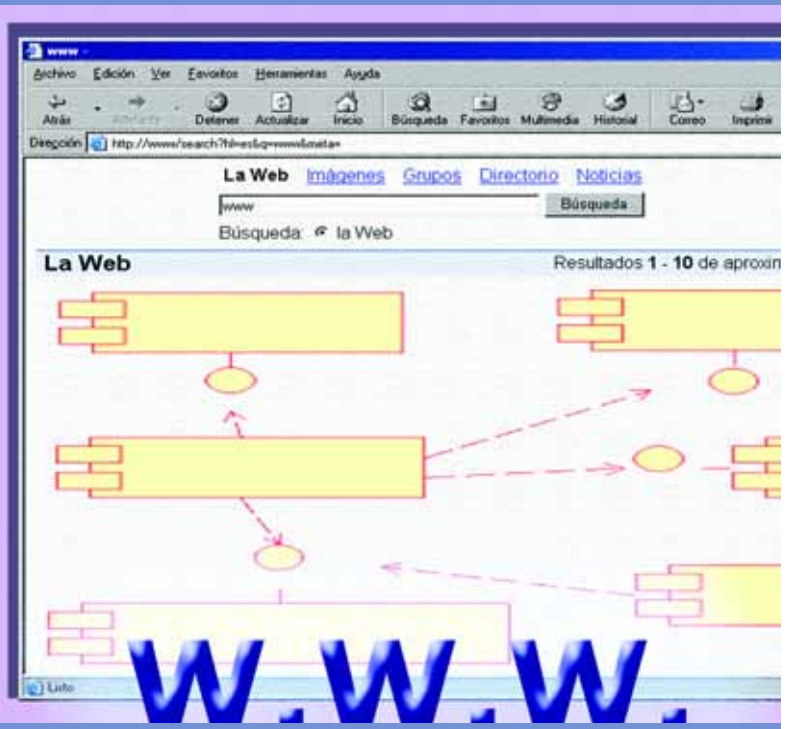
Software libre

Alberto Otero García

XP06/M2119/02156



Proyecto web

UOC
www.uoc.edu

David Megías Jiménez

Coordinador

Ingeniero en Informática por la UAB.

Magíster en Técnicas Avanzadas de Automatización de Procesos por la UAB.

Doctor en Informática por la UAB.

Profesor de los Estudios de Informática y Multimedia de la UOC.

Jordi Mas

Coordinador

Coordinador general de Softcatalà y desarrollador del procesador de textos libre Abiword.

Miembro fundador de Softcatalà y de la red telemática RedBBS.

En calidad de consultor, ha trabajado en empresas como Menta, Telépolis, Vodafone, Lotus, eresMas, Amena y Terra España.

Alberto Otero García

Autor

Ingeniero de Informática por la Universidad Ramon Llull. Licenciado en Investigación y Técnicas de Mercado por la UOC. Socio fundador y jefe de proyectos de Cometa Technologies, empresa dedicada a dar soluciones en tecnologías de la información, basadas en el uso de estándares y herramientas de código abierto.

Profesor titular de la asignatura Administración de Sistemas Operativos en Enginyeria i Arquitectura La Salle y consultor del Master Internacional en Software Libre de la UOC.

Segunda edición: febrero 2007

Av. Tibidabo, 39-43, 08035 Barcelona
Material realizado por Eureca Media, SL

Índice

Agradecimientos 5

Introducción 7

Objetivos 9

1. Estudio de viabilidad 11
1.1. Establecimiento del alcance del sistema 12
1.2. Estudio de la situación actual 15
1.3. Definición de los requisitos del sistema 18
1.4. Estudio de las alternativas de solución 21
1.5. Valoración de las alternativas 23
1.6. Selección de la solución 26

2. Análisis del sistema 29
2.1. Definición del sistema 29
2.2. Establecimiento de requisitos 33
2.3. Definición de interfaces de usuario 37
2.4. Especificación del plan de pruebas 40

3. Diseño del sistema 43
3.1. Arquitectura 44
3.1.1. Definición de niveles de arquitectura 44
3.1.2. Especificación de estándares, normas de diseño y construcción 47
3.1.3. Identificación de subsistemas 49
3.2. Revisión de casos de uso 50
3.2.1. Revisión de los subsistemas según los casos de uso 51
3.2.2. Elección de alternativas de componentes y licencias más adecuadas 53
3.2.3. Especificaciones de desarrollo y pruebas 56
3.2.4. Requisitos de implantación 59

4. Desarrollo 63
4.1. Planificación de las actividades de desarrollo e integración de sistema 64
4.2. Desarrollo 67
4.3. Documentación 68

5. Implantación 71
5.1. Formación 72
5.2. Implantación del sistema y pruebas 72
5.3. Nivel de servicios 74
5.4. Aceptación del sistema 74

6. Mantenimiento 75

Resumen 77

Bibliografía 79

Appendix A. GNU Free Documentation License 80

Agradecimientos

Los autores agradecen a la Fundación para la Universitat Oberta de Catalunya (http://www.uoc.edu) la financiación de la primera edición de esta obra, enmarcada en el Máster Internacional en Software Libre ofrecido por la citada institución.

ANOTACIONS

Introducción

Para llevar a cabo un **proyecto** de tecnologías de la información basado en la utilización de **herramientas de uso habitual en Internet** (como por ejemplo la World Wide Web), en entornos de software libre, como en cualquier otro tipo de proyecto, es necesario seguir un proceso que nos lleve desde la **comprensión del alcance del problema que queremos solventar hasta la implantación y mantenimiento** de la solución que hayamos elegido.

En este curso se pretenden repasar aquellas **fases** que es necesario seguir a lo largo de todo proyecto, tomando como referencia uno que basará su funcionamiento en la utilización de la web como herramienta principal.

Estas fases son las siguientes:

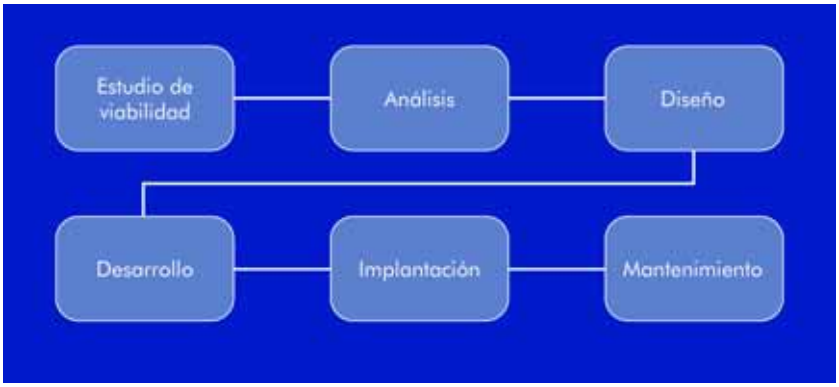
- **Estudio de viabilidad**: se estudiará en líneas generales qué problemas se desean resolver, qué soluciones posibles existen y cuál de ellas es la más adecuada.
- **Análisis**: se describirá detalladamente el sistema que se desea construir, qué requisitos debe cumplir y a qué usuarios debe satisfacer.
- **Diseño**: se realizará el planteamiento tecnológico de la solución.
- **Desarrollo**: se llevará a cabo la programación, integración, instalación, etc. de los diferentes subsistemas que compongan el proyecto.
- **Implantación**: se pasará el sistema construido a producción con el fin de que los usuarios de éste empiecen a utilizarlo.
- **Mantenimiento**: se realizarán tanto las correcciones de los posibles errores que puedan surgir en el sistema implantado, como las mejoras evolutivas que se consideren oportunas.

ANOTACIONS

Nota

A lo largo de todo el material del curso, se desarrolla un caso práctico con el fin de ejemplificar las explicaciones dadas. Este caso práctico no constituye de ningún modo un estudio exhaustivo del proyecto propuesto, sino que simplemente sirve como marco para ofrecer diferentes ejemplos de las partes de que se compone con fines únicamente didácticos.

Figura 1. Fases de un proyecto web



Estas fases estarán presentes, de una u otra forma, con estos nombres o con otros, en cualquier proyecto web, desde los gestionados mediante métodos "clásicos" (por ejemplo, en fases seguidas secuencialmente, en cascada, etc.), hasta los gestionados como sugiere el conjunto de metodologías conocidas como ágiles.

ANOTACIONS

Objetivos

Los **objetivos** que el lector deberá haber alcanzado al finalizar el curso de "Proyecto web" son los siguientes:

- Haber comprendido de manera global lo que representa llevar a cabo un proyecto basado en tecnologías de Internet, especialmente web, en un entorno tecnológico de software libre.
- Haber asimilado qué fases integran un proyecto de tecnologías de la información, y qué tareas se deben llevar a cabo en cada una de ellas, especialmente desde el punto de vista del desarrollo de soluciones basadas en web.
- Haber reflexionado sobre qué herramientas de software libre pueden ayudar en cada una de las fases de un proyecto web.
- Haber aplicado a un caso práctico los conocimientos adquiridos a lo largo de todo el máster en software libre.

ANOTACIONS

1. Estudio de viabilidad

El objetivo de la realización del **estudio de viabilidad** es el de, dado un conjunto de necesidades planteadas, elegir aquella solución que mejor las cubra de entre todas las posibles (o descartarlas todas en caso de que ninguna las satisfaga).

En el estudio de viabilidad se considerarán las diferentes soluciones posibles, teniendo en cuenta:

- El estado inicial del sistema.
- La situación actual.
- Los requisitos planteados.

Cada una de las **soluciones propuestas** en el estudio de viabilidad deberá recoger los siguientes aspectos:

- **Económicos**: se deberá incluir un estudio económico preliminar que contemple los costes asociados a cada una de las soluciones.
- **Técnicos**: se deberá incluir un estudio técnico preliminar de cada una de las soluciones.
- **Legales**: se deberá incluir un estudio de aquellos aspectos legales que puedan influir en la viabilidad de la solución.
- **Operativos**: se deberá incluir un estudio previo de la operativa de cada una de las soluciones propuestas.

Una vez planteadas cada una de las soluciones, se elegirá la mejor teniendo en cuenta:

- El impacto en la organización.
- La inversión que hay que realizar.
- Los riesgos asociados.

Los siguientes apartados describen con más detalle cada una de las tareas que hay que llevar a cabo para realizar el estudio de viabilidad.

ANOTACIONS

1.1. Establecimiento del alcance del sistema

En esta fase del estudio de viabilidad se pretende estudiar el alcance de las **necesidades planteadas por el cliente** (bien sean terceros, en caso de tratarse de un proyecto web dirigido a otras organizaciones, bien sean usuarios internos, en caso de tratarse de un proyecto para la propia organización).

Lo primero que será necesario hacer es la **descripción general** de las necesidades planteadas por el cliente. En esta descripción general se deberán incluir los aspectos básicos descritos en el anterior apartado (económicos, técnicos, legales y operativos) que tengan especial relevancia.

Caso práctico

Sitio web corporativo de Soluciones Abiertas, S. A.

La empresa Soluciones Abiertas, S. A., se dedica a la prestación de servicios relacionados con el software que ellos mismos desarrollan y que liberan en algunos casos bajo licencias libres. Los ingresos de Soluciones Abiertas provienen principalmente de la facturación realizada a sus clientes en concepto de horas/hombre dedicadas a la prestación de dichos servicios.

Con el fin de hacer más comprensible y atractiva su oferta, Soluciones Abiertas, S. A., ha decidido renovar su sitio web corporativo, teniendo como principales objetivos:

- Presentar la información del tipo "presencial" (p. ej. "Quiénes somos" "Qué ofrecemos") de forma visualmente más atractiva que en la actualidad.
- Implantar un nuevo sistema de comercialización de paquetes de horas de prestación de servicios a través de comercio electrónico, haciendo posible la contratación completa y el pago de éstos a través de su sitio web.
- Disfrutar de un sistema de gestión de contenidos (en lo referente tanto a la parte "presencial" como a la de "comercio electrónico" del sitio web), facilitando su actualización y haciendo que ésta pueda ser llevada a cabo por personas no técnicas.

ANOTACIONS

Figura 1-1. Descripción general del sistema

Desde el punto de vista económico, para que la renovación sea viable deberá implicar el menor gasto posible, dado que el coste de este proyecto no estaba contemplado inicialmente en los presupuestos anuales de la empresa.

A nivel técnico, las necesidades planteadas son muy poco restrictivas, ya que al ser Soluciones Abiertas una empresa dedicada a las tecnologías de la información, se dispone de personal cualificado que afronta y disfruta cualquier reto técnico sin mayores problemas.

Desde el punto de vista legal, se exige que las soluciones aportadas sean lo más flexibles posibles, ya que se valora muy negativamente el hecho de no disponer de la máxima libertad para copiar y/o modificar los sistemas software que se implanten.

A nivel operativo, la única necesidad planteada consiste en que en ningún caso se debe perder funcionalidad de la que en estos momentos ya se dispone en el sitio web de la empresa (p. ej. mostrar cualquier tipo de contenido, ordenarlo en secciones y subsecciones, posibilitar la descarga de ficheros), sino en todo caso, ampliarla.

Asimismo, además de describir de forma general el proyecto, se debe tener en cuenta **cómo afectará** éste a:

- **Otros proyectos de tecnologías de la información** ya en curso o que se piensan poner en marcha.
- Las diferentes **unidades de la organización**, teniendo en cuenta quiénes son los responsables de éstas y cuál es su estructura.

ANOTACIONS

Caso práctico

Alcance del proyecto

En el caso de Soluciones Abiertas, el proyecto de renovación de su sitio web corporativo afectará a:

- El proyecto de renovación del hardware de la empresa Soluciones Abiertas tiene en la actualidad un ordenador dedicado a servir su sitio web corporativo, que estaba previsto renovar junto con otros dedicados a otras funciones. Será necesario coordinar ambos proyectos para que el/los nuevo/s servidor/es web cumpla/n con las necesidades marcadas en el proyecto de renovación.

