



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Universidad Politécnica de Valencia
Escuela Técnica Superior de Ingeniería Informática

Ingeniería Técnica en Informática de Gestión

Integración Abox Document con Microsoft SharePoint 2010

Autor: Humberto Delgado Lázaro

Directora: María Carmen Penadés Gramaje

Valencia, Noviembre 2011

Resumen

La propuesta que se ha desarrollado en este proyecto trata sobre la integración de la plataforma de gestión documental Abox Document con la plataforma de colaboración Microsoft SharePoint Server 2010. El objetivo de esta propuesta es poder ampliar enormemente el mercado potencial de Abox a través de enlazar ambas plataformas.

Se presenta el entorno de la empresa Adapting S.L., en la cual se ha desarrollado el proyecto, destacando la estructura organizativa, la metodología empleada además de otros factores de interés de la misma. Posteriormente se presenta un amplio estudio de las dos plataformas, detallando sus virtudes, posibilidades y carencias.

El proyecto incluye la programación de una solución real de integración, la cual ha sido desarrollada durante varios meses, utilizando la metodología Extreme Programming (XP). Se presenta la planificación en iteraciones, artefactos generados y manual de usuario que ilustra su uso.

Palabras clave

Integración, SharePoint, Abox, XP, Adapting.

A mi madre Encarna
y a mis abuelos Encarna y Luis

Agradecimientos

Deseo expresar mi más sincero agradecimiento a todas aquellas personas que han colaborado de manera directa o indirecta en este proyecto.

En primer lugar, a mis compañeros de trabajo por el apoyo mostrado en todo momento. En especial a Belén Aguilera, por toda su ayuda y atención mostrada.

Quiero mostrar mi gratitud a mi directora María Carmen Penadés Gramaje, de la Universidad Politécnica de Valencia. Las reuniones celebradas han sido de gran impulso para este proyecto final de carrera.

Por último quiero dar las gracias a mi familia, por confiar en mí y apoyarme en todo momento. Ellos me han inculcado los valores y principios que han servido de cimiento, para que hoy se vea realizado uno de mis objetivos.

Índice general

1. Introducción	1
1.1. Motivación.....	1
1.2. Objetivos	1
1.3. Beneficios esperados	4
1.4. Formación para el proyecto	4
1.5. Estructura del documento	5
2. Entorno	7
2.1. La empresa: Adapting S.L.	8
2.1.1. Historia de la empresa	8
2.1.2. Estructura de la organización	10
2.1.3. Productos y servicios	12
2.1.4. Clientes	12
2.1.5. Plan de I+D+I	13
2.2. Metodología de desarrollo de software: XP.....	14
2.2.1. Prácticas	16
2.2.2. Roles.....	19
2.2.3. Artefactos	20
2.3. Tecnología	23

2.3.1.	Abox	23
2.3.1.1.	Módulos de Abox	25
2.3.1.2.	Caso de éxito	25
2.3.2.	SharePoint	29
2.3.2.1.	Historia de SharePoint	30
2.3.2.2.	Ventajas e inconvenientes	32
2.3.2.3.	Principales funcionalidades	33
2.3.2.4.	Formas básicas de extensibilidad	35
3.	Propuesta de integración	37
3.1.	Equipo	38
3.1.1.	Reunión con el Instituto Tecnológico de Informática	39
3.2.	Planificación	39
3.3.	Plan de entregas	43
3.4.	Prototipo	45
3.4.1.	Objetivos	45
3.4.2.	Implementación	46
3.4.3.	Grado de consecución de los objetivos	47
4.	Desarrollo	49
4.1.	Arquitectura	50
4.2.	Tecnologías	52
4.2.1.	Object Client Model	53
4.2.2.	NHibernate	56
4.2.3.	Model view controller	58
4.2.4.	Inversion of control	60
4.3.	Herramientas	61
4.4.	Pruebas	61
4.5.	Buenas practicas aplicadas	62

Índice general

5. Conclusiones	65
5.1. Conclusiones.....	66
5.1.1. Detalle de la consecución de objetivos.....	67
5.2. Líneas de trabajos futuros y posibles mejoras	68
 Bibliografía	 71
 Apéndices	 73
A.1. Manual del usuario	73
A.2. Historias de usuario, tareas de ingeniería y casos de prueba.....	95

Índice de figuras

<i>Figura 2.1.</i> Organigrama vertical de Adapting S.L.	9
<i>Figura 2.2.</i> Organigrama de desarrollo de proyectos	11
<i>Figura 2.3.</i> Buenas prácticas. Obtenida de [Be99]	16
<i>Figura 2.4.</i> El juego de la planificación.	16
<i>Figura 2.5.</i> Modelo propuesto para una historia de usuario. Obtenido de [Ca03]. ...	21
<i>Figura 2.6.</i> Modelo propuesto para una tarea de ingeniería. Obtenido de [Ca03]. ...	22
<i>Figura 2.7.</i> Modelo propuesto para una prueba de aceptación. Obtenido de [Ca03]. ..	22
<i>Figura 2.8.</i> Ventana principal de Abox.	24
<i>Figura 2.9.</i> Listado de módulos de Abox.	25
<i>Figura 2.10.</i> Antigua web de AgdoJavi.	26
<i>Figura 2.11.</i> Nueva extranet de Abox para Agdojavi.	27
<i>Figura 2.12.</i> Intranet de SharePoint 2010.	29
<i>Figura 2.13.</i> Características y funcionalidades en SharePoint.	31
<i>Figura 2.14.</i> Esquema SharePoint 2010. Obtenido de [Gm10].	33
<i>Figura 3.1.</i> Historia de usuario.	40
<i>Figura 3.2.</i> Tarea de ingeniería.	40
<i>Figura 3.3.</i> Prueba de aceptación.	41
<i>Figura 3.4.</i> Listado de librerías de SharePoint vistas desde el prototipo.	46
<i>Figura 3.5.</i> Información de una librería vista desde el prototipo.	47
<i>Figura 3.6.</i> Acciones del prototipo	47
<i>Figura 4.1.</i> Arquitectura 3 capas. Obtenido de [Wp11].	50
<i>Figura 4.2.</i> Arquitectura 3 capas. Obtenido de [Pr11].	51
<i>Figura 4.3.</i> Arquitectura de la aplicación.	52
<i>Figura 4.4.</i> Resumen de cómo funciona el <i>Client Object Model</i>	53
<i>Figura 4.5.</i> Hola mundo utilizando <i>Client Object Model</i> . Obtenido de [Us10]	55
<i>Figura 4.6.</i> Ejemplo de uso de <i>LoadQuery</i> . Obtenido de [Us10]	55
<i>Figura 4.7.</i> Selección de datos con <i>Client Object Model</i> . Obtenido de [Us10].	56

Índice de figuras

<i>Figura 4.8.</i> Fichero configuración de NHibernate.....	57
<i>Figura 4.9.</i> Archivo que mapea la tabla configuración	57
<i>Figura 4.10.</i> Tabla configuración dentro de la base de datos.....	58
<i>Figura 4.11.</i> Model view controller. Obtenido de [Ma11].....	59
<i>Figura 4.12.</i> Ejemplo MVC	59
<i>Figura 4.13.</i> Dependencia de los objetos. Obtenido de [In06]	60
<i>Figura 4.14.</i> Ejemplo de test unitario.....	62
<i>Figura 5.1.</i> Módulos Abox.....	67

Índice de tablas

<i>Tabla 3.1.</i> Roles en Adapting S.L.....	38
<i>Tabla 3.2.</i> Relación entre las historias de usuario, tareas y casos de prueba.....	41
<i>Tabla 3.3.</i> Plan de entrega.....	43
<i>Tabla 3.4.</i> Iteraciones.....	44
<i>Tabla 4.1.</i> <i>Web Services</i> y <i>Client Object Model</i> . Obtenida de [Wh10].....	54
<i>Tabla 5.1.</i> Funcionalidades y características.....	68

1

Introducción

1.1. Motivación

En el mes de octubre del año 2010, empecé a realizar prácticas no curriculares en Adapting S.L. una empresa relativamente joven que en 2008 empezó el desarrollo de su nuevo gestor de contenidos empresarial, denominado Abox.

Durante el primer mes de trabajo mi tutor en la empresa me planteo la opción de realizar el proyecto fin de carrera. La idea me entusiasmó, ya que me atraía el reto de realizar dentro de una empresa el proyecto final de carrera, además de poder participar con el grupo de trabajo que estaba desarrollando el gestor de contenidos y todas las ventajas que esto me aportaba, como aprender las metodologías de trabajo que utilizan, conocer nuevas herramientas además de aplicar y mejorar los conocimientos obtenidos en la carrera.

Tras investigaciones preliminares de mercado, se identificó el colectivo de implantadores de Microsoft SharePoint como un grupo de elevado interés para Adapting S.L., debido a su cercanía con la problemática de la gestión documental y las carencias de la plataforma SharePoint. La gran cantidad de implantaciones llevadas a cabo con SharePoint permitirían a Adapting S.L. ampliar enormemente el mercado potencial, si logramos enlazar nuestro producto con SharePoint de forma transparente y sencilla.

La idea de poder vincular Abox con la plataforma de colaboración Microsoft SharePoint 2010 me ilusionó, ya que iba a poder desarrollar una gran funcionalidad que podía permitir a Adapting S.L. obtener un mayor mercado de clientes y ampliar las características de su producto.

Con esta nueva funcionalidad los usuarios de Abox podrán utilizar y gestionar los contenidos albergados en la plataforma SharePoint, como si estuvieran trabajando desde la misma, pero beneficiándose de las ventajas que ofrece Abox.

1.2. Objetivos

Abox tiene la filosofía sobre SharePoint de no intentar sustituirlo si no de complementar este último, con las características específicas que ofrece Abox en el campo de la gestión documental.

El principal objetivo de este proyecto, es que aquellas empresas que tengan un sistema un SharePoint instalado y decidan aplicar la solución Abox puedan consultar, guardar y editar la información que ofrece SharePoint desde la interfaz de Abox. De esta forma las dos plataformas pueden trabajar entre sí, mediante el vínculo creado desde Abox.

Abox está realizado mediante distintos módulos, algunos de los cuales cooperan entre si y otros realizan funciones únicas. Cada uno de estos módulos es como una pieza de un puzle, por tanto para conseguir nuestro objetivo vamos tener que desarrollar un nuevo módulo, o lo que es lo mismo, crear una nueva pieza del puzle y que esta a su vez se podrá comunicar con las otras piezas del puzle.

Para la consecución del objetivo principal, se han planteado otros objetivos inherentes, que son los siguientes:

- Respetar la política de permisos de SharePoint.
- Realizar búsquedas simples y avanzadas en ambos productos simultáneamente.
- Navegar por documentos de SharePoint desde la interfaz de Abox, respetando su estructura y política de permisos.
- Realizar ciertas acciones de gestión sobre carpetas y documentos de SharePoint desde la interfaz de Abox.
- Utilizar en el proceso de desarrollo de la nueva funcionalidad la metodología de software *extreme programming* [Be99].

Adicionalmente de estos objetivos, también encontramos otros objetivos personales a los que me voy a tener que enfrentar. El principal es el de superar el reto que supone aprender y comprender la plataforma de colaboración SharePoint, ya que nunca antes había trabajado con ella. De igual modo también me gustaría destacar los siguientes objetivos personales:

- Participar en el desarrollo de un proyecto real, y por tanto aprender el procedimiento habitual de la realización de un proyecto, conocer y participar en sus diferentes fases, plazos, y dinámica general de trabajo.
- Preparar un entorno de trabajo para un proyecto de desarrollo de varios meses de duración

- Poner en práctica los conocimientos adquiridos en la carrera.
- Ganar experiencia laboral y realizar el proyecto en una empresa perteneciente al sector de la gestión documental.
- Idear soluciones para todos los problemas que me vayan surgiendo con la integración de ambos productos

1.3. Beneficios esperados

Con la realización del proyecto, queremos conseguir los siguientes beneficios:

- Desarrollando el nuevo módulo de software se pretende que mejoren las prestaciones funcionales que actualmente dispone Abox.
- Los futuros clientes podrán aprovechar los desarrollos anteriormente realizados en SharePoint de una forma sencilla y transparente al usuario.
- Que haya una mayor interacción entre el usuario y la plataforma, consiguiendo por ello un mayor grado de satisfacción por parte del cliente.
- Que se consiga un sistema útil y fiable y que además sirva para agilizar las tareas del cliente.
- Ampliar la cuota de mercado especialmente en el mercado latinoamericano en donde actualmente Adapting S.L. está realizando sus mayores esfuerzos por expandirse.

1.4. Formación para el proyecto

Durante el mes de Octubre Adapting me comunico que el mes siguiente tenía la posibilidad de participar en un curso de formación sobre la plataforma de colaboración SharePoint 2010, mi primera impresión sobre el comentario fue muy buena, ya que hasta el momento no tenía conocimientos sobre la plataforma y necesitaba alguna formación previa para la realización del proyecto.

En el mes de noviembre empecé un curso de creación de portales corporativos con Microsoft Office SharePoint Server 2010, el cual fue realizado en el Instituto Tecnológico de Informática, situado en las instalaciones de Universidad Politécnica de Valencia.

Durante 3 semanas pude disfrutar de una oportunidad para poder formarme en la plataforma. El curso estaba dividido en varios volúmenes y estaba abierto a todo tipo de preguntas y sugerencias.

El primer volumen fue una introducción a la plataforma de trabajo colaborativo, los objetivos que se pretenden utilizando SharePoint y capacidades de la plataforma. Por otra parte también destacaron los distintos tipos de licencias que podemos encontrar en SharePoint y los requisitos del mismo. Acto seguido se realizó una instalación desde 0 de un sistema SharePoint con todos sus componentes. Esto me vino muy bien ya Adapting S.L. no disponía de ningún servidor con SharePoint y para realizar el proyecto necesitaba realizar una instalación en algún servidor de la empresa. Los siguientes volúmenes se trataron temas de arquitectura de SharePoint, y distintas partes de la administración. A medida que nuestros conocimientos sobre la plataforma fueron creciendo se realizaron diversos ejercicios planteados por el profesor, en los cuales teníamos que realizar distintas acciones dentro del portal. Finalmente también se realizó una pequeña aplicación utilizando la API de la red social Twitter, la cual fue la conclusión del curso.

1.5. Estructura del documento

El presente proyecto se ha estructurado en cinco capítulos cuyos contenidos se resumen a continuación.

En el capítulo 2 se presenta a la empresa Adapting S.L. y los detalles que la rodean como su organigrama, productos y servicios que ofrece. También se prestara atención a la metodología de desarrollo utilizada y finalmente se aporta información sobre las tecnologías que utiliza y una descripción detallada de cada una de ellas.

En el capítulo 3 se expone información utilizada y generada mediante la aplicación de la metodología de desarrollo. Por último se explican detalles sobre el prototipo que se realizó.

En el capítulo 4 comenzamos explicando la arquitectura de la aplicación, las herramientas y las tecnologías empleadas para el desarrollo de la aplicación, razonando el

uso de cada una de ellas. Tras esto se comentaran las pruebas utilizadas y por ultimo las buenas prácticas empleadas.

En el capítulo 5 se plantean y valoran los resultados obtenidos, se determinan los objetivos alcanzados y se exponen las posibles líneas de desarrollo futuro. Además de valorar las recomendaciones realizadas por el ITI durante el proceso de desarrollo.

Finalmente a modo de información adicional y con la intención de completar el contenido expuesto en los capítulos anteriores, se incluyen varios anexos. El primero de ellos incluye un manual del usuario explicando con detalle aquellas operaciones que pueden realizarse con el nuevo módulo de Abox. El Segundo anexo incluye información utilizada en la gestión del proyecto: historias de usuario, tareas de ingeniería y casos de prueba.

2

El entorno

En este capítulo se introduce una visión general de Adapting S.L., así como los productos que ofrece, la estructura organizativa y los clientes. Además de esto se expone con detalle la metodología utilizada y las tecnologías empleadas.

La estructura de este capítulo es la siguiente. En la sección 2.1 se exponen temas como la estructura, los productos o los servicios que ofrece. En la sección 2.2 se expone con detalle la metodología de software utilizada. Finalmente en la sección 2.3 se exponen las dos tecnologías empleadas en este proyecto, Abox y SharePoint.

2.1. La empresa: Adapting S.L.

La misión de Adapting S.L. es integrar, desarrollar e implantar sistemas de información, con tecnología web, en organizaciones que requieran una gestión avanzada de información y conocimientos, mediante proyectos a medida que garanticen la rentabilidad y la confianza a largo plazo de clientes y colaboradores.

2.1.1. Historia de la empresa

Adapting, S.L. es una PYME, creada en Valencia en Febrero de 1999, que ofrece servicios de consultoría y desarrollo de aplicaciones a medida, todo ello con un alto componente de Investigación y Desarrollo. Se constituyó con un capital social de 12.000 Euros, ampliado posteriormente a 42.000 Euros y repartido entre sus socios, todos ellos jóvenes emprendedores con un importante bagaje académico y profesional.

Los socios fundadores, antiguos compañeros de colegio, Rubén Lafuente, Luis Llorens y Luis Blasco, asumieron la dirección de las diferentes áreas de la empresa en función de la formación académica y la experiencia profesional previa de cada uno.

En el año 2008 Adapting S.L. incorporó al inversor privado Pandora Lys¹ a su capital social con objeto de obtener los recursos financieros necesarios para dar el salto al mercado nacional e internacional. El equipo humano y un producto claramente orientado a medianas y pequeñas empresas de ámbito nacional e internacional, convenció a Pandora Lys a participar en el proyecto de Adapting S.L. Gracias a esta inversión y al apoyo de un

¹ Los servicios que ofrece Pandora Lys los podemos encontrar en <http://www.pandalys.com/>

gran equipo hace unos meses se inauguró la primera sede en Colombia, denominada Adapting América.

Actualmente cuenta con dos departamentos, LAB y ONE. El primero de ellos, LAB, está especializado en nuevas tecnologías informáticas, que nos permite disponer de productos y soluciones de software de desarrollo propio basado en las últimas tecnologías, además de ser los desarrolladores del gestor de contenidos Abox. Por otra parte el segundo departamento, ONE, cuenta con una amplia experiencia en el desarrollo de software en tecnología web, integración de sistemas e implantación de soluciones de gestión Abox, se dedica al desarrollo de soluciones a medida y al posterior mantenimiento de aplicaciones.

En la figura 2.1 se puede observar la representación gráfica de la estructura de la empresa, en ella se representan los diferentes departamentos así como las relaciones jerárquicas que existen entre ellos.

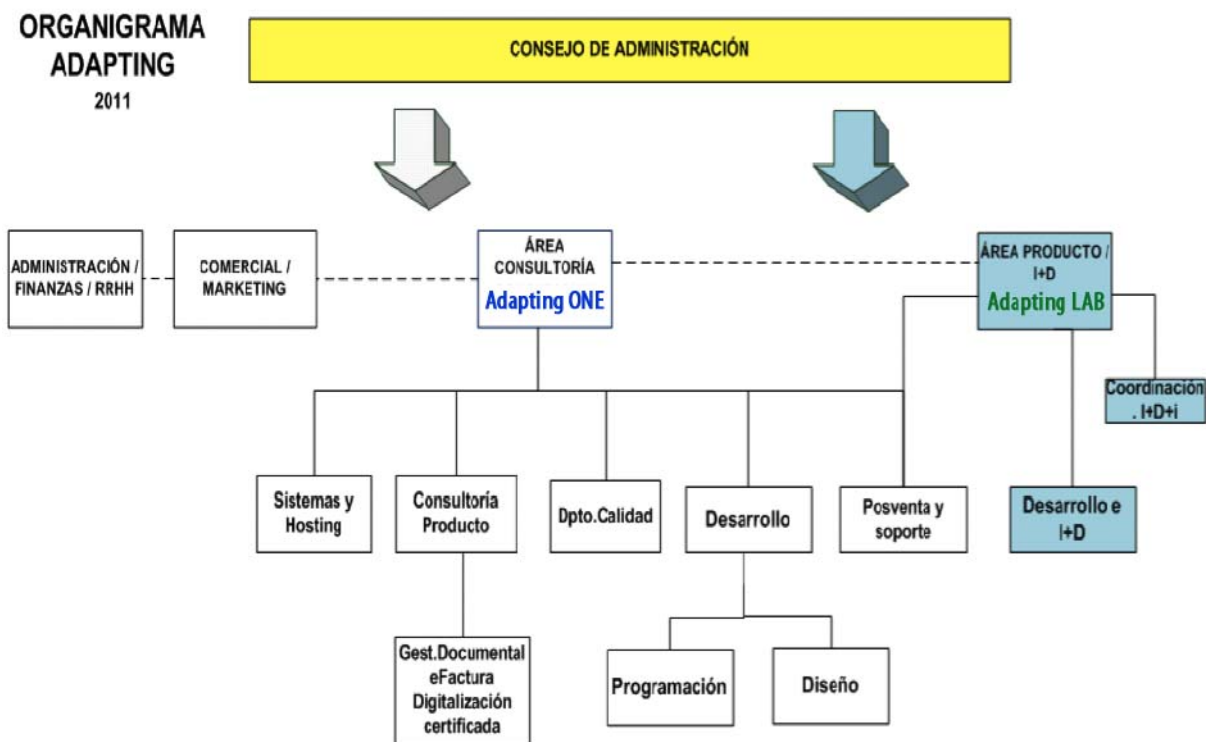


Figura 2.1. Organigrama vertical de Adapting S.L.

2.1.2. Estructura de la organización

La organización funcional se estructura en diferentes áreas que agrupan las distintas líneas de negocio de la empresa y los servicios auxiliares:

- **Área de gestión documental:** Es el área de mayor crecimiento de la empresa. Implantan sistemas de gestión documental para la optimización del volumen y flujo de documentos en cualquier tipo de organización.
- **Área de e-learning:** Área que pivota sobre los productos de formación *on-line* con clientes activos en temas de formación presencial y a distancia.
- **Área de servicios técnicos:** Incluye servicios accesorios prestados a los clientes, como alojamiento de aplicaciones web, reservas de dominios, etc.
- **Área comercial:** Gestiona la política comercial y de marketing de la empresa.
- **Área financiera-administrativa:** Gestiona la política financiera y de recursos humanos de la empresa.
- **Área de I+D:** Sus responsabilidades se concretan en los siguientes puntos:
 - Desarrollo de proyectos para clientes y servicios de post-venta.
 - Diseño de nuevos productos, así como actualización de los ya existentes.
 - Control y ensayo de las características técnicas en el diseño y en la Programación.
 - Gestión y control de los proveedores y subcontratas.
 - Actualización del *know how* de la Empresa.
 - Asistencia al Área Comercial en materias relacionadas con las reclamaciones de tipo técnico que se pudieran presentar dentro de su área.
 - Asistencia al Área Comercial en captación y mantenimiento de clientes.
 - Participación en proyectos de I+D nacionales y europeos.
- **Área de procesos y calidad:** Sus responsabilidades se concretan en:
 - Gestión y control de los proyectos de los clientes.

- Actualización del *know how* de la Empresa.
- Asistencia al Director Comercial en materias relacionadas con las reclamaciones de tipo técnico que se pudieran presentar dentro de su área.
- Colaboración con el departamento de desarrollo en la definición y transformación de los procesos relacionados con el desarrollo de aplicaciones.
- Supervisión del Comité de Calidad

Para el desarrollo de proyectos el organigrama vigente es el que se muestra en la figura 2.2:

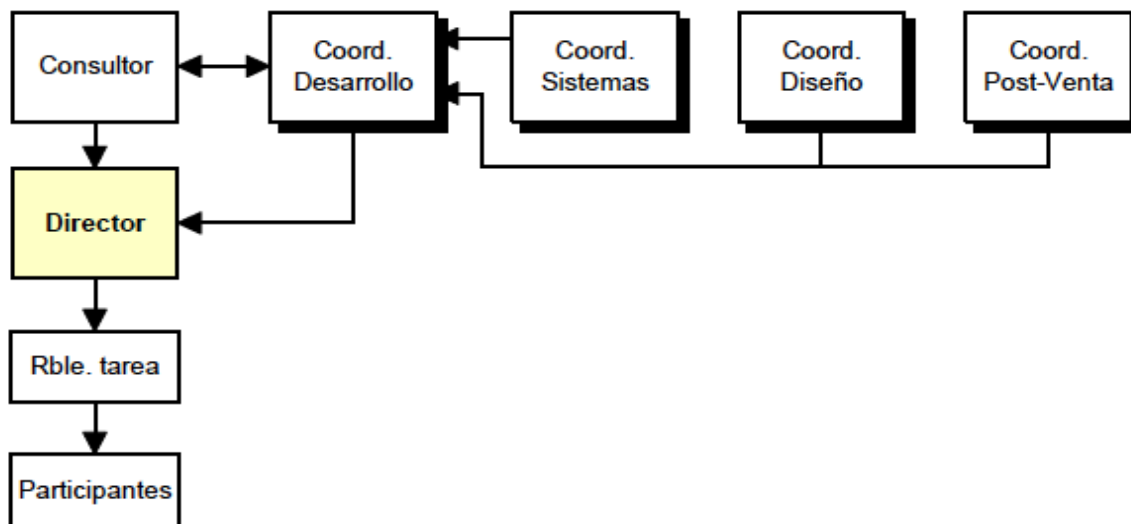


Figura 2.2. Organigrama de desarrollo de proyectos

El Director del proyecto tiene la responsabilidad única sobre el buen desarrollo del proyecto. Sin embargo, el director está supeditado a las decisiones de dos figuras que colaboran en el proyecto sólo de forma parcial:

- El consultor o *project manager*, que es el que tiene el contacto con el cliente.
- El coordinador de desarrollo, que conoce las metodologías internas, y decide sobre las técnicas utilizadas.

