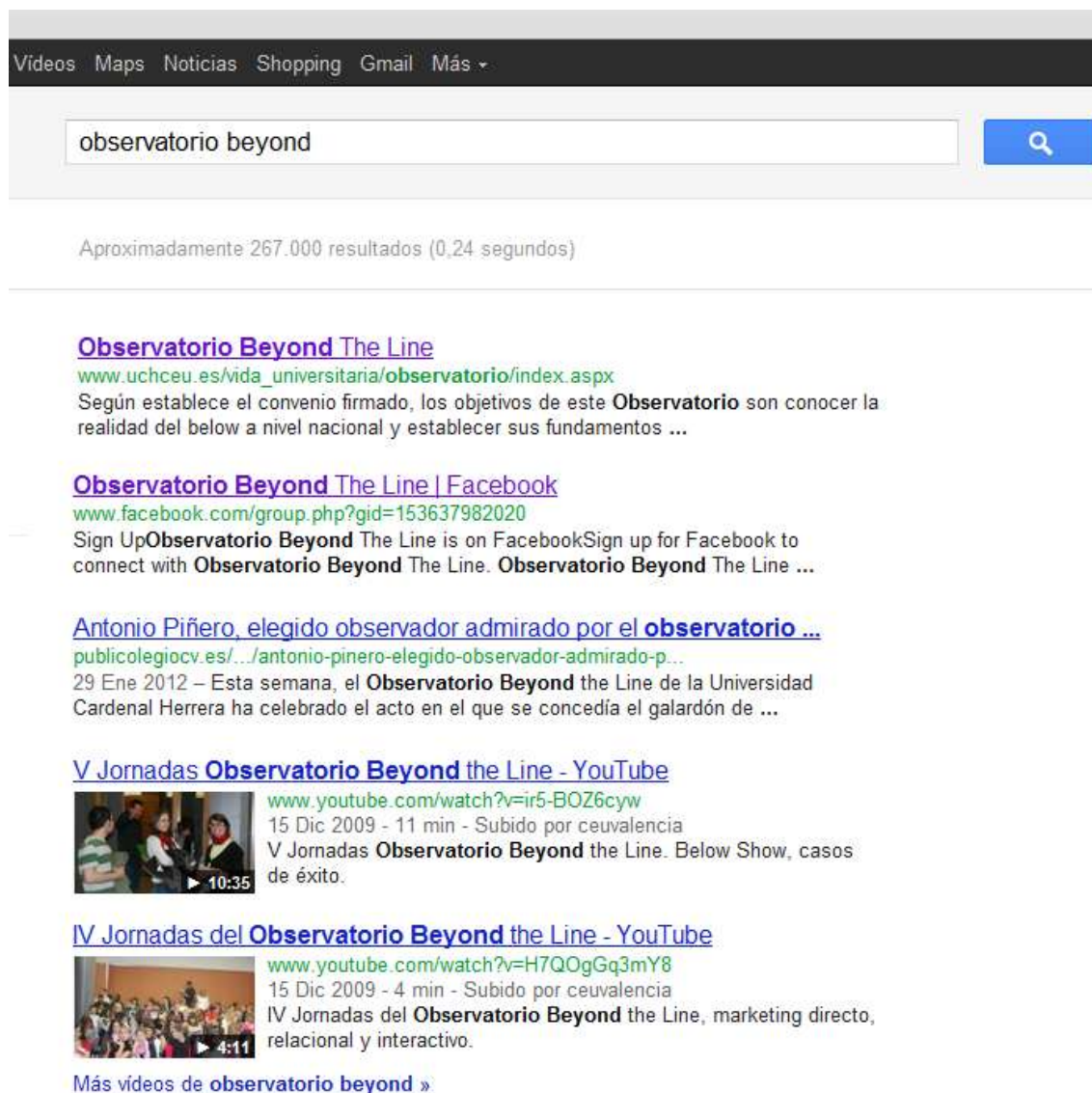


Figura 6.21. Resultados de la búsqueda “observatorio beyond” en el buscador de Google.

En cuanto a minería, cabe recordar que se introdujo el script de Google Analytics en la página inicial de sitio web para hacer un seguimiento de los accesos a la misma. Estos son algunos de los resultados obtenidos en el periodo que va desde el 20 de enero al 19 de febrero de 2012.

- 327 visitas.
- 162 visitantes únicos.
- 1127 páginas vistas.

- Los usuarios visitan 3,45 páginas por cada vez que acceden.
- 1:40 minutos de promedio por visita.
- 34,25 % son visitantes que han abandonado el sitio web habiendo navegado únicamente por la página principal.
- El 36,39 % de los usuarios que acceden lo hacen por primera vez.

