



Universidad de Valladolid

E.T.S. Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Sistema web gestor de convenios internacionales

Autor:

D. Ismael Medina Muñoz

Tutor:

D. Manuel Barrio Solórzano

Para mi familia y amigos que
de una manera u otra han hecho posible este proyecto

Resumen

Este documento muestra el desarrollo del sistema gestor de convenios internacionales para el Servicio de Relaciones Internacionales de la Universidad de Valladolid. Este proyecto busca la automatización del proceso de trámite y gestión de los convenios internacionales, ya que actualmente es un proceso largo y realizado a papel por el personal de la universidad, pudiendo provocarse errores humanos que provoquen retrasos y problemas para los alumnos.

Índice de Contenidos

Índice de Contenidos	7
Capítulo 1.....	11
Introducción y Objetivos	11
1.1. Introducción	12
1.2. Objetivos	13
1.3. Objetivos secundarios o derivados de éstos.....	13
Capítulo 2.....	15
Marco Conceptual.....	15
2.1. Marco Conceptual	16
2.2. Proceso de tramitación	19
2.3. Renovación.....	22
Capítulo 3.....	23
Planificación y Seguimiento	23
3.1. Planificación y Seguimiento.....	24
3.2. Actividades realizadas	24
3.3. Primera fase.....	25
3.4. Segunda fase.....	27
3.5. Tercera fase	29
Capítulo 4.....	31
Desarrollo de la Aplicación	31
4.1. Desarrollo de la aplicación	32
4.1.1. Diseño preliminar de la interfaz de usuario	32
4.2. Requisitos del sistema	37
4.2.1. Introducción	37
4.2.2. Requisitos Funcionales	37
4.2.3. Requisitos No Funcionales.....	42
4.3. Modelo de Casos de uso.....	44
4.3.1. Actores	44
4.3.2. Casos de uso.....	44
4.3.3. Diagrama de Casos de Uso	46
Diagrama de casos de uso de añadir	47
Diagrama de casos de uso de modificar.....	47
Diagrama de casos de uso de eliminar	48

4.3.4.	Descripción de los Casos de Uso.....	49
4.3.5.	Diagrama de clases de análisis	58
4.4.	Diseño	60
4.4.1.	Arquitectura del sistema	60
4.4.2.	Funcionamiento de los patrones utilizados.....	62
4.4.3.	Gestión de la persistencia	65
4.4.4.	Vista lógica del sistema.....	67
4.4.5.	Topología del sistema.....	68
4.4.6.	Diagrama de despliegue.....	70
4.4.7.	Aspectos de seguridad.....	71
4.4.8.	Diseño del sistema	73
4.4.9.	Casos de uso de Diseño.....	76
4.4.9.1.	Diagrama de casos de uso.....	77
4.4.9.2.	Descripción de los casos de uso.....	78
4.4.9.3.	Diagramas de clases de diseño	86
4.4.9.4.	Realización de casos de uso.....	95
4.4.9.5.	Diagramas de clases de diseño (operaciones).....	100
4.5.	Implementación.....	104
4.6.	Pruebas.....	114
4.7.	Despliegue	123
	Capítulo 5	125
	Manuales	125
5.	Manuales	126
5.1.	Manual de usuario.....	126
5.1.1.	Login de administrador	126
5.1.2.	Menú inicial	126
5.1.3.	Buscador de convenios	127
5.1.4.	Trámites de convenios	130
5.1.5.	Convenios caducados y pendientes	134
5.1.6.	Gestión de los datos	135
5.1.7.	Documentación	138
5.2.	Manual de instalación	140
	Capítulo 6	141
	Conclusiones	141
6.	Conclusiones	142
6.1.	Resumen y valoración del trabajo	142

Sistema web gestor de Convenios Internacionales

6.2.	Problemas y dificultades	143
6.3.	Líneas futuras	143
	Capítulo 7.....	145
	Bibliografía	145
7.	Bibliografía.....	146
	Apéndices	147
8.	Apéndice I	147

<i>Figura 1: Relaciones Internacionales</i>	17
<i>Figura 2: Eye-tracking del patrón F</i>	33
<i>Figura 3: Patrón F</i>	34
<i>Figura 4: Estructura pagina inicial</i>	34
<i>Figura 5: Estructura de la web sobreexpuesta con el contenido</i>	35
<i>Figura 6: Vista página convenios caducados</i>	36
<i>Figura 7: Diagrama de casos de uso</i>	46
<i>Figura 8: Diagrama de casos de uso de Añadir</i>	47
<i>Figura 9: Diagrama de casos de uso de Modificar</i>	47
<i>Figura 10: Diagrama de casos de uso de Eliminar</i>	48
<i>Figura 11: Diagrama de clases</i>	59
<i>Figura 12: Esquema patrón FrontController</i>	64
<i>Figura 13: Esquema patrón DAO Factory</i>	66
<i>Figura 14: Vista lógica inicial</i>	68
<i>Figura 15: Vista lógica final</i>	69
<i>Figura 16: Diagrama de despliegue</i>	70
<i>Figura 17: Formulario de login</i>	71
<i>Figura 18: Esquema SQL Injection</i>	72
<i>Figura 19: Diagrama de Clases ER</i>	74
<i>Figura 20: Modelo Relacional de la Base de datos</i>	75
<i>Figura 21: Diagrama de casos de uso Diseño</i>	77
<i>Figura 22: Vista lógica</i>	87
<i>Figura 23: Paquete Persistencia</i>	88
<i>Figura 24: Paquete Controlador</i>	89
<i>Figura 25: Paquete FrontController</i>	90
<i>Figura 26: Paquete Controlador AJAX</i>	92
<i>Figura 27: Paquete Vista</i>	93
<i>Figura 28: Paquete Vista Admin</i>	93
<i>Figura 29: Paquete Vista User</i>	94
<i>Figura 30: Paquete Vista AJAX</i>	94
<i>Figura 31: Diagrama de secuencia de Ver Legislacion</i>	96
<i>Figura 32: Diagrama de secuencia de Buscar Convenio</i>	97
<i>Figura 33: Diagrama de secuencia de Crear Convenio</i>	98
<i>Figura 34: Diagrama de secuencia de Añadir Coordinador UVA</i>	100
<i>Figura 35: Paquete de Persistencia (operaciones)</i>	101
<i>Figura 36: Diagrama de clases ER (operaciones)</i>	103
<i>Figura 37: Vista de la estructura generada por el código</i>	105
<i>Figura 38: Formulario emergente generado por jQuery y Javascript</i>	110

Capítulo 1

Introducción y Objetivos

1.1. Introducción

Desde que en 1957 se hiciese realidad el proyecto de una Europa unida, las fronteras entre los países comunitarios han ido difuminándose en favor de una conexión que, gracias a la globalización, se ha magnificado a escala global. En el campo de los estudios, y concretamente en el ámbito de educación superior, las instituciones universitarias son algunos de los principales organismos más interesados en una cooperación mutua que contribuya a mejorar la formación de los estudiantes con el objetivo esencial de prepararlos de cara al futuro.

Así, en las universidades surge la iniciativa de facilitar a los estudiantes y personal de la universidad la posibilidad de trasladarse a otros países para conocer otras culturas y empaparse de ellas. La movilidad a países diferentes a la nación de origen, ya no solo con el propósito de estudio, sino también desde la perspectiva laboral, supone una fuente de enriquecimiento personal e intelectual que mejora las cualidades profesionales de aquellos que han tomado esa decisión.

Asimismo, la cooperación supone un beneficio para el país que acoge a la persona porque se produce una fusión entre las costumbres y el *modus operandi* de la cultura emisora y de la cultura receptora, lo que se traduce en una nueva forma de trabajar o estudiar con unos horizontes más amplios y complementada por ciertos elementos que, de no haber llevado a cabo la movilidad, serían prácticamente imposibles.

Estas ventajas mutuas en la movilidad hacen que desde hace varias décadas la distribución de las universidades cree un departamento específico que se centra de forma exclusiva en gestionar y llevar a cabo los intercambios tanto de personas como de información. Nacen así los departamentos de Relaciones Internacionales con la intención de focalizar las movibilidades de las diferentes facultades en una única oficina internalizada.

Como iniciativa propuesta desde el departamento de Relaciones Internacionales de la Universidad de Valladolid, y en colaboración con el tutor Manuel Barrio, este proyecto pretende crear un sistema informático basado en web que permita la automatización de la gestión de los convenios internacionales que este departamento lleva a cabo con universidades de todo el mundo. Dicho sistema deberá encargarse tanto de la gestión de dichos convenios, como de todos los elementos relacionados, ya sean coordinadores, universidades, departamentos, etc.

Actualmente, todo este proceso está realizado por el personal del servicio de Relaciones Internacionales mediante un software no especializado para estas tareas. Por este motivo, surge la idea de la presente aplicación web, un sistema accesible desde cualquier navegador, conectado a los servidores del Departamento, que permita realizar la gestión de los convenios de una manera fiable y automatizada.

1.2. Objetivos

Los objetivos que planteamos en este proyecto son los siguientes:

- Realizar una aplicación web que permita la gestión informatizada de los convenios internacionales del Servicio de Relaciones Internacionales de la Universidad de Valladolid.
- Utilizar tecnologías web que permitan acceder a los estudiantes y profesores a la información que brinda Relaciones Internacionales.
- Poner en práctica conocimientos adquiridos en el curso académico sobre tecnologías web, funcionalidad y usabilidad.
- Permitir a los administradores de la aplicación web realizar las mismas tareas que realizan actualmente en papel.
- Mejorar el tiempo de trámite de los convenios transcurrido en el Servicio de Relaciones Internacionales mediante conceptos de usabilidad.
- Centralizar todas las tareas del personal de Relaciones Internacionales en una única aplicación.
- Poner a disposición de la comunidad universitaria toda la información relacionada con los convenios internacionales.
- Simular un proyecto profesional real, siguiendo todos los pasos de análisis, diseño e implementación, tratando de cumplir con todos los requisitos establecidos en un periodo de tiempo establecido.

1.3. Objetivos secundarios o derivados de éstos.

- Eliminar el uso de aplicaciones no especializadas con la gestión de convenios.
- Realizar una aplicación atractiva que busque el interés del usuario.
- Eliminar el trabajo realizado a mano por el personal de Relaciones Internacionales, suprimiendo posibles fallos que se puedan cometer.
- Promover la interactividad de la aplicación con los estudiantes y la comunidad universitaria de manera que haya una comunicación más eficiente, mejorando la productividad del Departamento.

Capítulo 2

Marco Conceptual

2.1. Marco Conceptual

Los convenios internacionales son la herramienta de la que se valen las universidades para establecer acuerdos entre ellas.

A lo largo de la historia ha quedado claro que una entidad que no se relacione con el resto del mundo, o con el entorno que le comprende, está destinado a fracasar. Al igual que lo hacen las empresas privadas, las universidades buscan socios, acuerdos y relaciones que les lleven a obtener beneficios en sus profesiones, pero las universidades son entidades sin ánimo de lucro cuyo objetivo principal radica en que sus estudiantes alcancen un alto grado de excelencia tanto a nivel académico como investigador.

Por este motivo, si miramos por los intereses de la universidad, buscaremos que nuestros estudiantes y profesores puedan ponerse en contacto con otras universidades, ya sea mediante intercambios físicos o mediante intercambios de información. De un modo u otro, lo que lograremos será mejorar y aumentar el conocimiento de la comunidad universitaria sobre otras culturas, costumbres y sobre el conocimiento que éstos han adquirido con el paso del tiempo.

Así, queda claro que las universidades deben relacionarse entre ellas para mejorar la calidad de la enseñanza que se da tanto a los alumnos que saldrán titulados como a los profesores, que serán los encargados de enseñar a dichos estudiantes.

Aunque actualmente nos encontremos en tiempos difíciles para llevar a cabo estas tareas, conviene recordar que cuanto mejores sean los estudiantes salidos de una universidad, mayor será su prestigio, lo que derivará en un mayor número de matriculaciones y por ende, en mayores ingresos. Analizando esto, parece lógico que deba invertirse en este aspecto.

Ahora que conocemos la importancia del contacto entre universidades, pasaremos a contemplar cómo se lleva esto a cabo. Como hemos dicho al inicio, la llave de este proyecto son los Convenios Internacionales.

Los convenios internacionales son el reflejo de conversaciones, reuniones e intereses entre universidades. En ellos se refleja de qué manera se va a establecer la comunicación y de qué manera van a relacionarse.

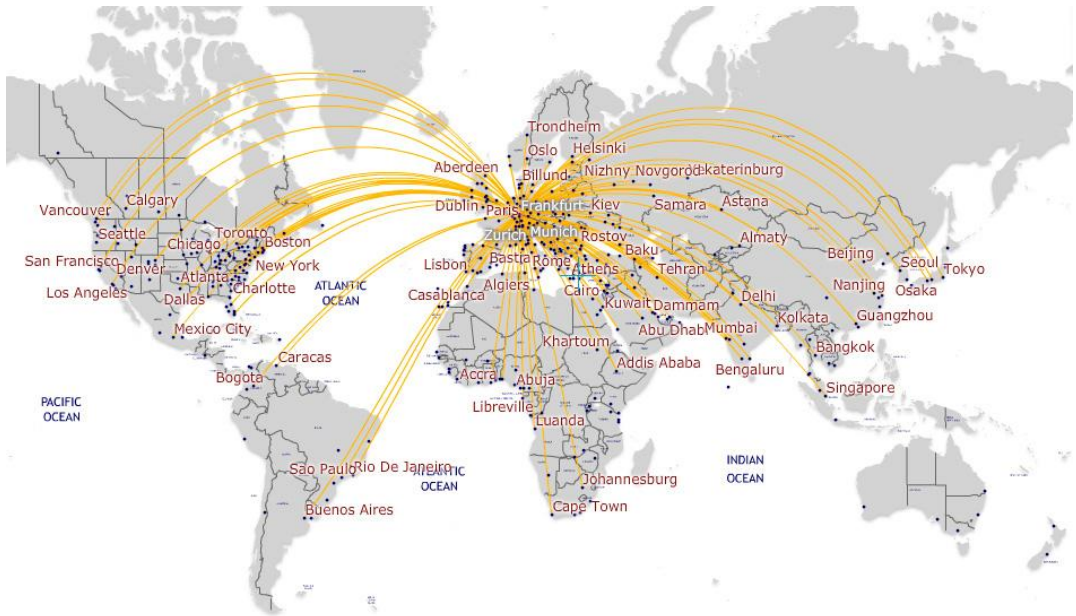


Figura 1: Relaciones Internacionales

Existen dos tipos de convenios que pasamos a explicar a continuación:

- **Convenios Marco:** como su propio nombre indica, son el “marco” de la relación. Son la base de la que partir y establecen las pautas de la comunicación, abriendo el camino a proyectos específicos.
- **Proyectos específicos:** en ellos se concreta de qué manera vamos a relacionarnos, es decir, qué tipo de actividad se va a llevar a cabo. Por ejemplo, estableceremos si habrá intercambio de estudiantes, profesores, colaboración en publicaciones, proyectos de investigación...

Si llevamos estos convenios a un punto menos formal, con el convenio marco establecemos que somos amigos y con el proyecto específico establecemos a qué vamos a jugar. De hecho, existen convenios marco que son firmados para que en un futuro se establezcan más relaciones, pero en el momento de la firma, no se tiene intención de firmar ningún proyecto específico.

Para concretar las diversas acciones que se pueden llevar a cabo entre dos o más instituciones socias se establecen las actividades tipo. Estas también se dividen en dos grupos, las actividades de los convenios marco, más generales, y las actividades de los proyectos específicos, más concretas.

Capítulo 2: Marco Conceptual

Actividades tipo de los convenios marco:

- Intercambio de estudiantes, profesores o personal de administración.
- Organización de actividades docentes coordinadas.
- Realización de proyectos de investigación y/o innovación.
- Realización conjunta de publicaciones.
- Organización de coloquios internacionales.

Actividades tipo de los convenios específicos:

- Doble Titulación de Grado.
- Doctorado Conjunto.
- Tesis Doctorales.
- Máster Conjunto.
- Impartición de Doctorado en la Universidad socia.
- Doble Titulación de Máster.
- Colaboración para la realización de títulos propios.
- Investigación e innovación.
- Intercambio de información científica.
- Desarrollo e impartición de currículos conjuntos.
- Docencia conjunta.
- Impartición de cursos de postgrado.
- Intercambio de profesores.
- Intercambio de lectores.
- Intercambio de estudiantes.
- Cotutela de tesis.

Las actividades listadas anteriormente son establecidas por parte de la Universidad de Valladolid con las universidades con las que se asocia.

Hasta aquí se ha descrito lo que es un convenio, su utilidad y sus funciones, ahora vamos a pasar a hablar de cuál es su proceso de tramitación, desde las primeras conversaciones entre universidades, a la firma del convenio.

2.2. Proceso de tramitación

El inicio de un convenio entre dos universidades surge normalmente por intereses comunes o bien por contactos entre el personal de las universidades, véase rectores, vicerrectores o profesores. Otro caso serían aquellos convenios promovidos por los propios alumnos que buscan un coordinador que les organice un convenio y que han contemplado la posibilidad al encontrar información en la web del Servicio de Relaciones Internacionales.

Cuando se decide poner en marcha un convenio marco y quizá también un proyecto específico (normalmente sin un convenio marco no se suelen hacer proyectos específicos, aunque se han dado casos) se establece un primer contacto con el Servicio de Relaciones Internacionales de la UVa, que explica al profesor de la universidad, a partir de ahora responsable del convenio, los documentos que debe presentar para iniciar la tramitación.

- **Documento de formalización de convenio marco.**
 - En el caso de que se quiera tramitar un proyecto específico, dentro del marco, se deberá presentar otro documento. Hay diferentes modelos según se trate de un proyecto para el intercambio de estudiantes, profesores, investigadores, etc. o para una cotutela de tesis doctoral, para la realización de una Doble Titulación, etc.
- **Formulario:** Es una ficha que debe cumplimentar el profesor responsable del convenio en la UVa (si es necesario, con la ayuda del profesor responsable en la Universidad socia).
- **Carta de Apoyo:** Es un documento que debe firmar el Coordinador de Relaciones Internacionales de la Facultad o Escuela Universitaria a la que pertenece el profesor responsable del convenio, e indica que el coordinador está de acuerdo en que se ponga en marcha el convenio y el proyecto específico, si lo hubiera, ya que considera que será interesante o enriquecedor para el Centro.

Estos documentos son los que el sistema pone a disposición en la web, junto con la legislación sobre los convenios.

Una vez cumplimentados y entregados estos documentos se da de alta un convenio en el sistema con los datos suministrados por los responsables tanto de la Universidad de Valladolid como de la universidad socia con la que se desea establecer el convenio.

Capítulo 2: Marco Conceptual

A partir de este momento se conocen la universidad extranjera, los responsables académicos de las dos universidades, la oficina gestora, el tipo de convenio y su forma de renovación. En el caso de la Universidad de Valladolid, la Oficina Gestora es el Servicio de Relaciones Internacionales, pero debido a que no en todas las universidades los convenios se gestionan en este Servicio, nos vemos en la obligación de hacer esta referencia en el sistema.

Una vez que disponemos de esta información, se debe especificar qué actividades tipo de las anteriores especificadas se contemplan en el convenio. Esto viene dado por los responsables del convenio y el tipo de convenio seleccionado.

A partir de este momento es cuando el Servicio de Relaciones Internacionales comienza su trabajo de verdad. Ahora los documentos rellenados y entregados por los responsables pasarán por diferentes fases que pasamos a enumerar:

- **Documentación de Relaciones Internacionales:** Se verifica si se han entregado todos los documentos necesarios para dar de alta un convenio nuevo.
 - Documento de formalización de convenio (Marco y Específico, si lo hubiera).
 - Formulario de solicitud
 - Carta de Apoyo
- **Asesor Jurídico / Comisión de Doctorado:** El asesor jurídico es el responsable de emitir un informe indicando si los documentos que se van a firmar (convenios marco y proyectos específicos) se ajustan a la legislación española y en particular a la de la UVa (por ejemplo a la "Normativa sobre tramitación, aprobación y ejecución de acuerdos con otras universidades o instituciones extranjeras" a los Estatutos de la UVa, etc.). Los proyectos de cotutelas no se envían al Asesor Jurídico sino a la Comisión de Doctorado. Ellos hacen lo mismo, es decir asegurarse de que el proyecto de cotutela está correctamente redactado de acuerdo con la legislación sobre tesis doctorales que hay a nivel nacional y dentro de la UVa.
- **Fecha de envío a la Comisión de Relaciones Institucionales:** Está formada por diez personas, entre ellas, el Vicerrector de Relaciones Internacionales y Extensión Universitaria y la Directora de Área de Relaciones Internacionales. Esta Comisión da el visto bueno o resuelve determinados temas como convocatorias de becas, firma de convenios, etc. Normalmente no se oponen a ninguna de estas actuaciones y dan el visto bueno a todas, pero podría darse el caso de que en algún momento no estuvieran de acuerdo con algún tema y éste quedara suspendido a la espera de llevar a cabo la modificación que indicara la Comisión e incluso podría quedar totalmente paralizado.
- **Fecha de envío a la Comisión Permanente del Consejo de Gobierno:** Está formada también por diez personas, entre ellas, el Secretario General de la Universidad y la Vicerrectora de Docencia. Hace más o menos la misma función que la anterior pero con temas más generales, procedentes de diferentes Vicerrectorados o Servicios de toda la Universidad. Podemos decir que jerárquicamente esta Comisión está por encima de la anterior.

- **Fecha de envío y recepción de la firma del Rector:** el convenio marco se envía al Rector de la UVa y se recibe firmado con su aprobación. Se procede de la misma forma con el proyecto específico, si lo hubiera, pero en este caso es el Vicerrector de Relaciones Internacionales quien lo firma.
- **Fecha de envío y recepción de las Firmas Extranjeras:** toda la documentación se envía al Rector de la universidad extranjera y se recibe firmado con su aprobación.
- **Fecha de envío al coordinador:** fecha en la que se comunica al profesor responsable de la UVa que el convenio se ha firmado por las dos partes y por lo tanto ya ha entrado en vigor.
- **Fecha de envío al secretario general:** fecha de envío de/del el/los documento/s original/es firmados al Secretario General para su custodia.
- **Fecha de envío al negociado de tesis:** Cuando se trata de un proyecto específico para una cotutela de tesis, fecha de envío de una copia del documento al Negociado de Tesis.
- **Observaciones privadas:** serán útiles en el caso de que se realicen nuevos proyectos con esa universidad, o simplemente a modo de anotación para los distintos pasos. Estas anotaciones no son del interés de la comunidad universitaria y por ello no aparecen en la descripción del convenio.
- **Observaciones públicas:** se dispone de un campo para añadir las anotaciones que el personal del Servicio de Relaciones Internacionales crea oportuno mostrar a la comunidad universitaria.
- **Vigencia:** En un último paso, y si el gestor de ese convenio lo cree oportuno, el convenio entra en vigor y es dado de alta en el sistema. Para llegar a este último paso, puede que cualquiera de las fechas anteriores de envío y recepción estén vacías por cualquier motivo, y aun así se pueda dar de alta el convenio (Esta es una característica especificada como obligatoria para el sistema). Debido a esto, se permite la modificación de un convenio una vez esté dado de alta, a través del buscador.

En el momento de la entrada en vigor del convenio, según su tipo de renovación, se establece la fecha de renovación. Si se establece que la renovación es automática, no existe fecha de renovación, pero si el tipo de renovación es expresa, se establece la fecha de renovación en años completos desde el momento de la vigencia.

Ahora que el convenio comienza a funcionar, tan solo tendrá dos momentos clave. Uno de ellos es cuando el convenio pasa de estar vigente a pendiente. Esto se produce cuando faltan seis meses o menos para que el convenio caduque, algo que se notifica al administrador mediante un mensaje al inicio del sistema y mediante una tabla con todos los convenios pendientes de renovar.

Capítulo 2: Marco Conceptual

El otro momento importante para un convenio es cuando llega su fecha de renovación y se ha decidido, por cualquiera de las partes, que no se renovará el convenio. El hecho de que el convenio pase a estar pendiente antes de caducar es para poner en alerta al administrador, y que éste se encargue de advertir a las partes interesadas en el convenio de la posibilidad de renovar el convenio ante su caducidad en un periodo inferior a seis meses.

Si finalmente un convenio pasa a estar caducado, estos se mantienen en un apartado por si en un momento posterior se desea crear o renovar uno de ellos.

2.3. Renovación

En cualquier momento de la vida del convenio, éste puede ser renovado, no es necesario esperar a que el convenio pase a estar pendiente o caducado. La renovación afecta al sistema de diferente manera si se trata de un tipo de convenio u otro:

- **Convenio Marco:** si se renueva un convenio marco, el nuevo convenio sustituye al antiguo, no dejando rastro de lo anterior acordado. Tan solo se mantiene el más nuevo porque es lo que más interesa a la administración. Los anteriores convenios marco carecen de importancia.
- **Proyecto específico:** si se renueva un proyecto específico, el nuevo convenio se crea a partir de los datos del anterior y éste se deja caducar. De esta manera, mantenemos un historial sobre los proyectos que mantiene a la Universidad de Valladolid con otras universidades.

Atendiendo a lo arriba descrito, la estructura de convenios se ajusta a su jerarquía en la vida real. En lo referente a una universidad socia, disponemos de un convenio marco que establece las pautas del acuerdo y a éste se asocian los proyectos específicos que se generen a partir del acuerdo marco.

Capítulo 3

Planificación y Seguimiento

3.1. Planificación y Seguimiento

En este apartado vamos a exponer cómo se ha llevado a cabo la consecución de este Trabajo Fin de Grado. Explicaremos todos los pasos seguidos desde el primer acuerdo con el Tutor hasta los primeros prototipos que conllevarían la finalización del proyecto.

Este proyecto comienza en Marzo de 2012 como propuesta entre el Servicio de Relaciones Internacionales de la Universidad de Valladolid y el tutor de este proyecto, Manuel Barrio. Su desarrollo hasta su finalización podríamos dividirla en tres etapas.

- La primera etapa transcurre desde el inicio del proyecto en marzo 2012 a septiembre de 2012.
- La segunda desde octubre 2012 hasta enero 2013
- La tercera desde febrero 2013 hasta junio 2013, con la entrega de este proyecto.

Para cada una de las fases vamos a exponer la planificación estimada, los riesgos, los retrasos y las causas de las actividades acontecidas en dicho periodo, pero antes descompondremos el proyecto en actividades que se deben cumplimentar para llegar a su finalización.

3.2. Actividades realizadas

Es importante saber qué vamos a hacer antes de empezar a hacerlo y por ello, realizaremos una lista con las tareas que debemos llevar a cabo en orden secuencial para lograr la consecución del Trabajo Fin de Grado:

- Reunión con el Servicio de relaciones internacionales para conocer el alcance.
 - Durante la primera fase se establecieron cinco reuniones para establecer qué se deseaba implementar, qué funcionalidad debería proporcionar y hacia qué público iba dirigido.
- Alcance del sistema y análisis de requisitos.
 - Decisión sobre el diseño de las vistas.
 - Documentación sobre MVC web y otros patrones.
- Instalación de la plataforma de desarrollo y tecnologías elegidas para el desarrollo.
 - Instalación de Netbeans con el plugin PHP.
 - Instalación de servidor local Apache XAMPP.
 - Instalación PHPMyAdmin.
 - Puesta a punto de las diferentes configuraciones.
- Elaboración de prototipo sin funcionalidad.

Sistema web gestor de Convenios Internacionales

- Diseño de la base de datos.
 - Análisis y diseño de la base de datos en MySQL
 - Implantación en el servidor local.
 - Modificaciones de *encoding*.
- Diseño e implementación de los patrones de arquitectura, base de datos y control del flujo.
- Implementación login de administrador.
- Implementación de la tramitación de convenios.
- Implementación de convenios caducados y pendientes.
- Implementación de gestión de los datos de las entidades del sistema.
 - Prototipo con funcionalidad media.
- Implementación de búsqueda de convenios para usuarios.
- Implementación de búsqueda de convenios para administrador.
- Implementación de muestra de resultados y resumen de convenios.
- Implementación de gestor de ficheros.
 - Prototipo con funcionalidad completa.
- Despliegue del sistema en el Servidor de Relaciones Internacionales.
- Elaboración de la memoria del proyecto.

3.3. Primera fase

De marzo a septiembre de 2012

Estimación

Desde un primer momento se planteó terminar este proyecto en un plazo que no superara septiembre. Se planeaba realizar todas las actividades terminando el proyecto en el curso de inicio del proyecto.

En este periodo, existen 130 días laborales a 4 horas al día de trabajo. Las estimaciones se medirán en días.

Capítulo 3: Planificación y Seguimiento

- Reunión con el Servicio de relaciones internacionales para conocer alcance.
 - 5 días
- Alcance del sistema y análisis de requisitos.
 - 15 días
- Instalación de la plataforma de desarrollo y tecnologías elegidas para el desarrollo.
 - 2 días
- Elaboración de prototipo sin funcionalidad.
 - 8 días.
- Diseño de la base de datos.
 - 5 días
- Diseño e implementación de los patrones de arquitectura, base de datos y control del flujo.
 - 15 días
- Implementación *login* de administrador.
 - 2 días
- Implementación de la tramitación de convenios
 - 5 días
- Implementación de convenios caducados y pendientes.
 - 5 días
- Implementación de gestión de los datos de las entidades del sistema.
 - 10 días
- Implementación de búsqueda de convenios para usuarios.
 - 4 días
- Implementación de búsqueda de convenios para administrador.
 - 2 días
- Implementación de muestra de resultados y resumen de convenios.
 - 5 días
- Implementación de gestor de ficheros.
 - 5 días
- Despliegue del sistema en el Servidor de Relaciones Internacionales.
 - 15 días
- Elaboración de la memoria del proyecto.
 - 30 días

En total 118 días, sin contar la elaboración de la memoria que debe hacerse en paralelo a las actividades.

Riesgos

- Problemas con la tecnología de desarrollo.
- Desconocimiento de desarrollo de aplicaciones web.
- Problemas de tiempo dedicado por entregas atrasadas de clase.
- Realización de prácticas en empresa.

Retrasos y causas

El proyecto finalmente no fue entregado en la fecha acordada en septiembre de 2012, principalmente por los dos últimos puntos de los riesgos descritos anteriormente. Durante el transcurso del segundo cuatrimestre se desarrolló otro proyecto pendiente para una asignatura del primer cuatrimestre y además se realizaron prácticas en empresa desde el mes de abril.

Actividades conseguidas

- Reunión con el Servicio de Relaciones Internacionales para conocer alcance.
- Alcance del sistema y análisis de requisitos.
- Instalación de la plataforma de desarrollo y tecnologías elegidas para el desarrollo.
- Elaboración de prototipo sin funcionalidad.
- Diseño de la base de datos.
 - Análisis y diseño de la base de datos en MySQL.
 - Modificaciones de *encoding*.
- Diseño e implementación de los patrones de arquitectura, base de datos y control del flujo.
- Implementación *login* de administrador.
- Implementación de la tramitación de convenios.

3.4. Segunda fase

De septiembre 2012 a enero 2013

Estimación

Al finalizar el plazo anterior, se estima un nuevo plazo de entrega de finales de año, enero 2013. Septiembre coincide con el comienzo un curso de postgrado en Madrid y me traslado a vivir Madrid. Actividades propuestas para realizar en este periodo y su estimación en días:

- Implementación de convenios caducados y pendientes.
 - 7 días
- Implementación de gestión de los datos de las entidades del sistema.
 - 15 días
- Implementación de búsqueda de convenios para usuarios.
 - 8 días
- Implementación de búsqueda de convenios para administrador.
 - 4 días
- Implementación de muestra de resultados y resumen de convenios.
 - 10 días

Capítulo 3: Planificación y Seguimiento

- Implementación de gestor de ficheros.
 - 15 días
- Despliegue del sistema en el Servidor de Relaciones Internacionales.
 - 15 días
- Elaboración de la memoria del proyecto
 - 20 días

En total 74 días de los 98 días de este periodo, al igual que antes sin contar el trabajo dedicado a la memoria.

Riesgos

- Dejadez y relajación en el proyecto.
- Pesimismo en el alumno.
- Inicio de curso de postgrado en Madrid.
- Mudanza de lugar de residencia.
- Distanciamiento entre tutor, Relaciones Internacionales y el alumno.
- Dificultad de comunicación y establecer reuniones y entregas.

Retrasos y causas

El proyecto vuelve a no quedar rematado en este periodo principalmente por el desarrollo en paralelo del curso de postgrado en Madrid y todas las dificultades que esto atañe, sumando todo el trabajo del proyecto al establecido en dicho curso.

En este caso es diferente, ya que en enero todo el desarrollo de la aplicación queda terminado a falta de ser testeado y probado en el servidor real donde estará funcionando. Además, también queda pendiente la entrega de toda la documentación.

Actividades conseguidas

- Implementación de convenios caducados y pendientes.
- Implementación de gestión de los datos de las entidades del sistema.
 - Prototipo con funcionalidad media.
- Implementación de búsqueda de convenios para usuarios.
- Implementación de búsqueda de convenios para administrador.
- Implementación de muestra de resultados y resumen de convenios.
- Implementación de gestor de ficheros.
 - Prototipo con funcionalidad completa.

3.5. Tercera fase

Esta tercera fase no es como las anteriores. En el punto en el que nos encontramos nos falta plasmar lo elaborado durante el proyecto en la documentación definitiva y llevar a cabo el despliegue del sistema en los servidores del Servicio de Relaciones Internacionales.

En este despliegue también se incluyen las pruebas que el sistema debe pasar con su uso en situaciones reales y posibles modificaciones en el caso de encontrar algún error.

En un principio no se establecen fechas límite para final del proyecto ya que la parte técnica está terminada y tan solo quedan las pruebas.

Esta parte se alarga hasta la fecha del proyecto en parte porque el alumno continúa con el curso de postgrado y también porque el alumno está realizando unas prácticas en empresa a jornada completa.

Capítulo 4

Desarrollo de la Aplicación

4.1. Desarrollo de la aplicación

4.1.1. Diseño preliminar de la interfaz de usuario

Uno de los puntos más importantes a la hora de comenzar a pensar en un sistema basado en web es su interfaz de usuario. En este tipo de sistemas donde se busca un uso frecuente por parte de los usuarios, la interfaz de usuario es clave, ya que un error que dificulte una determinada tarea en la interfaz puede conllevar pérdidas innecesarias de tiempo, pesadez sobre las tareas, desmotivación y abandono.