2.1.3. Productos y servicios

Los productos y servicios que oferta Adapting S.L. abarcan dos grandes necesidades:

- **Información y contenidos:** Se trata mediante la plataforma de gestión de contenidos empresariales Abox, la cual permite gestionar grandes cantidades de información y los diferentes contenidos digitales de las organizaciones: documentos, catálogos, webs, comunicación y más...
- **Personas y conocimientos:** Soluciones a medida para facilitar el trabajo entre personas y la gestión del conocimiento: proyectos, portales corporativos, *e-learning*, procesos y más...

El catálogo de los servicios prestados por Adapting ONE se presenta a continuación:

- Servicios de consultoría para selección, especificación e implantación de soluciones de negocio electrónico, gestión documental y facturación electrónica
- Servicios de análisis y programación de aplicaciones avanzadas para Internet e intranet, a partir de las últimas tecnologías.
- Servicios de creatividad y diseño de interfaces y aplicaciones multimedia (presentaciones, CDs corporativos, sitios web), combinando siempre el atractivo visual con la funcionalidad y usabilidad necesarias.
- Servicios posventa de mantenimiento y soporte de las aplicaciones desarrolladas por Adapting S.L.
- *Hosting* o alojamiento web
- Servicio de correos electrónicos

2.1.4. Clientes

En el siguiente listado podemos encontrar los clientes más importantes de Adapting S.L.:

- | | |
|-----------------------------|----------------------------|
| ▪ Generalitat Valenciana | ▪ Ayuntamiento de Valencia |
| ▪ Canal 9-RTVV | ▪ Grupo NATRA |
| ▪ Grupo Opel Palma | ▪ Ayuntamiento de Torrente |
| ▪ Ayuntamiento de Alicante | ▪ EuroUniforms |
| ▪ ETRA | ▪ Hurtado Muebles |
| ▪ Ayuntamiento de Castellón | ▪ Ayuntamiento de Alicante |
| ▪ FutbolCity | ▪ Iberodoc |
| ▪ Redita | ▪ Viva Europa |

Además de este listado de empresas, Adapting S.L. cuenta con otro gran número de empresa con las que coopera ya sea en proyectos de una envergadura más pequeña o mantenimiento de productos.

2.1.5. Plan de I+D+I

Adapting S.L. se creó con la premisa de prestar servicios innovadores, además de una filosofía de negocio sencilla, pero eficaz: trabajar junto a nuestros clientes, diseñando con ellos las soluciones que mejor se adapten a sus necesidades, utilizando para ello la tecnología más idónea, filosofía que se está cumpliendo escrupulosamente en la actualidad.

Adapting S.L. dispone de un departamento de I+D propio que provee de soluciones avanzadas e innovadoras a las distintas áreas de negocio de la empresa. La Dirección cree firmemente que, dentro de un contexto empresarial muy competitivo, las inversiones en I+D son la única forma de obtener una diferenciación, asegurar el futuro de la empresa y poder satisfacer plenamente las demandas del cliente.

En un sector tan dinámico como el de las aplicaciones tecnológicas, es aventurado hablar de proyectos concretos en un horizonte superior al año. No obstante, Adapting S.L. sí dispone de unas líneas básicas de actuación para los próximos años:

- Creación de una comunidad de desarrolladores de Abox, impulsando el desarrollo

de *plugins* o extensiones por parte de los socios o desarrolladores externos.

- Utilización de herramientas de software libre.
- Utilización de HTML5 para el desarrollo en nuevas plataformas como Android o Iphone.
- Apuesta por el *cloud computing*. Permitiendo ofrecer a los clientes servicios estandarizados de forma flexible.
- Mejoras en la seguridad de las aplicaciones y los servicios ofertados.

2.2. Metodología de desarrollo de software: XP

Cada proyecto de software es diferente al resto, es por ello que en una empresa en la que distintas personas participan en muchos proyectos es necesario adoptar una metodología o lo que es lo mismo, establecer un marco de trabajo para estructurar, planificar y controlar el proceso de desarrollo.

Durante los últimos 3 años Adapting S.L. ha estado poniendo en práctica la disciplina del desarrollo de software *extreme programming*, desarrollada por Kent Beck² [Ke11], centrada en potenciar las relaciones interpersonales como clave para el éxito del desarrollo de software, preocupándose por el aprendizaje de los desarrolladores, promoviendo el desarrollo del trabajo en equipo y proporcionando un buen clima de trabajo. Kent Beck define *extreme programming* como “*una forma de desarrollar software: liviana, eficiente, de bajo riesgo, flexible, predecible científica y divertida*” [Be99].

Extreme programming se basa en 5 pilares [Ar10] fundamentales que son la comunicación, simplicidad, retroalimentación y el valor. Aunque también posteriormente se ha añadido otro pilar que es el respeto. A continuación describimos cada uno de ellos en detalle:

- **Comunicación:** está presente en todas las prácticas. *Extreme programming* declara que la mejor forma de comunicación entre los desarrolladores y el clientes es cara a cara. Esto permite evitar parte de la documentación escrita que se utiliza en las comunicaciones con el cliente por parte de los desarrolladores.
- **Simplicidad:** se simplifica el diseño para agilizar el desarrollo y facilitar a su vez el mantenimiento. Este demostrado que un diseño complejo de código junto a

² Kent Beck y Ward Cunningham son los creadores y principales impulsores de Extreme Programming.

sucesivas modificaciones por parte de deferentes desarrolladores hacen que la complejidad aumente exponencialmente. Es por ello que para mantener la simplicidad del código es necesario la refactorización, esta es la manera de mantener el código simple a la vez que el proyecto va evolucionando.

- **Retroalimentación:** En la disciplina *extreme programming* el cliente está integrado dentro del desarrollo del proyecto, lo que permite conocer a tiempo real su opinión. Además al aplicarse ciclos muy cortos de desarrollo, los resultados son visibles cada poco tiempo, lo que permite que se minimizan rehacer las partes del código que no cumplen con los requisitos.
- **Valor:** requiere que los desarrolladores vayan a la par con el cambio. Los cambios de requisitos son inevitables en todo proyecto de software, pero hay que estar preparado para aceptarlos.
- **Respeto:** Este punto se añadió en una segunda revisión que se realizó a la metodología. Los miembros del equipo respetan su trabajo porque siempre están luchando por la alta calidad en el producto y buscando el diseño optimo o más eficiente, pero además de esto, los miembros del equipo respetan el trabajo del resto, orientándolos a realizar mejor su trabajo y obteniendo como resultado una mejor autoestima en el equipo y elevando el ritmo de producción en el equipo.

Las 4 actividades [Vp10] más habituales en las que participan todos o algunos de los miembros del equipo en esta metodología son las siguientes:

- **Escuchar:** Como habíamos comentado anteriormente uno de los pilares fundamentales de la *extreme programming* es la comunicación. Con menos documentación se hace necesaria una comunicación de muy alta calidad, ya que los desarrolladores no saben del negocio del sistema en construcción y este conocimiento debe ser transmitido muy claramente por el cliente.
- **Probar:** No se puede estar seguro de algo a menos que se lo pruebe, es por esto que la *extreme programming* impulsa el desarrollo de pruebas antes de realizar la codificación del sistemas. De esta manera se puede estar seguro que lo que se entrega realmente funciona sin errores.
- **Codificar:** El código es la parte más importante en el desarrollo de un sistema, es por esto que debe ser entendible por todo el equipo, es decir, un miembro con el hecho de solo leer el código puede entender la lógica y los algoritmos que se elaboran, no se habla de comentarios, quiere decir que el código debe ser comunicativo.
- **Diseñar:** Debemos crear una estructura de diseño que organiza la lógica del sistema, por medio de esto se eliminan dependencias, de manera que si existen cambios no afectarán a todo el sistema

2.2.1. Practicas

La *extreme programming* se caracteriza también por incluir 13 practicas básicas [Ca03], las cuales podemos ver relacionadas entre sí en la figura 2.3. Estas promueven y garantizan el éxito de un proyecto, pero a continuación comentaremos tan solo aquellas que son aplicadas en el desarrollo de proyectos de Adapting S.L.

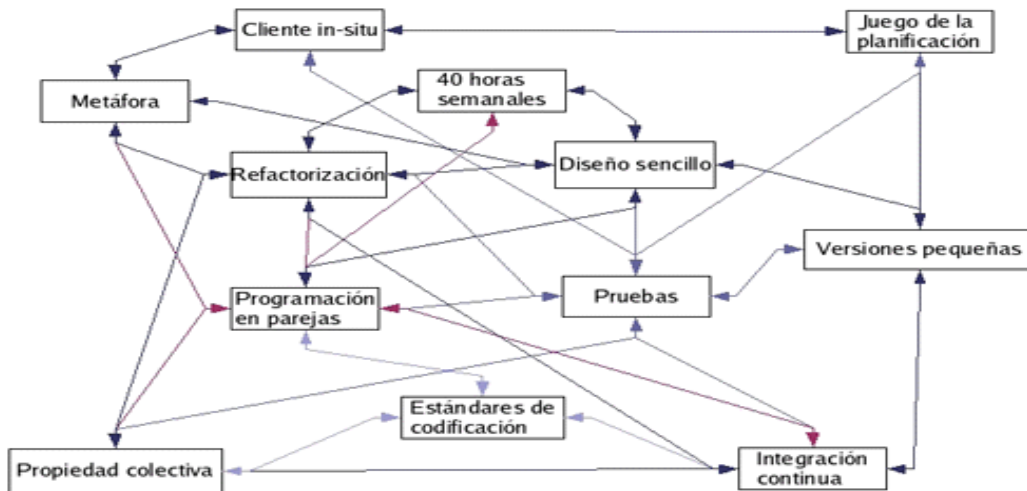


Figura 2.3. Buenas prácticas. Obtenida de [Be99]

- **Juego de planificación:** es la práctica que define la forma de trabajar, que está compuesta por la planificación de la *release* y la planificación de la iteración. En la figura 2.4 se puede observar un pequeño resumen de las mismas:

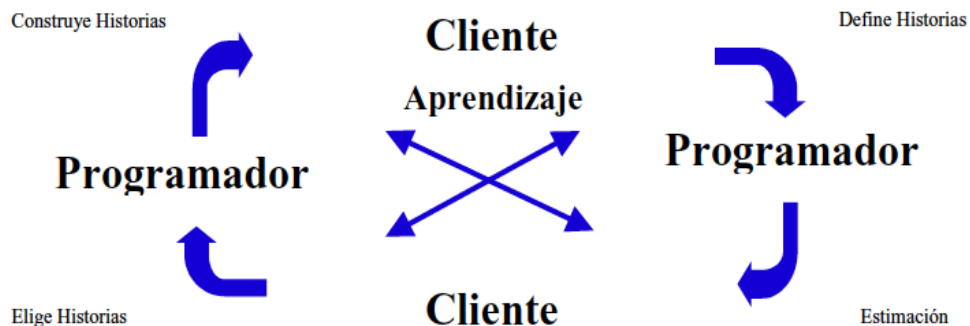


Figura 2.4 El juego de la planificación

En la planificación de la *release*, se define lo que se quiere tener como producto en los próximos 3 meses aproximadamente. El cliente define un gran número de requerimientos, llamados historias de usuario, que expresan requisitos funcionales y no funcionales del sistema. Una vez definidas un conjunto de historias de usuario, estas son analizadas y estimadas por un grupo de programadores, para que finalmente el cliente las ordene en función de su valor. Una vez ordenadas se proceden a elegir aquellas que no superen el tiempo de la *release*.

En la planificación de la iteración, se definen las actividades para las 3 o 4 próximas semanas. El cliente elige el conjunto de historias de usuario de mayor valor para que sean implementadas, a continuación los programadores dividen estas historias en tareas más pequeñas denominadas tareas de Ingeniería y se realiza una estimación de su tiempo de desarrollo. Finalmente el cliente ordena en función de sus necesidades las historias de usuario, dejando para iteraciones posteriores aquellas que sobrepasen la capacidad productiva de la iteración.

Durante el desarrollo de la iteración y al final de cada una de ellas, se realiza un seguimiento del plan de iteración y de la *release*, en las cuales es de esperar situaciones como la aparición de nuevas historias de usuario, modificación de las mismas o no finalización de la historia de usuario en la iteración planificada.

- **Pruebas:** La ejecución automatizada de tests es un elemento clave de la metodología *extreme programming*. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema, por otra parte los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse.

Uno de los objetivos principales de los tests no es corregir errores, sino prevenirlos. Elaborar los tests antes de escribir el código exige pensar por adelantado cuáles son los problemas más graves que se pueden presentar, y cuáles son los puntos dudosos. Esto evita muchos problemas y dudas, en lugar de dejar que aparezcan "sobre la marcha".

- **Versiones pequeñas:** La idea es producir rápidamente versiones del sistema tan pequeñas como sean posibles, aunque deben añadir suficiente valor como para que resulten valiosas para el cliente. Una entrega no debería tardar más 3 meses

- **Diseño sencillo.** El objetivo es utilizar el diseño más sencillo que consiga que todo funcione³. Se evita diseñar características extra porque a la hora de la verdad la experiencia indica que raramente se puede anticipar qué necesidades se convertirán en reales y cuáles no.

Kent Beck define [Be99] el diseño simple como aquel:

- Pasa todos los tests.
 - No contiene código duplicado.
 - Deja clara la intención de los programadores en cada línea de código.
 - Contiene el menor número posible de clases y métodos.
- **40 horas semanales:** Se debe trabajar un máximo de 40 horas por semana. Esto es debido a que el trabajo extra desmotiva al equipo, es por ello que hay que intentar que no se trabajen horas extras en dos semanas seguidas. Si esta situación se produce repetidamente quizás esté ocurriendo un problema que deba corregirse.
 - **Propiedad Colectiva:** En un proyecto de *extreme programming* cualquier programador puede mejorar cualquier porción de código, en cualquier momento. Lo cual resulta en una mayor calidad de código, en un menor número de defectos, contribuye con nuevas ideas y ayuda a compartir el conocimiento con todo el equipo, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.
 - **Refactorización:** es una actividad constante que consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, legible y no duplicar el código. La refactorización también se utiliza cuando resulta conveniente modificar código existente para hacer más fácil implementar nueva funcionalidad.
 - **Cliente in-situ:** Uno de los requerimientos de la metodología *extreme programming* es tener al cliente disponible durante todo el proyecto. Es por ello que el cliente formara parte del equipo.

³ En la metodología *extreme programming* este concepto se define como “Do the simplest thing that could be possibly work”.

Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo aportando mayor valor de negocio y para que los programadores puedan resolver de manera inmediata cualquier duda asociada.

La forma de comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada.

- **Estándares de codificación:** Los estándares de codificación mantienen el código legible para los miembros del equipo, facilitando los cambios. El objetivo es que el código del sistema se vea como si fuera escrito por una única persona muy competente.

Según Kent Beck, todas estas prácticas interactúan y se refuerzan mutuamente, de manera que el éxito del proceso estará dado por la puesta en marcha de todas las practicas [Be99].

2.2.2. Roles

Un principio básico de una metodología ágil, como es la *extreme programming*, es tener un número pequeño de roles. Los siguientes son los roles básicos que definió Beck en su propuesta original:

- **Programador:** es el encargado de escribir las pruebas unitarias, producir el código del sistema y de la toma de decisiones técnicas.
- **Cliente:** en *extreme programming* el cliente juega un papel fundamental y forma parte del equipo. El cliente será el encargado de escribir las historias de usuario y las pruebas funcionales para validar su implementación. Además asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- **Gestor:** también conocido como *big boss* o *manager*. Es el encargado de la coordinación, debe ser el vínculo entre clientes y programadores, además de esto también será el encargado de crear un ambiente que facilite las actividades del equipo de desarrollo.
- **Encargado de pruebas:** también conocido como *tester*. Proporcionará ayuda al cliente a escribir las pruebas funcionales, pero también ejecutara las pruebas regularmente, difunde los resultados en el equipo.
- **Encargado de seguimiento:** también conocido como *tracker*. Realiza el

seguimiento de las estimaciones realizadas y el tiempo que efectivamente se ha dedicado para mejorar futuras estimaciones. Analiza el avance de cada iteración y proporciona retroalimentación al grupo.

- **Entrenador:** Conocido como *coach*. Guía y vigila que se sigan correctamente las prácticas de *extreme programming*, es responsable del proceso global.
- **Consultor:** Especialista en algún tema específico, es una persona externa al grupo de trabajo. Su función es la de guiar al equipo para resolver algún problema.

De todos estos roles, los dos que destacarían serían el de entrenador y el tracker. Para elegir el entrenador de un grupo se escoge a una persona que ha cumplido roles como programador, jefe o arquitecto del sistema. El papel del entrenador es promover que todo el mundo tome sus propias decisiones, ayudar a los programadores con habilidades técnicas, explicar los procesos directivos de nivel superior y asegurarse que el grupo se sienta a gusto.

2.2.3. Artefactos

Las historias de usuario son la técnica utilizada en la metodología *Extreme Programming* para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el usuario describe la funcionalidad deseada desde la perspectiva del cliente. Los componentes básicos de una Historia de usuario se pueden resumir en tres elementos:

- **Tarjeta:** es la descripción escrita de la historia, que sirve como identificación, recordatorio y también ayuda a planificar.
- **Conversación:** es el núcleo de la historia; es el diálogo que ocurre con los usuarios, notas, grabaciones, prototipos y documentos.
- **Pruebas de Aceptación:** Son los criterios que el usuario va a utilizar para confirmar que la historia fue terminada.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Beck [Be99] afirma que la única información que se debe recoger es el nombre de la historia y una descripción de la misma.

Pero existen una gran cantidad de plantillas, algunas aconsejadas por otros autores y rebatidas por otros.

Un ejemplo de la plantilla de Historia de usuario lo podemos apreciar en la figura 2.5. El número de la historia de usuario y el nombre se utilizan como identificadores de la misma, el resto de campos son utilizados para anotar las observaciones de la misma. La prioridad del negocio es definida por el cliente, de esta manera se declaran aquellas tareas que tienen mayor valor, en cambio el riesgo de desarrollo es definido por los programadores al igual que los puntos estimados, los cuales pueden servir para comprar el esfuerzo de desarrollo entre las historias de usuario. El Equipo debe discutir y debatir cada uno de las estimaciones, y llegar a un acuerdo sobre los puntos de una historia de usuario.

<i>Historia de usuario</i>	
Número:	Usuario:
Nombre historia:	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados:	Iteración asignada:
Programador responsable:	
Descripción:	
Observaciones:	

Figura 2.5 Modelo propuesto para una historia de usuario. Obtenido de [Ca03]

Una buena historia de usuario, tiene que cumplir el modelo INVEST: Independiente, valiosa, negociable, estimable, pequeña (*small*) y testable.

- **Independiente:** una historia de usuario debe ser independiente de las otras.
- **Negociable:** La tarjeta tan solo incluye una pequeña descripción de la historia, los detalles se añaden mediante la conversación.
- **Valiosa:** Cada historia de usuario tiene que tener un valor para el cliente.
- **Estimable:** El equipo necesita poder estimar cada una de las historias de usuario.
- **Pequeña:** Una buena historia debe ser pequeña en esfuerzo.

- **Testable:** Una historia de usuario necesita poder probarse y saber que la historia de usuario

<i>Tarea de ingeniería</i>	
Número tarea:	Historia de usuario :
Nombre tarea:	
Tipo de tarea:	Puntos estimados:
Fecha inicio:	Fecha fin:
Fecha inicio:	
Programador responsable:	
Descripción:	

Figura 2.6 Modelo propuesto para una tarea de ingeniería. Obtenido de [Ca03]

<i>Caso de prueba de aceptación</i>	
Código:	Historia de usuario :
Nombre:	
Descripción:	
Entrada / Pasos de ejecución:	
Resultados esperados:	
Evaluación de la prueba:	

Figura 2.7. Modelo propuesto para una prueba de aceptación. Obtenido de [Ca03]

Una vez tenemos las historias de usuario redactadas, podemos definir las tareas relacionadas con cada historia. En la figura 2.6 se muestra un ejemplo de la tarjeta de tareas de ingeniería, las cuales son redactadas por los programadores.

Cada historia de usuario a su vez debe tener alguna prueba de aceptación. Estas indican al desarrollador y al cliente que la tarea ha sido completada. En la figura 2.7 se expone la plantilla utilizada para los casos de prueba, la cual tendrá que ser redactada por el cliente pero con la ayuda de la persona encargada de las pruebas en el equipo.

2.3. Tecnología

El uso de la tecnología dentro de la empresa es de suma importancia y proporciona un gran valor al negocio. Por ello en Adapting S.L. decidió a parte de aprovechar la tecnología que se encuentra en su entorno, desarrollar su propia tecnología, con los gastos humanos y económicos que esta decisión conlleva.

2.3.1. Abox

Adapting S.L. basada en su experiencia de más de once años en el desarrollo de sistemas de gestión de contenidos empresariales, es el fabricante de Abox, un gestor de contenidos empresarial, en tecnología 100% web, integrable con los ERPs más implantados del mercado.

Abox es la plataforma de gestión de contenidos empresariales idónea para todas las organizaciones, empresas medianas y grandes, que tienen la necesidad de gestionar de forma más eficiente la información no estructurada de cualquiera de sus áreas o departamentos.

Los contenidos que gestiona Abox son cualquier elemento de información en formato electrónico que pueda ser almacenado, que contenga información adicional de su tipo (Metadatos), que pertenezca a una jerarquía o que necesite permisos individuales.

Un ejemplo de los contenidos que gestiona son los siguientes:

- Documentos de texto
- Imágenes, vídeos, multimedia
- Facturas, albaranes, digitalizados
- Archivos de ofimática

- Registros, evidencias, contratos, etc.
- Catálogos digitales
- Páginas web
- Contenidos formativos

El diseño de Abox ha sido tutorizado por expertos en usabilidad de prestigiosos centros de investigación, un ejemplo de su apariencia se aprecia en la figura 2.8. Es tan sencillo de usar, como navegar por una página web.



Figura 2.8 Ventana principal de Abox

Ventajas de Abox sobre el uso de las carpetas de Windows:

- La asignación de permisos y la capacidad de auditoría es mucho más sofisticada y granular que en Windows.
- Puedes acceder a él simplemente disponiendo de un navegador y de conexión a Internet.
- Cada documento se caracteriza en esa base de datos con una ficha de Metadatos, que contiene mucha más información de la que te ofrece Windows.

- Abox Document no es sólo un repositorio de documentos en carpetas; es también una base de datos relacional.

2.3.1.1 Módulos de Abox

Núcleo de la plataforma de gestión de contenidos empresariales *ECM*.



Núcleo de la plataforma Abox.



Gestor documental avanzado y colaborativo para intranet e Internet.



Solución para la emisión y recepción de *e-facturas*. Digitalización certificada de facturas.



Editor de *workflows* documentales configurables para definir los ciclos de vida de los documentos y los flujos de trabajo.



Gestor de contenidos de sitios web *CMS*. Facilita la gestión de la estructura de los sitios web y la actualización de los contenidos de las páginas web.



Catálogo web de productos y portales de comercio electrónico.



Plataforma de *e-learning* para la gestión de formación *online*

Figura 2.9 Listado de módulos de Abox

2.3.1.2 Caso de éxito

A continuación se exponen un ejemplo en el cual la implantación de la plataforma Abox se realizó con un resultado satisfactorio y proporcionando grandes ventajas al cliente.

