

Software libre

Remo Suppi Boldrito
Josep Jorba Esteve

XP06/M2118/02155



Proyecto en Administración de Redes y Sistemas Operativos basados en GNU/Linux

Remo Suppi Boldrito

Autor

Doctor en Informática y profesor del Departamento de Arquitectura de Computadores y Sistemas Operativos de la Universidad Autónoma de Barcelona. Desde el punto de vista académico, se dedica esencialmente a sistemas operativos y proyectos de final de carrera de la Ingeniería Informática (Técnica y Superior) mientras que la investigación la realiza en sistemas distribuidos de cómputo oportunístico, simulación distribuida y sistemas de vídeo bajo demanda.

Josep Jorba Esteve

Autor

Ingeniero superior en Informática por la UAB. Magíster en Arquitectura y Procesamiento Paralelo por la UAB. Profesor de Arquitectura y Sistemas Operativos del Departamento de Arquitectura de Computadores y Sistemas Operativos de la UAB. Consultor de los Estudios de Informática y Multimedia de la UOC. Profesor ayudante en el Departamento de Arquitectura de Computadores y Sistemas Operativos ETSE-UAB.

Segunda edición: febrero 2007
© Fundació per a la Universitat Oberta de Catalunya
Av. Tibidabo, 39-43, 08035 Barcelona
Material realizado por Eureka Media, SL
© Autores: Remo Suppi Boldrito, Josep Jorba Esteve

Se garantiza permiso para copiar, distribuir y modificar este documento según los términos de la *GNU Free Documentation License, Version 1.2* o cualquiera posterior publicada por la *Free Software Foundation*, sin secciones invariantes ni textos de cubierta delantera o trasera. Se dispone de una copia de la licencia en el Apéndice A, junto con una traducción no oficial en el Apéndice B.

Índice

Agradecimientos	5
Introducción	7
Objetivos	13
1. Estudio de viabilidad	15
1.1. Necesidades y requisitos del cliente	17
1.2. Análisis de la situación actual	21
1.3. Definición de requisitos del sistema	21
1.4. Estudio de alternativas de solución	23
1.5. Valoración y elección de las posibles soluciones	26
2. Análisis del sistema	30
2.1. Definición del sistema	30
2.2. Requisitos exactos del proyecto	31
2.3. Establecimiento de requisitos	35
2.4. Definición de interfaces de usuario	37
2.5. Especificación del plan de pruebas	38
3. Diseño del sistema	41
3.1. Arquitectura	42
3.2. Definición de niveles de arquitectura	42
3.3. Especificación de estándares, normas de diseño y construcción	45
3.4. Identificación de subsistemas	47
3.5. Casos de usos reales	48
3.6. Revisión de casos de uso por subsistema	49
3.7. Especificaciones de desarrollo y pruebas	50
3.8. Requisitos de implantación	55
4. Desarrollo	60
4.1. Planificación de las actividades de integración del sistema	61

4.2. Cómo elegir la licencia más adecuada	65
4.3. Entorno de desarrollo	67
4.4. Documentación	68
5. Implantación	70
5.1. Formación	72
5.2. Implantación del sistema, pruebas y nivel de servicio	73
5.3. Aceptación del sistema	76
5.4. Mantenimiento	76
Resumen	78
Bibliografía	80
Licencia Pública General de GNU	81
Preámbulo	81
Licencia pública general de GNU	82

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 de Software Libre ofrecido por la citada institución.

Introducción

El desarrollo de proyectos en administración de redes y sistemas operativos bajo entornos de software libre es similar a cualquier otro tipo de proyecto con algunas sutiles diferencias. Es necesario para llegar a buen fin, seguir un proceso que pase por la definición clara y cuidadosa del alcance del problema que se quiere solucionar, el diseño, el desarrollo, la puesta en funcionamiento y el mantenimiento de la solución que se haya escogido. Es importante, según el tipo de proyecto, tener en cuenta los aspectos formativos para los usuarios y administradores que deberán poner en producción la solución desarrollada. Estos aspectos no se deben descuidar, ya que la bondad de la solución propuesta pasa por que los usuarios finales puedan resolver sus problemas desde el primer momento. En caso contrario, el proyecto necesitará una serie de modificaciones y retoques que no son compatibles con un simple mantenimiento y será necesario desarrollar nuevas versiones con el coste añadido y los problemas de puesta en marcha.

Aunque el responsable de proyectos en entornos de software libre para la administración de redes y sistemas operativos, o de un desarrollo específico basado en la utilización de software libre, no necesita ser un experto en toda la tecnología y aplicaciones utilizadas, sí que es necesario que conozca todos los pasos o fases del mismo. También deberá conocer todos los puntos (hitos) críticos en el desarrollo del proyecto, teniendo claro cuál debe ser el estado de avance y qué elementos se deben generar después de cada paso o fase, y si se está en condiciones de pasar a la siguiente con garantías de éxito o se debe revisar la actual.

El responsable del proyecto debe, con una visión global, dominar cada uno de los elementos que intervienen en el proyecto y poner en juego las habilidades para:

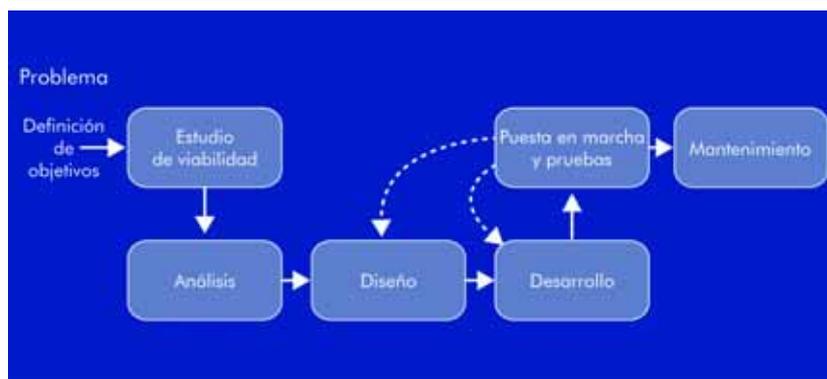
1. **Planificar.** Se deben planificar los recursos (físicos y humanos), así como la información necesaria en cada una de las fases del proyecto. Es necesario que la asignación de recursos se realice de

forma racional en función de la disponibilidad, el coste, la especificidad y el tiempo.

2. **Organizar.** Los aspectos de organización son fundamentales en el éxito del proyecto y son responsabilidad del encargado del mismo. La optimización de la utilización de los recursos y el coste de los servicios son dos de los elementos que están directamente correlacionados con la organización y previsión (planificación). Dentro de esta organización no deben descuidarse aspectos de coordinación entre los proyectos vinculados de la empresa, la gestión y administración de la información y tener en cuenta factores humanos y/o imponderables que puedan afectar al buen desarrollo del mismo.
3. **Verificar y validar.** El responsable del proyecto debe estar en condiciones de seguir el avance del proyecto y poder tomar decisiones respecto a la calidad del mismo, verificando si los resultados parciales coinciden con lo planificado y validándolo parcialmente con los requisitos y objetivos propuestos.

El objetivo del presente curso es definir las fases vitales que debe contemplar un proyecto y que el responsable deberá seguir y supervisar durante la vida del mismo.

Figura 1. Ciclo de vida de un proyecto



Como se muestra en la figura anterior, el ciclo de vida del proyecto está formado por fases e interrelaciones entre ellas. Las interrelaciones mostradas en líneas de punto dependen del grado de cumplimiento de los objetivos y no necesariamente están en todos los proyectos. Pueden existir otras interrelaciones y el responsable del proyecto deberá evaluar y tomar la decisión sobre si es necesario modificar el diseño/desarrollo sobre la base de la consecución de las

metas propuestas y el grado de cumplimiento de los objetivos establecidos. Es evidente que estas decisiones son complejas, ya que se pueden salir fuera del alcance del proyecto y pueden generar costes adicionales. Es responsabilidad del jefe de proyecto evaluar si el proyecto cumple o no los requisitos del cliente y tomar las medidas oportunas.

A partir del problema, las fases de un proyecto pueden describirse como:

- **Definición de los objetivos y alcance.** En muchos proyectos, éstos son requisitos del cliente y no se cuenta como una fase propiamente dicha, pero en otros, a partir de la especificación del problema, deben definirse los objetivos en forma clara y concisa. Esta definición se consigue después de una serie de reuniones entre cliente (quién encarga/utiliza el proyecto) y jefe de proyecto – responsable del grupo de trabajo que diseña, desarrolla e instala la solución informática en cuestión– (se pueden incluir colaboradores de ambas partes). A la aceptación de los objetivos del proyecto se llega después de la exposición de ideas funcionales sin detalles técnicos (lluvia de ideas) y con el compromiso por ambas partes.
- **Estudio de viabilidad.** Dentro de esta fase se analizarán cuidadosamente las posibles soluciones a los requisitos del proyecto para la consecución de los objetivos, qué soluciones técnicas existen, y cuál es la más adecuada (calidad, funcionalidad y coste).
- **Análisis.** Tiene por finalidad describir detalladamente el sistema que se desea realizar, qué requisitos funcionales debe tener y cómo se cumplirán los objetivos establecidos.
- **Diseño.** En esta fase se desarrolla la idea desde el punto de vista tecnológico, seleccionando qué herramientas se utilizarán y cuál es la arquitectura de la aplicación.
- **Desarrollo.** En esta fase, se realizará la programación de la solución adoptada, proponiendo, en función del equipo, diferentes modelos de desarrollo. Es importante que durante el desarrollo se vayan haciendo pruebas de validación parciales para la depuración de errores.

- **Puesta en marcha y pruebas.** Durante esta fase se instalará el sistema y se realizarán pruebas de integridad antes de pasarlo a producción. Esta fase puede generar que se deban realizar retoques en el desarrollo o en el diseño para solventar problemas que hayan surgido durante las pruebas. Una vuelta atrás en el desarrollo es aceptable en la mayoría de los proyectos, pero se debe tener en cuenta que durante la fase de diseño es complicado y que la decisión debe ser meditada con cuidado. Posteriormente, si se da el visto bueno, se pasará el sistema a producción con la finalidad de que los administradores y usuarios comiencen a utilizarlo.
- **Mantenimiento.** Con la práctica, los usuarios encontrarán los errores que puedan surgir de situaciones particulares y se analizarán las posibles mejoras (versiones evolutivas) de la aplicación. Durante esta fase o la anterior, dependiendo del proyecto, se suelen incluir sesiones de formación para los administradores/usuarios para realizar la cesión gradual de responsabilidades desde el equipo del proyecto hacia los usuarios finales del sistema. Este aspecto es importante para evitar la dependencia de un equipo con respecto al otro y la sobrecarga de trabajo, sobre todo si el equipo de proyecto pertenece a la misma institución/empresa.

Con la definición anterior o bajo otros nombres/apartados, estas fases estarán presentes en cualquier proyecto que involucre las tecnologías de la información y la comunicación independientemente de la metodología utilizada para su gestión (secuenciales, cascada, concurrentes, etc.).

Con relación a este documento-guía, se utilizará un **caso práctico** con el fin de aplicar los conocimientos teóricos, y para cada apartado se presentará un ejemplo (basado en el caso práctico) para mostrar aspectos concretos de cómo se debe desarrollar por parte de los estudiantes el punto en cuestión (se identificará con *Caso práctico: ...*). Es necesario tener en cuenta que la solución propuesta no es la única posible a dicho proyecto y que sólo tiene como fin presentar una de las posibles alternativas en el desarrollo del mismo sirviendo como marco de referencia para su desarrollo. Durante la etapa de aprendizaje, es conveniente trabajar por analogía, para, de esta manera, ejercitar todas las fases propuestas, aprender a resolver las dificultades que el mismo genera y seguir (durante este período de formación) una metodología adecuada en su consecución.

En el presente proyecto se considera que será desarrollado por un equipo formado por un jefe de proyecto (responsable de todas las fases) y un grupo de trabajo (analistas, diseñadores, programadores, etc.). Dependerá del volumen del grupo de trabajo y la complejidad del proyecto la distribución de responsabilidades entre el jefe de proyecto y su grupo. Normalmente, en función del proyecto, el jefe de proyecto define el grupo de trabajo y qué tareas/responsabilidades debe asumir cada uno de los miembros formando una organización jerárquica. En otros casos, el jefe de proyecto es un miembro más del grupo de analistas/diseñadores/desarrolladores y además de la dirección del proyecto realiza otras tareas específicas.

Objetivos

Hay una serie de objetivos que el estudiante debe conseguir al final del curso de **Proyecto final en entornos de software libre para la administración de redes y sistemas operativos**. Dentro de los mismos, se pueden enumerar los siguientes:

- Asimilar los conocimientos en cuanto a la realización de proyectos en entornos de software libre, que si bien se puede aplicar una metodología general en el desarrollo de proyectos, presentan ciertas particularidades en cuanto a la tecnología, la capacidad de decisión y la bondad de la solución.
- Adquirir experiencia y conocimientos en determinar cuáles son los puntos vitales en un proyecto de administración de redes y sistemas operativos, adaptando a la situación particular las fases que se han definido anteriormente.
- Analizar, seleccionar y probar qué herramientas de software libre se utilizarán para desarrollar el proyecto y cuáles se utilizarán para administrar las redes y los sistemas operativos, teniendo en cuenta elementos tales como el entorno, la capacitación de los usuarios y la autonomía que deberán tener los mismos cuando el proyecto haya finalizado.
- Aplicar al caso propuesto no sólo la metodología y los conocimientos adquiridos a lo largo de los diferentes módulos, sino vincular todos estos conocimientos en un proyecto que deberá ser desarrollado utilizando herramientas basadas en software libre. Asimismo, el proyecto (resultado del trabajo del estudiante) deberá estar basado en el mismo tipo de herramientas.
- Generar toda la documentación y la información necesaria para que, una vez el proyecto se encuentre en producción (aplicado a

la empresa y con usuarios de la misma), permita resolver las tareas básicas vinculadas al mismo como son la instalación y configuración, supervisión, adaptación al entorno cambiante de una empresa o interpretación de los resultados. Esta información también deberá incluir manuales de usuario para que éstos puedan aprender a trabajar con la solución propuesta sin la intervención del equipo de proyectos.

1. Estudio de viabilidad

La importancia vital para cualquier proyecto dentro del ámbito de administración de redes y sistemas operativos es el estudio de viabilidad, ya que será éste, a partir de unas necesidades planteadas, el que permitirá escoger la mejor solución que abarque las necesidades del cliente. Es importante, por lo tanto, que el estudiante **asuma** el rol del jefe de proyectos, e interprete los requisitos y necesidades del enunciado del proyecto. En un caso real, el jefe de proyecto deberá mantener un diálogo en profundidad con los responsables de la empresa que encargan el proyecto para que le transmitan todas las necesidades, especificaciones y requisitos que deberá cumplir la solución adoptada y evitar soluciones incompletas o añadidos a la solución inicial.

En el estudio de viabilidad, se debe comenzar considerando tres aspectos básicos en toda solución informática:

- El estado actual del sistema y su configuración (hardware y software).
- Qué problemas se deben solucionar y cuáles son los requisitos de dicha solución, tanto estructurales como funcionales, sin descuidar las particularidades de quién trabajará (instalará y configurará, mantendrá, utilizará, ...) con dicha solución y su grado de formación.
- Cuáles son las restricciones al sistema, al cual se debe aplicar la solución, ya que en general el jefe de proyectos se encontrará con un *statu quo* que no podrá o le costará mucho cambiar y que puede ir desde aspectos de comportamiento social (los usuarios están acostumbrados a...) hasta aspectos tecnológicos (se utiliza/debe utilizar tal hardware, tal base de datos o la conexión es de X Mbps, ...).

El estudio de viabilidad debe considerar las diferentes propuestas para llegar a un acuerdo con el cliente que será quien tomará la de-

cisión final guiado por sus asesores y por el jefe de proyectos. Por lo tanto, este estudio, además de un resumen ejecutivo, deberá incluir los siguientes aspectos:

- **Económicos:** un informe detallado preliminar pero con un alto grado de detalle sobre el coste de la solución propuesta (o de cada una de ellas si existen alternativas u opciones). Este análisis de costes debe ser riguroso e incluir todos los gastos previstos en la consecución del proyecto, evitando la costumbre de incrementar los costes en un determinado porcentaje “por las dudas” o “por lo que pueda pasar”. En proyectos donde existen costes muy difíciles de valorar en el momento del análisis de viabilidad, se fijan los procedimientos (por contrato) mediante los cuales se puedan hacer frente económicamente sin tener que redefinir todo el proyecto.
- **Técnicos:** estos aspectos permitirán tomar una decisión precisa sobre cuál de las propuestas es la más adecuada a las necesidades, y que junto con el análisis económico permitirán la adecuación de la misma. Puede darse el caso de que las soluciones más adecuadas desde el punto de vista tecnológico signifiquen cambiar la arquitectura del sistema o formar a todo el personal transformando esta opción en inviable. Ante estas situaciones, se deberá tomar una solución de compromiso.
- **Legales:** dependiendo del tipo de instalación y datos que maneje la empresa y dada las normativas legales (protección de datos, propiedad intelectual, confidencialidad, seguridad industrial, etc.) el estudio deberá incluir aspectos vinculados a estos temas y cómo la solución propuesta los aborda.
- **Aspectos de funcionamiento:** se deberán incluir los métodos y procedimientos operativos para cada uno de los diferentes aspectos contemplados en la solución propuesta.

