

3. ANTECEDENTES, ESTADO DE LA CUESTIÓN

Las empresas se han dado cuenta que ahora, y muy especialmente en un futuro muy cercano, la Intranet va a ser un recurso indispensable, debido a la gran cantidad de datos que generan y que ayudaría a resolver algunos problemas de información. Una Intranet bien implementada puede resolver, por ejemplo, el problema de la distribución de información para todos los empleados, así pues se pueden publicar manuales, planes de acción, procedimientos, material de formación, folletos de marketing y productos, listas de precios, información comercial, anuncios, promociones, etc. Y son accesibles para el empleado o cliente de forma inmediata, y con un ahorro considerable respecto a los métodos clásicos, tales como panfletos, circulares, notas informativas, etc. Además, cualquier actualización de datos es inmediata y no supone ninguna carga para la empresa, como los métodos tradicionales.

Una intranet permite organizar la distribución de las empresas, pues cada división podría tener su apartado en la intranet. Se puede organizar también una lista de encuentros y reuniones a la que cada empleado podrá acceder rápidamente, planificando así las reuniones de cada departamento de una forma más eficaz. Se mejora de esta forma la comunicación entre todos los trabajadores, y las sugerencias, peticiones o cualquier comunicación en general, se realizarían de una forma más rápida y eficiente.

Se aprovecharía también la potencia de una intranet para tener acceso rápido a cualquier documento de la empresa, siempre que se tenga el nivel de privilegios adecuado. Esta es otra de las ventajas de una intranet, su seguridad. Solo tendrán acceso a los recursos aquellos empleados que lo necesiten realmente. Siguiendo con la potencia y velocidad de acceso a datos de una intranet, el tiempo empleado en realizar cualquier búsqueda de datos de cualquier departamento de la empresa se reduciría considerablemente, por lo que la productividad de la empresa mejoraría.

Así, en los últimos años son varios los grupos que han trabajado y han propuesto metodologías que ofrecen procesos, modelos y técnicas adecuadas para trabajar con este tipo de sistemas¹. Sin embargo, si se analizan las diferentes propuestas, la gran mayoría de ellas enfocan su trabajo en la etapa de diseño del ciclo de vida, dando menor importancia a la ingeniería de requisitos, al testeo y a la gestión de calidad.

¹ Koch, 1999 & 2001, Retschitzegger & Schwinger, 2000 y Escalona, Mejías & Torres, 2002

El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema, así como las restricciones sobre las que se deberá operar. Consiste en un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido². La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen, se deben a una inadecuada ingeniería de requisitos.

3.1 La Ingeniería de Requisitos

En el proceso de desarrollo de un sistema, sea o no para la Web, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos. La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes.

Para realizar este proceso, no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio un conjunto de técnicas, cuyo uso proponen las diferentes metodologías para el desarrollo de aplicaciones Web. Se debe tener en cuenta que la selección de las técnicas y el éxito de los resultados que se obtengan, depende en gran medida tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen.

El proceso de especificación de requisitos se puede dividir en tres grandes actividades³:

- 1- Captura de requisitos
- 2- Definición de requisitos
- 3- Validación de requisitos

El proceso comienza con la realización de la captura de requisitos, el equipo de desarrollo toma la información suministrada por los usuarios y clientes. Esta información puede provenir de fuentes muy diversas: documentos, aplicaciones existentes, a través de entrevistas, etc. Con base en esta información, el equipo de desarrollo elabora el catálogo de requisitos. Finalmente, con la validación de requisitos se realiza la valoración de los mismos, comprobando si existen inconsistencias,

² Ferreira & Loucopoulos, 2001

³ Lowe & Hall, 1999

errores o si faltan requisitos por definir. El proceso de definición-validación es iterativo y en algunos proyectos complejos resulta necesario ejecutarlo varias veces.

En las siguientes páginas se presentan brevemente algunas técnicas clásicas para realizar las actividades de captura, definición y validación de requisitos. Estas técnicas pueden resultar más o menos apropiadas a la ingeniería de requisitos para aplicaciones en el entorno Web. Resulta muy difícil establecer criterios para seleccionar técnicas apropiadas. Entre estos criterios pueden ser considerados la facilidad de aprendizaje y de uso, el costo, la calidad y completitud de los resultados y el tiempo requerido para aplicar las técnicas. Así, podemos decir que el uso de lenguajes naturales produce resultados más imprecisos que una descripción con casos de uso y ésta a su vez es más imprecisa que los requisitos descritos formalmente. Los casos de uso son apropiados tanto para pequeños como para grandes sistemas, mientras que el uso de plantillas resulta menos apto para grandes sistemas. Así mismo, técnicas como JAD son más difíciles de usar y consumen mucho más tiempo que las entrevistas, permitiendo en cambio obtener resultados de mayor calidad.

3.1.1 Captura de requisitos

La captura de requisitos es la actividad mediante la cual el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir dicho sistema⁴. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa. A continuación se presentan un grupo de técnicas que de forma clásica han sido utilizadas para esta actividad en el proceso de desarrollo de todo tipo de software.

3.1.1.1 Entrevistas

Resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada. A través de esta técnica

⁴ Díez, 2001

el equipo de trabajo se acerca al problema de una forma natural. Existen muchos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización⁵.

Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados (protocolo de la entrevista).

A pesar de que las entrevistas son esenciales en el proceso de la captura de requisitos y con su aplicación el equipo de desarrollo puede obtener una amplia visión del trabajo y las necesidades del usuario, es necesario destacar que no es una técnica sencilla de aplicar⁶. Requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. Aquí desempeña un papel fundamental la preparación de la entrevista (ver cuestionarios y checklists).