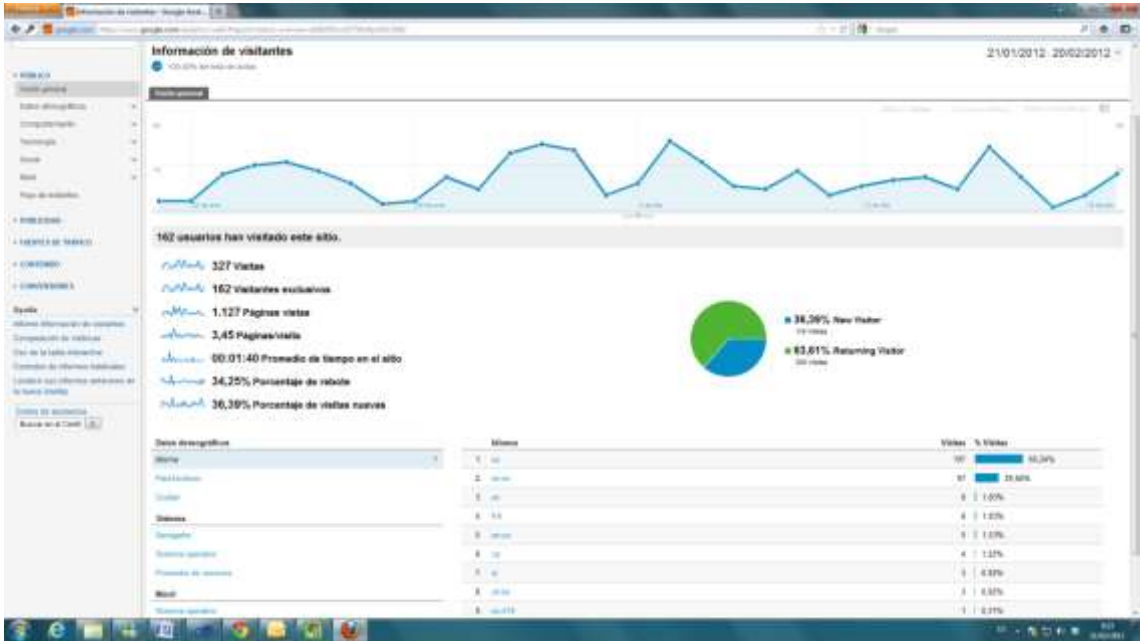


Figura 6.22. Visión general de estadísticas del Observatorio en Google Analytics.

Atendiendo al contenido, se puede observar que el apartado más visto, con diferencia, es la página inicial, seguido de la última noticia publicada. El apartado de noticias, la noticia anterior, el apartado de vídeos, el de experiencias creativas y el de El Observatorio estuvo en... se llevan un porcentaje residual de las visitas.

Esto puede ser debido a la falta de interés del resto de apartados, lo que apoyaría la reflexión acerca de un exceso de secciones, o una falta de actualización de contenidos, algo que se confirma al observar que apartados como Estudios, Expertos o Memorias apenas han sido actualizados durante el año 2012.

Página		Páginas vistas	% páginas vistas
1.	/observatorio/index.aspx	324	28,75%

2.	http://www.uchceu.es/vida_universitaria/observatorio/Noticias.aspx?id=64	238	21,12%
3.	/observatorio	112	9,94%
4.	http://www.uchceu.es/vida_universitaria/observatorio/estuvo.aspx?id=36	87	7,72%
5.	http://www.uchceu.es/vida_universitaria/observatorio/index.aspx?id=30	50	4,44%
6.	/observatorio/	44	3,90%
7.	http://www.uchceu.es/vida_universitaria/observatorio/NuestrosVideos.aspx	34	3,02%
8.	http://www.uchceu.es/vida_universitaria/observatorio/Noticias.aspx	34	3,02%
9.	http://www.uchceu.es/vida_universitaria/observatorio/experiencias.aspx	30	2,66%
10.	http://www.uchceu.es/vida_universitaria/observatorio/Noticias.aspx?id=62	25	2,22%

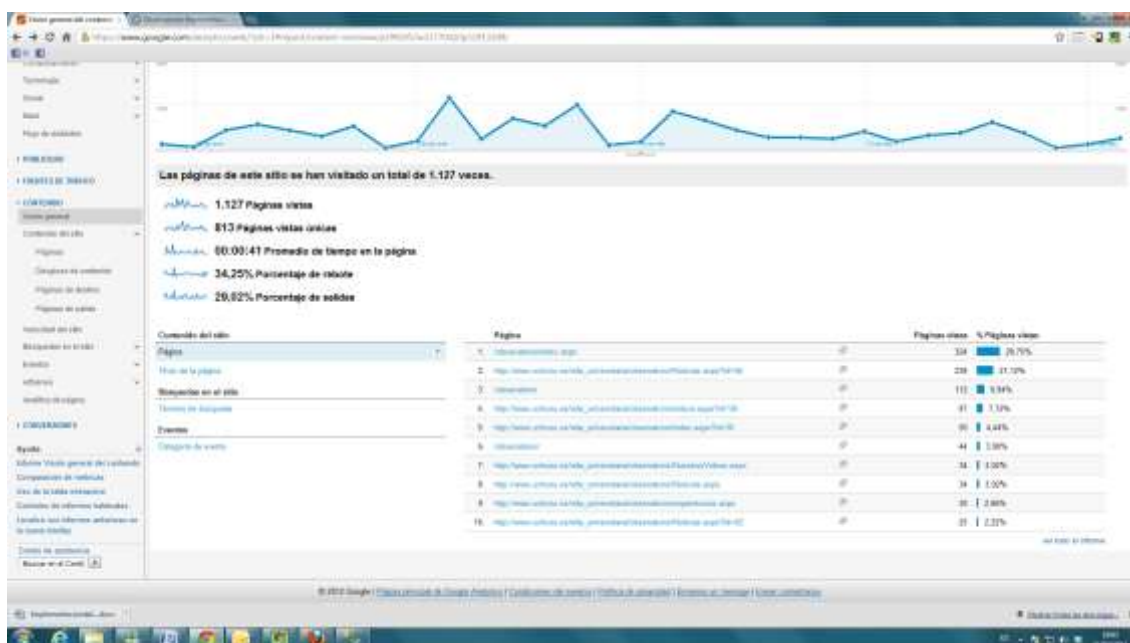


Figura 6.23. Visión general del contenido

6.2.6 – Pruebas de carga y estrés

Como ya se ha explicado a lo largo de esta memoria, el servidor en el cual se ha alojado el sitio y la aplicación web del Observatorio Beyond The Line es el mismo en el cual se aloja el sitio web corporativo de la Universidad CEU Cardenal Herrera y todas las aplicaciones web relacionadas con esta entidad. Por esa razón el servidor debe estar preparado para soportar una carga muy superior de la que realmente necesita la aplicación que se ha desarrollado en este proyecto, ya que la universidad maneja una gran cantidad de información que puede ser consultada tanto por personal y estudiantes como por cualquier interesado en los estudios o servicios ofrecidos por la misma. En total se puede estimar una media de 20.000 visitas diarias al conjunto de aplicaciones y sitios web alojados en el servidor.