- El proyecto de digitalización total de los procesos de la empresa Soluciones Abiertas ha puesto en marcha un proyecto de "organización sin papel", lo cual implica que puede existir información que se desea compartir en el sitio web de la empresa (p. ej. los comentarios de las encuestas de satisfacción de los clientes) y que antes no podía ser fácilmente incorporada a éste (p. ej. porque se disponía de ella únicamente en formato papel).

Por otro lado, el proyecto de renovación del sitio web de la empresa afectará a los siguientes departamentos de ésta:

- Marketing: la renovación a nivel gráfico y de contenidos del sitio web debe ser liderada por el equipo del departamento de marketing, ya que es éste el que se encarga de la comunicación externa en la empresa. Además se abre un nuevo canal de comercialización de los servicios de la organización, con todo lo que ello comporta (posible canibalización de otros canales, necesidad de unificación y simplificación de las tarifas de precios aplicados, etc.).

- Administración: la posibilidad de que los clientes contraten los servicios ofrecidos por Soluciones Abiertas directamente a través de su sitio web implica la necesidad de crear un nuevo canal de pago (además de la domiciliación bancaria y la recepción de transferencias), la emisión de facturas automáticamente (manteniendo la coherencia con las generadas a través del programa de contabilidad), etc.

ANOTACIONS

- Atención al cliente: las personas dedicadas a la atención al cliente deberán poder solucionar nuevas consultas, como las referentes a los métodos de pago a través del sitio web, reclamaciones en referencia a éstos, dudas acerca de la contratación directa en lugar de a través de un agente comercial, etc.

- Producción/Servicios: el hecho de ofrecer la contratación de servicios mediante Internet puede suponer que el número de clientes extranjeros aumente, ya que la oferta comercial de la empresa se hará más palpable para este público. Este hecho implica que las personas encargadas de la prestación de los servicios a los clientes deben estar preparadas para realizar su labor dependiendo cada vez menos de su presencia física (intervención a través de conexiones remotas, entrega de informes por correo electrónico sin posibilidad de realizar reuniones presenciales posteriores, etc.). Además, a la hora de prestar el servicio se deberán tener en cuenta las prioridades marcadas por los tipos de contrataciones disponibles en el nuevo sitio web (p. ej. "platinum", con respuesta y solución en 12 h, "gold", con respuesta en 24 h y solución "silver" en 48 h).

Se deberá informar puntualmente e implicar a los responsables de cada uno de los departamentos de los detalles y el alcance del proyecto, en la medida en que se vean afectados.

1.2. Estudio de la situación actual

En esta fase del estudio de viabilidad se pretende **estudiar la situación en la que se encuentra el sistema de información de la empresa y realizar un diagnóstico de éste**, siempre en lo referente al proyecto que nos ocupa.

La primera tarea que hay que realizar dentro del estudio de la situación actual es la de **identificar aquellos sistemas que deben descri-**

ANOTACIONS

birse, esto es, de qué sistemas vamos a hacer el estudio debido a que se ven afectados de alguna manera por el proyecto contemplado en el estudio de viabilidad. Es interesante también fijar qué usuarios participarán en el estudio de la situación actual de cada uno de los sistemas escogidos.

Caso práctico

Identificación de los sistemas actuales

Dado que el proyecto consiste en la renovación y ampliación del sitio web corporativo de Soluciones Abiertas, deberemos estudiar como mínimo la situación actual del sistema que deseamos renovar. En este caso este sistema será el del sitio web actual. Con el fin de realizar un diagnóstico lo más completo posible, hemos decidido trabajar junto con el personal de marketing, ya que es este departamento el que hace un uso más intensivo del mencionado sistema.

Por otro lado, dado que el proyecto sobre el que se está haciendo el estudio de viabilidad implica la comercialización de servicios directamente a través del sitio web de la empresa, se ha decidido estudiar también el estado de las aplicaciones de gestión con las cuales éste se tendrá que integrar (p. ej. la de contabilidad). Este trabajo se hará junto con el personal del departamento de administración.

El siguiente paso dentro del estudio de la situación actual será el de **describir cada uno de los sistemas identificados en el paso anterior**. La descripción se hará teniendo en cuenta la información recogida en las sesiones de trabajo con los usuarios seleccionados como representativos, y deberá alcanzar el nivel de detalle suficiente como para poder realizar un diagnóstico acertado del estado real de cada uno de los sistemas estudiados. Asimismo, la dedicación de recursos en esta fase dependerá de la información de partida de la cual se disponga (la descripción de la situación actual puede ser trivial en algunos casos o tremendamente complicada en otros).

ANOTACIONS

Caso práctico

Descripción de los sistemas actuales

A continuación se describe la situación actual de los sistemas de Soluciones Abiertas seleccionados con anterioridad:

- Sistema web: en la actualidad este sistema consiste en una serie de páginas HTML que son gestionadas a través de un editor web. Este editor web debe ser utilizado por una persona técnica o bien por una persona no técnica con conocimientos amplios de HTML, JavaScript y CSS.

Por otro lado, el sistema contiene también una pequeña aplicación que permite la introducción de las novedades y noticias que son mostradas en la página principal del sitio.

Figura 1-2. Descripción sistema web actual

- Aplicaciones de gestión: en la actualidad estos sistemas contemplan las vías de pago y facturación de servicios que hasta ahora eran habituales, es decir, la transferencia y domiciliación bancarias y la emisión de facturas de forma supervisada.

Para completar el estudio de la situación actual de los sistemas, se deberá realizar un **diagnóstico** de éstos, es decir, analizar la información obtenida detectando posibles problemas y puntos de mejora.

ANOTACIONS

Caso práctico

Diagnóstico de los sistemas actuales

Una vez analizada la información obtenida en la descripción de la situación actual de los sistemas estudiados en la empresa Soluciones Abiertas, se ha llegado a las siguientes conclusiones:

- Sistema web: se ha detectado que la posibilidad de gestión del contenido del sitio web corporativo es muy limitada, ya que únicamente permite que la sección de novedades y noticias sea verdaderamente dinámica y por tanto se actualice cada cierto tiempo. Además, los conocimientos necesarios para la gestión de los contenidos del sitio son demasiados, ya que esta labor debería poder ser llevada a cabo por personal experto en comunicación, no en tecnología. El personal de marketing consultado ha dejado también patente la necesidad de actualizar el sitio web a nivel visual.
- Aplicaciones de gestión: se ha detectado la necesidad de incorporar a estas aplicaciones nuevas funcionalidades que permitan explotar las posibilidades que brindan los métodos de comercialización que se pretenden poner en práctica. Estos cambios se podrán incorporar al mantenimiento evolutivo del sistema que ya se está realizando, ya que éste está vivo y ha venido cambiando junto con las necesidades de la empresa desde que se implantó.

1.3. Definición de los requisitos del sistema

Una vez descrita la situación actual del sistema, y teniendo en cuenta las opiniones de los diferentes usuarios implicados, se pasará a **describir de forma general los requisitos que deberá cumplir el proyecto** del cual se está estudiando la viabilidad.

La descripción del conjunto de requisitos a cumplir por el proyecto servirá posteriormente para evaluar cada una de las posibles soluciones alternativas existentes. Es por ello por lo que además de la citada descripción, es interesante incluir una **calificación de la prioridad de cada uno de los requisitos**, con el fin de tener presente su importancia relativa respecto al resto.

ANOTACIONS

La descripción de cada requisito incluirá una explicación de éste, la prioridad que se le asigna y una **catalogación** dentro de un conjunto de categorías definidas.

Caso práctico

Definición general de requisitos del sistema

Mediante el estudio del sistema del sitio web actual, los puntos de mejora y problemas detectados, y las entrevistas con los usuarios de éste, se han identificado y catalogado los siguientes requisitos (la prioridad de cada uno de ellos está indicada como un número entre 0 y 100, siendo 100 el prioritario).

Requisitos técnicos:

- (100) Arquitectura: el contenido del sitio web deberá poderse administrar mediante la utilización de cualquier navegador.
- (80) Arquitectura: el contenido del sitio web deberá estar almacenado en un sistema gestor de bases de datos relacionales, sobre el cual se puedan realizar futuras consultas no previstas en la actualidad.
- (80) Seguridad: el contenido del sitio web únicamente podrá ser modificado por aquellas personas autorizadas para ello.
- (80) Seguridad: se podrán realizar copias de seguridad por separado y conjuntamente del contenido del sitio web y de la forma en que éste se mostrará.
- (80) Normativas y/o estándares: el sitio web deberá cumplir con los estándares marcados por el World Wide Web Consortium (HTML, CSS, etc.).
- (80) Normativas y/o estándares: el sitio web deberá cumplir con las normas de accesibilidad marcadas por el World Wide Web Consortium (Web Accessibility Initiative).

Requisitos operativos:

- (100) Operativa: el sitio web deberá ser visualmente atractivo.

ANOTACIONS

- (10) Operativa: el sitio web deberá poder ser consultado, manteniendo sus características visuales, a través de dispositivos diferentes a ordenadores personales que posean conexión a Internet y un navegador web, tales como televisores, PDA (***personal digital assistant***), etc.
- (80) Operativa: el sitio web deberá posibilitar la visualización de cualquier tipo de contenido multimedia (texto, gráficos, vídeos, etc.).
- (90) Operativa: el sitio web deberá tener una estructura clara, ordenando el contenido de éste en secciones y subsecciones que abarquen cualquier aspecto de los que se quieran comunicar.
- (100) Operativa: el sitio web deberá permitir la contratación de paquetes de horas a través del pago de éstas con tarjeta de crédito, generándose la correspondiente factura, pedido, etc.
- (100) Operativa: la gestión del contenido del sitio web deberá poder ser realizada por una persona no técnica, es decir, que no tenga conocimientos de HTML, JavaScript, etc., de forma fácil e intuitiva.
- (40) Administración: la administración del sitio web (consulta de estadísticas, mantenimiento de cachés, etc.) deberá poder realizarse a través de un navegador web.

Requisitos legales:

- (60) La licencia de uso del software de gestión de contenidos debe ser lo menos restrictiva posible.
- (60) La licencia de uso del sistema operativo del servidor web debe ser lo menos restrictiva posible.

Requisitos económicos:

- (80) En el caso de ser necesario un gasto en concepto de licencia de uso del software de gestión de contenidos, éste deberá ser lo más pequeño posible.
- (80) El gasto correspondiente al sistema operativo del servidor web debe ser lo más pequeño posible.

Nota

En el resto del curso nos centraremos en el estudio del sistema del sitio web como ejemplo particular del caso práctico planteado hasta el momento. Para el sistema de aplicaciones de gestión sería necesario seguir el mismo proceso que se describirá de este punto en adelante.

ANOTACIONS

1.4. Estudio de las alternativas de solución

Una vez expresados los requisitos que deberá cumplir el proyecto sobre el que se está realizando el estudio de viabilidad, se pasará a **proponer varias soluciones alternativas** que cumplan con éstos. En esta fase se tendrá en consideración, asimismo, toda la información recogida hasta el momento: descripción general, alcance, situación actual, etc.

Para cada alternativa se deberá **especificar en qué consiste, tanto a nivel funcional como técnico** (estudiando en qué medida se cubren los requisitos descritos previamente), si está basada o no en algún producto ya existente en el mercado (en tal caso, se estudiará éste, describiendo posibles costes de licencias, evolución prevista, estándares que cumple, etc.), y si supone o no la necesidad de realizar algún desarrollo a medida (en tal caso, se deberá describir éste de tal modo que quede claro cuál es su alcance).

Caso práctico

Soluciones alternativas del sistema

Como ejemplo de alternativas al sistema web se proponen tres soluciones posibles, las cuales tienen las siguientes características:

- Microsoft Windows + aplicación propietaria: en este caso, el sistema operativo del servidor web será Microsoft Windows 2000, y la gestión de contenidos se hará mediante la adquisición de un software específico dedicado a tal efecto, por ejemplo HardCore Web Content Management (http://www.hardcoreinternet.co.uk/). Se ha comprobado que el software de gestión de contenidos elegido cumple con los requisitos funcionales y técnicos definidos al respecto. En lo referente a los requisitos legales y económicos de la solución propuesta, el sistema operativo no cumple con lo expresado, y el software de gestión de contenidos sólo lo hace en parte.

ANOTACIONS

- GNU/Linux + aplicación propietaria: en este caso el sistema operativo del servidor web sería GNU/Linux (en cualquiera de sus distribuciones), sin embargo el software de gestión de contenidos sería propietario, por ejemplo HardCore Web Content Management. Se ha comprobado que el software de gestión de contenidos elegido cumple con los requisitos funcionales y técnicos definidos al respecto. En lo referente a los requisitos legales y económicos, el sistema operativo cumple con ellos (ya que no es necesario pagar ninguna licencia de uso, y además se nos permite hasta estudiar y modificar el código fuente de éste), pero no así el software de gestión de contenidos (ya que éste es propietario y no los cumple en su totalidad).

- GNU/Linux + aplicación libre: en este caso tanto el sistema operativo del servidor de web como el software de gestión de contenidos serían libres (por ejemplo ezPublish, http://www.ez.no/, Drupal, http://www.drupal.org). Se ha comprobado que el software de gestión de contenidos elegido cumple con los requisitos funcionales y técnicos definidos al respecto, siendo su configuración inicial ligeramente más complicada que la del software propietario tomado como referencia en los anteriores puntos. En cuanto a los requisitos legales y económicos, se cubren perfectamente, ya que las licencias son muy flexibles y los costes de adquisición son nulos.

En los tres casos será necesaria la realización de un módulo de software que recoja la información generada en el proceso de compra de paquetes de horas vía web (datos de cliente, datos de producto adquirido, tarjeta de crédito, etc.) y la incorpore al programa de gestión administrativa de la empresa.

Sobre la base de la información obtenida por diferentes canales (informes, foros de discusión, experiencia del personal de la empresa, etc.), se ha considerado que el coste de instalación y mantenimiento de los componentes es igual en los tres casos.

ANOTACIONS

1.5. Valoración de las alternativas

Una vez se han estudiado las soluciones alternativas dentro del proyecto sobre el que se está haciendo el estudio de viabilidad, se debe pasar a **valorarlas considerando su viabilidad económica** (análisis costes/beneficios) **y riesgos que comportan**.