3.1.1.2 JAD (Joint Application Development/Desarrollo conjunto de aplicaciones)

Esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes⁷. Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, “lo que ve es lo que obtiene”), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará. Tras una fase de preparación del JAD al caso concreto, el equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación.

Durante la sesión se discute en grupo sobre estos temas, llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema.

Esta técnica presenta una serie de ventajas frente a las entrevistas tradicionales, ya que ahorra tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado, pero requiere un grupo de participantes bien integrados y organizados.

⁵ Durán, Bernáldez, Ruíz & Toro, 1999 y Pan, Zhu & Jonhson, 2001

⁶ Pan, Zhu & Johnson, 2001

⁷ IBM, 1997

3.1.1.3 Brainstorming (Tormenta de ideas)

Es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre⁸. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador.

Como técnica de captura de requisitos, es sencilla de usar y de aplicar, contrariamente al JAD, puesto que no requiere tanto trabajo en grupo como ésta. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no sirve para obtener detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.

3.1.1.4 Concept Mapping

Los *concept maps*⁹ son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en el lenguaje del usuario. Sin embargo, deben ser usados con cautela porque en algunos casos pueden ser muy sugestivos y pueden llegar a ser ambiguos en casos complejos, si no se acompaña de una descripción textual.

3.1.1.5 Sketches y Storyboards

Esta técnica es frecuentemente usada por los diseñadores gráficos de aplicaciones en el entorno Web. La misma consiste en representar sobre el papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces, dando idea de la estructura de navegación (storyboard).

⁸ Raghavan, Zelesnik & Ford, 1994

⁹ Pan, Zhu & Johnson, 2001

3.1.1.6 Casos de Uso

Aunque inicialmente se desarrollaron como técnica para la definición de requisitos¹⁰, algunos autores proponen casos de uso como técnica para la captura de requisitos¹¹. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales, expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los *actores* del mismo para realizar una determinada función. Los actores son elementos externos (personas, otros sistemas, etc.) que interactúan con el sistema como si de una caja negra se tratase. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria¹² si no se acompañan con una información textual o detallada con otra técnica, como pueden ser los diagramas de actividades¹³.

3.1.1.7 Cuestionarios y Checklists

Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas, por unas cuantas opciones en el propio cuestionario (Checklist). Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista.

3.1.1.8 Comparación de terminología

Uno de los problemas que surge durante la elicitación de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras técnicas para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos (contraste)¹⁴.

¹⁰ Jacobson, 1995

¹¹ Pan, Zhu & Johnson, 2001 y Liu & Yu, 200

¹² Vilain, Schwabe & Sieckenius, 2002 y Insfrán, Pastor & Wieringa, 2002

¹³ UML, 2001

¹⁴ Pan, Zhu & Johnson, 2001

Existen más técnicas para la captura de requisitos (análisis de otros sistemas, estudio de la documentación, etc.), incluso también es común encontrar alternativas que combinen varias de estas técnicas¹⁵. Sin embargo, las presentadas ofrecen un conjunto representativo de las más utilizadas y resultan suficientes para el objetivo de este trabajo.

3.1.2 Definición de requisitos

También, para la actividad de definición de requisitos en el proceso de ingeniería de requisitos, hay un gran número de técnicas propuestas. Describiremos brevemente las más relevantes para este trabajo.

3.1.2.1 Lenguaje natural

Resulta una técnica muy ambigua para la definición de los requisitos. Consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. A pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando.

3.1.2.2 Glosario y ontologías

La diversidad de personas que forman parte de un proyecto software hace que sea necesario establecer un marco de terminología común. Esta necesidad se vuelve más patente en los sistemas de información Web, puesto que el equipo de desarrollo en ellas suele ser más interdisciplinario¹⁶. Por esta razón son muchas las propuestas que abogan por desarrollar un glosario de términos en el que se recogen y definen los conceptos más relevantes y críticos para el sistema.

En esta línea se encuentra también el uso de ontologías, en las que no sólo aparecen los términos, sino también las relaciones entre ellos. Ninguna de las metodologías para el entorno Web incluidas en este estudio comparativo por su relevancia en la ingeniería de requisitos, propone la definición de una ontología. Por ello la técnica ontología no ha sido incorporada.

¹⁵ Pan, Zhu & Johnson, 2001

¹⁶ Koch, 2001

3.1.2.3 Plantillas o patrones

Esta técnica, recomendada por varios autores¹⁷, tiene por objetivo describir los requisitos mediante el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumpliendo, usando para ello, el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea ésta, menos ambigüedad ofrece. Sin embargo, si el nivel de detalle elegido es demasiado estructurado, el trabajo de rellenar las plantillas y mantenerlas, puede ser demasiado tedioso.

3.1.2.4 Escenarios

La técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos¹⁸. La representación del escenario puede variar, dependiendo del autor. Esta representación puede ser casi textual o ir encaminada hacia una representación gráfica en forma de diagramas de flujo¹⁹. El análisis de los escenarios, hechos de una forma u otra, pueden ofrecer información importante sobre las necesidades funcionales del sistema²⁰.

3.1.2.5 Casos de uso

Como técnica de definición de requisitos, han sido más ampliamente aceptados los casos de uso. Actualmente se ha propuesto como técnica básica del proceso RUP²¹. Sin embargo, son varios los autores que defienden que pueden resultar ambiguos a la hora de definir los requisitos²², por lo que hay propuestas que los acompañan de descripciones basadas en plantillas o diccionarios de datos que eliminen su ambigüedad.