En cualquier caso, Visual Studio 2010 incorpora una herramienta de Microsoft llamada Application Center Test (ACT) que está diseñada para realizar pruebas de carga y stress, que permite obtener información para detectar problemas de rendimiento y escalabilidad.

Las especificaciones del servidor, apuntadas en el apartado Especificaciones de esta memoria son los siguientes:

Intel® Xeon® CPU E554 @2.83 GHz 2.83 GHz, con 4 GB de RAM. Sistema operativo Microsoft Windows Server 2003 e Internet Information Service 6.

Las pruebas realizadas con esta herramienta se han realizado sobre una máquina gemela que contiene una copia exacta de las bases de datos y de las aplicaciones web del servidor original, (se hace para no saturar el servidor principal), y para ello se ha necesitado de más de una máquina cliente para conseguir mantener el procesador del servidor por encima del 80 % de uso.

En cualquier caso se va a realizar las pruebas de estrés con el sitio web del Observatorio, que es de esperar que reciba más visitantes que la aplicación privada de gestión. Se van a presuponer 5 peticiones por cada visita para comprobar su respuesta, y para ello se van a registrar los tiempos de respuesta y algunos datos estadísticos.

Las peticiones elegidas son:

- Visita a página principal, que contiene una consulta a las últimas 5 noticias de la base de datos, además del contenido de la página principal, también obtenido dinámicamente.
- Visita a la sección expertos en marketing promocional.
- Ver información sobre un experto.
- Ver RSS, que realiza una consulta a las últimas 25 noticias de la base de datos.
- Ver mapa completo del sitio, que requiere realizar consultas para mostrar todas las secciones y sus elementos en un único índice.

Para encontrar el número óptimo y crítico de usuarios se han ido realizando varias pruebas incrementales, comprobando así el rendimiento del sistema. El caso óptimo resultó estar en 325 usuarios y el caso crítico en 350. La diferencia parece pequeña, pero para 350 usuarios se

empiezan a reportar errores que ya afectan a algunos de los usuarios que están utilizando el sistema.

El informe contiene varias filas y columnas. Las filas representan cada una de las 5 peticiones que se realizaron y las columnas representan algunas medidas dentro de la prueba:

- URL: La actividad que se realiza, es decir, la petición.
- # Muestras: La cantidad de veces que se realiza la petición (una vez por usuario).
- Media: Media aritmética del tiempo en milisegundos.
- Mediana: Mediana del tiempo en milisegundos.
- Min: Tiempo mínimo de todas las peticiones de ese tipo.
- Max: Tiempo máximo de todas las peticiones de ese tipo.
- Porcentaje de error: Porcentaje de las peticiones fallidas.
- Rendimiento: Rendimiento medido en peticiones por segundo.
- Kb/Sec: Media de la velocidad en kilobytes por segundo.

URL	#Muestras	Media	Mediana	Min	Max	% Error	Rendimiento	Kb/Sec
Principal	350	9	15	0	47	0,00%	19,6/sec	47,53
Sección Expertos	350	514	16	0	5688	0,00%	18,4/sec	47,59
Experto	350	8	0	0	47	0,00%	20,9/sec	46,96
RSS	350	2231	1141	0	8422	0,00%	21,6/sec	105,93
Mapa de Sitio	350	13969	13735	312	27500	9,71%	11,2/sec	148,80
Total	1750	2788	16	0	27500	1,62%	13,3/sec	65,48

En la tabla se aprecia que en la actividad de visita al mapa del sitio existe un porcentaje de error del 9,71 %, éste representa 33 usuarios de los 350 con los que se realizó la prueba. Estos 33 usuarios obtendrían una página de error al intentar conectarse al sistema. Ésta es una cantidad importante de usuarios por lo que se tiene que considerar como caso crítico.

Por otra parte, la media total es de 2788 ms, es decir, que en promedio el sistema tarda en responder 2,7 segundos, un tiempo bastante bueno para la web de Observatorio si se tiene en cuenta que las visitas a la web del Observatorio no se acercan ni por asomo a este número de pruebas.

Para el caso de la web principal es distinto, ya que sí que se puede alcanzar y superar el número de peticiones. Estas peticiones a la página principal llevan consigo una conexión a la base de datos para mostrar contenido dinámico. Para solucionarlo se aplica un tiempo de caché de 10 minutos, tanto en el fichero de configuración de ASP.NET como en las herramientas administrativas del IIS.

Para el caso óptimo se han utilizado 325 usuarios, y como se muestra en la tabla posterior no hay porcentaje de error y que todas las peticiones se responden de manera adecuada y correcta. En cuanto a la media, la diferencia es apenas de 0,3 segundos. En cualquier caso se trata de un buen tiempo de respuesta, teniendo en cuenta que hay conexiones a bases de datos.