El caso que vamos a relatar es el del grupo Agdojavi, dedicado al transporte internacional del mueble, disponía anteriormente de una intranet-extranet que se nutria de diferentes sistemas y procesos internos que fueron implantados por Adapting S.L., en la figura 2.10 se aprecia la página web anterior que disponía el grupo Agdojavi.



Figura 2.10 Antigua web de AgdoJavi

La dirección de Agdojavi encargo a Adapting S.L. la renovación de dichos sistemas, con el objetivo principal de incrementar la productividad de sus procesos internos y de mejorar la comunicación con los clientes. Durante una primera fase el equipo de Adapting S.L. realizo un detallado análisis y una toma de datos, las cuales presentaron que la empresa también requiera de las siguientes necesidades:

- Reducción de costes (papel + envíos + gestión)
- Mejora de la imagen frente al cliente

- Facturación electrónica con Extranet para clientes
- Envío de documentación al cliente y registro de entrega
- Reconocimiento automático del CMR (integración Kofax)
- Integración con ERP (PICK)
- Aprovechamiento para gestión calidad (modo Intranet)



Figura 2.11 Nueva extranet de Abox para Agdojavi

Todos estas necesidades suponen la modernización tecnológica del grupo de forma que se incluyen mayores automatismos en la generación de documentos y contenidos puestos a disposición del cliente final, se modernizan las funcionalidades y los interfaces actuales, se aporta una mayor operatividad con una menor complejidad del uso de las aplicaciones y todo ello consiguiendo el objetivo principal de mejorar la productividad.

El entorno tecnológico que con el cual nos encontrábamos era un servidor dedicado, alojada dentro de la empresa, el cual ofrece un entorno de *Intranet/Extranet* que trabaja bajo un sistema operativo Windows 2000 Server y un SQL Server 2000. Además de esto la empresa dispone de un escáner Kodak I80 y un Kofax Ascent Capture. Tras una fase de parametrización y de revisión del entorno, se planteó al grupo Agdojavi la aplicación de la plataforma Abox con la siguiente solución técnica:

- *Abox Document*
- Pack de captura de datos (*Hot-Folder*)
- *Abox Invoice*

También se plantearon desarrollarlas siguientes soluciones a medida para completar las necesidades del grupo Agdojavi:

- Conector ERP
- Acceso a Extranet desde página web

Tras la aceptación por parte del grupo Agdojavi se procedió a realizar la instalación del sistema y la posterior carga y puesta en marcha de la aplicación. Finalmente se entregó documentación con los manuales de la aplicación. Tras ello la organización consiguió los objetivos marcados anteriormente además de los siguientes:

- Organización de la documentación del sistema de gestión de la calidad con *Abox Document*
- Imágenes digitalizadas de los *CMRs* firmados, así como documentos anexos a estos *CMRs* (copia de cheques, notas de entrega, ...), con información extraída con el programa *OCR* de reconocimiento de caracteres *KOFAX*
- Facturas electrónicas generadas a partir de los ficheros.

En resumen, el grupo Agdojavi alcanza una mejora de la productividad en procesos administrativos, vía la modernización tecnológica y la simplificación de alguno de los procesos y una mejora en el servicio y la imagen externa al cliente como se aprecia en la figura 2.11.

2.3.2. SharePoint

SharePoint es una plataforma de colaboración desarrollada por Microsoft que nos permite crear portales y sitios para disponer y compartir la información con todas las personas de una empresa u organización, es decir, los socios, empleados, clientes y colaboradores puedan compartir contenidos, formando parte del proceso de negocio de la empresa de una manera activa. En la figura 2.12 observamos la intranet de SharePoint.

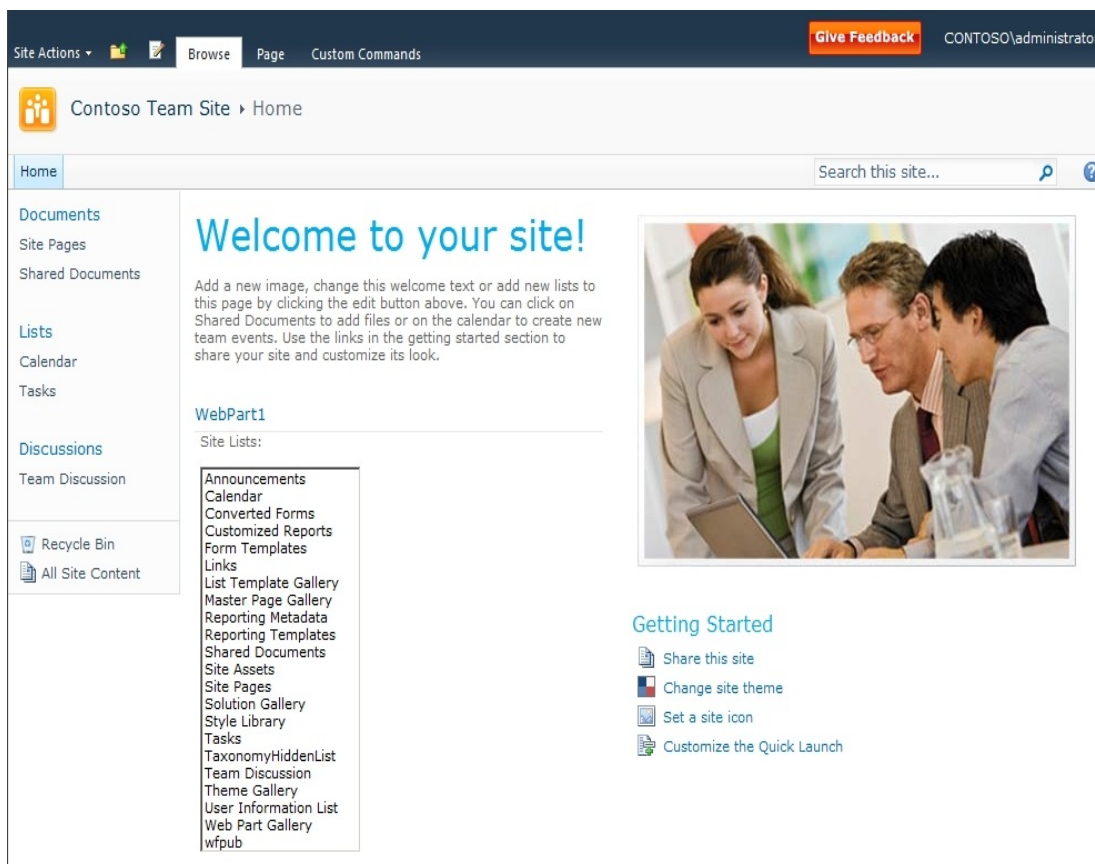


Figura 2.12 Intranet de SharePoint 2010

Gran parte del contenido es gestionado a través de la conocida interfaz Office, pueden ser documentos, hojas de cálculo, vídeos, ideas, conocimientos, gráficos, experiencias, fotos, etc. Microsoft define a su herramienta como “La plataforma de Colaboración para la Empresa e Internet”.

2.3.2.1. Historia de SharePoint

Mucho antes de la llegada de SharePoint⁴ [Ri06] existía un producto denominado Site Server [Si11], liberado por Microsoft en el año 1996, que contenía características de gestor de contenidos web, comercio electrónico, búsqueda y gestión e indexación de documentos. Su última versión fue liberada en 1998, correspondiendo a su versión 3.0. Después de esto Microsoft discontinuó el desarrollo de Site Server, dividiendo su funcionalidad.

Por una parte decidió crear el Microsoft Commerce Server [Mi11], la cual mantendría las partes de comercio electrónico y estadísticas. Mientras que por otro lado a comienzos del 2001 Microsoft compró el gestor de contenidos nCompass y lo relanzó bajo el nombre de Microsoft Content Management Server, este programa fue evolucionando durante distintos años, empezando como gestor de contenido, pero acabó compartiendo ciertas características con SharePoint en su versión 2003, lo que provocó que los grupos de desarrollo de SharePoint y Content Management Server se unificaran.

Pero antes de que ocurriera todo esto, a comienzos de 2001 Microsoft lanzó la primera versión de Tahoe Beta 2 bajo el nombre comercial de SharePoint Portal Server 2001. Este producto incluía características de Gestión de documentos, Indexación. El desarrollo posterior de la versión de SharePoint Portal Server 2001 fue sustituir el almacenamiento web por SQL Server, pero también mejoraron la escalabilidad y las características del portal.

También en el año 2001, con la liberación Microsoft Office XP se liberó SharePoint Team Services o también llamado STS, era un componente gratuito que proporcionaba un portal web y estaba disponible como parte de Microsoft FrontPage. STS se podía ejecutar en Windows 2000 Server o Windows XP.

En la siguiente revisión de STS, se rediseñó el producto por completo y se le cambió el nombre del producto por Windows SharePoint Services 2.0 o también conocido como WSS, esto fue debido a que el grupo de desarrollo del WSS se unió con el grupo de SharePoint Service a mediados del año 2002, esto provocó que las posteriores versiones de WSS siempre han ido ligadas con las nuevas apariciones de SharePoint.

En el año 2003, se lanzó la siguiente versión de Microsoft Office 2003 además de la siguiente versión de Microsoft SharePoint 2003 y la versión anteriormente comentada SharePoint Team Services 2.0.

⁴ Grafico detallado de la historia de SharePoint
http://www.joiningdots.net/downloads/SharePoint_History.jpg

SharePoint 2003 se alejó de la gestión de documentos y se centró en las capacidades de portal, pero el crecimiento continuo por parte de otras compañías de portales y la demanda por parte de las empresas dejó claro que los portales y la gestión de contenidos web estaban en curso de colisión, con la gestión de documentos.

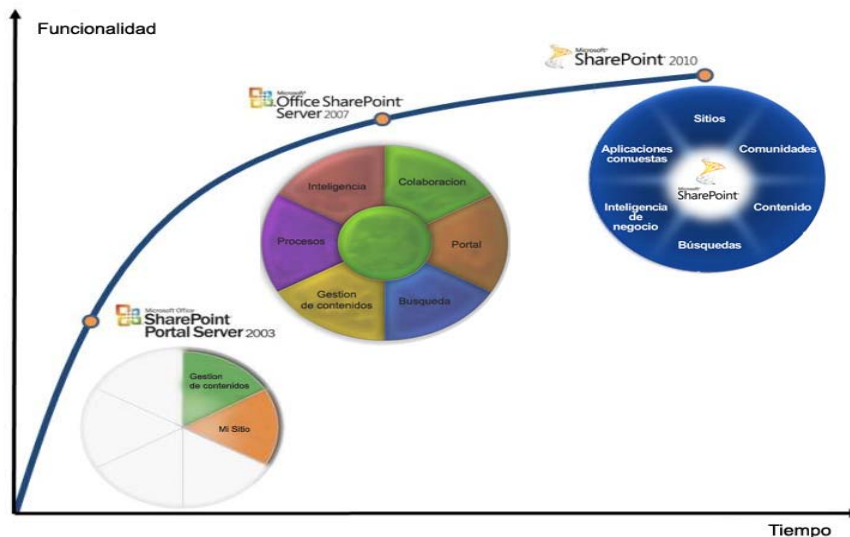


Figura 2.13 Características y funcionalidades en SharePoint

Como ya habíamos comentado anteriormente, en el año 2004 los equipos de desarrollo de Microsoft SharePoint y Microsoft Content Management Server se unificaron, con ello los elementos web con ASP.NET comenzaban a tomar vida propia. La unión de estos dos grupos de trabajos provocó el desarrollo en 2007 de la tercera versión de Microsoft Office SharePoint Server 2007 o también conocido como MOSS 2007, la cual tenía su versión gratuita conocida como Windows SharePoint Services 3.0.

Antes del año 2010, fueron liberadas las primeras versiones beta del que sería el actual producto Microsoft Office SharePoint Server 2010 o también conocido como MOSS 2010. Su versión gratuita cambió de nombre, anteriormente conocida como Windows SharePoint Service pasó a denominarse Microsoft SharePoint Foundation.

A modo de resumen se puede observar en la figura 2.13, la evolución de las últimas versiones de SharePoint.

2.3.2.2. Ventajas e inconvenientes de SharePoint

SharePoint es ideal para personas que no son muy exigentes sobre la gestión documental, ya que permite que cualquier persona pueda realizar tareas básicas con los flujos de trabajo, gestión de proyectos básicos e incluso generar portales que tengan una gran integración con los productos de Microsoft.

Es una herramienta que requiere muy poco entrenamiento por parte del usuario para poder ser utilizada, pero para ser empleada con la mayor eficiencia requiere de un gran conocimiento de las tareas y objetivos que el propio usuario realiza en la empresa. Además de esto, la colaboración entre el personal que utiliza el producto es fundamental, es por ello que la empresa que invierte en una solución como SharePoint, además de formar al personal en el entrenamiento de la plataforma también tiene que formar al personal explicando la forma de colaborar entre ellos.

El sistema es flexible a la hora de crear usuarios, grupos o modificar permisos, todo ello gracias a un panel de administración desde el cual se pueden gestionar un gran número de acciones, pero el conocimiento y la experiencia que se requiere para realizar estas acciones tiene que ser mayor. Es por ello que quizás algunas de estas tareas son encargadas a personas más cualificadas o incluso a personal externo de la empresa, con el sobrecoste que esto puede producir.

Algo parecido ocurre con la instalación y configuración inicial del producto, la cual es una tarea muy tediosa. En muchas ocasiones se contrata a personal externo, especializado en TI para ayudar en el proceso de iniciación del producto dentro de la empresa.

Para finalizar habría que destacar los siguientes puntos:

- Requiere una gran inversión en licencias, tanto por parte del producto SharePoint, como del SO y SQL.
- La seguridad es una parte muy importante y por ello requiere un mantenimiento por parte de un administrador de sistemas.
- Extensibilidad, ciertas desarrollos a medida que se realicen para una versión concreta del producto, no garantizan que en las siguientes versiones sean compatibles, poniendo en peligro gran parte de la inversión realizada.

- Actualmente algunos proveedores empiezan a dar posibilidad de contratar este producto en la nube.

2.3.2.3. Principales funcionalidades de Microsoft SharePoint 2010

En la figura 2.14 se muestra una representación habitual utilizada para explicar las características de SharePoint.



Figura 2.14 Esquema SharePoint 2010. Obtenido de [Gm10]

- **Sitios:** Esta funcionalidad del portal es utilizada para la de creación de espacios, en los cuales almacenar documentos de manera fácil y flexible, permitiendo recuperar el contenido de manera sencilla e incluso utilizando la conectividad de las aplicaciones Microsoft Office. En esta nueva versión se ha mejorado el acceso a los portales desde dispositivos móviles, permitiendo consultar e incluso editar documentos desde estos.
- **Comunidades:** Esta funcionalidad permite que los usuarios trabajen conjuntamente, facilita la búsqueda, posibilita la interacción con personas además de la clasificación y el etiquetado del contenido. Cada usuario tiene una página de perfil, que contiene información de los empleados o usuarios, datos acerca de su biografía, puesto, ubicación, intereses, habilidades y proyectos anteriores en los que han trabajado. Además de toda esta información también podemos tener un acceso rápido a distintos destinos que podemos predefinir nosotros mismos.

- **Contenido:** Permite a todos los usuarios participar en la gestión de contenidos, ya sea una página web, un documento o un conjunto de estos, siempre teniendo en cuenta los permisos y normativas. También se ha añadido la capacidad para poder validar el contenido de las listas, definido por el usuario. Además es posible administrar contenidos multimedia como vídeo, audio e imágenes, facilitando el almacenamiento, la administración y la localización.
- **Búsqueda:** La funcionalidad de buscar contenido dentro y fuera de SharePoint por parte de los usuarios, incluida la información en sistemas de bases de datos estructurados. Destacar especialmente, FAST Search Server 2010, esta opción proporciona mejores resultados en las búsquedas, adaptados de una forma más dinámica al trabajo y a los roles y funcionalidades de la organización.
- **Conocimientos:** Grandes funcionalidades, como la de usar Microsoft Excel para obtener acceso a los datos de una página web y mostrarlos, así como aprovechar los indicadores clave de rendimiento para transformar datos que proporcionen información a la organización.
- **Composiciones:** La funcionalidad de composiciones permite a los usuarios crear sus propias soluciones con o sin código de forma local o en nube, gracias a un conjunto de herramientas como son SharePoint Designer y Visual Studio 2010.
- Además de las funcionalidades citadas, me gustaría destacar otras características que contiene esta nueva versión de SharePoint 2010 que otorgan un gran valor al usuario.
 - Recibir actualizaciones de archivos en listas y bibliotecas mediante RSS.
 - Wiki.
 - Ribbon, la nueva barra superior que permite un acceso rápido a las acciones más habituales de los usuarios.
 - Multilinguaje.
 - Nuevo sitio web de administración central del mismo. Este recurso permite la administración y la supervisión de la granja de SharePoint.
 - Auditoria de documentos, es posible realizar el seguimiento de cualquier documento, desde su captura hasta que es archivado.

2.3.2.4. Formas básicas de extensibilidad

Disponemos de distintas formas para extender el comportamiento de la plataforma de colaboración SharePoint 2010, dependiendo de la utilidad o función que se desee desarrollar se tendrá que escoger entre una de las siguientes opciones:

- Creación de un elemento web: Es la más sencilla, los elementos web son unidades de información que constan de una barra de título, un marco y contenido. Cada uno de estos elementos Web incluyen contenidos o datos a una página, es posible que el usuario utilice estos datos para trabajar, o que los mismos sean modificados por el usuario.

Los programadores pueden crear una parte típicamente pequeña de una página web, como un elemento web, en Visual Studio 2010⁵ y agregar el código requerido. A continuación, los no programadores pueden agregar elementos web a páginas de contenido a medida que crean sitios.

Un ejemplo de un elemento web, podría ser un simple reloj creado en SilverLight⁶.

- Creación de una acción de flujo de trabajo: Los programadores pueden crear nuevas acciones de flujo de trabajo que contienen código personalizado en Visual Studio 2010 e implementarlas en SharePoint.
- Creación de un tipo de contenido externo BCS: Los programadores pueden crear un tipo de contenido externo que incluye la estructura de algunos datos externos, los métodos requeridos para obtener acceso a los datos y los requisitos de autenticación. Después de crear el tipo de contenido externo e implementarlo en SharePoint, los no programadores pueden usar esos datos mientras crean un sitio.

⁵ Página web de Visual Studio <http://msdn.microsoft.com/es-es/vstudio>

⁶ Página web de Silverlight <http://www.silverlight.net/>

3

Propuesta de integración

El objetivo de esta propuesta es la integración de la plataforma de gestión documental Abox con la plataforma de colaboración Microsoft SharePoint Server 2010, para ello vamos a presentar el marco de trabajo con el que se va actuar y para completar la información se expondrá el prototipo realizado.

La estructura del capítulo es la siguiente. En la sección 3.1 se presentan los roles que desempeñaran los miembros del equipo. En la sección 3.2 se mostrara la planificación del proyecto. En la sección 3.3 se expone el plan de entregas generado para la propuesta. Finalmente en la sección 3.4 se muestra el prototipo generado, explicando la razón de este, los objetivos y las conclusiones obtenidas del mismo.

3.1. Equipo

En nuestro caso particular contamos con un grupo pequeño, esto nos obliga a que algunos de los roles son gestionados por la misma persona, por ejemplo el rol de *manager* y el de *tracker* es gestionado por la misma persona. Pero hay que tener en cuenta que hay ciertos roles que no son compatibles que sean gestionados por la misma persona, un programador no debe ser a la vez el que gestione el rol de *tracker*, ni tampoco el rol de *tester*.

En la tabla 3.1 vamos a mostrar la organización del equipo, según sus roles

Rol	Empleados	
Programador	Humberto Delgado	Diego Hernández
Cliente	Rubén Lafuente	
Gestor (manager)	Luis Blasco	
Encargado de pruebas (tester)	Carlos Millán	
Encargado de seguimiento (tracker)	Víctor Sáez	
Entrenador (Coach)	Víctor Sáez	
Consultor	Equipo técnico del ITI	

Número total de personas en el equipo : 7
--

Tabla 3.1. Roles en Adapting S.L.

3.1.1. Reunión con el Instituto Tecnológico de Informática

Durante el inicio del proyecto se tuvo una reunión en las oficinas de Adapting S.L. con el Instituto Tecnológico de Informática, el cual presto un servicio de asesoría y consultoría con el objetivo de tratar la problemática de la integración de Abox con SharePoint.

En esta reunión participaron 2 miembros del Instituto Tecnológico de Informática y 3 miembros del personal de Adapting S.L., entre los cuales se encontraba Rubén Lafuente, el cual es el mi coordinador del proyecto final de carrera en la empresa y yo mismo. En mi opinión esta reunión fue muy positiva para el desarrollo del proyecto ya que se establecieron las bases del mismo. De esta reunión se sacaron las siguientes conclusiones:

- Adapting S.L. ofrece un producto que no pretende competir con SharePoint si complementarse.
- Varios escenarios de integración.
- Esfuerzo que costaría cada una de las posibilidades de integración.

3.2. Planificación

En la figura 3.1 se puede observar una historia de usuario que ha sido redactada para la realización de este proyecto. Esta historia de usuario denominada “Consulta de carpetas y documentos” corresponde a una de las funcionalidades básicas que el cliente ha solicitado, de ella se desprende que el cliente le ha otorgado una gran prioridad, por lo cual tiene un gran valor para el negocio y va a ser realizada en la primera iteración. Los programadores han establecido un riesgo de desarrollo medio.

En la figura 3.2 se puede apreciar una tarea de ingeniería. Esta está ligada con la anterior historia de usuario, la cual es asignada a un programador, el cual será responsable de ella. Existen distintos tipos de tareas de ingeniería, desarrollo, mejora, corrección y otras.

Por ultimo en la figura 3.3 vamos a ver un ejemplo de prueba de aceptación para la historia de usuario “Consulta de carpetas y documentos”.

<i>Historia de usuario</i>	
Número: 2	Usuario: Operador
Nombre historia: Consulta de carpetas y documentos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Humberto Delgado	
Descripción: Se desean consultar las carpetas y documentos existentes en la plataforma SharePoint 2010, de manera que queden organizados de la misma forma en la que se representan en su origen.	
Observaciones: Se consultan las carpetas y los archivos independientemente de los permisos.	

Figura 3.1 Historia de usuario

Todas las historias de usuario comparte el mismo tipo de usuario. Esto es debido a que el futuro usuario de la aplicación compartirá las mismas características, por ello hemos decidió reconocer a este usuario con el nombre de operador.

<i>Tarea de ingeniería</i>	
Número tarea: 2	Historia de usuario : 2
Nombre tarea: Obtener el listado de librerías de un sitio SharePoint 2010	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14 de febrero del 2011	Fecha fin: 25 de febrero del 2011
Fecha inicio: 15 de febrero del 2011	
Programador responsable: Humberto Delgado	
Descripción: Un sitio web, está compuesto de librerías, que estas a su vez son las que contiene los documentos y las carpetas.	

Figura 3.2 Tarea de ingeniería

Caso de Prueba de Aceptación	
Código: 2	Historia de usuario : 2
Nombre: Listado de librerías de un sitio web de SharePoint	
Descripción: Se desea consultar el listado de bibliotecas existentes del sitio web situado en el puerto 8080.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1) Iniciar sesión en el sistema 2) Acceder al módulo SharePoint 3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web sobre el que se desea consultar las librerías. Una vez listo pinchar sobre el botón guardar. 4) Ir a la opción explorador, en la cual se mostraran todo el listado de bibliotecas del sitio web. 	
Resultados esperados: El resultado tiene que mostrar un listado con 6 bibliotecas.	
Evaluación de la prueba: Satisfactorio	

Figura 3.3 Prueba de aceptación

Siguiendo esta filosofía y a modo resumen, en la tabla 3.2 se relacionan todas las historias de usuario obtenidas, con los tareas de ingeniería y casos de prueba.