3.1.2.6 Lenguajes Formales

Otro grupo de técnicas que merece la pena resaltar como extremo opuesto al lenguaje natural, es la utilización de lenguajes formales para describir los requisitos de un sistema. Las especificaciones algebraicas como ejemplo de técnicas de descripción

¹⁷ Durán, Bernáldez, Ruíz & Toro, 1999 y Escalona, Torres & Mejías, 2002

¹⁸ Liu & Yu, 2001

¹⁹ Weidenhaupt, Pohl, Jarke & Haumer, 1999

²⁰ Lowe & Hall, 1999

²¹ Kruchten, 1998

²² Díez, 2001 y Vilain, Schwabe & Sieckenius, 2002 y Insfrán, Pastor & Wieringa, 2002

formal, han sido aplicadas en el mundo de la ingeniería de requisitos desde hace años²³. Sin embargo, resultan muy complejas en su utilización y para ser entendidas por el cliente. El mayor inconveniente es que no favorecen la comunicación entre cliente y analista. Por el contrario, es la representación menos ambigua de los requisitos y la que más se presta a técnicas de verificación automatizadas.

3.1.3 La validación de requisitos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos establece realmente el sistema que el usuario necesita o el cliente desea²⁴. Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de los requisitos son correctos. Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias.

Aún así, existen algunas técnicas que pueden aplicarse para ello:

3.1.3.1 Reviews o Walk-throughs

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.

3.1.3.2 Auditorías

La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist predefinida o definida a comienzos del proceso, es decir, sólo una muestra es revisada.

²³ Peña, 1998

²⁴ Lowe & Hall, 1999

3.1.3.3 Matrices de trazabilidad

Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo²⁵. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos, y así poder lograr su corrección.

3.1.3.4 Prototipos

Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario²⁶. Esta técnica tiene el problema que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

3.2 Tratamiento de Requisitos en Propuestas para la Web

El desarrollo de sistemas Web agrupa una serie de características que lo hacen diferente del desarrollo de otros sistemas²⁷. Por un lado, hay que tener en cuenta que roles muy diferentes de desarrolladores participan en el proceso: analistas, clientes, usuarios, diseñadores gráficos, expertos en multimedia y seguridad, etc. Por otro lado, la existencia en estos sistemas de una importante estructura de navegación obliga a un desarrollo preciso de este aspecto que garantice que el usuario no se “pierde en el espacio navegacional del sistema”²⁸. Estas ideas, unidas al hecho que los sistemas Web suelen tratar con múltiples medios y es esencial que ofrezcan una interfaz adecuada en cada momento, obligan a que estos aspectos propios de la Web deban ser tratados de una forma especial en el proceso de desarrollo. Estas características especiales también hay que tenerlas en cuenta en la fase de especificación de requisitos²⁹. Por ello, la mayoría de las propuestas estudiadas ofrecen diferentes clasificaciones de los requisitos. Sin embargo, la terminología usada no es siempre la misma. Para facilitar la comprensión de las propuestas, antes de presentarlas, se muestra una clasificación de requisitos relevantes en sistemas Web.

²⁵ Durán, Bernáldez, Ruíz & Toro, 1999

²⁶ Olsina, 1999

²⁷ Koch, 2001

²⁸ Olsina, 1999

²⁹ Escalona, 2002

3.2.1 Requisitos de datos, también denominados requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información. Estos requisitos responden a la pregunta de qué información debe almacenar y administrar el sistema.

3.2.2 Requisitos de interfaz (*al usuario*), también llamados en algunas propuestas requisitos de interacción o de usuario. Responden a la pregunta de cómo va a interactuar el usuario con el sistema.

3.2.3 Requisitos navegacionales, recogen las necesidades de navegación del usuario.

3.2.4 Requisitos de personalización, describen cómo debe adaptarse el sistema en función de qué usuario interactúe con él y de la descripción actual de dicho usuario.

3.2.5 Requisitos transaccionales o funcionales internos, recogen qué debe hacer el sistema de forma interna, sin incluir aspectos de interfaz o interacción. También son conocidos en el ambiente Web como requisitos de servicios.

3.2.6 Requisitos no funcionales, son por ejemplo los requisitos de portabilidad, de reutilización, de entorno de desarrollo, de usabilidad, de disponibilidad, etc. Son muchos los autores que han propuesto técnicas y modelos para el desarrollo de Aplicaciones Web, pero como se ha comentado, estas propuestas se centran principalmente en las últimas fases del proceso de desarrollo. En este apartado se van a presentar aquellas técnicas que incluyen el proceso de definición de requisitos dentro del ciclo de vida. Algunas de las propuestas que se describen, cubren la captura y definición de requisitos desde sus primeras propuestas. Otras, como se verán, la incluyen en revisiones a sus trabajos iniciales.

El criterio que se ha seguido para presentar las propuestas es el cronológico, desde la primera publicación que incluye tratamiento de requisitos. Ello permitirá en la comparativa evaluar la evolución de las propuestas para requisitos.

3.3 Otras herramientas para el acoplamiento Web

Hace 30 o 40 años, el software no era considerado un producto sino un añadido que los vendedores de los grandes computadores de la época aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios con el uso de acuerdos de licencia, práctica que hasta hace unos años había generado una dependencia absoluta entre consumidores y fabricantes. Fue ésta la razón por la cual se dio inicio nuevamente a la creación, desarrollo y difusión del software libre, software que refiere su razón de ser, a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar su código fuente, dando libertad y restricción a las posibilidades de apropiación del software.

Es por esto, que se ha hecho un acoplamiento con servicios de herramientas como Servidor HTTP Apache, E-GroupWare, protocolo LDAP, servicios OpenLDAP, entre otros.