En cuanto a la selección de la mejor solución informática, podemos diferenciar dos escenarios posibles:

- **Proyecto final del curso.** Durante el desarrollo del curso, el estudiante deberá escoger la mejor solución teniendo en cuenta tanto el impacto en la organización como en los diferentes miembros de la misma, el coste asociado, no sólo económico sino en adap-

tación y esfuerzo de los usuarios, etc., y los riesgos asociados que implica la puesta en marcha de la solución propuesta. Este último aspecto debe tenerse especialmente en cuenta, ya que será uno de los más importantes en la empresa: no tiene que haber pérdida de la información y hay que tener en cuenta que se deben evitar o minimizar (sin son inevitables) las interrupciones del servicio durante y después de la implantación de la solución.

- **Proyecto en entorno real de la empresa.** Si bien se pueden aplicar los mismos criterios del párrafo anterior se deben incluir los representantes de la empresa, a los cuales habrá que explicar cuál es la balanza de beneficios-riesgos de la solución adoptada para que, conjuntamente, se tome la decisión adecuada y la responsabilidad no recaiga únicamente en el jefe de proyectos (o por lo menos que ésta sea delegada al jefe de proyectos con el consentimiento de la empresa).

Los siguientes párrafos presentan el problema y detallan, para el caso del ejemplo propuesto, cómo se lleva adelante el estudio de viabilidad.

1.1. Necesidades y requisitos del cliente

Como se mencionó anteriormente, ésta es una fase muy importante en la vida del proyecto: interpretar bien las especificaciones, requisitos y necesidades del cliente permitirá diseñar la solución adecuada para este problema.

Las necesidades pueden ser planteadas por un cliente externo si trabajamos para una empresa que se dedica a desarrollar soluciones de estas características, es decir el proyecto será para otra organización (en este caso es aconsejable dedicarle un tiempo a analizar la características de la empresa, modelo de negocio, hábitos y costumbres, tecnología utilizada, etc.), o la propia empresa dispone de un departamento de proyectos con un responsable y el proyecto es para otro departamento o unidad de la misma empresa. En este último caso, el jefe de proyectos ya parte con un conocimiento base de cómo funciona la empresa y las cuestiones vinculadas.

Es recomendable para esta primera parte del estudio es hacer un análisis “desde arriba hacia abajo” (*top-down*) comenzando por una

descripción general e ir profundizando en detalles de los aspectos comentados en el párrafo anterior que tenga especial relevancia.

Es importante que el jefe de proyecto (estudiante en nuestro caso) obtenga una visión global del problema sin detenerse en detalles y a continuación comience a desglosar cada uno de los aspectos mencionados.

Como caso práctico se utilizará una empresa que necesita actualizar su sistema informático y un conjunto de servicios nuevos (ver recuadros posteriores). El estudiante debe tener en cuenta que este ejemplo se utiliza para explicar en forma aplicada los conceptos teóricos y por lo tanto los detalles que se dan son únicamente a modo de ampliación de este caso específico, pero que son útiles en la mayoría de proyectos de estas características.

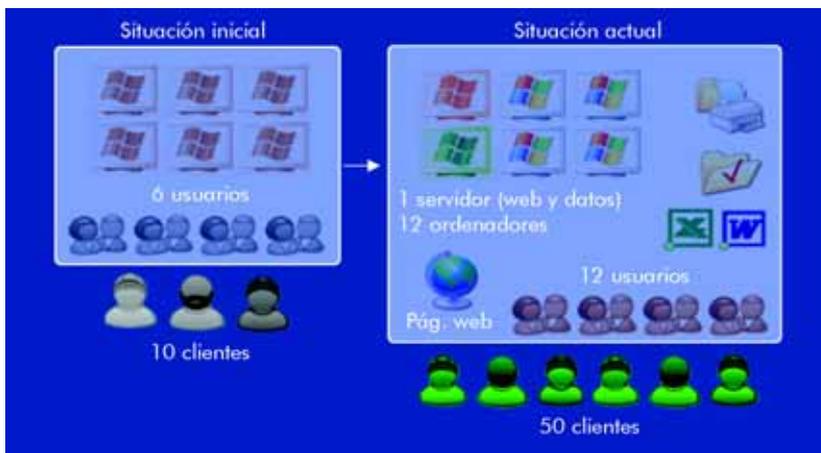
Caso práctico: Situación actual

Proyecto de adecuación y renovación de red y servicios informáticos para la empresa NtEum S.A.

La empresa **NtEum** es una empresa dedicada a la prestación de servicios en el ámbito editorial y cuenta con un sistema informático basado en soluciones propietarias que ha tenido un crecimiento muy elevado en un período de tiempo corto y no planificado.

La empresa ha pasado de un parque informático de 6 ordenadores basados en tecnología Windows[®]98, autogestionado, con 6 usuarios en ofimática, administración y servicios a clientes externos (10 clientes) a un sistema basado en la misma tecnología pero diferentes versiones (Windows[®]98, Windows[®]2000) de 12 ordenadores más 1 servidor (página web y datos internos) con 12 usuarios (empleados de la empresa) y 50 clientes externos. Este cambio se ha producido en 6 años y ha sido regido por las necesidades de la empresa sin planificación ni renovación tecnológica (sólo con nuevas adquisiciones).

Figura 1-1. Pasado y presente de NtEum



Caso práctico: Proyecto

Proyecto de adecuación y renovación de red y servicios informáticos para la empresa NtEum S. A.

La empresa solicita un proyecto de renovación, adaptación y crecimiento para hacer frente a un conjunto de servicios no prestados en la actualidad, un aumento a 100 clientes externos, que integre parte del sistema informático actual (6 máquinas de usuarios con datos compartidos e impresión en red), y que dicha solución esté basada en software libre.

El proyecto debe contemplar la integración/sustitución de los servicios actuales, el soporte para los nuevos servicios y la formación de los usuarios y administradores del sistema con la ampliación a 20 usuarios (administrativos/comerciales-empleados de la empresa) y 3 técnicos (administradores y/o gestores del sistema informático -empleados de la empresa). La solución adoptada debe permitir a la empresa cumplir con toda la legalidad vigente, tener presencia en Internet a través de una plataforma de contratación y gestión de servicios electrónico (**e-services**) y poder realizar la transferencia de información (productos y gestión) vía canales de comunicación seguros entre empresa y cliente externo.

Figura 1-2. Proyecto en administración de redes y SO de NtEumactual



El nuevo sistema debe integrar los 6 ordenadores Windows®, debe prestar los siguientes servicios internos basados en software libre: acceso único (NIS), datos compartidos (NFS, y datos Windows®), seguridad (zona de servidores y DMZ) impresión en red, servicio de web y mail, base de datos (clientes y personal), y ofimática. Además, para los clientes de la empresa debe proporcionar los siguientes servicios: WWW, mail, servicio de impresión en las máquinas de la empresa, FTP, foros, comercio electrónico, base de datos. Todos los servicios de datos deberán incluir políticas de resguardo –backup– adecuadas a sus características.

Es importante considerar que el cambio tecnológico deberá ir acompañado de formación (si bien se incorporará soporte técnico como empleados de la empresa), que toda la infraestructura afectará al trabajo de la empresa y que este trabajo debe ser cuidadosamente planificado.

1.2. Análisis de la situación actual

Después de la descripción de necesidades y requisitos, es necesario hacer un análisis de los sistemas de la empresa y realizar un diagnóstico lo más preciso posible haciendo una lista de tareas y actividades (*checklist*) para no descuidar ninguna (por evidente o mínima que sea). Esta lista permitirá tener en mente la integración del sistema actual con el nuevo y evitar improvisaciones cuando el proyecto esté en pruebas.

En el presente caso, se incluirán: las características técnicas de los ordenadores (*hardware*, sistema operativo y *software*) y las aplicaciones y servicios que deberán integrarse en el nuevo sistema. Asimismo, es importante fijar qué usuarios participarán en el estudio de la situación actual para resolver cuestiones y/o tomar decisiones de diseño de la nueva infraestructura. En nuestro ejemplo práctico, será necesario describir cada uno de los sistemas:

- Ordenadores: tipo, SO, aplicaciones y servicios.
- Impresoras: tipos y servicios.
- Red: tipo, estructura y características de seguridad.
- Datos compartidos: tipo de servicio, características de seguridad.
- Estructura física: dónde se encuentran los equipos y dónde serán alojados los nuevos sistemas/usuarios/personal técnico.
- Para los puntos que así lo justifiquen (por ejemplo, los que estén vinculados a estructuras físicas o lógicas), se pueden incluir diagramas para clarificar la estructura de los servicios o su dependencia, y del hardware.

1.3. Definición de requisitos del sistema

Con la descripción del sistema (y teniendo en consideración la opinión de los usuarios del mismo, sobre todo en la parte que se deberá integrar), se deben describir los requisitos que debe cumplir el proyecto del cual se estudiará la viabilidad.

Con ellos se podrán evaluar las alternativas posibles para solucionar el problema propuesto. Es necesario tener en cuenta que en esta descripción debe incluirse una prioridad (p. ej. entre 1 y 10, siendo 10 la máxima prioridad) con el objetivo de tener una valoración relativa de cada uno. Así, en la fase de diseño de la arquitectura se sabrá cuántos recursos hay que asignarle. La forma más simple de elaborar esta lista de requisitos es sencillamente con una descripción, una breve explicación y la prioridad que se le asigna.

Caso práctico

En nuestro caso se deberá realizar una lista descriptiva de los servicios que se necesitará instalar para dimensionar, más adelante, la arquitectura del sistema y decidir cuál es el hardware que hay que comprar. En el caso del ejemplo presentado, el detalle de los servicios se puede enumerar como:

- Acceso único (10): servicio de NIS y NFS para que cada usuario pueda acceder a los datos independientemente del terminal que se use.
- Acceso controlado a web y mail seguro (10).
- Base de datos de la empresa con datos sobre clientes y personal para la facturación y nóminas (10) (separadas).
- Sistema de seguridad a través de cortafuegos y DMZ (10).
- Impresión en red (9).
- Servicio de acceso a publicar datos en el sistema web de la empresa (9).
- Gestión de copias de respaldo e información de los sistemas (logs)(8).

Para los clientes de la empresa se deben ofrecer los siguientes servicios:

- Portal web (8): información de servicios y punto de entrada a la compra de servicios por comercio electrónico.
- Impresión remota (7): los clientes pueden enviar sus impresiones de alta calidad a los dispositivos de la empresa.
- Servicio de almacenamiento de archivos (7): los clientes pueden alquilar un espacio de almacenamiento para mantener allí sus archivos particulares (FTP en forma segura –SFTP–).

- Foros de discusión, atención al cliente y soporte (FAQ) (5).
- Servicio de base de datos a los clientes (4).

Los requisitos legales deben ser cubiertos por el control de seguridad y la separación de datos con identificación de los usuarios y seguimiento de la actividad en cada servidor. En cuanto a licencias de software, deben basarse (siempre que sea posible) en soluciones de software libre y lo menos restrictivas posible. Los requisitos económicos deben fundamentarse en el equilibrio entre el coste y el servicio que se debe dar (es importante contemplar si los servidores necesitan sistemas de 24 horas y 7 días/semana de funcionamiento en cuanto al mantenimiento del hardware y a los sistemas de alimentación interrumpida –SAI–). Los requisitos operativos pasan por la responsabilidad del personal técnico de la empresa para mantener todos los servicios una vez configurados y puestos en marcha por el equipo de proyectos.

1.4. Estudio de alternativas de solución

Con los requisitos que plantea el nuevo proyecto, se estudiarán diferentes soluciones, siempre y cuando sea posible, que cumplan con todos éstos. Para ello, es necesario contemplar toda la información recogida hasta el momento y para cada alternativa se deberá especificar tanto su definición funcional, como técnica, sus puntos fuertes y sus puntos débiles, y describir el valor económico que supone tanto en costes directos (p. ej. licencias) como en costes derivados (p. ej. mejoras –updates–). Tampoco se debe descuidar la información acerca de estándares que cumple, implantación en el mercado y soporte que ofrece.

Caso práctico

En nuestro ejemplo práctico, uno de los requisitos es que la solución se base en software libre, pero pueden analizarse diferentes opciones dentro de estas variantes:

Licencia GPL –Gnu public license– (mínimo coste): servidores y máquinas de escritorio basadas en distribuciones como Fedora, Debian, etc.

- Parcialmente en licencia GPL: servidores en Fedora, Debian o similares y escritorios en opciones con licencia GPL, como por ejemplo Linspire, SuSE, Mandrake, etc.
- Licencia LGPL u otras similares (mayor coste): servidores y escritorio basados en distribuciones como Red Hat, Mandrake, SuSE, etc.

Se deben analizar cuidadosamente estas opciones teniendo en cuenta el coste derivado de las licencias que, en este proyecto, aparte del hardware, será el mayor coste junto con el de personal.

Se debe recordar que la definición de código abierto es una especificación del tipo de código y no es en sí misma una licencia de software. Una de las licencias de este tipo más comunes es la GPL, y bajo ésta, el software puede ser copiado y modificado, pero las modificaciones han de hacerse públicas bajo la misma licencia. La GPL se diferencia de la licencia LGPL, que es prácticamente igual, pero permite la mezcla con software propietario. Existe una gran cantidad de software libre que incluye parte propietaria, como por ejemplo, MPL y NPL (*Mozilla public license* y *Netscape public license*) en Mozilla, BSD en Apache, etc.

Para contabilizar y facilitar el siguiente paso, es interesante confeccionar una tabla que presente las características posibles como la siguiente que se muestra a modo de ejemplo (el alumno deberá completar la información sobre el precio de las distribuciones y en la última fila deberían incluirse las necesidades mínimas para procesador, memoria, disco y el precio resultante de este ordenador –no incluidos en este ejemplo–):

Caso práctico

Figura 1-3. Comparación entre los diferentes sistemas operativos

Sistema Operativo	Red Hat Linux Woody	Manjaro Linux Community 10	Red Hat Linux Versión 9	Windows Professional
Recomendado Para	Servidores-Escritorio	Escritorio-Servidores	Servidores-Escritorio	Escritorio-Servidores
Tipo de usuario	Técnico	No Técnico	Técnico	No Técnico
Coste Aproximado	€	€	€	€
Tiempo instalación (min) (mismo PC aprox.)	27' (dependiendo elección)	20' (dependiendo elección)	40' (dependiendo elección)	58'
Seguridad	Excelente	Excelente	Avanzada	Regular (afectada por virus especialmente)
Fiabilidad	Muy Alta	Muy Alta	Muy Alta	Normal (incluyendo todos los SPack)
Ejecución desde CD sin Instalación (LiveCD)	No (directamente)	Si	No (directamente)	No
Soporte para Ofimática	Incluye OpenOffice (compatible con MSOffice)	Incluye OpenOffice (compatible con MSOffice)	No por defecto (se puede instalar OpenOffice costo 0€)	No por defecto Opciones: OpenOffice Of, MSOffice STD 2003 3995
Permite ejecución y creación de aplicaciones MS-Windows	Si a través de Wine -debe incluirse-	Si a través de Wine -incluido-	Si a través de Wine -debe incluirse-	Native
Permite la ejecución y creación de aplicaciones Linux	Native	Native	Native	Parcial a través de software libre CygWin
Herramientas: Adm. control de red, desarrollo software, etc.	Si	Si	Si	Parcial (algunas de Adm. y unidades)
Base de Datos, Firewall, Servidores, etc.	Si	Si	Si	Firewall: Si (XP) Web: Si (XP, .Net)
Necesidades mínimas de Hardware	Procesador Memoria Disco	Procesador Memoria Disco	Procesador Memoria Disco	Procesador Memoria Disco

Relacionado con este aspecto, y dependiendo del proyecto, es interesante calcular lo que se denomina TCO (*total cost of ownership*, coste total de la propiedad de un ordenador) que comporta las partidas siguientes:

- Coste total del hardware y software
- Mejoras previstas en hardware y software (*updates*)
- Mantenimiento
- Soporte técnico
- Entrenamiento o aprendizaje

Se puede tomar como base para una estimación de este coste 3-4 veces el costo de compra del PC y puede aportar información sobre la plataforma y/o arquitectura más adecuada a seleccionar. En nuestro caso, este cálculo lo haremos de manera separada para ver la influencia de los diferentes elementos que definen la inversión total (aproximada) del proyecto.

1.5. Valoración y elección de las posibles soluciones

Con una lista de requisitos y características técnicas de las posibles alternativas, se debe escoger la solución más apropiada para el proyecto. En esta fase se tendrá en consideración toda la información recogida hasta el momento (descripción general, alcance, situación actual, etc.) y para cada alternativa se deberá especificar en qué consiste, tanto a nivel funcional como técnico.