Historia de usuario	Tareas de ingeniería	Casos de prueba
1) Consultar los sitios web disponibles en SharePoint	Sitios webs de SharePoint 2010	Sitios web
2) Consulta de carpetas y documentos	Obtener el listado de librerías de un sitio SharePoint 2010	Listado de librerías de un sitio web de SharePoint
	Consultar el listado de carpetas y documentos de una librería	Listado de carpetas de la raíz de la biblioteca "Documentos compartidos" I
	Obtener la extensión de los archivos.	Listado de documentos de la raíz de la biblioteca "Documentos compartidos" II
	Interfaz usable	
	Obtener datos básicos	

Historia de usuario	Tareas de ingeniería	Casos de prueba
3) Utilizar los usuarios existentes en SharePoint 2010	Obtener usuarios existentes en SharePoint	Iniciar sesión con un usuario específico
	Configuración de los datos del usuario	Comprobar que se guardan los datos de inicio de sesión.
	Comprobación de los datos de configuración del usuario	
4) Respetar los permisos de lectura	Comprobación de permisos de lectura	Lectura de la biblioteca "Documentos privados"
5) Respetar los permisos de escritura	Comprobación de permisos de escritura	Modificar archivos en una biblioteca
6) Descarga de documentos	Descarga de documentos	Descarga de documentos
7) Realizar búsquedas simples	Obtener acceso al <i>WebService</i> de búsqueda de SharePoint	Búsqueda simple
	Creación de la consulta de búsquedas	
	Tratamiento de la respuesta del <i>WebService</i> de búsqueda	
8) Realizar búsquedas avanzadas	Búsqueda avanzada	Búsqueda avanzada
9) Versiones de los documentos	Listado de versiones de un documento	Versiones de documentos
10) Consultar los metadatos y otras informaciones de un documento	Obtener el tipo documental del archivo	Consulta de los metadatos de un archivo
11) Documentos y carpetas entre distintas plataformas	Utilización del HotFolder para el envío de ficheros de SharePoint a Abox	Enviar documento de la SharePoint a Abox
	Envío de ficheros de Abox a SharePoint	
12) Modificación de los metadatos y otras informaciones de un documento	Modificación de los metadatos del documento	Modificar el metadato situación de un archivo de imagen

Tabla 3.2 Relación entre las historias de usuario, tareas y casos de prueba

En el anexo al final de este documento, podremos encontrar todas las historias de usuario, tareas de ingeniería y pruebas de aceptación que han sido utilizadas para la realización de este proyecto.

3.3. Plan de entregas

Después de tener ya definidas las historias de usuario, es necesario crear el *release planning*, o también conocido como el plan de entrega, este es pactado entre el cliente y los desarrolladores. El plan de entrega contiene un listado con todas las historias de usuario y sus puntos más relevantes, como en que iteración van a ser entregados, prioridad, y puntos estimados.

Podemos apreciar en la tabla 3.2 que en el plan de entrega encontramos 5 iteraciones diferentes, en la primera iteración se tratará de asentar la arquitectura del proyecto y en las subsecuentes iteraciones se irán incluyendo las historias de usuario seleccionadas, dándole cada vez mayor funcionalidad al producto. Cada una de estas iteraciones tiene una duración de 4 semanas aproximadamente. En la tabla 3.3 describimos con mayor detalle cada una de estas iteraciones.

Historias de usuario	Puntos estimados	Prioridad en el negocio	Iteración
Consultar los sitios web disponibles en SharePoint	2	Alta	1
Consulta de carpetas y documentos	3	Alta	1
Utilizar los usuarios existentes en SharePoint 2010	1	Baja	1
Respetar los permisos de lectura	1	Alta	2
Respetar los permisos de escritura	2	Media	2
Descarga de documentos	3	Media	3
Realizar búsquedas simples	3	Alta	2
Realizar búsquedas avanzadas	4	Baja	3
Versiones de los documentos	1	Media	4
Consultar los metadatos y otras informaciones de un documento	4	Baja	4
Documentos y carpetas entre distintas plataformas	4	Baja	5
Modificación de los metadatos y otras informaciones de un documento	4	Baja	5

Tabla 3.3 Plan de entrega

Primera iteración

Historias de usuario	Puntos estimados	Prioridad
Consultar los sitios web disponibles en SharePoint	2	Alta
Consulta de carpetas y documentos	3	Alta
Utilizar los usuarios existentes en SharePoint 2010	1	Baja

Puntos totales 6

Segunda iteración

Historias de usuario	Puntos estimados	Prioridad
Respetar los permisos de lectura	1	Alta
Respetar los permisos de escritura	2	Media
Realizar búsquedas simples	3	Alta

Puntos totales 6

Tercera iteración

Historias de usuario	Puntos estimados	Prioridad
Descarga de documentos	3	Media
Realizar búsquedas avanzadas	4	Baja

Puntos totales 7

Cuarta iteración

Historias de usuario	Puntos estimados	Prioridad
Consultar los metadatos y otras informaciones de un documento	4	Baja
Versiones de los documentos	1	Media

Puntos totales 5

Quinta iteración

Historias de usuario	Puntos estimados	Prioridad
Documentos entre distintas plataformas	4	Baja
Modificación de los metadatos y otras informaciones de un documento	3	Baja

Puntos totales 7

Tabla 3.4 Iteraciones

En la metodología XP, se define la idea de velocidad para establecer la rapidez con la que se desarrollara el proyecto. De esta forma se sabrá cuantas historias de usuario se pueden desarrollar en las distintas iteraciones. Para nuestro proyecto se ha establecido una velocidad de 6 puntos.

Para calcular el número de iteraciones que se deberían realizar en el proyecto se ha planificado según el alcance del sistema, aplicando la siguiente fórmula: puntos de las historias de usuario/ velocidad del proyecto. De manera que se ha obtenido que el número de iteraciones deba ser 5.

En la tabla 3.4 se aprecia un resumen detallando que historias de usuario se van a realizar en cada una de las iteraciones. Se puede apreciar como en la primera iteración se va a desarrollar una historia de usuario con prioridad baja, esto es debido a que esta historia de usuario es necesaria para asentar las bases del módulo.

3.4. Prototipo

Podemos definir un prototipo como un modelo o una demostración a escala que no contiene toda la funcionalidad del producto final, pero que permite proporcionar una retroalimentación temprana al usuario y al equipo de programadores.

Mientras el cliente escribía las historias de usuario el equipo de programadores inicio la creación de un prototipo con la meta principal de determinar la viabilidad, los posibles problemas con los que nos podíamos encontrar y los efectos potenciales de la toma de decisiones a lo largo del proyecto.

3.4.1. Objetivos

Después de unas tareas iniciales de investigación se establecieron ciertos puntos importantes, algunos de los cuales proporcionarían una gran retroalimentación y marcarían una ruta de trabajo. A continuación se muestran una lista de los mismos.

- Comprobar las posibilidades que nos ofrecen los *Web Services* proporcionados por SharePoint Server 2010.
- Utilización de la *API Client Object Model*.
- Uso de los usuarios de SharePoint.
- Integración de usuarios Abox y SharePoint Server 2010.
- Adquirir experiencia en el desarrollo.

3.4.2. Implementación

El prototipo se desarrolló bajo el entorno de Visual Studio 2010, el cual tiene soporte para una gran amplitud de lenguajes como son C#, ASP .NET y Visual Basic. Este *IDE* proporciona un único entorno de desarrollo, el cual incluye multitud de herramientas que permiten el desarrollo rápido de aplicaciones y de servicios Web, depuración de las mismas, exploración del código de una forma sencilla y muchas otras acciones.

A la hora de seleccionar un lenguaje de programación, necesitábamos que soportara la orientación a objetos, con una sintaxis sencilla, conocida y que resultara muy familiar al resto del equipo de desarrollo. Debido a estas condiciones se decidió utilizar el lenguaje C# utilizando la *API* de *Windows Forms* para la creación de aplicaciones gráficas.

Una vez se empezó a desarrollar el prototipo, el primer objetivo fue conseguir iniciar sesión con un usuario y obtener un listado de información de SharePoint, para ello se utilizaron los *Web Services* que proporciona SharePoint para consultar información y realizar tareas, pero tras unos días pruebas y de ensayos se decidió cambiar por el uso de la *API Client Object Model* la cual resultó mucho más eficaz y cómoda a la hora de programar y de obtener un listado de todas las librerías de un sitio web. Un ejemplo del listado de las librerías de un sitio web visto desde el prototipo se puede ver en la figura 3.4.

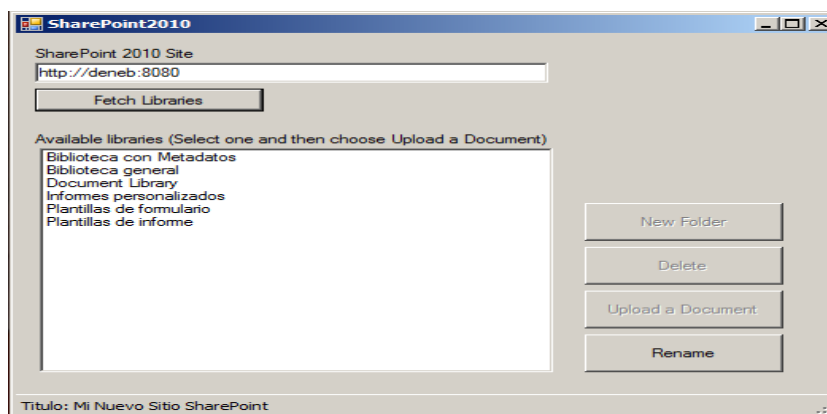


Figura 3.4 Listado de librerías de SharePoint vistas desde el prototipo

Los siguientes pasos fueron la obtención de información proveniente de una librería, figura 3.5, así como las carpetas que están incluidas dentro de la misma, figura 3.6.

Después de conseguir una serie de objetivos de consulta, se decidió probar otro tipo de objetivos, como son la creación, modificación y eliminación de elementos contenidos en

la plataforma SharePoint, siempre teniendo en cuenta la limitación de los permisos de cada usuario

3.4.3. Grado de consecución de los objetivos

El desarrollo del prototipo se alargó durante 2 semanas, en ellas se obtuvo un gran y valiosa información para el desarrollo del producto, la cual sirvió a su vez para obtener experiencia a los programadores, reducir el coste de la programación, facilito la toma de decisiones y encontró algunos de los problemas técnicos con los que nos podíamos encontrar en la implementación del producto real.

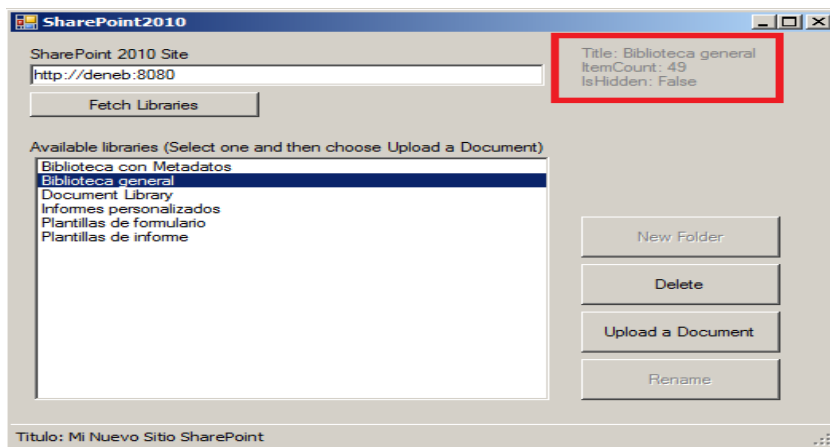


Figura 3.5 Información de una librería vista desde el prototipo

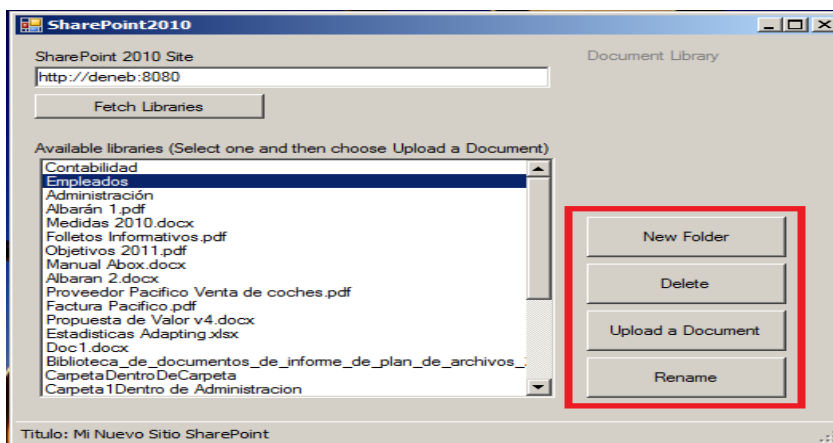


Figura 3.6 Acciones del prototipo

Al contrario que otros prototipos este no fue mostrado al cliente, debido a que el objetivo principal de este no era ese, si no comprobar que acciones se podían llevar a cabo y el alcance final del proyecto.

Como conclusión podemos afirmar que el desarrollo de este prototipo contribuyo al éxito de la implementación del producto final. Después de esto, se decidió desechar el prototipo y empezar de nuevo la implementación del producto final.

4

Desarrollo

En este capítulo se expone con detalle la arquitectura empleada y cada una de las tecnologías empleadas en el desarrollo de este nuevo módulo de Abox. Además de esto también se explican los tipos de pruebas realizadas y las buenas prácticas que hemos intentado aplicar a lo largo de todo el proceso.

La estructura del capítulo es la siguiente. En la sección 4.1 se expone la implementación en 3 capas que tiene el módulo y se razón su elección. En la sección 4.2 se explican los artefactos y técnicas utilizadas. En la sección 4.3 se explican los tipos de pruebas y se expone un ejemplo. Para terminar en la sección 4.4 se comentan las buenas prácticas que se han aplicado.

4.1. Arquitectura

En la ingeniería del software encontramos distintos tipos de arquitecturas, las más habituales que podemos encontrar en cualquier software son, monolítica, cliente-servidor y tres capas. Cada una de estas contiene a su vez ciertas ventajas y desventajas⁷.

Durante el desarrollo de esta propuesta hemos seguido una arquitectura de 3 capas. Esto permite separar la capa de presentación, la capa de negocio y los datos. En ocasiones esta separación puede ser física, como se muestra en la figura 4.1. Es decir podemos tener en distintos ordenadores cada una de las capas, con lo cual obtenemos una aplicación distribuida físicamente.

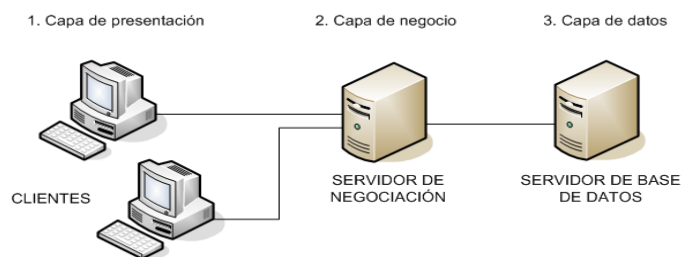


Figura 4.1 Arquitectura 3 capas. Obtenido de [Wp11]

⁷ Ventajas y desventajas de las arquitecturas
<http://www.mitecnologico.com/Main/ArquitecturaAplicacionesWeb>



Figura 4.2 Arquitectura 3 capas. Obtenido de [Pr11]

De esta manera nuestra aplicación queda dividida en 3 partes diferenciadas como se aprecia en la figura 4.2:

- La capa de presentación se refiere al mecanismo de interacción del usuario con el sistema. Su función es la de presentar la información al usuario y tomar los eventos que el usuario genere. Los tipos de interfaces software más comunes son las aplicaciones web y las aplicaciones de ventana.
- La lógica de negocio se refiere al conjunto de reglas que determina como funciona un sistema, según su naturaleza, bajo que parámetros y condiciones de acuerdo a las necesidades de los clientes o los usuarios. Los elementos fundamentales de esta capa son los objetos de dominio, los cuales son simples objetos que sólo contienen los datos que representan.
- La capa de acceso a datos se refiere a la media a través del cual podemos acceder y manipular los datos existentes en un sistema. En nuestro proyecto esta tarea será desarrollada en la mayor parte por NHibernate del cual hablaremos en el punto 4.2.2.

El uso de esta arquitectura a lo largo del desarrollo de esta propuesta nos ha aportado los siguientes beneficios:

- El desarrollo se realiza en distintos niveles permitiendo el desarrollo en paralelo.
- La aplicación es más robusta debido al encapsulamiento.
- Las tareas de mantenimiento y soporte son más sencillas.
- Mayor flexibilidad, se pueden añadir y eliminar funcionalidades de una forma más simple.

4.2. Tecnologías

A lo largo del proyecto hemos utilizado una larga lista de tecnologías y herramientas que han permitido la obtención de grandes y beneficioso resultados, es por ello que vamos a comentar las más importantes, explicando para que sirve, como se han empleado y que resultados nos han permitido obtener, pero antes de ello en la figura 4.3 se muestra un gráfico con la arquitectura de la aplicación.

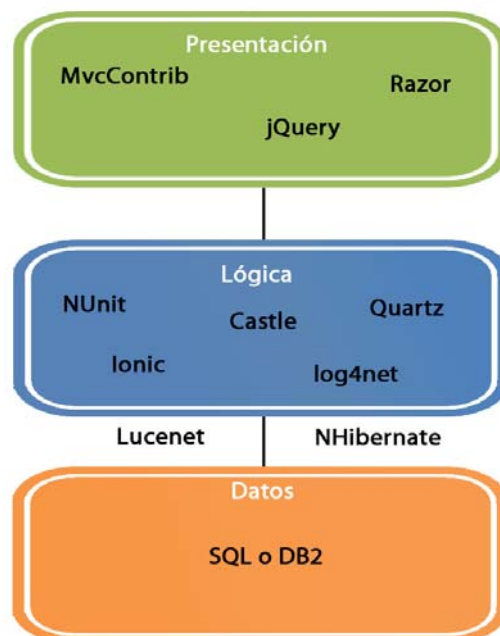


Figura 4.3 Arquitectura de la aplicación

4.2.1. Object Client Model

El *Client Object Model* [Wh10] proporciona un mecanismo de trabajo remoto con sitios de SharePoint 2010 y 2007, actuando como una capa de abstracción que nos permite trabajar con objetos SharePoint. Es decir el *Client Object Model* actúa como una API de manera que nos permite obtener, actualizar y gestionar datos en SharePoint.

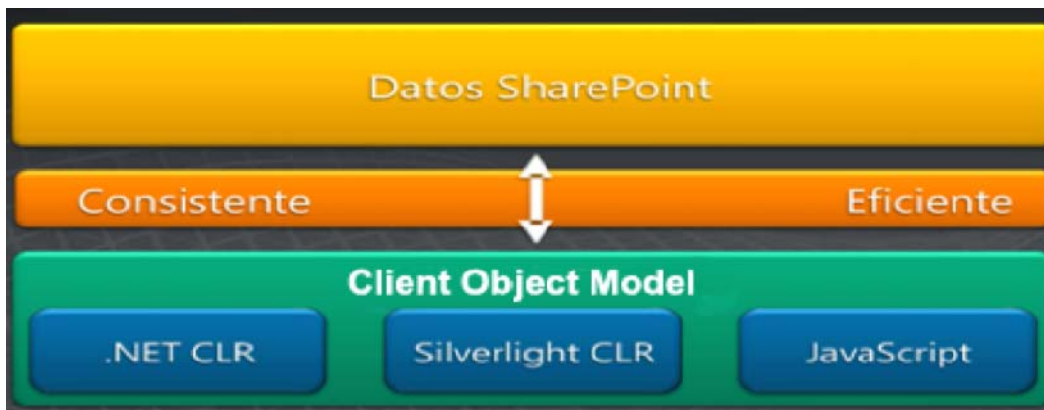


Figura 4.4 Resumen de cómo funciona el *Client Object Model*

Como se aprecia en la figura 4.4 podemos identificar 3 API's distintas, cada una de ellas para distintas tecnologías .NET, Silverlight y ECMAScript.

Una de las principales ventajas que nos proporciona es que no tenemos la necesidad de tener instalado SharePoint 2010 en nuestro entorno de desarrollo. Para ello Microsoft ha creado el *Client Object Model Redistributable*.

El *Client Object Model* es más sencillo de utilizar respecto a los Web Services, ya que el resultado de este último son valores XML, los cuales se tienen que analizar manualmente para obtener los resultados deseados de la operación. En cambio el *Cliente Object Model* trabaja con objetos SharePoint lo que facilita en gran parte la tarea al programador ya que le permite acceder a los métodos y a las propiedades de los objetos.

Si realizamos una comparativa entre estas dos formas de acceso a datos en SharePoint y desde donde podemos utilizarlas, observamos que el modelo objeto cliente es el que mayor soluciones de acceso nos proporciona en la mayoría de situaciones, tabla 4.1.

	<i>Web Services</i>	<i>Client Object Model</i>
<i>Web Parts</i>	NO	SÍ
<i>Event Handlers</i>	NO	SÍ
<i>Workflows</i>	NO	SÍ
<i>Desktop Apps</i>	SÍ	SÍ

Tabla 4.1. *Web Services* y *Client Object Model*. Obtenida de [Wh10]

A continuación vamos a listar los tipos de datos que podemos gestionar con el *Client Object Model*:

- Colecciones de sitios y sitios.
- Listas, elementos de las listas, vistas y listas de esquemas.
- Ficheros, carpetas y librerías.
- *Web Parts*.
- Seguridad
- Tipos de contenido.
- Plantillas del sitio y características de los sitios.
- Webs y propiedades de un sitio web.

Pero también existen otras partes de SharePoint no soporta el *Client Object Model*:

- Perfiles de usuario
- Usuario
- Búsqueda
- Publicación

- *Metadata* empresarial
- Administración
- *Web Services* REST Excel

También nos podemos encontrar con otras limitaciones como son la no existencia de capacidades para la elevación de privilegios, o que todas aquellas peticiones realizadas a la *API* son reguladas. Es por ello que actualmente existen ciertas acciones que aún no pueden ser realizadas y se tienen que realizar utilizando *Web Services*.

En la figura 4.5 podemos ver un ejemplo del uso de *Client Object Model* utilizando la tecnología .NET

```
1. using System;
2. using Microsoft.SharePoint.Client; //Cargamos las librerías de SharePoint
3.
4. class HolaMundo
5. {
6.     static void Main()
7.     {
8.         ClientContext clientContext = new ClientContext("http://adapting.com");
9.         Web site = clientContext.Web;
10.        clientContext.Load(site);
11.        clientContext.ExecuteQuery();
12.        Console.WriteLine("Titulo: {0}", site.Title);
13.    }
14. }
```

Figura 4.5 Hola mundo utilizando *Client Object Model*. Obtenido de [Us10]

```
1. using System;
2. using Microsoft.SharePoint.Client;
3.
4. class HolaMundo
5. {
6.     static void Main()
7.     {
8.         ClientContext ctx = new ClientContext("http://adpting.com");
9.         var list = ctx.Web.Lists.GetByTitle("NombreDeLaLista");
10.        //Buscamos una lista por su nombre
11.        ctx.Load(list);
12.        ctx.ExecuteQuery();
13.    }
14. }
```

Figura 4.6 Ejemplo de uso de *LoadQuery*. Obtenido de [Us10]

Podemos encontrar dos formas distintas de acceder a los datos en SharePoint, mediante la instrucción *LoadQuery* figura 4.7 y *Load* figuras 4.5 y 4.6. La primera de ellas es pasándole como argumento a la instrucción una consulta, por su parte *Load* le envía como argumento el objeto del cual queremos obtener información.

De esta forma se cargan una serie de datos básicos del objeto. Pero es posible seleccionar aquellos datos que se desea que sean cargados en el objeto, de esta forma se reduce el tamaño del JSON desde el servidor al cliente.

```
1.     using System;
2.     using Microsoft.SharePoint.Client;
3.
4.     class HolaMundo
5.     {
6.         static void Main()
7.         {
8.             ClientContext clientContext = new ClientContext("http://adpting.com");
9.             Web site = clientContext.Web;
10.            clientContext.Load(site,
11.                s => s.Title,
12.                s => s.Description); //Cagaremos el título y la descripción del sitio web.
13.            clientContext.ExecuteQuery();
14.        }
15.    }
```

Figura 4.7 Selección de datos con *Client Object Model*. Obtenido de [Us10]

4.2.2. NHibernate

La primera tecnología empleada en el proyecto fue el mapeador objeto-relacional denominado NHibernate⁸, la cual resuelve una de las problemáticas que surgen a la hora de trabajar con bases de datos es que estas siguen un modelo relacional y en cambio el código que nosotros desarrollamos suele estar en alguna tecnología que está orientada objetos. Es por ello que NHibernate nos permite convertir objetos en registros o viceversa sin necesidad de implementar complejas estructuras.

Para realizar este proceso de mapeo es necesario previamente configurar la conexión con la base de datos. Esta información de conexión estará alojada en un fichero de configuración el cual estará disponible para todos los elementos de la aplicación. En la figura 4.8 mostramos un ejemplo del mismo.