Para cada una de las posibles soluciones, se deberá **estudiar su viabilidad económica**, esto es, confeccionar un análisis costes/beneficios que deje patente el gasto que será necesario realizar y lo que se espera obtener a cambio (tanto de forma tangible como intangible).

Caso práctico

Análisis costes/beneficios del sistema

Los costes de adquisición imputados a cada una de las soluciones son:

- Microsoft Windows + aplicación propietaria = = 550 € + 2.000 € = 2.550 €.
- GNU/Linux + aplicación propietaria = = 0 € + 2.000 € = 2.000 €.
- GNU/Linux + aplicación libre = = 0 € + 0 € = 0 €.

Dado que consideramos que los costes de instalación y mantenimiento son los mismos para los tres casos, éstos no tendrán efecto en la comparación que estamos realizando (en un caso real podrían tener mucha importancia, ya que no únicamente se compararían las diferentes soluciones, sino que se estudiaría la viabilidad económica de elegir cualquiera de ellas).

En el caso de la tercera opción, GNU/Linux + aplicación libre, debemos tener en cuenta el coste añadido asociado a la complejidad de su configuración inicial, dado que suponemos que ésta no es tan fácil de realizar como la de la aplicación propietaria. Suponemos que el hecho de utilizar la aplicación libre frente a la aplicación propietaria requerirá 10 horas extra de dedicación (a un precio medio de 50 €/hora) acerca de la utilización de la primera. Es necesario sumar por tanto este coste al de adquisición: 0 € + 500 € = 500 €. Los beneficios de cada una de las soluciones son los descritos en apartados anteriores (requisitos, descripción, etc.).

Nota

Los precios indicados son ficticios, utilizados únicamente a modo de ejemplo.

Además de estudiar la viabilidad económica de las diferentes soluciones, deberemos tener en cuenta los **riesgos asociados** a cada una de ellas. Para cada una de las alternativas existentes, describiremos qué incertidumbres, problemas potenciales, etc. existen.

Caso práctico

Riesgos en las alternativas del sistema

Los riesgos asociados a cada una de las soluciones alternativas son los siguientes:

- *Microsoft Windows + aplicación propietaria:*
  - Sistema operativo: cambio en la estrategia de negocio del fabricante, desapareciendo el soporte dado hasta el momento y haciéndose necesaria una actualización.
  - Sistema operativo: fallos de seguridad detectados pero no subsanados por el fabricante en un periodo de tiempo razonable.
  - Aplicación propietaria: desaparición del fabricante del producto, o cambio de estrategia de negocio (dado que la licencia, aunque propietaria, permite disponer del código fuente de la aplicación, este hecho supondría que cualquier error o problema debería ser subsanado por el equipo de Soluciones Abiertas).

- *GNU/Linux + aplicación propietaria*
  - Sistema operativo: se podría dar la falta de soporte en determinados casos, ya que no existe un solo fabricante que centralice el desarrollo del sistema operativo.
  - Aplicación propietaria: desaparición del fabricante del producto, o cambio de estrategia de negocio (dado que la licencia, aunque propietaria, permite disponer del código fuente de la aplicación, este hecho supondría que cualquier error o problema debería ser subsanado por el equipo de Soluciones Abiertas).

- *GNU/Linux + aplicación libre*
  - Sistema operativo: se podría dar la falta de soporte en determinados casos, ya que no existe un solo fabricante que centralice el desarrollo del sistema operativo.
  - Aplicación libre: desaparición del equipo principal de desarrolladores que mantienen la aplicación.

ANOTACIONS

Llegados a este punto, se deberá realizar una **propuesta de enfoque con el fin de paliar en la medida de lo posible los riesgos** antes descritos. De este modo intentaremos reflejar si estos riesgos son salvables, haciéndose relevante su importancia relativa.

Caso práctico

Paliación de riesgos en las alternativas del sistema

Los posibles enfoques con el fin de paliar los riesgos asociados a cada una de las soluciones alternativas son los siguientes:

- *Microsoft Windows + aplicación propietaria:*
  - Sistema operativo: firma de contrato de soporte del sistema operativo con el fabricante de éste por un periodo de tiempo igual al que estimemos será la vida del sistema web tal y como lo estamos estudiando. Esta solución debe ser aceptada por el fabricante para poder llevarse a cabo.
  - Sistema operativo: firma de contrato de soporte con indemnizaciones en caso de producirse fallos en la seguridad del sistema debido a problemas en el sistema operativo. Esta solución debe ser aceptada por el fabricante para poder llevarse a cabo.
  - Aplicación propietaria: firma de contrato en el cual el fabricante se comprometa a licenciar de forma libre el código fuente de su aplicación en caso de que cese su actividad.

- *GNU/Linux + aplicación propietaria*
  - Sistema operativo: puede contratarse el soporte de una empresa externa que se comprometa a centralizar y resolver los posibles problemas que puedan surgir.
  - Aplicación propietaria: firma de contrato en el cual el fabricante se comprometa a licenciar de forma libre el código fuente de su aplicación en caso de que cese su actividad.

- *GNU/Linux + aplicación libre*
  - Sistema operativo: puede contratarse el soporte de una empresa externa que se comprometa a centralizar y resolver los posibles problemas que puedan surgir.

ANOTACIONS

– Aplicación libre: se debe valorar la estabilidad y alcance de la comunidad formada en torno a la aplicación, ya que en caso de que el equipo principal de desarrolladores desaparezca, la continuidad de ésta dependerá del número de personas que la utilizan y desarrollan esporádicamente en todo el mundo.

1.6. Selección de la solución

Para acabar con el estudio de viabilidad, se **elegirá una solución de entre las diferentes alternativas estudiadas**.

La decisión sobre cuál es la mejor solución (o si ninguna lo es) se tomará teniendo en cuenta la información acumulada hasta el momento:

- Descripción general y alcance del proyecto.
- Situación actual del sistema.
- Requisitos que deberá cumplir la solución adoptada.
- Descripción de las soluciones alternativas consideradas.
- Análisis de costes/beneficios de las diferentes soluciones y riesgos asociados a cada una de ellas.

Caso práctico

Selección de la solución adoptada en el sistema

Dada la descripción general del sistema y la situación actual de éste, se han considerado los siguientes factores con el fin de realizar la elección de la solución:

- Requisitos planteados y descripción de cada una de las soluciones: todas las soluciones cubren en mayor o menor medida los requisitos básicos a nivel funcional y técnico. En cuanto a los aspectos económicos y legales, la solución GNU/Linux + aplicación libre es la ganadora.

ANOTACIONS

- Análisis costes/beneficios: este análisis ha dado como resultado tres costes, entre los cuales la solución GNU/Linux + aplicación libre es la más barata. Dado que los beneficios aportados por cada solución son parecidos en términos generales (desde luego podrían discutirse ciertos detalles en los que sí que hay diferencias significativas, pero que no decantan definitivamente la balanza por una u otra solución), se ha optado por valorar como más positiva la solución GNU/Linux + aplicación libre.

- Riesgos: se han detectado posibles problemas de diferentes tipos en cada una de las soluciones, siendo los de más fácil solución los relacionados con el sistema operativo GNU/Linux y la aplicación de gestión de contenidos libre (precisamente por su carácter marcadamente abierto en comparación con el resto).

Se decide por tanto que la solución en GNU/Linux + aplicación libre de copias de seguridad es la más adecuada de entre todas las consideradas.

ANOTACIONS

2. Análisis del sistema

El objetivo de la realización del **análisis del sistema** es el de, dada la solución escogida de entre las descritas en el estudio de viabilidad, llevar a cabo una **especificación detallada de ésta** (orientada a facilitar el diseño del sistema, fase cubierta en el siguiente capítulo).

Los siguientes apartados describen con más detalle cada una de las tareas que hay que llevar a cabo para realizar el análisis del sistema.

2.1. Definición del sistema

En esta fase del análisis se deberá **describir el sistema**, establecer **cómo se comunicará con otros** en caso de ser necesario y **qué usuarios serán representativos** en el uso del mismo.

Como el lector recordará, ya en la fase de estudio de viabilidad se procedió a describir el sistema genéricamente, así como a definir cómo afectaba éste al resto de sistemas ya existentes (o proyectos que se pensaba llevar a cabo). El trabajo realizado en dicha fase servirá como base de las tareas realizadas en el presente análisis.

Utilizando como punto de partida la descripción de los requisitos hecha en el estudio de viabilidad, se **determinarán los requisitos exactos del sistema**. Asimismo, se estudiará **cómo se comunica el sistema** con el resto de sistemas existentes (ya sea recibiendo o enviado información a éstos).

Caso práctico

Requisitos exactos del sistema web

El sistema web de Soluciones Abiertas deberá cumplir los siguientes requisitos:

- La información deberá ser presentada de forma atractiva, en consonancia con la imagen corporativa de la empresa (colores, fuentes, logotipo, etc.).

ANOTACIONS

- El contenido del sitio web deberá ser administrado mediante una herramienta que permita crearlo, actualizarlo y borrarlo de forma fácil e intuitiva.
- Deberán poder almacenarse diferentes versiones del contenido del sitio web, siendo una de éstas la marcada como "publicada" (será por tanto la que se muestre a los visitantes del sitio web).
- Deberá poderse establecer un flujo de trabajo que marque la evolución del contenido creado. Así, por ejemplo, una persona podría crear cierto nuevo contenido, pero no publicarlo. Otra persona podría únicamente editar aquel contenido ya existente, como el mencionado anteriormente. Finalmente, una tercera persona podría aprobarlo para su publicación y efectuar la publicación de éste.
- La edición del contenido del sitio web deberá ser lo más fácil posible, evitando que las personas que desempeñen esta tarea deban conocer HTML, CSS, etc. Para ello, deberá existir un editor WYSIWYG (***what you see is what you get***) que permita crear texto, subrayarlo, añadir negritas, crear tablas, insertar imágenes, etc.
- La edición del contenido del sitio web deberá poder realizarse mediante las versiones más recientes de cualquiera de los navegadores más populares, especialmente Microsoft Internet Explorer (versión 6.0 o posterior) y Mozilla Firefox (versión 1.0 o posterior).
- Únicamente aquellas personas autorizadas para ello podrán acceder al sistema de gestión de contenidos. Esta autorización consistirá en que cada una de dichas personas poseerá un nombre de usuario y una contraseña válidos en el sistema.
- Deberán poderse definir perfiles de usuarios del sistema de gestión de contenidos, que tengan relación con las tareas que estén autorizados a realizar éstos (relacionado con el requisito de existencia de flujos de trabajo).
- El sitio web público deberá cumplir el estándar HTML 4.01.

ANOTACIONS

- El sitio web público deberá hacer un uso intensivo de hojas de estilo, siguiendo el estándar CSS (***cascading style sheets***) 2.0 (teniendo en cuenta que algunos navegadores no lo soportan en su totalidad).
- El sitio web público deberá seguir las recomendaciones marcadas en las Web Content Accessibility Guidelines versión 1.0 del World Wide Web Consortium.
- El contenido del sitio web deberá poder ser ordenado de forma jerárquica, es decir, en apartados y subapartados. No existirá ningún límite técnico en cuanto a la posible adición de nuevos apartados.
- Los administradores del sitio web deberán poder consultar las estadísticas de acceso a éste, que recogerán valores como el número de páginas vistas diariamente, mensualmente y anualmente, número de visitas, páginas más consultadas, navegadores clientes más habituales, etc.
- El sitio web deberá permitir la compra de paquetes de horas de prestación de servicios, efectuando el pago de éstos mediante tarjeta de débito o crédito.
- Las comunicaciones existentes entre el sistema web y cualquier otro sistema, especialmente el de las aplicaciones de gestión de ésta, se realizarán mediante llamadas a servicios web. Así, por ejemplo, el sistema web comunicará que se ha realizado una compra de paquete de horas a la aplicación de gestión a través de un servicio web ofrecido por esta última.
- La licencia de uso del software de gestión de contenidos deberá ser lo menos restrictiva posible, en concreto deberá ser de código abierto o libre.
- Las licencias de uso de las aplicaciones utilizadas por el software de gestión de contenidos (sistema gestor de base de datos, intérprete de ***scripts***, etc.) deberán ser lo menos restrictivas posibles, en concreto deberán ser de código abierto o libre.
- La licencia de uso del sistema operativo del servidor web será la correspondiente a GNU/Linux, es decir, GNU General Public License.

ANOTACIONS

En la definición del sistema también será necesario definir el **entorno tecnológico** del proyecto, respecto del cual ya se incluyó información en el estudio de viabilidad.

Caso práctico

Entorno tecnológico del sistema

El entorno tecnológico del sistema web será el siguiente:

- Sistema operativo: GNU/Linux (distribución por determinar).
- Sistema de gestión de contenidos: deberá poder ejecutarse en el sistema operativo GNU/Linux y estar hecho en un lenguaje que el equipo de personas de Soluciones Abiertas conozca (p. ej., PHP, Perl, Python, Java).
- Desarrollos a medida: en el caso de ser necesario realizar algún tipo de desarrollo, se llevará a cabo utilizando las tecnologías habituales en el proyecto que se esté modificando (p. ej., PHP en caso de que el desarrollo esté relacionado con el gestor de contenidos, C++ en caso de que el desarrollo esté relacionado con la aplicación de gestión de la empresa).

Para completar la descripción del sistema, será necesario hacer referencia al conjunto de **estándares y normas** que hay que considerar en la implementación de éste.

Caso práctico

Normas que cabe seguir en el sistema web

Las normas y estándares que hay que seguir en la implementación del sistema web serán las siguientes:

- En cuanto al sistema operativo, se seguirá el proceso documentado como "Instalación de servidores GNU/Linux" de Soluciones Abiertas, S. A.
- El software de gestión de contenidos deberá permitir el uso de los estándares web ***de facto*** y ***de jure*** más habituales (HTML, CSS, JavaScript, etc.). Para más información, podéis consultar el apartado "Requisitos exactos del sistema".