3.3.1 Servidor HTTP Apache. El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1³⁰ y la noción de sitio virtual.

Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 48% de los sitios web en el mundo y decreciendo su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft³¹).

³⁰ RFC 2616, Fielding, 1999

³¹ NETCRAFT, Internet Services

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas, puede en la mayoría de los casos ser abusada solamente por los usuarios locales y no puede ser accionada remotamente. Sin embargo, algunas de las ediciones anteriores se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Las principales ventajas de Apache son:

- Estructura modular
- Open source (Código abierto)
- Multiplataforma
- Extensible
- Popular
- Gratuito

3.3.2 E-GroupWare. eGroupware es una solución de trabajo en grupo vía web, de código abierto. Está escrita en PHP utilizando bases de datos, tales como PostgreSQL o MySQL. Incluye un calendario, una libreta de direcciones, un gestor de contactos, un cliente de correo electrónico IMAP, un InfoLog, funciones de CRM, un gestor de proyectos, un gestor de recursos, un gestor de ficheros, una planilla de tiempos, un wiki, una base de conocimiento y un motor de flujos de trabajo. El equipo principal detrás de eGroupware es de habla alemana y está localizado en Alemania.

eGroupWare es una suite de programas libres para la empresa que están listos para el trabajo en grupo en su red corporativa. Permite la gestión de contactos, citas, tareas y muchas más cosas para todo su negocio. Viene con una interfaz web nativa que permite el acceso a los datos desde cualquier plataforma y desde cualquier punto del planeta. Es más, también permite elegir un cliente para acceder a los datos del servidor (Kontact, Evolution, Outlook) y también mediante teléfono móvil o PDA mediante SyncML. Es internacional y en este instante, soporta más de 25 idiomas, incluyendo soporte rtl.

eGroupWare es independiente de la plataforma; su servidor corre bajo Linux, Mac, Windows y otros muchos sistemas operativos. En el lado del cliente, lo único que hace falta es un navegador de internet como Firefox, Konqueror, Internet Explorer y muchos más.

3.3.3 Protocolo LDAP. LDAP (*Lightweight Directory Access Protocol*) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas.

Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse, aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc).

En conclusión, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red, que se usó inicialmente como un *front-end* o interfaz final para X.500, pero que ahora puede usarse con servidores de directorio únicos y con otros tipos de servidores de directorio.

3.3.3.1 Servicio de Directorio

Un directorio es como una base de datos, pero en general contiene información más descriptiva y más basada en atributos. La información contenida en un directorio normalmente se lee mucho más de lo que se escribe. Como consecuencia, los directorios no implementan normalmente los complicados esquemas para transacciones o esquemas de reducción (*rollback*), que las bases de datos utilizan para llevar a cabo actualizaciones complejas de grandes volúmenes de datos. Por el contrario, las actualizaciones en un directorio son usualmente cambios sencillos de “todo o nada”, si es que se permiten en algo.

Los directorios están afinados para proporcionar una respuesta rápida a operaciones de búsqueda o consulta. Pueden tener la capacidad de replicar información de forma amplia, con el fin de aumentar la disponibilidad y la fiabilidad, y a la vez reducir el tiempo de respuesta. Cuando se duplica (o se replica) la información del directorio, pueden aceptarse inconsistencias temporales entre la información que hay en las réplicas, siempre que finalmente exista una sincronización.

Existen muchas maneras distintas de proporcionar un servicio de directorio. Los diferentes métodos permiten almacenar en el directorio diferentes tipos de información, establecer requisitos diferentes para hacer referencias a la información, consultarla y actualizarla, la forma en que protege al directorio de accesos no autorizados, etc. Algunos servicios de directorio son locales, proporcionando servicios a un contexto restringido (por ejemplo, el servicio de finger en una única máquina). Otros servicios son globales, proporcionando servicio en un contexto mucho más amplio.

3.3.3.2 Funcionamiento del LDAP

El servicio de directorio LDAP se basa en un modelo cliente-servidor. Uno o más servidores LDAP contienen los datos que conforman el árbol del directorio LDAP o base de datos troncal. El cliente ldap se conecta con el servidor LDAP y le hace una consulta. El servidor contesta con la respuesta correspondiente, o bien con una indicación de dónde puede el cliente hallar más información (normalmente otro servidor LDAP). No importa con qué servidor LDAP se conecte el cliente: siempre observará la misma vista del directorio; el nombre que se le presenta a un servidor LDAP hace referencia a la misma entrada a la que haría referencia en otro servidor LDAP. Es ésta una característica importante de un servicio de directorios universal como LDAP.

3.3.3.3 Las ventajas de los directorios LDAP

Tal vez la mayor ventaja de LDAP es que los usuarios de las empresas pueden acceder al directorio LDAP desde cualquier plataforma de computación, es decir, desde cualquier aplicación disponible para LDAP. Es también fácil personalizar las aplicaciones internas de las empresas para añadirles soporte LDAP.

El protocolo LDAP es utilizable por distintas plataformas y basado en estándares, de ese modo, las aplicaciones no necesitan preocuparse por el tipo de servidor en que se hospeda el directorio. De hecho, LDAP está encontrando mucha más amplia aceptación a causa de ese estatus como estándar de Internet. Los vendedores están más deseosos de codificar en sus productos integración con LDAP, porque no tienen que preocuparse de lo que hay al otro lado. El servidor LDAP puede ser cualquiera de un gran número de servidores de directorio LDAP de código abierto o comercial, puesto que acarrean el mismo protocolo, paquete de conexión cliente y comandos de consulta.

A diferencia de las bases de datos relacionales, no hay que pagar por cada conexión de software cliente o por licencia. La mayoría de los servidores LDAP son simples de instalar, de fácil mantenimiento y fácilmente optimizables.