⁸ Página web de NHibernate <http://nhforge.org>


```
1. <?xml version="1.0" encoding="utf-8" ?>
2. <hibernate-configuration xmlns="urn:nhibernate-configuration-2.2">
3.   <session-factory>
4.     <!-- Cadena de conexión -->
5.     <property name="connection.connection_string">Data Source=.;Initial
      Catalog=AboxSHpfc;User=Usuario;Password=Contraseña;Min Pool Size=2</property>
6.     <!-- Nombre de la base de datos -->
7.     <property name="default_schema">Aboxpfc.dbo</property>
8.     <property
9.       name="connection.provider">NHibernate.Connection.DriverConnectionProvider</property>
10.    <property
11.      name="connection.driver_class">NHibernate.Driver.SqlClientDriver</property>
12.    <property name="connection.isolation">ReadCommitted</property>
13.    <property name="dialect">NHibernate.Dialect.MsSql2005Dialect</property>
14.    <property name="cache.use_query_cache">>true</property>
15.    <property name="cache.use_second_level_cache">>true</property>
16.    <property name="cache.provider_class">NHibernate.Caches.SysCache.SysCacheProvider,
      NHibernate.Caches.SysCache</property>
17.    <property name="adonet.batch_size">100</property>
18.    <property name="use_outer_join">>true</property>
19.    <property name="command_timeout">120</property>
20.    <property
21.      name="proxyfactory.factory_class">NHibernate.ByteCode.Castle.ProxyFactoryFactory,
      NHibernate.ByteCode.Castle</property>
22.    <property name="show_sql">>false</property>
23.    <property name="generate_statistics">>false</property>
24.  </session-factory>
25.</hibernate-configuration>
```

Figura 4.8 Fichero configuración de NHibernate

```
1. namespace Adapting.SharePoint.Domain.Mappings
2. {
3.   using FluentNHibernate.Mapping;
4.   public class SharePointConfigMap : ClassMap<SharePointConfig>
5.   {
6.     public SharePointConfigMap(){
7.       Table("SharePointConfig");
8.       Id(x => x.Id);
9.       References(x => x.UserAbox).Column("userId").Unique(); //Usuario abox
10.      Map(x => x.PasswordSharePoint); //Contraseña SharePoint
11.      Map(x => x.UserSharePoint); //Usuario SharePoint
12.      Map(x => x.DomainSharePoint); //Dominio SharePoint
13.      Map(x => x.AddressSharePoint); //Dirección
14.    }
15.  }
16. }
```

Figura 4.9 Archivo que mapea la tabla configuración

El proceso para la utilización de esta tecnología requiere la creación de un archivo dentro de nuestro proyecto, que mapeara la entidad de la base de datos. En la figura 4.10 podemos ver un ejemplo real en el cual un archivo mapeara la tabla de la base de datos.

```
1. create table Aboxpfc.dbo.SharePointConfig (  
2.     Id UNIQUEIDENTIFIER not null,  
3.     PasswordSharePoint NVARCHAR(MAX) null,  
4.     UserSharePoint NVARCHAR(MAX) null,  
5.     DomainSharePoint NVARCHAR(MAX) null,  
6.     AddressSharePoint NVARCHAR(MAX) null,  
7.     userId UNIQUEIDENTIFIER null unique,  
8.     primary key (Id)  
9. )
```

Figura 4.10 Tabla configuración dentro de la base de datos

NHibernate es una herramienta muy potente, la cual ha sido más protagonista de lo que inicialmente esperaba en este proyecto y no sólo lo es por la forma de trasladar objetos a entidades de base de datos y viceversa sino que además permite crear relaciones entre objetos de forma sencilla y es capaz incluso de crear automáticamente la base de datos tomando como referencia el mapeo de objetos.

4.2.3. Model view controller

Es un patrón de la arquitectura de las aplicaciones software, también conocido como MVC. En nuestro proyecto lo hemos utilizado principalmente para separar la lógica del negocio de la interfaz del usuario. De esta manera hemos incrementado la reutilización de código y la flexibilidad del mismo. Para ellos hemos utilizado el *Fraemwork* ASP.NET MVC⁹

Este patrón se divide en 3 módulos que son *model*, *view* y *controller* (modelo, vista y controlador en castellano) :

- Modelo: Se caracteriza por ser independiente de cualquier representación de salida o entrada.
- Vista: Muestra la información al usuario, es decir es la interfaz del usuario.

⁹ Podemos consultar distintos ejemplos en su sitio web <http://www.asp.net/mvc>

- Controlador: Actúa como intermediario entre la vista y el modelo. Sera el encargado de recibir los eventos de entrada.

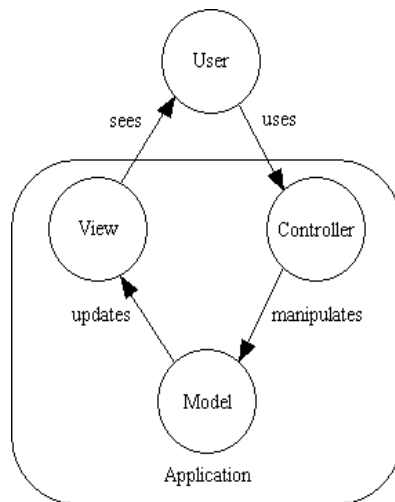


Figura 4.11 Model view controller. Obtenido de [Ma11]

En la figura 4.11 se muestra un resumen grafico del patrón. Un ejemplo de utilización se daría cuando un usuario crea una petición http desde su ordenador, esta llegara al servidor que aloja la página web, el cual se encargara de analizar la dirección URL y la despachara al controlador encargado. Aquí se realizara parte lógica de la aplicación y se generara un modelo el cual se rellenara con la información necesaria. Finalmente el controlador enviara el modelo hacia la vista la cual será la encargada de presentar la información al usuario.

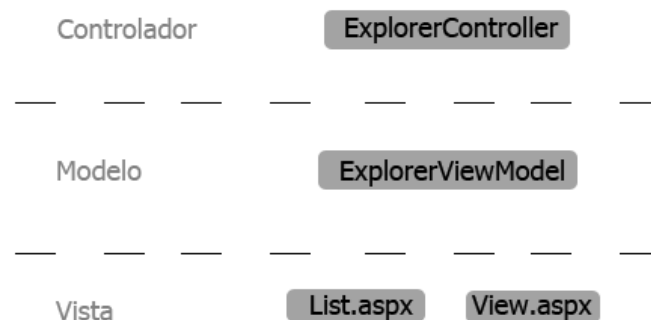


Figura 4.12 Ejemplo MVC

En la figura 4.12 se puede observar un ejemplo real que ha sido utilizado en este proyecto. En esta ocasión vemos como existe un único controlador denominado ExplorerController, el cual tiene un modelo denominado ExplorerViewModel. Este controlador está compuesto de dos acciones list y view, las cuales tienen asociadas dos vistas con su mismo nombre.

4.2.4. Inversion of control

Uno de los grandes problemas en el desarrollo de software es cómo manejar la dependencia de los objetos. La dependencia de los objetos se refiere que para crear un objeto, es necesario utilizar otro objeto y así sucesivamente. Un ejemplo de dependencia de los objetos se puede observar en la figura 4.13 .

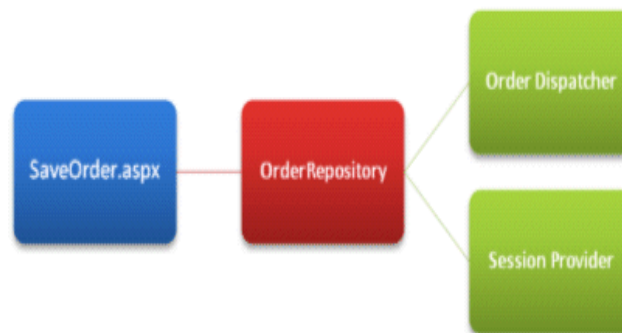


Figura 4.13 Dependencia de los objetos. Obtenido de [In06]

Los beneficios que inversión of control (inversión de control en español) [Fo04] nos proporciona, es la eliminación de las dependencias entre los objetos y una mayor flexibilidad en el código. La aplicación de este método de programación nos puede ser muy útil también para crear pruebas unitarias con otras herramientas como NUnit como se verá en el apartado 4.4.

Existen distintas maneras de aplicar inversión of control, en nuestro caso hemos decidido aplicarlo mediante Windsor Container¹⁰ .

¹⁰ Podemos obtener más información en <http://www.castleproject.org/container/>

4.3. Herramientas

Quizás la herramienta que más nos ha ayudado a la hora de la implementación ha sido Resharper, este software nos permite extender ampliamente las capacidades del *IDE* Visual Studio 2010, para realizar análisis de código, refactorización, navegación, búsqueda, entre otras tantas tareas, todo con el objetivo de hacernos más fácil y agradable la vida a los que desarrollamos software.

Por otra parte también encontramos otra herramienta que nos ha resultado de gran ayuda en el testeo de la aplicación, esta es NUnit. Esta aplicación es un *framework* para realizar pruebas, que no depende en lo absoluto de Visual Studio, el cual también aporta herramientas para realizar la misma función, pero en nuestro caso hemos preferido utilizar esta opción debido a que ya conocíamos anteriormente su funcionamiento y en mi caso ya la había utilizado previamente en la universidad. El motivo del uso de NUnit es la automatización de pruebas.

4.4. Pruebas

Existen diferentes tipos distintos de pruebas, en la siguiente enumeración destacamos las que quizás sean más importantes y las cuales garantizan la calidad del software:

- Unitarias
- Integración
- Funcionales
- Carga/Estrés
- Aceptación

Las ventajas del uso de pruebas además de conseguir el objetivo principal de garantizar la calidad de la aplicación, son la disminución de la necesidad de depuración, que pueden ser utilizadas como documentación, ayudan a mejorar el diseño además de posibilitar el trabajo en equipo.

En nuestro proyecto y en la metodología *extreme programming* las pruebas son una de las partes más importantes en el proceso de desarrollo de una aplicación. Las pruebas unitarias son ejecutadas constantemente y aseguran que el software funciona, en cambio las pruebas de aceptación son ejecutadas para asegurar el funcionamiento que el usuario espera que realice.

En la figura 4.14 vemos un ejemplo de prueba unitaria que ha sido utilizado en el proyecto.

```
1. public class SharePointServiceFixture : ComplexTestBase
2. {
3.     private IRepository<SharePointConfig> SharePointConfigRepository;
4.     private MockRepository Mocks;
5.     private ISharePointService SharepointService;
6.
7.     [SetUp]
8.     public void Setup()
9.     {
10.         Mocks = new MockRepository();
11.         SharePointConfigRepository = Mocks.StrictMock<IRepository<SharePointConfig>>();
12.         SharepointService = new SharePointService(SharePointConfigRepository);
13.     }
14.
15.     [Test]
16.     public void AddressSharePoint()
17.     {
18.         string url = SharepointService.GetUrlSharePoint().ToString();
19.         Assert.IsFalse(string.IsNullOrEmpty(url));
20.     }
21.
22. }
```

Figura 4.14 Ejemplo de test unitario

4.5. Buenas prácticas aplicadas

El desarrollo de software no es una tarea sencilla y en ella inciden un gran número de variables como son el producto a desarrollar, el cliente, el grupo de programadores, la metodología, los cambios de las especificaciones, entre muchas otras. Las metodologías y modelos existentes han contribuido sin duda a mejorar la calidad del software, estas se han formado sobretodo de las experiencias y proyectos exitosos, además de universidades y empresas que han tomado muy enserio la calidad del software.

En mi corta experiencia como programador he visto en el trabajado diario algunas prácticas que ofrece la metodología *extreme programming*, las cuales mejoran la experiencia y ayudan a conseguir el éxito de proyecto. En especial me gustaría destacar que

unos de los resultados más importantes que he visto al aplicar este tipo de prácticas en conjunto ha sido que ha mejorado la motivación del equipo y el ambiente de trabajo, lo cual ha sido muy beneficioso para el trabajo del día a día.

Desde el primer momento, se puso en marcha algunas prácticas como el cliente *in-situ* y el juego de la planificación, esta última es la cual promueve la forma de trabajar. Para ello se crearon las historias de usuario más tarde se siguió con los casos de pruebas y las tareas de ingeniería. Finalmente con toda la información redactada se desarrolló el plan de entrega.

Durante el periodo de implementación y el de mantenimiento del producto es necesario validar el correcto funcionamiento de toda la aplicación y comprobar que los nuevos cambios que se apliquen no provoquen fallos en las anteriores funciones implementadas, es por ello que requerimos de algún sistema que comprobara las pruebas unitarias de una forma automatizada.

El desarrollo del producto se realizó en 5 iteraciones, cada una de ellas era lo más pequeña posible de manera que le otorgaba suficiente valor al producto en la siguiente versión. En todo momento se intentó no hacer horas extras, que provocan en el desarrollador un desgaste.

Se intentó que los dos programadores fueran conscientes de todo el código, esto mejora la flexibilidad del equipo, la compartición de conocimientos además de las islas de conocimientos¹¹.

Por ultimo me gustaría destacar un listado de buenas prácticas que se han llevado a la práctica a la hora de la implementación, de forma que se ha conseguido un código más simple, más claro de entender y eficaz.

- Tabulaciones, permiten que la lectura sea mucho más simple y sencilla. En el equipo de desarrollo teníamos ciertas reglas que permitían que todos utilizáramos el mismo tipo de formato en el desarrollo del código.
- Uso correcto de los nombres de las variables, es decir que el nombre de la variable corresponda con el contenido y que además el nombre de la variable se pueda buscar fácilmente por el código.
- Uso de funciones e inclusión de comentarios sobre la función que realizan.

¹¹ Partes del código que se desconoce su utilidad.

- Comentarios a lo largo del código.
- Captura de los errores y correcto uso de las excepciones
- Uso de test unitarios sobre funciones.

5

Conclusiones

En este capítulo se presentan las conclusiones del proyecto final de carrera. En la sección 5.1 se relatan las contribuciones del desarrollo del proyecto. Finalmente en la sección 5.2 se exponen ciertas líneas de investigación en las cuales se puede seguir trabajando.

5.1. Conclusiones

El proyecto elaborado ha cumplido el principal objetivo marcado, conseguir hacer un vínculo desde Abox hacia la plataforma de colaboración SharePoint. De esta manera cualquier usuario pueda consultar, guardar y abrir la información que nos ofrecer SharePoint desde la interfaz de Abox. Esto ha sido posible gracias al gran esfuerzo y dedicación de todo el equipo.

Realizar una solución a un problema real, dentro de una empresa y con ayuda del ITI como consultor, ha cumplido todas las expectativas que tenía puestas en el proyecto, ya que me ha permitido estar involucrado en todas las fases del desarrollo de un proyecto de software, además de poder contribuir mejorando la calidad de Abox y dotando al mismo de nuevas funcionalidades y permitiéndole su expansión en un nuevo mercado. Es por ello valoro de forma muy positiva la experiencia personal alcanzada en la realización de este proyecto.

He podido poner en práctica alguna de los muchos conocimientos que he aprendido en la universidad, a la vez que he ampliado los mismos a través de la experiencia que mis compañeros me han transmitido y de las situaciones en las que nos hemos encontrado.

Asimismo durante todo el proceso hemos trabajado utilizando una metodología de desarrollo ágil, que con la ayuda de esta y de otros factores ha llevado al éxito del proyecto. Quizás esta no sea una metodología apropiada para todos los equipos de desarrollo, pero en nuestro caso ha demostrado ser conveniente y ha proporcionado libertad para que el grupo estableciera sus propias recomendaciones y prácticas.

La elaboración del proyecto, también me ha servido para conocer mejor mis capacidades y poder medir con más éxito el tiempo requerido en la planificación de las tareas. Así como, para ser más metódico y ordenado en el desarrollo. Pero además también me ha permitido ser consciente de que aún me falta mucho por aprender pero que tengo una buena base de conocimiento.

5.1.1. Detalle de consecución de los objetivos

La integración del nuevo módulo con el resto ha sido satisfactoria, en la figura 5.1 podremos apreciar como el modulo desarrollado para el vínculo entre SharePoint y Abox se relaciona con el módulo Document, permitiendo enviar información entre ellos. También ha sido posible relacionarlo con el módulo de Capture, permitiendo dar de alta documentos de forma masiva en la plataforma SharePoint.

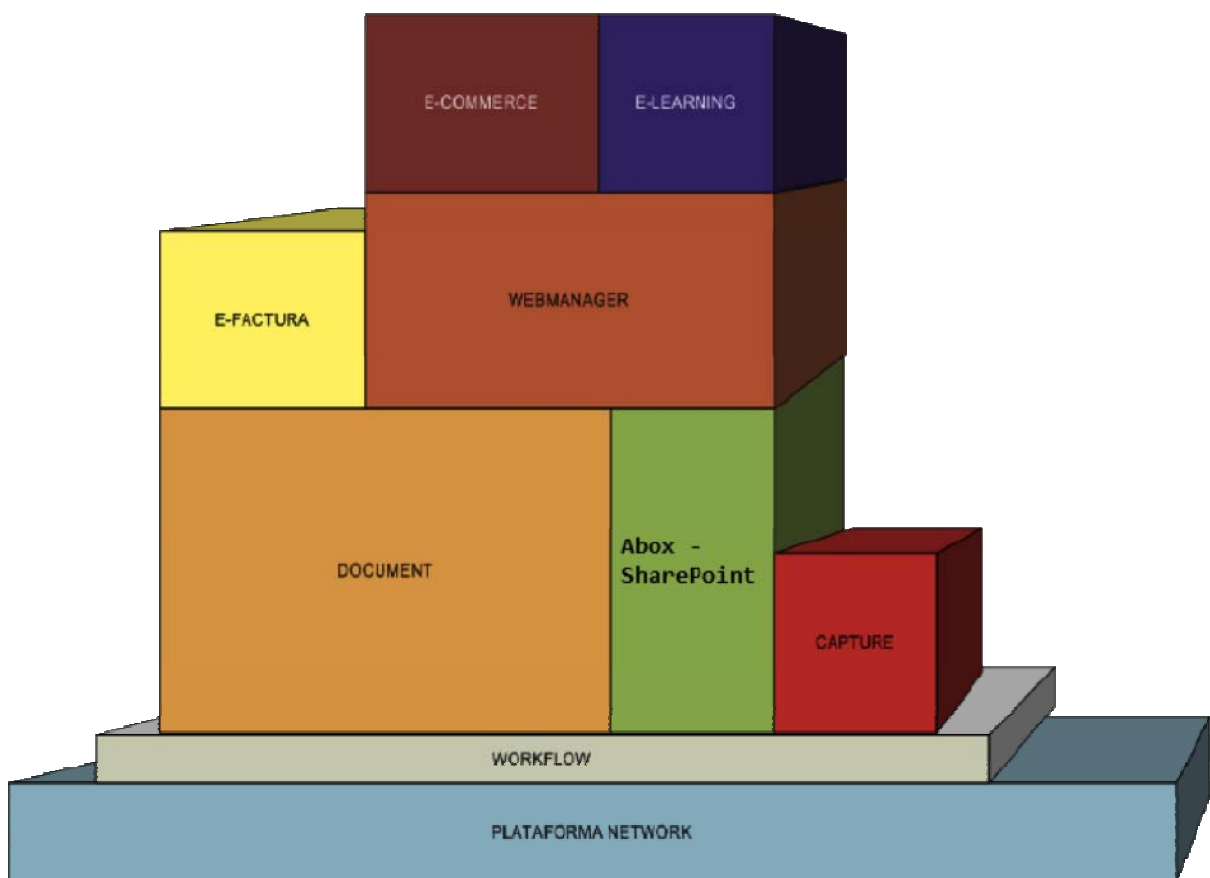


Figura 5.1 Módulos Abox

En la Tabla 5.1 vamos a mostrar un amplio resumen de algunas de las funcionalidades y características que hemos conseguido y de otras que no hemos llegado a lograr durante los meses que ha durado el desarrollo del proyecto.

Listado de funcionalidades específicas	Objetivo alcanzado	Línea de trabajo futura
Listar ficheros de acuerdo a la estructura de SharePoint	X	
Respetar los permisos de lectura, escritura, modificación y borrado	X	
Filtrado de las bibliotecas de interés para el usuario de un sitio web	X	
Testeo de la configuración del usuario	X	
Obtener información de la biblioteca		X
Obtener información de las carpetas		X
Obtener información de los archivos	X	
Descarga de documentos	X	
Información de las distintas versiones de un documento	X	
Descarga de las distintas versiones de un documento	X	
Consulta de los metadatos de un documento	X	
Añadir, modificar o eliminar metadatos a un documento		X
Obtención del tipo documental de un documento	X	
Creación y eliminación de carpetas	X	
Modificación de carpetas		X
Añadir y eliminar documentos	X	
Modificar documentos		X
Enviar documentos a la plataforma Abox desde SharePoint		X
Rápida y sencilla configuración del inicio de sesión del usuario de SharePoint	X	
Estructura en árbol de las bibliotecas y carpetas de un sitio SharePoint	X	
Búsqueda de un documento o carpeta	X	
Búsqueda de cualquier fuente de información de SharePoint	X	
Previsualización del contenido del archivo en las búsquedas	X	

Tabla 5.1 Funcionalidades y características

5.2. Líneas de trabajos futuros y posibles mejoras

Si bien es cierto que se han logrado alcanzar gran parte de los objetivos marcados al inicio del proyecto hay algunos aspectos del mismo que son mejorables pero además de esto, también hemos encontrado nuevas líneas de trabajo que podrían aumentar la potencia y la utilidad del mismo.

A continuación mostraremos los posibles trabajos futuros relacionados con la mejora de funcionalidades de la aplicación:

- Un sistema de envío de ficheros de SharePoint a Abox y viceversa. De esta manera se podrán enviar documentos entre las distintas plataformas.
- Integración entre los *workflows* de Abox y SharePoint.
- Obtención de información de las bibliotecas, las carpetas y de los tipos de contenido
- Subir documentos, definiendo el tipo de contenidos y metadatos que deseé el usuario.
- Mejorar la gestión de acciones sobre los documentos y carpetas. Entre ellas me gustaría destacar la adición y modificación de metadatos a los archivos, modificar el nombre de las carpetas y modificar los datos de los documentos.
- Añadir al árbol de documentos de Abox, el listado de librerías de SharePoint.
- Mejorar el ámbito de búsqueda de documentos en SharePoint de manera que se añadan más campos de filtrado.
- Búsqueda en ambas plataformas, tanto en SharePoint como en Abox de manera simultánea.
- Ampliar el registro de *logs* por parte del nuevo módulo.
- Refactorización del código y aplicación de patrones

Después de este listado, vamos a mostrar otro tipo de mejoras, aquellas que tienen relación sobre la tecnología utilizada en el proyecto:

- Toda la aplicación ha sido realizada mediante la tecnología MVC2, sería muy interesante aplicar las nuevas técnicas que se han desarrollado en la siguiente versión de esta tecnología MVC3.
- Cambiar las actuales vistas de ASPX por el nuevo motor Razor.
- Establecer contacto con el ITI para futura colaboración.

Bibliografía

- [Ar10] Armas, M. Valores, principios, actividades y procesos en la extreme programming, 2010. http://www.liderdeproyecto.com/articulos/extreme_programming.html .
- [Be01] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, Robert C., Mellor, S., Schwaber, K., Sutherland, J. y Thomas, D. Manifest for Agile Software Development. 2001. <http://agilemanifesto.org/>
- [Be99] Beck, K. Extreme Programming Explained. Embrace Change. 1999.
- [Ca03] Canós, José H., Letelier, P. y Penadés, M^a Carmen. Metodologías Agiles en el de Desarrollo de Software. 2003. <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf> .
- [Fo04] Fowler, M. Inversion of Control Containers and the Dependency Injection pattern. 2004. <http://www.martinfowler.com/articles/injection.html>
- [Gm10] González Martín, C. SharePoint 2010: Otro resumen + novedades. 2009. <http://jcgonzalezmartin.wordpress.com/2009/11/10/sharepoint-2010-otro-resumen-de-novedades/>
- [In06] Microsoft. Inversion of control. 2006. <http://msdn.microsoft.com/en-us/library/aa973811.aspx> .