ANOTACIONS

- Los posibles desarrollos a medida seguirán las normas internas de Soluciones Abiertas, S. A., es decir, las que se recogen en el documento "Normas de desarrollo de Soluciones Abiertas, S. A.". En este documento quedan recogidas las normas que hay que seguir en el desarrollo de cualquier proyecto, tales como utilización de diagramas UML, formato de documentación del código, etc.

Una vez descrito el sistema, se procederá a **identificar a aquellos usuarios que intervendrán en la definición de requisitos de éste y en su aceptación definitiva**. Es especialmente importante contar con la colaboración de los usuarios a lo largo de todo el proceso de desarrollo del sistema.

Caso práctico

Identificación de usuarios del sistema web

El personal involucrado en la definición de requisitos y aceptación de la solución final del sistema web de Soluciones Abiertas es:

- El personal del departamento de marketing encargado de la creación y actualización del contenido que se muestra en el sitio web de la empresa.
- El personal del departamento de administración encargado de la facturación de la empresa, gestión de nuevos pedidos, etc.
- Los administradores del sistema web, encargados de que éste funcione en todo momento, así como el personal técnico encargado del mantenimiento correctivo y evolutivo del sistema web.

2.2. Establecimiento de requisitos

El objetivo de esta fase será **completar los requisitos definidos anteriormente**, contando con la información suministrada por los usuarios. En la medida de lo necesario, **se dividirá el sistema en subsistemas** que permitan su estudio por separado, con el fin de facilitar el análisis de éstos.

ANOTACIONS

La comparación de la descripción de cada uno de los requisitos expresados en esta fase del proyecto con el diseño creado con posterioridad nos permitirá verificar la corrección de este último.

El primer paso en el proceso de establecimiento de requisitos será el de **obtener los requisitos a partir de la información suministrada por los usuarios**. Los requisitos recogidos en las reuniones mantenidas con los usuarios elegidos en la fase anterior serán básicamente de los siguientes tipos:

- **Funcionales** (p. ej., mediante el sistema de gestión de contenidos se deberá poder modificar cualquier página del web corporativo, sea cual sea su contenido).
- **Rendimiento** (p. ej., la carga de cualquier página del sitio web corporativo, en condiciones normales de utilización de la red, no puede tardar más de 8 segundos).
- **Seguridad** (p. ej., sólo podrán modificar el contenido del sitio web aquellas personas que estén autorizadas para ello).
- **Implantación** (p. ej., el servidor web estará alojado en un lugar físicamente seguro).
- **Disponibilidad** (p. ej., el sistema web deberá ser monitorizado cada 30 minutos con el fin de comprobar su correcto funcionamiento).

Caso práctico

Definición del requisito "compra de paquetes de horas"

Con relación a la compra –mediante el sitio web de la empresa– de paquetes de horas de prestación de servicios, efectuando el pago de éstos mediante tarjeta de débito o crédito, se han determinado los siguientes requisitos:

- Junto con el personal del Departamento de Administración se ha determinado que el pago de los paquetes de horas se realizará mediante la utilización del terminal punto de venta virtual ofrecido por la entidad bancaria con la que Soluciones Abiertas trabaja habitualmente. De esta forma se obtendrán mejores condiciones económicas por transacción y se facilitará la operativa habitual.

ANOTACIONS

- De acuerdo con el personal del Departamento de Marketing se ha determinado qué datos serán los que los clientes deberán suministrar para la correcta realización de la compra de paquetes de horas: datos personales (nombre y apellidos, teléfono, dirección de correo electrónico, departamento dentro de la empresa), datos de la empresa (razón social, número de identificación fiscal, dirección postal, teléfono, fax, dirección de correo electrónico). Junto con los mencionados datos, se permitirá también introducir cualquier tipo de observación sobre el pedido que se esté realizando.
- A petición del personal del Departamento de Marketing, y teniendo en cuenta la legislación vigente, se ha decidido que la contratación de paquetes de horas mediante el sitio web debe ser lo más segura posible, evitando que los datos del cliente (domicilio, número de tarjeta, etc.) puedan ser descubiertos por terceros (utilización de Secure Sockets Layer tanto en el sitio web corporativo, como en el terminal punto de venta virtual utilizado). El objetivo es vencer las posibles reticencias que los clientes tengan en cuanto a la seguridad a la hora de facilitar sus datos y realizar el pago mediante Internet.
- El personal del departamento de marketing ha determinado que la contratación de un paquete de horas deberá tener como resultado el envío por correo electrónico de una factura a la dirección indicada por el cliente. Los datos de esta factura (especialmente el número de ésta) serán obtenidos mediante la comunicación con la aplicación de gestión de la empresa (esta decisión ha sido contrastada con la opinión del departamento de administración).
- La contratación de un paquete de horas tendrá como resultado la creación de un pedido en la aplicación de gestión de la empresa. Este pedido servirá para identificar el servicio contratado de forma unívoca en cualquier momento, cuáles son sus características y cuál es su estado (tipo de servicio contratado, horas consumidas hasta el momento, etc.). Esta información es la que habitualmente maneja el Departamento de Administración, y que también sirve como referencia al de producción/servicios.

ANOTACIONS

Una vez descritos cada uno de los requisitos, se procederá a la especificación de los **casos de uso** de cada uno de ellos. Los casos de uso, además de la descripción en sí del problema, incluirán cómo interactuarán los usuarios con el sistema, qué interfaces utilizarán y cómo se tratarán las condiciones de fallo.

Caso práctico

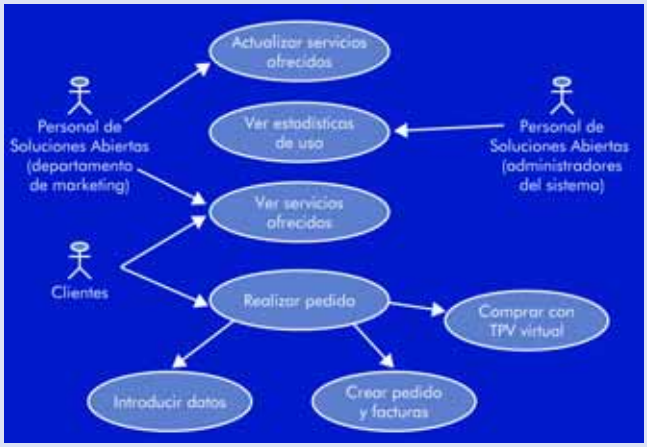
Caso de uso "compra de paquetes de horas"

Los usuarios del departamento de marketing actualizarán el contenido del sitio web corporativo para que en éste figure la información más actualizada sobre los servicios ofrecidos por Soluciones Abiertas, los honorarios sobre éstos, posibles promociones, etc. Los clientes o futuros clientes podrán acceder a esta información, y en caso de estar interesados realizar un pedido (la compra de un paquete de horas de prestación de servicios).

La realización de un pedido comportará llevar a cabo las siguientes tareas: introducción de los datos personales y de la empresa que realiza la contratación, realización de la compra a través del terminal punto de venta virtual del banco, introduciendo el correspondiente número de tarjeta de crédito, y la creación de un pedido y una factura asociadas a la contratación realizada. Estos dos últimos pasos se llevarán a cabo teniendo en cuenta que ambos deberán coordinarse con sistemas externos.

La navegación a través de la información del sitio web generará una serie de mensajes que se procesarán a posteriori para la realización de las estadísticas de acceso a éste, que únicamente los administradores podrán consultar.

Figura 2-1. Diagrama de caso de uso



ANOTACIONS

A la vez que se describen los requisitos y sus casos de uso correspondientes, se analizarán éstos con el fin de **detectar posibles inconsistencias** (dado que pueden intervenir diferentes usuarios con diferentes necesidades), duplicidades, etc., y las **posibles asociaciones** entre éstos.

Caso práctico

Asociaciones de un caso de uso

El caso de uso referente a la "compra de paquetes de horas" por parte de los clientes está directamente relacionado con los siguientes casos de uso:

- El proceso y la información manejada al recoger los datos de un cliente deben ser comunes o muy parecidos al de la sección "Póngase en contacto con nosotros" del sitio web, descrito en el caso de uso "Contacto".
- El método de actualización de la información acerca de los servicios ofrecidos por Soluciones Abiertas debe ser exactamente el mismo que el descrito en el caso de uso más general "Actualización de contenido".
- Las estadísticas almacenadas sobre el acceso a la compra de paquetes de horas deben ser las mismas que las referidas en el caso de uso "Estadísticas del sitio web".

2.3. Definición de interfaces de usuario

En esta fase del análisis **se especifican cómo serán las diferentes interfaces que habrá entre el sistema que estamos describiendo y los usuarios** de éste. Esta especificación se hará teniendo en cuenta los diferentes perfiles de usuarios, flexibilidad necesaria, tipos de acciones que hay que llevar a cabo, etc.

El primer paso en la definición de las interfaces de usuario será el de **definir los perfiles de usuarios que utilizarán el sistema**. De este modo, se podrá describir posteriormente a qué tipos de interfaces accederá cada uno de ellos.

ANOTACIONS

Caso práctico

Perfiles de usuarios

La aplicación de gestión de contenidos, que permitirá mantener la información mostrada en el sitio web de Soluciones Abiertas, será utilizada principalmente por los usuarios del Departamento de Marketing, que en general tendrán las siguientes características:

- Usuarios con un perfil no técnico.
- Usuarios acostumbrados a la utilización de programas de edición de documentos y hojas de cálculo.
- Usuarios acostumbrados a la utilización de programas para la realización de presentaciones.
- Usuarios no acostumbrados a la edición de ficheros de texto, especialmente aquellos que contienen marcas, instrucciones, etc.

A continuación, se deberán **especificar los principios generales de la interfaz de usuario**, por ejemplo, si se utilizarán interfaces de texto o gráficas, cómo se mostrarán los mensajes de error, cómo se obtendrá ayuda, etc.

Caso práctico

Principios generales de la interfaz de usuario

La aplicación de gestión de contenidos tendrá las siguientes características:

- El acceso a la aplicación y su uso se realizará a través de un navegador web.
- La edición de cualquier tipo de contenido se realizará a través de formularios web en los que se mostrará la información ya existente para cambiarla o se introducirá la nueva.
- Existirá un tipo de formulario especial, que servirá para introducir el grueso de la información del sitio web, que dispondrá de controles especiales que permitan la edición WYSIWYG.
- Los mensajes de error serán mostrados en general por pantalla, en la medida de lo posible acompañados de un número que los identifique de forma unívoca.

ANOTACIONS

- Aquellos mensajes de error que no se puedan mostrar por pantalla de forma detallada a los usuarios de la aplicación (p. ej., si se produce un error cuando un cliente está realizando un pago a través del módulo de comercio electrónico), serán enviados por correo electrónico al administrador del sitio web.
- La ayuda a nivel funcional de la aplicación de gestión de contenidos estará integrada dentro de ésta, formando parte de las páginas que permitan la edición de la información o presentándose como enlaces dentro de éstas.

Una vez identificadas las características generales de la interfaz de usuario, se pasará a **especificar ésta para cada uno de los casos de uso** definidos en el apartado anterior.

Ejemplo

Interfaz de usuario

El siguiente diagrama recoge cuál deberá ser la información ofrecida por la interfaz de usuario de la aplicación de gestión de contenidos, y cuál será su disposición (en la medida de lo posible se intentará seguir este esquema, aunque está sujeto a posteriores decisiones de diseño).

Figura 2-2. Esquema de interfaz de usuario

ANOTACIONS

2.4. Especificación del plan de pruebas

Para acabar con la fase de análisis, se procederá a realizar la especificación del plan de pruebas, que nos **servirá para establecer si el sistema cumple con los requisitos establecidos por los usuarios**.

Se podrán realizar **pruebas** del sistema a varios **niveles**:

- Pruebas unitarias, con el fin de testar por separado cada uno de los componentes que forman el sistema (p. ej., prueba de conexión al terminal punto de venta virtual del banco elegido).
- Pruebas de integración, con el fin de testar el funcionamiento de los componentes actuando de manera coordinada, es decir, testar cada uno de los subsistemas que forman el sistema (p. ej., prueba de creación de un nuevo pedido en la aplicación de gestión de la empresa).
- Pruebas de sistema, con el fin de testar el funcionamiento de los subsistemas actuando de manera coordinada (p. ej., prueba de acceso intensivo a los contenidos del sitio web de Soluciones Abiertas).
- Pruebas de implantación, con el fin de testar el funcionamiento del sistema en el entorno de operación de éste (p. ej., prueba de volcado y recuperación de la base de datos que alberga el contenido del sitio web, una vez instalada en el/los servidor/es de producción).
- Pruebas de aceptación, con el fin de que los usuarios del sistema validen el correcto funcionamiento de éste (p. ej., realización de una compra completa de un paquete de horas de servicio por parte de un usuario del Departamento de Marketing). Este conjunto de pruebas es crítico, ya que será el que permitirá validar el sistema completo. Además del correcto funcionamiento del sistema, se deberán tener en cuenta parámetros como la seguridad, rendimiento, disponibilidad, etc.

Para cada una de las pruebas que hay que realizar, se deberá definir el **alcance** de éstas (p. ej., usuarios implicados en las pruebas, productos de las pruebas, criterios de aceptación de las pruebas, etc.), y los **requisitos en el entorno de pruebas** (hardware necesario, librerías disponibles, configuración de accesos, etc.).

ANOTACIONS

Caso práctico

Prueba de integración

La prueba de integración del sistema web con la aplicación de gestión de la empresa tendrá las siguientes características:

- Permitirá al personal del departamento de administración comprobar que las compras realizadas a través del sistema web se integran de manera correcta con la aplicación de gestión utilizada habitualmente por ellos.
- Como producto de la prueba se obtendrá un nuevo pedido con los datos introducidos a través del sistema web (cliente que lo ha encargado, número de horas contratadas, etc.). El número de este pedido deberá tener sentido dentro de la aplicación (no estar repetido, ser consecutivo con respecto al último realizado antes de él, etc.).
- Como producto de la prueba también se obtendrá una factura con los datos correspondientes al cliente que ha realizado el pedido. El número de esta factura deberá haberse asignado de forma consecutiva a la anterior a ella en el tiempo.
- La prueba se dará por correcta cuando los usuarios de la aplicación de gestión de la empresa hayan validado el pedido y la factura generados, dedicando especial atención a los números de éstos.