En nuestro caso, la tabla anterior será de vital importancia para la toma de decisiones. En esta tabla (aquí sólo hecha a modo de ejemplo y con datos figurativos) debe ampliarse para que incluya una lista de todos los servicios que son necesarios para cada requisito y así pueda facilitar la selección de la solución más adecuada. Esta tabla permitirá, además, poder hacer un informe ejecutivo de cada opción.

Caso práctico

Por ejemplo, para el punto 4 de las necesidades del caso práctico propuesto: *Sistema de seguridad a través de cortafuegos y DMZ (prioridad 10)*

- **Debian:** integrado y de alta confiabilidad y funcionalidad (iptables). Configuración muy técnica. Grado de adaptabilidad: 100%.
- **Mandrake:** integrado (Shorewall). Versatilidad limitada (es posible instalar iptables). Muy fácil de configurar y controlar.
- **RedHat:** análogo a Debian.
- **Windows®:** sólo en XP y 2003 con SP2. Limitado y configuración aceptable.

Es necesario tener en cuenta que en el análisis económico se deben considerar los costes directos y los indirectos de cada solución, tal y como ya se mencionó anteriormente. Además de estudiar la viabilidad económica de las diferentes soluciones, se deberán analizar los riesgos asociados a cada una de ellas.

Para cada una de las alternativas existentes será necesario describir qué incertidumbres, problemas potenciales, etc. existen. Por ejemplo, se deberá valorar como riesgo: seguridad, continuidad del SO, responsabilidades legales, ciclo de mejoras estable (*updates*), etc. Para cada uno de los riesgos potenciales detectados, se deben incluir las alternativas posibles para hacer frente a estas posibles amenazas y en qué forma pueden afectar a la capacidad de funcionamiento del sistema.

Además, como toda la infraestructura debe ser autónoma una vez instalada y configurada, no deben descartarse los costes asociados a la formación de los usuarios y los técnicos de la empresa. Existen proyectos en los que es posible contratar empresas dedicadas al mantenimiento y actualización de estos servicios (*outsourcing*) y los cuales tendrán también unos costes asociados. En este último caso, se debe realizar un contrato preciso que detalle todas las opciones y prestaciones que deben ser realizadas, así como las penalizaciones que se deriven por incumplimiento de las condiciones.

Caso práctico

Teniendo en cuenta todos los elementos de juicio antes planteados, la solución adoptada (una de las posibles) para nuestro sistema ejemplo será:

- **Sistema operativo:** Fedora Core 3.
- **Acceso y administración:** NIS (acceso a los datos independiente del ordenador), Webmin, LinuxConf (administración y configuración).
- **Archivos e impresión:** NFS (compartición de archivos), Samba (compartición de archivos desde sistemas Windows), CUPS (servidor de impresión).
- **Red y seguridad:** DHCP (servicio de información para máquinas), NAT (translación de direcciones de red), DNS (servicios de información de nombres), Xinetd (servicio de conexiones seguro), SSH (servicios de conexión interactivo seguro), Wu-FTP, SFTP (servicio de transferencia de ficheros –normal y seguro–), Iptables (cortafuegos), Nmap, Nessus, Snort, Logcheck, Tripwire (herramientas de seguimiento, seguridad y control), GnuPG (gestión de firma digitales y encriptación).

- **Servicios web, mail y news:** Apache-Tomcat (servicio de www), Exim (servicio de correo), IMAP (servicio de correo a clientes), LeafNode (servicio de noticias).
- **Servicio de base de datos:** PostgreSQL.

Con toda esta información se procederá a generar un informe, que incluirá un presupuesto detallado de los costes del proyecto (tanto los costes directos como indirectos). El informe deberá ser aceptado y aprobado por ambas partes (clientes – grupo de proyectos). Dependiendo de la relación entre cliente y grupo de proyecto, es decir, si es un cliente externo o si el proyecto es para la misma empresa, se firmará un anexo al contrato que contenga los elementos necesarios que indiquen cuáles son las obligaciones/responsabilidades de cada una de las partes.

2. Análisis del sistema

Una vez definida la solución que hay que adoptar en el estudio de viabilidad, es necesario realizar una especificación detallada de la misma con el objetivo de preparar su diseño y su arquitectura. Esta especificación se realiza durante la fase de “*análisis del sistema*”, donde es muy importante la interacción con los usuarios de éste para descartar la posibilidad de omisión y/o errores que lleven a diseños no adecuados.

2.1. Definición del sistema

Como se verá en los párrafos siguientes, se debe describir el sistema con un alto grado de detalle, especificando no sólo la función de cada elemento, sino su comunicación con los restantes componentes. También deberá ser necesario especificar qué usuario será el responsable de cada uno de estos componentes y hasta dónde llegará su responsabilidad.

Es importante destacar que en el presente apartado y para el caso práctico propuesto, la descripción se realizará teniendo en cuenta la de alto nivel realizada en el estudio de viabilidad del apartado anterior, pero que se hará con mayor profundidad. Cabe considerar que este análisis debe servir para poder, más adelante, hacer el diseño de la solución adoptada.

Es por ello por lo que dentro de este apartado se utilizará la enumeración de requisitos del estudio de viabilidad para establecer las necesidades exactas del mismo y cuál será el intercambio de información (si la hay) entre los diferentes componentes del sistema. Se debe remarcar la importancia de este apartado, ya que una definición inadecuada del sistema de intercambio de información puede dar lugar a un diseño que no contemple todas las necesidades de la

empresa. Esta situación implicaría vueltas atrás en el proyecto con los consabidos retrasos y costes añadidos a la finalización del proyecto.

2.2. Requisitos exactos del proyecto

- **Requisitos legales.** Dado que la empresa procesará datos relativos a cliente y empleados, se debe cumplir todas las normativas vigentes en protección de datos y acceso a la información. Por ello, el diseño deberá contemplar los métodos de identificación y control de acceso a la información de acuerdo a la ley y con los mecanismos de certificación, control/cifrado adecuados para cumplir lo establecido legalmente.
- **Requisitos de propiedad intelectual y licencias.** El proyecto debe basarse en software libre y con licencias lo menos restrictivas posible, teniendo en cuenta las inversiones necesarias, tanto en la adquisición inicial como en el servicio postventa de actualizaciones y mantenimiento.
- **Requisitos de acceso único.** Los datos de la empresa se centralizarán en un servicio distribuido (interno) de archivos mediante NFS. Esta opción deberá contemplar los requisitos de seguridad para la aceptación del servicio de las máquinas clientes y con los criterios adecuados en cuanto a los permisos de lectura-escritura para cada máquina cliente. También los datos de usuario de la red deberán estar centralizados por un servicio de información distribuida (NIS), de modo que cada usuario tenga en la empresa una única cuenta y una palabra clave para el acceso a los servicios desde todas las máquinas. Dada la existencia de máquinas Windows, será necesario la integración con un servidor Samba para el acceso unificado a los archivos desde estas máquinas.
- **Requisitos de acceso web, mail y grupos de noticias.** Los servicios de acceso a páginas de información de la empresa deberán contemplar la posibilidad de acceso cifrado (https) mediante SSL, ya que el acceso a algunas partes de la información deberá ser confidencial para algunos clientes. Asimismo, deberá contemplar módulos de identificación para el acceso personalizado a dicha información. Los servicios de correo electrónico también deberán ser cifrados a través de SSL, tanto si son mediante clientes IMAP o

POP (imaps y pops), como si lo es mediante webmail (https). Los servicios de correo deberán contemplar la inclusión de recursos antivirus y antispam, tanto para los mensajes de entrada como para los de salida. El acceso a los grupos de noticias podrá ser abierto pero la inclusión de noticias deberá ser mínimamente controlado, por ejemplo, a través de registro previo y con moderadores para cada tema, ya que este sistema será utilizado tanto para las preguntas frecuentes (FAQ), como para la información de soporte.

- **Requisitos sobre la base de datos.** La base de datos de la empresa deberá ser sólo accesible a las máquinas de la Intranet, ya que contendrá información sensible sobre clientes, facturación y nóminas. Esta base de datos podrá accederse a través de peticiones SQL desde servicios web (p. ej. a través de páginas web con PHP) o interactivamente, pero siempre con palabra clave en ambos casos.
- **Requisitos del sistema de seguridad.** El sistema informático de la empresa (servidores y máquinas clientes) deberá tener un diseño basado en un cortafuegos institucional con una definición clara de zonas (DMZ), con direcciones privadas y traslación (p. ej. a través de NAT). El sistema deberá contar con las herramientas de detección y análisis adecuadas que permitan a los administradores realizar un control adecuado del sistema informático. La organización interna de la red se basará en un sistema de distribución de direcciones IP automático (DHCP) y un servidor de nombres secundario interno (DNS). Deberá tenerse en cuenta la posibilidad de que cada usuario posea un certificado digital interno (GnuPG) gestionado a través de una unidad certificadora local (CA).
- **Requisitos de impresión en red.** Desde cada máquina de la empresa se podrá acceder a un servicio de impresión centralizado mediante colas (CUPS), donde cada máquina cliente deberá ser identificada y contabilizado el trabajo de impresión realizado.
- **Requisitos de servicio de acceso para publicar datos en el sistema web de la empresa.** La empresa dispondrá de un sistema de acceso a la información vía un servicio de web, y los usuarios del sistema (empleados de la empresa) podrán publicar y/o actualizar información de la empresa, siendo ellos mismos responsables de dicha publicación, todo lo cual deberá realizarse mediante métodos de identificación adecuados. Habrá que implementar un control de acceso para este servicio y un registro de mo-

dificaciones. Asimismo, se deberá habilitar un sistema de registro de actividad de servicio de web con el fin de tener información analítica para la empresa (desde dónde se accede y a qué información) como desde el punto de vista del análisis de seguridad.

- **Requisitos de gestión de las copias de respaldo e información de los sistemas.** El sistema informático deberá contar con un servicio de copias de respaldo de la información planificada teniendo en cuenta que éstas habrán de cumplir unas condiciones de almacenamiento físicas de seguridad cuando los datos tengan requisitos de confidencialidad. Asimismo, deben integrarse todos los datos de seguridad (logs) en un sistema centralizado para que el análisis sea efectivo y real del sistema en conjunto, no sólo de máquinas particulares.

Para los clientes de la empresa, se ofrecerán los siguientes servicios:

- **Requisitos del portal web.** El portal de información de la empresa deberá admitir servicios complementarios como puede ser la compra de servicios ofrecidos por la empresa, impresiones remotas en dispositivos de alta calidad u otros servicios con valor añadido. Para ello, se tendrán en cuenta las posibilidades de petición del servicio, identificación del cliente, modo de entrega y pago del mismo. Deberán establecerse modelos diferentes si son clientes registrados/habituales o clientes nuevos u ocasionales.
- **Requisitos de servicio de almacenamiento de archivos.** La empresa ofrecerá un servicio de almacenamiento de información a clientes registrados, por lo cual se deberá montar un servicio de gestión de espacio (cuotas) y acceso de clientes remotos a sus espacios de disco (conocido generalmente como FTP, pero que deberá ser implementado como SFTP para evitar problemas de seguridad). Este servicio deberá ser fiable, por lo cual el sistema contará con copias de respaldo gestionadas por la empresa de manera separada a las copias de respaldo propias.
- **Requisitos de servicio de base de datos a los clientes.** Los clientes podrán contar con un servicio de base de datos con acceso remoto. La empresa sólo proveerá un servicio de alojamiento y acceso seguro a la información de la base de datos del cliente, pero no será responsable de la creación de la base ni de los datos

almacenados. Su responsabilidad será sólo de servicio y seguridad para el acceso a dicha base de datos.

Por otro lado, la empresa cumplirá con unas cuestiones de organización en cuanto al mantenimiento del sistema informático, tanto hardware como software:

- **Requisitos tecnológicos, mantenimiento y administración.** La empresa deberá contratar un plan de mantenimiento hardware o formar a las personas adecuadas para prestar este servicio. Dada la magnitud del proyecto, es crucial para el buen funcionamiento posterior a la implantación/integración de la formación de los administradores y de los usuarios. Esta formación podrá realizarse durante los períodos de prueba para generar confianza y para que acepten la nueva tecnología sin preconceptos ni falsas expectativas.

En cuanto a la base de las aplicaciones, el sistema se basará en GNU/Linux Fedora Core 3 y aplicaciones de software libre (CUPS, Apache, Postgres, Samba, etc.) y si fuera necesario desarrollar algún servicio complementario, éste se efectuará bajo el mismo esquema utilizando lenguajes que los administradores-programadores de la empresa conozcan (C, C++, PHP, Shell Script, etc.)

- **Requisitos de organización.** La empresa deberá realizar un organigrama de recursos humanos para redefinir las tareas y las responsabilidades en el nuevo sistema. Esta reorganización tendrá por objetivo asignar las nuevas responsabilidades entre los empleados de la misma teniendo en cuenta los nuevos servicios y productos de la empresa. Asimismo, es conveniente que la empresa genere un "Libro Blanco" que recoja no sólo la funcionalidad de la empresa, sus servicios y productos, sino los procedimientos y las responsabilidades de cada empleado responsable en forma funcional. Esta información la conocerán todos los empleados y será donde quedará anotado por escrito las responsabilidades de cada uno de los integrantes de la plantilla de la empresa.
- **Requisitos de seguridad.** La empresa, además, deberá contar con un documento de seguridad que describa tanto las responsabilidades de los usuarios que trabajan con información sensible, como los procedimientos que hay que seguir en caso de que se descu-

bran problemas relativos a la seguridad (intrusos, pérdida de confianza, vulnerabilidades, etc.) con el fin de incluir todos los pasos que deben seguir los responsables ante situaciones de este tipo.

2.3. Establecimiento de requisitos

El objetivo de esta fase será complementar los requisitos definidos anteriormente incorporando la información de los usuarios y directivos de la empresa. Es aconsejable que la discusión de la primera fase (usuarios-directivos) se realice en función de los niveles de responsabilidad, ya que se tiende a hablar, por ejemplo, con los directivos, de cuestiones técnicas o de muy bajo nivel, y con usuarios técnicos, sobre decisiones de empresa. Para iniciar esta tarea, es importante dividir el sistema en subsistemas, lo cual facilitará el análisis y permitirá clarificar los diferentes aspectos sobre servicios y recursos con los cuales contará el sistema.

La obtención de los requisitos estipulados por los usuarios (directivos de la empresa, usuarios técnicos, etc.) será el primer paso en el establecimiento de requisitos. Durante reuniones de discusión con ellos, se deben fijar y esbozar los criterios mínimos, averiguando cuáles son las necesidades, cuáles son “expresiones de deseo” y cuáles aquellos elementos que son indiferentes para unos u otros usuarios. Es conveniente una distribución de los requisitos en ámbitos para centrar el debate y las discusiones, como por ejemplo en los siguientes:

- **Funcionales.** Seguir los lazos de movimiento, procesamiento y utilización de la información determinando los servicios necesarios para ello.
- **De prestaciones.** Cómo deberán ser optimizados los recursos y dónde se encontrará la información, cómo se accederá a ella y qué criterios se utilizarán para hacerlo.
- **De seguridad.** Cómo afectará el movimiento de la información a la seguridad de la misma y qué elementos/servicios se deberán tener en cuenta en su organización.
- **De implantación.** Cómo se organizará la información, dónde se almacenará y quién será su responsable, tanto para los servicios

como para la atención a los clientes externos de la empresa. En determinados casos, es necesario también especificar cómo será el transitorio entre el actual sistema de información y el nuevo, y cómo se realizará este cambio.

- **De disponibilidad.** Cómo deben garantizarse los servicios y qué debe ocurrir en caso de fallo de uno de ellos.

Los responsables de la empresa (directivos) deberán establecer los procedimientos de acceso, seguridad, confidencialidad, o en su caso de negocio, de la información que se controla y/o manipula en la empresa.

Los responsables técnicos (administradores) deberán establecer los criterios y/o necesidades para ofrecer el servicio adecuado y discutir la prestación de nuevos servicios. Deberán también opinar sobre cuestiones como interfaces de usuario, funcionalidad y distribución de los servicios, programación de nuevos recursos, etc. Es importante tener en cuenta los requisitos de los usuarios técnicos y aprovechar la experiencia en el sistema.

El siguiente paso será la especificación formal de los **casos de uso**, donde se deberá incluir una descripción del problema, cómo los usuarios y responsables interactuarán con el sistema, qué interfaces utilizarán y qué procedimientos seguirán en caso de errores, fallos o averías en alguno de los componentes del servicio.

Caso práctico

Ejemplo de caso de uso: acceso a un servicio de soporte

La empresa determina que el sistema de soporte de servicio de impresión remota se realizará mediante el portal de servicios integrados de la empresa. El cliente (externo) accederá a este servicio mediante una página web en la cual deberá registrarse o, si ya lo está, introducir su identificador y clave, lo cual se validará en el registro respectivo de la base de datos.