Por este inconveniente, lo que buscamos en una interfaz usable para todo tipo de usuarios, fácil de entender e intuitiva. Está claro que éstas son las características perseguidas por toda aplicación o sistema y que son el resultado de un pequeño estudio por parte de los desarrolladores y diseñadores.

En principio, estudiaremos qué clases de usuarios van a usar nuestra aplicación. Esto es importante ya que no es lo mismo diseñar una interfaz de usuario para un niño de unos doce años que para un experto en una determinada materia, al igual que no es lo mismo hacerlo para el personal de una oficina o para una persona de avanzada edad.

Si tuviéramos que dividir a grandes rasgos los grupos de usuarios que van a manejar la aplicación, los dividiríamos en tres grupos:

- **Alumnos:** Personas de mediana edad entre dieciocho y treinta años en su mayoría. Podríamos asumir gran conexión con el mundo de Internet y que usan este medio prácticamente a diario, ya sea por estudios o por entretenimiento.
- **Profesores:** Personas adultas con conocimientos en plataformas virtuales tipo Moodle y acceso a Internet, mayoritariamente por trabajo y entretenimiento (correo electrónico, webs de interés, noticias, etc).
- **Personal de Relaciones Internacionales:** Personas adultas y de mediana edad con altos y específicos conocimientos en Convenios Internacionales, y conocimientos sobre Internet gracias al uso diario de herramientas web en su oficina.

Con respecto a lo explicado en apartados anteriores, se deben diferenciar dos partes en el sistema. Una parte pública que pueda ser vista por cualquier persona interesada en los convenios internacionales, ya sean Alumnos o Profesores, y otra privada para el Personal de Relaciones Internacionales, donde se gestionan y publican los convenios disponibles para la comunidad de estudiantes de la Universidad de Valladolid. Por esto podríamos haber diferenciado dos tipos de usuarios, pero hemos decidido hacer tres grupos por las diferencias de edad que se manifiestan entre alumnos y profesores que a veces son importantes para el diseño de la interfaz.

Para empezar a diseñar la interfaz de usuario deberemos conocer qué funcionalidades necesitan los usuarios anteriormente clasificados.

Sistema web gestor de Convenios Internacionales

- Alumnos y Profesores: Buscan consultar la información que el departamento de Relaciones Internacionales publica en su web, principalmente convenios con otras universidades, documentación y legislación al respecto.
- Personal de Relaciones Internacionales: Llevan a cargo la funcionalidad principal del sistema, la gestión y mantenimiento de los convenios vigentes. Por otro lado también gestionan los convenios que caducan pronto, los caducados y todas las entidades relacionadas con los convenios.

Para llevar a cabo estas funcionalidades el sistema debe dividirse en un *'front-end'* público y un *'back-end'* privado, siendo el primero la cara pública del sistema y el segundo la zona donde se gestiona lo que se va a publicar.

Para cumplir con la funcionalidad solicitada y que a su vez la interfaz de usuario cumpla con las características mencionadas de usabilidad, fácil de entender e intuitivo, recurrimos a los **patrones de diseño**. No son más que buenas prácticas llevadas a cabo a partir de estudios sobre los usuarios.

Uno de los patrones más usados y probados es el Patrón F. Este patrón defiende que en lugar de forzar al usuario lo que debe mirar, deja que las costumbres de la mayoría de los usuarios les enseñen la información que el desarrollador quiere mostrar.



Figura 2: Eye-tracking del patrón F

Lo que ha probado la eficiencia de este patrón es el *'eye-tracking'* de los usuarios (seguimiento de los ojos). Mediante unas cámaras se graba lo que los usuarios primero miran ante una aplicación cuando se abre en un navegador, dejando ver patrones comunes entre todos ellos.

El comportamiento que muestra este patrón es el siguiente:

- Los usuarios comienzan mirando la esquina superior izquierda de la pantalla.
- A continuación, contemplan toda la parte superior.
- Después los usuarios siguen observando línea a línea de arriba abajo y de izquierda a derecha.

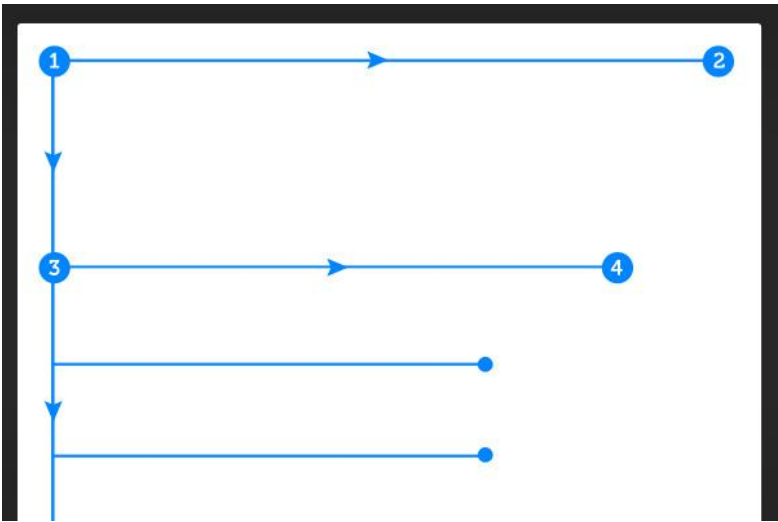


Figura 3: Patrón F

Complementando este patrón hemos decidido incluir unas pautas defendidas con la inserción de HTML5 que son la inclusión de una cabecera (*header*) y un pie de página (*footer*). Con todo esto, la estructura quedaría de la siguiente manera:

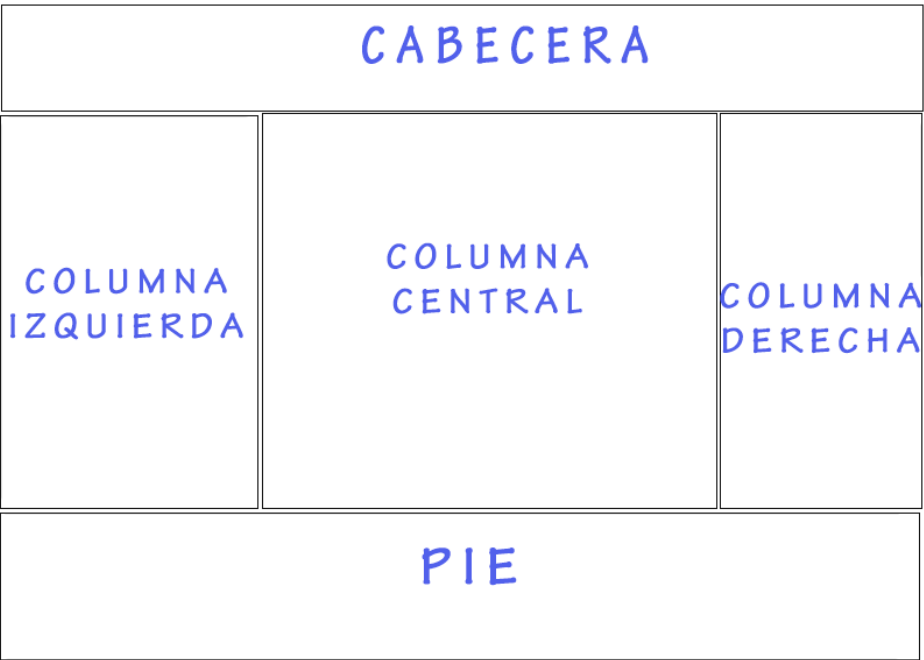


Figura 4: Estructura pagina inicial

Sistema web gestor de Convenios Internacionales

A partir de lo anterior comentado, en el cuerpo del documento, hablaríamos de tres columnas claramente diferenciadas. En la columna izquierda, colocaríamos un menú con las opciones disponibles para los usuarios. En la columna central, la información y subcategorías de las mismas, para terminar dejando en la derecha el *login* para el administrador.

Es un aspecto que no queremos que el usuario normal preste atención, por ello, lo dejamos a la derecha. Para el administrador, esta parte es muy importante ya que es su puerta de acceso y aunque esté a la derecha, permanece arriba para su mayor visibilidad.

Según las pautas anteriores el diseño inicial quedaría:



Figura 5: Estructura de la web sobreexpuesta con el contenido

En el menú izquierdo mostramos toda la diversidad de opciones que actualmente permite Relaciones Internacionales. Además esta columna izquierda mantiene la estructura con el resto de la web.

Según el usuario navegue a través de las opciones, la estructura se mantiene haciendo que el visitante mantenga la atención en el centro y no se distraiga buscando lo que desea.

Al acceder como administrador, la estructura sufre un cambio. Debido a que ya nos hemos registrado, la columna derecha pierde sentido. Por ello, ensanchamos la columna central que va a contener toda la información y de esta manera ganamos claridad en el contenido.



Figura 6: Vista página convenios caducados

Como podemos ver en la imagen, mantenemos la columna izquierda, ahora, con las opciones del administrador. Y en la nueva columna “central-derecha” mostramos la información.

4.2. Requisitos del sistema

4.2.1. Introducción

A continuación vamos a presentar los requisitos de nuestro sistema, todas las restricciones y obligaciones que la aplicación debe cumplir una vez este puesta en marcha. Están organizados por una primera distinción entre requisitos funcionales y no funcionales. Dentro de los requisitos funcionales hemos realizado una segunda división, dos bloques funcionales diferenciados. Estos dos bloques hacen referencia por un lado a todo lo relacionado con los convenios, y por el otro a la funcionalidad que el sistema debe prestar al administrador.

4.2.2. Requisitos Funcionales

Requisitos de Administrador

1. El sistema deberá soportar un único tipo de usuario, Administrador. El resto de usuarios accederán a la aplicación como usuarios sin registro alguno.
2. El sistema no permitirá la creación de nuevos usuarios privilegiados.
 - Si se desea añadir un nuevo administrador será necesario contactar con el departamento de Relaciones Internacionales.
3. El sistema deberá permitir el acceso a una zona restringida de la aplicación al personal de administración de la misma.
 - 3.1. El sistema deberá disponer de un acceso con usuario y contraseña para el administrador.
 - El nombre de usuario debe de tener como mínimo 6 caracteres.
 - La contraseña deberá de tener una longitud mayor de 6 caracteres. No puede contener caracteres que no sean alfanuméricos.
 - El sistema deberá codificar la contraseña para evitar el robo de la contraseña.
 - El sistema deberá añadir un valor 'salt' que se utilizará para aumentar la dificultad de la contraseña y hacer más difícil una posible decodificación.
4. El sistema será capaz de reconocer al usuario actual tras haberse identificado.
 - 4.1. El sistema mantendrá registrado al usuario administrador hasta que éste decida desconectarse del sistema.

Capítulo 4: Desarrollo de la aplicación

5. El sistema deberá proporcionar al administrador una zona restringida para la gestión de la base de datos asociada a la aplicación.
 - 5.1. El sistema deberá permitir la inserción, modificación y eliminación de los siguientes tipos de datos de la Universidad de Valladolid:
 - Centro, Departamento, Gestor de Convenios UVa, Rector UVa, Coordinador UVa, Vicerrector UVa, Actividad Tipo, dejando los proyectos y convenios marco aparte.
 - No se deberá poder añadir nuevas oficinas gestoras de la Uva ya que solo existe una.
 - 5.2. El sistema deberá permitir la inserción, modificación y eliminación de los siguientes tipos de datos de la Universidad Extranjera:
 - Oficinas Gestoras EXT, Gestores de Convenios EXT, Rectores EXT, Coordinadores EXT y Universidades EXT
6. El sistema deberá obligar al Administrador a insertar una Oficina Gestora, un Gestor de Convenios, un Coordinador y un Rector cada vez que se inserte una nueva Universidad.
7. El sistema deberá permitir la edición de cualquier dato presente en la base de datos relacionado con los listados en los requisitos 5.1 y 5.2
 - 7.1. El sistema mostrará todos los campos del dato que se quiere modificar para editar su contenido y volverá a ser almacenado en la base de datos.
8. El sistema deberá permitir que la eliminación de un dato no sea definitiva, ante la posibilidad de ser requerida por consultas futuras.
 - 8.1. El sistema deberá almacenar estos valores eliminados, cambiándolos de estado.
 - La entidad permanecerá en un estado no accesible para la creación de nuevos convenios, pero permanecerá almacenado para los ya creados.
9. El sistema deberá exigir al administrador rellenar un mínimo de campos para poder insertar o modificar un elemento del sistema.
 - Para entidades personales: Nombre y Apellidos.
 - Para entidades físicas: Nombre.
 - Para todas las entidades: además de los anteriores, elementos que relacionen dicha entidad con otras (Universidad, Departamento, etc.).

Sistema web gestor de Convenios Internacionales

10. El sistema deberá bloquear el borrado de un Rector, Vicerrector o Rector extranjero que estén vigentes.
 - 10.1. El sistema deberá permitir solo el borrado de estas entidades siempre y cuando no estén como Vigentes para el sistema.
11. El sistema deberá borrar aquellas entidades que dependan de un elemento borrado.
 - 11.1. El sistema deberá hacer un borrado en cascada de aquellos elementos que dependan de un elemento borrado.
12. El sistema deberá mostrar todos los listados por orden alfabético ascendente.
13. El sistema deberá permitir la búsqueda de convenios por campos para todos los tipos de usuario
 - 13.1. El sistema mostrará al usuario los siguientes campos de búsqueda para el usuario sin registrar:
 - País, Universidad y Actividad Tipo.
 - 13.2. El sistema mostrará al usuario los siguiente campos de búsqueda para el administrador:
 - País, Universidad, Actividad Tipo, Coordinador Uva y Centro de la UVa.
 - 13.3. El sistema mostrará el listado completo en el caso de que no se introduzcan criterios de búsqueda.
 - 13.4. El sistema mostrará los resultados de la búsqueda ordenados por el país y la universidad asociada a los convenios hallados.
 - 13.5. El sistema proporcionará al usuario los datos ya disponibles en la base de datos para facilitar la búsqueda.
14. El sistema deberá mostrar un listado completo de los convenios disponibles en la base de datos.
 - 14.1. El sistema deberá clasificar el listado primero por país, y a continuación por universidad, distinguiendo entre convenios marco y específicos, y a su vez entre vigentes y caducados.
 - 14.2. El sistema deberá mostrar un contador del número de convenios de cada tipo.
15. El sistema mostrará información acerca de los convenios.
 - 15.1. El sistema mostrará la siguiente información:
 - Nombre del Convenio, Universidad Extranjera, Coordinador UVa y extranjero, Gestores UVa y extranjero, Rectores UVa y extranjero, Vicerrector UVa.
 - Tipo de Convenio (si fuese Específico, convenio marco asociado), tipo de renovación (si fuese expresa, fecha de renovación) y actividades tipo asociadas.
 - Observaciones.

Capítulo 4: Desarrollo de la aplicación

15.2. El sistema permitirá consultar la información sobre entidades relacionadas con el convenio en mayor detalle.

16. El sistema deberá mostrar en todo momento la información sobre la Oficina Gestora de los convenios internacionales de la Universidad de Valladolid, para cualquier tipo de duda.

16.1. El sistema deberá mostrar:

- Nombre Oficina, nombre de la persona responsable, dirección completa, número de teléfono y fax, correo electrónico, horario de atención al público y página web.

Requisitos de Convenios

17. El sistema deberá permitir la tramitación de los convenios, tanto marcos como específicos.

17.1. El sistema no almacenará un convenio en la base de datos visible al público hasta que la tramitación de este haya sido completada.

- El momento de la finalización de la tramitación del convenio será establecido por el Administrador cuando éste lo estime oportuno.
- Se permite que haya convenios que no tengan todos sus campos de tramitación completos.

18. El sistema deberá obligar a la cumplimentación de un mínimo de campos del convenio para que se le considere en trámite.

18.1. El sistema deberá obtener un número mínimo de campos rellenados para que el convenio sea creado y pase a estar en estado de tramitación.

- Los campos a rellenar son: Nombre del convenio, Universidad extranjera, País, Coordinador de la UVa, Coordinador Extranjero, Gestor de convenios de la Uva, Gestor de convenios extranjero, tipo de convenio, tipo de renovación y actividades tipo relacionadas.

19. El sistema almacenará los diferentes estados por los que transcurre un convenio desde su iniciación, tramitación, vigencia y caducidad, sin llegar a ser eliminado como se establece en el Requisito 8.

19.1. El sistema permitirá añadir comentarios al convenio que sirvan de recordatorio al administrador para su tramitación.

- Se habilitará un campo que recoja todos los comentarios sobre los distintos parámetros del convenio.

Sistema web gestor de Convenios Internacionales

- 19.2. El sistema deberá permitir añadir comentarios públicos acerca del convenio que aparecerán en la información sobre el convenio.
- Se habilitará un campo que recoja todos los comentarios sobre la información del convenio.
20. El sistema deberá permitir llevar el control de los convenios disponibles en la base de datos.
- 20.1. El sistema deberá suministrar al administrador una visión de los convenios disponibles para su control.
- 20.2. El sistema mostrará los convenios que vayan a caducar en el plazo de seis meses en otro apartado para la gestión del administrador.
- 20.3. El sistema mostrará los convenios que se encuentran caducados en la base de datos en otro apartado para la gestión de administrador.
21. El sistema deberá permitir que los convenios pendientes y caducados puedan reinsertarse.
- 21.1. El sistema deberá permitir renovar un convenio marco que se encuentre caducado o pendiente.
- El sistema sobrescribe la información sobre el primer convenio marco.
- 21.2. El sistema deberá permitir renovar un convenio específico que se encuentre caducado o pendiente, manteniendo los anteriores como un historial
- El sistema deberá dejar caducar el convenio creando uno nuevo a partir de los datos del anterior.
 - El administrador realiza los cambios oportunos sobre la información y se activará un convenio específico nuevo.
- 21.3. El sistema deberá mantener los convenios caducados visibles a los usuarios y al administrador.
- 21.4. El sistema deberá mantener los convenios pendientes visibles a los usuarios y al administrador.
22. El sistema deberá permitir añadir y modificar datos a los convenios.
- 22.1. El sistema permitirá añadir y modificar datos de coordinadores y gestores de convenios mientras se está creando o tramitando un convenio.
- 22.2. El sistema no permitirá crear o modificar una universidad mientras se crea o modifica un convenio ya que se deberá añadir como se especifica en el Requisito 6.
23. El sistema deberá avisar al administrador de los convenios que estén cerca de caducar, para el conocimiento de éste.

Capítulo 4: Desarrollo de la aplicación

- 23.1. El sistema mostrará un mensaje al iniciar el sistema avisando al administrador de la existencia de convenios que caducarán en seis meses.
- 23.2. El sistema mostrará un mensaje para aquellos convenios que caduquen el mismo día.
- 24. El sistema deberá proporcionar el acceso al documento de información del departamento de Relaciones Internacionales.
- 25. El sistema deberá proporcionar acceso a toda la legislación disponible acerca de los convenios internacionales.
- 26. El sistema deberá proporcionar toda la documentación necesaria para realizar la solicitud de inscripción a un convenio internacional.
 - 26.1. Esta documentación deberá estar en varios idiomas:
 - Español, portugués, italiano, inglés y francés.
- 27. El sistema deberá proporcionar un medio para subir nuevos documentos para ser visualizados por los usuarios sin registro.
 - Estos documentos mantienen relación con la legislación o la documentación de los convenios.

4.2.3. Requisitos No Funcionales

- 1. El sistema deberá usarse en ordenadores con una conexión a Internet, preferiblemente, banda ancha que favorezca la conexión y el intercambio de información con la base de datos.
- 2. El sistema deberá ejecutarse en un navegador web que cuente como mínimo con las siguientes versiones instaladas en su ordenador:
 - 2.1. Google Chrome 10
 - 2.2. Mozilla Firefox 14
 - 2.3. Internet Explorer 9
 - 2.4. Opera 11

La justificación de estas versiones viene dada por la utilización de tecnologías en el lado del cliente que favorecen la interacción entre usuario y máquina.

Sistema web gestor de Convenios Internacionales

3. El sistema deberá desplegarse en un servidor web con soporte a tecnologías PHP y MySQL.
 - 3.1. PHP versión 5.
 - 3.2. MySQL versión 5.
4. El sistema deberá desarrollarse con las siguientes tecnologías:
 - 4.1. Interfaz estática: HTML.
 - 4.2. Interfaz dinámica: JavaScript y jQuery.
 - 4.3. Gestión dinámica: PHP.
 - 4.4. Administración de la Base de Datos: PHPMyAdmin.
 - 4.5. Acceso a Base de Datos: MySQL.
5. El sistema deberá desarrollarse en un servidor Apache, similar al servidor donde se va a desplegar la aplicación.
6. El sistema deberá desarrollarse mediante el uso de la aplicación XAMPP para montar un servidor Apache y la estructura de la base de datos MySQL necesaria para realizar las pruebas y la implementación.
7. El sistema deberá desarrollarse teniendo en cuenta patrones de diseño de usabilidad para su interfaz.
8. El sistema deberá estar disponible en todo momento siempre y cuando el servidor donde se encuentre desplegado no sufra ningún problema.
9. El sistema deberá mostrar mensajes de éxito y fallo cuando uno de estos hechos se produzca.
10. El sistema no deberá mantener, de manera obligatoria, la estética de las webs de la UVA, ya que se encuentra en un servidor externo.

4.3. Modelo de Casos de uso

4.3.1. Actores

Administrador: este actor representa al personal de relaciones internacionales que dispones de un acceso restringido para el control y la gestión de los convenios. Su funcionalidad es mayor que la de un usuario normal: es el encargado de llevar la gestión de la información de la aplicación. Supone el usuario objetivo de la aplicación, facilitando el trabajo que debe realizar.

Usuario: este actor representa a cualquier usuario que acceda al sistema sin acceso a la zona de administración. Normalmente son estudiantes que acceden a la información que el servicio de relaciones internacionales pone a su disposición.

4.3.2. Casos de uso

En este apartado pasamos a dar una pequeña descripción de los casos de uso de nuestro sistema. Estos son una selección de los más importantes o significativos del sistema, ya que su número total es demasiado elevado.

CU1: Ver legislación

El sistema lista la legislación relacionada con los convenios propuesta por parte del Servicio de Relaciones Internacionales.

CU2: Descargar Documentación

El sistema lista y permite descargar la documentación disponible por parte del Servicio de Relaciones Internacionales para la comunidad universitaria.

CU3: Buscar Convenio

El sistema dispone un buscador para permitir a los administradores y a los estudiantes y profesores acceder a la información sobre los convenios disponibles.

CU4: Crear Convenio

El sistema permite la creación de nuevos convenios que pasarán a ser dados de alta cuando el administrador lo crea oportuno.

CU5: Editar Convenio

El sistema permite editar la información almacenada sobre los convenios del sistema

CU6: Reactivar Convenio Caducado

El sistema almacena información acerca de los convenios caducados para que si en algún caso se deciden reactivar, se realicen de manera sencilla.

CU7: Renovar Convenio Pendiente

El sistema almacena los convenios pendientes para que en el caso de que se desee renovar un convenio, su renovación sea sencilla.

CU8: Añadir documentación

El sistema almacena los documentos que el Servicio de Relaciones Internacionales pone al servicio de la comunidad universitaria. De esta manera se permite añadir nuevos documentos al sistema.

CU9: Añadir Coordinador UVA

El sistema permite añadir nuevos coordinadores de la Universidad de Valladolid para la firma y coordinación de los convenios.

CU10: Añadir Universidad EXT

El sistema almacena las universidades con las que tiene relaciones pasadas, presentes y futuras, pudiéndose añadir en cualquier momento.

4.3.3. Diagrama de Casos de Uso

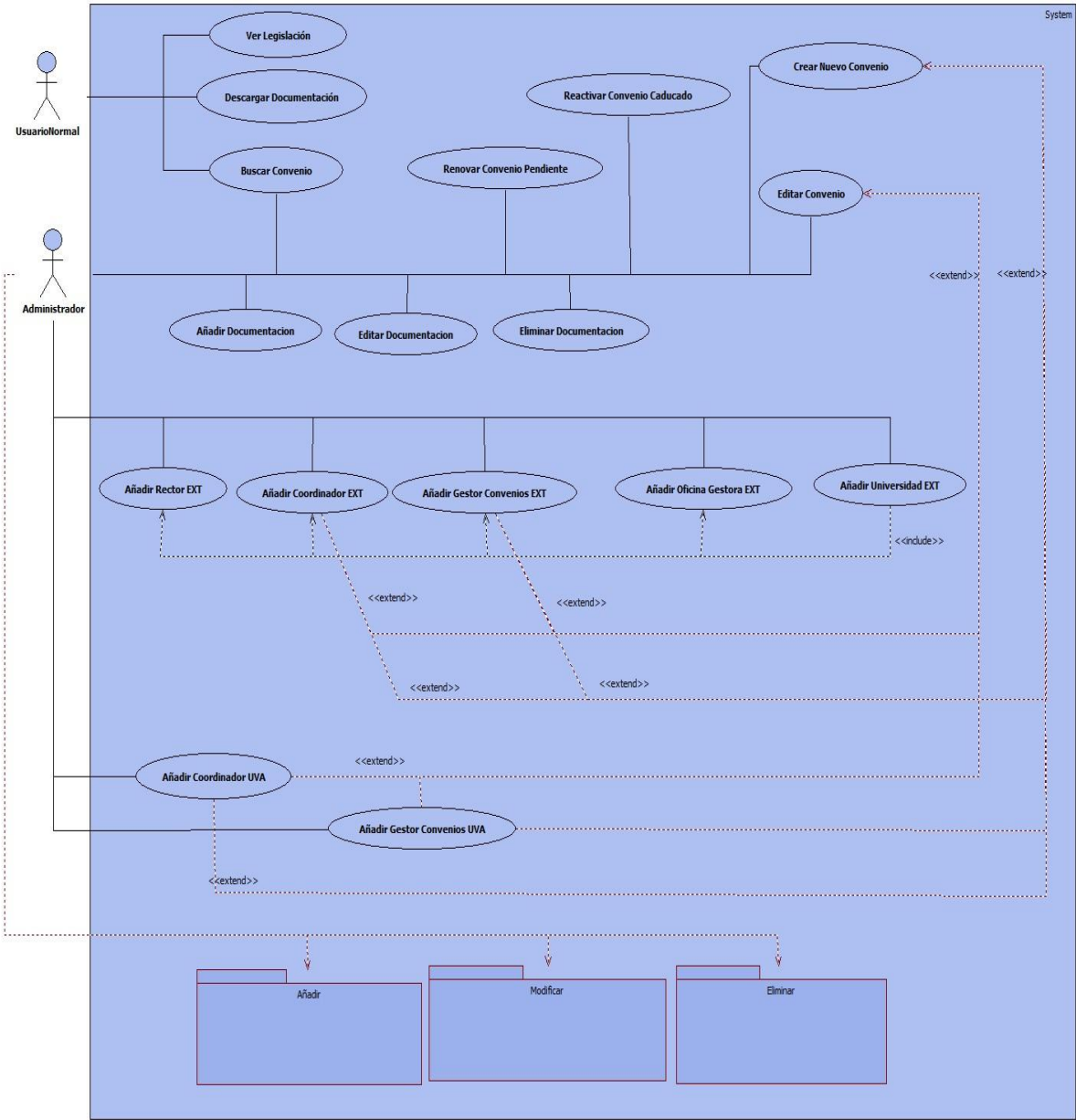


Figura 7: Diagrama de casos de uso

Debido al alto número de casos de uso relacionados con la gestión de los elementos del sistema (inserción, modificación y eliminación) y las relaciones existentes entre ellos, nos vemos obligados a separarlos en paquetes de casos de uso para mejorar su legibilidad.

Diagrama de casos de uso de añadir

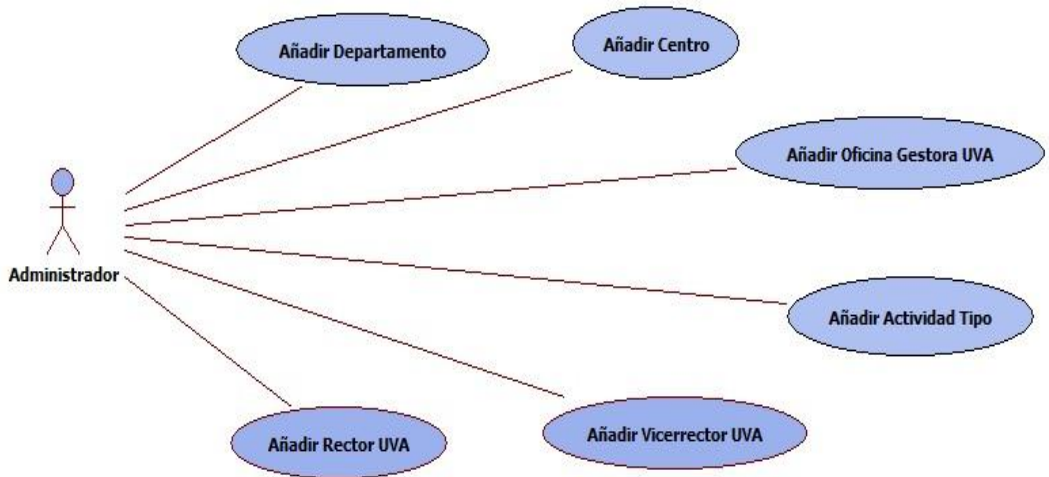


Figura 8: Diagrama de casos de uso de Añadir

Diagrama de casos de uso de modificar

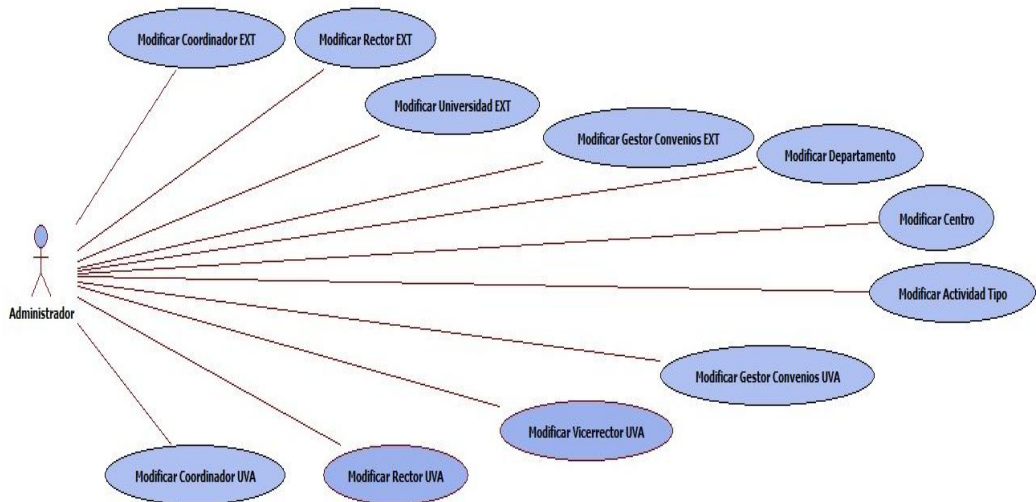


Figura 9: Diagrama de casos de uso de Modificar

Capítulo 4: Desarrollo de la aplicación

Diagrama de casos de uso de eliminar

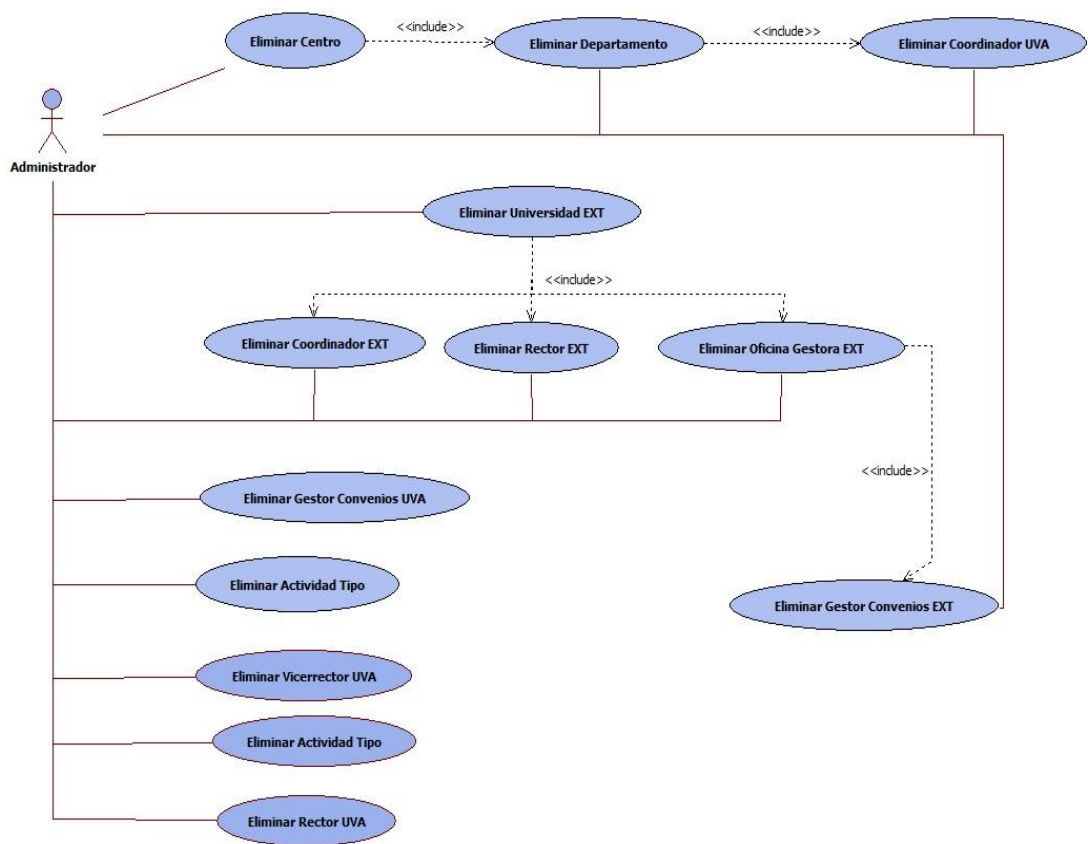


Figura 10: Diagrama de casos de uso de Eliminar

4.3.4. Descripción de los Casos de Uso

Al igual que en el caso anterior con los diagramas, dado el alto número de casos de uso, nos centraremos en detallar aquellos cuyas relaciones y flujos de acciones sean más importantes. Los que se centran en la inserción, modificación y eliminación de elementos del sistema los dejaremos referenciados en los diagramas, pero no detallados en este apartado ya que de esta manera se refleja que se deben implementar pero que los únicos cambios entre ellos son los campos que se usan para cada uno de ellos.