Con el fin de poder realizar la prueba, será necesario:

- Disponer del módulo de comercio electrónico del sistema web completamente acabado, ejecutándose en un servidor alojado en la red sobre la cual se implantará definitivamente.
- Disponer de un acceso al terminal punto de venta de la entidad bancaria escogida, preferiblemente en modo de pruebas (aceptando pagos con tarjetas inexistentes).
- Disponer del módulo de atención de peticiones remotas desde el sistema web a la aplicación de gestión, así como un usuario y contraseña válidos para acceder a éste.

ANOTACIONS

3. Diseño del sistema

El objetivo de la fase de **diseño** de un proyecto web es obtener los **modelos y especificaciones** que lo definen a partir del análisis realizado en la fase anterior. Las actividades que llevemos a cabo en esta fase nos permitirán determinar las especificaciones de desarrollo e integración, así como definir el entorno de pruebas e implantación necesarios para su correcto funcionamiento.

Concretamente, los resultados que deberemos obtener en esta fase serán:

- La definición del modelo arquitectónico del sistema. Mediante la identificación de sus componentes, sus interacciones y la ayuda de herramientas de modelado obtendremos un mapa de los subsistemas y recursos que intervienen en todos los procesos.
- Las especificaciones y estándares que se usarán tanto en esta misma fase como durante el desarrollo del sistema.
- La identificación de cada subsistema, sus requisitos de integración, licencia y funcionalidades cubiertas.
- Los casos de uso aplicados de los subsistemas anteriormente identificados, debidamente revisados para reflejar el modelo y especificaciones definidos.
- Los componentes, clases o interfaces que deberemos construir en la fase de desarrollo.
- Los requisitos necesarios para proceder con éxito a la implantación del sistema.

Como vemos, muchos conceptos y especificaciones van a determinarse en esta fase, y aunque algunas metodologías recientes aconsejan mezclarla con la fase de desarrollo en un ciclo combinado de diseño y construcción iterativo, con el objetivo de obtener resultados pronto o de identificar fallos en el diseño a tiempo, es obvio que las decisiones en cuanto a especificaciones, estándares o subsistemas que tomemos aquí van a facilitar todas las tareas futuras.

ANOTACIONS

Especialmente importante es la identificación de los componentes que hay que usar y sus licencias, ya que éstos pueden determinar parte de la funcionalidad, la necesidad de desarrollos internos de comunicación entre subsistemas o el tipo de licencia con que deberemos distribuir (si procede) el resultado de nuestro proyecto.

3.1. Arquitectura

La definición de la **arquitectura** del sistema es el primer paso para **identificar los componentes** del mismo y da lugar a las siguientes fases de diseño en las que profundizaremos en cada uno de ellos. El objetivo es disponer de un conjunto de documentos y diagramas completos y concisos que sean comprensibles para la dirección y a la vez sirvan de base para profundizar en el diseño del sistema.

Antes de realizar el diseño más detallado del sistema, será necesario definir las normas y estándares de diseño y construcción. También puede darse el caso de que estas normas y estándares hayan sido ya definidas en anteriores proyectos, o sean comunes a todos los que lleva a cabo la organización.

Una vez acordados los estándares de diseño, podremos ya profundizar y determinar los subsistemas, repitiendo el mismo proceso que seguimos con la arquitectura general del sistema, esta vez con mayor granularidad en cada uno de ellos.

3.1.1. Definición de niveles de arquitectura

Existen varias formas de ver o entender la arquitectura de un sistema:

- Arquitectura conceptual: su propósito es dirigir la atención sobre los grandes bloques que forman el sistema, sin entrar en detalles, e identificar las relaciones entre dichos bloques. Es muy útil para comunicar a la dirección o a departamentos no técnicos una visión global del sistema.
- Arquitectura lógica: añade detalle a la anterior e incorpora la definición de las interfaces de comunicaciones entre los componen-

ANOTACIONS

tes, lo que permitirá a los desarrolladores de cada componente trabajar sin dependencias entre ellos.

Como apoyo a los diagramas, podemos utilizar tarjetas CRC (***class responsibility collaborator***), que tienen las siguientes características:

- En la parte superior figura el nombre del componente.
- En la columna izquierda deberemos reflejar todo lo que el componente sabe o hace acerca de él mismo. Allí se incluirá todo aquello que creemos que es de su responsabilidad y la información que deberá mantener.
- En la parte derecha figurarán los componentes con que se relaciona para poder llevar a cabo las responsabilidades de la parte izquierda.

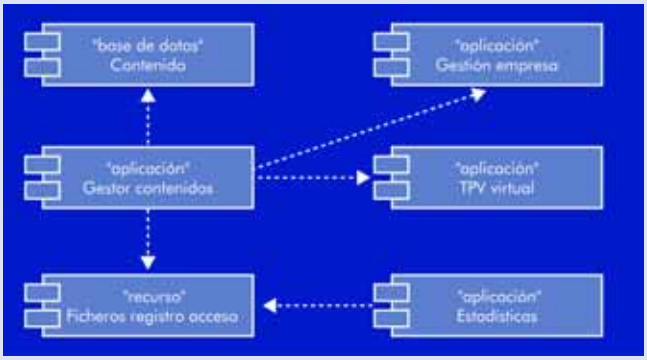
Estas tarjetas son muy utilizadas en las metodologías eXtreme Programming y Agile, y permiten "crear" el diagrama de componentes del sistema dinámicamente encima de una mesa simplemente situando las tarjetas próximas o lejanas entre sí según su grado de comunicación, para consensuar así una visión general de la arquitectura lógica del sistema durante una reunión.

Caso práctico

Definición de la arquitectura

Para expresar la arquitectura del proyecto del sistema web de Soluciones Abiertas, usamos la notación UML en los diagramas y tarjetas CRC (Clase-Responsabilidad-Colaborador).

Figura 3-1. Diagrama UML de componentes

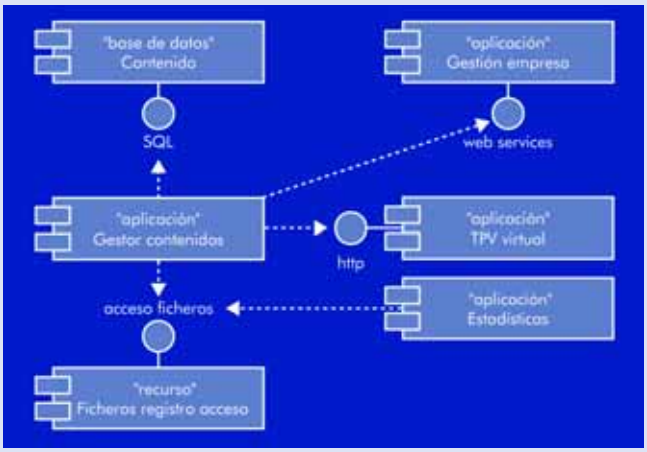


ANOTACIONS

En el diagrama anterior vemos los componentes del sistema web y los conectores que los unen. Estos conectores indican que algún tipo de comunicación se produce entre ellos. Los diferentes componentes en este diagrama están identificados mediante los estereotipos que los acompañan (<<basededatos>>, <<aplicación>>, etc.).

Una vez consensuada esta visión general del sistema, pasamos a profundizar en las interfaces de los componentes para obtener la arquitectura lógica del sistema. Para ello, extendemos el diagrama de componentes anterior detallando los procesos de comunicación.

Figura 3-2. Diagrama UML de componentes con interfaces



Como apoyo a la generación del diagrama anterior, utilizamos tarjetas CRC.

Tabla 3-1. Ejemplo de tarjeta CRC

Gestión de contenidos	
Permite administrar el contenido del sitio web. Muestra el contenido público del sitio web. Incorpora comercio electrónico al sitio web. Permite sincronizar los pedidos hechos mediante el sitio web con la gestión habitual de la empresa. Mantiene el registro de accesos realizados al sitio web.	Base de datos de contenidos. Aplicación de gestión de la empresa. Aplicación de terminal punto de venta virtual. Ficheros registro de accesos.

ANOTACIONS

3.1.2. Especificación de estándares, normas de diseño y construcción

Es importante acordar con las personas encargadas del diseño del sistema y con los equipos que procederán a su construcción unas normas que habrá que seguir en la notación de diagramas y documentos. Estas normas pueden venir dadas por estándares o por recomendaciones, o bien ser de uso y creación interna. Obviamente, siempre es recomendable apoyarse en estándares, ya que va a facilitar la comunicación, consistencia, reusabilidad y comprensión por parte de entidades externas o recién incorporadas al equipo.

Deberemos definir:

- Formato y plantilla de los documentos de diseño.
- Notación a usar en los diagramas de diseño.
- Recomendaciones en cuanto al estilo, idioma y formato de la documentación técnica.

Caso práctico

Definición del conjunto de normas y notaciones

Es conveniente que todos los documentos creados de ahora en adelante, y que van a ser objeto de revisión por parte de equipos diferentes, compartan unas características y mantengan un formato coherente. Para ello, después de estudiar los estándares y recomendaciones sobre el tema, se llega a las siguientes conclusiones:

- Documentos de diseño: estos documentos se deben poder consultar tanto por el personal técnico implicado, como por personal no técnico que pueda revisarlo o consultarlo. Se acuerda que se trabajen en formato OpenDocument y que la versión más reciente esté simultáneamente en PDF para su consulta. Se creará una plantilla que contenga en la primera página:
  - Título del documento.
  - Responsable del documento.
  - Lista de autores que han intervenido y la fecha de su primera intervención.

ANOTACIONS

  - Lista resumida de cambios introducidos en el documento a medida que se vayan produciendo (cambio, fecha y autor).

- Diagramas de diseño: para los diagramas de diseño se acuerda usar la notación Unified Modeling Language, UML (http://www.omg.org/uml/) en su versión 1.5, definida por el Object Management Group (www.omg.org).

- Documentación técnica: la documentación técnica será posiblemente la que más revisiones sufrirá y contendrá también enlaces a documentaciones de las herramientas usadas, especificaciones de programación (API), etc., por lo que se recomienda usar un formato lo más flexible posible e integrable con las propias herramientas de desarrollo que se usen. Para ello, se decide usar DocBook (www.docbook.org, http://www.oasis-open.org/docbook/), que nos permitirá:

  - Partición de un documento en varios ficheros estructurados, susceptibles de ser revisados independientemente.
  - Fácil inclusión de referencias a otros documentos (enlaces http, figuras, etc.).
  - Fácil generación de varios formatos para su visualización (PDF, HTML) y con la posibilidad de separar el contenido del documento de su formato.
  - Independencia de editor usado, ya que es una implementación de XML y, por lo tanto, modificable en cualquier editor de texto.
  - Incorporar documentación contenida en el código fuente generado en la fase de desarrollo, de forma automática en muchos casos.

Cabe destacar que al tomar las decisiones se ha dado importancia a la implantación del formato o notación en la industria y a la accesibilidad del mismo, es decir, a la disponibilidad de ejemplos y documentación, así como a un amplio conjunto de herramientas que trabajen con ellos.

ANOTACIONS

3.1.3. Identificación de subsistemas

Para reducir la complejidad que supondría diseñar al detalle todo el sistema, éste deberá dividirse en subsistemas para facilitar su comprensión, revisión y reutilización. Este tipo de divisiones se realizará iterativamente hasta reducir significativamente la complejidad de las funciones a realizar por los subsistemas resultado.

Para realizar esta división, podemos tener en cuenta varios aspectos de los componentes:

- Funcionalidad común o relacionada por características de la ejecución.
- Gestión de datos, acceso a datos comunes.
- Integración en una interfaz de usuario común.
- Optimización de líneas de comunicaciones o recursos.

Caso práctico

Identificación y diseño de subsistemas

Realizando una primera división por funcionalidad, identificamos claramente los siguientes subsistemas dentro del sistema web:

- Subsistema gestor de contenidos (permitirá administrar y consultar todo el contenido del sitio web).
- Subsistema aplicación gestión de la empresa (permitirá integrar los pedidos realizados a través del sitio web en la operativa normal de la empresa).
- Subsistema de estadísticas (permitirá consultar la información agregada de acceso al sitio web).

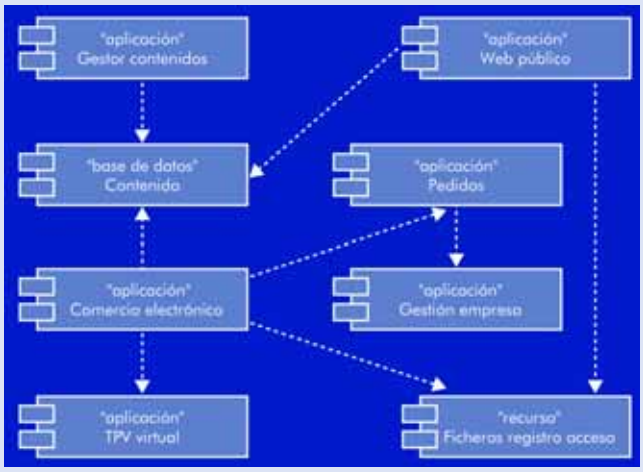
Como ejemplo de subdivisión, el subsistema gestor de contenidos estará compuesto a su vez por los siguientes subsistemas:

- Subsistema de administración de contenido (permitirá modificar el contenido del sitio web).
- Subsistema de web pública (permitirá mostrar el contenido a los usuarios del sistema web).

ANOTACIONS

- Subsistema de comercio electrónico (permitirá realizar pedidos de paquetes de horas de servicio a través del sitio web, utilizando el terminal punto de venta virtual seleccionado).
- Subsistema de pedidos (permitirá integrar la información de los pedidos realizados en la aplicación de gestión de Soluciones Abiertas).