En el caso de que el usuario no esté registrado, se procederá al registro y se enviará por correo electrónico el identificador y clave de acceso de servicio. El responsable del servicio de soporte a clientes será notificado de esta nueva petición.

Cuando el cliente externo introduzca su pregunta, el sistema enviará una petición al responsable de servicio y éste asignará la petición a un técnico después de verificar las cuestiones relativas a la petición. El técnico responderá al cliente quedando registrado cuándo se ha realizado la petición y qué tipo de respuesta se ha dado al problema. Quedará en manos del responsable del servicio la cuestión de si esta consulta pasará como FAQ o quedará como una consulta interna.

Mediante la consulta con los directivos de la empresa, los administradores y los responsables de servicio, se debe verificar si éste es el procedimiento correcto, si todos están de acuerdo en cuanto a criterios de calidad, servicio y responsabilidad y si es así, se deberá fijar qué elementos hardware y software serán necesarios para cumplir estos requisitos.

Habrá que tener en cuenta que cuantos más casos de usos se generen, mejor quedarán establecidas (y además, por escrito) las necesidades de la empresa y menos probabilidades surgirán de vuelta atrás durante la implantación, debido a procedimientos no contemplados u olvidados. También es necesario recordar que el jefe de proyectos debe escuchar todas las propuestas y ser receptivo a las necesidades intentando hacer prevalecer criterios de servicio, negocio y empresa, de manera que el usuario por sí mismo descarte opciones personales o conveniencias de un departamento.

2.4. Definición de interfaces de usuario

En esta fase del análisis se especificará cómo serán las interfaces de comunicación no sólo usuario-máquina, sino usuario-usuario, usuario-responsable y usuario-cliente externo. En el caso de usuario-máquina (considerando como usuarios a los empleados de la empresa para la cual se está desarrollando el proyecto), que es donde deberá prestar la mayor atención el jefe de proyecto, se realizará una especificación teniendo en cuenta diferentes perfiles de usuarios, qué tipos de permisos tendrán, y con qué capacidades y flexibilidades

contarán, así como qué nivel jerárquico ocupa cada uno y cómo se accede a información de servicios vinculados o dependientes.

Una vez descritos los perfiles de usuario, se identificará qué clase de interfaces tendrán y cómo accederá a la información cada uno de ellos.

Caso práctico

Definición de interfaz de usuario: interfaz del responsable de actualización de base de datos de clientes y empleados.

- ¿Se utilizará desde un sistema Unix o Windows?
- ¿Es usuario técnico o administrativo?
- ¿La interfaz deberá ser en modo texto o gráfica?
- ¿Los registros de transacciones deben ser enviados al responsable del servicio?
- ¿La validación del usuario se realizará mediante variables de sesión o mediante claves incorporadas al acceso de la base de datos?
- ¿La memoria caché de la máquina de acceso debe ser eliminada una vez terminada la transacción?

Mediante tablas de definiciones de este estilo, se definirán las interfaces para cada perfil teniendo en cuenta que éste sólo debe tener los privilegios de acceso a la información de la cual es responsable.

2.5. Especificación del plan de pruebas

El último aspecto que hay que contemplar en la fase de análisis es la especificación de plan general de pruebas que servirá para establecer si el sistema cumple con todos los requisitos de las partes involucradas en el proyecto (directivos, administrativos, usuarios, técnicos, etc.)

Normalmente, se acostumbra a dividir el plan de pruebas en diferentes niveles que suelen ser útiles para desvincular servicios, departamentos o unidades funcionales.

- **Pruebas unitarias:** del servicio (por ejemplo, conexión a la base de datos desde perfiles administrativos o de gestión económica).
- **Pruebas de integración:** acceso de clientes a servidores en DMZ cumpliendo los criterios de seguridad y protección de la información estipulados.
- **Pruebas de sistema:** verificación de si el servicio cumple con todos los requisitos estipulados por el usuario/cliente.
- **Prueba de implantación:** cómo se realizará la transferencia de información entre el sistema actual y el nuevo y cómo se administrarán los transitorios.
- **Pruebas de aceptación:** cómo se identificarán los usuarios ante el nuevo sistema y cómo accederán a la información y aceptarán sus nuevas responsabilidades.

Este conjunto de pruebas es sumamente crítico, ya que permitirá validar el sistema en su conjunto teniendo en cuenta desde aspectos obvios, como la funcionalidad, a aspectos más subjetivos, como seguridad, rendimiento, disponibilidad, etc.

Para cada prueba se deberá estipular el **horizonte** de la misma, entendiendo por horizonte a qué usuarios implica, qué servicios contempla, qué criterios de aceptación debe cumplir la prueba para ser superada, cuál será el entorno de pruebas (hardware y software), cómo se realizarán las pruebas y en qué condiciones, etc.

Si bien este paso de pruebas puede ser tedioso y poco creativo, garantiza que el diseño se realizará sobre las condiciones adecuadas y que los resultados podrán cumplir los requisitos especificados. Es por ello por lo que se recomienda al jefe de proyectos especial cuidado en la definición de las pruebas, ya que éstas serán la garantía, una

vez superadas, de que el sistema funciona de acuerdo a los criterios preestablecidos durante las primeras fases del proyecto.

Caso práctico

Especificación de pruebas: prueba de integración entre el servicio de web y de base de datos.

Permitirá conocer si un cliente puede acceder a cambiar información personal guardada en la base de datos de la empresa mediante un acceso seguro a través de una página web.

- El cliente sólo deberá poder cambiar su filiación, pero no otra información como código de cliente, información económica, etc.
- El acceso deberá ser seguro.
- Se deberá comprobar la comunicación entre servidor web y servidor base de datos.
- Se deberá verificar la imposibilidad de cambios no autorizados de información.
- Se deberá verificar la comunicación autorizada entre servidor web y base de datos.
- Se trabajará sobre datos reales y con usuarios reales del sistema accediendo desde Internet verificando el cifrado de las comunicaciones.
- Se verificarán accesos y elementos de seguridad tanto dentro de la intranet (servidor-servidor), como externos (navegador del cliente-servidor web).

La prueba será considerada como superada cuando se haya podido cambiar sólo la información deseada y verificado todos los aspectos de seguridad mencionados.

3. Diseño del sistema

El objetivo de la fase de diseño de un proyecto de estas características es obtener los elementos (modelos y especificaciones) definidos en la etapa de análisis (etapa anterior), pero de forma concreta. Básicamente, se debe tener en cuenta que en esta etapa se inicia el proceso de selección de los servicios y su distribución física, y se analiza cómo serán ejecutados estos servicios en una arquitectura hardware. Además, se deberán determinar las especificaciones de desarrollo e integración, así como definir el entorno de pruebas y seleccionar qué criterios se utilizarán para que éstas sean representativas del correcto funcionamiento del sistema.

Se pueden resumir las tareas en esta fase como:

- Definición de la arquitectura del sistema: identificación de los componentes hardware, su interconexión, jerarquía software, seguridad y privilegios. Es decir, todo lo que es necesario para que el sistema pueda ser configurado y puesto en marcha con garantías.
- Especificaciones, requisitos de seguridad y estándares que se utilizarán tanto en esta fase, como en el desarrollo del sistema.
- Identificación de los subsistemas, requisitos de integración, licencias y funcionalidad.
- Casos de utilización especificados anteriormente, revisados para obtener las condiciones de funcionamiento determinadas en las fases anteriores.
- Servicios, componentes, interfaces, métodos y clases que se deberán instalar en la fase de desarrollo.
- Asegurar la garantía del éxito de la implantación del sistema.

Como se puede observar, esta fase será esencial, donde muchos conceptos se determinarán y, aunque existen algunas tendencias de

desarrollo y construcción iterativas, con el fin de obtener resultados de forma inmediata e identificar así fallos en el diseño, es importante recordar que todas las decisiones en cuanto a seguridad, especificaciones o estándares tomadas aquí facilitarán la tareas futuras.

Es importante identificar aquí los servicios y garantizar aspectos legales, ya que éstos pueden afectar a la funcionalidad y comprometer la utilización del sistema por cuestiones no derivadas de su ejecución, sino de elementos que contravienen normas o aspectos jurídicos. Estas cuestiones pueden impedir dar los servicios con legalidad y garantía, no sólo para la empresa, sino para los clientes y los trabajadores de la misma.

3.1. Arquitectura

La arquitectura del sistema es el primer paso para identificar los elementos hardware y dónde se ejecutarán los servicios. El objetivo es disponer de un conjunto de documentos y diagramas completos (que contengan todo el nivel de detalle necesario y suficiente), que sean comprensibles para personal no técnico, como por ejemplo la dirección de la empresa, y a la vez, que puedan ser utilizados como base para profundizar en el diseño del sistema.

3.2. Definición de niveles de arquitectura

En nuestro caso utilizaremos la separación funcional-lógica de la arquitectura, pero existen otros esquemas para realizar la definición de la misma:

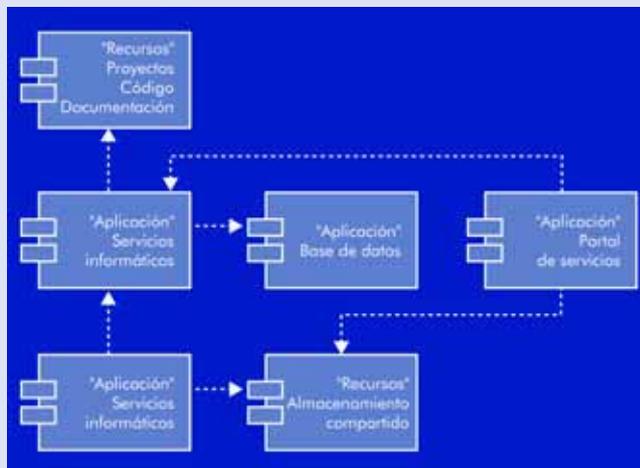
- **Arquitectura funcional:** descripción a grandes bloques del sistema, sin entrar en detalles e identificando las interrelaciones entre ellos. Será útil para comunicar y verificar con la dirección de la empresa y con personal no técnico la visión global de sistema.
- **Arquitectura lógica:** añade detalle a la anterior e incorpora los detalles de la interacción de cada uno de los subsistemas (bloques), que permitirán a cada instalador-desarrollador trabajar preocupándose sólo del trabajo encargado.

Caso práctico

**Caso práctico de definición de arquitectura:
arquitectura global de la empresa NtEum S. A.**

Se puede representar la arquitectura del nuevo sistema con notación UML (*unified modeling language*) definida por el Object Management Group y tarjetas CRC (Clase-Responsabilidad-Colaborador) de soporte.

Figura 3-1. Arquitectura al nivel de componentes



En el diagrama anterior se pueden observar a modo de ejemplo los componentes del sistema (una distribución de ellos) y los conectores que los unen. Éstos indican que algún tipo de comunicación se produce entre ellos y es recomendable a este nivel mezclar componentes técnicos y no técnicos (por ejemplo, de negocio), los cuales deberán estar debidamente identificados con un estereotipo: <<recursos>>, <<aplicación>>, u otros.

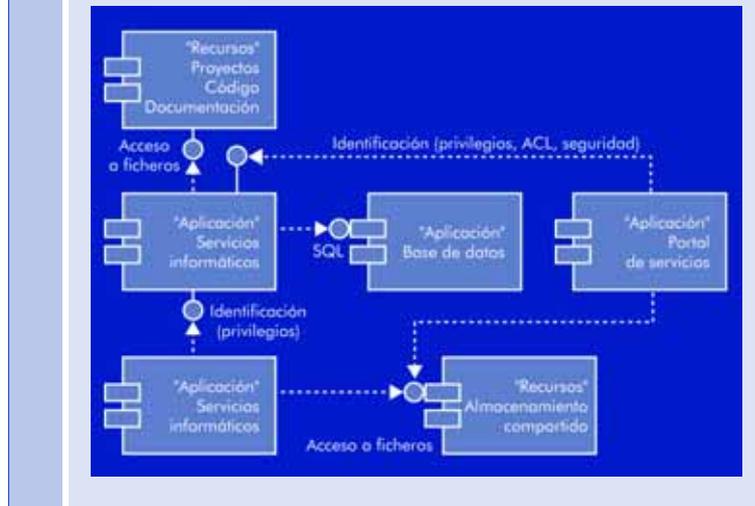
Con este diagrama y el consentimiento del cliente, se pasará a profundizar sobre las interfaces de los componentes para obtener así la arquitectura lógica del sistema extendiendo el diagrama anterior y detallando los procesos de comunicación.

Nota

El UML se puede consultar en <http://www.uml.org/uml> y el Object Management Group se puede consultar en <http://www.omg.org>.

Caso práctico

Figura 3-2. Arquitectura lógica



Como apoyo a la generación del diagrama anterior o para validar las interfaces, se pueden utilizar tarjetas CRC (class responsibility collaborator) tal como:

Tabla 3-1. Ejemplo de tarjetas CRC para el portal de servicios

Portal de servicios	
Ofrece los servicios a los clientes de la empresa <ul style="list-style-type: none"> - Web, mail y news - Impresión remota - SFTP - Comercio electrónico - Base de datos 	Identificación y ficheros

Debe figurar en primer lugar el nombre del componente. En la columna izquierda se deberá reflejar todo lo que el componente hace (servicios que presta en nuestro caso) o también los datos que posee. Se incluirá todo lo que es de su responsabilidad y la información que debe mantener. En la parte derecha se incorporarán los elementos con los cuales se relaciona para llevar a cabo los servicios indicados en la izquierda.

Estas tarjetas son utilizadas generalmente en las últimas tecnologías de diseño y planificación del software y permiten "definir" diagramas de componentes dinámicamente, utilizando un espacio (por ejemplo, una mesa) para colocar las tarjetas en función de su grado de comunicación (cerca o lejos del recurso/aplicación), para lograr así una visión global de la arquitectura lógica durante una sesión de planificación.

3.3. Especificación de estándares, normas de diseño y construcción

Es importante que todas las personas vinculadas al proyecto (tanto para los que desarrollarán como para los que implantarán y mantendrán) seguirán unas normas de generación de documentación. Se debe elegir no sólo la forma como se realizarán los diagramas, con qué herramientas y cómo se especificarán, sino también cómo se generará toda la documentación. Es recomendable utilizar herramientas bien conocidas y probadas que utilicen formatos estándares. En cuanto al lenguaje, detalles y notación, se recomienda utilizar también estándares predefinidos o en caso de adoptar uno interno, es preceptible que éste sea conocido (o que su aprendizaje sea simple) por todas las personas vinculadas de una u otra forma al proyecto, ya que no sólo facilitará la comunicación con los diferentes grupos involucrados, sino que permitirá su reusabilidad y comprensión por parte de personas externas o recién incorporadas al equipo de trabajadores.

Se debe definir entre otras cosas:

- Herramienta, formato y plantilla de los documentos de diseño.
- Notación utilizada en los diagramas de diseño.
- Recomendaciones de estilo, detalles, idioma y formato de la documentación técnica, así como las herramientas utilizadas para generarla y/o modificarla.

Caso práctico

Definición de las normas y notaciones del proyecto de adecuación renovación de servicios informáticos de NtEum S. A.

Es recomendable que los documentos creados a partir de este punto, y que serán motivo de revisión por parte de diferentes personas con diferentes grados de conocimientos técnicos, compartan características y uniformidad, así como un formato adecuado. En nuestro caso (particular) se optará por:

- **Documentos de diseño:** estos documentos se deben poder consultar tanto por el personal técnico, como

por otros miembros de la empresa sin formación de esta características. Se acuerda trabajar en formato SXW, que es el estándar de Open Office para los documentos susceptibles de sufrir modificaciones con control de correcciones y basado en una plantilla predefinida que contenga: *título* del documento, *responsable* del documento, *lista de autores* que han intervenido y la *fecha* en la cual lo han hecho, *resumen ejecutivo* de cambios introducidos (control de versiones) indicando cambio, fecha y autores. Asimismo, se utilizará cada documento y se generará simultáneamente en el formato PDF para su comunicación y publicación a personas vinculadas al proyecto, pero sin capacidad de introducir cambios.

- **Diagramas de diseño:** para ellos se utilizará la notación UML.
- **Documentación técnica:** ésta será la que probablemente más revisiones y controles deberá pasar y tendrá enlaces a las herramientas utilizadas, interfaces de programación (API), configuración y ejemplos, etc., por lo que se recomienda utilizar un formato lo más flexible posible e integrable con las propias herramientas utilizadas. Es recomendable por las características del proyecto utilizar XML para este tipo de documentación aprovechando estándares como podrían ser DocBook (<http://www.docbook.org> o también <http://www.oasis-open.org/docbook/>) para facilitar la interacción y comprensión. Con éste se podrá:
 - Jerarquizar un documento y estructurarlo en diferentes ficheros, los cuales podrán ser revisados independientemente.
 - Fácil inclusión de referencias a otra información como archivos, URL, figuras, etc.
 - Generación simple de otros formatos para su visualización como PDF y HTML.
 - Como se trabaja con XML, los ficheros son independientes del editor utilizado.
 - Permitirá obtener documentación incluida en el código fuente de las aplicaciones e incluirla de forma automática en la mayor parte de situaciones.