URL	#Muestras	Media	Mediana	Min	Max	% Error	Rendimiento	Kb/Sec
Principal	325	9	0	0	63	0,00%	16,9sec	44,21
Sección Expertos	325	97	16	0	2563	0,00%	19,0/sec	58,83
Experto	325	8	0	0	46	0,00%	18,3/sec	40,17
RSS	325	1839	1063	0	6140	0,00%	16,2/sec	62,75
Mapa de Sitio	325	12913	12640	93	24906	0,00%	10,9/sec	159,10
Total	1625	2788	16	0	27500	0,00%	13,3/sec	65,48

6.3 - Evaluación presupuestaria

El proyecto ha sufrido una sensible variación respecto a la planificación temporal debido a un factor fundamental expuesto como riesgo potencial en el apartado de planificación. El becario asignado al proyecto terminó en junio junto al curso académico, siendo contratado en octubre como programador del Servicio de Comunicación Digital y aumentando la jornada laboral a 8 horas, pero abandonando la dedicación al proyecto a tiempo completo y repartiendo dicho tiempo entre todas las tareas asignadas a este departamento de la universidad. Esto ha implicado un retraso en la realización de 2 meses respecto de la planificación inicial.

En cuanto a los costes, hay que reconocer el ahorro respecto a lo dispuesto en dicha planificación, en la que se recomendó utilizar a 3 programadores a tiempo completo y con un sueldo en torno a los 1700 euros mensuales cada uno durante 9 meses, y habiendo utilizado finalmente un becario durante 6 meses con un sueldo de 400 euros al mes y un programador junior durante 5 meses con un sueldo de 1000 euros al mes. Esto ha supuesto un gasto de a los 7400 euros frente a los 45900 euros estimados mediante el método COCOMO, lo que implica un ahorro de 38500 euros en sueldos. Por otra parte, el coste de recursos de hardware amortizado durante 9 meses ascendía a 500 euros, por lo que en los 11 meses que finalmente ha durado el desarrollo se ha incrementado 111 euros. Esto supone un ahorro final de 37389 euros sobre la estimación presupuestaria, aunque también una demora de 2 meses.

7 – Conclusiones y Trabajo Futuro

7.1 - Conclusiones

El uso del Framework .NET para el desarrollo de aplicaciones web y de gestión no puede considerarse una mala idea. Aporta herramientas de desarrollo interesantes a nivel de presentación de contenidos y de funcionalidad, así como soluciones de fáciles de implementar frente a las que pueden aportar otros lenguajes de desarrollo nativos, inclusive los orientados a aplicaciones de ámbito empresarial.

El conocimiento de la plataforma y el hecho de disponer de un sistema preparado para su uso han decantado la balanza a su favor también frente a otros frameworks como Struts o Spring (basados en Java) y Code Igniter o Symfony (basados en PHP) que iban a exigir más tiempo de aprendizaje y que iba a suponer tener que poner en marcha otros recursos tecnológicos sin que ello proporcionara grandes ventajas a nivel de presentación, funcionalidad, potencia o capacidad de integración con otras tecnologías.

Sin embargo, sí que es discutible la conveniencia de aprovechar los Sistemas de Gestión de Contenidos ya existentes en el mercado para el desarrollo de aplicaciones de esta índole. La presumible personalización de la aplicación y el sitio web en función de las especificaciones del cliente, que en un principio ha servido para intentar justificar el uso de un Framework que diera mayor libertad y posibilidades de desarrollo, finalmente no dio tan buen resultado como se esperaba.

Los requisitos del cliente y los continuos cambios en sus peticiones a medida que avanzaba el desarrollo del proyecto han ido complicando su resolución frente al coste que hubiera supuesto afrontarlo con las herramientas ofrecidas por un CMS. Además, los resultados obtenidos no han supuesto una diferenciación lo suficientemente interesante como para compensar ese esfuerzo.

No obstante, el escenario y las condiciones que se daban, que hacían inviable el uso de otras plataformas por la incompatibilidad con la infraestructura que la Universidad CEU Cardenal Herrera pone a disposición de los programadores, han servido para probar las posibilidades de la plataforma en un ámbito de aplicación que se va a extender en un futuro próximo y conocer las limitaciones de estos sistemas para el desarrollo de este tipo de sitios y aplicaciones.

Así pues, habrá que seguir investigando para encontrar soluciones a este problema de compatibilidad con las nuevas plataformas de gestión de contenidos que faciliten el desarrollo de estas aplicaciones hasta que se modernice la infraestructura de la universidad, algo que

puede demorarse teniendo en cuenta que no vive aislada de la situación económica en la que está sumida el país y el coste que implica la actualización de las licencias de los sistemas utilizados.

En cualquier caso, o al menos hasta que se cumpla uno de los dos supuestos mencionados en el párrafo anterior, ASP.NET es una solución fiable y potente para ésta y cualquier otra aplicación web, que ha ido aumentando su capacidad de integración con otras tecnologías con el paso del tiempo, como Ajax o JSON, y que ofrece una libertad de desarrollo que jamás será alcanzada por la que pueden otorgar las plantillas prediseñadas que ponen a disposición los sistemas de gestión de contenidos que hay en el mercado.

7.2 - Trabajo Futuro

El trabajo futuro puede debe ir orientado en la búsqueda de sistemas de gestión de contenidos compatibles con la infraestructura de la universidad e integrables en el sitio web corporativo. Las dos principales opciones a tener en cuenta son las siguientes:

- Esperar al lanzamiento de la última versión del CMS Umbraco, basado en .NET y testarlo para comprobar que ha superado todos los bugs y limitaciones que arrastraban las versiones anteriores como se vio en el estado del arte.
- Buscar alguna utilidad gratuita que facilite la integración de Wordpress (el CMS que mejor impresión ha dejado durante el periodo de investigación) en el sistema operativo Windows Server 2003 y el servidor IIS 6. Esta opción es la que va a centrar la atención de este apartado.