Caso de Uso: Ver Legislación	
ID	CU1
Descripción	El actor interesado accede a la legislación establecida por la Universidad de Valladolid
Actores Principales	Usuario Normal
Actores Secundarios	
Precondiciones	El actor debe haber accedido a la aplicación a través de un navegador.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor selecciona “Consultar Legislación”. 2. El sistema accede a la página donde se encuentra toda la legislación vigente. 3. El actor selecciona un documento de la legislación dentro del conjunto mostrado por el sistema. 4. El sistema muestra el documento seleccionado. 5. El caso de uso finaliza.
Postcondiciones	El sistema ha mostrado al usuario la legislación existente seleccionada.
Flujos Alternativos	Volver <ol style="list-style-type: none"> 3. El actor selecciona “Volver”. 4. El sistema vuelve a la página inicial y el caso de uso finaliza.

Caso de Uso: Descargar Documentación	
ID	CU2
Descripción	El actor consulta la documentación existente en el sistema y puede descargársela para sus trámites.
Actores Principales	Usuario normal
Actores Secundarios	
Precondiciones	El actor debe haber accedido a la aplicación a través de un navegador.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Descargar Documentación”.2. El sistema muestra al actor toda la documentación presente en el sistema dividida por secciones y en diferentes idiomas.3. El actor selecciona un documento.4. El sistema muestra el documento seleccionado por el usuario y la opción de descargar.5. El actor selecciona descargar.6. El sistema descarga el documento y muestra un mensaje de éxito.7. El caso de uso finaliza.
Postcondiciones	El actor ha descargado la documentación solicitada.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none">3. El actor selecciona “Volver”.4. El sistema vuelve a la página inicial y el caso de uso finaliza. <p>Solo consulta</p> <ol style="list-style-type: none">5. El actor selecciona “Volver”.6. Paso 2

Caso de Uso: Buscar Convenio	
ID	CU3
Descripción	El actor quiere consultar un convenio buscándolo entre todos los disponibles en el sistema.
Actores Principales	Usuario Normal
Actores Secundarios	Administrador
Precondiciones	Si el actor es el administrador, deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor selecciona “Buscar Convenio”. 2. El sistema muestra los campos para restringir la búsqueda. <ol style="list-style-type: none"> 2.1. Si el actor es Administrador, el sistema mostrará los campos de País, Universidad, Actividad Tipo, Coordinador UVa y Centro UVa. 2.2. Si el actor es Usuario Normal, el sistema mostrará los campos de País, Universidad y Actividad Tipo. 3. El actor rellena los campos que crea oportunos y selecciona “Buscar” 4. El sistema muestra los convenios que coincidan con los campos rellenos. 5. El caso de uso finaliza.
Postcondiciones	El sistema ha mostrado al actor el resultado de la búsqueda.
Flujos Alternativos	<p>No se encontró ningún convenio</p> <ol style="list-style-type: none"> 4. El sistema no encuentra ningún resultado y muestra un mensaje de “Convenios no encontrados”. 5. El actor selecciona “Volver” y el caso de uso finaliza. <p>No se introducen campos de búsqueda</p> <ol style="list-style-type: none"> 3. El actor no rellena ningún campo de búsqueda. 4. El sistema muestra al usuario el listado completo de Convenios. 5. Paso 4. <p>Volver</p> <ol style="list-style-type: none"> 3. El actor selecciona “Volver”. 4. El sistema vuelve a la página inicial y el caso de uso finaliza.

Caso de Uso: Crear Convenio	
ID	CU4
Descripción	El actor da de alta un nuevo convenio
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Crear Nuevo Convenio”.2. El sistema muestra los campos para la creación del convenio.3. El actor rellena los campos y selecciona “Dar de alta”.4. El sistema verifica los campos introducidos y almacena el convenio.<ol style="list-style-type: none">4.1. Si el actor ha marcado el convenio como “En trámite”, el convenio permanece como convenio en trámite.4.2. Si el actor ha marcado el convenio como “Completo”, el convenio pasa a estar a disposición de los usuarios públicos.5. El caso de uso finaliza.
Postcondiciones	Se registra un nuevo convenio en el sistema.
Flujos Alternativos	<p>Datos erróneos</p> <ol style="list-style-type: none">4. El sistema verifica que no se han introducido los campos mínimos y lo notifica al actor.5. Paso 2. <p>Datos no encontrados</p> <ol style="list-style-type: none">3. El actor no dispone de los datos para rellenar el convenio.4. <<extend>>Añadir Coordinador UVa5. <<extend>>Añadir Coordinador Exterior6. <<extend>>Añadir Gestor de Convenios UVa7. <<extend>>Añadir Gestor de Convenios Exterior <p>Datos disponibles pero incorrectos</p> <ol style="list-style-type: none">3. El actor verifica que algún dato de los disponibles para rellenar el convenio son incorrectos.4. <<extend>>Modificar Coordinador UVa5. <<extend>>Modificar Coordinador Exterior6. <<extend>>Modificar Gestor de Convenios UVa7. <<extend>>Modificar Gestor de Convenios Exterior

Caso de Uso: Editar Convenio	
ID	CU5
Descripción	El actor edita un convenio
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor selecciona “Editar Convenio”. 2. El sistema muestra los campos del convenio rellenados anteriormente. 3. El actor modifica los datos en los campos y selecciona “Editar”. 4. El sistema verifica los campos introducidos y almacena el convenio. <ol style="list-style-type: none"> 4.1. Si el actor ha marcado el convenio como “En trámite”, el convenio permanece como convenio en trámite. 4.2. Si el actor ha marcado el convenio como “Completo”, el convenio pasa a estar a disposición de los usuarios públicos. 5. El caso de uso finaliza.
Postcondiciones	Se registran los cambios del convenio en el sistema.
Flujos Alternativos	<p>Datos erróneos</p> <ol style="list-style-type: none"> 4. El sistema verifica que no se han introducido los campos mínimos y lo notifica al actor. 5. Paso 2. <p>Datos no encontrados</p> <ol style="list-style-type: none"> 3. El actor no dispone de los datos para editar el convenio. 4. <<extend>>Añadir Coordinador UVa 5. <<extend>>Añadir Coordinador Exterior 6. <<extend>>Añadir Gestor de Convenios UVa 7. <<extend>>Añadir Gestor de Convenios Exterior <p>Datos disponibles pero incorrectos</p> <ol style="list-style-type: none"> 3. El actor verifica que algún dato de los disponibles para rellenar el convenio son incorrectos. 4. <<extend>>Modificar Coordinador UVa 5. <<extend>>Modificar Coordinador Exterior 6. <<extend>>Modificar Gestor de Convenios UVa 7. <<extend>>Modificar Gestor de Convenios Exterior

Caso de Uso: Reactivar Convenio Caducado	
ID	CU6
Descripción	El actor procede a dar de alta de nuevo un convenio caducado.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Reactivar Convenio”2. El sistema muestra al actor todos los convenios caducados.3. El actor selecciona el convenio que desea reactivar.4. El sistema muestra la información disponible sobre el convenio.5. El actor realiza las modificaciones necesarias y selecciona “Reactivar”.6. El caso de uso finaliza.
Postcondiciones	El convenio queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none">3. El actor selecciona “Volver”.4. Paso 2. <p>Datos Erróneos</p> <ol style="list-style-type: none">5. El actor introduce algún dato incorrecto en los campos6. El sistema notifica el error.7. Paso 4.

Caso de Uso: Renovar Convenio Pendiente	
ID	CU7
Descripción	El actor procede a renovar un convenio pendiente (que caduca en menos de 6 meses).
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor selecciona “Renovar Convenio” 2. El sistema muestra al actor todos los convenios pendientes de renovación. 3. El actor selecciona el convenio que desea renovar. 4. El sistema muestra la información disponible sobre el convenio. 5. El actor realiza las modificaciones necesarias y selecciona “Renovar”. 6. El caso de uso finaliza.
Postcondiciones	El convenio queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none"> 5. El actor selecciona “Volver”. 6. Paso 2. <p>Datos Erróneos</p> <ol style="list-style-type: none"> 5. El actor introduce algún dato incorrecto en los campos 6. El sistema notifica el error. 7. Paso 4.

Caso de Uso: Añadir documentación	
ID	CU8
Descripción	El actor procede a añadir un documento nuevo en el listado de documentación.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Añadir Documento”2. El sistema muestra al actor todos los campos que debe rellenar para añadir el documento.3. El actor rellena los campos del documento.4. El sistema muestra un mensaje de que el documento ha sido añadido con éxito y muestra el listado de todos los documentos subidos.5. El caso de uso finaliza.
Postcondiciones	El documento queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none">3. El actor selecciona “Volver”.4. El caso de uso finaliza. <p>Datos Erróneos</p> <ol style="list-style-type: none">3. El actor introduce algún dato incorrecto en los campos4. El sistema notifica el error.5. Paso 2.

Caso de Uso: Añadir Coordinador UVA	
ID	CU9
Descripción	El actor procede a añadir un nuevo coordinador de la UVa al sistema.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor selecciona “Añadir Coordinador UVA” 2. El sistema muestra al actor todos los campos relacionados con el coordinador. 3. El actor rellena los campos del coordinador. 4. El sistema muestra un mensaje de que el coordinador ha sido añadido con éxito. 5. El caso de uso finaliza.
Postcondiciones	El coordinador queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none"> 3. El actor selecciona “Volver”. 4. El caso de uso finaliza. <p>Datos Erróneos</p> <ol style="list-style-type: none"> 3. El actor introduce algún dato incorrecto en los campos 4. El sistema notifica el error. 5. Paso 2.

Caso de Uso: Añadir Universidad EXT	
ID	CU10
Descripción	El actor procede a dar de alta una nueva universidad en el sistema.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Añadir Universidad EXT”2. El sistema muestra al actor todos los campos de la universidad.3. El actor rellena los campos relacionados con la universidad.4. <<include>> Añadir Rector EXT5. <<include>> Añadir Coordinador EXT6. <<include>> Añadir Oficina Gestora EXT7. <<include>> Añadir Gestor Convenios EXT8. El sistema muestra un mensaje de éxito.9. El caso de uso finaliza.
Postcondiciones	El convenio queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none">3. El actor selecciona “Volver”.4. El caso de uso finaliza. <p>Datos Erróneos</p> <ol style="list-style-type: none">3. El actor introduce algún dato incorrecto en los campos4. El sistema notifica el error.5. Paso 2.

4.3.5. Diagrama de clases de análisis

A continuación se presenta el diagrama de clases de análisis previo a la realización de casos de uso. En este diagrama se detallan las clases más relevantes obtenidas en un análisis superficial de los requisitos del sistema, junto con el análisis realizado previamente de la usabilidad de nuestra aplicación. Con este diagrama nos hacemos una idea básica de nuestro sistema, que nos permitirá ir dando forma a la aplicación completa.

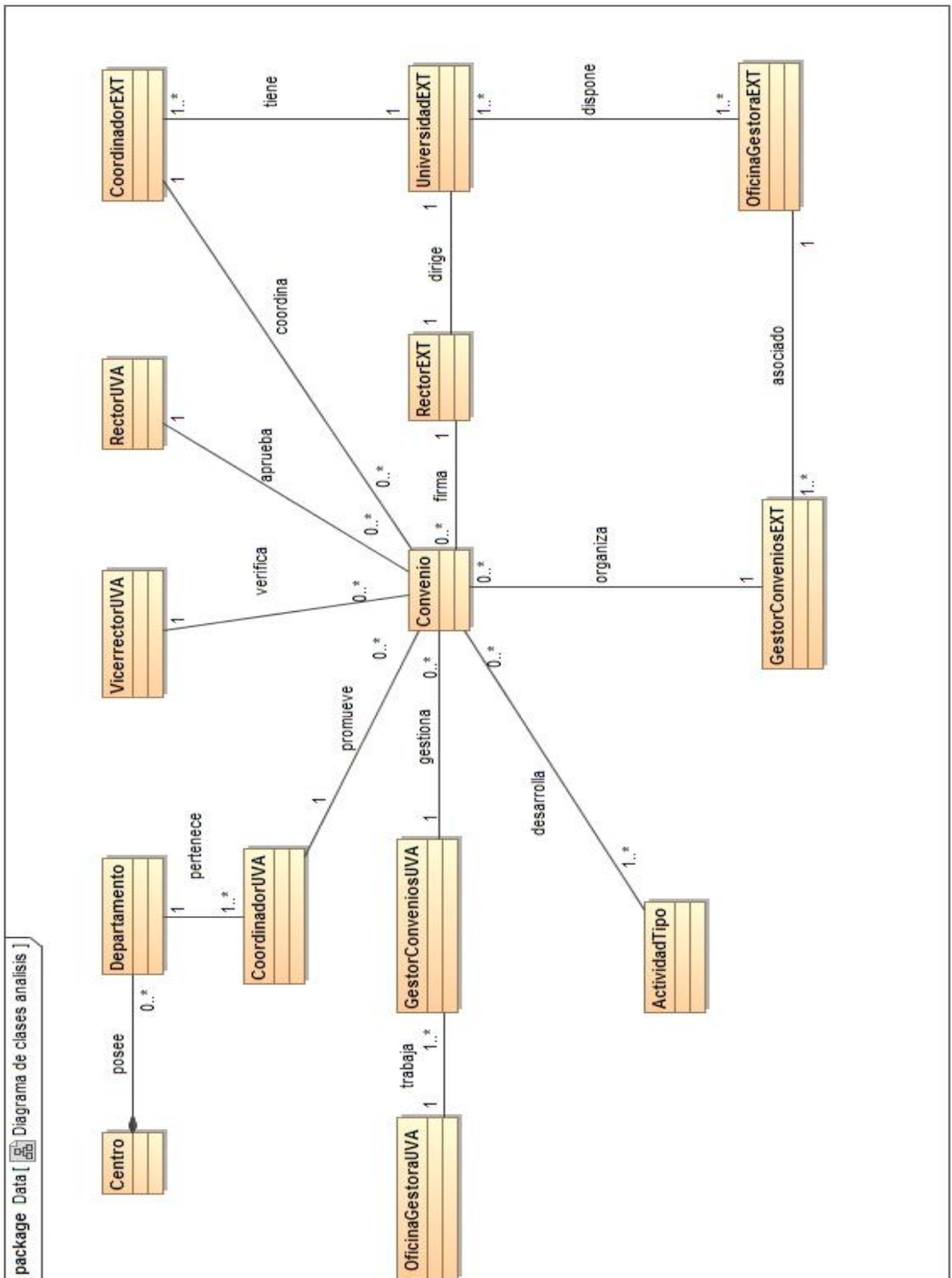


Figura 11: Diagrama de clases

Capítulo 4: Desarrollo de la aplicación

Una vez obtenidas las clases que van a formar parte de nuestro sistema pasamos a estudiar las interacciones que sufren entre ellas. Para ello, utilizamos las herramientas propias de la ingeniería de software a nivel de análisis, como son los diagramas de secuencia. Su función es representar y explicar cómo interaccionan las clases entre ellas para llevar a cabo las tareas que se les asignan gracias a los casos de uso que ya hemos detallado. En el nivel en el que nos encontramos ahora, el análisis, todas las interacciones son a un nivel muy básico. La interacción entre clases se detalla a un nivel más cercano a la realidad en el diseño.

4.4. Diseño

Una vez completado todo el trabajo relacionado con el análisis, pasamos al diseño. En esta parte deberemos acercar todo aquello que hemos afianzado durante en el análisis en un nivel más próximo a la implementación, ya que el diseño nos sirve para traducir las especificaciones del producto en algo que se desea construir, actuando de puente entre ambas fases del desarrollo software.

Para ello, vamos a empezar estudiando la arquitectura que va a tomar nuestra aplicación. Posteriormente, una vez tengamos claro esto, pasaremos a las clases de diseño del sistema y a la realización de casos de uso a nivel de diseño

4.4.1. Arquitectura del sistema

Nuestra aplicación es un sistema con dos vertientes, una para la difusión del trabajo realizado por parte del Servicio de Relaciones Internacionales, y otra para que dicho trabajo sea más eficiente y suponga una mejora del trabajo.

Acorde con esto, y desde el primer momento, se desea un sistema web, distribuido en Internet, que permita la difusión de la información y el trabajo de personal de la oficina. Por ello, el modelo que mejor se ajusta a las circunstancias es la arquitectura de Cliente-Servidor.

Este patrón arquitectónico es el usado en Internet, donde el cliente serían todas las personas que acceden al sistema a través de su navegador u otro programa, siendo el servidor la máquina que nos suministra la información a los clientes.

Al usar esta arquitectura debemos tener en cuenta la existencia de tres capas bien diferenciadas:

- **Capa de presentación:** esta es la capa que se encarga de presentar la información al usuario y proporcionar la interacción con él.
- **Capa de Aplicación:** es la capa donde se gestiona la funcionalidad básica del sistema y donde se encuentra su implementación.
- **Capa de Gestión de datos:** o también llamada capa de Base de Datos o capa de Persistencia, es aquella en la que la información que se gestiona en la capa de Aplicación se hace persistente, permitiendo futuros accesos a la misma y otorgando dinamismo al sistema.

Sistema web gestor de Convenios Internacionales

La distribución de estas tres capas entre cliente y servidor provoca la existencia dos tipos de arquitectura Cliente-Servidor:

- **Cliente Ligero (*thin client*):** en un modelo de cliente ligero, todo el procesamiento de la aplicación y la gestión de la información la lleva a cabo el servidor; es decir, que las capas de Aplicación y Base de Datos se establecen en el servidor. El cliente simplemente es responsable de la capa de presentación del software.
- **Cliente Rico o Pesado (*rich client*):** en este modelo, el servidor solamente es responsable de la gestión de los datos. En el lado del cliente se dejan la lógica de la aplicación y la presentación y la iteración con el usuario.

Ajustándonos a lo que hemos explicado anteriormente, nuestro sistema debe ser un cliente ligero, ya que nuestro sistema va a depender de la información que publica el Servicio de Relaciones Internacionales en su servidor. No tendría sentido utilizar un sistema de cliente rico debido a que el usuario solo consulta la información.

Otros usuarios, como los administradores, podrían dotarse de un cliente grueso para acceder desde sus ordenadores al servidor, pero cualquier cambio que se produjese en la lógica de la aplicación sobre la gestión de los convenios llevaría a cambiar la instalación realizada en todos los clientes.

Además de lo explicado anteriormente, los clientes ligeros son la base de la mayoría de los sistemas web actuales. Su efectividad se basa en su robustez, ya que los cambios que queramos aplicar en el sistema son posibles de aplicar en el servidor. Desde ese momento, cualquier cliente que se conecte a través de una petición recibirá la respuesta del sistema con los nuevos cambios. Estos sistemas han provocado un cambio en la concepción de las aplicaciones. Por todo esto, la decisión de montar un cliente ligero para el Gestor de Convenios parece acertada.

Una vez definida la estructura del sistema vamos a definir cómo interaccionan las capas entre ellas y cuál es su comportamiento ante la interacción del usuario.

De acuerdo con el patrón estructural que hemos seleccionado, el patrón arquitectónico que mejor se ajusta es el patrón Modelo-Vista-Controlador (MVC).

Este patrón fue descrito por primera vez en 1979 por Trygve Reenskaug, que trabajaba como desarrollador en Smalltalk para los laboratorios de Xerox. Supuso un gran avance en las aplicaciones, aunque el patrón que se diseñó en un principio ha cambiado mucho con respecto al actual a pesar de que la base sigue siendo la misma.

Su mejor característica es que nos permite desacoplar el acceso a datos y la capa de negocio del sistema de la representación que finalmente será mostrada al usuario. Con esto se nos permite también la facilidad de introducir diferentes vistas del modelo, ya que al encontrarse independiente del dominio podemos implementarlas por separado, al igual que el resto de componentes de este patrón. Esta separación otorga gran flexibilidad y reutilización a nuestra aplicación.

Capítulo 4: Desarrollo de la aplicación

En esta arquitectura existen tres tipos de objetos:

- **Modelo:** el modelo representa los datos de la aplicación y las reglas de modificación de dichos datos.
- **Vista:** la vista nos muestra el contenido del modelo. Especifica exactamente como debe ser presentado el modelo. En el caso de que se produzca un cambio en el modelo, la vista se actualiza con los eventos del usuario. En función de cómo sea la acción, se dará una respuesta u otra desde el controlador. Esto se debe a que en aplicaciones web, la vista es estática mientras que no se genere un evento en el cliente.
- **Controlador:** el controlador transmite las acciones que el usuario genera sobre las vistas en acciones que el modelo debe llevar a cabo. Debido a que hablamos de un sistema web, la comunicación se produce de manera síncrona o asíncrona a través del controlador. Si la acción se produce de manera síncrona, los eventos son disparados con acciones del usuario, generando una respuesta por parte del servidor y provocando un cambio de vista en el cliente. Pero si la acción invocada por el usuario es asíncrona, no se cambia la vista, sino que se cambia la información mostrada. El usuario no percibe el evento ni la comunicación establecida entre Modelo, Vista y Controlador.

4.4.2. Funcionamiento de los patrones utilizados

A partir del MVC tradicional, en el que la comunicación entre el modelo y las vistas se realiza directamente, ha surgido otro patrón modificado a partir de éste. En el caso de las aplicaciones web, el patrón que se utiliza no usa las notificaciones, ya que en esta clase de sistemas la vista no puede actualizarse sin la interacción del usuario.

Vamos a explicar un poco más a fondo cómo funciona este patrón mediante sus interacciones:

- Las vistas representan la información almacenada en el modelo. Esta parte carece de lógica de negocio y tan solo se encarga de recibir la información y mostrarla como el diseñador crea oportuno.
- El controlador recibe el evento de la vista y solicita la acción adecuada en función del evento generado. Una vez que el modelo genera la información necesaria, se encarga de enviársela a la vista para que se muestre. Si la información es compleja, su misión consiste en facilitar la presentación de la información.
- El modelo ejecuta la acción que le ordena el controlador y en el caso de generar información solicitada, se la entrega al controlador.
- Por último, el modelo se nutre de información a partir de la capa de base de datos, que es la encargada de mantener la información del sistema. Esta capa responde a las peticiones del modelo (nuevas inserciones, modificaciones, borrado, etc.).

Sistema web gestor de Convenios Internacionales

Utilizando este patrón lo que conseguimos es desacoplar aún más todos los componentes que forman parte de sistema, haciendo que el Controlador sea el encargado de mantener el orden de comunicación en nuestra aplicación.

Como ya especificamos antes, ahora vemos a designar el lugar que ocupa nuestro MVC dentro de las capas del Cliente Ligero. El modelo de capas nos permite mantener la división de nuestro sistema, haciendo posible la comunicación entre capas en sentido descendente. De esta manera, la capa inferior es sirviente de la capa superior, solo pudiendo comunicarse con ella a través de respuestas surgidas de las peticiones, no de peticiones directas.

Todo esto nos lleva a que la comunicación principal vaya desde la capa de presentación hasta la capa de persistencia pasando por la de aplicación, generando una respuesta por cada una de ellas, que será la que finalmente se muestre al usuario. Con esto conseguimos que el Patrón MVC elegido se ajuste perfectamente

Aplicado a nuestro sistema MVC, tendríamos:

- ❖ Capa de presentación: o también llamada capa de interfaz de usuario, en esta capa estará ubicado todo aquello relacionado con la presentación de nuestro sistema. Tendrá la responsabilidad de mantener las acciones de los usuarios bien atendidas y de la forma correcta, al igual que deberá establecer las diferentes visualizaciones que queramos dar sobre nuestra aplicación. Como veremos posteriormente, esta capa se encuentra en el cliente y en relación con el MVC, en esta capa disponemos de:
 - Vista: como ya hemos explicado será la encargada de mostrar al usuario el estado actual del sistema, así como de recoger todos los eventos ocurridos y comunicárselos al controlador.
- ❖ Capa de aplicación: se encargará de mantener todas las clases del dominio de la aplicación, mantener la gestión de los eventos de la capa de presentación y preparar las peticiones del modelo. También se encarga de las solicitudes de datos a la capa de persistencia. Esta capa se encuentra en el servidor y es el encargado de recibir las peticiones del cliente. En relación con el MVC:
 - Controlador: encargado de recoger los eventos recibidos desde el usuario y gestionar responsabilidades. Define el modo en el que la interfaz reacciona a las entradas del usuario.
 - Modelo: como ya hemos dicho anteriormente, el encargado de mantener toda la funcionalidad de la aplicación. Proporciona a las vistas todos los datos que necesiten, siendo estos solicitados a través de su controlador correspondiente.
- ❖ Capa de base de datos: esta capa es la encargada de mantener todos los datos que la aplicación necesite almacenar en memoria para que estos puedan ser recuperados en cualquier momento de la ejecución. Se encarga de mantener la persistencia de los datos y dar acceso a ellos a la capa de aplicación. Esta capa puede estar ubicada en el mismo servidor que la capa de aplicación o puede estar en otro servidor diferente.

Capítulo 4: Desarrollo de la aplicación

Front Controller

Una vez comentado los patrones más importantes del sistema, dejando aparte la conexión con la base de datos que se expone en el siguiente apartado, vamos a exponer un patrón que se encuentra en la capa de aplicación y que actúa como un guardia de tráfico, especificando hacia qué controladores se derivan las peticiones.

Su nombre es *Front-Controller* y, como ya hemos perfilado, se encarga de recibir todas las peticiones que el usuario realiza en la Vista (en la Capa de Presentación). En función del tipo de petición delega el flujo de control a otros controladores más específicos que generarán las respuestas a las solicitudes realizadas.

Realmente, es similar al patrón Fachada, cuya diferencia más plausible es que éste se suele utilizar en la conexión con la capa de Base de Datos, y el *Front-Controller* se usa en sistemas web para la conexión entre la capa de Presentación y la de Aplicación.

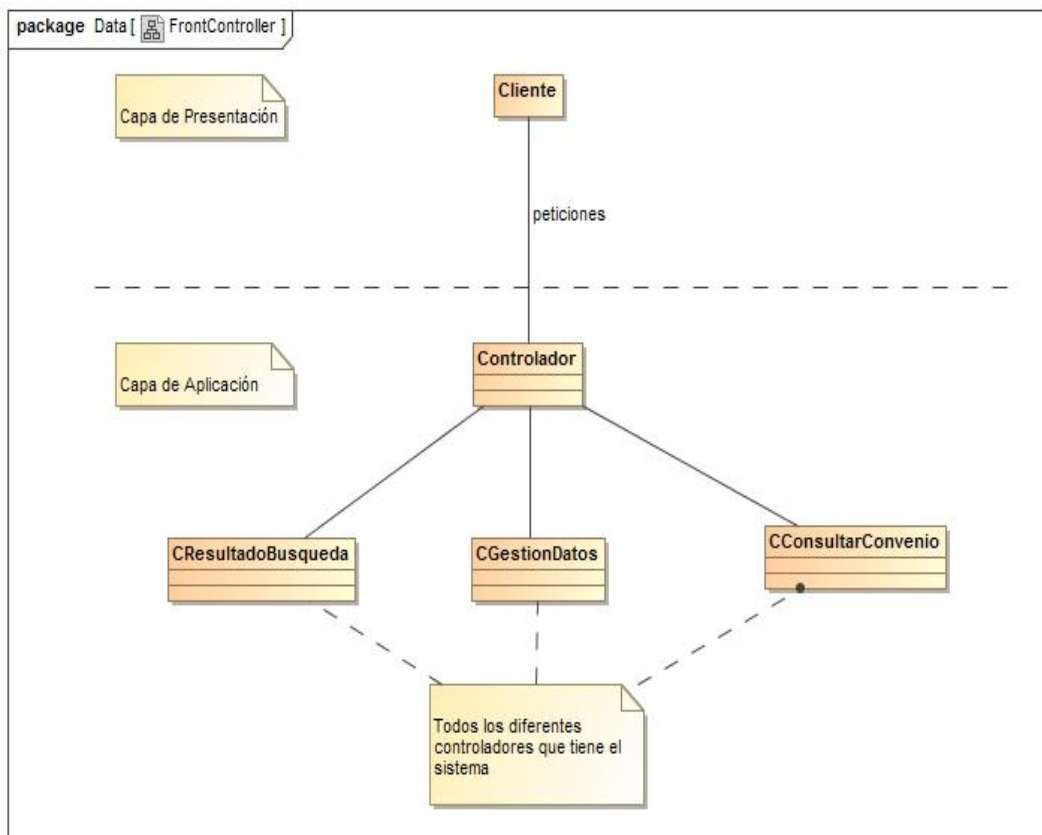


Figura 12: Esquema patrón FrontController

4.4.3. Gestión de la persistencia

Resulta una buena práctica de diseño el hecho de construir aplicaciones independientes de la tecnología de almacenamiento de datos, facilitando el diseño, desarrollo, mantenimiento y futuras ampliaciones del sistema. En el momento del diseño, utilizar una base de datos empleando una tecnología en concreto resulta ser la solución más adecuada. En nuestro caso, esa tecnología es MySQL, pero por cambios tecnológicos o razones de cualquier otra índole es posible que en el futuro se desee cambiar de base de datos. Incluso en el caso de que nunca se cambie el sistema de almacenamiento, se habrá conseguido un diseño más limpio y correcto, facilitando la posterior construcción y el mantenimiento del sistema.

En el caso en el que nos encontramos, es complicado que una universidad o el Servicio de Relaciones Internacionales, decidan cambiar los servidores de base de datos a día de hoy. Pero nunca sabremos lo que puede ocurrir en el futuro, o por lo menos, no sabremos lo que pasará en el futuro de nuestro sistema. Por lo tanto, parece muy oportuno prestar atención a esta faceta del diseño de software.

En resumen, conviene cubrir dos objetivos estrechamente relacionados, la aplicación debe adaptarse lo más fácilmente posible a un tipo de almacenamiento distinto al inicialmente previsto y debe ser lo más independiente posible del mismo.

Lo primero que hay que preguntarse es qué objetos deben ser persistentes en el sistema. En este caso la respuesta es sencilla, deben ser persistentes todas las entidades del dominio, que son las que nos interesa conservar en la base de datos y que están presentes en el diagrama de clases del análisis.

Seguidamente, hay que determinar quién es el encargado de la persistencia de cada objeto persistente. Según el patrón GRASP Experto en Información, se determina que la responsabilidad de insertar, modificar o borrar un objeto persistente debería recaer en el propio objeto. Aunque este enfoque no es completamente adecuado, primero porque las entidades del dominio no deben preocuparse por el almacenamiento, y segundo porque se convertirían en dependientes de la tecnología de almacenamiento, que es una de las cosas que pretendemos evitar. Está claro que necesitamos utilizar el patrón Factoría Abstracta para obtener el acceso a la persistencia y que éste no puede formar parte de la capa de presentación ni puede formar parte de la capa de acceso a datos, por lo tanto estará ubicada en la capa de aplicación.

Se va a utilizar un paquete “Persistencia” para agrupar la lógica de la gestión de los objetos persistentes en el sistema. Para empezar, hay que tener en cuenta que su responsabilidad principal es almacenar y recuperar datos persistentes del sistema de almacenamiento a petición del modelo, haciendo necesario por lo tanto que el modelo deba ser dependiente de este nuevo paquete. El modelo se comunicará con el acceso a datos por medio de objetos del dominio, y esto hace necesario que nuestro paquete haga referencia, y por lo tanto sea dependiente, a las entidades del dominio.

El paquete proporcionará una interfaz totalmente independiente del sistema de almacenamiento elegido, que actuará a modo de fachada, haciendo transparente la implementación del acceso físico a los datos. Hay varias maneras de lograrlo, y para este diseño se ha escogido el uso de un patrón que se adapta perfectamente al problema. Se trata del patrón DAO (*Data Access Object*, Objeto de Acceso a Datos).

Capítulo 4: Desarrollo de la aplicación

La idea del patrón DAO es proporcionar una interfaz para el acceso a los objetos de datos persistentes independiente del sistema de almacenamiento. Para cumplir este objetivo, todas las operaciones deberán ser abstractas, obligando a definir interfaces. Se implementarán interfaces para cada método de acceso a datos que se utilice.

La potencia del patrón DAO aumenta si se usa en combinación con el patrón Factoría Abstracta. La factoría abstracta es una clase que genera una factoría implementada para una tecnología en concreto. Ésta recibe peticiones para servir objetos abstractos y decide cuál de las implementaciones existentes debe instanciar y devolver. La instanciación implementa el patrón Singleton, que restringe la creación de instancias a una única instancia por clase.