Es importante destacar que este tipo de herramientas son de última generación y puede ser que algunas de las personas involucradas en su desarrollo/implantación/mantenimiento no esté al corriente de las mismas. Sin embargo, se deberá contemplar esta situación y tomar las medidas correctoras necesarias, ya que su utilización redundará en una mejor accesibilidad a la información y mejorará el ciclo de vida de mantenimiento del sistema. Y puesto que toda la información se encontrará disponible, será muy fácil su actualización y revisión.

3.4. Identificación de subsistemas

Para reducir la complejidad de diseñar todo a un nivel de detalle determinado, se puede dividir el sistema en secciones o subsistemas agrupados por funcionalidad (más efectivo desde el punto de vista técnico) o por organización (más comprensible desde el punto de vista no técnico) para facilitar su comprensión, revisión y utilización.

Para realizar una división funcional, es importante tener en cuenta:

- Funcionalidad común o relacionada por interacción y/o ejecución.
- Acceso a recursos compartidos o validados por la misma identificación (datos o sistemas).
- Integración en una interfaz de usuario única a pesar de que los servicios puedan estar físicamente distribuidos.
- Mejoras en la utilización de la información sobre todo cuando ésta debe tener niveles altos de seguridad o ser de gran volumen.

Caso práctico

Identificación de subsistemas:

En nuestro caso particular, analizando según la funcionalidad, se obtiene:

- Subsistema servicios informáticos (integrador modelo 3-tier con respecto a la base de datos y módulo de seguridad)
- Subsistema portal de servicios
- Subsistema servicios internos
- Subsistema de base de datos
- Subsistema de almacenamiento compartido
- Subsistema de acceso a la documentación

Aplicando un criterio uniformador, sería deseable definir una única interfaz integrada para obtener así un único subsistema (no aplicable en este caso) o bajo un criterio de optimización de recursos, intentar reducir los subsistemas al mínimo posible, pero sin perder definición o separación de funciones en el sistema real.

Con esto se dispone de una visión (mapa) de la arquitectura del sistema global, que una vez verificado y validado, permitirá profundizar en cada uno de sus componentes a través de los casos de usos del sistema.

3.5. Casos de usos reales

Una vez identificados los subsistemas, es el momento de analizar los “casos de uso” establecidos en la fase de análisis y determinar las operaciones que se deben desarrollar de cada uno de ellos.

A partir de los escenarios escogidos en la fase de análisis, se determinarán qué subsistemas están incluidos en cada uno de ellos y se diseñará su funcionamiento teniendo en cuenta:

- El ámbito donde se aplican considerando no sólo aspectos tecnológicos.
- Las situaciones particulares de cada caso de uso (excepciones).
- Detalles vinculados a la implementación identificados en esta fase. Se deben tener en cuenta a este nivel de diseño todos los aspectos vinculados a la seguridad global del sistema.
- Restricciones aplicables a nivel de interfaz.
- Nuevos requisitos o funcionalidades necesarias y no contempladas en el análisis inicial.

Si el sistema que se está desarrollando incluye bases de datos como es nuestro caso, se deberá definir la estructura de la misma (tablas, métodos, sentencias SQL, validación, etc.).

Es importante durante esta fase revisar todos los requisitos del sistema y establecer quién los servirá (incluyendo aspectos de seguridad,

interacción y vinculación). De forma consecuente, también se obtendrá el diseño de todas las pruebas que asegurarán el correcto funcionamiento del sistema durante el desarrollo y las condiciones bajo las cuales se implantará el mismo.

3.6. Revisión de casos de uso por subsistema

Para cada caso de uso se debe definir:

- Subsistemas y actores que intervienen.
- Información que intercambian.
- Qué perfil de usuario está habilitado en este caso y qué funcionalidad está habilitada.

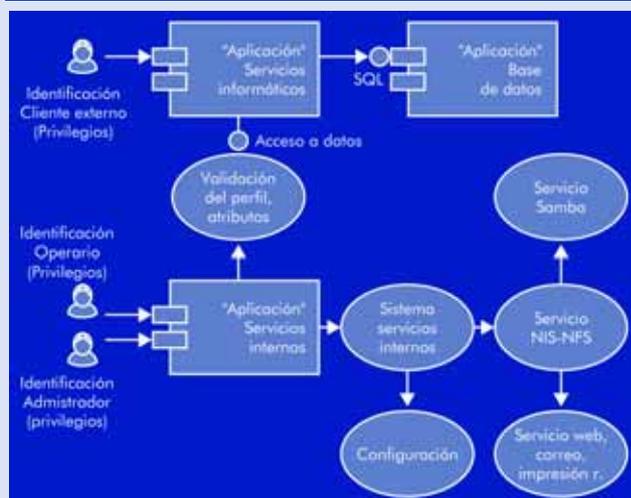
La definición del intercambio de información permitirá verificar y detallar las interfaces de cada subsistema, teniendo en cuenta todos los casos de uso en que intervienen e ir completando, así, la definición de subsistemas realizadas en fases anteriores.

Caso práctico

Diagrama UML de revisión de casos de uso con componentes implicados del subsistema servicios internos.

A continuación se desarrollará la definición de subsistemas realizada con la revisión de casos de uso implicados en servicios internos.

Figura 3-3.



Como complemento de este esquema, se deben definir los mensajes que intervienen, los ficheros a los cuales se tiene acceso para lectura y escritura y bajo qué perfiles ocurren estos cambios, con el fin de detallar cada una de las elipses del diagrama con cada una de las flechas, es decir, cada elemento con sus atributos, métodos, etc.

3.7. Especificaciones de desarrollo y pruebas

A partir de este punto, se estará en condiciones de establecer las características del sistema en los siguientes aspectos:

- Tecnología: hardware, software y comunicaciones.
- Servicios y herramientas de desarrollo: identificación de los servicios y por qué software será prestado, y adecuación del mismo al entorno tecnológico y herramientas que permitirán dar la funcionalidad requerida.
- Herramientas de documentación.
- Seguridad y protección: niveles, separación y control.
- Restricciones técnicas.

Con todo esto hay que definir el conjunto de pruebas necesarias que se deben realizar una vez implantado el sistema. Éstas deben definirse como pruebas unitarias, con el mínimo nivel de dependencia posible para permitir un desarrollo, implantación y prueba por componentes. Por ejemplo, la prueba del NFS. No será necesario probarlo con todo el sistema en primer nivel, se podrán realizar pruebas de montar recursos locales y con sólo una máquina cliente para los diferentes perfiles de uso. Será necesario también definir una prueba global (de integración) para probar más adelante el sistema de NFS a nivel de toda la arquitectura, pero con la certeza de que el servidor funciona y está parcialmente validado.

La especificación de las pruebas unitarias se puede dividir en:

- **Pruebas de caja negra:** se considera el componente o servicio desde el punto de vista funcional, analizando sus entradas y salidas y comparando sus respuestas con los resultados esperados.

- **Pruebas de caja blanca:** se considera el componente o servicio como una estructura con una secuencia lógica de eventos y se comprueba la validez de ésta, los mensajes/eventos incorrectos, etc.

Normalmente, se utilizará una combinación de los dos tipos de pruebas adecuadas para cada servicio/componente. Es recomendable invertir el proceso tradicional de realizar las pruebas de los componentes una vez implementados por pruebas parciales (dejando para el final las pruebas de integración). De esta manera se pueden probar los servicios y validar (parcialmente) su funcionalidad.

En un proyecto de estas características, es importante considerar que el desarrollo de una única prueba puede ser muy complejo y tener un elevado coste debido a la cantidad de requisitos, perfiles y restricciones que se manifiestan. Además, una única prueba puede tener el agravante de dejar interrelaciones o elementos sin probar (dada la complejidad de la misma) o introducir en ésta efectos colaterales, que luego pueden afectar a los resultados obtenidos.

Es importante considerar también que el apartado de pruebas frecuentemente provoca retrasos en el proyecto. Este enfoque de pruebas unitarias (parciales) disminuye este efecto y optimiza el desarrollo, ya que las pruebas se enfocan a obtener resultados (satisfactorios) del subsistema desarrollado o implantado.

Caso práctico

Especificaciones de desarrollo.

En nuestro caso práctico, teniendo en cuenta los servicios internos, las especificaciones de desarrollo serían:

- Servicio NIS: integración de todos los usuarios (excepto el *root*) a través de ordenadores basados en GNU/Linux. Los ordenadores basados en Windows tendrán usuarios locales con acceso al sistema Samba validado en un sistema GNU/Linux.
- Documentación:
http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/NIS-HOWTO.html

- Servicio NFS: nivel 3 con exportación sólo para las máquinas del sistema local sin inclusión de convalidación del usuario *root* en el sistema importado.
- Documentación:
http://www.ibib.io.org/pub/Linux/docs/HOWTO/other-formats/html_single/NFS-HOWTO.html
<http://www.nfsv4.org>
- Servicio de web: servidor Apache-Tomcat y autenticación y seguridad (SSL). Configuración segura de directorios, seguimiento de enlaces y acceso por servidores virtuales.
- Documentación: <http://apache.org/>
<http://www.php.net/> <http://www.postgresql.org/>
- Servicio de red: gestión de IP a través de DHCP, intranet basada en direcciones privadas con acceso exterior a través de NAT, servicio de DNS local, servicios seguros a través de Xinetd, SSH, e Iptables.
- Documentación:
<http://tldp.org/HOWTO/HOWTO-INDEX/howtos.html>
- Servicio de monitorización: basado esencialmente entre Nmap, Nessus, Snort, Logcheck y Tripwire.
- Documentación:<http://www.insecure.org/nmap/>
<http://www.nessus.org/>
<http://www.snort.org/>
<http://sourceforge.net/projects/logcheck/>
<http://www.tripwire.org/>
- Servicio de configuración: basado en Webmin y Linuconf.
- Documentación: <http://www.webmin.com/>
<http://sourceforge.net/projects/linuxconf/>
- Servicio de impresión remota: a través de servicio de impresión CUPS con validación de usuario. Control local de recursos y contabilidad de utilización a través de la base de datos.

- Documentación: <http://www.cups.org>
- Servicio de correo electrónico: servicio basado en Exim con servidor seguro de IMAP. Control de cuotas de correo e integración con un servicio de web-mail basado en Squirrel.
- Documentación: <http://www.exim.org>
<http://www.washington.edu/imap/>
<http://www.squirrelmail.org/>
- Servicio de ofimática: acceso a recursos compartidos basados en OpenOffice para edición de texto, hojas de cálculo, dibujo o presentaciones.
- Documentación: <http://www.openoffice.org/>
- Instalación de aplicaciones del cliente: en este caso, se instalarán las aplicaciones existentes del cliente (contabilidad, gestión y control) validando su funcionalidad.

Se deberán desarrollar las interfaces y las páginas web que permitan acceder a los recursos integrándolas todas bajo un aspecto corporativo y unificado de la empresa. La gestión y control se deberá sintonizar para los administradores futuros en forma de reducir la complejidad (por ejemplo, instalando herramientas de configuración y/o gestión gráficas –o creando las interfaces web adecuadas– para las tareas más rutinarias).

Para la codificación de páginas web, se utilizará PHP con validación local por JavaScript en el caso de los formularios, y en el lado del servidor, si es necesario, se utilizará Perl (www.perl.org) como lenguaje de soporte para las tareas de mayor complejidad.

A continuación se deberán enumerar las pruebas unitarias, extraídas de las funcionalidades e interfaces del sistema:

- Conexión, desconexión de los usuarios en función de los servicios/perfiles.

- Configuración automática de la red (DHCP), control de acceso a Internet por NAT y verificación de acceso a servicios desde Xinetd.
- Gestión, errores, accesos no permitidos y log de los servicios web.
- Acceso a los recursos con los criterios de seguridad especificados (NFS-NIS).
- Acceso a la base de datos (lectura, modificación) de acuerdo al perfil del usuario y lectura/escritura de diferentes elementos de la misma.
- Acceso a sistema local (intranet) desde máquinas no habilitadas o desde Internet.
- Acceso al sistema de correo y verificación de envío-recepción.
- Análisis de seguridad (Nessus, Nmap, Snort, Tripwire y Logcheck).
- Control de transacciones de todos los servicios antes mencionados.
- Control de integridad y seguridad de los datos intercambiados entre cada servicio.
- Control de configuración de los servicios (WebAdmin).
- Acceso a sistema de compartición de ficheros sobre máquinas Windows con criterios de seguridad y control.
- Acceso a sistema de impresión (CUPS) de acuerdo a los criterios de seguridad y control de acceso.

Para cada una de estas pruebas, se deben definir los parámetros o información de entrada y sus posibles resultados o información de salida. Esto permitirá al grupo de desarrollo programar cada una de las pruebas particulares elegidas dentro del marco de trabajo especificado.

3.8. Requisitos de implantación

Los requisitos de implantación serán los que se deberán cumplir para cada componente o subsistema cuando se trabaje en el entorno real conjuntamente con el resto de subsistemas. Por entorno no se considerará solamente el tecnológico, sino que se tendrá en cuenta a los usuarios (de todos los tipos) del subsistema. Esto permitirá definir el plan de formación adecuado a los diferentes niveles para que los usuarios finales del subsistema puedan interactuar y obtener las prestaciones para las cuales fue definido el servicio.

Desde el punto de vista tecnológico, se deben determinar las condiciones del entorno donde se implantará el nuevo subsistema y las condiciones de funcionamiento para determinar que este subsistema funcionará a pleno rendimiento sin agotar los recursos y con las condiciones de seguridad adecuadas, y sin afectar a los demás subsistemas con los cuales interrelaciona.

El documento que recoja los requisitos de implantación deberá contemplar además:

- Gestión de la documentación: quién y cómo se tendrá acceso a ella y bajo qué condiciones.
- Necesidades de migración de servicios/usuarios.
- Formación de los usuarios, administradores y clientes para las nuevas herramientas/servicios.
- Necesidades del hardware y software básico.
- Necesidades de comunicación.
- Niveles de seguridad y control de implantación.
- Recomendaciones en casos de contingencia, emergencia y procedimientos de recuperación.
- Recomendaciones de integración global.
- Recomendaciones de análisis de utilización de recursos y posibilidades de ampliación.

Este documento se tendrá en cuenta en la fase de implantación y pueden figurar en él cuestiones adicionales tales como procedimientos de seguridad adicionales, pruebas locales y desde Internet, procedimientos de bloqueo o también cuestiones vinculadas a intentos de acceso no permitidos a datos y/o recursos.

Caso práctico

Requisitos de implantación para el sistema de servicios internos

La implantación deberá ser realizada bajo dos puntos de vista: los usuarios del subsistema (todos los tipos existentes) y el tecnológico y sus recursos.

Desde el punto de vista del **usuario**:

- Definición del responsable de seguridad y administración del subsistema de servicios internos. Esta figura tendrá como responsabilidades:
 - Conocer el funcionamiento del subsistema teniendo acceso a toda la documentación.
 - Definir las políticas tanto de altas, bajas, como de bloqueo de usuarios-recursos y servicios.
 - Comprobar y monitorizar el correcto funcionamiento, así como verificar las políticas de seguridad con un plan previamente especificado y consensuado.
 - Analizar los riesgos y tener planes de contingencia para cada uno de ellos, así como las actuaciones que cabe realizar por parte de los equipos de trabajo.
- Dada la vital importancia de este subsistema para la empresa, es importante contar con un grupo de trabajo para cubrir todos los turnos y definir si los servicios quedarán en funcionamiento permanente o si sólo se podrá acceder a ellos en franjas horarias predeterminadas.

- Para el resto de los usuarios, se describirán las tareas de acuerdo a cada perfil y se determinará qué es lo que pueden hacer y a dónde pueden acceder para realizar la actividad diaria a la que habilita su perfil. Además, se deben especificar los canales de comunicación para el envío de incidencias y cómo hacer la comunicación de errores o las peticiones de actividades-acciones necesarias no contempladas en su perfil actual.

Desde el punto de vista **tecnológico**, la implantación del subsistema servicios internos tendrá un gran impacto sobre la forma de trabajo de la empresa, tanto a nivel de recursos, como a nivel de adaptación/formación del personal existente. Es importante tener en cuenta esta situación para aprovechar, desde el primer momento, y vencer las reticencias iniciales de los usuarios, con la finalidad de que adquieran confianza con el nuevo sistema. Se recomienda realizar sesiones informativas preparatorias para evitar el recelo con el cual una persona se adapta a un nuevo entorno tecnológico, cambiando sus hábitos en forma de colaboración y no de imposición.

Se deberán contemplar una serie de cuestiones como prerrequisitos tecnológicos de implantación que mejorarán los aspectos mencionados:

- El hardware debe funcionar de forma transparente para los usuarios locales y con buenos índices de prestaciones mejorando el rendimiento actual por lo menos en un factor de dos (recomendable 5).
- Los usuarios deben estar dados de alta con anterioridad para que puedan probar los nuevos sistemas e irse habituando a éstos (siempre que sea posible por cuestiones de espacio o reutilización del hardware).
- Las cuentas de correo deben estar migradas y durante un tiempo mantener la duplicidad del mismo.
- Los archivos deben montarse de modo que el usuario ubique sus archivos de manera análoga al sistema anterior.