En el Estado del Arte se ha hablado de la existencia de módulos desarrollados por Microsoft que se comportan de manera análoga a como lo hacen los ficheros "httpd.conf" y ".htaccess" en Apache, pero exclusivos para versiones antiguas de IIS (IIS 7 es capaz de leer estos ficheros sin necesidad de traductores). El carácter de pago de estos módulos impidió valorar la integración de Wordpress en el sistema de la universidad para desarrollar aplicaciones de gestión de contenidos integrados en el sitio web.

Sin embargo, durante el curso en el que se ha ido desarrollando el proyecto ha aparecido una aplicación gratuita equivalente al ISAPI Rewrite de Microsoft. Se llama URL Rewrite y está pensado para ser utilizado con IIS 6. Si este módulo funcionara sería posible instalar Wordpress en el sistema y configurar una red multisitio de web/blogs/aplicaciones para los distintos servicios y facultades emergentes en la universidad.

En el caso concreto de este proyecto, una posible solución para homogeneizar todos los sitios web de la universidad ya creados con los nuevos e integrarlos fácilmente en el diseño de la web sin perder información, es aprovechar el conocimiento de la tecnología RSS y su facilidad de uso. Wordpress tiene un plugin que permite la importación de contenidos a partir de un fichero RSS. Mediante las clases RSS.cs y RSSBLL.cs ya implementadas en este proyecto, se puede almacenar y categorizar todas las entradas del sitio web del Observatorio en un fichero XML-RSS que se importará a través de Wordpress sin perder un ápice de información.

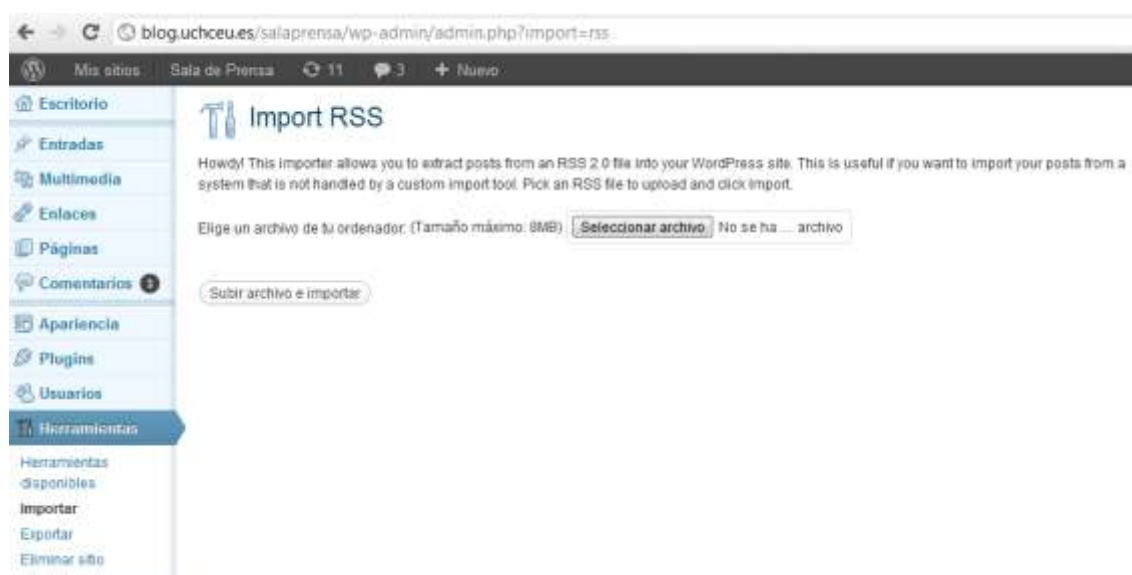


Figura 7.1. Importar RSS a blog en Wordpress.

A partir de ahí, las posibilidades de maquetación y estructuración del nuevo sitio web son tantas como permita el CMS y sus módulos, plugins y plantillas disponibles, cuya implantación puede realizarse de forma sencilla y a través de los cuales se garantiza una alta usabilidad y accesibilidad. Eso sí, aunque las posibilidades de modificar el desarrollo mediante PHP no son desdeñables, no se puede aspirar a conseguir la potencia y la libertad de desarrollo que ofrece ASP.NET y las tecnologías complementarias que ofrece el marco de trabajo .NET para el desarrollo de aplicaciones más específicas y/o complejas.

Por esta última razón no se puede desaconsejar el uso de ASP.NET para ésta o cualquier otra aplicación web ni dejar de investigar y probar las nuevas actualizaciones del Framework y las nuevas librerías y el soporte que éstas ofrecen a las nuevas tecnologías emergentes. La más interesante, sin lugar a dudas, será las posibilidades de integración de la nueva versión del Framework NET 4.5 con HTML 5.

8 – Bibliografía

8.1 – Bibliografía impresa

1. Programador de Aplicaciones en Redes Internet/Extranet/Intranet. ISIConsulting.
2. Visual Studio .NET (Anaya Multimedia). Julian Templeman, David Vitter.
3. La biblia de C# (Anaya Multimedia). Ferguson, Patterson, Boutquin y Gupta.
4. Visual Studio 2010, .NET 4.0 y ALM (Editorial Kasis Press). Bruno Capuano.
5. MS SQL Server 2005 Manual de Administrador (Editorial McGraw-Hill). Stanek, William.

8.2 – Enlaces Web.