Figura 3-3. Diagrama UML de componentes, subsistema de gestor de contenidos



3.2. Revisión de casos de uso

Una vez identificados los subsistemas, es el turno de revisar los **casos de uso** hechos en la fase de análisis y **determinar las operaciones** que deberán implementar las interfaces de cada uno de ellos.

Así pues, a partir de los escenarios recogidos en la fase de análisis, determinaremos qué subsistemas están implicados en ellos y diseñaremos su funcionamiento teniendo en cuenta:

- El entorno tecnológico en el que se aplican.
- Las excepciones que se produzcan en cada caso de uso.
- Detalles relacionados con la implementación que ya podamos identificar en esta fase.
- Restricciones o características de la interfaz de usuario.
- Nuevos requisitos que podamos identificar.

ANOTACIONS

Si el sistema que estamos diseñando está centrado en el desarrollo, el estudio de los subsistemas deberá incorporar la definición de las clases u objetos y por lo tanto los diagramas que obtengamos deberán también representar la interacción entre los mismos.

Así pues, durante esta fase estableceremos las características, revisaremos los requisitos y diseñaremos las clases (con sus atributos, operaciones y relaciones) de todo el sistema. De manera natural durante el proceso, obtendremos también el diseño de las pruebas que asegurarán el buen funcionamiento del sistema durante el desarrollo y las condiciones de implantación del mismo.

3.2.1. Revisión de los subsistemas según los casos de uso

Para cada caso de uso deberemos definir:

- Los subsistemas que intervienen en el mismo.
- Cuáles serán los objetos que compongan cada subsistema, y qué mensajes intercambiarán entre ellos.

La definición de los mensajes nos servirá para verificar y detallar las interfaces de cada subsistema teniendo en cuenta todos los casos de uso en que interviene y completar así la definición de subsistemas realizada en fases anteriores.

Caso práctico

Revisión de los subsistemas según los casos de uso

El caso de uso referido a la realización de una compra de un paquete de horas de servicio a través del sitio web de Soluciones Abiertas está relacionado con los siguientes subsistemas:

- Subsistema de web público (se mostrará el contenido referente a los paquetes de horas comercializados, sus características, etc.).
- Subsistema de comercio electrónico (permitirá la realización del pedido de paquetes de horas de servicio, utilizando el terminal punto de venta virtual seleccionado).

ANOTACIONS

- Subsistema de pedidos (permitirá integrar la información del nuevo pedido en la aplicación de gestión de Soluciones Abiertas).

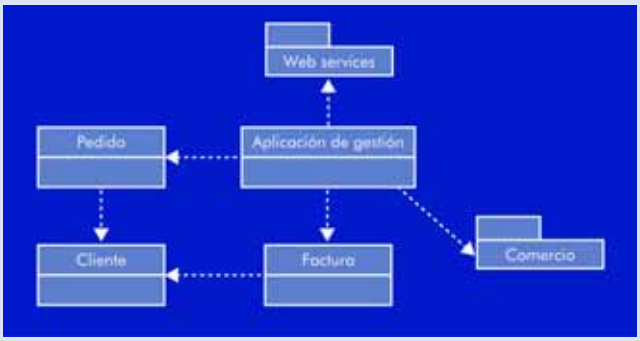
El estudio detallado del caso de uso mencionado, y de los subsistemas que intervienen en él, da como resultado el cambio de algunos de estos últimos, para que soporten características como:

- Carrito de la compra: el subsistema de web público y de comercio electrónico debe permitir realizar un pedido que incluya más de un paquete de horas de servicio.
- Formas de pago: el subsistema de comercio electrónico, junto con el terminal punto de venta virtual, debería dar la posibilidad de realizar el pago mediante tarjeta de débito, de crédito, o domiciliación bancaria.
- Resolución de problemas de pago: el subsistema de comercio electrónico debe contemplar la posibilidad de que el pago del producto contratado no pueda realizarse (debido, por ejemplo, a una falta de fondos en la tarjeta suministrada).

Los cambios propuestos implican cambios en las interfaces existentes en los diferentes subsistemas, así como en los mensajes que éstos intercambian.

Además de la introducción de los cambios propuestos anteriormente, después del estudio detallado de los casos de usos y los subsistemas, se han confeccionado los diagramas de clases relacionados con estos últimos (en los casos en los que se trata de subsistemas que impliquen un desarrollo a medida).

Figura 3-4. Diagrama UML inicial de clases del subsistema de pedidos



ANOTACIONS

3.2.2. Elección de alternativas de componentes y licencias más adecuadas

El estudio detallado de los casos de uso del proyecto web que estemos realizando, junto con la identificación de aquellas partes de éste que necesariamente debamos desarrollar a medida, nos permitirá identificar los subsistemas candidatos a implementarse a través de la utilización de productos software ya existentes en el mercado.

Caso práctico

Concretar alternativas de los componentes

Una vez realizada la revisión de los casos de uso del sistema web de Soluciones Abiertas, hemos identificado que únicamente los subsistemas relacionados con el terminal punto de venta virtual y con la aplicación de la empresa deberán ser desarrollados a medida. El resto de subsistemas podrán ser desarrollados a partir de productos software existentes en el mercado (ya en el estudio de viabilidad del proyecto se determinó que estos productos deberán estar licenciados como software libre).

El estudio de las diferentes alternativas existentes en el mercado, junto con los casos de uso que se desean satisfacer, han dado como resultado la siguiente tabla, que resume los principales componentes a utilizar en la fase de desarrollo.

Tabla 3-2 Principales componentes en la fase de desarrollo

Componente	Paquete	Versión prevista	Licencia
Gestor de contenidos	ezPublish	3.0	GPL
Base de datos	MySQL	4.11	GPL
Sistema operativo	GNU/ Linux	2.6.11	GPL
Servidor web	Apache	2.0.48	Apache Software License
Intérprete de *scripts*	PHP	4.3.11	PHP License

ANOTACIONS

Tanto si se trata de un proyecto web interno, como de un producto que tenemos previsto comercializar, elegir la licencia del mismo o de las partes que vamos a desarrollar antes del inicio del mismo es altamente aconsejable. Más aún cuando estamos integrando componentes software o servicios, ya que su licencia puede condicionar los términos de la nuestra, u obligarnos a incluirla.

La licencia elegida va a tener algún efecto sobre:

- Los ficheros de código fuente de nuestro desarrollo. Deberemos incluirla en todos y cada uno de ellos, y en determinados casos hacer mención de partes de código o librerías que son propiedad de otras organizaciones.
- La documentación y los materiales de formación. Ahí debe reflejarse la licencia escogida desde la primera versión del sistema. Esto es especialmente importante en proyectos de código libre que vayan a ponerse a disposición públicamente, para evitar malentendidos.
- Los componentes elegidos. Si el desarrollo va a ser comercializado bajo una licencia propietaria, debemos asegurarnos de que los componentes que integremos nos lo permiten.
- El cliente que recibe el producto. Debemos informar al cliente respecto a qué derechos tiene sobre el producto y qué garantías le proporcionan sus derechos.
- El mantenimiento del mismo, y el soporte que vamos a ser capaces de proporcionar al producto. En la fase de implantación veremos las estrategias de mantenimiento posibles según el tipo de licencia escogido.

Básicamente, debemos concentrarnos en las incompatibilidades entre los distintos modelos de licencias existentes, dentro de alguno de estos escenarios:

- ¿Vamos a combinar código propietario con el nuestro y distribuirlo bajo licencia comercial? ¿Hemos adquirido estos derechos por parte de los distribuidores del código propietario?
- ¿Vamos a combinar código libre (en alguna de sus variantes, BSD, GPL, LPGL, etc.) con el nuestro y distribuirlo bajo licencia co-

ANOTACIONS

mercial? ¿Nos lo permiten todas las licencias de los distintos componentes?

- ¿Vamos a combinar código libre con el nuestro y distribuirlo bajo licencia libre? ¿Cuál deberá ser la licencia libre resultante? ¿Deseamos imponer alguna restricción a nuestra licencia libre? ¿Nos interesa mantener el ***copyright***? ¿Dar garantía y soporte?

En todo caso, la respuesta a las preguntas que surgen en cada escenario deberán resolverse después de una lectura detenida de las licencias de los componentes y de un análisis de nuestro modelo de negocio.

Caso práctico

Elección de la licencia de desarrollo

Los desarrollos que hay que realizar son para consumo interno, por lo que la licencia elegida no tendrá efectos sobre el modelo de negocio ni sobre su distribución a clientes. Esto no evita que debamos incluir una licencia en el código que desarrollemos, y en este caso, tenemos por ejemplo las siguientes alternativas:

- Licencia propietaria: en nuestro sistema estamos combinando diversas licencias de software libre. Si no vamos a redistribuir nuestro sistema, podemos desarrollarlo bajo licencia propietaria.

- Licencia estilo BSD: esta licencia nos permite mantener el ***copyright*** sobre nuestro desarrollo, y es coherente con las licencias del resto de componentes. No nos obliga a distribuir el código fuente resultante, pero sí que permite al destinatario del software su uso, copia, modificación, redistribución o venta del mismo. También nos permitirá su incorporación futura a un producto comercializable bajo una licencia propietaria.

- Licencia GPL: esta licencia nos permite mantener el ***copyright*** sobre nuestro desarrollo, y es coherente con las licencias del resto de componentes. Nos obliga a distribuir el código fuente resultante e impide su futura comercialización bajo una licencia propietaria.

Una vez analizadas las alternativas, decidimos adoptar la licencia GPL para lo referido a la integración de la herramienta de comercio electrónico y gestor de contenidos con el terminal punto de venta virtual de nuestra entidad bancaria.

ANOTACIONS

3.2.3. Especificaciones de desarrollo y pruebas

Llegados a este punto, estaremos en condiciones de establecer las condiciones y características del entorno de desarrollo en los siguientes términos:

- Entorno tecnológico: hardware, software y comunicaciones.
- Herramientas de desarrollo: IDE, generadores de código, compiladores, etc.
- Herramientas de documentación.
- Restricciones técnicas.
- Requisitos de seguridad del entorno.

También deberemos ser capaces de definir las pruebas necesarias que se deberán realizar para asegurar el funcionamiento del sistema una vez implantado. Éstas deberían definirse como pruebas unitarias, es decir, pruebas con el mínimo nivel posible de dependencia entre ellas para permitir un desarrollo o una integración software por componentes, donde cada equipo pueda trabajar y probar independientemente, dejando para la fase final las pruebas de integración.

La especificación de las pruebas unitarias se divide en pruebas de caja blanca y pruebas de caja negra:

- Caja negra: se considera el componente desde el punto de vista funcional, analizando sus entradas y salidas, y comparando todas sus posibilidades con los resultados esperados.
- Caja blanca: se considera el componente como una estructura con una secuencia lógica de eventos y se comprueba la validez de ésta, el código no usado, comprobaciones contempladas, etc.

Normalmente, se usará una combinación de los dos tipos de pruebas, según el componente o su funcionalidad. Tradicionalmente, se tiende a realizar las pruebas de los componentes una vez desarrollados. Las metodologías más recientes recomiendan invertir este proceso, y justifican la realización de las pruebas previamente a las de los propios componentes software por las siguientes razones:

- Al disponer de la funcionalidad requerida de los componentes, estamos en disposición de diseñarlas.

ANOTACIONS

- Desde el primer momento podremos probar que nuestros componentes cumplen o van cumpliendo las funcionalidades.
- Es mejor crear un componente con el objetivo de pasar las pruebas diseñadas, que crear uno que tenga la funcionalidad requerida. De este modo se evita que los programadores introduzcan efectos colaterales o crean que determinada funcionalidad también será necesaria y la introduzcan.
- Se ha demostrado que si las pruebas están bien diseñadas y son completas, los programadores tardarán igual o menos en crear un componente que cumpla las funcionalidades que uno que simplemente pase los tests diseñados.
- Aunque tradicionalmente se dejan las pruebas para el final del desarrollo, éstas son las que más frecuentemente provocan retrasos en el proyecto. Este enfoque de pruebas unitarias mitiga estos retrasos y optimiza el desarrollo que ahora se enfoca directamente a obtener resultados, entendidos como tests pasados satisfactoriamente.

Para cada una de las pruebas, deberemos definir sus posibles parámetros o información de entrada, y sus posibles resultados o información de salida. Esto nos permitirá más adelante programar cada uno de los tests bajo el marco de trabajo de tests unitarios elegido.

Caso práctico

Definición de las especificaciones de desarrollo y pruebas del subsistema

El subsistema gestor de contenidos debe permitir administrar el contenido del sitio web, así como ofrecer la posibilidad de contratar paquetes de horas de servicio. Algunos de los subsistemas que lo integran, como el que permitirá la edición de contenidos o el de comercio electrónico, serán implementados a través de los productos elegidos de entre los existentes en el mercado. Por el contrario, la utilización del terminal punto de venta virtual de la entidad bancaria escogida, o la integración con la aplicación de gestión de la empresa, exigirán un desarrollo a medida que permita su correcta integración.

ANOTACIONS

Así pues, se deberá desarrollar una interfaz que permita la comunicación con la aplicación de gestión de la empresa, de tal forma que cuando se realice un pedido éste pase a constar de forma automática en dicha aplicación.

Al determinar el lenguaje elegido para el desarrollo, se consideran las siguientes alternativas:

- Perl (http://www.perl.org/): lenguaje de script muy flexible y versátil, dispone de multitud de librerías de apoyo y una comunidad de usuarios extensa. Permite orientación a objetos.
- PHP (http://www.php.net/): lenguaje de script especialmente diseñado para la realización de aplicaciones web. Dispone de multitud de librerías de apoyo y una comunidad de usuarios extensa.
- Java (http://java.sun.com/): lenguaje interpretado y orientado a objetos, con tecnologías complementarias indicadas para llevar a cabo ciertas tareas (p. ej. servlets, JSP). Dispone de multitud de librerías de apoyo y una comunidad de usuarios extensa.

Por su amplia aceptación y disponibilidad de librerías de apoyo, así como su fácil integración con posibles herramientas de gestión de contenidos, Soluciones Abiertas, S. A. decide utilizar el lenguaje PHP.