- Las páginas web deben ser totalmente funcionales y operativas, así como las interfaces de usuarios, a los servicios y las aplicaciones utilizadas por los usuarios de la empresa.
- Debe estar probada y verificada la interrelación con los demás subsistemas desde el punto de vista de las comunicaciones y la coherencia de los datos.

En este apartado se deberán incluir todas las cuestiones relativas a la implantación del subsistema en cuestión para evitar situaciones que luego en las pruebas de integración puedan generar retrasos o modificaciones por falta de compatibilidad en la interacción de los subsistemas. Es importante tener en cuenta una secuenciación de implantación, ya que se deberá disponer de otros subsistemas implantados con anterioridad para que el actual pueda funcionar totalmente.

4. Desarrollo

La construcción ordenada del sistema del cual se ha evaluado la viabilidad, se ha hecho, analizado y diseñado, es el objetivo principal de esta fase. En las metodologías tradicionales, el desarrollo se produce cuando se han superado satisfactoriamente las fases anteriores, pero métodos más recientes aconsejan iniciar esta fase cuando sea posible, ya que permite una mayor agilidad en el ciclo de vida del proyecto permitiendo disponer de prototipos para ser evaluados antes que con la metodología clásica. Algunos autores denominan a esta metodología *evolutiva* y es aconsejable, si el jefe de proyecto lo considera conveniente, empezarla en forma concurrente con la fase de análisis.

No obstante, siempre es necesario adaptar las metodologías a las necesidades del proyecto, utilizando submodelos (de entre otros más extensos) que se adapten mejor a la realidad que el que se está desarrollando.

Generalmente, es muy productivo que parte de los desarrolladores se involucren en el diseño participando por ejemplo en el desarrollo de un subsistema y en el diseño de otro. Esto permitirá una alta motivación y una gran calidad en el producto software desarrollado, siendo capaces de detectar, además, necesidades de implantación o errores de diseño que en otro caso habrían sido pospuestos hasta la primera revisión con los problemas añadidos que ello implica (retrasos, modificación de otros subsistemas interconectados, etc.).

También se debe prestar atención para que esta dualidad diseñador-desarrollador sea analizada por el jefe de proyectos con detalle, ya que es necesario que las personas tengan disponibilidad y puedan aportar en ambos sentidos. Muchos fracasos en grandes proyectos provienen de que todos los miembros del equipo están en todas las fases, lo cual provoca vicios en el razonamiento, diseño o desarrollo. Un ambiente de trabajo basado en la crítica constructiva es el am-

biente ideal para que un proyecto salga adelante en un grupo organizado de trabajo.

Independientemente de estas cuestiones, existe un conjunto de aspectos que conciernen al desarrollo y que deben ser definidas al inicio de la presente fase. Se debe planificar el inicio y fin del desarrollo sincronizando las diferentes actividades del proyecto que se deberán implantar al final de la fase (incluyendo no sólo el propio desarrollo, sino la instalación, las pruebas, la documentación, los planes de formación, etc.)

4.1. Planificación de las actividades de integración del sistema

Al llegar a este punto, se dispone de información sobre qué necesita ser desarrollado, qué componentes software-hardware se utilizarán y cómo se integrarán en el sistema actual, qué herramientas/servicios se utilizarán, en qué entorno, etc.

Las actividades de desarrollo que permitirán llegar al objetivo planteado serán:

- Concretar las versiones del software o componentes software (incluido librerías, módulos, clases, etc.) que se utilizarán.
- Estudiar dichos componentes y realizar su implantación en el entorno de desarrollo.
- Desarrollar las pruebas unitarias.
- Desarrollar/instalar/configurar los componentes necesarios.
- Realizar la documentación.
- Planificar la formación a los usuarios del sistema.
- Realizar las pruebas de integración.
- Validar y aprobar el sistema.

El objetivo final de la presente fase es la aprobación del sistema para que pueda ser implantado. En muchos casos (como el ejemplo tra-

tado aquí) el desarrollo a veces se realiza sobre el mismo hardware que luego se pondrá en producción en la fase de implantación. Esta forma de trabajo es adecuada cuando son proyectos nuevos que se realizan dentro de la empresa en la cual será implantado.

Las actividades deben planificarse para seguir un secuencia de pasos que conlleven a la finalización del proyecto en forma ordenada hasta su aprobación (fase de implantación, apartado de aceptación del sistema). El orden de las actividades no tiene por qué ser secuencial, y su grado de simultaneidad estará fijado por el número de desarrolladores, su perfil y la diversidad de recursos con los cuales se cuente. Se debe tener en cuenta que existen actividades que son perfectamente concurrentes (por ejemplo, desarrollo de un subsistema y la documentación del mismo) mientras que existen otro tipo de situaciones en las que es mejor una secuencia, ya que hay relaciones de precedencia entre ellas. Estas relaciones pueden ser complejas y la experiencia del jefe de proyecto permitirá encontrar el camino óptimo para su resolución.

Caso práctico

Planificación de adecuación y renovación de red y servicios informáticos para la empresa NtEum S.A.

Un método adecuado para realizar este tipo de planificación es a través de Diagramas de Gantt. Normalmente, se procede incluyendo una duración estimada de las tareas y la vinculación entre ellas, marcando los diferentes hitos poniendo de manifiesto la sincronización y la secuenciación entre las diferentes actividades.

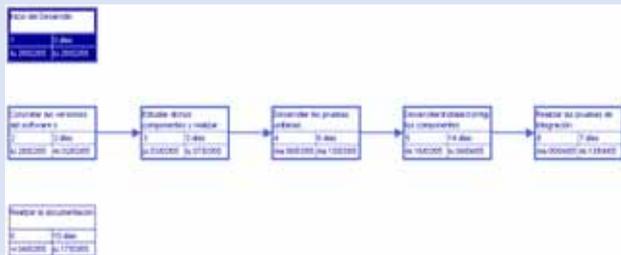
Este diagrama puede ser revisado a medida que avance el proyecto para incluir las diferencias con la realidad haciendo versiones del mismo, las cuales servirán, una vez haya finalizado el proyecto, para analizar las problemáticas que han impedido cumplir el primero de ellos y extraer experiencia para futuros proyectos. (<http://www.imendio.com/projects/planner/>).

A continuación se muestra un ejemplo de tres gráficos obtenidos con una herramienta de planificación de proyectos que son el diagrama de Gantt, el diagrama PERT (*program evaluation and review technique, diagrama orientado al plazo de ejecución*) y el calendario.

Figura 4-1. Diagrama de Grantt



Figura 4-2. Diagrama de PERT



En estos gráficos también se puede incorporar el seguimiento de los recursos, de manera tal que se pueda controlar que no hay sobreasignaciones o problemas derivados de la planificación sobre los recursos (humanos y físicos).

Estos diagramas permitirán a cada uno de los actores ver la secuencia de hitos y tomar las decisiones oportunas para lograr el objetivo final del proyecto. Estos gráficos se pueden complementar con un documento (plantilla) que refleje el estado del mismo de forma periódica:

Figura 4-3. Calendario



Tabla 4-1

Adecuación y renovación de red y servicios informáticos para la empresa NtEum S. A.	
Fecha del reporte	dd/mm/aaaa
Fecha de finalización:	dd/mm/aaaa Estimado original dd/mm/aaaa Estimada actual
Cambio desde el último reporte	+/- dd días
Elementos pendientes de desarrollo	n Defectos, m funcionalidades
Elementos pendientes de aprobación	0 Defectos, j funcionalidades
Elementos cerrados	l funcionalidades
Recursos usados durante este período	Persona1: j horas Persona2: i horas Persona3: k horas
Resumen del estado	El proyecto está avanzado según lo planificado
Documentos relacionados	Plan de proyecto Diseño de módulos
Estado detallado	
El desarrollo de esta semana se ha concentrado en la tarea X...	
Se ha realizado aproximadamente el 30% del proyecto y se llevan dos días de retraso respecto a la fecha prevista...	
Control de riesgos	Problemas con las versiones de la librería XX utilizada en la identificación de los usuarios Verificar subsistemas
Actividades programadas	Actividad ... Evento ... Acciones ...
Actualización de la planificación	Se puede realizar una copia del diagrama de Gantt de la semana

El jefe de proyecto deberá cumplimentar para cada tabla los valores relacionados con las fechas (dd/mm/aa) y los valores particulares de cada fila de la tabla (i,j,k,n,m,X,XX,etc.).

4.2. Cómo elegir la licencia más adecuada

Tanto si se trata de un proyecto interno como de un proyecto con fines comerciales se debe escoger la licencia de los componentes o de los módulos/partes que se van a utilizar/desarrollar al comienzo del mismo. Más aún cuando se integran componentes software o servicios de diferentes desarrolladores, ya que su licencia puede condicionar la licencia final del proyecto actual.

La licencia escogida tendrá repercusiones sobre los ficheros de código fuente de nuestro desarrollo haciendo mención de las partes que son propiedad de otros desarrolladores. También deberá mencionarse en la documentación y los materiales de formación teniendo en cuenta que en proyectos *open-source* generalmente deben publicarse todas las licencias.

Si el proyecto se comercializa bajo una licencia propia, debe asegurarse que los componentes, herramientas y el código resultante lo permitan para que esto no implique problemas legales al cliente final, el cual deberá ser informado de los derechos y deberes respecto a la licencia del producto que acaba de adquirir. La licencia también afecta a los planes de mantenimiento y soporte del producto y deben ser tenidos en cuenta a la hora de escogerla.

El jefe del proyecto y el director responsable deben concentrarse en las incompatibilidades de los distintos modelos de licencia existente especificando el escenario concreto en el cual se desarrollará el software del mismo identificando las posibles alternativas: código propietario y licencia comercial, código parcialmente propietario y licencia comercial (implica tener licencias de explotación), código libre y licencia comercial, código libre y licencia libre. Deberán además resolverse preguntas relacionadas con los derechos de autor, la garantía y el soporte leyendo detenidamente las especificaciones de cada licencia en cada caso.

Caso práctico

Elección de las licencias de desarrollo

En nuestro caso será un desarrollo basado en *open-source* de utilización interna, por lo cual el modelo de licencia no tendrá efectos sobre el modelo de negocio de la compañía. Esto, sin embargo, no evita que no se deban poner las referencias adecuadas en el código desarrollado ni en la documentación. Las posibles alternativas son:

- Licencia propietaria: si no se va a distribuir el sistema, se pueden desarrollar bajo licencia propietaria.
- Licencia BSD: permite mantener los derechos de autor (*copyright*) sobre el desarrollo y es coherente con las licencias del resto de los componentes (las cuales se basan en BSD y GPL). Esta licencia NO obliga a distribuir el código fuente final y permite al destinatario de este software su utilización, copia, modificación, redistribución y venta. También permite la integración con otro producto de licencia propietaria.
- Licencia GPL: permite mantener los derechos de autor sobre el desarrollo y es coherente con el resto de licencias de nuestro proyecto (BSD y GPL). Pero a diferencia de BSD, obliga a distribuir el código fuente e impide su futura comercialización bajo licencia propietaria.
- Licencia LGPL: es prácticamente igual a la GPL, pero permite que software con esta licencia sea integrado en software propietario. Un ejemplo clásico es la librería C de Linux (con licencia LGPL).
- Otras licencias para código abierto: Apache (Basada en BSD), Mozilla (MPL y NPL de Netscape).

Dadas las características de nuestro proyecto, nos decantaremos por una licencia LGPL, ya que el proyecto es más de integración que de desarrollo y satisface las expectativas de utilización del cliente.

4.3. Entorno de desarrollo

El objetivo de esta tarea es transportar el software desarrollado/integrado al equipo de producción, donde deberá prestar el servicio. Esta tarea implicará más o menos dificultades si el equipo de desarrollo difiere del equipo de implantación o no, y esto se deberá tener en cuenta.

Caso práctico

En nuestro caso particular, como se trata de un sistema nuevo, los equipos de desarrollo e implantación coinciden, por lo cual, una vez finalizado el desarrollo, la puesta en producción será casi inmediata.

En un amplio sentido y entrando en detalle sobre las actividades descritas en la planificación del desarrollo, se debe tener en cuenta:

- Preparación del entorno de generación y desarrollo
- Generación del código de los componentes
- Instalación de los servicios especificados
- Ejecución de pruebas unitarias
- Ejecución de pruebas de integración

Siguiendo las recomendaciones de las fases de análisis y diseño, el seguimiento y la asistencia al desarrollo del código facilitará al grupo de trabajo su consecución con éxito y en los tiempos previstos. Debe tenerse en cuenta que las herramientas tipo RAD (*rapid application design*) facilitan enormemente el trabajo de los desarrolladores, pero que se necesita un período de aprendizaje para que su utilización sea eficiente.

Si los desarrolladores han participado en el diseño, el tiempo de estudio y distribución de las tareas será fácil y eficiente, reduciéndose el tiempo necesario de preparación e imprimiendo un ritmo mayor en el desarrollo del proyecto.

En caso contrario, si las fases de diseño/desarrollo han estado desvinculadas, hasta que no finalice la etapa de diseño no podrá co-

menzar la de desarrollo y se necesitará un tiempo (*startup*) para que los desarrolladores analicen los requisitos y realicen un estudio de los mismos. Si se solapan parcialmente las fases de diseño y desarrollo, se puede caer en situaciones no deseadas interrumpiendo, por ejemplo, el desarrollo de un subsistema porque de otro, con el cual se vincula todavía, no ha sido aprobado su diseño. Además, esta táctica presenta un riesgo añadido dadas las constantes revisiones que se deberán hacer del código desarrollado cuando se deba integrar globalmente.

Los resultados de las pruebas unitarias diseñadas y desarrolladas anteriormente son el indicador más relevante del ritmo del proyecto y permitirá conocer en todo momento si la planificación es la adecuada o si se desvía de las previsiones iniciales o reflejadas en la planificación.

4.4. Documentación

Una vez llegados a esta etapa, es necesaria la elaboración de la documentación de usuario, tanto de administración como de explotación. Teniendo en cuenta las decisiones de fases anteriores respecto al formato, herramientas de generación y disponibilidad, debe desarrollarse su estructura y contenido.

Es importante mencionar que esta tarea debe ser realizada en función de la planificación (diagrama de Gantt) y que no necesariamente tiene que ir al final del desarrollo, si bien es cierto que deberá ser completada al final de esta etapa (en algunas empresas la documentación se rellena después de haber realizado un prototipo sintético del código pero antes de escribir el código final).

Es importante que este tipo de recurso, que luego será utilizado por diferentes usuarios de perfiles muy variados, sea consensuado para encontrar el estilo y profundidad adecuada al lector. Suele pasar que la documentación no técnica y escrita por personal técnico sólo es inteligible para personal con formación técnica, por lo cual, el jefe de proyecto deberá seguir de cerca el proceso de documentación para que este proceso tenga todos los requisitos de calidad necesarios.

Caso práctico

En el presente proyecto se utilizarán los formatos adecuados (DocBook) y dadas las características del proyecto, la mayor parte de la documentación será la correspondiente a los servicios integrados que ya poseen una documentación adecuada. Mediante el formato elegido, se procederá a integrar toda esta documentación (transcribiéndola si fuera necesario) para darle un aspecto uniformador y coherente. Dentro de la documentación técnica, ésta se deberá complementar con las estructuras funcionales y de relación del código desarrollado, así como de los diagramas adecuados al tipo de proyecto en el cual se está trabajando (clases, métodos, objetos, eventos, mensajes, etc.).

En el caso de integración de documentación externa, hay que citar la fuente de donde se ha extraído y la versión de la misma para evitar diferencias o situaciones de discrepancia que afecten o bien al software o bien a la documentación.

5. Implantación

La implantación es la fase en la cual se traspasa el programa en desarrollo a producción, bajo usuarios y conexión reales en el entorno que operará. Esta fase generalmente es crítica en la vida del proyecto y es una etapa de nerviosismo y de alta presión psicológica. Por lo tanto, es importante mantener la calma durante este período y planificar cuidadosamente el cambio.

Asimismo, cabe destacar que existirá una cierta reticencia al cambio por parte de los usuarios nuevos o que tengan deficiencias en la formación y que el jefe de proyectos deberá preparar el cambio con los jefes de área haciendo sesiones informativas y contestando a todas las preguntas de los usuarios para ir preparando el camino hacia el cambio.

En los casos en los que el sistema se sustituya por otro similar, el grupo de desarrollo se encontrará con un ambiente receptivo o un tanto indiferente en función de la medida de los problemas que solucione el nuevo sistema. En cambio, si el cambio (como es nuestro caso) es radical, ya que modifica la forma de funcionamiento del sistema en su totalidad, la lista de incidencias será distinta y de mayor longitud (generalmente).