1. <http://www.purosoftware.com/desarrollo-web-scripts-blog/09-umbraco-cms-para-asp-net.html>
2. <http://umbraco.com/>
3. <http://es.wikipedia.org>
4. <http://es.wordpress.org/>
5. <http://img.redusers.com/imagenes/libros/lpcu096/capitulogratias.pdf>
6. <http://materias.fi.uba.ar/7500/blanco-tesisingenieriainformatica.pdf>
7. http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/tTema1.pdf
8. <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/378/1/36610tfc.pdf>
9. <http://books.openlibra.com/pdf/Analisis-desarrollo-patrones-J2EE.pdf>
10. <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>
11. <http://dotnetnuke.com.es/cms-en-asp-net/>
12. <http://www.liferay.com/>
13. <http://drupal.org/>
14. <http://sentidoweb.com/2007/07/24/comparativa-entre-j2ee-asp-net-y-php.php>
15. <http://www.php.net/manual/es/history.php.php>
16. <http://www.danysoft.com/free/aspnet.pdf>
17. <http://www.ecured.cu>
18. <http://www.metadatos-xmlrdf.com>
19. <http://www.bib.uc3m.es/~mendez/publicaciones/7jc99/rdf.htm#3>
20. <http://www.isapirewrite.com/docs/>
21. <http://www.blogtecnologico.net/url-rewrite-para-iis-gratis/>
22. <http://dotcms.com/>

9 - Anexos

9.1 - Web a analizar: Observatorio Beyond The Line

Echa un vistazo a las páginas que van apareciendo para intentar quedarte con la mayor cantidad de detalles posibles y luego trata de realizar las tareas.

- **1ª Tarea.** Descargar archivo adjunto al estudio “Libro blanco del marketing de afiliación 2010”.
- **2ª Tarea.** Encontrar el mismo estudio a través del buscador.
- **3ª Tarea.** Encontrar el mismo estudio a través del mapa del sitio.

9.1.1 - Test de usuario 1.

Variable a medir	Resultado
Tiempo empleado en realizar...	
Tarea 1	23 segundos
Tarea 2	17 segundos
Tarea 3	50 segundos
Número de clics empleado en realizar...	
Tarea 1	4
Tarea 2	3
Tarea 3	3
Errores en la realización de...	
Tarea 1	0
Tarea 2	0
Tarea 3	0
Memoria de reconocimiento.	
El enlace para acceder al grupo de Facebook del Observatorio se puede encontrar en: a) Página principal, parte superior. b) Página principal, parte inferior c) No se encuentra en la página principal. d) No lo recuerdo.	A
Las últimas noticias se pueden encontrar en: a) Parte derecha. b) Parte izquierda. c) Parte superior. d) Parte inferior.	A (incorrecta)

e) No los he visto.	
En la parte izquierda de la página de inicio he podido reconocer (válida más de una opción) a) Enlaces y directorios. b) Noticias. c) Memorias. d) Buscador	A y B. (Falta C)
La sección “El Observatorio estuvo en...” se encuentra en: a) Parte superior derecha. b) Parte superior izquierda. c) Parte inferior derecha. d) Parte inferior izquierda.	b)
Memoria de recuerdo.	
Enumera las opciones que recuerdes de las proporcionadas en el menú superior de la página principal.	Nosotros, Expertos, Estudios, El Observatorio estuvo en... (Falta seminarios, jornadas y congresos, Experiencias creativas y Diccionario Below).
Enumera Las opciones que recuerdes del menú de la parte superior izquierda de la página principal	Observador Admirado, Contacta con nosotros, Enlaces y directorios. (Falta Objetivos, Memorias, Equipo, Premios y Partners. Sobra Enlaces y Directorios).
Enumera las categorías en las que se agrupan los videos disponibles en la web.	No recuerda ninguna. (Nuestros Vídeos y Observamos el Sector).
¿Dónde se encuentra el icono del RSS?	No lo recuerda
Tiempo empleado en recordar la estructura de...	
Página principal.	2 minutos 6 segundos (incompleta)
Página de Sección de estudios.	1 minuto 23 segundos (incompleta)
Página del Estudio.	41 segundos (incompleta)
Satisfacción con la página web. Puntúa de 0 a 10.	
Calidad.	7
Facilidad de uso.	6
Agrado.	6

Valoración: El usuario 1 ha sido muy rápido en realizar las tareas encomendadas, pero no ha dedicado tiempo ni prestado atención para memorizar tanto los detalles como la estructura de la web. La tarea 3 ha resultado ser la más engorrosa ya que la cantidad de información obtenida del mapa del sitio le ha provocado confusión.

9.1.2 - Test de usuario 2.

Variable a medir	Resultado
Tiempo empleado en realizar...	
Tarea 1	1 minuto y 3 segundos
Tarea 2	16 segundos
Tarea 3	30 segundos
Número de clics empleado en realizar...	
Tarea 1	11
Tarea 2	3
Tarea 3	3
Errores en la realización de...	
Tarea 1	4
Tarea 2	0
Tarea 3	0
Memoria de reconocimiento.	
El enlace para acceder al grupo de Facebook del Observatorio se puede encontrar en: a) Página principal, parte superior. b) Página principal, parte inferior c) No se encuentra en la página principal. d) No lo recuerdo.	A
Las últimas noticias se pueden encontrar en: a) Parte derecha. b) Parte izquierda. c) Parte superior. d) Parte inferior. e) No los he visto.	B
En la parte izquierda de la página de inicio he podido reconocer (válida más de una opción) a) Enlaces y directorios. b) Noticias. c) Memorias. d) Buscador	A, B y C
La sección "El Observatorio estuvo en..." se encuentra en: a) Parte superior derecha. b) Parte superior izquierda. c) Parte inferior derecha. d) Parte inferior izquierda.	B
Memoria de recuerdo.	