Al determinar el entorno de desarrollo, se consideran las siguientes alternativas:

- Jedit (http://www.jedit.org/): se trata de un entorno muy potente desarrollado en Java y por tanto multiplataforma, con un conjunto de ***plug-ins*** que proporcionan funcionalidades adicionales como detección de sintaxis y navegación avanzada por el código, integración con sistemas de control de versiones, etc.
- Vim (http://www.vim.org/): se trata de una extensión del editor 'vi' incluido en casi todos los sistemas Unix. Ha sido mejorado para facilitar su uso, y dispone de interfaz gráfica e interfaz de texto.

ANOTACIONS

- Emacs (http://www.gnu.org/software/emacs/): se trata de una herramienta muy potente y compleja. En su origen es un intérprete de Lisp con funcionalidades de edición de textos. Su curva de aprendizaje es muy pronunciada.

Por su facilidad de uso y potencia, con posibilidad de desarrollo de extensiones y de incorporación de las muchas ya existentes, se decide utilizar el entorno de desarrollo Jedit.

El resto de especificaciones de desarrollo provienen de las tomadas anteriormente, ya que el formato de documentación, así como el marco de trabajo de las pruebas unitarias, surgen de forma natural a partir del lenguaje de programación escogido.

- Marco de trabajo de pruebas unitarias: módulo PHP PEAR PHPUnit.
- Documentación del propio desarrollo: formato phpDocumentor (http://www.phpdoc.org/).
- Documentación técnica de las interfaces y su uso: DocBook.

A continuación, deberemos enumerar las pruebas unitarias, extraídas de las funcionalidades e interfaces del subsistema, por ejemplo:

- Conexión y desconexión a la aplicación de gestión de la empresa.
- Obtención de cada uno de los datos que se deberán mostrar al cliente (p. ej. el número de factura suministrado por la aplicación de gestión).
- Inicio y fin de transacciones con la aplicación de gestión para evitar datos duplicados durante el proceso de obtención de éstos.

3.2.4. Requisitos de implantación

Los requisitos de implantación serán los que tenga que cumplir cada componente o subsistema cuando trabaje en el entorno real conjuntamente con el resto de subsistemas. Por entorno no entenderemos

ANOTACIONS

únicamente entorno tecnológico, sino que tendremos en cuenta también a los usuarios del subsistema.

Así pues, la implantación del subsistema tendrá implicaciones para los usuarios y habrá que determinar si sus conocimientos actuales son suficientes para usar el nuevo subsistema, o bien deberemos desarrollar un plan de formación.

De la misma manera, desde el punto de vista tecnológico, deberemos determinar las condiciones del entorno donde implantaremos el subsistema, la capacidad actual de sus recursos y sus condiciones de funcionamiento para verificar que nuestro nuevo subsistema no va a agotar los recursos existentes y no va a provocar cambos en los niveles de servicio del resto del sistema.

El documento que recoja los requisitos de implantación deberá contemplar:

- Gestión de la documentación. Quién tendrá acceso a ella y en qué formato y condiciones.
- Formación de los usuarios.
- Necesidades hardware y software.
- Necesidades de comunicaciones.
- Restricciones de rendimiento que tengan lugar en el entorno de implantación.

Este documento se tendrá en cuenta en la fase de implantación, y puede contemplar casos adicionales como la migración del sistema, recuperación frente a desastres, alternativas o posibles ampliaciones de algunos de los recursos más críticos.

Caso práctico

Definición de los requisitos de implantación del subsistema

Desde el punto de vista tecnológico se prevé que el sistema web tenga dos tipos de entornos:

- Entorno de desarrollo: servirá para realizar pruebas antes de pasar cualquier cambio realizado en el sistema web a producción. Deberá consistir en un servidor que alojará toda la solución.

ANOTACIONS

- Entorno de producción: será el que aloje la solución final y el que usen los clientes de ésta. Estará formado por dos servidores en los cuales se ejecutarán el sistema gestor de base de datos y el sistema gestor de contenidos.

Se prevé que las características de los servidores deberán ser las siguientes:

- Servidor de desarrollo:
  - CPU: Intel Pentium 4 3,2 Ghz.
  - Memoria: 2 Gbytes.
  - Disco: SCSI 100 Gbytes.
  - Red: Ethernet 1 Gb.

- Servidor de producción 1 (sistema gestor de base de datos):
  - CPU: Intel Pentium 4 3,2 Ghz.
  - Memoria: 2 Gbytes.
  - Disco: SCSI 150 Gbytes.
  - Red: Ethernet 1 Gb.

- Servidor de producción 2 (sistema gestor de contenidos):
  - CPU: Intel Pentium 4 3,2 Ghz.
  - Memoria: 2 Gbytes.
  - Disco: SCSI 50 Gbytes.
  - Red: Ethernet 1 Gb (conexión con el servidor de producción 1), Ethernet 1 Gb (conexión a Internet).

Todos los servidores deberán estar conectados a un cuarto sistema, del que la empresa ya dispone, en el cual se realizarán las copias de seguridad pertinentes.

ANOTACIONS

4. Desarrollo

El objetivo de la fase de **desarrollo** es la **construcción ordenada del sistema** del que se ha evaluado la viabilidad, analizado y diseñado. El inicio del desarrollo se produce, en metodologías tradicionales, cuando las fases anteriores se han completado satisfactoriamente y en su totalidad. Metodologías más recientes, y en las que se da preferencia a la ***agilidad*** del ciclo de vida del proyecto, aconsejan pasar a esta fase lo antes posible, avanzando simultáneamente con el análisis del sistema.

Nota

Para obtener más información sobre *metodologías ágiles*, podéis consultar la bibliografía.

No obstante, siempre es conveniente adaptar las metodologías a nuestras necesidades, y siempre es interesante, al igual que en muchos otros ámbitos del software, utilizar los modelos, notaciones o módulos dentro de una metodología muy extensa, que se adapten a nuestra realidad.

Si parte de los desarrolladores deben implicarse en el diseño, puede ser ventajoso para el mismo que se empiece el desarrollo de un subsistema, a la vez que se trabaja en el diseño de otro, y quizá seremos capaces de detectar necesidades de implantación o excepciones que de otro modo provocarían una revisión de parte del diseño con las consecuencias que podría tener al revisar otros subsistemas o componentes.

Existen, sin embargo, un conjunto de especificaciones que conciernen al desarrollo, que deben ser definidas cuando se empiece esta fase, independientemente del ritmo que queramos imprimir al proyecto.

Deberemos ser capaces de planificar el inicio y fin del desarrollo sincronizado de las distintas actividades que al final dejarán el proyecto en condiciones de implantarlo, como son el alcance del propio desarrollo, componentes de terceros a usar o adquirir, sus pruebas, los manuales de usuario, documentación, formación, etc.

ANOTACIONS

4.1. Planificación de las actividades de desarrollo e integración de sistema

Una vez llegados a este punto, ya tenemos información sobre qué necesita ser desarrollado, qué tipo de componentes de software vamos a integrar en nuestro sistema, qué herramientas vamos a utilizar, en qué entorno, etc.

Las actividades de desarrollo que nos permitirán alcanzar el objetivo planteado son:

- **Concretar versiones** o alternativas de los componentes de software, servicios o librerías que vayamos a usar.
- **Estudiar esos componentes**, servicios o librerías.
- Implantar el **entorno de desarrollo**.
- Desarrollar las **pruebas unitarias**.
- **Desarrollar los componentes** necesarios.
- Realizar la **documentación**.
- Planificar la **formación a usuarios** del sistema.
- Desarrollar las **pruebas de integración** del sistema.
- **Aprobar** el sistema.

El objetivo final de esta fase es la **aprobación del sistema para que pueda ser implantado**. Por lo tanto, todo el resto de actividades debe planificarse hacia el cumplimiento de las condiciones de aprobación del mismo.

El orden de las actividades no tiene por qué ser secuencial, y en su grado de paralelismo tendrá mucho que ver la diversidad de recursos y de perfiles que intervengan en el desarrollo. Aun así, hay actividades que mejoran su despliegue desarrollándose en paralelo (como son el propio desarrollo de componentes, su documentación técnica y sus pruebas unitarias) y otras en las que su inicio depende de resultados obtenidos en otras actividades.

ANOTACIONS

Caso práctico

Planificación del desarrollo

Para expresar la planificación del desarrollo del proyecto web de Soluciones Abiertas, utilizaremos diagramas de Gantt. En una primera versión, nos limitaremos a decidir la duración relativa y la sincronización de las diferentes tareas e hitos, así como los recursos dedicados a cada una de ellas.

Figura 4-1. Planificación del proyecto

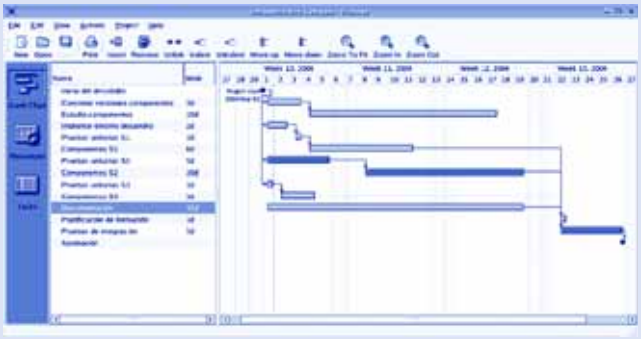


Figura 4-2. Reporte de la planificación completa del proyecto

Estos diagramas (generados mediante la herramienta Planner, ver bibliografía), acompañados de sistemas de control de tiempo empleado en tareas (y registrados en hojas de seguimiento de las tareas del proyecto), permitirán a Soluciones Abiertas seguir la evolución del desarrollo y ajustar sus recursos, hitos y fechas estimadas según la marcha del mismo.

ANOTACIONS

La plantilla que se usará para reflejar el estado del proyecto periódicamente tendrá este formato:

Tabla 4-1.

Proyecto	Sistema web de Soluciones Abiertas
Fecha de finalización	DD/MM/AAAA Estimado original DD/MM/AAAA Estimada actual Cambio desde el último reporte +/- DD días
Ítems pendientes de desarrollo	X defectos X funcionalidades
Ítems pendientes de aprobación	X defectos X funcionalidades
Ítems cerrados	X defectos X funcionalidades
Recursos usados durante este periodo	Persona A: 8 horas Persona B: 23 horas Persona C: 5 horas
Resumen del estado	El proyecto está avanzado según lo planificado.
Documentos relacionados	Plan de proyecto. Diseño de módulos.
Estado detallado	Esta semana nos hemos concentrado en.... Estamos aproximadamente en el X% del proyecto y llevamos dos días de antelación respecto a la fecha prevista...
Gestión de riesgos	Pueden surgir problemas con la versión de uno de los componentes a integrar. Quizá deberíamos evaluar las versiones siguientes o anteriores.
Actividades planeadas	Arreglar ítem #XXX Arreglar ítem #XXX Próxima reunión de desarrollo el DD/MM/AAAA Avanzar en el desarrollo del componente Y
Actualización de la planificación	Aquí podemos copiar el diagrama de Gantt visto en la planificación del proyecto, debidamente actualizado según el uso de recursos, desviaciones, etc.

ANOTACIONS

4.2. Desarrollo

El objetivo de esta fase es **implantar el entorno de desarrollo seleccionado en el equipo que lo va a llevar a cabo y proceder con el mismo**. Entrando en detalle sobre las actividades vistas en la planificación del desarrollo, identificamos las siguientes tareas:

- Preparación del entorno de generación y desarrollo.
- Generación del código de los componentes o procedimientos.
- Ejecución de las pruebas unitarias.
- Ejecución de las pruebas de integración.

Si se han seguido las recomendaciones en cuanto a metodología de análisis y diseño en las fases anteriores, y se dispone de herramientas de asistencia al desarrollo o de generación de código a partir de los diagramas de componentes o clases, el equipo de desarrolladores dispone de toda la información y herramientas necesarias para terminar con éxito su participación en el proyecto.

Si hemos implicado a los desarrolladores en actividades de diseño (como sugieren algunas metodologías recientes) el tiempo de estudio del mismo y de los requisitos del sistema se verá reducido, imprimiendo un mayor ritmo al proyecto en el inicio del desarrollo. Por contra, si se han dejado fases del diseño pendientes de aprobación por haber empezado el desarrollo de otros subsistemas completamente analizados, tenemos que incorporar estas actividades que interrumpirán el desarrollo dentro de la planificación.

Los resultados de las pruebas unitarias diseñadas y desarrolladas anteriormente son un perfecto testigo del avance del proyecto, y nos permitirán conocer en todo momento si el ritmo del desarrollo es el deseado o nos estamos desviando de nuestra planificación.

ANOTACIONS

Caso práctico

Desarrollo del software

Según las decisiones tomadas en las fases anteriores, estamos en condiciones de llevar a cabo el desarrollo del subsistema integración con la aplicación de gestión de la empresa. A partir de la planificación del desarrollo, debemos:

- Instalar el IDE Jedit en los ordenadores de los desarrolladores.
- Acordar un estilo de codificación. Elegimos basarnos en el definido en PEAR: http://pear.php.net/manual/en/standards.php.
- Acordar un conjunto de preferencias del funcionamiento del editor, tamaño de tabulación, estilo de código, etc., de acuerdo al estilo de codificación acordado.
- Generación de código a partir de los diagramas de clases, los casos de uso revisados y las pruebas unitarias.
- Ejecución de pruebas unitarias concurrentemente con el desarrollo.

4.3. Documentación

El objetivo de esta fase es la **elaboración de la documentación de usuario**. Sobre la base de las decisiones tomadas al respecto en fases anteriores, relativas a su formato y disponibilidad, debe desarrollarse su estructura y contenido.

Si se trata de una documentación técnica, deberemos poder incorporar documentación de otras fuentes o componentes que integremos en el sistema. De igual manera, deberemos poder incorporar la documentación de las operaciones y procedimientos que se encuentren en el código fuente.