Los servidores LDAP pueden replicar tanto algunos de sus datos como todos, a través de métodos de envío o recepción, lo que permite enviar datos a oficinas remotas, incrementar la seguridad y demás. La tecnología de replicación está incorporada y es fácil de configurar. LDAP permite delegar con seguridad la lectura y modificación basada en autorizaciones según las necesidades, utilizando ACIs (Lista de Control de Acceso por sus siglas en inglés). Las ACIs pueden controlar el acceso, dependiendo

de quién está solicitando los datos, qué datos están siendo solicitados, dónde están los datos almacenados, y otros aspectos del registro que está siendo modificado. Todo esto, directamente a través del directorio LDAP, así que no hay necesidad de preocuparse por hacer comprobaciones de seguridad en el nivel de aplicación de usuario.

LDAP es particularmente utilizable para almacenar información que se desee leer desde muchas localizaciones, pero que no sea actualizada frecuentemente. Por ejemplo, la empresa podría almacenar todos los datos siguientes en un directorio LDAP:

- La lista de teléfonos de empleados de la empresa y el esquema organizacional.
- Información de contacto de clientes externos.
- Información de la infraestructura de servicios, alias de email y demás.
- Información de configuración para paquetes de software distribuidos.
- Certificados públicos y claves de seguridad.

3.3.4 Open LDAP. OpenLDAP es una implementación libre y de código abierto del protocolo Lightweight Directory Access Protocol (LDAP) desarrollado por el OpenLDAP Project. Está liberada bajo su propia licencia OpenLDAP Public License. LDAP es un protocolo independiente de la plataforma. Muchas distribuciones Linux incluyen el software OpenLDAP para su soporte. El software también corre en variantes de BSD, AIX, HP-UX, Mac OS X, Solaris, Microsoft Windows (2000, XP), y z/OS.

3.3.5 Base de Datos MySQL. MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual³². Por un lado lo ofrece bajo la GNU GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso.

Está desarrollado en su mayor parte en ANSI C. Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual; MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es

³² MySQL AB, Dispelling Myths

lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

3.3.6 PHP MyAdmin. Es proyecto de código abierto en PHP para administrar la base de datos MySQL a través de una interfaz web, creado por una comunidad sin ánimo de lucro, que sólo trabaja en el proyecto por amor al arte. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.

La aplicación en sí no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

4. METODOLOGÍA

El método de trabajo seguido para la realización del proyecto ha estado influido por las fases en las que se ha desarrollado:

- a) Realización de un estudio profundo sobre el estado de la cuestión, incidiendo en el desarrollo de software de aplicaciones Web (metodologías, arquitecturas, tecnologías, etc.).
- b) Elección de la metodología y la tecnología para resolver el problema y conseguir el objetivo principal de este proyecto de fin de carrera.
- c) Estudio y análisis de los requisitos de la aplicación a desarrollar.
- d) Diseño de la aplicación.
- e) Implementación de la aplicación.
- f) Realizar el manual de usuario de la aplicación.
- g) Puesta en funcionamiento de una instancia de la INTRANET AUTEKO, que ha servido como producto prototipo para pruebas.

No se han numerado las fases porque no se han realizado secuencialmente. De hecho, algunas se han solapado en el tiempo. Otras se han realimentado a medida que avanzaba el proyecto.

El resultado de la fase *a)* se ha redactado en el *Capítulo 3*. En la fase *b)* se tomó la decisión de utilizar una “metodología propia”, teniendo como base la metodología OOWS (Object Oriented Web Solution)³³. Algunos de los conceptos aprendidos en la fase *c)* han sido reflejados en el apartado dedicado a los sistemas gestores de contenidos.

La fase *g)* se incluye en el *capítulo 5*.

El resto de este capítulo se dedica a la exposición de las fases *d)*, *e)* y *f)*.

En definitiva, a continuación se realiza un estudio del ciclo de vida seguido para el desarrollo de la aplicación. Debido a la extensión de su funcionalidad, se han elegido los elementos claves para la comprensión de la funcionalidad, el diseño y los detalles de implementación de la aplicación.

³³ Fons J., Pelechano V., Pastor O., Albert M., and Valderas P. *Extending an OO Method to Develop Web Applications*. The Twelfth International World Wide Web Conference. Budapest, Hungary. 20-24 May, 2003

4.1 Estudio y análisis de los requisitos de la aplicación.

Este apartado realiza un estudio del ciclo de vida seguido para el desarrollo de la aplicación.

4.1.1 Organización y funcionamiento de AUTOTÉCNICA COLOMBIANA S.A.

La historia de Auteco se remonta más de sesenta años en el pasado, cuando a finales de la década de los 30's y principios de los 40's del siglo pasado, un emprendedor antioqueño educado en Inglaterra, se dio a la tarea de crear una nueva empresa de carácter comercial. En este proceso analizó varios factores inherentes a la época: el primero, que la población de las ciudades – que apenas eran unos pueblos grandes– estaba creciendo de manera acelerada; y el segundo, que estaban por abrirse la mayoría de las carreteras Inter-Departamentales que conocemos hoy. Estas observaciones lo llevaron a la conclusión de que el sector automotor tendría un importante desarrollo en las siguientes décadas, tanto en lo relativo a automóviles para las ciudades, como a camiones para el transporte de carga y, además, que habría una clase trabajadora que necesitaría un transporte económico y práctico, como las motocicletas, prácticamente desconocidas en esa época.