En todo caso, es en esta fase donde se debe implicar a los usuarios participantes en los casos de uso analizados del sistema y enseñarles mediante la formación adecuada sus nuevas responsabilidades o tareas.

También es importante establecer los criterios de calidad de servicio bajo los cuales el sistema debe funcionar, es decir, los servicios mínimos bajo los cuales el sistema se considera totalmente funcional, lo cual se deberá redactar de acuerdo a las especificaciones funcionales iniciales. Obviamente, después de la implantación (o en un

breve período posterior acordado de antemano) el sistema implantado deberá cumplirlos.

Es importante tener en cuenta si la implantación es total (nuevos servicios, nueva funcionalidad) o si el sistema se debe integrar con otro ya existente y trabajar conjuntamente con él. En el primer caso no existirán demasiados problemas en su instalación/puesta en funcionamiento, mientras que en el último caso, debe darse garantía de que la implantación no afectará al funcionamiento de los servicios que existen, los cuales deberán continuar funcionando a pleno rendimiento.

Caso práctico

Planificación de la implantación del subsistema servicios internos

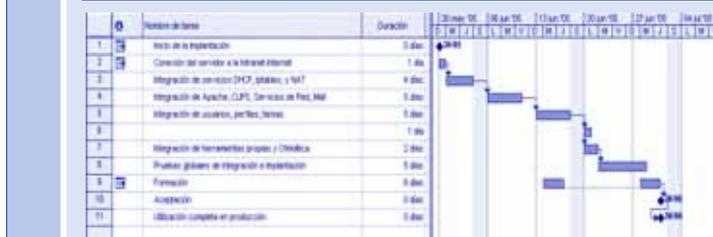
El jefe de proyecto, en este caso, deberá diseñar un calendario de implantación que será consensuado tanto con los desarrolladores como con los usuarios de casos de uso. Para ello se pueden utilizar diagramas y herramientas como los que se han utilizado en otras fases de proyecto (por ejemplo, en la fase de desarrollo).

Como se ha mencionado anteriormente, la etapa de implantación requiere la dedicación total por parte de los equipos de trabajo, usuarios del propio subsistema y de otros subsistemas con el cual el presente interrelaciona. En estos casos, es importante disponer (si el tamaño de proyecto lo justifica) de herramientas de trabajo colaborativo (Groupware) de comunicación y registro de incidencias. También se pueden utilizar métodos más tradicionales como hojas de incidencias o registros para ir centralizando todos los problemas y/o incidencias que se vayan produciendo durante la puesta en marcha.

Independientemente del método utilizado, deberá haber un equipo (generalmente del equipo de diseño-desarrollo) de soporte que dé respuestas a estas incidencias y solucione los problemas que surjen bajo la ejecución real.

Un ejemplo de un diagrama de Gantt con la planificación de la implantación podría ser el siguiente:

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



5.1. Formación

La comunicación es un elemento de vital importancia en la aceptación del proyecto y en su correcta utilización por parte de los usuarios, por esta razón, el jefe de proyecto deberá realizar sesiones informativas (o delegar en personas de su equipo con la formación adecuada) con los futuros usuarios a lo largo del mismo. Una vez que los usuarios conozcan los objetivos del proyecto, su diseño y en algunos casos su desarrollo, estas personas deberán recibir formación para que puedan trabajar eficientemente con el sistema desarrollado y donde también se les explique la potencialidad del mismo, sus deberes y obligaciones con respecto al nuevo sistema.

Si bien en sí misma la formación no suele formar parte de los proyectos en el ámbito de las tecnologías de la información y la comunicación (TIC), es importante considerarlo como un elemento determinante en la aceptación del producto final.

En este caso, para evitar tópicos muy conocidos, hay que determinar diferentes perfiles de usuario que necesitarán ser formados sobre los diversos aspectos del sistema que se está implantando. En función de las obligaciones y responsabilidades, la formación puede ir desde la utilización de metodologías no-presenciales hasta la formación presencial con evaluación y análisis de casos prácticos.

Para realizar los planes de formación y desarrollar su contenido, se deberá comenzar por los estudios de casos de uso (utilizados en las fases de análisis y diseño) y ampliarlos en función de las necesidades de la empresa y las características del proyecto.

5.2. Implantación del sistema, pruebas y nivel de servicio

Las pruebas de implantación son aquellas que se realizan sobre el sistema funcional final. Generalmente, difieren de las unitarias y de las de integración, que se realizan no sobre las máquinas de desarrollo, sino sobre las del cliente. En nuestro caso particular, como será equipamiento nuevo, las pruebas de integración se realizarán sobre el mismo hardware donde se han realizado las pruebas restantes. No obstante, esto se dará cuando puedan determinarse los errores de conectividad, seguridad, etc., es decir, cuando el sistema trabaje en el “mundo real” y no en el departamento de desarrollo.

Es importante asegurar la disponibilidad de los recursos actuales para minimizar el impacto de “parar” la empresa o servicios de la misma durante el menor espacio de tiempo posible. Normalmente estos cambios-pruebas se realizan en días no laborables para evitar molestias a los clientes, pero implica que todas las personas vinculadas deben estar presentes como si de un día normal se tratara.

Caso práctico

En resumen, en nuestro caso y en esta actividad se debe:

- Integrar todos los subsistemas en el entorno real, modificando las IP, NAT y servicios de comunicación para adecuarlos al entorno de trabajo (no al de desarrollo).
- Integrar todo el hardware-software existente con los nuevos subsistemas y reencaminar las peticiones de datos-servicios hacia los nuevos subsistemas.
- Verificar la integración de usuarios, servicios, subsistemas a nivel de seguridad y autenticación (se debe recordar que esto ya se ha realizado en pruebas unitarias, pero no con todo el hardware/usuarios de la empresa).

- Integrar los nuevos servicios (por ejemplo, web, correo electrónico) con el sistema real sin que haya interferencias con los existentes. Esto dependerá de la política de la empresa si el cambio se realiza de forma abrupta (reemplazo directo de servicios) o de forma gradual (coexistencia de servicios durante un tiempo prudencial).
- Traspaso de los datos al nuevo gestor de datos (o actualización si ya se han hecho traspasos parciales) con validación de los datos y verificación de las políticas de seguridad.
- Configuración de los servicios externos (direcciones, IP, etc.) para que apunten al nuevo sistema.

Una vez comprobada la correcta instalación de todos los subsistemas, de las políticas de seguridad, de la accesibilidad y de la conectividad, se lanzarán los procesos automáticos (en *background*) teniendo en cuenta que se deberá realizar una monitorización periódica para ver su evolución.

Después de las pruebas de implantación de los primeros resultados estables (y se hayan corregido las deficiencias-errores que se han producido en los subsistemas), se deberá evaluar el comportamiento de éstos para obtener medidas de prestaciones o eficiencia que permitirán, comparándolas con los requisitos iniciales, establecer criterios de cumplimiento o no del sistema desarrollado (nivel de servicio).

En caso de no llegar a unos mínimos resultados, se deberán efectuar los cambios necesarios en el desarrollo (o también podría ser en el diseño) de los subsistemas que afecten a estos resultados. Esta tarea es complicada, ya que se debe volver atrás (aún peor si se debe volver al diseño) e implicará retrasos en el pleno funcionamiento de la aplicación.

En este caso, el jefe de proyectos, junto con los directivos de la empresa, deben, en primer lugar, tomar una determinación sobre si el sistema sigue en funcionamiento y si se trabaja en paralelo para solucionar estos

En este caso, el jefe de proyectos, junto con los directivos de la empresa, deben, en primer lugar, tomar una determinación sobre si el sistema sigue en funcionamiento y si se trabaja en paralelo para solucionar estos problemas o se quita “de línea” al sistema y se deja su integración para cuando los problemas estén solventados. En segundo lugar, el jefe de proyectos deberá analizar dónde se han producido los problemas y en qué fase se han cometido los errores para que esta situación no se repita en un futuro (es una buena práctica dejar constancia documental sobre los errores o desviaciones para este fin). Este tipo de retraso/problemas genera normalmente pérdidas económicas a la empresa, por lo cual, si el equipo de proyecto es externo, se le pedirán responsabilidades – económicas, legales, etc.– mientras que si es interno, se deberá analizar donde se han dado los problemas para depurar responsabilidades.

Cada caso es particular, pero generalmente se debe dar una prioridad a cada uno de los problemas-errores para que puedan ser analizados dentro de contexto. Esto permitirá tener información de dónde está el problema y cuánto tiempo se tardará en resolverlo. Por ejemplo, un problema de autenticación generará una serie de problemas derivados, que no son tales, sino causa del primero, el cual puede ser simplemente porque el usuario que se conecta a través de un servicio NIS no ha sido dado de alta correctamente en el servidor o porque el mismo usuario ha cambiado su palabra clave de acceso y todavía no se han refrescado las memorias caché de las máquinas clientes. Esta situación no significa que el subsistema no funcione, sino que se debe determinar la causa y encontrar cuál es el error inicial y cuáles son los errores correlacionados (es decir, que dependen de otro).

Generalmente, el jefe de proyecto con su equipo deberá realizar una serie de medidas tales como utilización de los recursos, tiempo de respuesta, tiempo de retorno, ancho de banda, etc., que permitan identificar el nivel de servicio que se está obteniendo del sistema para realizar previsiones de funcionamiento cuando el sistema se encuentre a plena carga.

5.3. Aceptación del sistema

Esta actividad consiste en presentar a los responsables o a la dirección de la empresa toda la documentación relativa a la implantación de proyecto, incluyendo los resultados desde las pruebas unitarias, de integración y de implantación a los niveles de servicio obtenidos para su aprobación.

Esta actividad es muy importante para la empresa si el equipo de proyecto es externo porque implica que se acepta el proyecto tal y como está, teniendo en cuenta (generalmente regido por un contrato) que pasará con las incidencias que puedan surgir. Una vez aceptado el sistema, el equipo de proyecto podrá considerar que el proyecto ha finalizado y que posteriores modificaciones o añadidos serán considerados como tal y generalmente irán aparte del coste económico del anterior (no ocurrirá así si son deficiencias o faltas en el desarrollo inicial y que no han sido detectadas). La situación es mucho más sencilla desde este punto de vista si el equipo de proyecto es interno a la empresa, aunque se debe estipular muy bien los canales de comunicación para la corrección de errores o problemas, ya que si no el equipo de proyecto siempre vivirá pendiente (por diversas peticiones de los usuarios) del sistema desarrollado.

5.4. Mantenimiento

Esta fase debe ser planificada durante el resto de las fases de proyecto, puesto que una gran parte de la información incluida en la documentación del proyecto debe ir orientada al mantenimiento del mismo.

Existen diferentes opciones en cuanto al mantenimiento en función del tipo de instalación de que se disponga: mantenimiento hardware, de comunicaciones, de infraestructura (aire acondicionado, por ejemplo), mantenimiento software, etc., las cuales se integrarán dentro de la política de la empresa al respecto. Existen situaciones particulares, como por ejemplo, ¿el mantenimiento hardware lo realiza la misma empresa o es un servicio externo?

En nuestro caso consideramos que todo el software implantado es código abierto (*open source*) por lo cual, se puede optar por mante-

nimiento interno (con el equipo adecuado) o subcontratar a una empresa externa (para el software integrado propietario se seguirá en este caso con la política anterior definida por la empresa). En el primer caso pasa por contratar a un técnico de soporte especializado o formar a personal interno para este trabajo, ya que se dispone de todo el código y documentación necesaria para ello. La formación podría estar a cargo del equipo de proyecto u obtenerse en el mercado como formación estándar (si bien es recomendable por su eficiencia el primer caso).

En el caso de que se opte por una empresa externa de mantenimiento, será importante que la dirección de la empresa, junto con el equipo de proyecto, redacten un pliego de condiciones de mantenimiento para que no haya problemas en la externalización (*outsourcing*) de este servicio y que se realice de acuerdo a los criterios de quienes han desarrollado e implantado el proyecto.

Resumen

Durante el desarrollo de este material se han descrito con detalle todas las fases de las que consta un proyecto teniendo en cuenta que, de acuerdo a la tipología del mismo, algunas pueden ser diferentes o muy cortas, pero que en su gran mayoría aparecen en todos los proyectos del ámbito de las TIC.

Estas fases, dependiendo de la metodología empleada (más clásicas o más dinámicas –por ejemplo las evolutivas–), pueden estar presentes en mayor o menor grado, hecho que deberá tenerse en cuenta a la hora de escoger la metodología de trabajo, pero, generalmente, las que siempre aparecen en proyectos de estas características son:

- **Estudio de viabilidad:** en esta fase se considera si el proyecto se puede realizar, teniendo en cuenta las circunstancias internas y externas, las diferentes soluciones posibles y los recursos de los cuales se dispone.
- **Análisis:** se analizan las necesidades que se desea satisfacer con el nuevo proyecto, se ajustan los objetivos finales y se centra la solución tecnológica. También en esta fase se definen las interfaces entre los diferentes subsistemas que formarán el proyecto y las de usuario que permitirán interactuar con el sistema.
- **Diseño:** se realizan en esta fase el diseño tecnológico de la solución escogida, proponiendo una arquitectura global y analizando y estudiando todos los casos de usos (o los más representativos) existentes.
- **Desarrollo:** se construye en esta fase la solución propuesta teniendo en cuenta el entorno utilizado, se escogen las licencias, se genera la documentación y se ejecutan las pruebas acordes al tipo de proyecto y metodología utilizada.
- **Implantación:** se traspa del entorno de desarrollo al sistema real y se realizan todas las pruebas y medidas de niveles de pres-

taciones que conducirán a la aceptación definitiva de proyecto. Se define también el plan de mantenimiento y se toman las decisiones adecuadas para el correcto funcionamiento del mismo durante el resto de vida de éste.

En el caso particular de un proyecto basado en software libre, se debe considerar en cada una de las fases anteriores aspectos diferenciales con otros proyectos basados en software propietario como pueden ser la adecuación de una solución existente a nuestros propósitos, viabilidad, arquitectura, licencias, etc., que normalmente no aparecen en el otro caso (por ejemplo, si se desarrolla un mismo proyecto con tecnología .NET, las licencias y la arquitectura son únicas y prefijadas por la tecnología).

Bibliografía

AgileAlliance y Extreme Programming: herramientas para el desarrollo de software. <http://www.agilealliance.org/> <http://www.extremeprogramming.org/>

Dawson, Christian W.; Martín, Gregorio (2002). *El proyecto de Fin de Carrera en Ingeniería Informática. Una Guía para el estudiante*. Prentice Hall.

Dia: Herramienta para la creación de diagramas. <http://www.gnome.org/projects/dia/>

DocBook: Herramientas y metodología para la generación de documentación <http://www.docbook.org> o también <http://www.oasis-open.org/docbook/>

Licencia Pública General de GNU (<http://www.gnu.org/licenses/>)

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

Newkirk, James; Martín, Robert (2001). *La programación extrema en la práctica*. Addison Wesley.

Planner: Herramienta de planificación de proyectos. <http://www.simpleprojectmanagement.com/planner/home.html>

ReadySet: Herramientas para la generación y mantenimiento de documentación basado en plantillas. <http://readysset.tigris.org/>

The Linux Documentation Project: <http://tldp.org/>

The Object Management Group – UML: <http://www.omg.org/uml/>

Licencia Pública General de GNU

Licencia Pública General de GNU
Versión 2, Junio de 1991
(<http://www.gnu.org/licenses/>)

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA Se permite la copia y distribución de copias literales de este documento, pero no se permite su modificación.

A.1. Preámbulo

Las licencias que cubren la mayor parte del software están diseñadas para quitarle a usted la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software de la Free Software Foundation y a cualquier otro programa cuyos autores se comprometen a utilizarla (existe otro software de la Free Software Foundation que está cubierto por la Licencia Pública General de GNU para Bibliotecas). Si quiere, usted también puede aplicarla a sus propios programas.

Cuando hablamos de software libre, estamos refiriéndonos a libertad, no a precio. Nuestras Licencias Públicas Generales están diseñadas para asegurarnos de que usted tenga la libertad de distribuir copias de software libre (y cobrar por ese servicio si quiere), de que reciba el código fuente o que pueda conseguirlo si lo quiere, de que pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que sepa que puede hacer todas estas cosas.

Para proteger sus derechos necesitamos hacer algunas restricciones que prohíban a cualquiera negarle a usted estos derechos o pedirle que renuncie los derechos. Estas restricciones se traducen en ciertas obligaciones que le afectan si usted distribuye copias del software, o si lo modifica.

Por ejemplo, si usted distribuye copias de uno de estos programas, ya sea gratuitamente o por cobrar, usted debe dar a todos los receptores todos los derechos que usted tiene. Usted debe asegurarse de que ellos también reciban o puedan conseguir el código fuente. Y debe mostrarles estas condiciones de manera que ellos pueden conocer sus derechos.

Nosotros protegemos sus derechos con estas dos medidas: (1) Ponemos el software bajo *copyright* y (2) le ofrecemos esta licencia que le da permiso legal a copiar, distribuir y/o modificar el software.