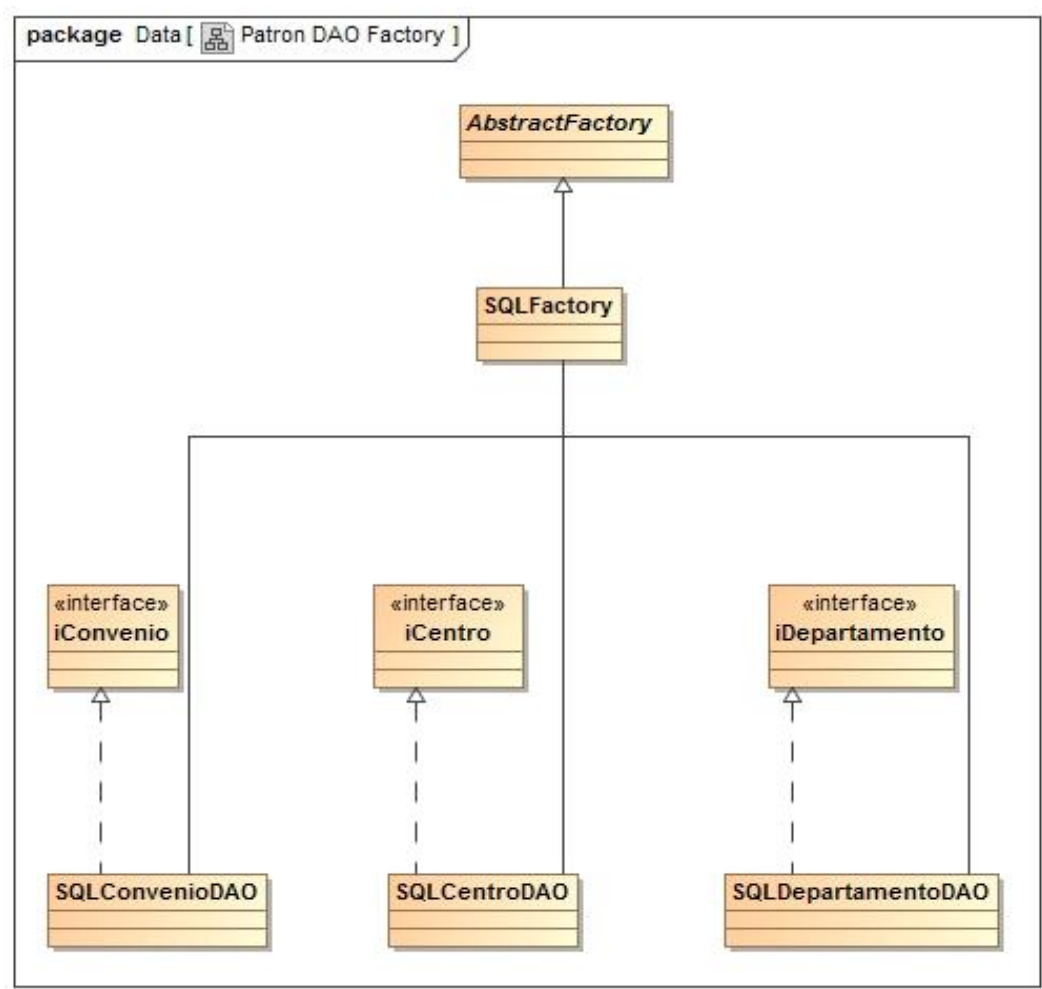


Figura 13: Esquema patrón DAO Factory

En cuanto a la implementación del patrón DAO, hay dos opciones posibles para utilizarlo: un único objeto de acceso a datos (DAO) para todos los objetos persistentes, o utilizar varios DAO independientes (uno para cada entidad). Para esta aplicación nos inclinamos por la segunda opción: se creará un objeto de acceso a datos para cada entidad del dominio. La ventaja de esta elección es que cada DAO es más cohesivo y tiene un número de métodos menor que si se siguiera la primera opción, sin embargo la gran desventaja es que aumenta el número de DAOs y se complica el mantenimiento y la obtención de instancias. No obstante, cualquiera de las dos opciones es perfectamente válida.

4.4.4. Vista lógica del sistema

Una tarea frecuente del diseño es dividir nuestro sistema en subsistemas que permitan aislar y dividir sus funcionalidades, con lo que conseguimos aumentar la cohesión del sistema y reducir el acoplamiento, aunque es cierto que debemos tener cuidado con las dependencias entre subsistemas. Estas dependencias provocan que dos o más subsistemas se encuentren acoplados debido a que un cambio producido en uno de los subsistemas, afecta al otro en relación a los datos que ambos intercambian.

A la hora de afrontar esta aplicación no nos encontramos en un gran sistema en el que podamos decir que existen diferentes subsistemas diferenciados. Simplemente dispondremos de nuestra aplicación separada por capas, que no son subsistemas, a su vez dentro de éstas, las clases del sistema agrupadas en paquetes funcionales, y cuyas relaciones son gestionadas mediante los patrones que hemos explicado anteriormente.

Para entender las relaciones existentes en nuestro sistema distribuido en capas con MVC y el resto de patrones comentados, vamos a exponer la vista lógica mediante un diagrama de paquete.

Capítulo 4: Desarrollo de la aplicación

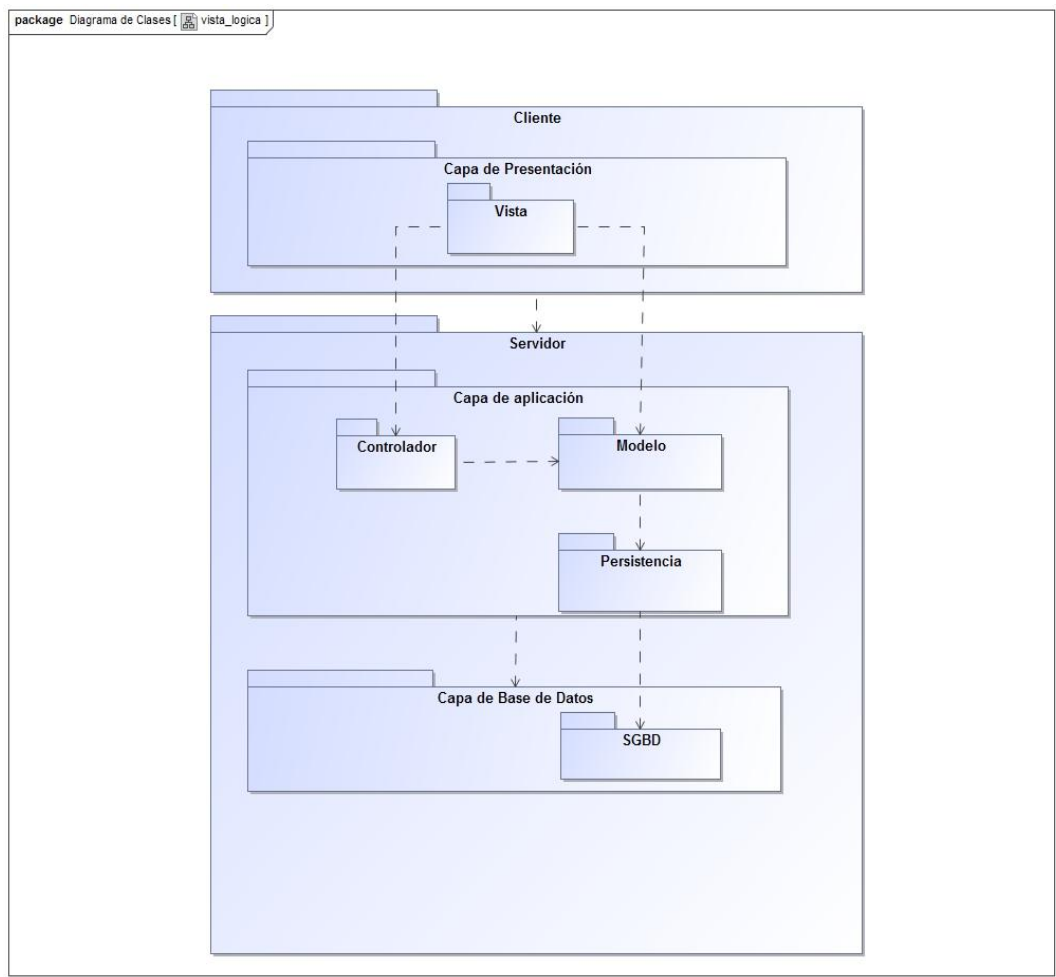


Figura 14: Vista lógica inicial

4.4.5. Topología del sistema

Nuestro sistema es una aplicación web basada en cliente-servidor con un cliente ligero, una aplicación que se ejecuta en el navegador de los usuarios dejando todo el procesamiento en un servidor dedicado al procesamiento de la información relacionada con los convenios.

El servidor se encarga de recibir las peticiones de los usuarios, procesar la información solicitada a partir de la solicitud y ciertos parámetros de entrada, generar una respuesta a partir de la información almacenada en la base de datos e incluso modificar los datos almacenados en ella. Hasta este punto, parece claro por qué el cliente es “ligero”, ya que todo el procesamiento se lo lleva el servidor.

Por todo esto, en algunos sistemas la base de datos también se distribuye, dejando que las peticiones a la base de datos se realicen a través de http, en un servidor externo. Este es el caso de la Universidad de Valladolid, que para dar servicio al Servicio de Relaciones Internacionales, dispone de un servidor de base de datos dedicado a la persistencia de su información. Esto provoca un cambio en nuestro diagrama de paquetes que expresa la vista lógica de nuestro sistema, estableciéndose lo siguiente:

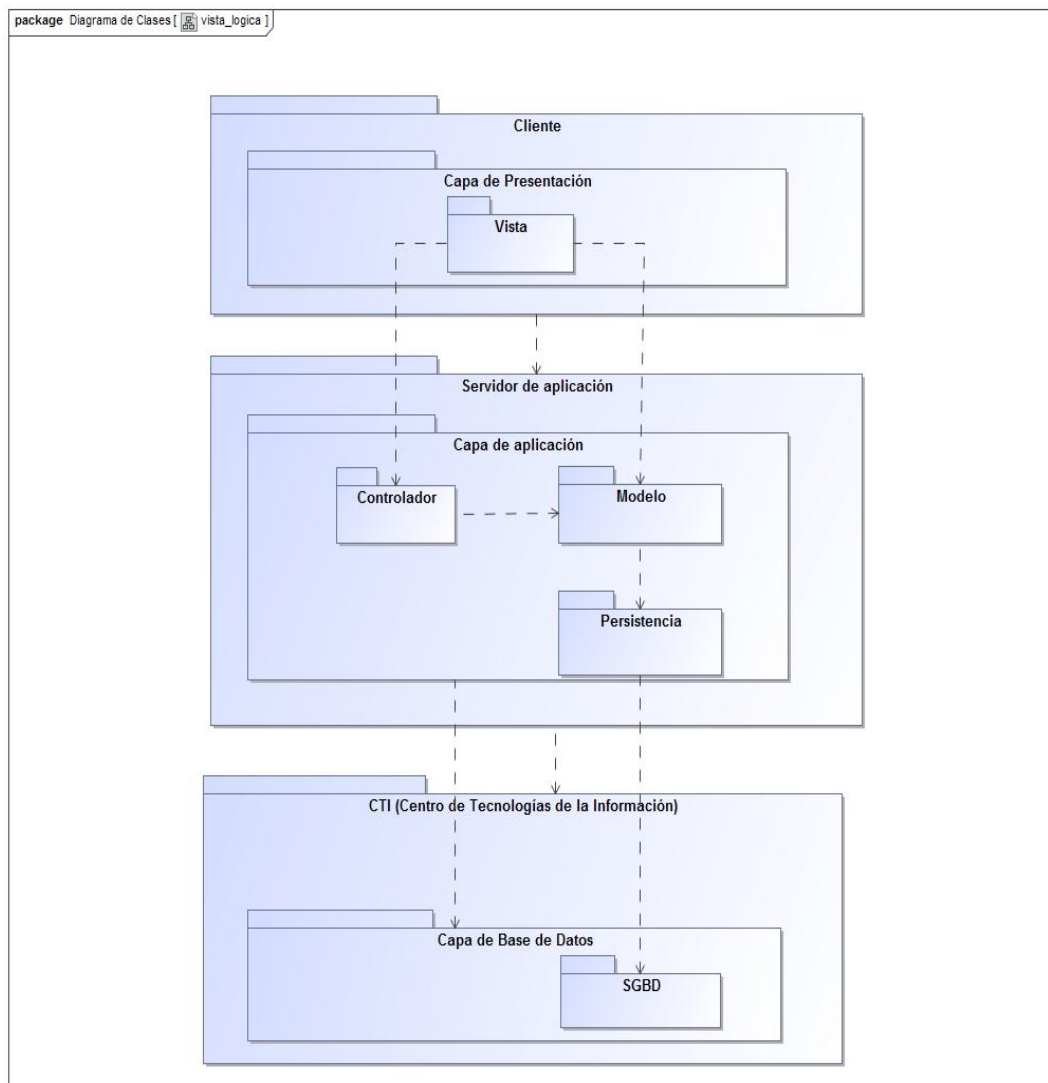


Figura 15: Vista lógica final

Capítulo 4: Desarrollo de la aplicación

Como podemos ver, ahora la capa de persistencia y el Sistema Gestor de la Base de Datos se encuentran en otro servidor separado, conectado mediante Internet, y se comunican mediante el protocolo http.

Por un lado tiene las ventajas de separar la carga de trabajo de un único servidor, en uno para la aplicación y otro para la base de datos, pero pierde eficiencia en la comunicación mediante la red de comunicación. Otro punto a su favor es que permite que otras aplicaciones o sistemas se puedan conectar a este servidor de base de datos para suministrar información.

Y por último, y quizá la más importante, permite separar el mantenimiento de dicho servidor a un departamento especializado que se encargue de los posibles problemas surgidos, ya que el personal del servicio de Relaciones Internacionales no es un personal especializado en estos aspectos. De no ser así, la Universidad de Valladolid debería crear un departamento de informática en cada departamento que disponga de una información accesible desde la red.

4.4.6. Diagrama de despliegue

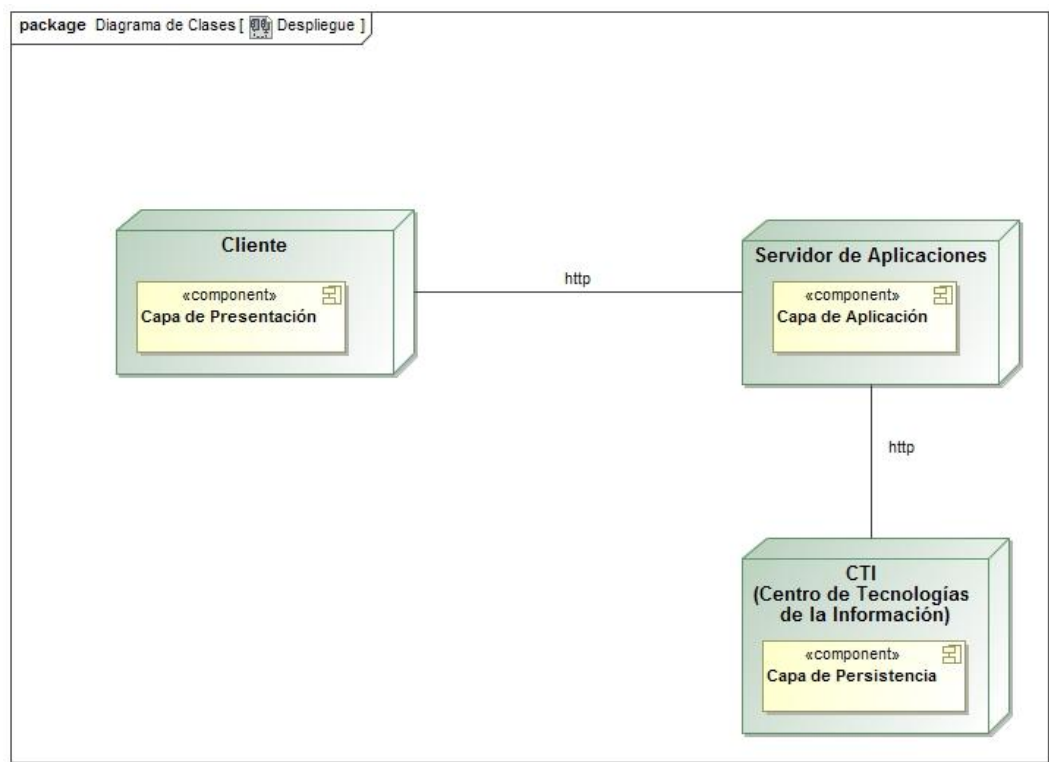


Figura 16: Diagrama de despliegue

4.4.7. Aspectos de seguridad

En una aplicación web la seguridad es un aspecto muy importante. No importa que la arquitectura y el sistema estén bien estructurados y bien implementados si dejamos un agujero de seguridad que puede provocar entre otros problemas, robo de información personal y de importancia. En la actualidad, lo que más se protege es la información. Se busca que el acceso a ésta sea controlado y restringido bajo medidas de seguridad.

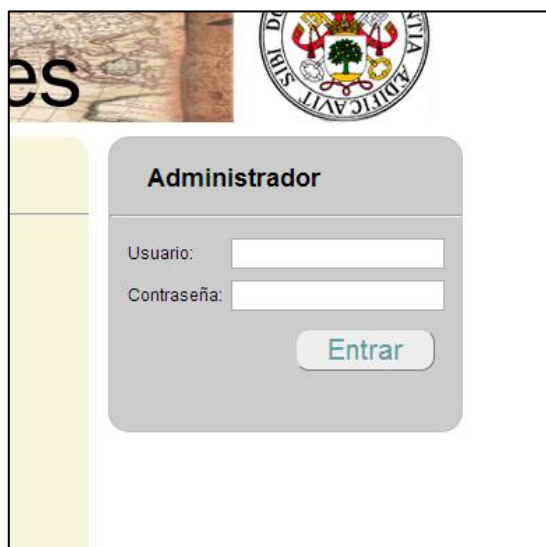


Figura 17: Formulario de login

En este sistema almacenamos información acerca de los convenios del Servicio de Relaciones Internacionales. Como ya hemos explicado, los convenios relacionan a coordinadores, rectores, gestores de convenios con sus respectivos datos personales, por lo tanto, es de obligación disponer de un mínimo de seguridad, restringiendo el acceso a los datos.

El método más común es la utilización de un nombre de usuario y contraseña para restringir dicho acceso. Se determinan dos palabras con caracteres alfanuméricos que especifican la identidad del administrador que debe ser verificada por el sistema.

Al igual que este método es el más utilizado en seguridad, también es uno de los más atacados ante vulnerabilidades de implementación. Uno de los ataques más conocidos es SQL Injection, por el que se consigue información de la base de datos desde la vista. Si no se soluciona, colocando la cadena adecuada en el campo de usuario o contraseña, podría dar acceso directo a la información.

Esto se produce porque la cadena de caracteres introducida por el usuario va viajando a través de las capas hasta llegar a la consulta de la base de datos. Si la cadena introducida es la apropiada, se podría modificar la consulta predeterminada para comprobar la identidad del usuario como se muestra en el ejemplo.

SQL Injection.

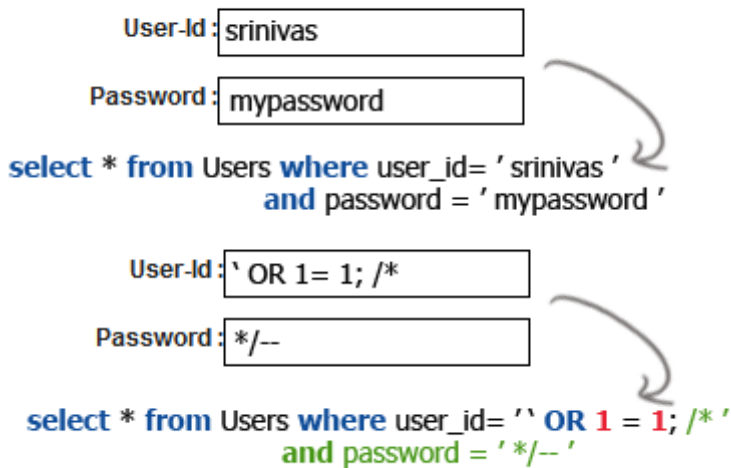


Figura 18: Esquema SQL Injection

Con esta técnica podemos tumbar la seguridad básica de un sitio web, o incluso, conocer todo el listado de usuarios y sus contraseñas.

Pero la seguridad no es solo una medida, hay que usar una combinación de ellas para evitar ataques maliciosos. Otra medida es el uso de un usuario sin privilegios para el acceso a la base de datos. Un error muy común, aunque suene extraño, es crear un usuario de la base de datos con control de administrador que le da permiso a éste para crear y modificar o borrar la estructura de la base de datos, claves, tablas, índices, etc. Por este motivo, el usuario que tiene acceso a la base de datos debe ser uno con los privilegios restringidos para añadir y modificar los datos de las tablas de datos.

Por otro lado, hemos dejado el registro de administradores en un apartado externo al sistema. Mediante un módulo separado del sistema y solo accesible para los administradores del Servicio de Relaciones Internacionales, permitimos la creación de nuevos usuarios en el sistema. El problema es que a la hora de crear nuevos administradores, las contraseñas no suelen ser demasiadas complejas y pueden ser descifradas mediante algoritmos de fuerza bruta.

Complejidad de la contraseña

Debido a que los nuevos usuarios no memorizan contraseñas demasiado complejas, introducimos la codificación de la contraseña mediante SHA1. Con esta codificación, cualquier cadena de caracteres se codifica en números haciendo que solo sea codificada en ese sentido. Si se quiere obtener la contraseña desde la codificación numérica a texto, el procesamiento de dicho proceso llega hasta los 30 años.

Aun así, palabras muy comunes son descifradas con una simple búsqueda en Internet. Para evitar esto se introduce el SALT. Este término no es más que una cadena que se concatena a la contraseña para aumentar la complejidad en la codificación. Esa cadena concatenada es la que se codifica haciendo que su decodificación sea imposible en tiempos razonables.

Normalmente se coge el valor del tiempo actual en milisegundos y se codifica y almacena junto al nombre de usuario y la contraseña.

Cuando el administrador introduce el usuario y la contraseña, el sistema recupera los valores del SALT y la contraseña a partir del nombre de usuario. Se concatena la contraseña introducida por el administrador sin registrar, con el SALT almacenado en la base de datos. Esa cadena se codifica y se comprueba si la contraseña codificada es la correcta y es igual al valor almacenado.

Con esta técnica, además, conseguimos que si un intruso consigue interceptar nuestro tráfico de red, solo recibirá la contraseña codificada, nunca la contraseña real.

Como nota final, en el acceso a cada vista de administrador se verifica que se haya registrado el administrador para evitar que se acceda a las rutas, memorizándolas.

4.4.8. Diseño del sistema

En este momento comenzamos la especificación del diseño del sistema en función de los patrones y otras medidas adoptadas en los apartados anteriores.

Como primer paso, se presentaría el modelo de Entidad-Relación (E-R) que representa cómo se estructuran los datos almacenados en la base de datos.

Por su tremendo parecido con el diagrama de clases del sistema, hemos decidido aunarlos en uno solo, ayudándonos con los estereotipos de UML para expresar qué clases tendrán una tabla persistente en la base de datos.

Debido a que de momento el diagrama de clases no ha sufrido ningún cambio con respecto al contemplado en el análisis, aprovecharemos para incluir en este diagrama de clases los atributos que se recogen a partir de las especificaciones del sistema. En estos atributos también introduciremos algún estereotipo que nos ayude, sobretodo, en las tablas con un gran número de atributos, como es el caso de los convenios.

Diagrama de clases-ER

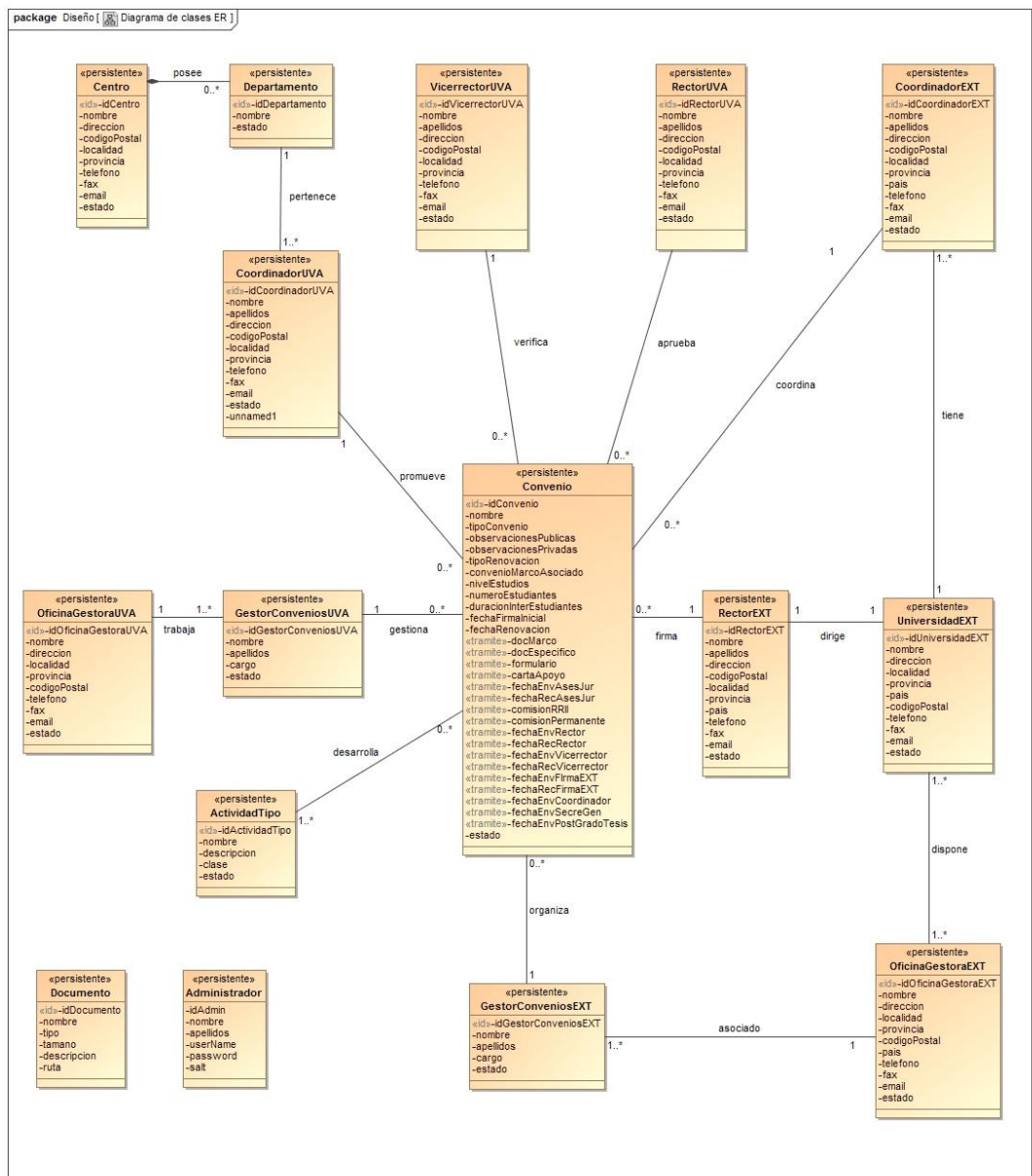


Figura 19: Diagrama de Clases ER

Sistema web gestor de Convenios Internacionales

A partir del modelo ER se obtiene el modelo relacional de los datos en el que se han traducido las relaciones en otras tablas, índices y claves foráneas. Este modelo es la representación de los datos en las tablas de la base de datos y las relaciones se establecen a través de las claves foráneas y claves principales.

Modelo Relacional

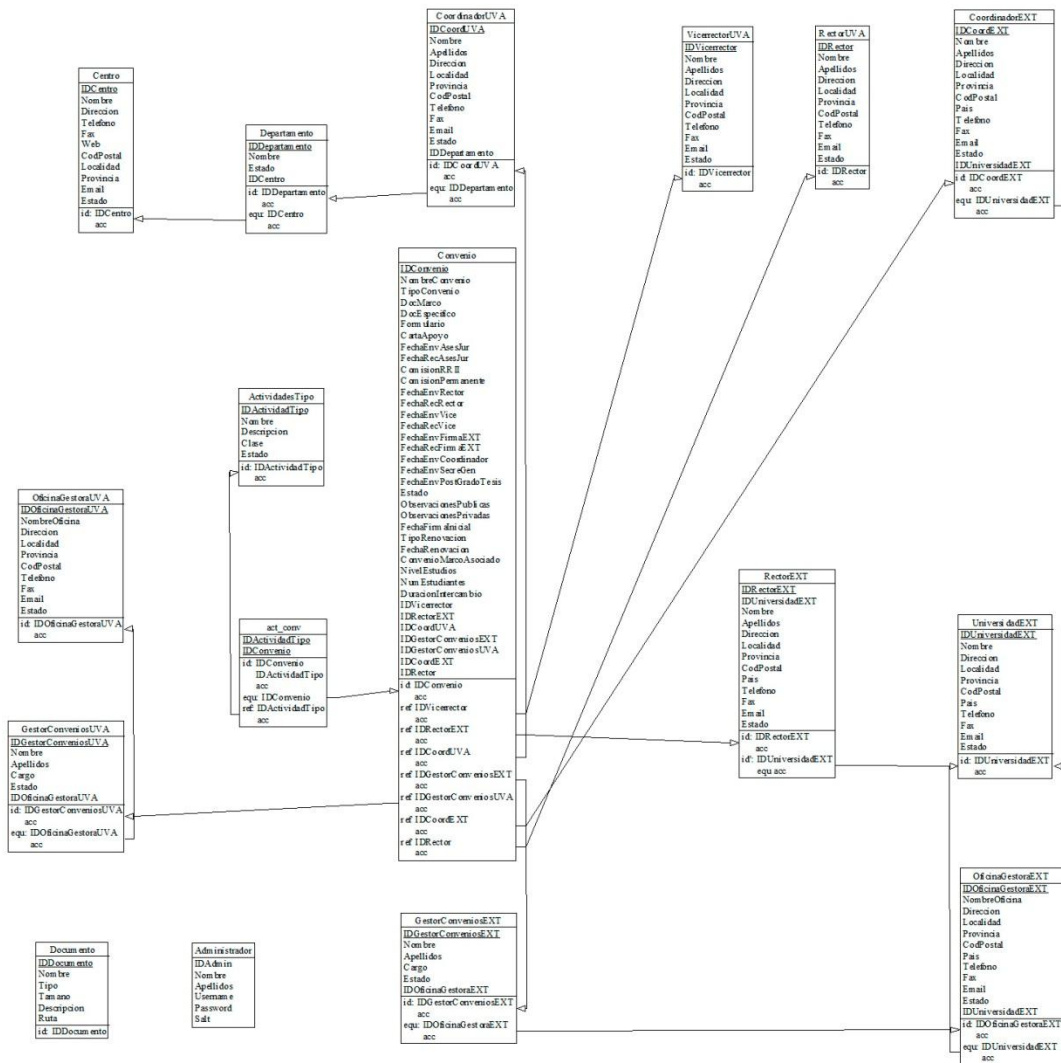


Figura 20: Modelo Relacional de la Base de datos

Capítulo 4: Desarrollo de la aplicación

La diferencia de este diagrama con el diagrama de clases es la creación una tabla intermedia entre el Convenio y las Actividades Tipo. Al ser esta relación del tipo 1 a muchos, es necesario crear una tabla intermedia que almacena los identificadores de las actividades tipo y del convenio.

Como podemos observar en la comparativa de los diagramas, todas las clases tienen representación en la base de datos en forma de tabla, donde sus datos se almacenarán. Las características del sistema y de los datos han llevado a que no exista mucha diferencia entre el diagrama de clases y el modelo relacional de la base de datos, pero esto no es muy común en otros sistemas. En nuestro caso, lo que los diferencia es la tabla intermedia que hemos comentado anteriormente, ya que no tiene representación en forma de clase del modelo, sino que forma parte de la clase Convenio como un atributo.

En primer lugar vamos a reeditar los casos de uso explicados en el análisis dando un paso más cercano a la tecnología de implementación. Luego, mostraremos un diagrama de clases distribuido en paquetes para entender un poco el sistema, para posteriormente observar sus interacciones mediante los diagramas de secuencia. Una vez hayamos completado estos diagramas pasaremos a mostrar toda aquella información que hemos obtenido en la realización de esta tarea en un diagrama de clases completo que nos ayude a dar el salto a la implementación.

4.4.9. Casos de uso de Diseño

Ahora vamos a proceder a una descripción desde el punto de vista del diseño de los casos de uso expuestos en análisis.

No se han descubierto nuevos actores del sistema, ni nuevos casos de uso. El sistema tiene una funcionalidad clara y definida, por lo cual la aparición de nuevos escenarios era poco probable.

Al realizar el paso del análisis al diseño hemos comprobado que dos de los casos de uso se pueden unificar en uno solo. Hemos unido los casos de uso del análisis “Reactivar Convenio Caducado” y “Renovar Convenio pendiente” en un único caso de uso en diseño, denominado “Renovar Convenio”. Esto se debe a que en el acercamiento hacia la implementación nos hemos dado cuenta de que de igual manera que se reactiva un convenio caducado, se renueva uno que esté pendiente de renovación; tan solo cambia ese estado del convenio. En el siguiente diagrama de casos de uso mostramos la modificación con respecto al análisis para resolver cualquier duda.

4.4.9.2. Descripción de los casos de uso

Pasamos a describir los casos de uso con un mayor grado de detalle, derivado de la elección de la tecnología y los patrones.

Caso de Uso: Ver Legislación	
ID	CU1
Descripción	El actor interesado accede a la legislación establecida por la Universidad de Valladolid
Actores Principales	Usuario Normal
Actores Secundarios	Administrador
Precondiciones	El actor debe haber accedido a la aplicación a través de un navegador.
Flujo Principal	<div><div>1.</div><div>2.</div><div>3.</div><div>4.</div><div>5.</div></div> <div>El caso de uso comienza cuando el actor selecciona “Consultar Legislación” en la página de inicio. El sistema accede a la página donde se encuentra toda la legislación vigente y la muestra en forma de listado. El actor selecciona un documento de la legislación dentro del conjunto mostrado por el sistema. El sistema muestra el documento seleccionado. El caso de uso finaliza.</div>
Postcondiciones	El sistema ha mostrado al usuario la legislación existente seleccionada.
Flujos Alternativos	<div><div>Volver</div><div>3.</div><div>4.</div></div> <div>El actor selecciona “Volver”. El sistema vuelve a la página inicial y el caso de uso finaliza.</div>

Caso de Uso: Descargar Documentación	
ID	CU2
Descripción	El actor consulta la documentación existente en el sistema y puede descargársela para sus trámites.
Actores Principales	Usuario normal
Actores Secundarios	
Precondiciones	El actor debe haber accedido a la aplicación a través de un navegador.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor selecciona “Descargar Documentación”, en la página de inicio. 2. El sistema muestra al actor toda la documentación presente en el sistema dividida por secciones y en diferentes idiomas. 3. El actor selecciona un documento. 4. El sistema muestra el documento seleccionado por el usuario. 5. El actor selecciona con <i>click</i> derecho sobre el documento y selecciona descargar. 6. El sistema descarga el documento y muestra un mensaje de éxito. 7. El caso de uso finaliza.
Postcondiciones	El actor ha descargado la documentación solicitada.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none"> 3. El actor selecciona “Volver”. 4. El sistema vuelve a la página inicial y el caso de uso finaliza. <p>Solo consulta</p> <ol style="list-style-type: none"> 5. El actor selecciona “Atrás”. 6. Paso 2

Caso de Uso: Buscar Convenio	
ID	CU3
Descripción	El actor quiere consultar un convenio buscándolo entre todos los disponibles en el sistema.
Actores Principales	Usuario Normal
Actores Secundarios	Administrador
Precondiciones	Si el actor es el administrador, deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Buscar Convenio”.2. El sistema muestra los campos para restringir la búsqueda.<ol style="list-style-type: none">2.1. Si el actor es Administrador, el sistema mostrará los campos de País, Universidad, Actividad Tipo, Coordinador UVa y Centro UVa.2.2. Si el actor es Usuario Normal, el sistema mostrará los campos de País, Universidad y Actividad Tipo.3. El actor rellena los campos que crea oportunos y selecciona “Buscar”4. El sistema muestra los convenios que coincidan con los campos rellenos.5. El caso de uso finaliza.
Postcondiciones	El sistema ha mostrado al actor el resultado de la búsqueda.
Flujos Alternativos	<p>No se encontró ningún convenio</p> <ol style="list-style-type: none">4. El sistema no encuentra ningún resultado y muestra un mensaje de “Convenios no encontrados”.5. El actor selecciona “Volver” y el caso de uso finaliza. <p>No se introducen campos de búsqueda</p> <ol style="list-style-type: none">3. El actor no rellena ningún campo de búsqueda.4. El sistema muestra al usuario el listado completo de Convenios.5. Paso 4. <p>Volver</p> <ol style="list-style-type: none">3. El actor selecciona “Volver”.4. El sistema vuelve a la página inicial y el caso de uso finaliza.