Teniendo esta idea como norte y con un capital inicial de \$10.000, prestados por un banco local, Autotécnica Colombiana (Auteco) inició operaciones el 1 de septiembre de 1941, con una bomba de gasolina y un pequeño almacén de repuestos situados cerca al Hospital San Vicente de Paúl en la ciudad de Medellín. En 1945, construyó su primera sede propia en un lote situado donde actualmente se encuentra Dismerca, en la esquina de la calle 37 (conocida como la Avenida 33) con Palacé, que en ese entonces era el extremo sur de la ciudad. Allí estableció un almacén de repuestos Ford y Chevrolet, una sala de exhibición y los más modernos talleres de servicio de la ciudad.

Fiel a la idea inicial de traer al país un medio de transporte económico, Auteco importó antes de 1945, las primeras motos, marca Indian, procedentes de Estados Unidos, de la cual le vendieron varias a la Guardia Presidencial, que un tiempo después fueron totalmente destruidas en los disturbios del 9 de abril de 1948, después del asesinato de Jorge Eliécer Gaitán en Bogotá. Sin embargo, las Indian resultaban costosas para nuestro medio, por lo cual pronto se cambiaron por motos Ariel, Excelsior y Monarch originarias de Inglaterra, que eran más económicas.

En los primeros años de la década de los 50 se estableció contacto con la fábrica italiana Innocenti, productora de las motonetas Lambretta, cuyos derechos de

ensamble, fabricación gradual y venta para Colombia fueron adquiridos por Auteco, comenzando su comercialización en 1954 a un precio de \$1.600 pesos. Durante esa década, se incrementó sustancialmente la producción de las Lambretta, gracias a su gran éxito comercial.

Un punto de quiebre se dio a principios de los años 70, cuando la fábrica Innocenti dejó de producir las Lambretta, debido en parte a conflictos laborales y a que los usuarios italianos de este producto fueron cambiando sus preferencias por automóviles de bajo costo que estaban cada vez más a su alcance, como el Fiat 500, más recordado como Topolino, o por motocicletas japonesas que venían imponiéndose en el mundo. Ante estas circunstancias, los directivos de la compañía viajaron a Japón y establecieron contactos con las cuatro grandes marcas, llegando finalmente a un acuerdo de ensamble y asistencia técnica con Kawasaki Heavy Industries, productora de equipo pesado, barcos, aviones, trenes y también motocicletas. El ensamble de modelos de esta marca comenzó en 1972.

La moto más popular del portafolio de Kawasaki fue la G7-100 “Korrecaminos”, también conocida como la Deluxe, que se ensambló por más de 20 años. Su compacto motor de 100cc y dos tiempos demostró ser muy fuerte y duradero, ganándose la confianza de toda una generación que la usaba para el transporte y para trabajo. La G7-100 fue la primera moto y el primer medio de transporte propio para muchas personas que la adquirieron a crédito o con los ahorros de mucho tiempo. Aún hoy, podemos ver una gran cantidad de estas máquinas rodando en la calle, siendo muy apreciada en el mercado de usadas.

El éxito inicial de Kawasaki en nuestro medio decayó en los 80, cuando se conjugaron varios factores. Por una parte, se fortaleció la competencia de las otras ensambladoras de motos que existen en nuestro país. Yamaha había llegado en 1978 y Suzuki y Honda unos años después, lo cual amplió el portafolio de opciones para los motociclistas, en especial en las cilindradas pequeñas, en las cuales Kawasaki no tenía un desarrollo tan importante como las otras marcas japonesas, porque la mayoría de sus esfuerzos tecnológicos los invertía en máquinas de alta cilindrada para el consumo del mercado americano, el europeo y el propio Japón, donde tenía y tiene una ganada reputación, que no satisfacían las necesidades de nuestro medio, lo que ocasionó que no pudieran ofrecer los modelos adecuados para competir con las otras marcas. Por otra parte, comenzó una fuerte recesión económica; el dólar vivió una gran devaluación, lo que incrementó el precio de las motocicletas en general y ocasionó una crisis financiera en Auteco. Para rematar, el Gobierno creó un régimen especial de ensamble de vehículos, carros y motos, por medio del cual restringió la

cantidad de modelos que se podían producir. Cada empresa podía ensamblar cuatro modelos; las demás importaciones estaban prácticamente prohibidas, con algunas excepciones para competencias deportivas.

Todas las experiencias vividas en los años 80 hicieron que la empresa, dentro de su visión estratégica, se preocupara por buscar otro fabricante que le permitiera complementar su gama de productos y proveer transporte económico y confiable a otros segmentos del mercado, que hasta ese momento había tenido desatendidos. De esta manera se iniciaron, a principios de los 90, contactos con el fabricante Bajaj de la India, al cual fueron remitidos por el propio Kawasaki, con quien Bajaj tiene acuerdos de cooperación tecnológica. Uno de los principales modelos de Bajaj era una motoneta que por sus características de motor y chasis era muy parecida a la mítica Vespa Italiana. A pesar de que su nivel de calidad no se equiparaba con las motos japonesas, Auteco decidió apostarle a volver a introducir la motoneta en nuestro país, porque, entre otras cosas, las mismas razones que hicieron tan exitosa a la Lambretta en los 60, seguían vigentes: la necesidad de un transporte práctico, económico y accesible para un grupo de usuarios que no podían comprar los modelos actuales.