En el caso de incorporar referencias externas, es de vital importancia **citar explícitamente la versión** que se ha integrado en el sistema, ya que los enlaces externos pueden hacer referencia a la última

ANOTACIONS

versión (que puede coincidir con la integrada en el momento de la creación de la documentación), y más adelante podrían producirse discrepancias.

El **estilo de redacción** de la documentación tiene que ser acorde con el destinatario final de la misma, y se deben indicar claramente los cambios producidos desde la versión anterior.

Caso práctico

Documentación del software

Las decisiones tomadas en fase anteriores indican que la documentación debe estar en formato DocBook.

La documentación de las librerías usadas para conexión al terminal punto de venta virtual, tests unitarios, etc. se encuentra en formato phpDocumentor, así como la documentación técnica contenida en el código fuente.

Mediante la utilidad de conversión de phpDocumentor, podremos volcar la documentación de librerías y de nuestro propio código en la documentación DocBook. Otros componentes o librerías que no dispongan de utilidades similares podrán incorporarse a la documentación como enlaces hipertexto.

ANOTACIONS

5. Implantación

El **paso a producción del sistema** en el entorno en que va a operar requerirá una cuidada planificación de sus actividades. En los casos en que el sistema sustituya a otro de similar funcionalidad, nos encontraremos con un tipo de escenarios determinado (será por ejemplo necesario realizar una migración de los datos). En los casos en que el sistema que se desea implantar añada prestaciones o funcionalidad a un sistema ya implantado, o sea, uno totalmente nuevo, la incidencia de determinadas actividades será distinta.

En todo caso, es en esta fase donde deberemos implicar a los usuarios participantes de los casos de uso analizados del sistema y formarlos en sus nuevas responsabilidades o cometidos.

También deberemos acordar el nivel de servicio que debe prestar el sistema, según las especificaciones, y comprobar que éste se cumple. En el caso en que el sistema deba integrarse en uno ya existente, verificar que su implantación no afecta a su funcionamiento será una de las actividades clave.

Caso práctico

Planificación de la implantación

El calendario de la implantación deberá realizarse atendiendo a las distintas fases implicadas. Para su representación y seguimiento, pueden usarse el mismo tipo de herramientas usadas en la planificación del desarrollo.

Al implantar todo el sistema, los equipos de trabajo y usuarios del resto de subsistemas deberán implicarse en su testeo, y por lo tanto es muy importante disponer de herramientas colaborativas de comunicación y registro de incidencias. El mismo gestor de proyectos puede servirnos para registrar los problemas o incidencias que se vayan produciendo y su solución.

ANOTACIONS

El diagrama de Gantt correspondiente a la planificación de la implantación del sistema podría ser el siguiente.

Figura 5-1. Diagrama de Gantt de la planificación de la implantación

5.1. Formación

La formación en sí misma no suele formar parte de las metodologías relacionadas con proyectos de sistemas de información. Lo que es importante en esta fase es determinar los distintos perfiles de usuario que van a necesitar ser formados sobre el sistema que estamos implantando y adaptar la misma a esos perfiles. El formato y tipo de formación (presencial, no presencial, etc.) también puede venir determinado por esos perfiles de usuario, como puede ser el caso de trabajadores que ejerzan su actividad fuera de las instalaciones de la empresa u otras organizaciones que deban usar el sistema implantado.

En cualquier caso, los estudios de los casos de uso realizados en las fases de análisis y diseño del sistema serán la base para definir el plan y contenidos de la formación.

5.2. Implantación del sistema y pruebas

En esta etapa se llevarán a cabo las **actuaciones para implantar el sistema en su entorno operativo definitivo**. Mientras que las pruebas unitarias y de integración tienen lugar en un entorno diferente del entorno de operación real, las **pruebas de implantación** deben realizarse en el sistema en producción.

Por tanto, se deberá verificar como paso previo que los recursos necesarios atendiendo a los requisitos especificados están disponibles, así como servicios auxiliares, bases de datos, etc.

ANOTACIONS

Adicionalmente, es posible que se deban cargar datos iniciales en el sistema. Esta tarea deberá ser contemplada en esta fase, e incorporada a las pruebas de implantación con los casos reales.

Caso práctico

Implantación y pruebas

Con todos los procedimientos documentados y el sistema probado a nivel de integración, la implantación se limita en este caso a realizar la instalación de los servicios en los servidores definitivos y a implantar las políticas de acceso a dichos servicios, junto con los datos que se especificaron en los casos de uso.

Así pues, deberemos, entre otras actividades:

- Instalar los servicios en los servidores designados.
- Instalar los componentes desarrollados en los servidores designados.
- Instalar el gestor de base de datos, definiendo sus usuarios y políticas de acceso.
- Crear las bases de datos, la estructura de sus tablas y cargar sus datos iniciales.
- Configurar los servicios externos que vayan a usarse por parte de los servicios, como por ejemplo el servidor de correo saliente.

Una vez comprobada la correcta instalación del sistema, activaremos las tareas periódicas que se ejecuten desatendidas (como las copias de seguridad) monitorizándolas durante un ciclo completo de su periodicidad.

A continuación ejecutaremos las pruebas de implantación, de acuerdo a las especificaciones establecidas. Entre otras, podrían ser:

- Navegación completa a través de todo el sitio web, tomando datos acerca de la velocidad de respuesta de éste.
- Compra de paquetes de horas a través del sitio web, comprobando la validez de los pedidos y facturas generados.

ANOTACIONS

Una vez revisados los resultados y contrastados con los requisitos, habrá que decidir si hay incidencias que solventar y qué equipo o subsistema es el responsable. Posteriormente, y según la gravedad de las desviaciones o incidencias registradas, decidiremos si volvemos a ejecutar el plan de pruebas completa o parcialmente.

5.3. Nivel de servicios

Según los resultados obtenidos en las pruebas de implantación, y los requisitos del sistema, estaremos en condiciones de **fijar un nivel de servicio** de cada subsistema. El acuerdo de nivel de servicio deberá contemplar:

- Identificación de los servicios en los que va a haber acuerdo. De qué tipo de servicios estamos hablando:
  - Soporte en línea: tiempo de respuesta, disponibilidad.
  - Comunicaciones.
  - Seguridad.
  - Gestión de recursos: capacidad, horas de servicio para el usuario.
- Propiedades de cada servicio. Unidades en las que se mida la prestación del servicio. Pueden ser numéricas (horas de tiempo de respuesta, capacidad de disco para los usuarios, ancho de banda, etc.) o bien expresadas en términos de restricción de las capacidades de sistema cuando hablemos de seguridad, por ejemplo.
- Estimación de los recursos empleados en prestar el acuerdo de nivel de servicio según los compromisos acordados.

5.4. Aceptación del sistema

Esta actividad consiste en **presentar a los responsables o a dirección toda la documentación relativa a la implantación del sistema**, incluyendo los resultados de las pruebas y el acuerdo de nivel de servicio, **para su aprobación**.

Sólo cuando el sistema ha sido presentado y aprobado formalmente, se considera su paso a producción, y por tanto empieza la prestación del nivel de servicio acordado y su mantenimiento.

ANOTACIONS

6. Mantenimiento

Aunque hemos situado esta fase después de la aprobación del sistema (ya que es cuando se inician sus actividades), su planificación se debe producir a lo largo de todas las fases del proyecto. Muchas de las actividades ya realizadas, como la documentación o el establecimiento de requisitos de desarrollo, van orientadas a facilitar también el **mantenimiento del sistema**.

A diferencia de los proyectos web en los que usamos software propietario, donde estamos obligados a contratar a la organización propietaria del software o alguno de sus socios, el mantenimiento en proyectos web de software libre admite mucha más flexibilidad. Nos podemos encontrar con todos o algunos de los siguientes escenarios:

- Posibilidad de contratar un soporte técnico del componente de software libre a la empresa desarrolladora.
- Posibilidad de formar a nuestros desarrolladores en el componente, ya que disponemos del código fuente, podemos asegurar que si poseen los suficientes conocimientos técnicos, podrán solventar la mayoría de incidencias que se produzcan. Si esto sucede, podemos proporcionar la solución de la incidencia a los desarrolladores del producto para que la incorporen. Durante este proceso, normalmente no tendremos ninguna dificultad en conseguir ayuda por parte de los desarrolladores del componente.
- Posibilidad de contratar a una tercera empresa, experta en la herramienta, que nos proporcione el soporte necesario si la empresa original no ofrece este servicio o el tiempo de respuesta a nuestra petición no se ajusta a nuestro calendario.

Dependiendo de nuestra capacidad técnica, de nuestros recursos y de la urgencia con que necesitemos solventar la incidencia o implementar la nueva funcionalidad, optaremos por una fórmula u otra.

ANOTACIONS

Caso práctico

Mantenimiento del subsistema

Durante la última actualización del contenido del sitio web, hemos encontrado un problema en la herramienta de gestión de contenidos. El software ezPublish escogido para realizar la edición del contenido no actualiza bien la base de datos en caso de que haya en éste cierta combinación de caracteres.

Nuestros desarrolladores nos indican que la única solución sería modificar el código fuente de ezPublish para realizar un chequeo del contenido que se desea actualizar antes de insertarlo en la base de datos, pero que este cambio no es trivial, ya que creen que afecta a múltiples módulos del software.

Consultando el sitio web del producto, vemos que existe una sección donde es posible contactar con empresas que dan soporte al mismo. Contactamos con varias de ellas que nos mandan diferentes presupuestos y después de decidirnos por la que nos ofrece mejores condiciones en cuanto a tiempo de respuesta, recibimos el parche para el producto que pasamos a incorporar en un entorno seguro.

A continuación, ejecutamos las pruebas de integración, y posteriormente implantamos y probamos el subsistema implicado.

ANOTACIONS

Resumen

A lo largo del material del curso se han podido repasar las diferentes **fases de las que consta un proyecto basado en la utilización de tecnologías web**. Dichas fases, dependiendo de la metodología utilizada (especialmente si forma parte del grupo de las denominadas clásicas, o por el contrario forma parte de las nuevas metodologías ágiles) pueden hacerse presentes antes o después dentro de la vida del proyecto, desarrollarse de manera secuencial o en paralelo, etc.

Las fases básicas en las que se suelen dividir los proyectos que hacen uso del web, al igual que cualquier otro proyecto de sistemas de información, son las siguientes:

- **Estudio de viabilidad**: en esta fase se considerará si el proyecto se puede llevar a cabo, teniendo en cuenta las diferentes soluciones existentes y los recursos de los cuales se dispone.
- **Análisis**: en esta fase se estudiarán las necesidades que se desea satisfacer con el nuevo proyecto, con el fin de poder enfocar la solución tecnológica. Asimismo, se especificarán las interfaces de usuario que permitirán a éstos interactuar con el sistema.
- **Diseño**: en esta fase se realizará el diseño tecnológico de la solución, proponiendo una arquitectura global de ésta y estudiando cada uno de los casos de uso existentes.
- **Desarrollo**: en esta fase se construirá la solución, teniendo en cuenta temas como el entorno de desarrollo utilizado, documentación generada, etc.
- **Implantación**: en esta fase se pasará la solución desarrollada a un entorno de producción, por lo que tiene lugar la aceptación definitiva de éste.
- **Mantenimiento**: durante esta fase, que se prolongará a lo largo del resto de la vida del proyecto, se realizará el mantenimiento de éste, tanto a nivel correctivo como evolutivo.

ANOTACIONS

En caso de tratarse de un proyecto que haya que desarrollar dentro de un marco de tecnologías de software libre, en cada una de las fases anteriores se deberá tener en cuenta ciertos aspectos que, si bien normalmente ya se tienen en cuenta, en este tipo de entorno adquieren una especial relevancia (p. ej., viabilidad de la solución, arquitectura global del sistema, licencias utilizadas).

ANOTACIONS

Bibliografía

AgileAlliance (http://www.agilealliance.org/).

Dia a drawing program (http://www.lysator.liu.se/~alla/dia/).

Extreme Programming (http://www.extremeprogramming.org/).

Free License Quick Reference (http://zooko.com/license_quick_ref.html).

Métrica 3 (http://www.csi.map.es/csi/metrica3). "Consejo Superior de Informática y para el impulso de la Administración Electrónica". Ministerio de Administraciones Públicas.

ReadySet (http://readyset.tigris.org/).

Planner (http://planner.imendio.org/).

The Object Management Group – UML (http://www.omg.org/uml/).

World Wide Web Consortium (http://www.w3c.org/).

ANOTACIONS

Appendix A. GNU Free Documentation License

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of *copyleft* which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

ANOTACIONS

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The *Document* below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as *you* You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A *Modified Version* of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A *Secondary Section* is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The *Invariant Sections* are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The *Cover Texts* are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

ANOTACIONS

A *Transparent* copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not *Transparent* is called *Opaque*.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The *Title Page* means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, *Title Page* means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section *Entitled XYZ* means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as *Acknowledgements*, *Dedications*, *Endorsements*, or *History*. To *Preserve the Title* of such a section when you modify the Document means that it remains a section *Entitled XYZ* according to this definition.

ANOTACIONS

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

ANOTACIONS

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the

ANOTACIONS

Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.

- I. Preserve the section Entitled *History*. Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled *History* in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the *History* section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled *Acknowledgements* or *Dedications* Preserve the Title of the section, and preserve in the section all the

ANOTACIONS

substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

- M. Delete any section Entitled *Endorsements* Such a section may not be included in the Modified Version.

- N. Do not retitle any existing section to be Entitled *Endorsements* or to conflict in title with any Invariant Section.

- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled *Endorsements*, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

ANOTACIONS

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled *History* in the various original documents, forming one section Entitled *History*; likewise combine any sections Entitled *Acknowledgements*, and any sections Entitled *Dedications*. You must delete all sections Entitled *Endorsements*.

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this

ANOTACIONS

License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an *aggregate* if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a

ANOTACIONS

translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled *Acknowledgements*, *Dedications*, or *History*, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License *or any later version* applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version

number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled *GNU Free Documentation License*

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the *with...Texts.* line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

ANOTACIONS


UOC
www.uoc.edu