Bibliografía

- [Ke11] Wikipedia. Kent Beck. Consultado en octubre 2011, http://es.wikipedia.org/wiki/Kent_Beck .
- [Ma11] Marston, T. The Model-View-Controller (MVC) Design Pattern for PHP. 2011. <http://www.tonymarston.net/php-mysql/model-view-controller.html> .
- [Mi11] Wikipedia. Microsoft Commerce Server. Consultado en septiembre del 2011, http://en.wikipedia.org/wiki/Microsoft_Commerce_Server .
- [Pr11] Wikipedia. Programación por capas, consultado en septiembre 2011. http://es.wikipedia.org/wiki/Programación_por_capas, 1999.
- [Rc08] Robert Cecil, M. Clean Code. 2008.
- [Ri06] Richardson, S. SharePoint History. 2006. <http://www.joiningdots.net/blog/2006/08/sharepoint-history.html> .
- [Si11] Wikipedia. Site Server. Consultado en octubre 2011. http://en.wikipedia.org/wiki/Microsoft_Site_Server .
- [Us10] Microsoft. Using the SharePoint Foundation 2010 Managed Client Object Model, 2010, <http://msdn.microsoft.com/en-us/library/ee857094.aspx>
- [Wh10] Microsoft. What's New: Client Object Model. 2010. <http://msdn.microsoft.com/en-us/library/ee535231.aspx>

A.1

Manual del usuario

Índice del manual de usuario

1. Introducción	77
1.1. Información general	78
2. Tutoriales	81
2.1. Primeros pasos	82
2.2. Configuración.....	83
2.3. El explorador	83
2.4. La ficha del documento.....	85
2.5. Creación de carpetas.....	87
2.6. Borrado de carpetas	88
2.7. Añadir documentos	88
2.8. Borrado de documentos	90
2.9. Búsqueda de documentos	90

Índice de figuras

<i>Figura 1.1.</i> Vínculo entre SharePoint y Abox	80
<i>Figura 1.2.</i> Vista de una biblioteca de SharePoint desde Abox	81
<i>Figura 1.3.</i> Vista de una biblioteca de SharePoint desde el mismo	81
<i>Figura 2.1.</i> Vista del <i>login</i> de Abox.....	84
<i>Figura 2.2.</i> Menú superior de la pantalla principal	84
<i>Figura 2.3.</i> Vista de la configuración del módulo de SharePoint.....	85
<i>Figura 2.4.</i> Listado de bibliotecas de SharePoint.....	86
<i>Figura 2.5.</i> Contenido de una biblioteca de SharePoint.....	86
<i>Figura 2.6.</i> Exploración en árbol de las bibliotecas de SharePoint.....	87
<i>Figura 2.7.</i> Ficha de un documento.....	88
<i>Figura 2.8.</i> Metadatos de un documento	88
<i>Figura 2.9.</i> Versiones de un documento.....	89
<i>Figura 2.10.</i> Vista de una nueva carpeta.....	90
<i>Figura 2.11.</i> Selección de una carpeta.....	91
<i>Figura 2.12.</i> Nuevo documento.....	91
<i>Figura 2.13.</i> Búsqueda de un documento	92
<i>Figura 2.14.</i> Resultados de búsqueda de un documento.....	93

1

Introducción

1.1. Información general

El objetivo de este manual es proveer información a los usuarios del módulo SharePoint para Abox. A fin de garantizar un correcto uso de la aplicación en este documento se describen acciones habituales que el usuario podrá realizar con la plataforma.

Podemos apreciar de forma gráfica en la figura 1.1 el principal beneficio que esta nueva funcionalidad con la cual los usuarios de Abox podrán utilizar y gestionar los contenidos albergados en la plataforma SharePoint, como si estuvieran trabajando desde la misma, pero beneficiándose de las ventajas que ofrece Abox. De esta forma las dos plataformas pueden trabajar entre sí, mediante el vínculo creado desde Abox

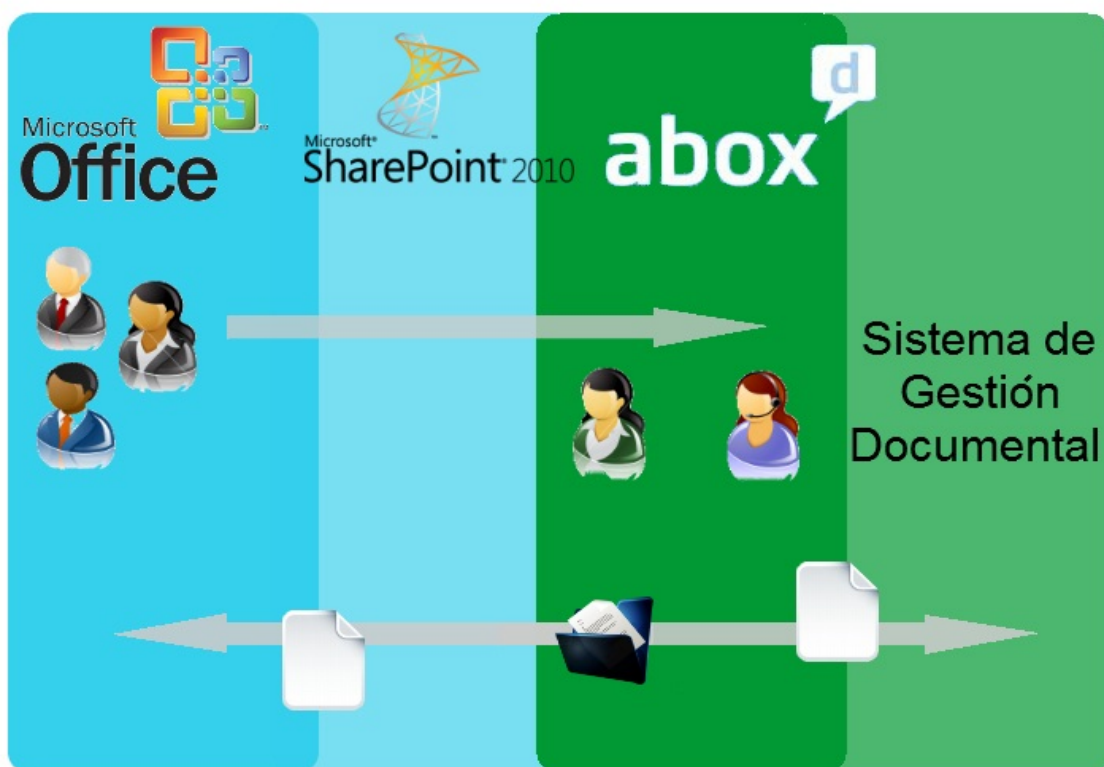


Figura 1.1. Vínculo entre SharePoint y Abox.

En las figura 1.2 y en la figura 1.3 se puede apreciar una comparativa del contenido de una misma biblioteca vista desde las dos plataformas, apreciando como el contenido listado es el mismo, esto es debido a que el módulo de SharePoint para Abox es una solución idónea para consultar y gestionar el contenido de la plataforma SharePoint.

Integración de la plataforma de gestión documental Abox Document con la plataforma de colaboración empresarial Microsoft SharePoint Server 2010

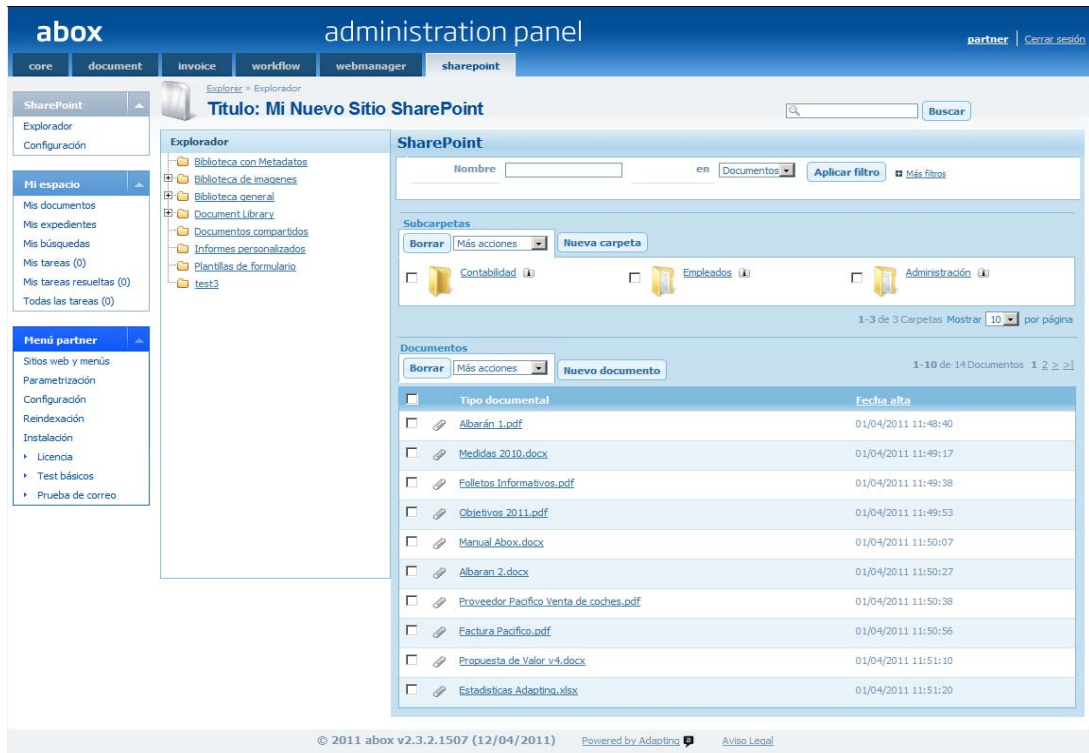


Figura 1.2. Vista de una biblioteca de SharePoint desde Abox.

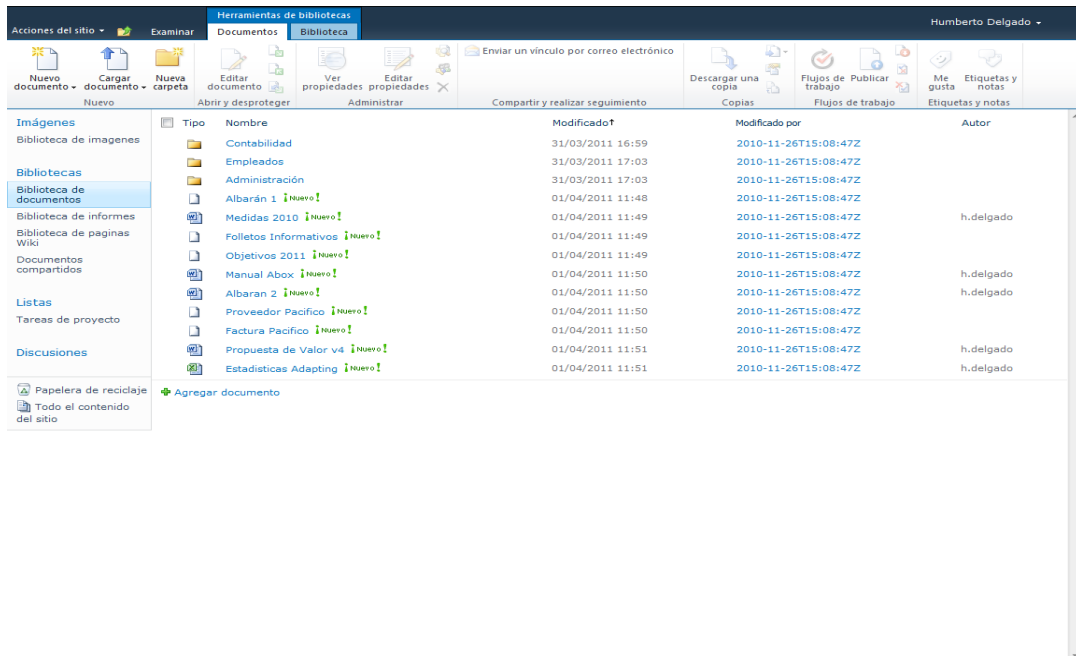


Figura 1.3. Vista de una biblioteca de SharePoint desde el mismo.

2

Tutoriales

2.1. Primeros pasos

Empezaremos iniciando sesión en la aplicación como se muestra en la figura 2.1 con nuestro usuario y contraseña de Abox.

Acto seguido apareceremos en la pantalla principal de la aplicación. En ella encontramos un menú superior en el cual se encuentran los módulos que tiene Abox, un ejemplo de este menú es la figura 2.2 en la cual podemos apreciar como existe una opción llamada SharePoint, ahí es donde estará incluido nuestro nuevo módulo de Abox.

En el caso de que esta pestaña no aparezca, habrá que consultar si el usuario con el que estamos accediendo a Abox tiene los permisos adecuados para utilizar este módulo.

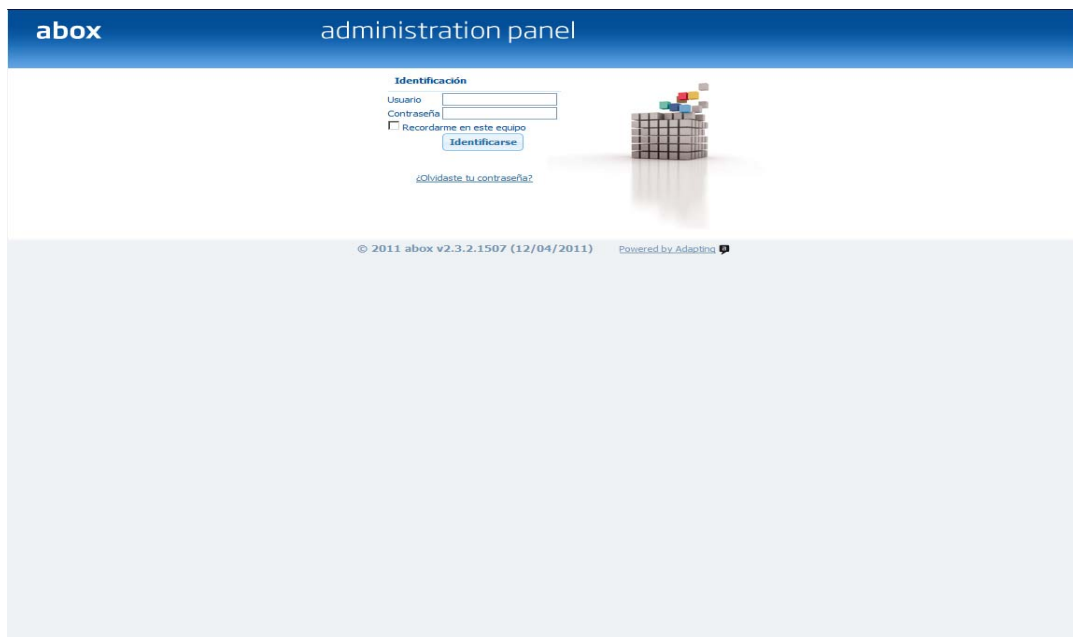


Figura 2.1. Vista del *login* de Abox.



Figura 2.2. Menú superior de la pantalla principal.

2.2. Configuración

Si es la primera vez que entramos al módulo con nuestro usuario, una vez pinchamos en la pestaña SharePoint nos enviara a la vista de configuración. Aunque siempre podremos acceder a ella a través del menú, en la opción configuración.

En esta vista encontraremos varios campos que tendremos que rellenar con la información de nuestro usuario de SharePoint. Antes de guardar la configuración se recomienda pinchar en el botón “Test conexión” para comprobar que los datos introducidos son correctos y el modulo funciona correctamente.

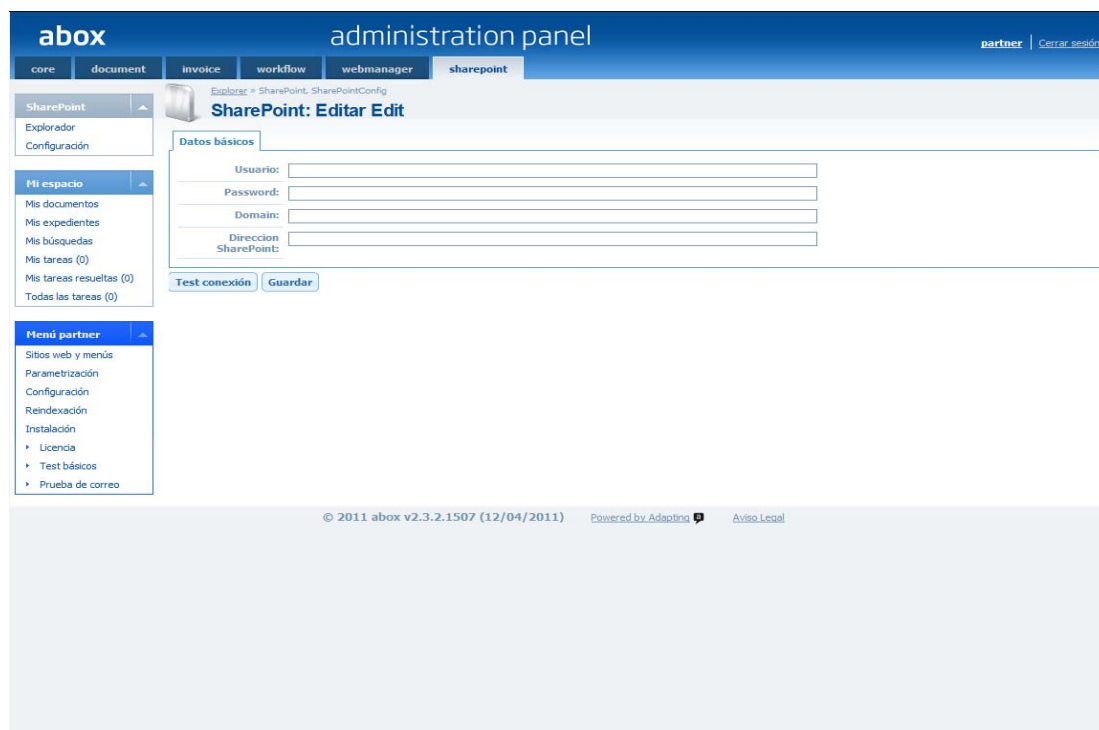


Figura 2.3. Vista de la configuración del módulo de SharePoint

2.3. El explorador

Podemos acceder al explorador de dos formas diferentes, la primera de ellas es pulsando en la pestaña superior denominada “SharePoint”, pero también podremos acceder a esta opción desde el menú de la izquierda, en una opción denominada “Explorador”.

En la figura 2.4 podemos apreciar que se listan el conjunto de bibliotecas que tiene el sitio web sobre el que hemos rellenado la configuración anteriormente.

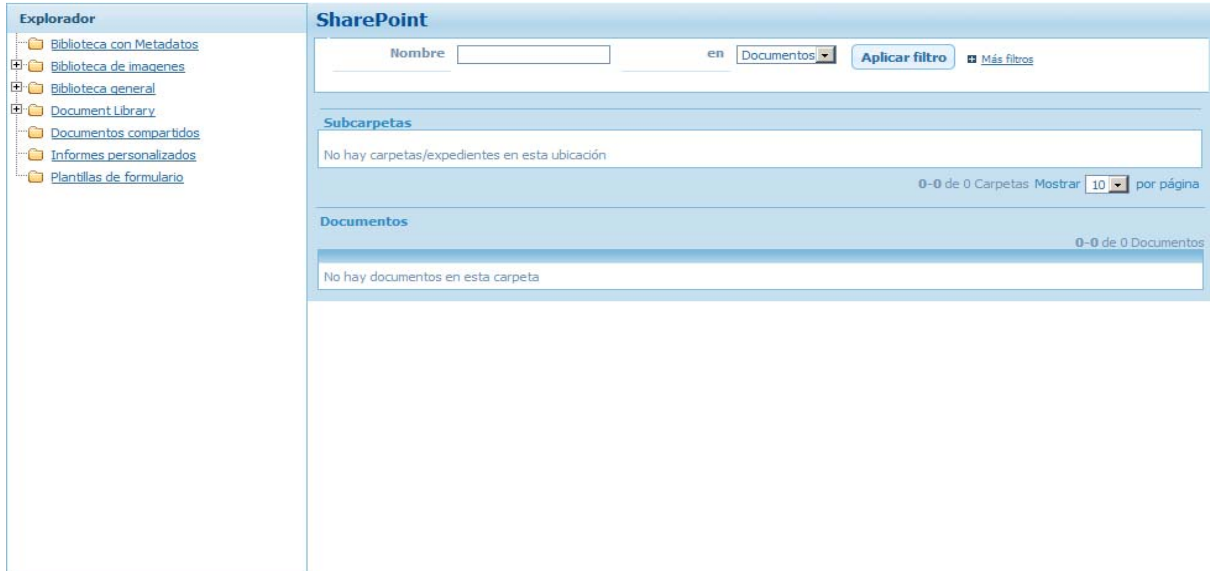


Figura 2.4. Listado de bibliotecas de SharePoint

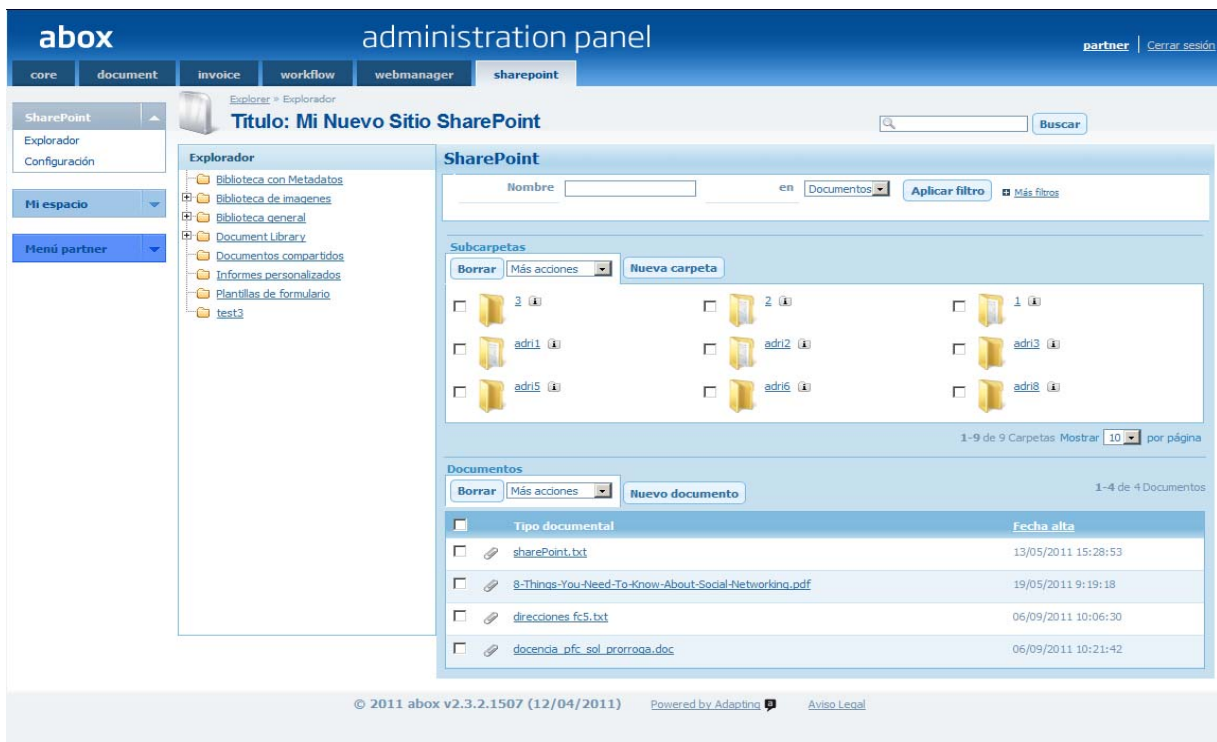


Figura 2.5. Contenido de una biblioteca de SharePoint

Si pinchamos sobre algunas de las bibliotecas veremos como en el espacio reservado a la derecha de la misma se muestra su contenido, un ejemplo de esto se puede ver en la figura 2.5.

Otra forma de navegar a través de las bibliotecas es como se muestra en la figura 2.6, pinchando en el icono de la izquierda de cada una de ellas, de esta manera se listarán en forma de árbol las carpetas que esta contiene.

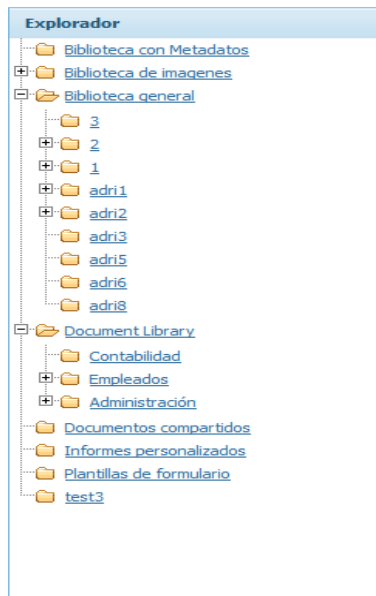


Figura 2.6. Exploración en árbol de las bibliotecas de SharePoint

2.4. La ficha del documento

En caso de que deseáramos ver información sobre un archivo, tan solo tendríamos que ponernos encima del nombre del archivo y pinchar sobre él, de manera que nos enviara a una pantalla con información sobre el archivo, como podemos apreciar en la figura 2.7.