Enumera las opciones que recuerdes de las proporcionadas en el menú superior de la página principal.	Nosotros, Expertos, Estudios, El Observatorio estuvo en... , Seminarios, Jornadas y Congresos, Experiencias creativas. (Falta Diccionario Below).
Enumera Las opciones que recuerdes del menú de la parte superior izquierda de la página principal	Observador Admirado, Contacta con nosotros, Objetivos, Memorias, Equipo y Premios, (Falta Partners).
Enumera las categorías en las que se agrupan los videos disponibles en la web.	Nuestros Vídeos y Observamos el Sector.
¿Dónde se encuentra el icono del RSS?	Parte inferior izquierda.
Tiempo empleado en recordar la estructura de...	
Página principal.	3 minutos 24 segundos
Página de Sección de estudios.	1 minuto 52 segundos (incompleta)
Página del Estudio.	1 minuto 2 segundos
Satisfacción con la página web. Puntúa de 0 a 10.	
Calidad.	8
Facilidad de uso.	5
Agrado.	7

Valoración: El usuario 3

9.1.3 - Test de usuario 3.

Variable a medir	Resultado
Tiempo empleado en realizar...	
Tarea 1	33 segundos
Tarea 2	18 segundos
Tarea 3	42 segundos
Número de clics empleado en realizar...	
Tarea 1	4
Tarea 2	3
Tarea 3	3
Errores en la realización de...	
Tarea 1	0
Tarea 2	0
Tarea 3	0
Memoria de reconocimiento.	
El enlace para acceder al grupo de Facebook del Observatorio se puede encontrar en: a) Página principal, parte superior. b) Página principal, parte inferior c) No se encuentra en la página principal. d) No lo recuerdo.	A
Las últimas noticias se pueden encontrar en: a) Parte derecha. b) Parte izquierda. c) Parte superior. d) Parte inferior. e) No los he visto.	B
En la parte izquierda de la página de inicio he podido reconocer (válida más de una opción) a) Enlaces y directorios. b) Noticias. c) Memorias. d) Buscador	A, B y C
La sección "El Observatorio estuvo en..." se encuentra en: a) Parte superior derecha. b) Parte superior izquierda. c) Parte inferior derecha. d) Parte inferior izquierda.	B
Memoria de recuerdo.	
Enumera las opciones que recuerdes de las proporcionadas en el menú superior de la página	Nosotros, Estudios, El Observatorio estuvo en... , Seminarios, Jornadas y Congresos,

principal.	Experiencias creativas y Diccionario Below. (Falta Expertos).
Enumera Las opciones que recuerdes del menú de la parte superior izquierda de la página principal	Observador Admirado, Contacta con nosotros, Objetivos y Memorias. (Falta Premios, Memorias y Partners).
Enumera las categorías en las que se agrupan los videos disponibles en la web.	Nuestros Vídeos y Observamos el Sector.
¿Dónde se encuentra el icono del RSS?	No lo recuerda
Tiempo empleado en recordar la estructura de...	
Página principal.	2 minutos 53 segundos (incompleta)
Página de Sección de estudios.	1 minuto 34 segundos (incompleta)
Página del Estudio.	58 segundos
Satisfacción con la página web. Puntúa de 0 a 10.	
Calidad.	7
Facilidad de uso.	6
Agrado.	7

Valoración: El usuario 3 ha realizado todas las actividades en tiempos adecuados y ha recordado la estructura de la web de forma satisfactoria.

9.2 - Web a analizar: Aplicación Observatorio

Echa un vistazo a las páginas que van apareciendo para intentar quedarte con la mayor cantidad de detalles posibles y luego trata de realizar las tareas.

- **1ª Tarea.** Crear una conferencia y asignarle un seminario.
- **2ª Tarea.** Crear un estudio y adjuntarle un archivo.
- **3ª Tarea.** Borrar un experto.
- **4ª Tarea.** Editar la noticia “Las Técnicas "Below the Line" superan a la publicidad” para cambiarle el estado a “no visible”.

9.1.1 - Test de usuario 1.

Variable a medir	Resultado
Tiempo empleado en realizar...	
Tarea 1	53 segundos
Tarea 2	21 segundos
Tarea 3	34 segundos
Tarea 4	45 segundos
Número de clics empleado en realizar...	
Tarea 1	3
Tarea 2	3
Tarea 3	3
Tarea 4	3
Errores en la realización de...	
Tarea 1	0
Tarea 2	0
Tarea 3	0
Tarea 4	0
Memoria de reconocimiento.	
El buscador se encuentra en: a) Parte inferior, encima del listado de entradas antiguas. b) Parte superior, encima del título. c) Parte superior, debajo del título. d) No he visto ningún buscador	A
Cada experto tiene obligatoriamente: a) Nombre, sección, y dos imágenes b) Nombre, sección, imagen y archivo.	A

c) Nombre, sección, descripción e imagen. e) No los recuerdo.	
Enlazar un seminario a una conferencia es obligatorio a) Sí b) No. c) No lo recuerdo.	A
En la sección de Estudios se permite insertar imágenes y archivos. a) Sí. b) No. c) Solo imágenes. d) No lo recuerdo.	B) incorrecto
Memoria de recuerdo.	
Enumera las opciones que recuerdes de las proporcionadas en el panel de Noticias	Título, subtítulo, descripción, calendario, buscador, visible, listado de noticias (falta sección, archivo e imagen).
Enumera Las opciones que recuerdes en el panel de Conferencias.	Título, seminario, descripción, archivo, imagen y listado de conferencias (falta buscador y visible).
Enumera Las opciones que recuerdes en el panel de Estudios.	Título, subtítulo, archivo, descripción (falta listado de estudios, imagen y buscador).
Enumera Las opciones que recuerdes en el panel de Expertos.	Título, subtítulo, sección, imagen, descripción, listado para editar y borrar (falta imagen pequeña y archivo)
Tiempo empleado en recordar la estructura de...	
Página de Conferencias.	1 minuto 5 segundos
Página de Estudios.	1 minuto 3 segundos (incompleta)
Página de Noticias.	58 segundos (incompleta)
Satisfacción con la página web. Puntúa de 0 a 10.	
Calidad.	7
Facilidad de uso.	8
Agrado.	6

Valoración: El usuario 1 ha realizado las actividades de forma bastante eficiente y sin apenas dificultades. Posteriormente ha tenido varios errores a la hora de recordar las opciones disponibles en los apartados sobre los que ha realizado la actividad. Tal vez la actividad propuesta no ha requerido la memorización de todas las posibilidades.