Caso de Uso: Crear Convenio	
ID	CU4
Descripción	El actor da de alta un nuevo convenio
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor accede a los trámites de convenios y selecciona “Crear Nuevo Convenio”. 2. El sistema recoge toda la información disponible sobre las entidades relacionadas y muestra los campos para la creación del convenio. 3. El actor rellena los campos específicos del convenio y las especificaciones del trámite, después selecciona “Guardar Convenio”. 4. El sistema verifica que los campos introducidos son los necesarios, revisa si son correctos, lo almacena en el sistema y muestra un listado con los convenios en trámite. <ol style="list-style-type: none"> 4.1. Si el actor no ha marcado la opción de “Vigente”, el convenio permanece como convenio en trámite. 4.2. Si el actor ha marcado la opción de “Vigente”, el convenio pasa a estar a disposición de los usuarios universitarios. 5. El caso de uso finaliza.
Postcondiciones	Se registra un nuevo convenio en el sistema.
Flujos Alternativos	<p>Datos erróneos</p> <ol style="list-style-type: none"> 4. El sistema verifica que no se han introducido los campos mínimos y lo notifica al actor. 5. Paso 2. <p>Datos no encontrados</p> <ol style="list-style-type: none"> 3. El actor no dispone de los datos para rellenar el convenio. 4. <<extend>>Añadir Coordinador UVa 5. <<extend>>Añadir Coordinador Exterior 6. <<extend>>Añadir Gestor de Convenios UVa 7. <<extend>>Añadir Gestor de Convenios Exterior <p>Datos disponibles pero incorrectos</p> <ol style="list-style-type: none"> 3. El actor verifica que algún dato de los disponibles para rellenar el convenio son incorrectos. 4. <<extend>>Modificar Coordinador UVa 5. <<extend>>Modificar Coordinador Exterior 6. <<extend>>Modificar Gestor de Convenios UVa 7. <<extend>>Modificar Gestor de Convenios Exterior

Caso de Uso: Editar Convenio	
ID	CU5
Descripción	El actor edita un convenio
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Editar Convenio” en el listado de convenios en trámite.2. El sistema muestra los campos del convenio rellenados anteriormente.3. El actor modifica los datos en los campos y selecciona “Editar”.4. El sistema verifica los campos introducidos y almacena el convenio.<ol style="list-style-type: none">4.1. Si el actor ha marcado el convenio como “En trámite”, el convenio permanece como convenio en trámite.4.2. Si el actor ha marcado el convenio como “Completo”, el convenio pasa a estar a disposición de los usuarios públicos.5. El caso de uso finaliza.
Postcondiciones	Se registran los cambios del convenio en el sistema.
Flujos Alternativos	<p>Datos erróneos</p> <ol style="list-style-type: none">4. El sistema verifica que no se han introducido los campos mínimos y lo notifica al actor.5. Paso 2. <p>Datos no encontrados</p> <ol style="list-style-type: none">3. El actor no dispone de los datos para editar el convenio.4. <<extend>>Añadir Coordinador UVa5. <<extend>>Añadir Coordinador Exterior6. <<extend>>Añadir Gestor de Convenios UVa7. <<extend>>Añadir Gestor de Convenios Exterior <p>Datos disponibles pero incorrectos</p> <ol style="list-style-type: none">3. El actor verifica que algún dato de los disponibles para rellenar el convenio son incorrectos.4. <<extend>>Modificar Coordinador UVa5. <<extend>>Modificar Coordinador Exterior6. <<extend>>Modificar Gestor de Convenios UVa7. <<extend>>Modificar Gestor de Convenios Exterior

Caso de Uso: Renovar Convenio	
ID	CU6
Descripción	El actor procede a dar de alta de nuevo un convenio caducado.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 7. El caso de uso comienza cuando el actor selecciona “Reactivar Convenio” 8. El sistema muestra al actor todos los convenios caducados. 9. El actor selecciona el convenio que desea reactivar. 10. El sistema muestra la información disponible sobre el convenio. 11. El actor realiza las modificaciones necesarias y selecciona “Reactivar”. 12. El caso de uso finaliza.
Postcondiciones	El convenio queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none"> 5. El actor selecciona “Volver”. 6. Paso 2. <p>Datos Erróneos</p> <ol style="list-style-type: none"> 8. El actor introduce algún dato incorrecto en los campos 9. El sistema notifica el error. 10. Paso 4.

Caso de Uso: Añadir documentación	
ID	CU8
Descripción	El actor procede a añadir un documento nuevo en el listado de documentación.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none">1. El caso de uso comienza cuando el actor selecciona “Añadir Documento” en la gestión de la documentación.2. El sistema muestra al actor todos los campos que debe rellenar para añadir el documento.3. El actor rellena los campos del documento.4. El sistema muestra un mensaje de que el documento ha sido añadido con éxito y muestra el listado de todos los documentos subidos.5. El caso de uso finaliza.
Postcondiciones	El documento queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none">3. El actor selecciona “Volver”.4. El caso de uso finaliza. <p>Datos Erróneos</p> <ol style="list-style-type: none">3. El actor introduce algún dato incorrecto en los campos4. El sistema notifica el error.5. Paso 2.

Caso de Uso: Añadir Coordinador UVA	
ID	CU9
Descripción	El actor procede a añadir un nuevo coordinador de la UVa al sistema.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<ol style="list-style-type: none"> 1. El caso de uso comienza cuando el actor selecciona “Añadir Coordinador UVA” en la gestión de los datos. 2. El sistema muestra al actor todos los campos relacionados con el coordinador y los posibles departamentos a los que el coordinador se puede asociar en una ventana emergente. 3. El actor rellena los campos del coordinador. 4. El sistema muestra un mensaje de que el coordinador ha sido añadido con éxito. 5. El caso de uso finaliza.
Postcondiciones	El coordinador queda almacenado.
Flujos Alternativos	<p>Volver</p> <ol style="list-style-type: none"> 3. El actor selecciona “Volver”. 4. El caso de uso finaliza. <p>Datos Erróneos</p> <ol style="list-style-type: none"> 3. El actor introduce algún dato incorrecto en los campos 4. El sistema notifica el error. 5. Paso 2.

Caso de Uso: Añadir Universidad EXT	
ID	CU10
Descripción	El actor procede a dar de alta una nueva universidad en el sistema.
Actores Principales	Administrador
Actores Secundarios	Ninguno
Precondiciones	El actor deberá haberse registrado en el sistema.
Flujo Principal	<div><div>1. El caso de uso comienza cuando el actor selecciona “Añadir Universidad EXT” en la gestión de los datos.</div><div>2. El sistema muestra al actor todos los campos de la universidad en un apartado, y a mayores un apartado para el resto de las entidades a añadir.</div><div>3. El actor rellena los campos relacionados con la universidad.</div><div>4. <<include>> Añadir Rector EXT</div><div>5. <<include>> Añadir Coordinador EXT</div><div>6. <<include>> Añadir Oficina Gestora EXT</div><div>7. <<include>> Añadir Gestor Convenios EXT</div><div>8. El sistema muestra un mensaje de éxito.</div><div>9. El caso de uso finaliza.</div></div>
Postcondiciones	El convenio queda almacenado.
Flujos Alternativos	<div><div>Volver<div><div>3. El actor selecciona “Volver”.</div><div>4. El caso de uso finaliza.</div></div></div><div>Datos Erróneos<div><div>3. El actor introduce algún dato incorrecto en los campos</div><div>4. El sistema notifica el error.</div><div>5. Paso 2.</div></div></div></div>

4.4.9.3. Diagramas de clases de diseño

Antes de mostrar los diagramas de secuencia de los casos de uso vamos a distribuir las clases del sistema a partir de los patrones utilizados en paquetes para su mejor entendimiento.

Vamos a volver a mostrar la vista lógica del sistema y a continuación la distribución de las clases en cada uno de los paquetes.

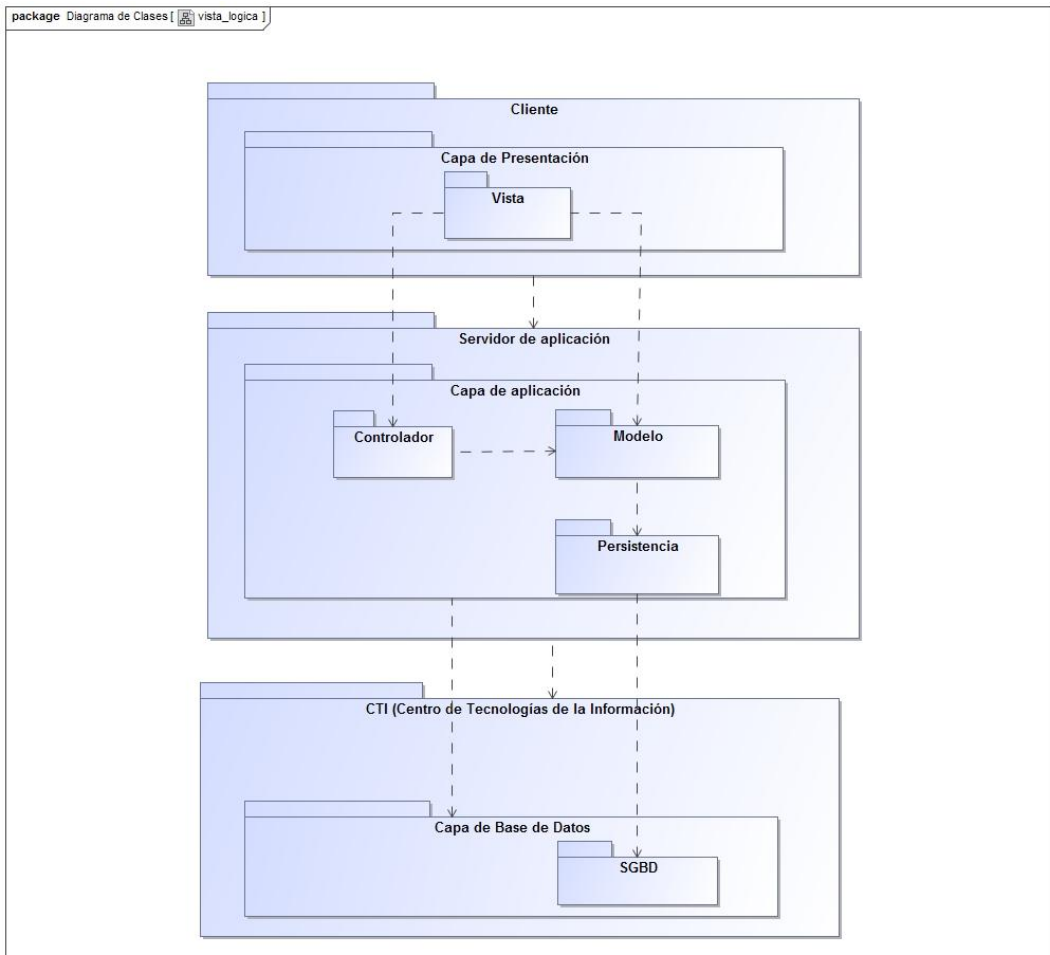


Figura 22: Vista lógica

Nuestro diseño del diagrama de clases se centrará en los dos primeros paquetes o capas, la de presentación y la de aplicación. La capa de persistencia es diseñada e implementada por parte del sistema gestor de base de datos del servidor de base de datos. Añadiremos en un anexo el script necesario para montar la estructura de la base de datos. Esto se lleva a cabo con la ejecución de un sencillo comando en el SGBD.

Capítulo 4: Desarrollo de la aplicación

Paquete de Persistencia en el Servidor de Aplicaciones

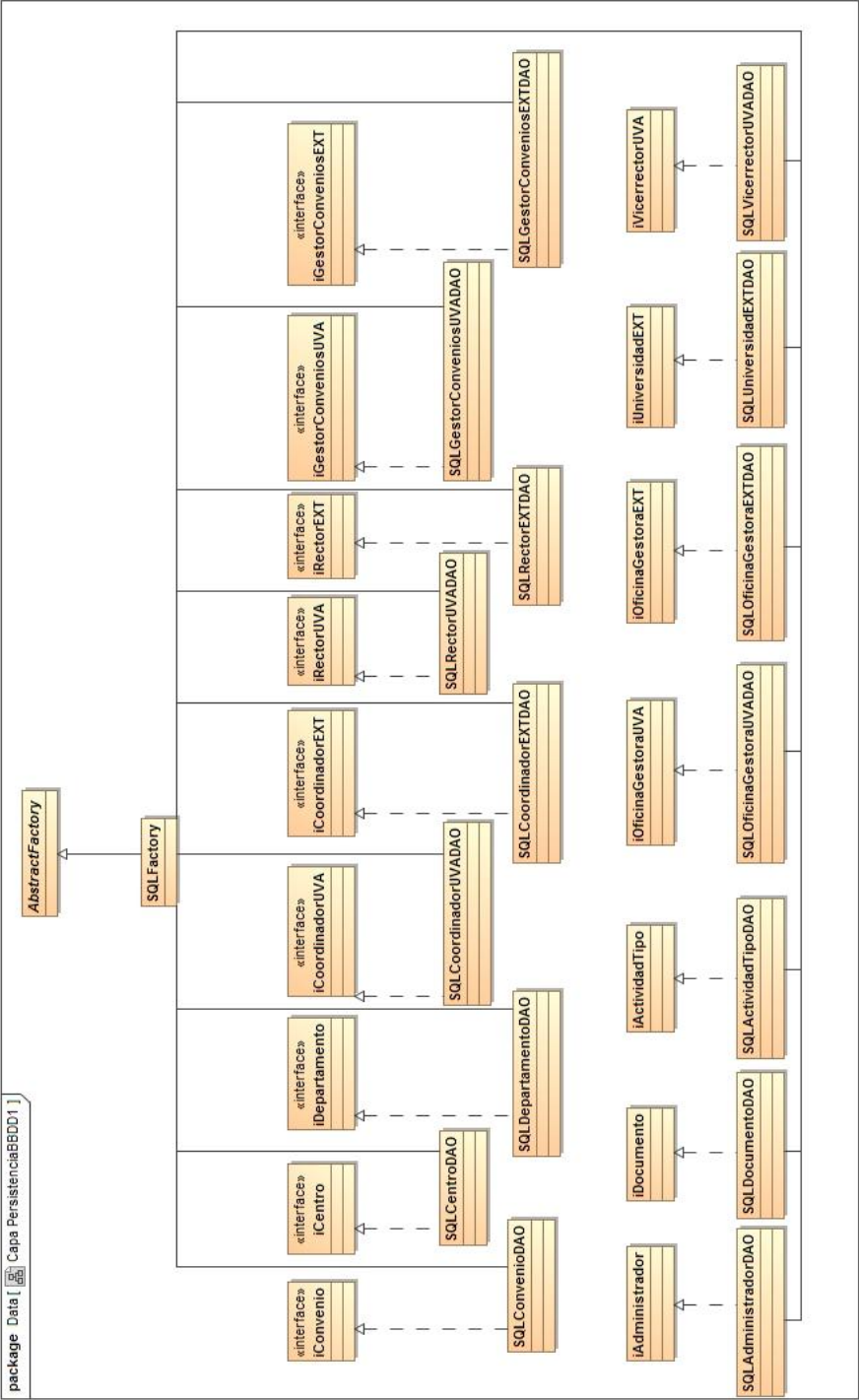


Figura 23: Paquete Persistencia

Paquete de Controlador en el Servidor de Aplicaciones

El paquete controlador está formado a su vez por dos subpaquetes. Por un lado, está el paquete de FrontController que se encarga de la navegación básica del sistema y de recibir sus peticiones síncronas que nos permiten observar los cambios en el sistema.

Por otro lado está el paquete de Controlador AJAX, que se encarga de recibir las peticiones asíncronas del sistema. AJAX es una tecnología que permite llamadas asíncronas a la base de datos sin que la vista del usuario sea recargada. Esta tecnología la explicaremos en el apartado de Implementación.

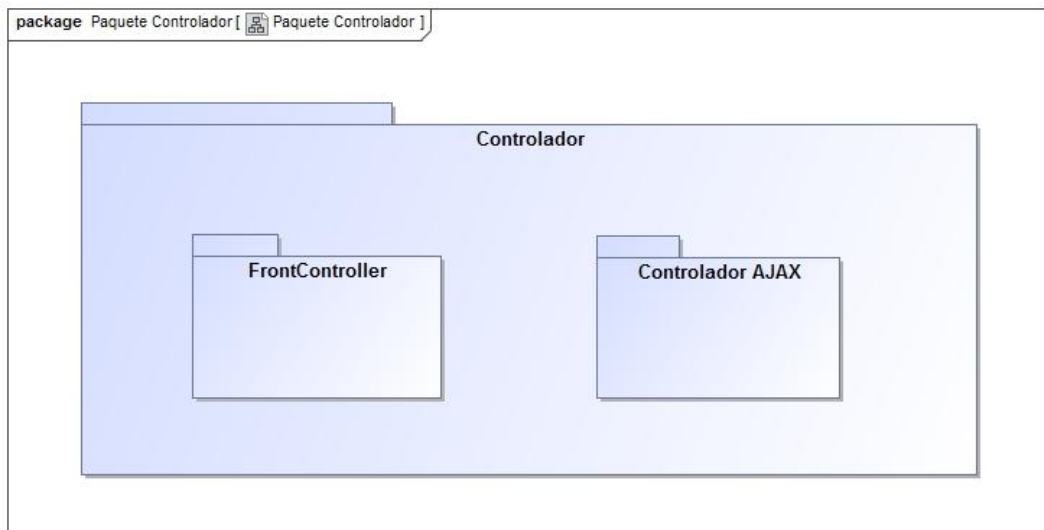


Figura 24: Paquete Controlador

A continuación pasamos a describir los paquetes del Controlador con más detalle.

En primer lugar, el paquete de FrontController encierra el patrón descrito anteriormente. A través de la clase Controlador, y en función de un parámetro, se redirigen las peticiones hacia otros controladores más genéricos y especializados, dejando a la clase Controlador, las funciones de agente de tráfico.

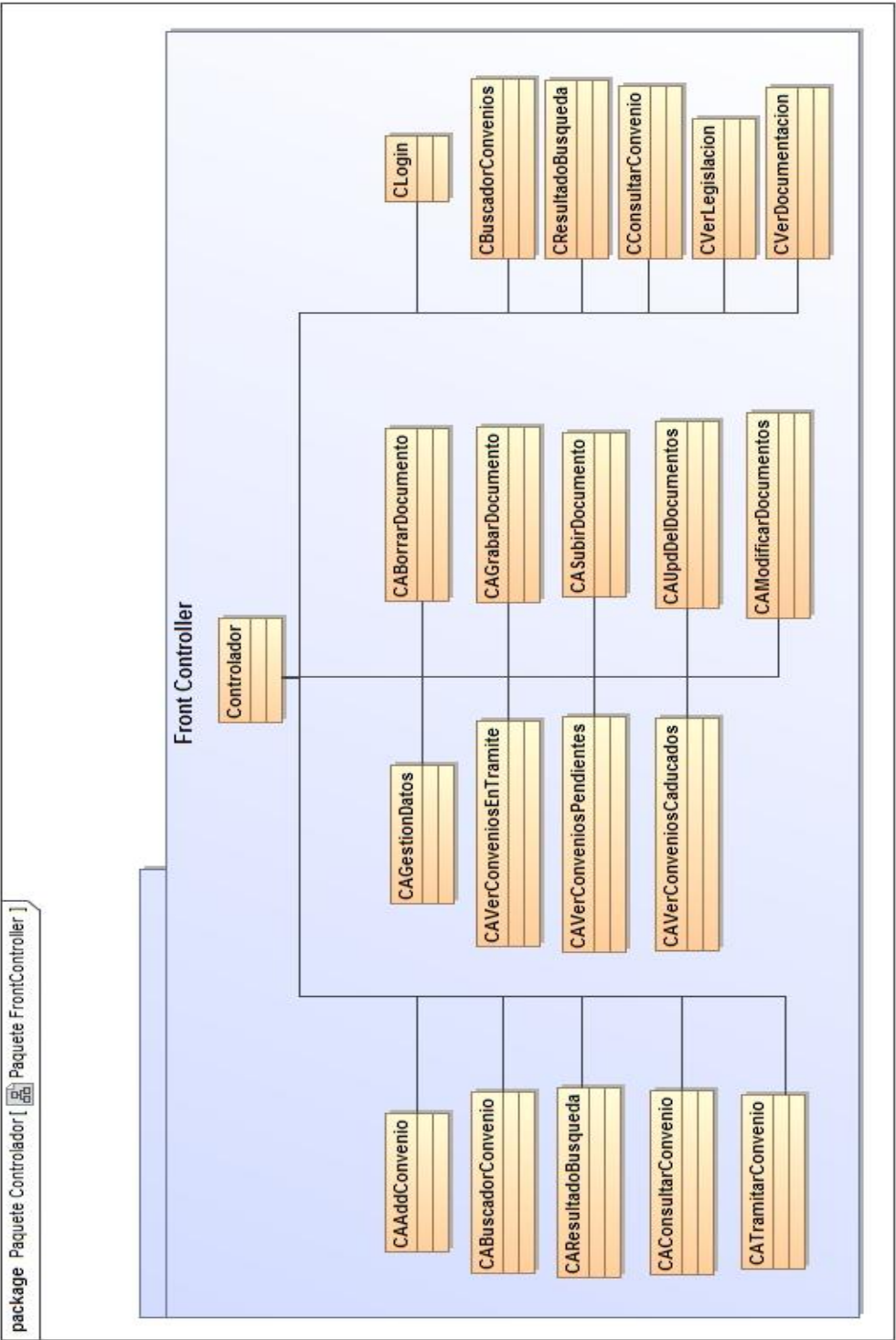


Figura 25: Paquete FrontController

A continuación, el paquete de Controladores AJAX.

Hemos separado estos controladores del FrontController porque son controladores con características y funcionalidades distintas.

Como ya hemos comentado, este paquete recibe llamadas asíncronas que realizan peticiones a la base de datos sobre información del sistema sin que el usuario perciba un cambio de vista.

Hemos centrado la funcionalidad de este paquete en la gestión de las entidades del sistema, como son la creación, modificación y eliminación de coordinadores, rectores, gestores, etc. Debido a que habría que crear una vista para cada una de las acciones relacionadas con las entidades, hemos preferido utilizar esta tecnología en el cliente para mejorar la interacción con el usuario.

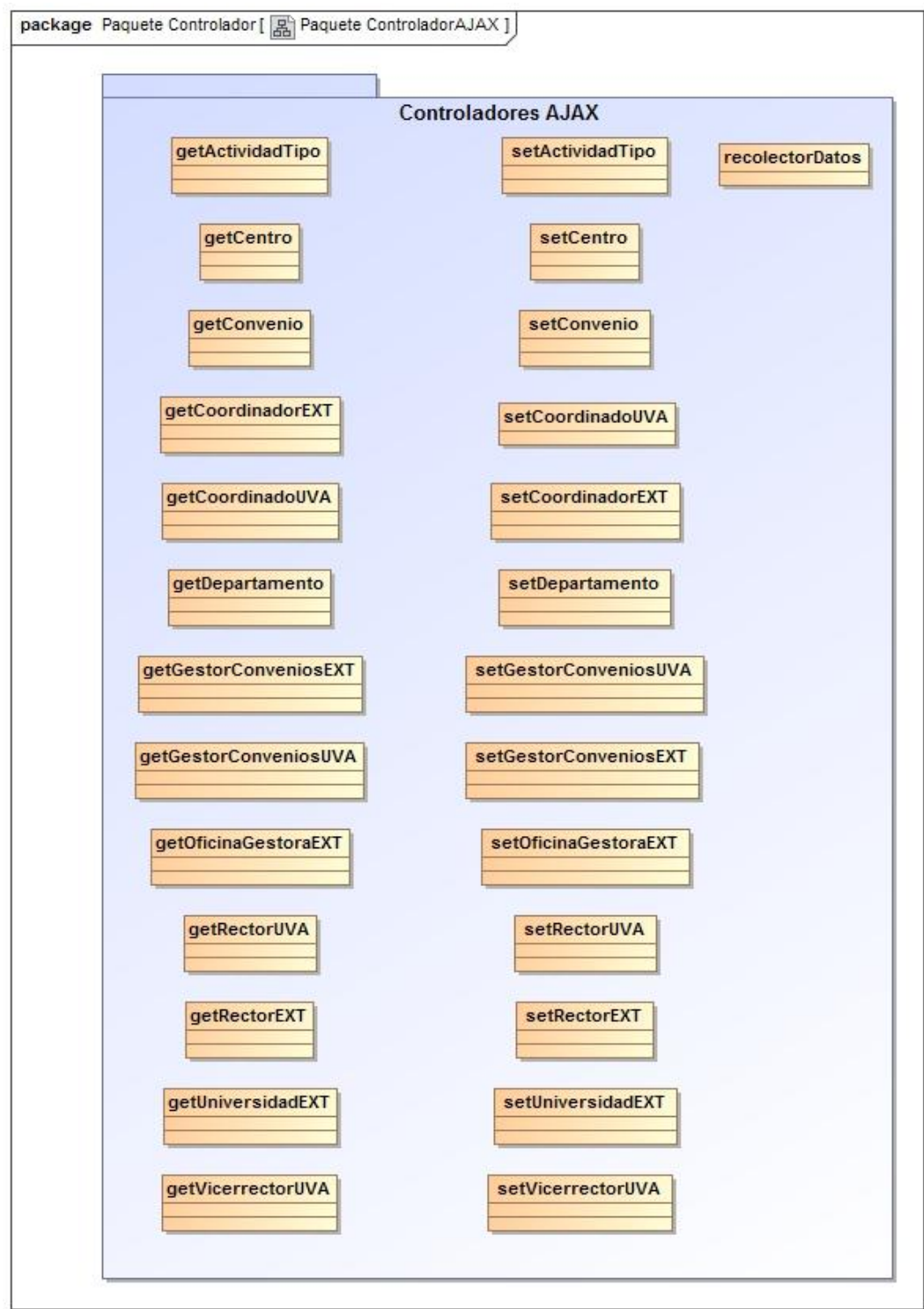


Figura 26: Paquete Controlador AJAX

Paquete de Vista en el Cliente

Al igual que ocurría con el paquete Controlador, en el caso de la vista también hemos separado las vistas en tres subpaquetes. El primero, la Vista Admin, recoge las vistas que tienen relación con las acciones del administrador del sistema. El segundo, la Vista User, tiene agregadas las vistas relacionadas con el usuario. Y por último, hemos separado las vistas relacionadas con las llamadas asincrónicas de AJAX en un paquete a parte. En este paquete no se recogen vistas completas, sino vistas parciales que se añaden a las vistas de los dos primeros para complementar su funcionalidad.

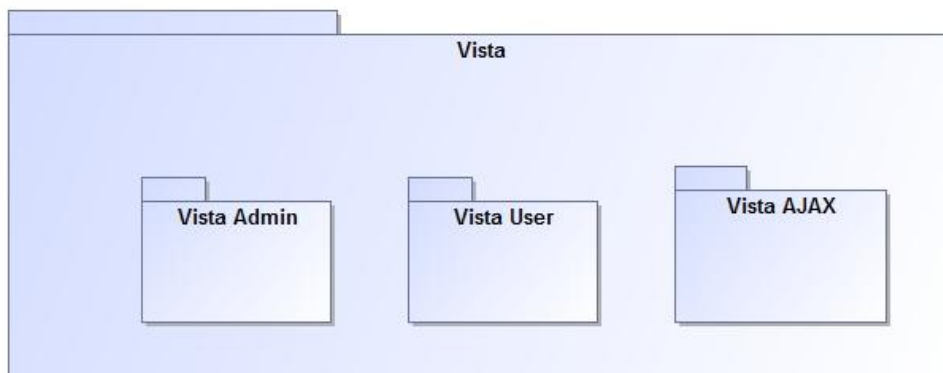


Figura 27: Paquete Vista

Paquete Vista Admin

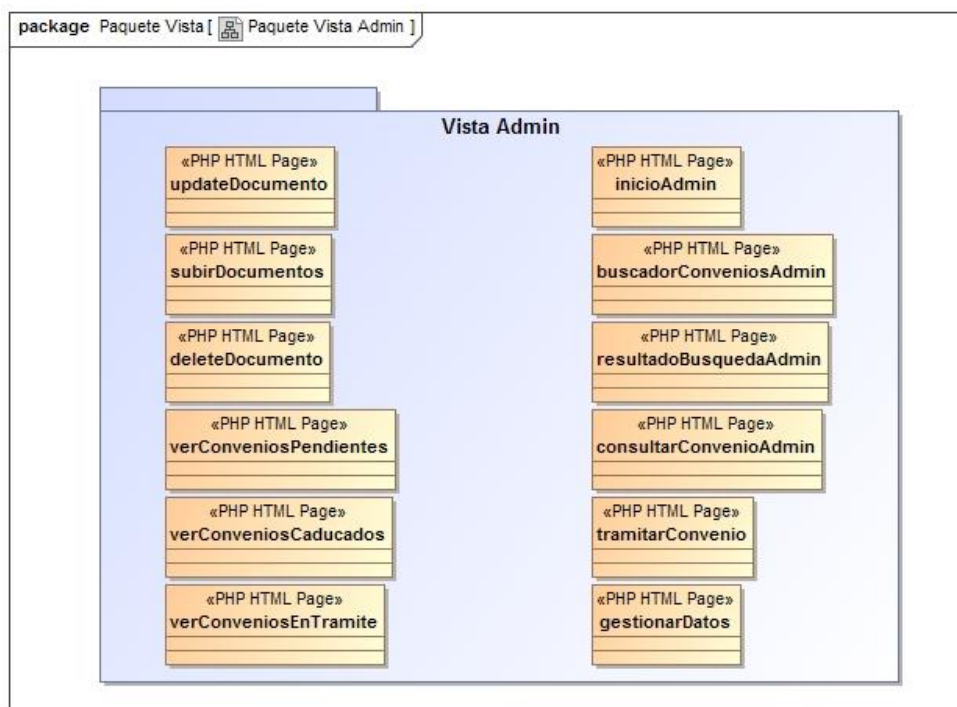


Figura 28: Paquete Vista Admin

Paquete Vista User

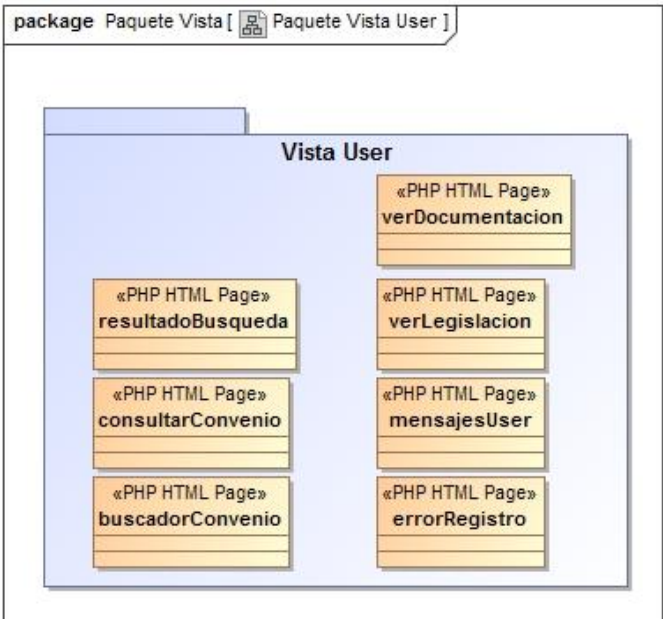


Figura 29: Paquete Vista User

Paquete Vista AJAX

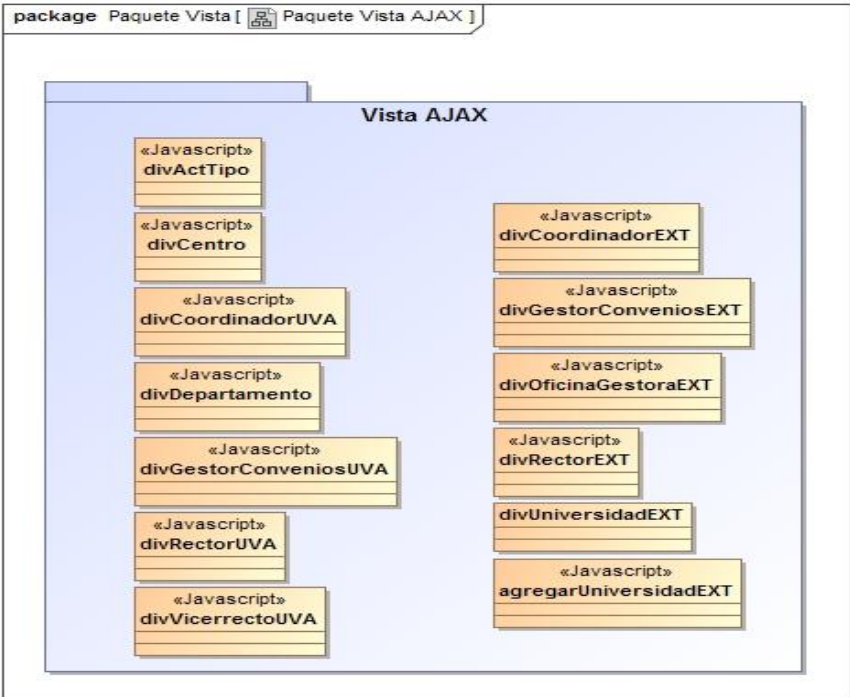


Figura 30: Paquete Vista AJAX

Si se revisan los paquetes de la vista lógica, se notará que falta el paquete de Modelo en el servidor de aplicaciones, pero es que realmente ese paquete de Modelo es el diagrama de clases expuesto al comienzo de este apartado.