La ficha del documento, se parte en dos, un encabezado y un cuadro donde se muestra información. En el encabezado podemos encontrar la versión actual del documento que se está consultando, un enlace que permite descargar el documento y un logo de la plataforma SharePoint cuando se esté consultando un archivo dentro de ella. La

otra parte de la ficha del documento se divide pestañas, las cuales hacen referencias a unos datos básicos, metadatos del documento y las versiones del mismo.

Finalmente en el caso que deseáramos volver a la biblioteca que estábamos consultando, tan solo tendríamos que pinchar sobre el enlace volver.

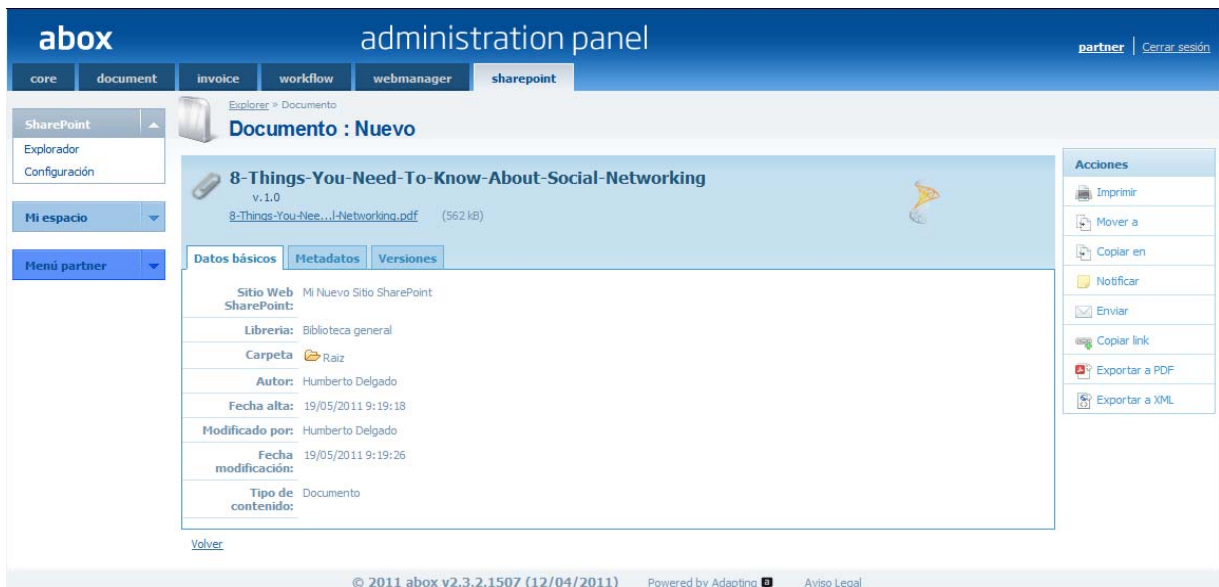


Figura 2.7. Ficha de un documento

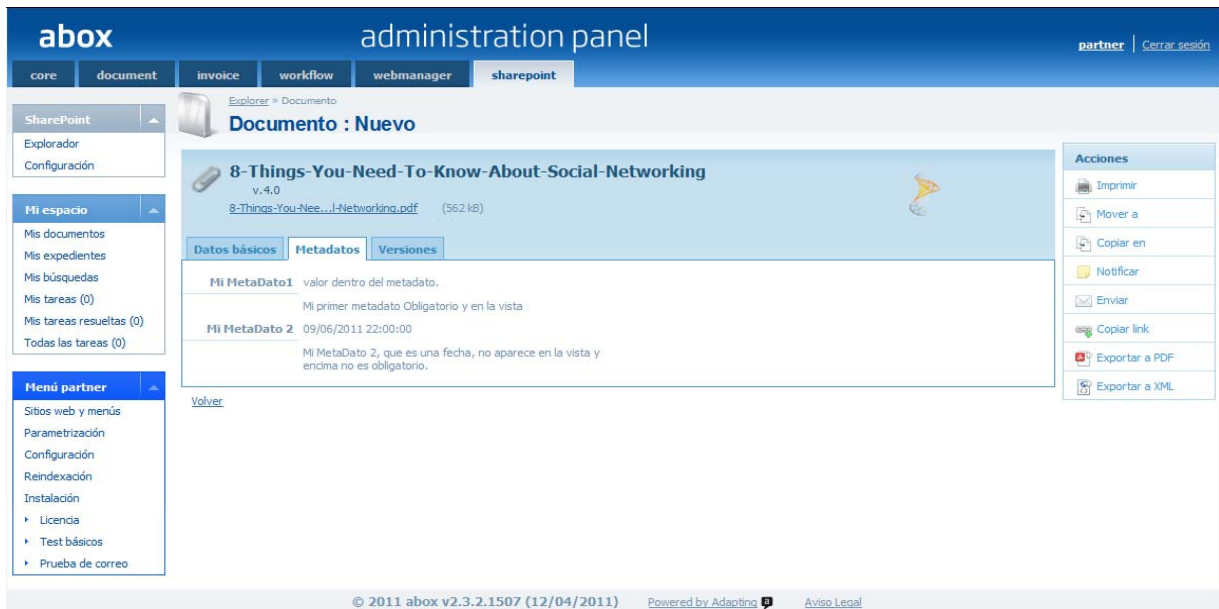


Figura 2.8. Metadatos de un documento

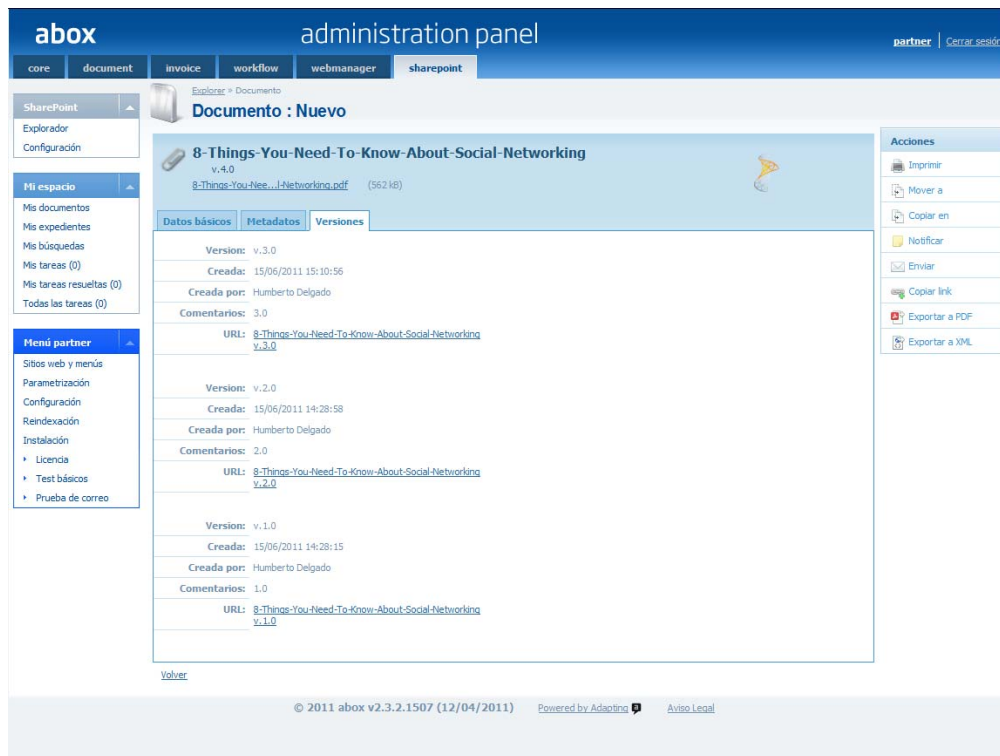


Figura 2.9. Versiones de un documento

2.5. Creación de carpetas

Una vez entremos en cada una de las bibliotecas podremos realizar distintas acciones, una de estas es la que nos permite crear nuevas carpetas.

Para ello tendremos que presionar el botón “Nueva carpeta” que está situado encima de listado de carpetas. Este botón nos enviara a una vista como la de la figura 2.10 en la que tendremos que rellenar algunos datos como el título y la descripción de la carpeta.

Una vez pulsemos en el botón guardar, la web nos redirigirá al lugar donde se ha creado la carpeta.

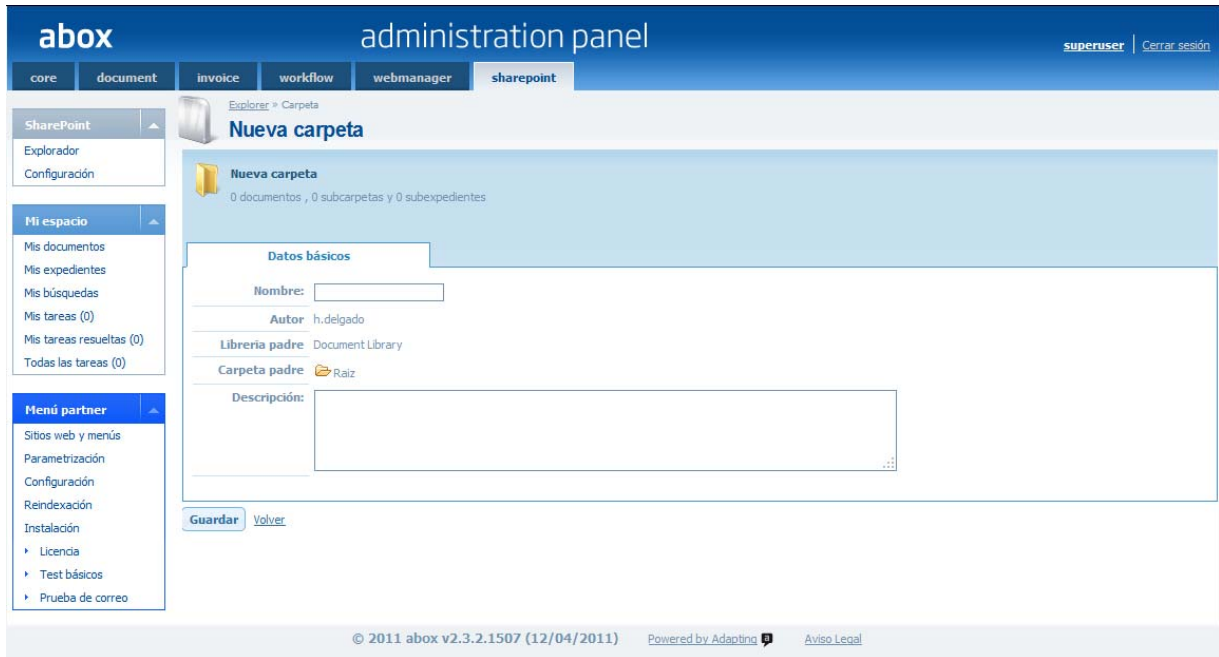


Figura 2.10. Vista de una nueva carpeta

2.6. Borrador de carpetas

Cada una de las carpetas tiene a su izquierda una casilla de verificación, la cual puede ser marcada para realizar acciones sobre estas. Si seleccionamos una o varias como se muestra en la figura 2.11 y pinchamos en el botón “borrar” estas serán eliminadas del sistema.

2.7. Añadir documentos

Quando nos encontremos consultando los documentos y carpetas de una librería, tendremos activado el botón “Nuevo documento” el cual nos enviara a una vista como la de la figura 2.12 en la cual tendremos que seleccionar el documento a subir a la plataforma SharePoint.

Una vez se haya completado la verificación del documento, podremos pinchar sobre el botón continuar el cual nos confirmara que el documento se ha subido correctamente y nos redirigirá a la vista donde se encuentra el documento.

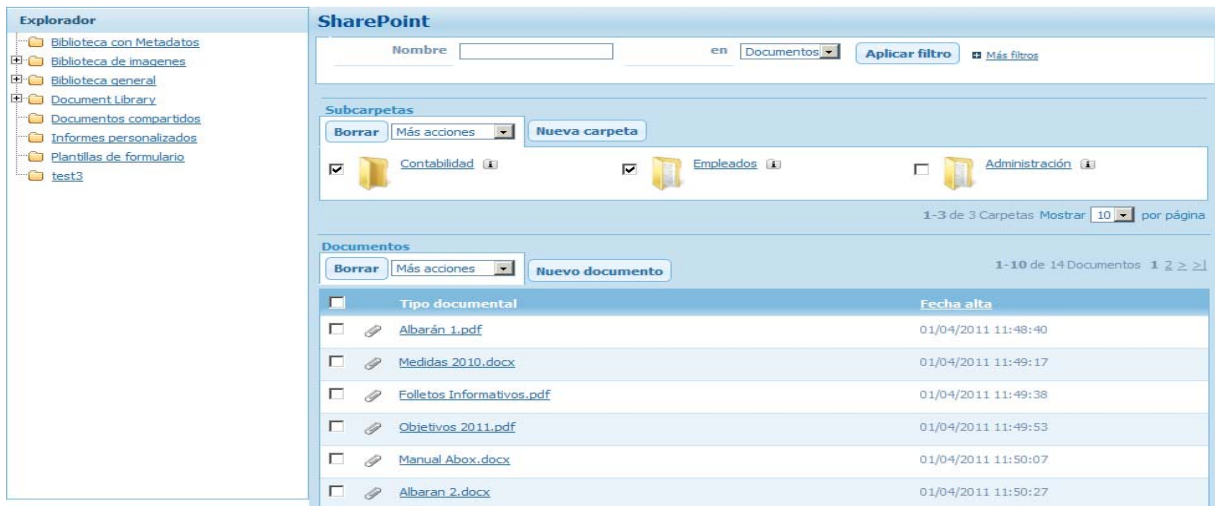


Figura 2.11. Selección de una carpeta

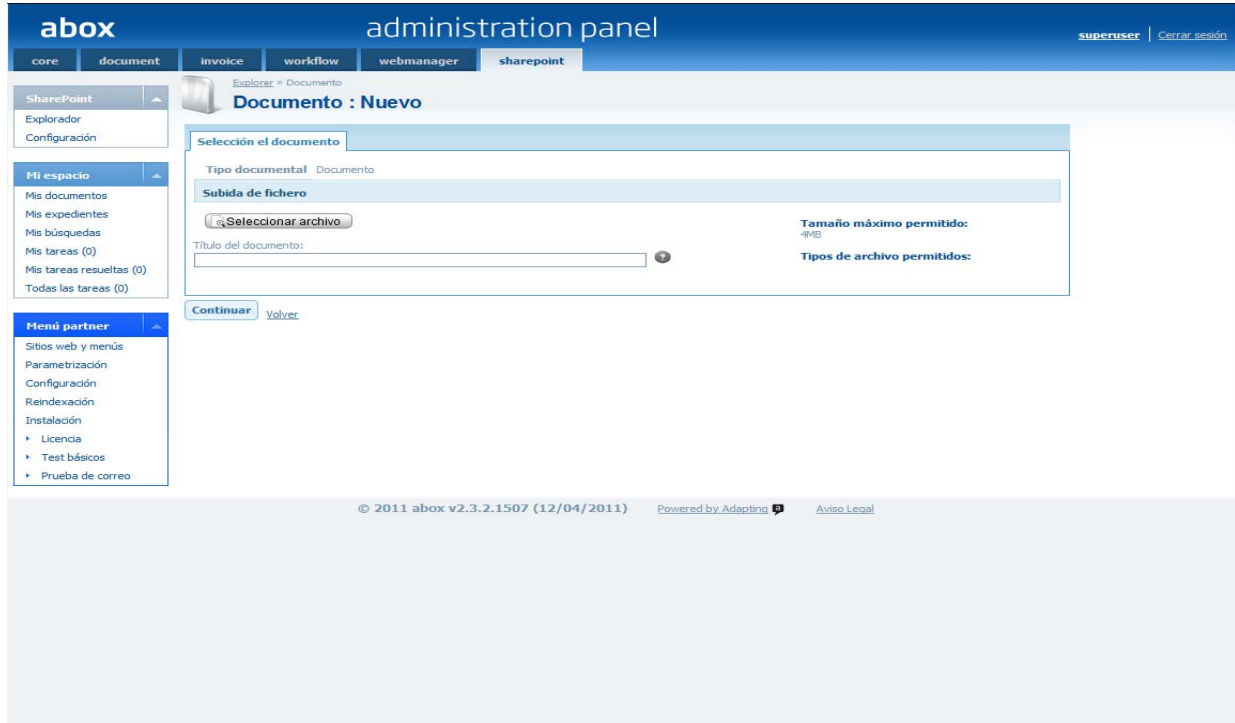


Figura 2.12. Nuevo documento

2.8. Borrado de documentos

Por cada fichero contenido dentro de una biblioteca o carpeta existe una casilla de verificación a su izquierda la cual puede ser marcada para realizar acciones sobre ese archivo.

En el caso de que deseáramos borrar un documento, tan solo tendríamos que marcar aquellos documentos que queramos eliminar pinchando en la casilla y volver a pinchar sobre el botón “Borrar” encima del listado de documentos.

2.9. Búsqueda de documentos

Desde cada una de las páginas del módulo de Abox, podemos encontrar en la esquina superior izquierda una caja de texto y botón llamado “Buscar”. Esto nos permite que desde cualquier punto de la aplicación sea posible realizar búsquedas. Un ejemplo de esta vista se puede ver en la figura 2.13.

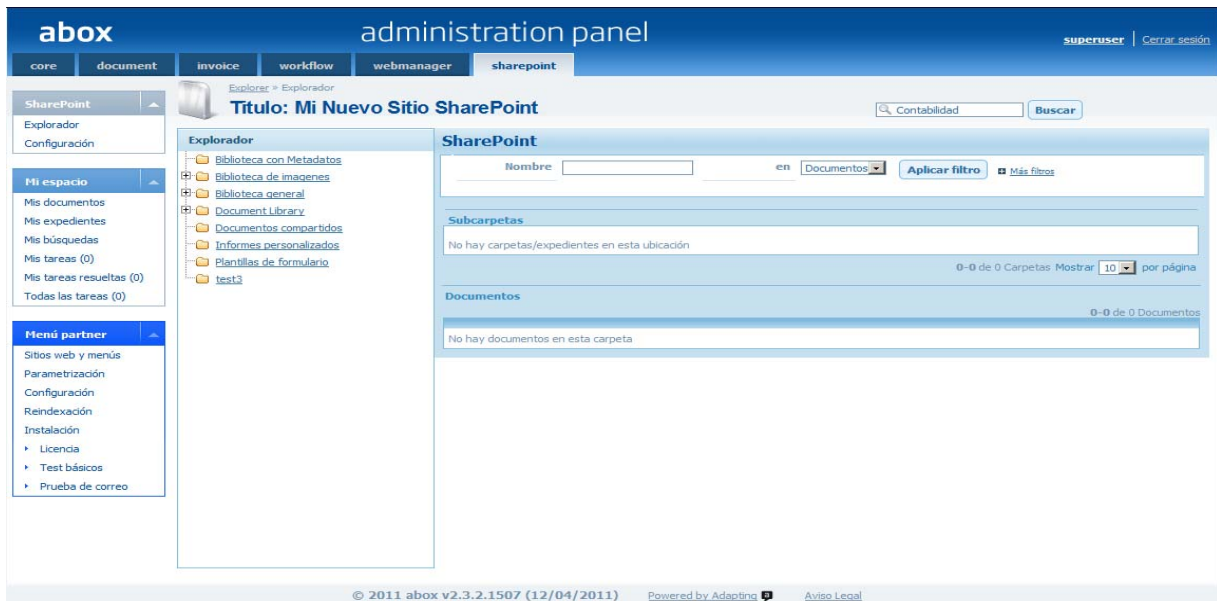


Figura 2.13. Búsqueda de un documento

Las búsquedas aceptan expresiones regulares, de esta forma podemos buscar una palabra concreta “Contabilidad” o podemos buscar toda aquella información que contiene SharePoint con la palabra “Cont*” .

En la figura 2.14 se muestran los resultados obtenidos al realizar una búsqueda con la palabra “Contabilidad”. En ella podemos observar como existen distintos tipos de resultados los cuales podremos diferenciar con los iconos que se les aplican. También podremos conocer en que carpeta están contenidos y en algunos casos las primeras líneas del documento.

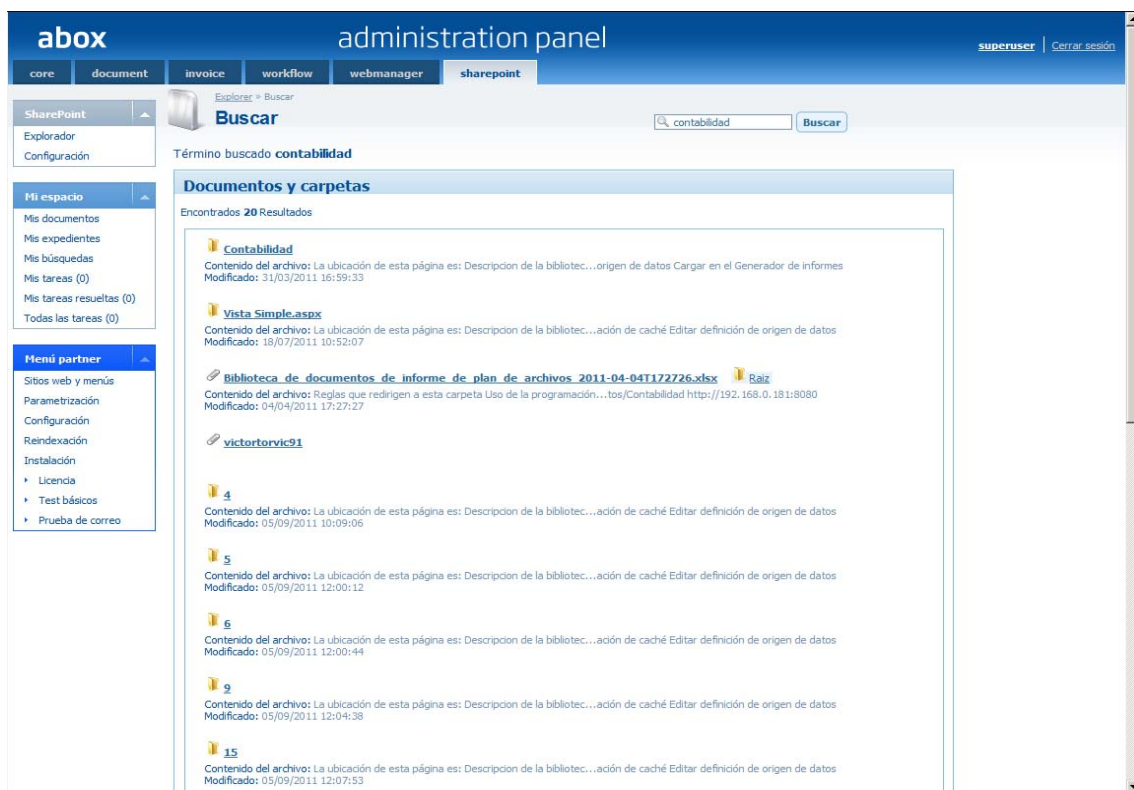


Figura 2.14. Resultados de búsqueda de un documento

A.2

Historias de usuario, tareas de
ingeniería y casos de prueba

1.1. Historias de usuario

<i>Historia de Usuario</i>	
Número: 1	Usuario: Operador
Nombre historia: Consultar los sitios web disponibles en SharePoint	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Humberto Delgado	
Descripción: La estructura de SharePoint empieza con los sitios webs. Estos son los que almacenan todo el contenido de la plataforma.	
Observaciones:	

<i>Historia de Usuario</i>	
Número: 2	Usuario: Operador
Nombre historia: Consulta de carpetas y documentos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Humberto Delgado	
Descripción: Se desean consultar las carpetas y documentos existentes en la plataforma SharePoint 2010, de manera que queden organizados de la misma forma en la que se representan en su origen.	
Observaciones: <i>Se consultan las carpetas y los archivos independientemente de los permisos.</i>	

<i>Historia de Usuario</i>	
Número: 3	Usuario: Operador
Nombre historia: Utilizar los usuarios existentes en SharePoint 2010	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Humberto Delgado	
Descripción: Antes de iniciar la aplicación se solicita el nombre de usuario y su contraseña. Los usuarios inician sesión en el sistema con el usuario que tienen en la plataforma SharePoint 2010	
Observaciones: Esta acción se realizara mediante LDAP	

<i>Historia de Usuario</i>	
Número: 4	Usuario: Operador
Nombre historia: Respetar los permisos de lectura	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Humberto Delgado	
Descripción: Existen distintos tipos de usuarios los cuales tienen acceso a diferente tipo de información contenida en el sistema.	
Observaciones:	

<i>Historia de Usuario</i>	
Número: 5	Usuario: Operador
Nombre historia: Respetar los permisos de escritura	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Humberto Delgado	
Descripción: Existen distintos tipos de usuarios los cuales tienen distintos permisos de escritura en la plataforma SharePoint 2010. Estos permisos se deben de mantener.	
Observaciones:	

<i>Historia de Usuario</i>	
Número: 6	Usuario: Operador
Nombre historia: Descarga de documentos	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Humberto Delgado	
Descripción: Se desea que se puedan descargar para trabajar en local con los archivos.	
Observaciones:	

<i>Historia de Usuario</i>	
Número: 7	Usuario: Operador
Nombre historia: Realizar búsquedas simples	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Humberto Delgado	
Descripción: Se desean poder realizar búsquedas simples en toda la plataforma SharePoint 2010.	
Observaciones: La búsqueda se realizara en aquellos lugares en los cuales tenga acceso el usuario que ha iniciado sesión.	