9.1.2 - Test de usuario 2.

Variable a medir	Resultado
Tiempo empleado en realizar...	
Tarea 1	1 minuto 4 segundos
Tarea 2	42 segundos
Tarea 3	22 segundos
Tarea 4	1 minuto 13 segundos
Número de clics empleado en realizar...	
Tarea 1	5
Tarea 2	3
Tarea 3	3
Tarea 4	6
Errores en la realización de...	
Tarea 1	2
Tarea 2	0
Tarea 3	0
Tarea 4	3
Memoria de reconocimiento.	
El buscador se encuentra en: a) Parte inferior, encima del listado de entradas antiguas. b) Parte superior, encima del título. c) Parte superior, debajo del título. d) No he visto ningún buscador	A
Cada experto tiene obligatoriamente: a) Nombre, sección, y dos imágenes b) Nombre, sección, imagen y archivo. c) Nombre, sección, descripción e imagen. e) No los recuerdo.	A
Enlazar un seminario a una conferencia es obligatorio a) Sí b) No. c) No lo recuerdo.	A
En la sección de Estudios se permite insertar imágenes y archivos. a) Sí. b) No. c) Solo imágenes. d) No lo recuerdo.	B) Incorrecto
Memoria de recuerdo.	

Enumera las opciones que recuerdes de las proporcionadas en el panel de Noticias	Título, subtítulo, descripción, calendario, visible, listado de noticias, sección, archivo e imagen. (Falta buscador).
Enumera Las opciones que recuerdes en el panel de Conferencias.	Título, seminario, descripción, buscador, visible, listado de conferencias (falta imagen y archivo).
Enumera Las opciones que recuerdes en el panel de Estudios.	Título, subtítulo, archivo, descripción (falta listado de estudios, imagen y buscador).
Enumera Las opciones que recuerdes en el panel de Expertos.	Título, subtítulo, sección, imagen, descripción, listado para editar y borrar (falta imagen pequeña y archivo)
Tiempo empleado en recordar la estructura de...	
Página de Conferencias.	1 minuto 10 segundos (incompleta)
Página de Estudios.	58 segundos (incompleta)
Página de Noticias.	1 minuto y 6 segundos (incompleta)
Satisfacción con la página web. Puntúa de 0 a 10.	
Calidad.	6
Facilidad de uso.	6
Agrado.	6

Valoración: El usuario 2 ha tardado más tiempo y ha utilizado rutas poco eficientes para completar las actividades. Tampoco ha logrado memorizar ni situar visualmente las opciones de cada sección. Probablemente la aplicación no esté adaptada a la usabilidad que se requiere.

9.1.3 - Test de usuario 3.

Variable a medir	Resultado
Tiempo empleado en realizar...	
Tarea 1	45 segundos
Tarea 2	32 segundos
Tarea 3	27 segundos
Tarea 4	38 segundos
Número de clics empleado en realizar...	
Tarea 1	3
Tarea 2	3
Tarea 3	3
Tarea 4	3
Errores en la realización de...	
Tarea 1	0
Tarea 2	0
Tarea 3	0
Tarea 4	0
Memoria de reconocimiento.	
El buscador se encuentra en: a) Parte inferior, encima del listado de entradas antiguas. b) Parte superior, encima del título. c) Parte superior, debajo del título. d) No he visto ningún buscador	A
Cada experto tiene obligatoriamente: a) Nombre, sección, y dos imágenes b) Nombre, sección, imagen y archivo. c) Nombre, sección, descripción e imagen. e) No los recuerdo.	A
Enlazar un seminario a una conferencia es obligatorio A) Sí B) No. C) No lo recuerdo.	A
En la sección de Estudios se permite insertar imágenes y archivos. A) Sí. B) No. C) Solo imágenes. D) No lo recuerdo.	A
Memoria de recuerdo.	

Enumera las opciones que recuerdes de las proporcionadas en el panel de Noticias	Título, subtítulo, descripción, calendario, visible, listado de noticias, sección, archivo, imagen. Y buscador.
Enumera Las opciones que recuerdes en el panel de Conferencias.	Título, seminario, descripción, buscador, visible, listado de conferencias, imagen y archivo.
Enumera Las opciones que recuerdes en el panel de Estudios.	Título, subtítulo, archivo, descripción, falta listado de estudios e imagen. (Falta buscador).
Enumera Las opciones que recuerdes en el panel de Expertos.	Título, subtítulo, sección, imagen, descripción, listado para editar y borrar (falta imagen pequeña y archivo)
Tiempo empleado en recordar la estructura de...	
Página de Conferencias.	1 minuto 5 segundos (incompleta)
Página de Estudios.	55 segundos (incompleta)
Página de Noticias.	1 minuto y 6 segundos
Satisfacción con la página web. Puntúa de 0 a 10.	
Calidad.	7
Facilidad de uso.	8
Agrado.	7

Valoración: El usuario 3 ha realizado las actividades de forma eficiente y ha recordado los distintos elementos de la aplicación de forma satisfactoria.