Ahora que ya conocemos las clases del sistema, estén en el paquete que esté, podemos realizar los casos de uso mediante los diagramas de secuencia. Éstos van a trazar las interacciones entre las clases ante las interacciones del usuario.

4.4.9.4. Realización de casos de uso

En este apartado vamos a mostrar con mayor detalle la interacción entre el sistema y las acciones del usuario a la hora de llevar a cabo una funcionalidad del mismo. En resumen, es un diagrama que muestra más en detalle el comportamiento del sistema.

Debido a que la extensión de los casos de uso es alta, y que alguno de los casos de uso descritos en diseño son muy parecidos, nos hemos centrado en analizar tan solo unos pocos al detalle.

Como producto de estos diagramas, se anotan las operaciones que se realizan en la cooperación entre clases para completar aún más al detalle los diagramas.

Diagrama de secuencia del caso de uso Ver Legislación

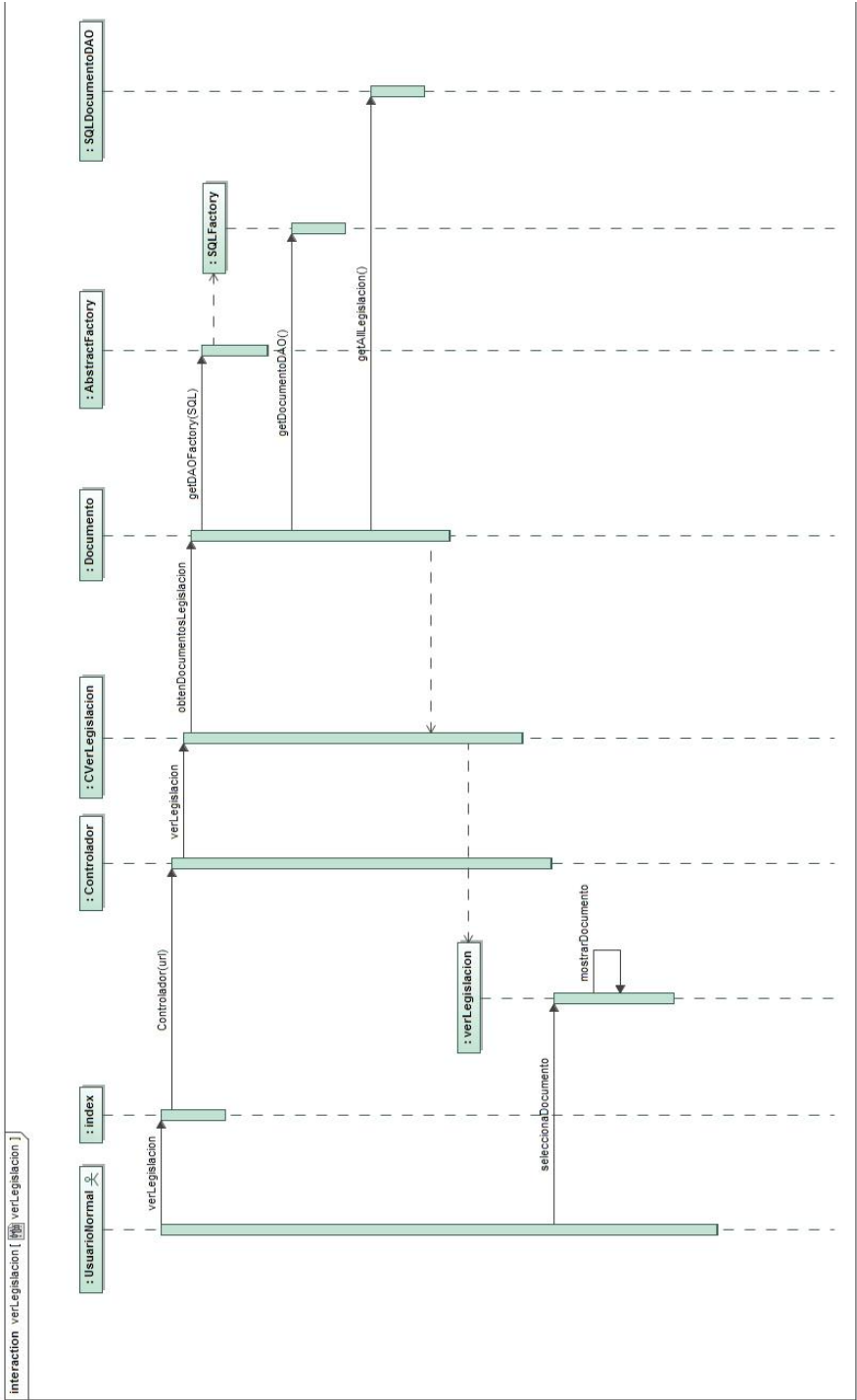


Figura 31: Diagrama de secuencia de Ver Legislacion

Diagrama de secuencia del caso de uso Buscar Convenio

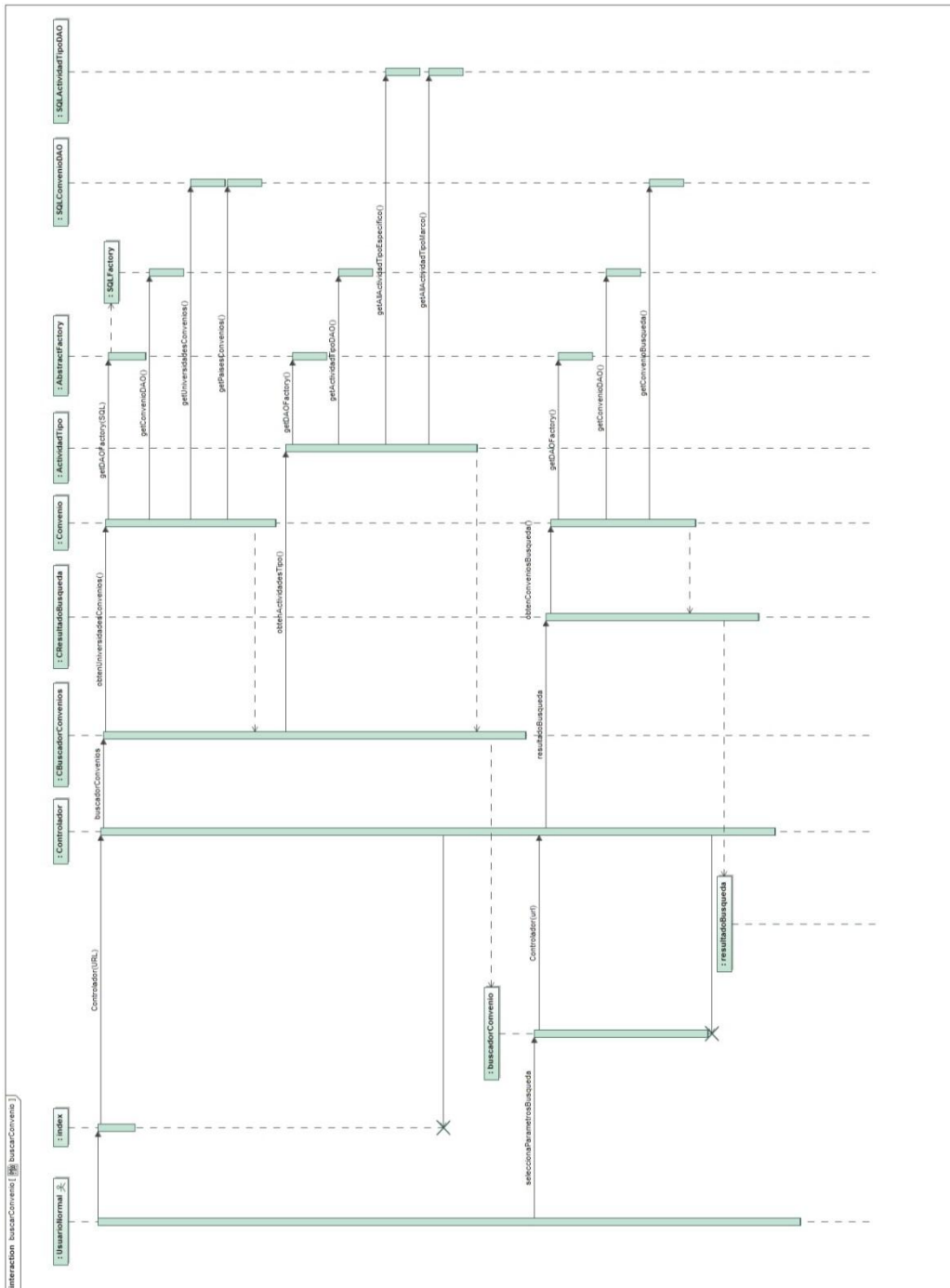


Figura 32: Diagrama de secuencia de Buscar Convenio

Diagrama de secuencia del caso de uso Crear Convenio

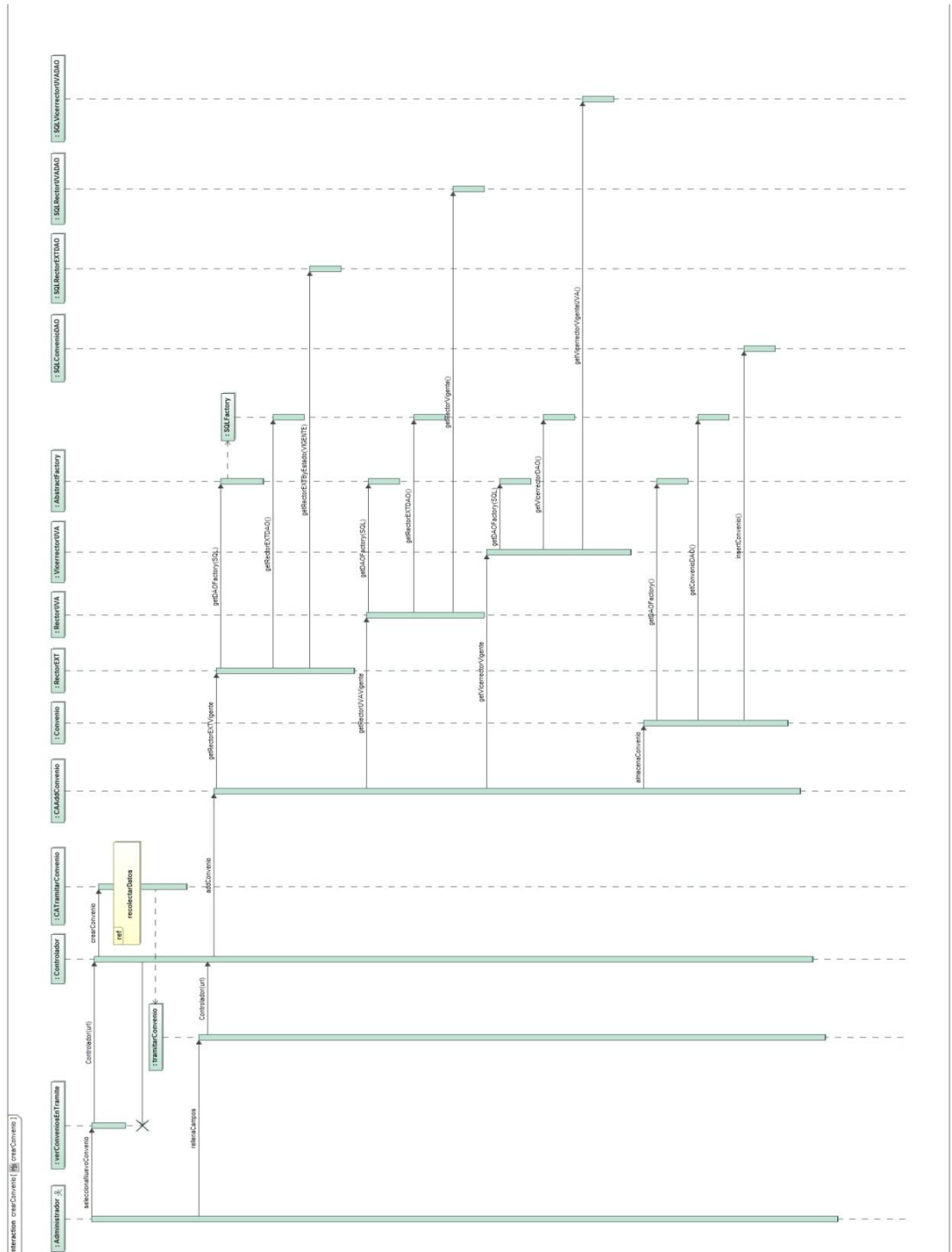


Figura 33: Diagrama de secuencia de Crear Convenio

Diagrama de secuencia de REF Recolectar Datos

<<Debido al tamaño de este diagrama, esta página
ha sido incluida en la carpeta Diagramas del CD-ROM
con el nombre de “Recolectar Datos>>

Diagrama de secuencia del caso de uso Añadir Coordinador UVA

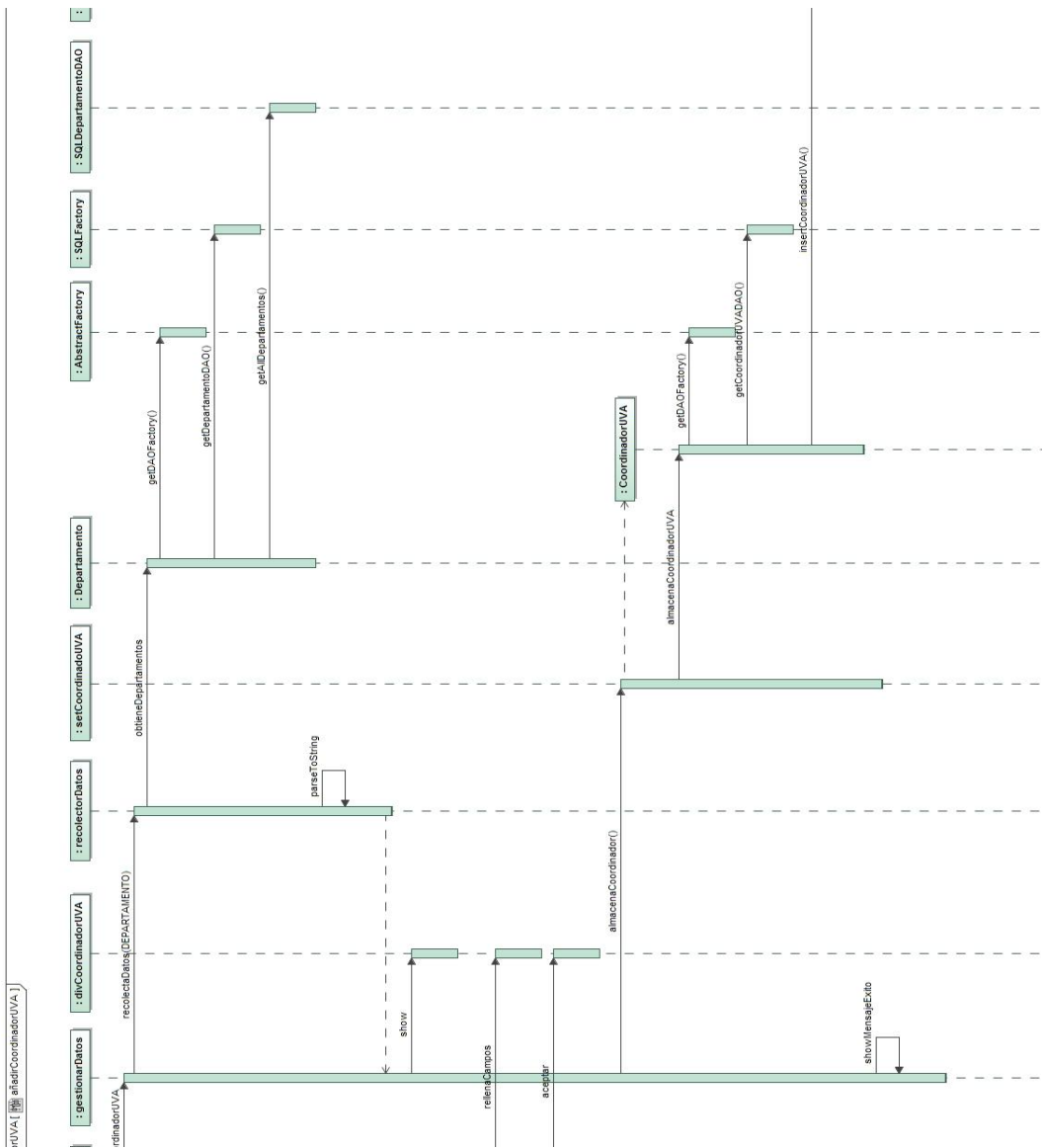


Figura 34: Diagrama de secuencia de Añadir Coordinador UVA

4.4.9.5. Diagramas de clases de diseño (operaciones)

Una vez que ya hemos comprobado las interacciones entre las clases, debemos plasmarlas a través de las operaciones en los diagramas de clases. Al igual que en el apartado anterior, las vamos a separar por paquetes para mantener la estructura del sistema y aclarar al lector.

Diagrama de clases del paquete de Persistencia

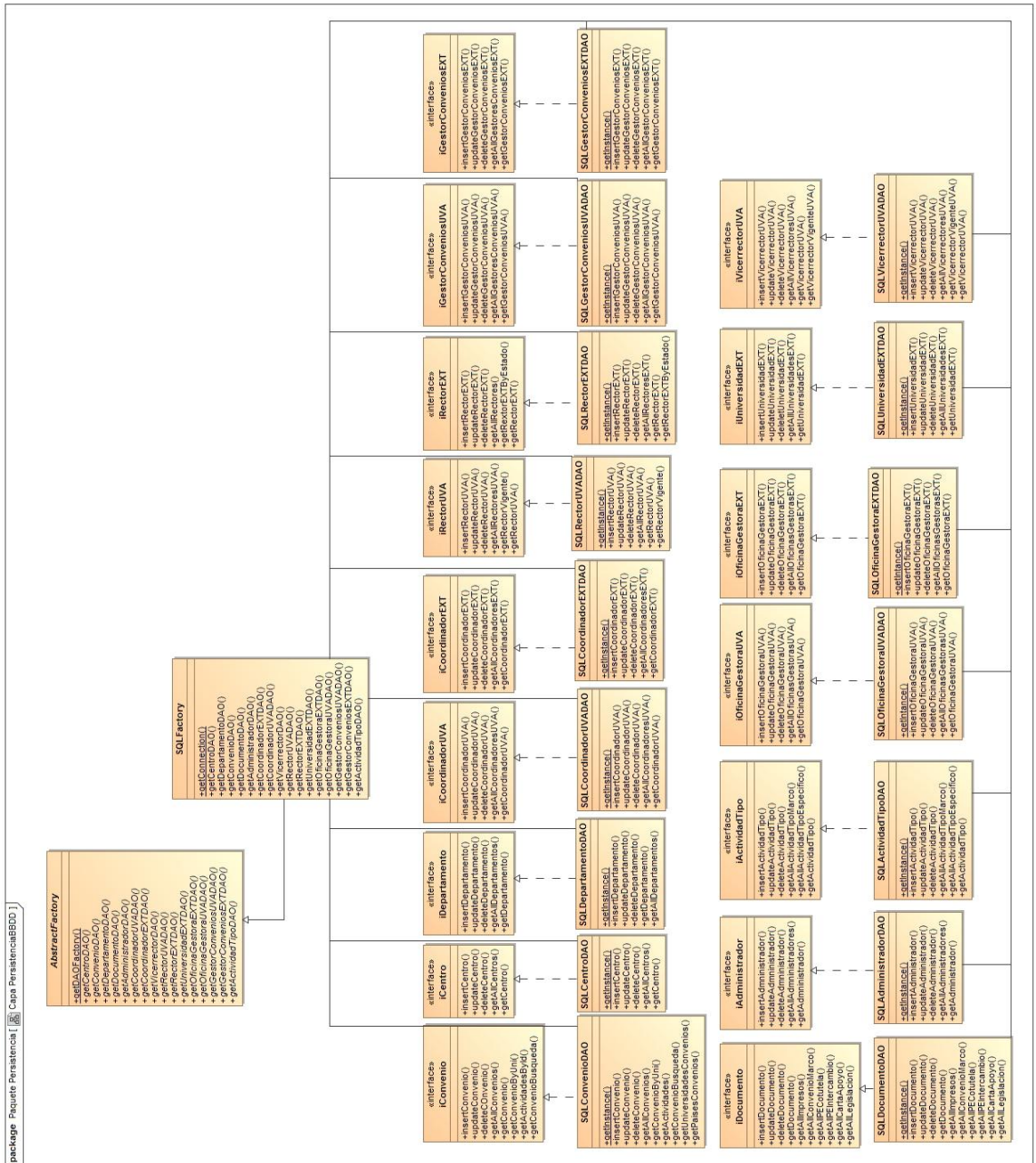


Figura 35: Paquete de Persistecia (operaciones)

Capítulo 4: Desarrollo de la aplicación

Diagrama de clases del paquete de Controlador

Los controladores empleados en el sistema no emplean operaciones como tal. Aunque en los diagramas se representan como llamadas a otras clases mediante operaciones, éstas no existen sino que lo que hacen es que reciben una petición POST/GET de las vistas y el controlador delega en un controlador más específico, que se encarga de reunir y procesar la información desde el modelo.

Por esto, no es necesario añadir los diagramas con operaciones ya que serían repetir los anteriormente mostrados.

Diagrama de clases del paquete Modelo

En el apartado anterior, decidimos no mostrar el diagrama de clases del modelo porque era repetir el diagrama de clases que mostramos al inicio del diseño. Pero ahora que se ha mostrado su interacción con el resto de clases, es necesario representarlo con las nuevas operaciones.

Para no añadir complejidad al diagrama, se omiten los *getters* y los *setters* de cada atributo de la clase.

Sistema web gestor de Convenios Internacionales

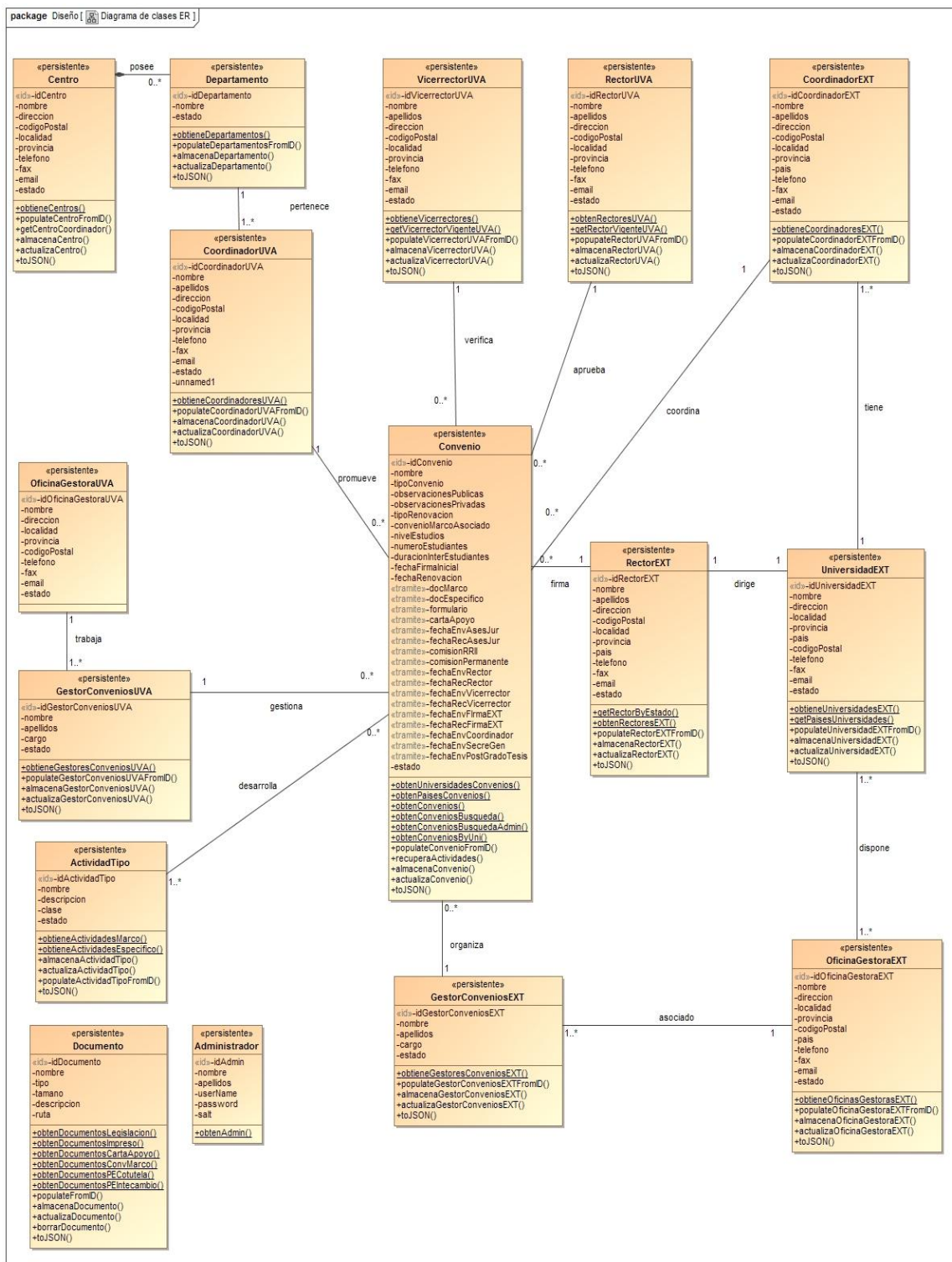


Figura 36: Diagrama de clases ER (operaciones)

Capítulo 4: Desarrollo de la aplicación

Diagramas de clases del paquete de Vista

Las clases que aparecen en estos paquetes son páginas PHP HTML y ficheros Javascript. Realmente no son clases como tal y por ello no poseen ni atributos ni operaciones. Tan solo reciben las acciones de los usuarios y las notifican a los controladores, ya sea en el cliente mediante Javascript y jQuery, o en el servidor PHP.

Por este motivo, creemos que no es necesario repetir los diagramas de la vista mostrados en los anteriores apartados.

4.5. Implementación

En este apartado realizaremos una exposición de los aspectos más importantes a la hora de realizar la implementación del sistema descrito. Todos los diagramas y procesos seguidos hasta ahora se centran en facilitar esta fase, la implementación, ya que al fin y al cabo es el resultado visible de los proyectos y finalmente donde se apoya todo lo que se ven en el funcionamiento del sistema.

Llegados a este punto debemos recoger todo lo extraído de los pasos anteriores (requisitos, casos de uso, análisis y diseño) para poder generar un sistema acorde a todo lo tratado. Al comenzar este apartado lo que más nos va a influir es el diseño, ya que en éste hemos decidido la arquitectura y el comportamiento de nuestro sistema ante las acciones del usuario.

Pero todo esto no quiere decir que dejemos el resto de etapas a un lado, sino que al tratar el desarrollo del software en un proceso iterativo, deberemos tener en cuenta aspectos muy importantes como por ejemplo, los requisitos, que en ningún momento debemos de perder de vista y que se definieron al principio del desarrollo de la aplicación.

Como ya hemos expuesto en los Requisitos del Sistema, la tecnología que emplearemos para llevar a cabo nuestro objetivo será el lenguaje PHP como lenguaje de servidor, HTML y CSS para las vistas y Javascript y jQuery para mejorar la experiencia del usuario y ayudar al comportamiento del sistema. Estas decisiones podían haber sido otras de las muchas que están disponibles para desarrollar sistemas web.

Si miramos hacia el servidor, PHP es la tecnología con una curva de aprendizaje muy rápida, con muchas facilidades en el desarrollo y una gran comunidad en la web, aunque también con muchas libertades, por ello, hay que tener cuidado a la hora de implementar el sistema para no romper los patrones de diseño y la estructura del mismo.

Para entender el funcionamiento del sistema en la implementación, primero es necesario conocer los patrones anteriormente descritos, y el más importante es el MVC.

Con PHP, nos encargamos de todos los paquetes del lado de servidor, como son los Controladores, el Modelo y paquete de acceso a la Persistencia. Digamos que el núcleo del sistema está hecho en PHP.

Sistema web gestor de Convenios Internacionales

Si vamos al cliente la cosa es distinta. La estructura de las vistas viene dada por HTML. Mediante etiquetas, se va estructurando la vista por zonas, como se ha descrito en el apartado del estudio sobre la interfaz de usuario. Con CSS se aplican estilos a dichas etiquetas creando el diseño estético de las vistas.

```
<div id="contenedor">
  <div id="cabecera">
  </div>
  <div id="cuerpo">
    <div id="lado_izquierdo">

    </div>

    <div id="centro">
    </div>

    <div id="lado_derecho">
    </div>
  </div>
  <div id="pie">
  </div>
</div>
```



Figura 37: Vista de la estructura generada por el código

Capítulo 4: Desarrollo de la aplicación

Una vez que ya tenemos la estructura es necesario introducir la información en las vistas y esto tiene lugar gracias a los controladores, que son los encargados de obtener la información del modelo y suministrarlo a las vistas mediante inserciones de código PHP.

Estas inserciones de valores PHP nunca deberán ser porciones de código que ejecuten complejas estructuras, sino exclusivamente valores numéricos o de cadena, o iteraciones sobre colecciones de objetos sencillos. Cuanto más reduzcamos estas inserciones y cuanto más sencillas sean, menor será la dependencia entre Vista y Controlador y entre Vista y Modelo, siendo más flexibles a cambios.

```
<table>
  <tr>
    <th>Nombre Fichero</th>
    <th>Tamaño</th>
    <th></th>
    <th></th>
  </tr>
  <?php
  foreach ($docImpreso as $doc) {
    echo "<tr>";
    echo "<td>";
    echo $doc->getNombre();
    echo "</td>";
    echo "<td>";
    echo ($doc->getTamano() / 1024) . " KBytes";
    echo "</td>";
    echo "<td>";
    echo "</td>";
    echo "</tr>";
  }
  ?>
</table>
```

Como podemos observar en esta sección de código, se realiza una iteración sobre una colección de objetos Documento, y se listan en una tabla.

Con la vista ya formada, los usuarios interaccionan con ésta y en un momento dado, desembocado por un evento del usuario, se lanza una acción. Dicha acción solicitará un cambio de la vista, ya sea mediante un click en un enlace o en un botón de la vista. El controlador frontal recibe el evento y en función de un parámetro de selección delega el control a un controlador más específico.

El controlador frontal, o también llamado *FrontController*, es tan solo una estructura selectiva que carga un nuevo controlador para que realice la acción.

```

$url = $_GET['url']; //parámetro selectivo

switch ($url) {
    case('cbcu'):
        include '../app/Controladores/CBuscadorConvenio.php';
        break;
    case('alu'):
        include '../app/Controladores/CVerLegislacion.php';
        break;
    case('adu'):
        include '../app/Controladores/CVerDocumentacion.php';
        break;
    case('cbcur'):
        include '../app/Controladores/CResultadoBusqueda.php';
        break;
    //////////ADMIN////////////////////////////////////
    case('badmin'):
        include '../app/Controladores/CABuscadorConvenio.php';
        break;
    case('bresadmin'):
        include '../app/Controladores/CARResultadoBusqueda.php';
        break;
    case('tcadmin'):
        include '../app/Controladores/CATramitarConvenio.php';
        break;
    case('gesdataadmin'):
        include '../app/Controladores/CAGestionDatos.php';
        break;
    case('addConv'):
        include '../app/Controladores/CAAddConvenio.php';
        break;
    case('borrardocadmin'):
        include '../app/Controladores/CABorrarDocumento.php';
        break;
    ...
}

```

Con el *include* de PHP se delega la responsabilidad al controlador concreto que obtiene la información del Modelo, y éste a su vez de la base de datos.

Capítulo 4: Desarrollo de la aplicación

```
<?php

$ConveniosMarco = Array();
$ConveniosEspecificos = Array();

//obtenemos los convenios en trámite de la base de datos a través del modelo
$convenios = Convenio::obtenConveniosEstado(TRAMITE);

foreach ($convenios as $c) {
    $aux = Array();

    array_push($aux, $c->getIDConvenio());
    array_push($aux, $c->getNombre());

    //recuperamos los datos relacionados con el convenio
    $coordinadorEXT = new CoordinadorEXT();
    $coordinadorEXT->populateCoordEXTFromID($c->getIDCoordEXT());
    array_push($aux, $coordinadorEXT->getIDCoordinadorEXT());
    array_push($aux, $coordinadorEXT->getNombre());

    $universidad = new UniversidadEXT();
    $universidad->populateUniversidadFromID($coordinadorEXT->getIDUniversidadEXT());
    array_push($aux, $universidad->getIDUniversidadEXT());
    array_push($aux, $universidad->getNombre());

    $coordinadorUVA = new CoordinadorUVA();
    $coordinadorUVA->populateCoordUVAFromID($c->getIDCoordUVA());
    array_push($aux, $coordinadorUVA->getIDCoordinadorUVA());

    array_push($aux, $coordinadorUVA->getNombre());
    array_push($aux, $c->getEstado());

    if (($c->getTipoConvenio()) == "Marco") {
        array_push($ConveniosMarco, $aux);
    } else {
        array_push($ConveniosEspecificos, $aux);
    }
}
include '../pub/admin/verConveniosEnTramites.php';
?>
```

Una vez que hemos obtenido la información para generar la nueva vista, la incluimos para que de esta manera tenga acceso a los datos obtenidos en el modelo. Al igual que mostramos en la inserción de código PHP, en esta vista se tendrá acceso a un listado de objetos convenio listos para listarlos, sin que la vista tenga que hacer computación.

Sistema web gestor de Convenios Internacionales

Esta navegación para los casos normales encaja perfectamente. Pero existe un caso en el sistema donde esta navegación habría creado una gran cantidad de vistas parciales y controladores de las mismas, hablamos de la gestión de los datos.

Existen en el sistema entre 12 y 14 entidades. Pues bien para cada una de ellas habría que haber creado una vista de creación, otra de modificación y otra de eliminación, y todas ellas con sus correspondientes controladores y pantallas mostrando un mensaje de éxito o fracaso al llevar a cabo la acción.