Esos primeros años de los noventa también son recordados por el lanzamiento en 1991 de la Kawasaki KMX, que revolucionó el mercado de las motos de 125cc enduro, porque incluía adelantos tecnológicos nunca antes vistos en las motos de producción nacional, como por ejemplo refrigeración por agua, que le daba un sonido completamente distinto respecto a los tradicionales motores de 2 tiempos refrigerados por aire, frenos de disco en las dos ruedas, y válvula de escape KIPS que se abre a altas revoluciones incrementando la potencia. Esto, unido a una estética muy moderna y a una potencia de 24 HP – muy superior a la de sus competidoras - hizo que fuera el objeto más deseado de todos los aficionados al motociclismo. Claro que también costaba más que las otras motos de su segmento, por lo cual sólo pudo ser adquirida por los afortunados que podían pagar su precio. Tener una KMX se volvió, en esos primeros años, un signo de status.

En 1995, la industria de la motocicleta nacional vivió un año record en ventas, con más de 132.000 unidades vendidas, pero decayó dramáticamente a partir de 1998 y hasta el año 2000, afectando a todas las ensambladoras. A pesar de esta época de crisis, hubo modelos que se destacaron por su alto desempeño y tecnología, como fue el caso de la Kawasaki Víctor 150cc, 2 tiempos, de corte deportivo, que contaba con 34 caballos, una increíble potencia para su cilindrada y que se ensambló desde 1997 hasta el 2002, así como la Kawasaki K-1 115cc, catalogada como la semiautomática más rápida y deportiva que se ha ensamblado en el país. Estos dos modelos se

dejaron de comercializar básicamente por su alto precio para el momento que vivía el mercado.

Después de la Plus introdujeron otros modelos Bajaj, tales como la Suma, la KTZ 100, la KB 125 y la Zip, que no fueron muy exitosos, hasta que en el 2001 y siguiendo la tendencia mundial hacia las motos de 4 tiempos, se introdujo la Boxer de 100cc, con un precio inferior a todas las motos de su segmento y una alta economía de combustible que la convirtió en una opción muy interesante como moto de trabajo, para los que querían adquirir su primera moto o que estaban pensando en comprar alguna de segunda mano. Su éxito comercial ha sido significativo y durante los tres últimos años ha sido el modelo más vendido de Auteco.

Después de la Boxer se han introducido otros modelos de Bajaj de cuatro tiempos, con muy buena aceptación, como la Legend y su versión hermana la NXT, primera motoneta de cuatro tiempos, la Pulsar 180, que tiene un excelente comportamiento dinámico y una estética impactante y la Avanti 90, que busca ser una opción aún más económica que la Boxer.

Por más de 65 años, Auteco ha desarrollado una inmensa actividad comercial e industrial en muchos campos, pero siempre ha estado ligada a la industria de la moto, porque para ellos más que un negocio ha sido una pasión, algo que se puede constatar cuando se habla con los empleados o con los directivos de esta empresa. Fueron unos de los primeros importadores a nivel comercial y los primeros que las ensamblaron, abriendo el camino hacia la masificación de este tipo de vehículos en nuestro país, con las Lambretta, y en los últimos años han sido los más dinámicos al introducir nuevos modelos y buscar alternativas en este competido mercado. Todo esto, unido a una merecida reputación de respaldo técnico y buen servicio, construido a lo largo de seis décadas, ha hecho que sea una de las marcas más queridas de nuestro medio y con mayor proyección hacia el futuro.

4.1.1.1 Misión de la Compañía

Nuestra misión consiste en servir con entusiasmo, excelencia y dinamismo, las necesidades y expectativas de nuestros clientes internos y externos, mejorando continuamente la tecnología y la calidad de nuestros productos y servicios en las áreas de fabricación y comercialización, de manera que respondan a las necesidades de transporte, recreación, diversión, equipos de trabajo, energía y suministros para terceros, y en las demás actividades a las cuales se diversifique.

Lo anterior, mediante la participación decidida y comprometida en la aventura empresarial y en el desarrollo de los mercados nacionales e internacionales por parte de nuestros trabajadores, accionistas, distribuidores y proveedores, con productos y servicios de alta calidad, competitivos y rentables, todo ello dentro de una cultura profundamente humana, con valores, principios, actitudes y prácticas que satisfagan las necesidades de nuestros clientes y de nuestra gente y que permita a ésta su realización integral en sus dimensiones espirituales, intelectuales, emocionales y materiales, buscando contribuir, sin detrimento del medio ambiente, al progreso económico y social y al mejoramiento de la calidad de vida de la comunidad.

4.1.1.2 Visión de la Compañía

Hacer de Auteco una organización líder que satisfaga las necesidades de sus clientes y, en especial la de dotar de transporte individual a los colombianos, realice su misión y trascienda ésta a través de su participación en el mejoramiento social del país, permitiendo que grupos cada vez mayores de personas ingresen a la vida moderna.

4.1.1.3 Organigrama de la Compañía

Ver Anexo I.

4.1.1.4 La Calidad, un compromiso

AUTECO S.A ha desarrollado un Sistema de Gestión Integral hacia la Calidad y a través de su Misión y Política de Calidad pone de manifiesto su filosofía de mejoramiento continuo. Dicha filosofía es el fundamento del Manual de Calidad, que describe el enfoque del Sistema de Aseguramiento de Calidad y la forma de implementación de cada uno de los requisitos del sistema, traducidos en procedimientos, instructivos y registros que lo soportan, con el fin de tener un criterio unificado para proceder en la elaboración de nuestros procesos y productos; todo ello orientado a la satisfacción de nuestros clientes internos y externos y al desarrollo de la Organización y de nuestra gente.

El Manual de Calidad describe la forma como AUTECO orienta a la organización para cumplir los requerimientos QS9000 y demás requerimientos para enriquecer el sistema de calidad que solicite cualquiera de sus clientes.

Algunas herramientas mencionadas en el manual han sido implementadas a partir del segundo semestre de 1997, por lo que algunos productos desarrollados con anterioridad a esta fecha pueden no cumplir con las mismas y por tanto no posean un registro evidente de su respectiva aplicación.