<i>Historia de Usuario</i>	
Número: 8	Usuario: Operador
Nombre historia: Realizar búsquedas avanzadas	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 4	Iteración asignada: 3
Programador responsable: Humberto Delgado	
Descripción: Se realizaran búsquedas en base a ciertos parámetros establecidos por el usuario, de manera que la búsqueda quede más centrada.	
Observaciones: Atributos como la fecha de creación, modificación, autor o la ubicación del archivo son requeridos por el cliente.	

<i>Historia de Usuario</i>	
Número: 9	Usuario: Operador
Nombre historia: Versiones de los documentos	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Humberto Delgado	
Descripción: Se requiere poder acceder a las distintas versiones de los documentos y obtener información de la versión del documento la cual se está consultando.	
Observaciones:	

<i>Historia de Usuario</i>	
Número: 10	Usuario: Operador
Nombre historia: Consultar los metadatos y otras informaciones de un documento	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 4	Iteración asignada: 4
Programador responsable: Humberto Delgado	
Descripción: Se desea obtener información de los metadatos que tiene cada documento asignado, así como otro tipo de información que contenga el documento	
Observaciones: No se tratara ningún tema relación con el <i>workflow</i> que tenga el archivo.	

Historia de Usuario	
Número: 11	Usuario: Operador
Nombre historia: Documentos y carpetas entre distintas plataformas	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 4	Iteración asignada: 5
Programador responsable: Diego Hernández	
Descripción: Se requiere que se puedan mover documentos entre las distintas plataformas, de forma que el contenido no quede duplicado.	
Observaciones: Todos los temas relacionados con el <i>workflow</i> quedaran obviados.	

Historia de Usuario	
Número: 12	Usuario: Operador
Nombre historia: Modificación de los metadatos y otras informaciones de un documento	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 5
Programador responsable: Diego Hernández	
Descripción: Se desea poder modificar información de los metadatos que tiene cada documento asignado, así como otro tipo de información que contenga el documento.	
Observaciones: Todos los temas relacionados con el <i>workflow</i> quedaran obviados.	

1.2. Tareas de ingeniería

<i>Tarea de ingeniería</i>	
Número tarea: 1	Historia de usuario : 1
Nombre tarea: Sitios webs de SharePoint 2010	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 7 de febrero del 2011	Fecha fin: 18 de febrero del 2011
Fecha inicio: 7 de febrero del 2011	
Programador responsable: Humberto Delgado	
Descripción: La estructura de SharePoint empieza con los sitios webs. Estos son los que almacenan todo el contenido de la plataforma.	

<i>Tarea de ingeniería</i>	
Número tarea: 2	Historia de usuario : 2
Nombre tarea: Obtener el listado de librerías de un sitio SharePoint 2010	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14 de febrero del 2011	Fecha fin: 25 de febrero del 2011
Fecha inicio: 15 de febrero del 2011	
Programador responsable: Diego Hernández	
Descripción: Un sitio web, está compuesto de librerías, que estas a su vez son las que contiene los documentos y las carpetas.	

Tarea de ingeniería	
Número tarea: 3	Historia de usuario : 2
Nombre tarea: Consultar el listado de carpetas y documentos de una librería.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 21 de febrero del 2011	Fecha fin: 4 de marzo del 2011
Fecha inicio: 23 de febrero del 2011	
Programador responsable: Humberto Delgado	
Descripción:	

Tarea de ingeniería	
Número tarea: 4	Historia de usuario : 2
Nombre tarea: Obtener la extensión de los archivos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 21 de febrero del 2011	Fecha fin: 24 de febrero del 2011
Fecha inicio: 23 de febrero del 2011	
Programador responsable: Humberto Delgado	
Descripción: Obtendremos la extensión de los archivos (.pdf, .txt, .docx) para obtener el <i>thumbnail</i> del archivo.	

<i>Tarea de ingeniería</i>	
Número tarea: 5	Historia de usuario : 2
Nombre tarea: Interfaz usable.	
Tipo de tarea: Diseño	Puntos estimados: 3
Fecha inicio: 7 de febrero del 2011	Fecha fin: 18 de febrero del 2011
Fecha inicio: 7 de febrero del 2011	
Programador responsable: Diego Hernández	
Descripción: La aplicación tener un buen diseño gráfico, buenos contenidos, fácil de usar y aprender por el usuario. Hay que tener presente que los usuarios de esta aplicación serán pequeñas organizaciones y por ellos la aplicación debe ser muy amigable.	

<i>Tarea de ingeniería</i>	
Número tarea: 6	Historia de usuario : 2
Nombre tarea: Obtener datos básicos.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 21 de febrero del 2011	Fecha fin: 25 de febrero del 2011
Fecha inicio: 21 de febrero del 2011	
Programador responsable: Diego Hernández	
Descripción: Por cada archivo se desea obtener la fecha de creación, la fecha de modificación y el autor.	

<i>Tarea de ingeniería</i>	
Número tarea: 7	Historia de usuario : 3
Nombre tarea: Obtener usuarios existentes en SharePoint.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 28 de febrero del 2011	Fecha fin: 28 de febrero del 2011
Fecha inicio: 2 de marzo del 2011	
Programador responsable: Humberto Delgado	
Descripción: Se desea obtener los usuarios dados de alta en la plataforma SharePoint 2010.	

<i>Tarea de ingeniería</i>	
Número tarea: 8	Historia de usuario : 3
Nombre tarea: Configuración de los datos del usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 28 de febrero del 2011	Fecha fin: 4 de marzo del 2011
Fecha inicio: 1 de marzo del 2011	
Programador responsable: Diego Hernández	
Descripción: Se creara una página, en la que el usuario introducirá el usuario y la contraseña que tiene en SharePoint 2010.	

<i>Tarea de ingeniería</i>	
Número tarea: 9	Historia de usuario : 3
Nombre tarea: Comprobación de los datos de configuración del usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 7 de marzo del 2011	Fecha fin: 11 de marzo del 2011
Fecha inicio: 7 de marzo del 2011	
Programador responsable: Humberto Delgado	
Descripción: Una vez que el usuario ha introducido los datos, debe existir una acción que comprueba si los datos introducidos son correctos y se puede acceder con ellos a la plataforma SharePoint.	

<i>Tarea de ingeniería</i>	
Número tarea: 10	Historia de usuario : 3
Nombre tarea: Conservación de los datos de configuración del usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 7 de marzo del 2011	Fecha fin: 11 de marzo del 2011
Fecha inicio: 7 de marzo del 2011	
Programador responsable: Diego Hernández	
Descripción: Después de comprobar que los datos son correctos, se debe permitir al usuario guardar los datos, para que no sean requeridos la próxima vez.	

<i>Tarea de ingeniería</i>	
Número tarea: 11	Historia de usuario : 4
Nombre tarea: Comprobación de permisos de lectura	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 14 de marzo del 2011	Fecha fin: 22 de marzo del 2011
Fecha inicio: 14 de marzo del 2011	
Programador responsable: Humberto Delgado	
Descripción: Una vez obtenido un listado de sitios web, librerías, carpetas o documentos, comprobar que el usuario tiene permisos de lectura para el objeto.	

<i>Tarea de ingeniería</i>	
Número tarea: 12	Historia de usuario : 5
Nombre tarea: Comprobación de permisos de escritura	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 22 de marzo del 2011	Fecha fin: 31 de marzo del 2011
Fecha inicio: 22 de marzo del 2011	
Programador responsable: Humberto Delgado	
Descripción: Una vez obtenido un listado de sitios web, librerías, carpetas o documentos, comprobar que el usuario tiene permisos de escritura para el objeto.	

<i>Tarea de ingeniería</i>	
Número tarea: 13	Historia de usuario : 6
Nombre tarea: Descarga de documentos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 26 de abril del 2011	Fecha fin: 6 de mayo del 2011
Fecha inicio: 26 de abril del 2011	
Programador responsable: Humberto Delgado	
Descripción: Cuando se hace <i>click</i> sobre el <i>thumbnail</i> en el listado de un documento, se debe permitir la descarga de la última versión del documento.	

<i>Tarea de ingeniería</i>	
Número tarea: 14	Historia de usuario : 7
Nombre tarea: Obtener acceso al <i>WebService</i> de búsqueda de SharePoint	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 28 de marzo del 2011	Fecha fin: 8 de abril del 2011
Fecha inicio: 28 de marzo del 2011	
Programador responsable: Diego Hernández	
Descripción:	

<i>Tarea de ingeniería</i>	
Número tarea: 15	Historia de usuario : 7
Nombre tarea: Creación de la consulta de búsquedas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 4 de abril del 2011	Fecha fin: 12 de abril del 2011
Fecha inicio: 4 de abril del 2011	
Programador responsable: Diego Hernández	
Descripción: Creación de la estructura de texto, que será enviada al <i>WebService</i> .	

<i>Tarea de ingeniería</i>	
Número tarea: 16	Historia de usuario : 7
Nombre tarea: Tratamiento de la respuesta del <i>WebService</i> de búsqueda.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 11 de abril del 2011	Fecha fin: 15 de abril del 2011
Fecha inicio: 11 de abril del 2011	
Programador responsable: Humberto Delgado	
Descripción: Una vez obtenida la respuesta, se debe transformar el documento XML en los objetos con los que se trabajan en el sistema.	

<i>Tarea de ingeniería</i>	
Número tarea: 17	Historia de usuario : 8
Nombre tarea: Búsqueda avanzada.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 9 de mayo del 2011	Fecha fin: 20 de mayo del 2011
Fecha inicio: 11 de mayo del 2011	
Programador responsable: Humberto Delgado	
Descripción: Clasificación, ordenación y filtrado de los resultados obtenidos por el <i>WebService</i> .	

<i>Tarea de ingeniería</i>	
Número tarea: 18	Historia de usuario : 9
Nombre tarea: Listado de versiones de un documento	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 23 de mayo del 2011	Fecha fin: 3 de junio del 2011
Fecha inicio: 25 de mayo del 2011	
Programador responsable: Humberto Delgado	
Descripción: En aquellos documentos que tienen la opción de versionado disponible, obtener el listado de documentos existentes y la versión que tiene cada documento.	

<i>Tarea de ingeniería</i>	
Número tarea: 19	Historia de usuario : 10
Nombre tarea: Obtener el tipo documental del archivo	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 30 de mayo del 2011	Fecha fin: 3 de junio del 2011
Fecha inicio: 30 de mayo del 2011	
Programador responsable: Diego Hernández	
Descripción: El tipo documental tiene asociados los metadatos que dispone el archivo.	

<i>Tarea de ingeniería</i>	
Número tarea: 20	Historia de usuario : 11
Nombre tarea: Utilización del HotFolder para el envío de ficheros de SharePoint a Abox	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 6 de junio del 2011	Fecha fin: 10 de junio del 2011
Fecha inicio: 6 de junio del 2011	
Programador responsable: Diego Hernández	
Descripción: Para el envío de archivo o documentos de SharePoint a Abox se utilizara la aplicación HotFolder la cual fue desarrollada para el envío de documentos a Abox.	

Tarea de ingeniería	
Número tarea: 21	Historia de usuario : 11
Nombre tarea: Envío de ficheros de Abox a SharePoint	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 6 de junio del 2011	Fecha fin: 16 de junio del 2011
Fecha inicio: 8 de junio del 2011	
Programador responsable: Humberto Delgado	
Descripción:	

Tarea de ingeniería	
Número tarea: 22	Historia de usuario : 12
Nombre tarea: Modificación de los metadatos del documento	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 20 de junio del 2011	Fecha fin: 5 de julio del 2011
Fecha inicio: 20 de junio del 2011	
Programador responsable: Humberto Delgado	
Descripción: El usuario debe ser capaz de modificar los metadatos existentes a la vez de poder añadir nuevos.	

1.2. Casos de pruebas de aceptación

Caso de Prueba de Aceptación	
Código: 1	Historia de usuario : 1
Nombre: Sitios web	
Descripción: La estructura jerárquica de SharePoint empieza desde los sitios web. Nuestro objetivo es consultar la existencia de ese sitio web. En nuestra máquina de pruebas existen dos sitios web SharePoint, uno en el puerto 8080 y otro en el puerto 9090.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1) Iniciar sesión en el sistema 2) Acceder al módulo SharePoint 3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información de cada uno de los sitios web de SharePoint. 4) Pinchar sobre el botón "test conexión" 	
Resultados esperados: Tiene que mostrarse un mensaje confirmando la existencia del sitio web para el cual se ha rellenado la información.	
Evaluación de la prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código: 2	Historia de usuario : 2
Nombre: Listado de librerías de un sitio web de SharePoint	
Descripción: Se desea consultar el listado de bibliotecas existentes del sitio web situado en el puerto 8080.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 5) Iniciar sesión en el sistema 6) Acceder al módulo SharePoint 7) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web sobre el que se desea consultar las librerías. Una vez listo pinchar sobre el botón guardar. 8) Ir a la opción explorador, en la cual se mostraran todo el listado de bibliotecas del sitio web. 	
Resultados esperados: El resultado tiene que mostrar un listado con 6 bibliotecas.	
Evaluación de la prueba: Satisfactorio	

Caso de Prueba de Aceptación	
Código: 3	Historia de usuario : 2
Nombre: Listado de carpetas de la raíz de la biblioteca “Documentos compartidos” I	
Descripción: Se desea comprobar el correcto listado de carpetas que contiene una biblioteca.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema2) Acceder al módulo SharePoint3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. Una vez listo pinchar sobre el botón guardar.4) Ir a la opción explorador, en la cual se mostraran todo el listado de bibliotecas del sitio web y buscar la biblioteca correspondiente.	
Resultados esperados: Se muestran 10 carpetas ordenadas por nombre.	
Evaluación de la prueba: Satisfactorio	

Caso de Prueba de Aceptación	
Código: 4	Historia de usuario : 2
Nombre: Listado de documentos de la raíz de la biblioteca “Documentos compartidos” II	
Descripción: Se desea comprobar el correcto listado de documentos que contiene la biblioteca “Documentos compartidos”.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema2) Acceder al módulo SharePoint3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. Una vez listo pinchar sobre el botón guardar.4) Ir a la opción explorador, en la cual se mostraran todo el listado de bibliotecas del sitio web y buscar la biblioteca “Documentos compartidos”.	
Resultados esperados: Se muestran 3 documentos ordenadas por nombre.	
Evaluación de la prueba: Satisfactorio	

Caso de Prueba de Aceptación	
Código: 5	Historia de usuario : 3
Nombre: Iniciar sesión con un usuario específico.	
Descripción: Se desea poder utilizar los usuarios existentes en SharePoint. El objetivo de esta prueba es que se pueda iniciar sesión con el usuario "h.delgado" que esta dado de alta en SharePoint.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1) Iniciar sesión en el sistema 2) Acceder al módulo SharePoint 3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. 4) Rellenar la información correspondiente al usuario "h.delgado" 5) Pinchar sobre el botón "Test conexión" 	
Resultados esperados: Se debe mostrar un mensaje confirmado que ha sido posible iniciar sesión con el usuario introducido.	
Evaluación de la prueba: Satisfactorio	

Caso de Prueba de Aceptación	
Código: 6	Historia de usuario : 3
Nombre: Comprobar que se guardan los datos de inicio de sesión.	
Descripción: Se desea comprobar la persistencia de los datos que se introducen en el inicio de sesión de un usuario.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1) Iniciar sesión en el sistema 2) Acceder al módulo SharePoint 3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. 4) Rellenar la información correspondiente al usuario "h.delgado" 5) Pinchar sobre el botón "Guardar" 6) Cerrar la ventana, borrar las cookies, borrar el historial y la información relevante con el inicio de sesión. 7) Volver a iniciar sesión en Abox, con el usuario anterior. Acceder a la configuración del módulo de SharePoint. 	
Resultados esperados: Deben mostrarse los datos anteriormente guardados en el sistema.	
Evaluación de la prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código: 7	Historia de usuario : 3
Nombre: Comprobar que se guardan los datos de inicio de sesión.	
Descripción: Se desea comprobar la persistencia de los datos que se introducen en el inicio de sesión de un usuario.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema2) Acceder al módulo SharePoint3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web.4) Rellenar la información correspondiente al usuario "h.delgado"5) Pinchar sobre el botón "Guardar"6) Cerrar la ventana, borrar las cookies, borrar el historial y la información relevante con el inicio de sesión.7) Volver a iniciar sesión en Abox, con el usuario anterior. Acceder a la configuración del módulo de SharePoint.	
Resultados esperados: Deben mostrarse los datos anteriormente guardados en el sistema.	
Evaluación de la prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código: 8	Historia de usuario : 4
Nombre: Lectura de la biblioteca "Documentos privados"	
Descripción: Se desea comprobar el correcto funcionamiento de los permisos de lectura. Para ello se intentara acceder con un usuario que no tiene permiso de lectura a una biblioteca para la cual no tiene permisos.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema2) Acceder al módulo SharePoint3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web.4) Rellenar la información correspondiente al usuario "h.delgado"5) Pinchar sobre el botón "Guardar"6) Dirigirse hacia la opción "Explorador" y buscar la biblioteca "Documentos privados"	
Resultados esperados: La biblioteca no tiene que mostrarse en el listado.	
Evaluación de la prueba: Satisfactorio	

Caso de Prueba de Aceptación	
Código: 9	Historia de usuario : 5
Nombre: Modificar archivos en una biblioteca	
Descripción: Se desea poder modificar un documento ya existente en una biblioteca. En este caso el usuario no tiene los permisos suficientes para poder modificar o escribir en la misma, tan solo para poder leer documentos.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1) Iniciar sesión en el sistema 2) Acceder al módulo SharePoint 3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. 4) Rellenar la información correspondiente al usuario "h.delgado" 5) Pinchar sobre el botón "Guardar" 6) Dirigirse hacia la opción "Explorador" y buscar la biblioteca "Documentos restringidos" 7) Descargarse el documento "Prueba de escritura.docx" modificarlo y volver a subirlo. 	
Resultados esperados: Se tiene que mostrar un mensaje informando al usuario de que carece de los permisos suficientes para poder realizar la acción deseada.	
Evaluación de la prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código: 10	Historia de usuario : 6
Nombre: Descarga de documentos	
Descripción: Se desea comprobar que un usuario con los permisos adecuados puede descargarse un documento de la plataforma SharePoint a través de Abox, siempre respetando los permisos.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1) Iniciar sesión en el sistema 2) Acceder al módulo SharePoint 3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. 4) Rellenar la información correspondiente al usuario "h.delgado" 5) Pinchar sobre el botón "Guardar" 6) Dirigirse hacia la opción "Explorador" y buscar la biblioteca "Documentos restringidos" 7) Descargarse el documento "Prueba de escritura.docx" 	
Resultados esperados: Se permite la descarga del documento.	
Evaluación de la prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código: 11	Historia de usuario : 7
Nombre: Búsqueda simple	
Descripción: El objetivo de esta prueba es realizar una búsqueda sobre un término concreto y comprobar que devuelve un número de resultados concretos.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema2) Acceder al módulo SharePoint3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web.4) Rellenar la información correspondiente al usuario "h.delgado"5) Pinchar sobre el botón "Guardar"6) Dirigirse hacia la opción "Explorador"7) En la parte superior derecha debe mostrarse una caja de texto y un botón buscar. Introducir en la caja de texto "Adapting" y presionar sobre el botón buscar.	
Resultados esperados: El resultado de la búsqueda tienen que ser el siguiente: <ul style="list-style-type: none">- 3 archivos- 2 carpetas	
Evaluación de la prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código: 12	Historia de usuario : 8
Nombre: Búsqueda avanzada	
Descripción: El objetivo de esta prueba es realizar una búsqueda sobre un término y aplicar expresiones regulares.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema2) Acceder al módulo SharePoint3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web.4) Rellenar la información correspondiente al usuario "h.delgado"5) Pinchar sobre el botón "Guardar"6) Dirigirse hacia la opción "Explorador"7) En la parte superior derecha debe mostrarse una caja de texto y un botón buscar. Introducir en el caja de texto "Adapt*" y presionar sobre el botón buscar.	
Resultados esperados: El resultado de la búsqueda tienen que ser el siguiente: <ul style="list-style-type: none">- 3 archivos- 2 carpetas	
Evaluación de la prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código: 13	Historia de usuario : 9
Nombre: Versiones de documentos	
Descripción: En la plataforma SharePoint existe una biblioteca con la característica de versionado activada. Dentro de ella existe un archivo con diferentes versiones. El objetivo de esta prueba es comprobar la correcta visualización de las distintas versiones que tiene el archivo.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1) Iniciar sesión en el sistema 2) Acceder al módulo SharePoint 3) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. 4) Rellenar la información correspondiente al usuario "h.delgado" 5) Pinchar sobre el botón "Guardar" 6) Dirigirse hacia la opción "Explorador" y entrar dentro de la biblioteca con el nombre "Biblioteca versionada" 7) La raíz de la biblioteca contendrá un documento denominado "TestVersionado.docx", sobre el cual pincharemos e iremos a la pestaña de versionado. 	
Resultados esperados: Se tendrá que mostrar un listado de 3 archivos con el mismo nombre cada uno de ellos.	
Evaluación de la prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código: 14	Historia de usuario : 10
Nombre: Consulta de los metadatos de un archivo.	
Descripción: Deseamos consultar los metadatos que SharePoint tiene sobre cada archivo de la plataforma.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1) Iniciar sesión en el sistema y acceder al módulo SharePoint 2) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web. 3) Rellenar la información correspondiente al usuario "h.delgado" 4) Pinchar sobre el botón "Guardar" 5) Dirigirse hacia la opción "Explorador" y entrar dentro de la biblioteca con el nombre "Biblioteca de imágenes" 6) La raíz de la biblioteca contendrá un documento denominado "arbol.jpg", sobre el cual pincharemos e iremos a la pestaña de metadatos. 	
Resultados esperados: En la pestaña se mostraran distintos metadatos entre ellos el autor, tamaño, calidad y la situación.	
Evaluación de la prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código: 15	Historia de usuario : 11
Nombre: Enviar documento de la SharePoint a Abox	
Descripción: Se desea seleccionar un documento y poder enviarlo a la plataforma Abox.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema y acceder al módulo SharePoint2) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web y rellenar la información correspondiente al usuario "h.delgado"3) Pinchar sobre el botón "Guardar"4) Dirigirse hacia la opción "Explorador" y entrar dentro de la biblioteca con el nombre "Biblioteca de imágenes"5) La raíz de la biblioteca contendrá un documento denominado "arbol.jpg", sobre el cual pincharemos.6) Pincharemos sobre la opción "enviar a Abox" y seleccionaremos la ubicación donde queremos guardar el archivo.	
Resultados esperados: Desde la plataforma Abox debe estar disponible este archivo.	
Evaluación de la prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código: 16	Historia de usuario : 12
Nombre: Modificar el metadato situación de un archivo de imagen.	
Descripción: Se desea modificar un metadato de un archivo de imagen que actualmente existe en la plataforma SharePoint.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none">1) Iniciar sesión en el sistema y acceder al módulo SharePoint2) Ir a la opción de menú de configuración y rellenar los campos que se presentan con la información del sitio web.3) Rellenar la información del usuario "h.delgado" y pinchar sobre el botón "Guardar"4) Dirigirse hacia la opción "Explorador" y entrar dentro de la biblioteca con el nombre "Biblioteca de imágenes"5) La raíz de la biblioteca contendrá un documento denominado "arbol.jpg", sobre el cual pincharemos e iremos a la pestaña de metadatos.6) Pincharemos sobre la opción editar y en el campo de texto "situación" cambiaremos su valor y pincharemos en el botón guardar.	
Resultados esperados: Si consultamos desde la plataforma SharePoint el metadato de ese mismo archivo, tiene que ser igual al valor introducido anteriormente.	
Evaluación de la prueba: Satisfactoria.	