Ante este problema surgen las tecnologías de clientes ligeros como Javascript y jQuery que permiten dar versatilidad a las vistas. Estas vistas y controladores están en los paquetes de Vistas AJAX y Controladores AJAX expuestos en el diseño.

Mediante estas dos tecnologías del cliente se ejecutan las llamadas AJAX, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas.

Por otro lado, jQuery y Javascript permiten recoger los eventos del usuario y mostrar y ocultar campos, zonas, información, etc.

Estas tecnologías nos vienen perfectas para nuestro caso. El hecho es que en el momento de seleccionar, por ejemplo, “Añadir un coordinador UVA”, mediante jQuery se muestra el formulario de creación por encima de la pantalla donde estábamos, permitiéndonos acceder tan solo a ese formulario y haciendo inaccesible la parte oculta. Además, y de manera invisible al usuario y mediante AJAX, se ha solicitado al servidor toda la información sobre los departamentos a los que se puede añadir a este coordinador. Esto se debe a que todo coordinador de la UVA pertenece a un departamento en concreto, por ello a la hora de añadirlo, es necesario relacionarlo con un departamento, que de esta manera ya nos aparece para ser seleccionado.

En el caso de la modificación y eliminación, el usuario debe seleccionar qué coordinador va a modificar/borrar del listado de coordinadores del sistema. Pues bien, cuando el usuario selecciona “Modificar/Borrar coordinador UVA”, al igual que en el caso anterior, jQuery oculta la vista base y muestra el formulario, precargando mediante AJAX el listado de todos los coordinadores. Al seleccionar uno de ellos, como ya tenemos todos sus datos presentes, aunque ocultos, tan solo hay que rellenar los campos gracias a Javascript y jQuery.

Las llamadas AJAX también mantienen la estructura del sistema, siendo las vistas los formularios que aparecen y se desvanecen, y los controladores son los encargados de solicitar la información en segundo plano, como ya hemos dicho, mostrados en los diagramas de Vista y Controlador AJAX.

Capítulo 4: Desarrollo de la aplicación

El fichero que recibe la acción del usuario sería el siguiente:

```
$('#addCoordUVA').click(function(){  
  
    reiniciaCamposCoordUVA();  
    $('#anadirCoordUVA').show();  
    $('#modificarCoordUVA').hide();  
    $('#eliminarCoordUVA').hide();  
    $('#coordinadorUVADIV').hide();  
    $('#fondoNegro').fadeIn('slow');  
    $('#contenidoCoordUVA').fadeIn('slow');  
  
});
```

Lo que hacemos es que cuando se haga click sobre el elemento HTML con id 'addCoordUVA' se muestre la vista con el formulario del coordinador, se establezca un fondo negro para evitar que el usuario realice acciones sobre el formulario de atrás y se muestran los botones de acción.

Con esta tecnología se consigue un alto grado de reutilización y permite un avance rápido en el desarrollo.

The image shows a web application interface. At the top, there's a header with 'Relaciones Internacionales' and a 'Desconexión' button. A sidebar on the left contains 'Opciones Administrador' and a list of menu items. The main content area features a modal form titled 'Datos Coordinador UVA:'. The form is divided into two columns. The left column lists labels: 'Nombre:', 'Apellidos:', 'Dirección Completa:', 'Código Postal:', 'Localidad:', 'Provincia:', 'Teléfono:', 'Fáx:', 'Correo electrónico:', and 'Departamento:'. The right column contains corresponding input fields: text boxes for the first eight labels, and a dropdown menu for 'Departamento:'. At the bottom of the modal, there are two buttons: 'Añadir' and 'Volver'.

Figura 38: Formulario emergente generado por jQuery y Javascript

Sistema web gestor de Convenios Internacionales

Ésta sería la composición de un controlador AJAX que solicita la información de todos los coordinadores de la UVA:

```
$coordinadoresUVA = CoordinadorUVA::obtieneCoordinadoresUVA();
$txt = "";
for ($i = 0; $i < count($coordinadoresUVA); $i++) {
    $txt = $txt .
        $coordinadoresUVA[$i]->getIDCoordinadorUVA() . "," .
        $coordinadoresUVA[$i]->getNombre() . " " .
        $coordinadoresUVA[$i]->getApellidos() . "," .
        $coordinadoresUVA[$i]->getIDDepartamento();
    $txt = $txt . " * ";
}
echo $txt;
```

Recoge la información a través del modelo y la parsea en una cadena para que la vista la pueda utilizar y mostrarla la vista en un selector.

Una vez que el usuario rellena todos los campos, seleccionará “Añadir”, la vista del formulario de creación desaparece y se recupera la visión de la gestión de los datos.

Para el usuario no ha ocurrido nada, tan solo un mensaje de éxito que le notifica la acción, pero no ha sufrido una recarga de la página, ni ésta se ha modificado. Para el usuario todo es transparente.

Lo que realmente ha ocurrido es que mediante jQuery hemos recogido los valores de los campos rellenos por el usuario. Los hemos añadido a una llamada AJAX y hemos solicitado a un controlador que cree un nuevo coordinador con los campos que le hemos pasado.

Código jQuery:

```
id = $('#idCoordUVA').val();
nombre = $('#nombreCoordUVA').val();
apell = $('#appCoordUVA').val();
direccion = $('#direccionCoordUVA').val();
cp = $('#cpCoordUVA').val();
localidad = $('#localidadCoordUVA').val();
provincia = $('#provinciaCoordUVA').val();
tel = $('#telCoordUVA').val();
fax = $('#faxCoordUVA').val();
email = $('#emailCoordUVA').val();
idDep = $('#depCoordUVA').val();
```

Capítulo 4: Desarrollo de la aplicación

Una vez tenemos los campos, creamos la llamada AJAX:

```
$.ajax({  
    url: "../pub/scripts/setCoordUVA.php", //ruta del controlador  
    data: {  
        id : id,  
        nombre : nombre,  
        apellidos : apell,  
        dir : direccion,  
        cp : cp,  
        local : localidad,  
        prov : provincia,  
        tel : tel,  
        fax : fax,  
        email : email,  
        estado : estado,  
        idDep : idDep  
    },  
    success: function(dato){  
        alert("Coordinador de la UVa añadido con éxito.")  
    }  
});
```

En el servidor está el controlador AJAX, que recibe esta solicitud de creación. Toma los parámetros y define un nuevo coordinador.

```
$coordUVA = new CoordinadorUVA();  
$coordUVA->setNombre($nombre);  
$coordUVA->setApellidos($apellidos);  
$coordUVA->setDireccion($direccion);  
$coordUVA->setCodPostal($cp);  
$coordUVA->setLocalidad($localidad);  
$coordUVA->setProvincia($provincia);  
$coordUVA->setTelefono($tel);  
$coordUVA->setFax($fax);  
$coordUVA->setEmail($email);  
$coordUVA->setEstado($estado);  
$coordUVA->setIDDepartamento($idDep);  
  
$coordUVA->almacenaCoordinadorUVA();
```


Sistema web gestor de Convenios Internacionales

Hasta aquí hemos concretado la implementación del MVC, ahora nos falta ver cómo se implementa el último paso en la cadena del funcionamiento del sistema. El modelo almacena los datos en la base de datos.

Gracias a los patrones en la capa de persistencia descritos en el diseño, veremos que esta acción es un hecho relativamente sencillo una vez que se ha montado y se entiende el patrón.

En un primer caso, el modelo necesita el DAO, implementado en tecnología MySQL para poder realizar el acceso a los datos. Para ello necesita la factoría que crea esos DAO concretos:

```
//conseguimos la factoría concreta que crea DAOs implementados en MySQL

$fact = AbstractFactory::getDAOFactory(SQL);

//con la factoria concreta, solicitamos en DAO que nos interesa

$coordUVADAO = $fact->getCoordinadorUVADAO();
```

Ahora que tenemos el DAO ya podemos realizar el acceso a la información de la base de datos, y por otro lado, acceso a la manipulación de las tablas, como por ejemplo:

```
$coordUVADAO->insertCoordinadorUVA($this);
```

Si comprobamos el código en el DAO:

```
public function insertCoordinadorUVA(CoordinadorUVA $CoordinadorUVA) {
    //conseguimos la conexion remota a la base de datos gracias a la factoria
    $conexion = SQLFactory::getConexion();

    //creamos la consulta que vamos a ejecutar
    $insert = sprintf(
        "INSERT INTO `coordinadoruva_new`(`IDCoordUVA`, `Nombre`, ...)
        VALUES (',%s',...);",
        $CoordinadorUVA->getNombre(), $CoordinadorUVA->getApellidos(), ...);

    //ejecutamos la consulta
    $query = mysql_query($insert, $conexion);

    //devolvemos el id del Nuevo coordinador si todo ha ido bien
    if ($query)
        $id = mysql_insert_id(); // Recoge la id del coordinador insertado.
    else
        die(mysql_error());
    mysql_close($conexion);

    return $id;
}
```

Capítulo 4: Desarrollo de la aplicación

Analizando el código, obtenemos la conexión con la base de datos, generamos la consulta para obtener la información y ejecutamos la consulta.

Tan solo hay que reproducir estas líneas con cada consulta que queramos realizar y obtendremos toda la información que necesitemos.

4.6. Pruebas

Tras haber realizado la implementación del proyecto, pasamos a explicar qué tipo de pruebas hemos realizado para lograr hacer de nuestro sistema una aplicación a prueba de errores.

Las pruebas realizadas y documentadas en este documento han sido realizadas en función de los casos de uso que se han diseñado para el sistema. Se realizarán pruebas para cada uno de los casos de uso y se realizará una evaluación de los resultados obtenidos, dependiendo a la manera en que afecte al sistema, los datos almacenados y la interfaz de la aplicación.

En cada caso de prueba se especificará la funcionalidad que afecta al resultado de la prueba, así como la salida de los datos obtenida y si corresponde con la información que se esperaba conseguir.

Existen algunos casos de prueba que no serán documentados, ya que se trata de casos con pequeñas variaciones, como el hecho de añadir una entidad al sistema o añadir otra.

Los Casos de Prueba que resultaron fallidos, se volvieron a repetir posteriormente de realizar la corrección de fallos. Los errores detectados y su posterior corrección o solución están reflejados en este documento.

Se han desarrollado en dos fases, una en un servidor local de desarrollo, y otra en el servidor real del Servicio de Relaciones Internacionales. Las primeras han sido realizadas por el desarrollador y las segundas por parte del personal de la oficina de Relaciones Internacionales.

Tanto en local como en el servidor real, es necesario probar cada caso de uso por separado.

Pruebas en local

CP01	Ver legislación correctamente
Objetivo	Comprobar que el sistema permite visualizar la documentación de la legislación correctamente
Prueba	Acceso a la sección de legislación desde el menú inicial
Salida Estimada	Listado de la documentación de legislación.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP02	Descargar documento correctamente
Objetivo	Comprobar que el sistema permite descargar un documento correctamente
Prueba	Seleccionar un documento en la sección de documentación.
Salida Estimada	Descargar el documento seleccionado
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP03	Inicio de sesión correcto
Objetivo	Comprobar que el sistema permite el acceso a un usuario registrado en el mismo.
Prueba	Introducir Nombre de usuario y contraseña correctamente
Salida Estimada	El sistema permite el acceso al administrador
Salida obtenida	Salida estimada
Resultado	Acción realizada correctamente

Capítulo 4: Desarrollo de la aplicación

CP04	Inicio de sesión incorrecto
Objetivo	Comprobar que el sistema no permite el acceso a un administrador con datos incorrectos
Prueba	Introducir un nombre de usuario y/o contraseña incorrecto/s
Salida Estimada	El sistema impide el acceso del administrador al sistema y lo notifica mediante un mensaje.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP05	Buscar convenio
Objetivo	Comprobar que el sistema muestra un listado de convenios existentes ante una búsqueda.
Prueba	Introducir unos parámetros de búsqueda
Salida Estimada	Listado con los convenios clasificados por país y universidad
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP06	Buscar convenio inexistente
Objetivo	Comprobar que el sistema muestra un mensaje informando al usuario de que no existen convenios con los criterios de búsqueda introducidos.
Prueba	Introducir parámetros de búsqueda de convenio inexistente
Salida Estimada	Se muestra un mensaje expresando la inexistencia de convenios con esos parámetros
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP07	Crear un nuevo convenio en trámite
Objetivo	Comprobar que el sistema permite añadir un nuevo convenio
Prueba	Completar el formulario de los datos del convenio para la creación, dejando sin seleccionar el campo “vigente”.
Salida Estimada	Se muestra un mensaje de éxito y un listado con todos los convenios en trámite.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP08	Crear un convenio vigente
Objetivo	Comprobar que el sistema permite añadir convenios vigentes
Prueba	Completar el formulario de los datos del convenio para la creación, completando el campo “vigente”.
Salida Estimada	Se muestra un mensaje de éxito y un listado de los convenios en trámite. (Se comprueba que se ha añadido con éxito, buscándolo en el buscador).
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP09	Renovar convenio pendiente
Objetivo	Comprobar que el sistema permite establecer un convenio pendiente a vigente/en trámite.
Prueba	Se selecciona un convenio pendiente y se rellenan los nuevos campos del convenio
Salida Estimada	Se muestra un mensaje de éxito y un listado con los convenios en trámite.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

Capítulo 4: Desarrollo de la aplicación

CP10	Reactivar convenio caducado
Objetivo	Comprobar que el sistema permite establecer un convenio caducado a vigente/en tramite
Prueba	Se selecciona un convenio caducado y se rellenan los campos nuevos del convenio
Salida Estimada	Se muestra un mensaje de éxito y un listado con los convenios en trámite
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP11	Añadir coordinador UVA
Objetivo	Comprobar que el sistema permite añadir un nuevo coordinador UVA
Prueba	Se rellenan los campos del formulario de coordinadores UVA
Salida Estimada	Se muestra un mensaje de éxito.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP12	Modificar coordinador UVA
Objetivo	Comprobar que el sistema permite añadir un nuevo coordinador UVA
Prueba	Se modifican los campos del formulario de coordinadores UVA
Salida Estimada	Se muestra un mensaje de éxito
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP13	Eliminar coordinador UVA
Objetivo	Comprobar que el sistema permite eliminar un nuevo coordinador UVA
Prueba	Se selecciona un coordinador a eliminar de un listado de todo los disponibles.
Salida Estimada	Se muestra un mensaje de éxito.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

Pruebas en real

CP01	Ver legislación correctamente
Objetivo	Comprobar que el sistema permite visualizar la documentación de la legislación correctamente
Prueba	Acceso a la sección de legislación desde el menú inicial
Salida Estimada	Listado de la documentación de legislación.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP02	Descargar documento correctamente
Objetivo	Comprobar que el sistema permite descargar un documento correctamente
Prueba	Seleccionar un documento en la sección de documentación.
Salida Estimada	Descargar el documento seleccionado
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP03	Inicio de sesión correcto
Objetivo	Comprobar que el sistema permite el acceso a un usuario registrado en el mismo.
Prueba	Introducir Nombre de usuario y contraseña correctamente
Salida Estimada	El sistema permite el acceso al administrador
Salida obtenida	Salida estimada
Resultado	Acción realizada correctamente

Capítulo 4: Desarrollo de la aplicación

CP04	Inicio de sesión incorrecto
Objetivo	Comprobar que el sistema no permite el acceso a un administrador con datos incorrectos
Prueba	Introducir un nombre de usuario y/o contraseña incorrecto/s
Salida Estimada	El sistema impide el acceso del administrador al sistema y lo notifica mediante un mensaje.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP05	Buscar convenio
Objetivo	Comprobar que el sistema muestra un listado de convenios existentes ante una búsqueda.
Prueba	Introducir unos parámetros de búsqueda
Salida Estimada	Listado con los convenios clasificados por país y universidad
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP06	Buscar convenio inexistente
Objetivo	Comprobar que el sistema muestra un mensaje informando al usuario de que no existen convenios con los criterios de búsqueda introducidos.
Prueba	Introducir parámetros de búsqueda de convenio inexistente
Salida Estimada	Se muestra un mensaje expresando la inexistencia de convenios con esos parámetros
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP07	Crear un nuevo convenio en trámite
Objetivo	Comprobar que el sistema permite añadir un nuevo convenio
Prueba	Completar el formulario de los datos del convenio para la creación, dejando sin seleccionar el campo “vigente”.
Salida Estimada	Se muestra un mensaje de éxito y un listado con todos los convenios en trámite.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP08	Crear un convenio vigente
Objetivo	Comprobar que el sistema permite añadir convenios vigentes
Prueba	Completar el formulario de los datos del convenio para la creación, completando el campo “vigente”.
Salida Estimada	Se muestra un mensaje de éxito y un listado de los convenios en trámite. (Se comprueba que se ha añadido con éxito, buscándolo en el buscador).
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP09	Renovar convenio pendiente
Objetivo	Comprobar que el sistema permite establecer un convenio pendiente a vigente/en trámite.
Prueba	Se selecciona un convenio pendiente y se rellenan los nuevos campos del convenio
Salida Estimada	Se muestra un mensaje de éxito y un listado con los convenios en trámite.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

Capítulo 4: Desarrollo de la aplicación

CP10	Reactivar convenio caducado
Objetivo	Comprobar que el sistema permite establecer un convenio caducado a vigente/en tramite
Prueba	Se selecciona un convenio caducado y se rellenan los campos nuevos del convenio
Salida Estimada	Se muestra un mensaje de éxito y un listado con los convenios en trámite
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP11	Añadir coordinador UVA
Objetivo	Comprobar que el sistema permite añadir un nuevo coordinador UVA
Prueba	Se rellenan los campos del formulario de coordinadores UVA
Salida Estimada	Se muestra un mensaje de éxito.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP12	Modificar coordinador UVA
Objetivo	Comprobar que el sistema permite añadir un nuevo coordinador UVA
Prueba	Se modifican los campos del formulario de coordinadores UVA
Salida Estimada	Se muestra un mensaje de éxito
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

CP13	Eliminar coordinador UVA
Objetivo	Comprobar que el sistema permite eliminar un nuevo coordinador UVA
Prueba	Se selecciona un coordinador a eliminar de un listado de todo los disponibles.
Salida Estimada	Se muestra un mensaje de éxito.
Salida obtenida	Salida estimada
Resultado	Acción realizada con éxito

4.7. Despliegue

El servicio de Relaciones Internacionales de la UVa cuenta con la ayuda de un departamento técnico encargado de los temas informáticos, asociado a la Universidad de Valladolid, el STIC.

Ellos son los encargados del mantenimiento del material informático de este departamento, así como su software y el conjunto de webs relacionadas. Este servicio es el encargado de dar soporte a la base de datos y a las aplicaciones web, por este motivo, debemos solicitarles que nos permitan el alojamiento en el servidor, acceso a la base de datos (usuario y contraseña) y un repositorio de ficheros al que tener acceso desde la aplicación.

Antes de subir ningún fichero, deberemos conocer el servidor de base de datos, su IP o URL, el nombre de la base de datos asociada al sistema y el nombre y contraseña que están ligados a dicha base de datos. Con esta información, deberemos acceder a la clase `SQLFactory.php`, que es la encargada de suministrar la conexión al sistema e introducir los datos.

```
public static function getConnection() {  
  
    $host = 'localhost';           //URL o IP de alojamiento de BBDD  
  
    $nameBD = 'relint';           //nombre de la base de datos  
  
    $nameUser = 'relint';         //usuario base de datos  
  
    $passUser = '';               //contraseña base de datos  
  
    $conexion = mysql_connect($host, $nameUser, $passUser);  
  
    if (!$conexion) {  
  
        die('Error en conexion BD' . mysql_error());  
  
    }  
  
    mysql_select_db($nameBD, $conexion);  
  
    return $conexion;  
  
}
```

Se cambian los datos establecido para pruebas en local, con los valores suministrados por el STIC.

Capítulo 4: Desarrollo de la aplicación

Una vez montada la conexión de la base de datos está realizada y conocemos la dirección de alojamiento de la aplicación, copiamos la estructura de carpetas del proyecto en el servidor mediante FTP. Con el acceso disponible al servidor de la base de datos, utilizamos un sistema gestor de bases de datos y ejecutamos el script asociado a esta memoria con la estructura de tablas del sistema.

Capítulo 5

Manuales

5. Manuales

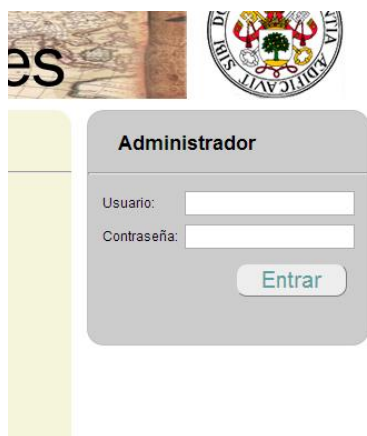
En este apartado, daremos instrucciones de cómo usar la aplicación para lograr conseguir los objetivos del sistema. Se intentará dar una descripción detallada de la funcionalidad del sistema, buscando la utilidad de los gráficos para favorecer el aprendizaje del usuario.

5.1. Manual de usuario

Este manual se centra en la parte de administrador. Para la parte de usuario normal, es ilógico incluir un manual, ya que lo que se lleva buscando en todo el desarrollo del sistema es que la intuitividad del usuario favorezca el éxito a la hora de realizar las tareas en la aplicación. Además debemos tener en cuenta que por parte de un usuario normal, la funcionalidad es bastante reducida.

Volviendo al administrador, vamos a dividir las funcionalidades de la zona privada e iremos describiendo cada una de ellas.

5.1.1. Login de administrador



En primer lugar, el administrador debe registrarse mediante un nombre de usuario y una contraseña almacenadas en el sistema para poder acceder a la zona privada del sistema.

El formulario de registro se encuentra en la parte derecha de la pantalla, permitiendo que las miradas de los usuarios se fijen en la parte central, con el contenido importante, y dejando a un lado dicho formulario.

5.1.2. Menú inicial

Una vez nos registramos con éxito, nos aparece un menú de inicio para el administrador, donde se pone a nuestra disposición la funcionalidad del sistema.

Sistema web gestor de Convenios Internacionales

En una barra lateral izquierda, dejamos los enlaces a las diferentes partes del sistema, y en un único bloque central, eliminando la columna derecha, establecemos el contenido.



Además, como podemos observar en la imagen, se nos muestran los avisos si en el momento de activar la sesión, se han añadido nuevos convenios al conjunto de los pendientes o de los caducados.

Desde el primer momento en el que entramos al sistema, ya estamos avisados de que se han producido cambios de interés en el sistema, y debemos tenerlo en cuenta.

5.1.3. Buscador de convenios

El buscador de convenios es una herramienta muy útil para los administradores. Debido a la gran cantidad de convenios que puede llegar a manejar este sistema, sería una tarea muy tediosa, tener que ir buscando uno a uno los convenios hasta encontrar el que deseásemos.

En un primer momento, se establecen hasta cinco parámetros de búsqueda:

- **País:** país de la universidad con la que realizamos el convenios.
- **Universidad:** universidad con la que hacemos el convenio.
- **Actividad tipo:** actividad tipo que se establece en el convenio.
- **Coordinador UVA:** Personal de la UVA que se encarga de gestionar el convenio.
- **Centro UVA:** Centro al que pertenece el coordinador del convenio.

Estas cinco características son restricciones que establecemos a la hora de filtrar el listado de los convenios. Esto es, para eliminar aquellos convenios que no nos interesen.



Relaciones Internacionales



Desconexión

Opciones Administrador

- Buscador de Convenios
- Trámites de Convenios
- Convenios Pendientes
- Convenios Caducados
- Gestión de Datos
- Subir documentación

Buscador de Convenios

País:

--País--

Universidad:

--Universidad--

Actividad Tipo:

----Actividad Tipo----

Coordinador UVA:

--Coordinador UVA--

Centro UVA:

--Centro UVA--

Buscar



Una vez introducidos los parámetros de búsqueda, seleccionamos el botón de Buscar y nos aparecerá un listado con los convenios que concuerden con lo requerido. De no encontrarse ningún convenio acorde a lo establecido, se mostrará un mensaje que así lo indica.

Francia

Universidad de Paris



Convenios Vigentes:



Convenios Marco:

Convenio Marco Total  

1 convenio(s)

Convenios Especificos:

Convenio Total Especifico  

Convenio Total Especifico ren 1  



2 convenio(s)

Convenios No Vigentes:

Convenios Marco:

0 convenio(s)

Convenios Especificos:

Convenio Total Especifico  

1 convenio(s)



Sistema web gestor de Convenios Internacionales

Si por otro lado, el administrador no desea introducir ningún parámetro de búsqueda, el sistema mostrará un listado completo de los convenios almacenados.

Se recuerda que tan solo aparecerán aquellos convenios que estén Vigentes, Pendientes o Caducados, dejando fuera a aquellos que estén en trámite, ya que todavía no están aprobados.

El listado de los convenios, en el caso de haber obtenido algún resultado, se muestra dividido en primer lugar por el país de la universidad, después por las universidades de ese país, y dentro de cada universidad, están separados por convenios vigentes y caducados. Por último, dentro de cada uno de ellos, los convenios están separados por convenios Marco y Específicos.

Cada elemento del listado permite su consulta y edición:

- **Consulta:** seleccionamos el nombre del convenio y se nos muestra una ventana con la información del convenio.
 - Dentro de este listado podemos seleccionar cada uno de los elementos
seleccionado en la lupa () para consultar el elemento en mayor detalle.
- **Edición:** seleccionamos () para acceder al modo edición del convenio seleccionado.











Consulta de un convenio

Resultado de la Búsqueda:

[Volver](#)

Información acerca del Convenio

Datos del Convenio:

Nombre del convenio	Convenio Total Especifico II	
Universidad Extranjera	Universidad de Paris	
Coordinador UVA	Laura Polo	
Departamento UVA	Departamento de Ingeniería de software	
Coordinador Univ. Ext	Frederic Vernoulli	
Gestor del Convenio UVA	Fatima Arévalo	
Gestor del Convenio Univ. Ext.	ArielA Ben-Kanaan	
Oficina Gestora de Convenios Univ. Ext.	Relations Intenationelle	
Rector UVA	Fernando Sacristan	
Vicerrector UVA	el Vicerrector	
Rector Extranjero	Sophie Dupont	
Tipo de Convenio	Especifico	
Convenio Marco asociado:	Convenio Marco Total	
Tipo de Renovación	Expresa	
Fecha Expiración	24-01-2014	

Actividades Tipo relacionadas con el Convenio:

Actividad Tipo para Convenio Especifico:

Intercambio de estudiantes

Intercambio de Estudiantes:

Detalle de un elemento del convenio

Buscador de

Trámites de

Convenios

Convenios

Gestión de

Subir docu

Datos Coordinador EXT:

Nombre:	Frederic
Apellidos:	Vernoulli
Dirección Completa:	La ru
Código Postal:	453343
Localidad:	Paris
Provincia:	Paris
País:	Francia
Teléfono:	9858249234882349
Fáx:	ao92349923940293
Correo electrónico:	ifvn@gmk.fr

Volver

5.1.4. Trámites de convenios

En este apartado se mantienen aquellos convenios que están en proceso de aprobación para pasar a estar vigentes y accesibles para la comunidad universitaria. Desde este apartado también podemos crear convenios nuevos que, dependiendo de la información de la que dispongamos, pasaran a ser vigentes o en trámite.

Los convenios aparecen separados entre convenios marco y específicos. En la parte superior derecha podremos seleccionar “Nuevo convenio” para acceder a un formulario vacío.

Tanto si creamos uno nuevo como si editamos uno existente, accederemos a la misma pantalla, tan solo cambiará el hecho de que algunos campos aparezcan seleccionados o no.

Sistema web gestor de Convenios Internacionales

En primer lugar nos aparece el listado de los convenios y el botón para crear uno nuevo.



Desconexión

Opciones Administrador

Buscador de Convenios

Trámites de Convenios

Convenios Pendientes

Convenios Caducados

Gestión de Datos

Subir documentación

Convenios en Trámite

+ Nuevo Convenio

Convenios Marco

Nombre	Universidad Extranjera	Coordinador UVA	Coordinador Exterior	Estado
marco-caducado	Universidad de Estocolmo	Agatha Silva	Sven Hukke	En trámite

Convenios Específicos

Nombre	Universidad Extranjera	Coordinador UVA	Coordinador Exterior	Estado
convenio para pruebas-tramite	Universidad de Estocolmo	Agatha Silva	Sven Hukke	En trámite
convenio para pruebas-tramite	Universidad de Estocolmo	Agatha Silva	Sven Hukke	En trámite
prueba-reactivar caducado	Universidad de Estocolmo	Laura Polo	Sven Hukke	En trámite

Servicio de Relaciones Internacionales - Universidad de Valladolid

Casa del Estudiante - Calle Real de Burgos, s/n

47011 Valladolid España

+34 983 423 719 +34 983 185 845 mariafatima.arevalo@uva.es http://www.relint.uva.es/relint/convenios/index.asp

Y si seleccionamos un convenio nuevo o uno ya existente, obtendremos:

Atrás Desconexión

Trámite de Convenios

Datos del Convenio:

Nombre del convenio

Universidad Extranjera

País

Coordinador UVA

Coordinador Univ. Ext

Gestor del Convenio (UVA)

Gestor del Convenio (Univ. Ext.)

Tipo de Convenio

Tipo de Renovación

--Universidad--

--Coordinador UVA--

--Coordinador Universidad Ext--

--Gestor de Convenios UVA--

--Gestor de Convenios Ext--

☐ Marco

☐ Específico

☐ Automática

☐ Expresa

Actividades Tipo relacionadas con el Convenio:

Atrás Desconexión

Trámite de Convenios

Datos del Convenio:

Nombre del convenio

Universidad Extranjera

País

Coordinador UVA

Coordinador Univ. Ext

Gestor del Convenio (UVA)

Gestor del Convenio (Univ. Ext.)

Tipo de Convenio

Tipo de Renovación

marco-caducado

Universidad de Estocolmo

Suecia

Agatha Silva

Sven Hukke

Fatima Aráozvala

Amandine Phonseque

☒ Marco

☐ Específico

☐ Automática

☒ Expresa






Años de vigencia: 3



Actividades Tipo relacionadas con el Convenio:

131

Capítulo 5: Manuales

Una vez que nos encontramos en el formulario del convenio, puede darse que nos falte algún dato en el sistema y no podamos completar el convenio. Para no tener que estar dando pasos atrás e insertarlo en otro menú, se da la opción de poder añadir coordinadores y gestores de convenios de la UVA y de la universidad extranjera, desde el menú del convenio. De esta manera nos ahorramos navegación innecesaria.

Coordinador UVA	Agatha Silva	 
Coordinador Univ. Ext	Sven Hukke	 
Gestor del Convenio (UVA)	Fatima Arevalo	 
Gestor del Convenio (Univ. Ext.)	Amandine Phonseque	 

Al seleccionar cualquiera de los iconos, tanto para añadir como para editar ( , ), aparece un formulario emergente que nos ahorra el redirigirnos a otra pantalla para rellenar los datos y confirmarlos.

Opciones Admin

Datos Coordinador UVA:

Nombre:

Apellidos:

Dirección Completa:

Código Postal:

Localidad:

Provincia:

Teléfono:

Fáx:

Correo electrónico:

Departamento:

--Departamento--

Añadir

Volver

Se rellenan los datos y se añade el nuevo elemento al sistema, pudiendo acceder a él para completar los datos del convenio.

A la hora de rellenar los convenios, es necesario rellenar un número mínimo de campos que corresponden con la información básica del sistema, que se corresponden con lo que aparecen en la imagen:

Trámite de Convenios

Datos del Convenio:

Nombre del convenio	marco-caducado
Universidad Extranjera	Universidad de Estocolmo
País	Suecia
Coordinador UVA	Agatha Silva
Coordinador Univ. Ext	Sven Hukke
Gestor del Convenio (UVA)	Fatima ArÃ©valo
Gestor del Convenio (Univ. Ext.)	Amandine Phonseque
Tipo de Convenio	<input checked="" type="radio"/> Marco <input type="radio"/> Específico
Tipo de Renovación	<input type="radio"/> Automática <input checked="" type="radio"/> Expresa
Años de vigencia: 3	

Estos datos son la base del convenio, sin ellos, no se puede inicializar un convenio, ya que de no disponer de esta información, no es necesario que figure aún en el sistema. El siguiente paso es la selección de las actividades tipo relacionadas con el convenio. Éstas dependerán de si el convenio es Marco o Específico.

El siguiente apartado relacionado con las fechas de entrega a los diferentes organismos encargados de firmar un convenio, viene explicada en el capítulo de Marco Conceptual par aque sirve cada una de ellas. Nos limitamos a mostrar el método de introducción de las fechas para facilitar su rápido cumplimentado.

Capítulo 5: Manuales

Trámites del Convenio:

Documento de Marco

Documento Entregado:

☐

Documento Especifico

Documento Entregado:

☐

Formulario

Documento Entregado:

☐

Carta de Apoyo

Documento Entregado:

☐

Envío Asesor Jurídico / Comisión Doctorado:

Enviado:

Recibido:

June 2013

July 2013

Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
					1	2	1	2	3	4	5	6	7
3	4	5	6	7	8	9	8	9	10	11	12	13	14
10	11	12	13	14	15	16	15	16	17	18	19	20	21
17	18	19	20	21	22	23	22	23	24	25	26	27	28
24	25	26	27	28	29	30	29	30	31				

Comisión RRH:

Entregado:

Comisión Permanente:

Entregado:

Envío Firma Rector:

Enviado:

Recibido:

Envío Firma Vicerrector:

Enviado:

Recibido:

Envío Firmas Extranjeras:

Enviado:

Recibido:

Envío Coordinador:

Entregado:

Envío a Secretario General:

Entregado:

Una vez hayamos cumplimentado los campos de los que dispongamos, podemos seleccionar su fecha de entrada en vigor y, si el administrador lo cree oportuno, dar de alta el convenio.

5.1.5. Convenios caducados y pendientes

Aunque en el sistema aparecen como dos apartados distintos, en el manual las explicaremos en conjunto debido a su similitud.

Haciendo memoria, un convenio pendiente es aquel al que le queda un periodo inferior o igual a seis meses para su caducidad. Por otro lado, un convenio caducado es aquel cuya fecha de renovación ha expirado, por lo que se encuentra fuera de vigor.