También, para la protección tanto de cada autor como la de nosotros mismos, queremos asegurar que todos entiendan que no hay garantía alguna para este software libre. Si alguien más ha modificado y distribuido el software, queremos que sus receptores sepan que no es el original, de manera que cualquier problema que otro introduzca no afecte la reputación de los autores originales.

Finalmente, cualquier programa libre se ve constantemente amenazado por las patentes de software. Deseamos evitar el peligro de que los re-distribuidores de un programa libre individualmente obtengan patentes, con el efecto de convertir el programa en un programa propietario. Para evitar esto, hemos dejado claro que cualquier licencia de patente debe ser conseguido para el uso libre de todos o no debe conseguirse.

Los términos y condiciones exactas para la duplicación, distribución y modificación se elaboran a continuación.

A.2. Licencia Pública General De Gnu

Términos y condiciones para la duplicación, distribución y modificación

0.

Esta licencia se aplica a cualquier programa u otra obra que contenga un aviso de parte del propietario del *copyright* diciendo que se puede distribuir bajo los términos de esta Licencia General Pública.

En adelante, "Programa" se refiere a cualquier dicho programa u obra, y "obra basada en el Programa" quiere decir ya sea el programa o cualquier obra derivada de él bajo las leyes de copyright. Es decir, una obra que contenga el Programa o una porción del mismo, ya sea literal o con modificaciones y/o traducido a otro idioma. (De aquí en adelante se incluye la traducción sin limitación en el término "modificación".) Se dirige a cada licenciatario como "Usted"

Esta licencia no cubre otras actividades fuera de la duplicación, distribución y modificación; éstas están fuera de su alcance. El acto de ejecutar el Programa no está restringido, y los datos que resultan de su uso están cubiertos solamente cuando constituyen una obra basada en el Programa, independientemente del hecho de haber sido producido por la ejecución del programa. El caso de que sea así o no depende de qué es lo que hace el Programa.

1.

Usted puede hacer y distribuir copias literales del código fuente del Programa tal como Usted lo recibió, en cualquier medio, con tal de que publique en cada copia, de manera visible y apropiada, un aviso sobre el copyright y repudiación de garantía; mantenga intactos todos los avisos que refieren a esta Licencia y la ausencia de garantía; y proporcione a cualquier otro receptor del Programa una copia de esta Licencia junto con el Programa.

Usted puede cobrar un honorario por el acto físico de transferir una copia, y a opción suya puede ofrecer protección de garantía a cambio de un honorario.

2.

Usted puede modificar su copia o copias del Programa o cualquier porción del mismo, y así formar una obra basada en el Programa, y duplicar y distribuir dichas modificaciones u obra bajo los términos de la antedicha apartado 1, con tal de que también cumpla con todas las siguientes condiciones:

- a) Usted debe hacer que los ficheros modificados lleven avisos indicando que Usted ha cambiado los ficheros, con la fecha de cualquier cambio.
- b) Usted debe hacer que cualquier obra que distribuya o publique que contenga o sea derivada del Programa o de una parte del

mismo, ya sea en su totalidad o en parte, sea licenciada en su totalidad sin coste a todas las terceras partes bajo los términos de esta licencia.

- c) Si el programa modificado normalmente lee órdenes interactivamente al ejecutarse, Usted debe hacer que, al iniciar dicho uso interactivo en la manera más habitual, el programa muestre un mensaje incluyendo un aviso apropiado de copyright y un aviso de que no hay garantía (o diciendo que Usted ofrece una garantía) y que los usuarios pueden redistribuir el programa bajo estas condiciones y avisando al usuario como ver esta Licencia. (Excepción: si el Programa mismo es interactivo pero no suele mostrar un mensaje de este tipo, entonces no se requiere que su obra basada en el Programa muestre un mensaje.)

Estos requisitos son aplicables a la obra modificada en su totalidad. Si secciones identificables de dicha obra no están derivadas del Programa y pueden ser razonablemente consideradas obras independientes y separadas en sí, entonces esta Licencia y sus términos no se aplican a esas secciones cuando Usted los distribuye como obras separadas. Pero cuando distribuye las mismas secciones como parte de una totalidad que es una obra basada en el Programa, la distribución del todo debe ser bajo los términos de esta Licencia, cuyos permisos para otros licenciatarios se extienden a la totalidad de la obra y por consiguiente a todas y cada una de sus partes, respectivamente de quien la haya escrito.

Por lo tanto, no es la intención de este apartado reclamar derechos o disputar sus derechos sobre obras escritas enteramente por Usted. Más bien, la intención es de ejercer el derecho de controlar la distribución de obras derivadas o colectivas basadas en el Programa.

Adicionalmente, la simple agregación de otra obra no basada en el Programa junta con el Programa (o con una obra basada en el Programa) en un volumen de un medio de almacenamiento o de distribución no extiende el alcance de esta Licencia a la otra obra.

3.

Usted puede duplicar y distribuir el Programa (o una obra basada en ella, bajo el apartado 2) en forma de código objeto o ejecutable bajo

los términos de los antedichos apartados 1 y 2, con tal de que también haga uno de los siguientes:

- a) Acompañarlo con el código fuente completo correspondiente en una forma legible por máquina, el cual debe ser distribuido bajo los términos de los antedichos apartados 1 y 2 en un medio habitualmente utilizado para el intercambio de software; o
- b) Acompañarlo con una oferta por escrito, válida por un mínimo de tres años, de proporcionar a cualquier tercera parte por un honorario que no exceda del coste de físicamente realizar la distribución de los fuentes, una copia completa en forma legible por máquina del código fuente correspondiente, el cual debe ser distribuido bajo los términos de los antedichos apartados 1 y 2 en un medio habitualmente utilizado para el intercambio de software; o
- c) Acompañarlo con la información que Usted recibió en cuanto a la oferta de distribución del código fuente correspondiente. (Esta alternativa sólo es permitida para distribución no comercial y solamente si Usted recibió el Programa en forma de código objeto o ejecutable con dicha oferta de acuerdo con el subapartado b anterior.)

El "código fuente" de una obra significa la forma preferida de la obra para hacer modificaciones a la misma. Para una obra ejecutable, el "código fuente completo" quiere decir todo el código fuente para todos los módulos que contiene, más cualesquier ficheros asociados de definición de interfaz, más los scripts que se utilizan para controlar la compilación e instalación del ejecutable. Sin embargo, como una excepción especial, el código fuente distribuido no necesita incluir algo que normalmente se distribuye (ya sea en forma de código fuente o en forma binaria) con los componentes principales (compilador, núcleo, etc.) del sistema operativo con el cual el ejecutable funciona, a no ser que dicho componente mismo acompañe el ejecutable.

Si la distribución del ejecutable o código objeto se hace al ofrecer acceso para copiarlo de un lugar designado, entonces el ofrecer acceso equivalente para copiar el código fuente del mismo lugar cuenta como distribución del código fuente, aunque no se exija a terceras partes que copien el código fuente junto con el código objeto.

4.

Usted no puede copiar, modificar, sublicenciar o distribuir el Programa excepto de la manera expresamente previsto por esta licencia.

Cualquier intento de copiar, modificar, sublicenciar o distribuir el Programa de otra manera es inválido y terminará sus derechos bajo esta Licencia automáticamente. Sin embargo, otras partes que hayan recibido copias o derechos de Usted bajo esta Licencia no perderán sus derechos mientras dichas partes sigan en pleno cumplimiento.

5.

Dado que no lo ha firmado, Usted no está obligado a aceptar esta licencia. Sin embargo, no hay nada más que le dé permiso para modificar o distribuir el Programa o sus obras derivadas. Estas acciones son prohibidas por la ley a no ser que Usted acepte esta Licencia. Por lo tanto, al modificar o distribuir el Programa (o cualquier obra basada en el Programa), Usted indica su aceptación de esta Licencia para hacerlo, y de todos sus términos y condiciones sobre la duplicación, distribución o modificación del Programa u obras basadas en él.

6.

Cada vez que Usted redistribuye el Programa (o cualquier obra basada en el Programa), el que lo recibe automáticamente recibe una licencia del licenciante original para copiar, distribuir o modificar el Programa, sujeto a estos términos y condiciones. Usted no puede imponer al receptor ninguna restricción adicional sobre el ejercicio de los derechos concedidos en la presente. Usted no es responsable de hacer que terceras partes cumplan con esta Licencia.

7.

Si como consecuencia de un fallo judicial o de una alegación de infracción de patente o por cualquier otra razón (no limitándose a cuestiones de patentes), se le imponga a Usted condiciones (ya sea por una orden judicial, acuerdo o de otra manera) que contradigan las condiciones de esta Licencia, éstas no le eximen de las condiciones de esta licencia. Si Usted no puede distribuirlo de manera que satisfaga simultáneamente sus obligaciones bajo esta licencia y cualquier otra obligación perteneciente, entonces en consecuencia Usted no puede distribuir el programa. Por ejemplo, si una licencia de patente no permite a todos los que reciban copias de Usted, ya sea directamente o indirectamente, redistribuir del Programa libre de regalías, entonces la única manera en que Usted puede cumplir tanto con ella como con esta Licencia sería de abstenerse del todo de la distribución del Programa.

Si cualquier parte de este apartado es considerado inválido o imposible de hacer cumplir bajo alguna circunstancia particular, el resto del apartado debe aplicarse y el apartado en su totalidad debe aplicarse en otras circunstancias.

No es el propósito de este apartado inducirlo a infringir algún patente u otro derecho de propiedad o a alegar contra la validez de algún derecho reclamado; este apartado tiene el único propósito de proteger la integridad del sistema de distribución de software libre, el cual se pone en práctica mediante licencias públicas. Muchas personas han hecho contribuciones generosas a la amplia gama de software distribuida mediante este sistema, confiando en la aplicación uniforme de dicho sistema. Depende del autor o donador decidir si está dispuesto a distribuir software mediante algún otro sistema y un licenciataria no puede imponer esa elección.

Este apartado pretende hacer abundantemente claro lo que se cree ser una consecuencia del resto de esta Licencia.

8.

Si la distribución y/o el uso del Programa está restringido en ciertos países debido a patentes o interfaces bajo copyright, el titular original del copyright que pone el Programa bajo esta Licencia puede añadir una limitación geográfica explícita a la distribución, excluyendo dichos países, de manera que la distribución quede permitido solamente en o entre países no así excluidos. En dicho caso, esta Licencia incorpora la limitación así como si estuviera escrita en el cuerpo de esta Licencia.

9.

La Free Software Foundation puede publicar versiones modificadas y/o nuevas de la Licencia Pública General de vez en cuando. Dichas versiones nuevas serán similares en espíritu a la versión presente, pero pueden ser diferentes en detalles para abarcar nuevos problemas o situaciones.

A cada versión se le dará un número de versión que lo distingue de otras. Si el Programa especifica un número de versión que se le aplica y "cualquier versión posterior", Usted tiene la opción de cumplir con los términos y condiciones ya sea de esa versión o de cualquier

versión posterior que publique la Free Software Foundation. Si el Programa no especifica un número de versión de esta Licencia, Usted puede escoger cualquier versión que la Free Software Foundation haya en algún momento publicado.

10.

Si Usted desea incorporar partes del Programa en otros programas libres cuyos condiciones para la distribución son diferentes, escriba al autor pidiendo permiso. Para software cuyo titular del copyright es la Free Software Foundation, escriba a la Free Software Foundation; en vezes hacemos excepciones para esto. Nuestra decisión será guiada por las dos metas de preservar el estado libre de todos los derivados de nuestro software libre y de promover que se comparta y reutilice el software en general.

11.**AUSENCIA DE GARANTÍA**

DEBIDO A QUE EL PROGRAMA SE LICENCIA LIBRE DE CARGAS, NO HAY GARANTÍA ALGUNA SOBRE EL PROGRAMA, EN LA MEDIDA PERMITA POR LAS LEYES APLICABLES. LOS TITULARES DEL COPYRIGHT Y/ U OTRAS PARTES PROVEEN EL PROGRAMA "TAL Y COMO ESTÁ" SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPRESA O IMPLÍCITA, INCLUYENDO SIN LIMITACIÓN LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN Y APTITUD PARA UN PROPÓSITO ESPECÍFICO, EXCEPTO CUANDO LO CONTRARIO SEA DECLARADO POR ESCRITO. TODO EL RIESGO EN CUANTO A LA CALIDAD Y ACCIÓN DEL PROGRAMA LO ASUME Usted SI EL PROGRAMA SE COMPRUEBA DEFECTUOSO, Usted ASUME TODO EL COSTE DE TODO SERVICIO, REPARACIÓN O CORRECCIÓN QUE SEA NECESARIO.

NINGÚN TITULAR DE COPYRIGHT NI OTRA PARTE QUE PUEDA MODIFICAR Y/O REDISTRIBUIR EL PROGRAMA SEGÚN SE PERMITE EN ESTA LICENCIA SERÁ RESPONSABLE ANTE Usted JAMÁS POR PERJUICIOS, INCLUYENDO CUALQUIER PERJUICIO GENERAL, ESPECIAL, INCIDENTAL O CONSECUENTE DEBIDO AL USO O LA IMPOSIBILIDAD DE PODER USAR EL PROGRAMA (INCLUYENDO SIN LIMITACIÓN LA PÉRDIDA DE DATOS O QUE DATOS SE VUELVAN INCORRECTOS O PÉRDIDAS SOSTENIDAS POR USTED O POR TERCERAS PARTES O LA IMPOSIBILIDAD DEL PROGRAMA A OPERAR

CON ALGÚN OTRO PROGRAMA), A NO SER QUE LEYES APLICABLES LO REQUIERAN O HAYA SIDO ACORDADO POR ESCRITO, AUNQUE DICHO TITULAR U OTRA PARTE HAYA SIDO AVISADO DE LA POSIBILIDAD DE TALES PERJUICIOS.

FIN DE LOS TÉRMINOS Y CONDICIONES

Anexo

Cómo aplicar estos términos a sus programas nuevos.

Si Usted desarrolla un nuevo programa y quiere que sea del mayor uso posible al público, la mejor manera de conseguir esto es convertirlo en software libre que todo el mundo puede redistribuir y cambiar bajo estos términos.

Para hacerlo, adjunte los siguientes avisos al programa. Lo más seguro es adjuntarlos al principio de cada fichero fuente para comunicar lo más eficazmente posible la ausencia de garantía, y cada fichero debe incluir por lo menos el renglón de "copyright" y una indicación del lugar donde se encuentra la notificación completa.

```
<un renglón para dar el nombre del programa y una
idea breve de lo que hace> Copyright (C) <año>
<nombre del autor>
```

```
Este programa es software libre. Usted lo puede
redistribuir y/o modificar bajo los términos de la
Licencia Pública General de GNU publicada por la
Free Software Foundation, ya sea la versión 2 de
la Licencia o (a opción suya) cualquier versión
posterior.
```

```
Este programa se distribuye con la esperanza de
que sea útil, pero SIN GARANTÍA ALGUNA, ni siquie-
ra la garantía implícita de COMERCIALIZACIÓN o AP-
TITUD PARA UN PROPÓSITO ESPECÍFICO. Véase la
Licencia Pública General de GNU para más detalles.
```

Usted debe haber recibido una copia de la Licencia Pública General de GNU junto con este programa; en caso que no, escriba a la Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Además, incluya información sobre cómo ponerse en contacto con Usted tanto por correo electrónico como por correo postal.

Si el programa es de tipo interactivo, debe mostrar un breve mensaje como el siguiente cuando comienza en la modalidad interactiva:

```
Gnomovision versión 69, Copyright ©<año> nombre
del autor Gnomovision viene ABSOLUTAMENTE SIN
GARANTIA; para detalles teclee 'show w' Este es
software libre, y Usted lo puede redistribuir li-
bremente bajo ciertas condiciones; para más de-
talles, teclee 'show c'.
```

Los comandos hipotéticos 'show w' y 'show c' deben mostrar las partes apropiadas de la Licencia Pública General de GNU. Claro, los comando que Usted use pueden ser diferentes que 'show w' y 'show c'. Incluso pueden ser pulsaciones al botón del ratón o elementos de un menú –lo que sea apropiado para su programa.

Usted también debe pedir a su empleador (si Usted trabaja como programador) o su universidad (en dicho caso), a firmar una renuncia de copyright para el programa en caso de que sea necesario. A continuación se ofrece un ejemplo; hay que cambiar los nombres:

```
Yoyodyne, Inc. con la presente renuncia todo in-
terés de derechos de copyright en el programa
'Gnomovision' (el cual hace pasadas a compilado-
res) escrito por Jaime Hacker.
```

```
<firma de Ty Coon>, 1 Abril 1989 Ty Coon, Presi-
dente de Alguien, Inc.
```

Esta Licencia Pública General no permite la inclusión de su programa en programas propietarios. Si su programa es una biblioteca de subrutinas, Usted puede considerar más útil el permitir el enlazado de aplicaciones propietarias con su biblioteca. Si es esto lo que Usted quiere hacer, use mejor la Licencia Pública General para Bibliotecas de GNU en lugar de esta Licencia.