Tanto para unos como para otros, en su apartado correspondiente, se encuentran listados los convenios que tienen estas características. Será desde este punto desde donde el administrador se encargara de reactivarlos o de renovarlos.

Para ello, seleccionará el botón con la acción correspondiente colocado a la derecha de la descripción de cada convenio, y éste será reconducido al formulario de edición de los convenios, con las mismas características que si se tratase de la tramitación de uno nuevo.

Si el convenio que se renueva o reactiva es un convenio marco, el convenio anterior se sobrescribe con la nueva información, pero con el caso de los específicos, el anterior convenio se deja caducar y se crea uno nuevo copiando los datos que el administrador encuentre oportuno.



UvA Relaciones Internacionales

Desconexión

Opciones Administrador

- Buscador de Convenios
- Trámites de Convenios
- Convenios Pendientes
- Convenios Caducados
- Gestión de Datos
- Subir documentación

Convenios Caducados

Convenios Caducados

Nombre	Universidad Extranjera	Coordinador UvA	Coordinador Exterior	F.Renovación	
Convenio por el buen trabajo	Otago Polytechnic	Eusebio Huerto-Cosme	John Stewart	01-07-2012	Reactivar
Convenio Vigente Marco	National Taiwan Universit	Francisco Santidrián	Shan Ku Fen	19-03-2013	Reactivar

Convenios Específicos Caducados

Nombre	Universidad Extranjera	Coordinador UvA	Coordinador Exterior	F.Renovación	
Convenio Vigente	Universidad de Tokio	Marta López	Coordin Exterior	12-01-2013	Reactivar
Convenio Vigente Especifico	National Taiwan Universit	Agatha Silva	Shan Ku Fen	23-01-2013	Reactivar
Convenio Vigente Tramo Especifico	Otago Polytechnic	Irene Prada	John Stewart	20-02-2013	Reactivar
El convenio mas largo del mundo ademas de todo vigente	UCLA	Pedro Rosales	Anton Martin	22-01-2013	Reactivar
Convenio Total Especifico	Universidad de Paris	Laura Polo	Frederic Vernoulli	24-01-2013	Reactivar

Servicio de Relaciones Internacionales - Universidad de Valladolid
 Casa del Estudiante - Calle Real de Burgos, s/n
 47011 Valladolid España




5.1.6. Gestión de los datos

Desde este apartado se nos permite acceder a todos los elementos modificables del sistema, como son los coordinadores, centros, oficinas, universidades, etc. Todos estos elementos deben poder ser creados, editados y borrados del sistema para el mantenimiento y funcionamiento de la aplicación.

Debido al gran número de elementos y navegación que debería añadirse a cada una de las operaciones descritas, todas ellas se realizan mediante formularios emergentes que recogen la información suministrada por el administrador y la almacenan, editan o borran del sistema.



Como ya hemos dicho, para cada elemento podemos:

- **Añadir:** al seleccionar () se nos muestra un formulario vacío donde completar la información de la entidad. Se selecciona “Añadir” y el elemento queda añadido.
- **Modificar:** se selecciona (). Se muestra un selector con todos los elementos disponibles en el sistema del tipo seleccionado (por ejemplo, todos los coordinadores de la UVA). El administrador elige el elemento a modificar y se muestra toda la información acerca de dicho elemento. El administrador puede ahora modificar los datos y almacénar las modificaciones.
- **Eliminar:** se selecciona () y se muestra un listado al igual que el caso anterior, para que el administrador elija el elemento a borrar. Se muestran los datos del elemento seleccionado para verificar que se borra el elemento que el administrador quiere. El borrado de un elemento en el sistema, como ya sea explicado, no se lleva a cabo de manera completa, ya que al existir relaciones entre muchos elementos a través del convenio, se producirían errores. Si borramos, por ejemplo un rector, deberíamos borrar del sistema todos los convenios que éste ha gestionado y todos los elementos relacionados con esos convenios. Como este caso no es el deseado, el borrado se realiza cambiando el estado del elemento a borrado, haciendo que no sea accesible para la creación de nuevos convenios, pero permitiendo su consulta para convenios ya creados.

Sistema web gestor de Convenios Internacionales

El formulario de creación es similar a los anteriores mostrados, mostraremos el de modificación y eliminación que son diferentes a los mostrados:

Formulario de modificación con selector de rectores UVA

Datos Rector UVA: ▼

Fernando Sacristan ▼

Nombre:	Fernando
Apellidos:	Sacristan
Dirección Completa:	Campus Miguel Delibes
Código Postal:	47011
Localidad:	Valladolid
Provincia:	Valladolid
Teléfono:	983333888
Fáx:	983333887
Correo electrónico:	rector@uva.es
Estado:	<input type="radio"/> Disponible <input checked="" type="radio"/> Vigente

[Modificar](#) [Volver](#)

Formulario de eliminación de oficinas gestoras exteriores

Datos Oficina Gestora EXT: ▼

Department of Foreign Relationships ▼

Nombre de la Oficina:	Department of Foreign Relationships
Dirección Completa:	Round Avd
Código Postal:	20086
Localidad:	Estocolmo
Provincia:	Estocolmo
País:	Suecia
Teléfono:	45671235
Fáx:	45671236
Correo electrónico:	fr@unistockholm.se
Universidad:	Universidad de Estocolmo ▼

[Eliminar](#) [Volver](#)

5.1.7. Documentación

Para llevar a cabo el mantenimiento de los ficheros que se ponen a disposición de la comunidad universitaria, es necesario tener un repositorio de documentos para administrarlos.

En este apartado se nos permite modificar o eliminar los documentos existentes o añadir nuevos. Para disminuir la navegación se han realizado sobre la misma pantalla, separando ambas funcionalidades en dos pestañas separadas.




Para subir un nuevo documento, vamos a la pestaña de “Subir documento”. Se nos muestra un formulario con el nombre del documento, una pequeña descripción, se selecciona el tipo de documento y el contenido del fichero en sí.



Sistema web gestor de Convenios Internacionales

Si todo ha ido bien, aparecerá un mensaje de éxito y dicho documento será accesible tanto para administrador como para los usuarios normales.

Si queremos gestionar los documentos ya subidos, vamos a la pestaña de “Ver documentos” y podemos realizar dos tareas distintas:

- **Edición:** seleccionamos () y se muestra un formulario con los datos del documento. Se modifican los datos oportunos, incluso el contenido del documento, añadiendo uno nuevo y borrando el anterior de manera transparente al usuario.



Subir Documentación: —

Nombre:
Anexo I Cotelas


Descripción:
Anexo I Cotelas de Tesis

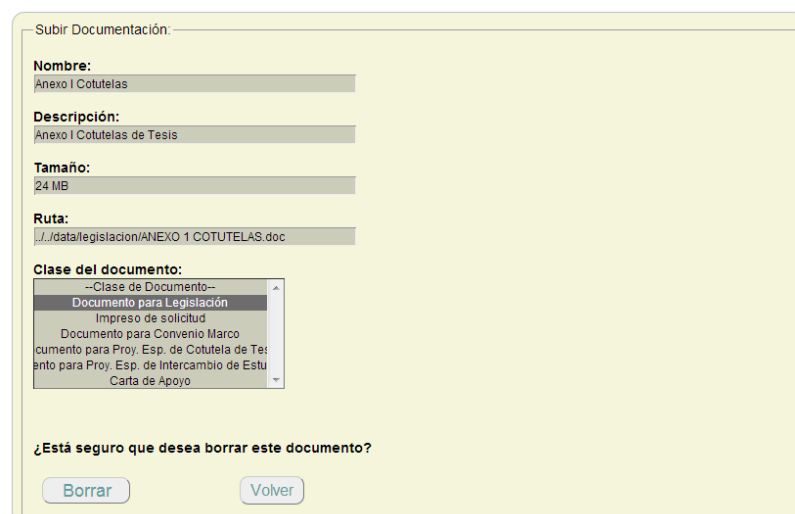
Clase del documento:
Documento para Legislación

Documento:
[Seleccionar archivo](#) No se eligió archivo

Si no se rellena este campo, se mantendrá el documento ya guardado.

[Actualizar](#) [Volver](#)

- **Borrado:** seleccionando () , se nos muestran los datos del documento para verificar que el documento seleccionado es el que queremos borrar. Todos los campos están desactivados para que no se puedan modificar antes del borrado.



Subir Documentación: —

Nombre:
Anexo I Cotelas

Descripción:
Anexo I Cotelas de Tesis

Tamaño:
24 MB

Ruta:
./../data/legislacion/ANEXO 1 COTUTELAS.doc

Clase del documento:
--Clase de Documento--
Documento para Legislación
Impreso de solicitud
Documento para Convenio Marco
Documento para Proy. Esp. de Cotelas de Te
Documento para Proy. Esp. de Intercambio de Estu
Carta de Apoyo

¿Está seguro que desea borrar este documento?

[Borrar](#) [Volver](#)

5.2. Manual de instalación

Como tal, no existe un manual de instalación mejor que seguir los pasos del despliegue. La mayoría de las aplicaciones web, y todos los sistemas basados en clientes ligeros, no tienen proceso de instalación en los ordenadores que los usan.

Tan solo hay que desplegar el proyecto en el servidor dedicado que vayamos a utilizar, realizar las conexiones necesarias y permitir a los usuarios acceder a través de una URL, ya sea pública o privada.

Capítulo 6

Conclusiones

6. Conclusiones

Aunque no se encuentren unas conclusiones obtenidas a partir del desarrollo de un sistema web, vamos a exponer los aspectos más relevantes de este proyecto.

6.1. Resumen y valoración del trabajo

La realización de este Proyecto Fin de Carrera ha sido un duro trabajo a lo largo de varios meses. Su consecución, como todo proyecto, ha tenido sus partes buenas y sus partes malas, pero al finalizar, parece que todas las partes malas desaparecen para quedarnos con lo bueno.

En cuanto al producto final, creemos que cubrimos todos los aspectos solicitados por el Servicio de Relaciones Internacionales de la Universidad de Valladolid reflejados en los requisitos del sistema.

En resumen, pensamos que esta aplicación se ha llevado a cabo de las mejores maneras posibles, siguiendo un plan bien establecido que se ajusta a aquello que nos encontraremos en las empresas de desarrollo de software y que hemos aprendido a lo largo de los estudios realizados.

En cuanto a lo que a esta memoria se refiere, nos parece que puede servir a futuros programadores o alumnos que realicen su Proyecto Fin de Carrera para tomarla como guía de desarrollo de su aplicación. Bien es cierto que no aportamos demasiada información acerca del código empleado, aunque este se encuentra íntegro en el CD-ROM adjunto.

Gracias a este Proyecto, hemos aprendido la utilización de ciertas tecnologías de desarrollo software como son el lenguaje de programación web como son PHP para el lado del servidor, HTML, CSS, Javascript y jQuery para el lado del cliente y MySQL para la base de datos. Gracias a la unión de estas tecnologías hemos desarrollado una aplicación vistosa, atractiva y llamativa que creemos que puede ser fácil de usar y de entender para el usuario.

También conviene añadir que el desarrollo de este sistema nos ha permitido llevar a cabo el estudio y uso del patrón MVC aplicado a las tecnologías web, del que tanto hemos oído hablar a lo largo de nuestros estudios y al que por fin nos hemos enfrentado .

Como fin a este apartado, sería necesario recalcar que la elaboración de este Proyecto Fin de Grado conlleva un aprendizaje inmenso que nos permite adquirir un nivel más de programación con respecto al que finalizamos los estudios.

6.2. Problemas y dificultades

A lo largo del desarrollo de este Proyecto, hemos ido afrontando diferentes obstáculos que se interponían entre nosotros y la consecución del sistema. Unas dificultades que se han solventado de mejor o peor manera y que han otorgado al desarrollador conocimientos a la hora de enfrentarse a diferentes problemas.

Unas de las partes más complicadas a la que nos hemos enfrentado en la duración del proyecto ha sido el desarrollo de las tecnologías en cliente, como son Javascript y jQuery. Esta dificultad viene dada por no haber visto nada similar durante los estudios cursados, por lo que nuestro punto de partida era cero.

Por otro lado, existe una gran documentación de calidad y una amplia comunidad de desarrolladores que favorecen el aprendizaje y desarrollo de los programadores en tecnologías tan demandadas como las citadas.

También conviene destacar la dificultad que tiene comenzar todo proyecto, esa fase de aprendizaje y documentación en la que no conseguimos ninguna meta visible y que puede ser un problema si no nos la tomamos como lo que es, un aprendizaje para el futuro del proyecto.

A esta fase hay que sumarle la dificultad de extraer los requisitos a través de reuniones con el cliente. Esta tarea ya es difícil en empresas dedicadas al desarrollo de software, con contratos de por medio y requisitos firmados, pero si bajamos de nivel hasta la universidad, donde hay un mayor nivel de cercanía y confianza, nos encontramos ante requisitos que aún son más variables y cambios más frecuentes en la manera de ver las cosas.

6.3. Líneas futuras

En cuanto a la visión de futuro, pocas cosas más hay que añadir. El sistema se ha ajustado a todo lo necesitado por parte del Servicio de Relaciones Internacionales de la Universidad de Valladolid, pero hay que tener en cuenta que esta funcionalidad ha sido desarrollada al trasladar un trabajo realizado en papel y mediante software no especializado, hacia un sistema web con mayor facilidad de uso y automatización. Todo esto puede hacer que nuevas funcionalidades aparezcan a partir de su uso, como nuevos criterios de búsqueda, nuevas facilidades a la comunidad universitaria, etc.

Capítulo 7

Bibliografía

7. Bibliografía

- Conallen, J. “Building web applications with UML”, 2ª edición : Addison-Wesley , 2002
- Sommerville, I. “Ingeniería del Software”, 7ª edición, Pearson Educación S.A., Madrid. 2004.
- Larman, C. “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development”, 3ª edición, Prentice Hall 2004.
- Arlow, J. Neustadt, I. “UML 2 and The Unified Process: Practica Object-Oriented Analysis And Desing”. Addison-Wesley, 2005
- Rumbaugh, J, Jacobson, I, Booch, G. “El proceso Unificado de Desarrollo de Software”. Addison-Wesley, 2000.
- E. Gamma. Patrones de diseño: elementos de software orientado a objetos reutilizable / Madrid : Addison-Wesley , 2006
- Ludwig-Maximilians-Universität München. “UWE – UML-based Web Engineering”. [en línea] 2007. <http://uwe.pst.ifi.lmu.de/exampleSimpleWebsite.html>
- jQuery foundation. “Documentacion oficial jQuery UI”. [en línea] 2013. <http://jqueryui.com/>
- jQuery foundation. “Documentacion oficial jQuery”. [en línea] 2013. <http://jquery.com/>
- jQuery foundation. “jQuery.ajax()”. [en línea] 2013. <http://api.jquery.com/jQuery.ajax/>
- Navarro, D. “Patrones de diseño web: Patron F”. La webera.[en línea] 2012. <http://www.lawebera.es/disenio-web/patrones-layout-patron-f.php>
- Herrington, J. “Five common PHP design patterns”. DevelopersWorks [en línea] 2006. <http://www.ibm.com/developerworks/library/os-php-designptrns/?ca=dgr-lnxw57PHP-Patterns>
- Google, Inc. “Google Fonts”. Google.com. [en línea] 2013. <http://www.google.com/fonts/#>
- “PHP Manual”. Php.net. [en línea] 2013. <http://php.net/manual/es/index.php>
- Santiago, R. “Estereotipos UML para aplicaciones web”. Universidad de Cauca. [en línea]. 2008. <http://www.slideshare.net/2008PA2Info3/extensiones-uml-para-aplicaciones-web-roco-santiago>

Apéndices

8. Apéndice I

Script ejecutable en el SGBD para la creación de la base de datos MySQL

```
-- phpMyAdmin SQL Dump
-- version 3.4.5
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 14-06-2013 a las 04:10:56
-- Versión del servidor: 5.5.16
-- Versión de PHP: 5.3.8

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

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

--
-- Base de datos: `relint`
--
-----

--
-- Estructura de tabla para la tabla `actividadestipo_new`
--

CREATE TABLE IF NOT EXISTS `actividadestipo_new` (
  `IDActividadTipo` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` char(100) DEFAULT NULL,
  `Descripcion` mediumtext,
  `Clase` char(15) DEFAULT NULL,
  `Estado` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`IDActividadTipo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=28 ;

-----

--
-- Estructura de tabla para la tabla `act_conv_new`
--

CREATE TABLE IF NOT EXISTS `act_conv_new` (
```

Capítulo 7: Bibliografía

```
`IDActividadTipo` int(11) NOT NULL DEFAULT '0',
`IDConvenio` int(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`IDConvenio`,`IDActividadTipo`),
KEY `FKdes_Act_IND` (`IDActividadTipo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
-----

--
-- Estructura de tabla para la tabla `administrador_new`
--

CREATE TABLE IF NOT EXISTS `administrador_new` (
  `IDAdmin` int(5) NOT NULL AUTO_INCREMENT,
  `Nombre` varchar(20) DEFAULT NULL,
  `Apellidos` varchar(40) DEFAULT NULL,
  `Username` varchar(12) DEFAULT NULL,
  `Password` varchar(100) DEFAULT NULL,
  `Salt` varchar(100) NOT NULL,
  PRIMARY KEY (`IDAdmin`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
-----

--
-- Estructura de tabla para la tabla `centro_new`
--

CREATE TABLE IF NOT EXISTS `centro_new` (
  `IDCentro` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` char(50) DEFAULT NULL,
  `Direccion` char(50) DEFAULT NULL,
  `CodPostal` char(5) DEFAULT NULL,
  `Localidad` char(15) DEFAULT NULL,
  `Provincia` char(15) DEFAULT NULL,
  `Telefono` char(20) DEFAULT NULL,
  `Fax` char(20) DEFAULT NULL,
  `Web` char(20) DEFAULT NULL,
  `Email` char(30) DEFAULT NULL,
  `Estado` varchar(25) NOT NULL,
  PRIMARY KEY (`IDCentro`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=19 ;
-----

--
-- Estructura de tabla para la tabla `convenio_new`
--

CREATE TABLE IF NOT EXISTS `convenio_new` (
  `IDConvenio` int(11) NOT NULL AUTO_INCREMENT,
  `NombreConvenio` char(100) DEFAULT NULL,
  `Estado` char(15) DEFAULT NULL,
  `TipoRenovacion` char(20) DEFAULT NULL,
  `FechaRenovacion` date DEFAULT NULL,
  `DocMarco` char(5) DEFAULT NULL,
  `DocEspecifico` varchar(5) DEFAULT NULL,
  `Formulario` char(5) DEFAULT NULL,
```

```
`CartaApoyo` char(5) DEFAULT NULL,
`FechaEnvAsesJur` date DEFAULT NULL,
`FechaRecAsesJur` date DEFAULT NULL,
`ComisionRRII` date DEFAULT NULL,
`ComisionPermanente` date DEFAULT NULL,
`FechaEnvRector` date DEFAULT NULL,
`FechaRecRector` date DEFAULT NULL,
`FechaEnvVice` date DEFAULT NULL,
`FechaRecVice` date DEFAULT NULL,
`FechaEnvFirmaEXT` date DEFAULT NULL,
`FechaRecFirmaEXT` date DEFAULT NULL,
`FechaEnvCoordinador` date DEFAULT NULL,
`FechaEnvSecreGen` date DEFAULT NULL,
`FechaEnvPostGradoTesis` date DEFAULT NULL,
`Observaciones` mediumtext,
`ObservacionesPublicas` text,
`FechaFirmaInicial` date DEFAULT NULL,
`TipoConvenio` char(12) NOT NULL,
`NivelEstudios` char(20) DEFAULT NULL,
`NumEstudiantes` int(3) DEFAULT NULL,
`DuracionIntercambioEst` int(3) NOT NULL,
`IDVicerrector` int(11) DEFAULT NULL,
`IDCoordUVA` int(11) DEFAULT NULL,
`IDGestorConveniosEXT` int(11) DEFAULT NULL,
`IDGestorConveniosUVA` int(11) DEFAULT NULL,
`IDCoordEXT` int(11) DEFAULT NULL,
`IDRector` int(11) DEFAULT NULL,
`ConvenioMarcoAsociado` int(11) DEFAULT NULL,
`IDRectorEXT` int(11) DEFAULT NULL,
PRIMARY KEY (`IDConvenio`),
KEY `FKrelacionado_IND` (`IDVicerrector`),
KEY `FKpromueve_IND` (`IDCoordUVA`),
KEY `FKgestion_a_IND` (`IDGestorConveniosEXT`),
KEY `FKgestion_IND` (`IDGestorConveniosUVA`),
KEY `FKcoordina_IND` (`IDCoordEXT`),
KEY `FKasociado_IND` (`IDRector`),
KEY `FKIDRectorEXT` (`IDRectorEXT`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=77 ;
```

```
--
--
-- Estructura de tabla para la tabla `coordinadorext_new`
--
```

```
CREATE TABLE IF NOT EXISTS `coordinadorext_new` (
  `IDCoordEXT` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` char(20) DEFAULT NULL,
  `Apellidos` char(40) DEFAULT NULL,
  `Direccion` char(30) DEFAULT NULL,
  `Localidad` char(20) DEFAULT NULL,
  `Provincia` varchar(20) DEFAULT NULL,
```

Capítulo 7: Bibliografía

```
`CodPostal` char(10) DEFAULT NULL,  
`Pais` char(40) DEFAULT NULL,  
`Telefono` char(25) DEFAULT NULL,  
`Fax` char(25) DEFAULT NULL,  
`Email` char(20) DEFAULT NULL,  
`Estado` varchar(20) NOT NULL,  
`IDUniversidadEXT` int(11) DEFAULT NULL,  
PRIMARY KEY (`IDCoordEXT`),  
KEY `FKpertenece_a_IND` (`IDUniversidadEXT`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=34 ;  
-----
```

```
--  
-- Estructura de tabla para la tabla `coordinadoruva_new`  
--
```

```
CREATE TABLE IF NOT EXISTS `coordinadoruva_new` (  
  `IDCoordUVA` int(11) NOT NULL AUTO_INCREMENT,  
  `Nombre` char(20) DEFAULT NULL,  
  `Apellidos` char(40) DEFAULT NULL,  
  `Direccion` char(30) DEFAULT NULL,  
  `Localidad` char(20) DEFAULT NULL,  
  `Provincia` char(20) DEFAULT NULL,  
  `CodPostal` char(5) DEFAULT NULL,  
  `Telefono` char(20) DEFAULT NULL,  
  `Fax` char(20) DEFAULT NULL,  
  `Email` char(20) DEFAULT NULL,  
  `Estado` varchar(20) NOT NULL,  
  `IDDepartamento` int(11) DEFAULT NULL,  
  PRIMARY KEY (`IDCoordUVA`),  
  KEY `FKpertenece_IND` (`IDDepartamento`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=48 ;  
-----
```

```
--  
-- Estructura de tabla para la tabla `departamento_new`  
--
```

```
CREATE TABLE IF NOT EXISTS `departamento_new` (  
  `IDDepartamento` int(11) NOT NULL AUTO_INCREMENT,  
  `Nombre` char(100) DEFAULT NULL,  
  `Estado` varchar(20) NOT NULL,  
  `IDCentro` int(11) DEFAULT NULL,  
  PRIMARY KEY (`IDDepartamento`),  
  KEY `FKtiene_IND` (`IDCentro`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=31 ;  
-----
```

```
--  
-- Estructura de tabla para la tabla `documentos_new`  
--
```

```
CREATE TABLE IF NOT EXISTS `documentos_new` (  
  `IDDocumento` int(11) NOT NULL AUTO_INCREMENT,  
  `Nombre` varchar(50) NOT NULL,  
  `Tipo` varchar(80) NOT NULL,
```

```
`Tamano` float NOT NULL,
`Descripcion` varchar(100) NOT NULL,
`Ruta` varchar(80) NOT NULL,
PRIMARY KEY (`IDDocumento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=27 ;
-----
```

```
--
-- Estructura de tabla para la tabla `gestorconveniosext_new`
--
CREATE TABLE IF NOT EXISTS `gestorconveniosext_new` (
  `IDGestorConveniosEXT` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` char(20) DEFAULT NULL,
  `Apellidos` char(45) DEFAULT NULL,
  `Cargo` char(50) DEFAULT NULL,
  `Estado` varchar(20) NOT NULL,
  `IDOficinaGestoraEXT` int(11) DEFAULT NULL,
  PRIMARY KEY (`IDGestorConveniosEXT`),
  KEY `FKtrabaja_en_IND` (`IDOficinaGestoraEXT`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=26 ;
-----
```

```
--
-- Estructura de tabla para la tabla `gestorconveniosuva_new`
--
CREATE TABLE IF NOT EXISTS `gestorconveniosuva_new` (
  `IDGestorConveniosUVA` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` char(20) DEFAULT NULL,
  `Apellidos` char(45) DEFAULT NULL,
  `Cargo` char(20) DEFAULT NULL,
  `IDOficinaGestoraUVA` int(11) DEFAULT NULL,
  `Estado` varchar(20) NOT NULL,
  PRIMARY KEY (`IDGestorConveniosUVA`),
  KEY `FKtrabaja_IND` (`IDOficinaGestoraUVA`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;
-----
```

```
--
-- Estructura de tabla para la tabla `oficinagestoraext_new`
--
CREATE TABLE IF NOT EXISTS `oficinagestoraext_new` (
  `IDOficinaGestoraEXT` int(11) NOT NULL AUTO_INCREMENT,
  `NombreOficina` char(50) DEFAULT NULL,
  `Direccion` char(50) DEFAULT NULL,
  `Localidad` char(15) DEFAULT NULL,
  `Provincia` char(15) DEFAULT NULL,
  `CodPostal` char(5) DEFAULT NULL,
  `Pais` char(40) DEFAULT NULL,
  `Telefono` char(9) DEFAULT NULL,
```

Capítulo 7: Bibliografía

```
`Fax` char(9) DEFAULT NULL,  
`Email` char(30) DEFAULT NULL,  
`Estado` varchar(20) NOT NULL,  
`IDUniversidadEXT` int(11) DEFAULT NULL,  
PRIMARY KEY (`IDOficinaGestoraEXT`),  
KEY `FKlleva_IND` (`IDUniversidadEXT`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=23 ;  
-----  
--  
-- Estructura de tabla para la tabla `oficinagestorauva_new`  
--  
CREATE TABLE IF NOT EXISTS `oficinagestorauva_new` (  
  `IDOficinaGestoraUVA` int(11) NOT NULL AUTO_INCREMENT,  
  `NombreOficina` char(50) DEFAULT NULL,  
  `Direccion` char(50) DEFAULT NULL,  
  `Localidad` char(30) DEFAULT NULL,  
  `Provincia` char(30) DEFAULT NULL,  
  `CodPostal` char(15) DEFAULT NULL,  
  `Telefono` char(20) DEFAULT NULL,  
  `Fax` char(20) DEFAULT NULL,  
  `Email` char(30) DEFAULT NULL,  
  `Estado` varchar(20) NOT NULL,  
  PRIMARY KEY (`IDOficinaGestoraUVA`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;  
-----  
--  
-- Estructura de tabla para la tabla `rectorext_new`  
--  
CREATE TABLE IF NOT EXISTS `rectorext_new` (  
  `IDRectorEXT` int(11) NOT NULL AUTO_INCREMENT,  
  `IDUniversidadEXT` int(11) DEFAULT NULL,  
  `Nombre` char(20) DEFAULT NULL,  
  `Apellidos` char(40) DEFAULT NULL,  
  `Direccion` char(30) DEFAULT NULL,  
  `Localidad` char(20) DEFAULT NULL,  
  `Provincia` char(50) DEFAULT NULL,  
  `CodPostal` char(10) DEFAULT NULL,  
  `Pais` char(40) DEFAULT NULL,  
  `Telefono` char(15) DEFAULT NULL,  
  `Fax` char(15) DEFAULT NULL,  
  `Email` char(20) DEFAULT NULL,  
  `Estado` char(15) NOT NULL,  
  PRIMARY KEY (`IDRectorEXT`),  
  KEY `FKdirige_ID` (`IDUniversidadEXT`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=32 ;  
-----  
--  
-- Estructura de tabla para la tabla `rectoruva_new`  
--  
CREATE TABLE IF NOT EXISTS `rectoruva_new` (  

```



```

`IDRector` int(11) NOT NULL AUTO_INCREMENT,
`Nombre` char(20) DEFAULT NULL,
`Apellidos` char(40) DEFAULT NULL,
`Direccion` char(30) DEFAULT NULL,
`Localidad` char(20) DEFAULT NULL,
`Provincia` char(20) DEFAULT NULL,
`CodPostal` char(5) DEFAULT NULL,
`Telefono` char(9) DEFAULT NULL,
`Fax` char(9) DEFAULT NULL,
`Email` char(20) DEFAULT NULL,
`Estado` char(20) NOT NULL,
PRIMARY KEY (`IDRector`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;
-----

--
-- Estructura de tabla para la tabla `universidadext_new`
--
CREATE TABLE IF NOT EXISTS `universidadext_new` (
  `IDUniversidadEXT` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` char(25) DEFAULT NULL,
  `Direccion` char(50) DEFAULT NULL,
  `Localidad` char(30) DEFAULT NULL,
  `Provincia` char(30) DEFAULT NULL,
  `CodPostal` char(10) DEFAULT NULL,
  `Pais` char(40) DEFAULT NULL,
  `Telefono` char(12) DEFAULT NULL,
  `Fax` char(12) DEFAULT NULL,
  `Email` char(20) DEFAULT NULL,
  `Estado` varchar(20) NOT NULL,
  PRIMARY KEY (`IDUniversidadEXT`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=22 ;
-----

--
-- Estructura de tabla para la tabla `vicerrectoruva_new`
--
CREATE TABLE IF NOT EXISTS `vicerrectoruva_new` (
  `IDVicerrector` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` char(20) DEFAULT NULL,
  `Apellidos` char(40) DEFAULT NULL,
  `Direccion` char(30) DEFAULT NULL,
  `Localidad` char(20) DEFAULT NULL,
  `Provincia` char(20) DEFAULT NULL,
  `CodPostal` char(5) DEFAULT NULL,
  `Telefono` char(9) DEFAULT NULL,
  `Fax` char(9) DEFAULT NULL,
  `Email` char(20) DEFAULT NULL,
  `Estado` char(12) NOT NULL,
  PRIMARY KEY (`IDVicerrector`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=3 ;

```

Capítulo 7: Bibliografía

```
--
-- Restricciones para tablas volcadas
--
--
-- Filtros para la tabla `act_conv_new`
--
ALTER TABLE `act_conv_new`
  ADD CONSTRAINT `FKdes_Act_FK` FOREIGN KEY (`IDActividadTipo`) REFERENCES
`actividadestipo_new` (`IDActividadTipo`),
  ADD CONSTRAINT `FKdes_Con` FOREIGN KEY (`IDConvenio`) REFERENCES
`convenio_new` (`IDConvenio`);
--
-- Filtros para la tabla `convenio_new`
--
ALTER TABLE `convenio_new`
  ADD CONSTRAINT `convenio_new_ibfk_43` FOREIGN KEY (`IDVicerrector`)
REFERENCES `vicerrectoruva_new` (`IDVicerrector`),
  ADD CONSTRAINT `convenio_new_ibfk_44` FOREIGN KEY (`IDCoordUVA`)
REFERENCES `coordinadoruva_new` (`IDCoordUVA`),
  ADD CONSTRAINT `convenio_new_ibfk_45` FOREIGN KEY (`IDGestorConveniosEXT`)
REFERENCES `gestorconveniosext_new` (`IDGestorConveniosEXT`),
  ADD CONSTRAINT `convenio_new_ibfk_46` FOREIGN KEY (`IDGestorConveniosUVA`)
REFERENCES `gestorconveniosuva_new` (`IDGestorConveniosUVA`),
  ADD CONSTRAINT `convenio_new_ibfk_47` FOREIGN KEY (`IDCoordEXT`)
REFERENCES `coordinadorext_new` (`IDCoordEXT`),
  ADD CONSTRAINT `convenio_new_ibfk_48` FOREIGN KEY (`IDRector`) REFERENCES
`rectoruva_new` (`IDRector`),
  ADD CONSTRAINT `convenio_new_ibfk_49` FOREIGN KEY (`IDRectorEXT`)
REFERENCES `rectorext_new` (`IDRectorEXT`);
--
-- Filtros para la tabla `coordinadorext_new`
--
ALTER TABLE `coordinadorext_new`
  ADD CONSTRAINT `FKpertenece_a_FK` FOREIGN KEY (`IDUniversidadEXT`)
REFERENCES `universidadext_new` (`IDUniversidadEXT`);
--
-- Filtros para la tabla `coordinadoruva_new`
--
ALTER TABLE `coordinadoruva_new`
  ADD CONSTRAINT `FKpertenece_FK` FOREIGN KEY (`IDDepartamento`) REFERENCES
`departamento_new` (`IDDepartamento`);
--
-- Filtros para la tabla `departamento_new`
--
ALTER TABLE `departamento_new`
  ADD CONSTRAINT `FKtiene_FK` FOREIGN KEY (`IDCentro`) REFERENCES `centro_new`
(`IDCentro`);
--
-- Filtros para la tabla `gestorconveniosext_new`
--
ALTER TABLE `gestorconveniosext_new`
```

Sistema web gestor de Convenios Internacionales

```
ADD CONSTRAINT `FKtrabaja_en_FK` FOREIGN KEY (`IDOficinaGestoraEXT`)
REFERENCES `oficinagestoraext_new` (`IDOficinaGestoraEXT`);
--
-- Filtros para la tabla `gestorconveniosuva_new`
--
ALTER TABLE `gestorconveniosuva_new`
ADD CONSTRAINT `FKtrabaja_FK` FOREIGN KEY (`IDOficinaGestoraUVA`)
REFERENCES `oficinagestorauva_new` (`IDOficinaGestoraUVA`);
--
-- Filtros para la tabla `oficinagestoraext_new`
--
ALTER TABLE `oficinagestoraext_new`
ADD CONSTRAINT `FKlleva_FK` FOREIGN KEY (`IDUniversidadEXT`) REFERENCES
`universidadext_new` (`IDUniversidadEXT`);
--
-- Filtros para la tabla `rectorext_new`
--
ALTER TABLE `rectorext_new`
ADD CONSTRAINT `FKdirige_FK` FOREIGN KEY (`IDUniversidadEXT`) REFERENCES
`universidadext_new` (`IDUniversidadEXT`);

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