La casa matriz que produce las motocicletas tiene un decidido compromiso con la calidad, como lo demuestra el hecho de contar con un completo sistema de aseguramiento de la calidad para sus procesos de producción.

El grupo Humano que conforma la compañía está plenamente convencido de la importancia de ofrecer productos de excelente calidad y desempeño en los diversos usos y terrenos. Por ello las motos son sometidas a exigentes exámenes y pruebas de comportamiento en la geografía colombiana antes de salir al mercado.

Prueba de ello, son los viajes por Colombia y otras pruebas realizadas por publicaciones especializadas en este tipo de vehículos para comprobar su funcionamiento en diferentes niveles de altitud, climas y terreno. Sus resultados le han conferido a las motos Auteco una buena calificación para rodar por las distintas vías del país.

4.1.2 Entrevistas con el cliente

Esta fase comienza con reuniones con jefes de los distintos Departamentos de la Organización, elegidos por la Gerencia, para que sus centros tengan presencia en la Intranet con un portal en el que participen de forma activa todos los miembros de la Organización que lo deseen.

Las conclusiones más importantes extraídas en diferentes reuniones sobre la funcionalidad del Portal, son las siguientes:

- Los miembros del Grupo de Trabajo Intranet serán los que tengan la responsabilidad de administración y configuración del sitio Web. Aunque si así lo desean, podrán delegar la responsabilidad en otra persona perteneciente a su mismo grupo laboral.
- Habrá una parte de la información incluida en el portal que será estática (manuales de procedimientos y políticas, organigrama de la empresa, entre otros). Es decir, no cambiará a lo largo del desarrollo. En cualquier caso, deberá facilitarse la modificación de este contenido.

- Las publicaciones estarán divididas en secciones y algunas podrán considerarse interesantes para ser publicadas como noticias de cada Departamento.
- Deberá existir un mecanismo que permita diferenciar entre información pública e información que sólo pueda ser visualizada por los miembros de cada Departamento.
- Los miembros de la comunidad podrán publicar información mediante un foro, sección dedicada a ellos para la difusión y la comunicación de información, gracias al acoplamiento con las herramientas de E-GroupWare.
- Los Jefes de Departamento jugarán un papel importante. Se encargarán de mantener una sección didáctica dentro de la información de su Departamento, en la que se publicarán las programaciones de eventos y noticias, entre otros, que tiene competencia con el Departamento.
- No se accederá a información personal por medio de la Intranet (información de nómina, información de cartera, información de contabilidad, direcciones y teléfonos residenciales, etc.; fundamentalmente, información confidencial de la organización y sus miembros), salvo algunas excepciones.
- Los miembros de la comunidad tendrán la posibilidad trabajar con un mecanismo de comunicación rápido por medio del cual podrán ahorrar tiempo, papel y gasto telefónico, gracias a un chat o Messenger interno.
- Existirá una ayuda FAQ (Frequently Asked Questions) con procedimientos de Sistemas que permitan a los usuarios con problemas técnicos, intentar dar una solución rápida a estos, antes de solicitar el soporte.
- Habrá una agenda compartida por todos los usuarios de la organización, por medio de la cual se conocerá la disponibilidad de tiempo de todos los miembros (aplican restricciones de acuerdo con los miembros, grupos y cargos dentro de la organización).
- Habrá un espacio para todos los miembros de la organización, por parte de Gestión Humana y de Servicios, con la información del Portal de gestión Humana (semanal) y con el Boletín de servicios (mensual) respectivamente.
- Habrá información del Directorio Telefónico Interno de la organización, así como la información de los correos internos y externos de todos los miembros.

- Los miembros de la organización tendrán la posibilidad de hacer solicitudes de servicios, para problemas que no puedan solucionar por ellos mismos, mediante un formulario de requisición.
- Existirá un medio para llevar un control sobre el ingreso de personas a la empresa; este será usado por los guardias de seguridad, porteros y personas encargadas de la logística y desarrollo de la seguridad de la organización.

Esta lista se ha considerado como los requisitos funcionales de la aplicación (no existen requisitos no funcionales que merezca la pena mencionar).

Con posterioridad se celebró una nueva reunión para tratar el aspecto de la presentación y estructura del portal. Las conclusiones más importantes extraídas fueron las siguientes:

- El portal tendrá los colores insignia de la organización, que normalmente están incluidos en su logotipo. Por este motivo, no sería posible configurar el colorido del contenido de cada Departamento.
- Aunque exista la posibilidad de que la interfaz incluya gráficos, es aconsejable permitir que se pueda reducir el número de elementos gráficos para mejorar la rapidez de acceso.
- Las opciones personales de los miembros del portal por Departamento, no se deben mezclar con los elementos de navegación del usuario anónimo.
- La lista de noticias publicadas depende única y estrictamente de cada Departamento.
- El acceso a las opciones debe ser intuitiva.

De estas conclusiones se han extraído los requisitos estructurales y de navegación.

4.1.3 Roles

Antes de hacer un estudio de los casos de uso de la aplicación Web a desarrollar, y en función de las conclusiones extraídas del apartado anterior, se considera que los usuarios estarán divididos en los siguientes grupos (roles):

- a) Usuarios anónimos (U): son los visitantes que no están registrados en el portal.
- b) Administrador (A): es un usuario especial que permite iniciar la fase de configuración, personalización y administración del portal.

- c) Administrador por Departamento (AD): son los miembros de la Organización que tienen a cargo la responsabilidad de administración y configuración del sitio Web. Existe un representante por cada Departamento perteneciente al Grupo de Trabajo Intranet.
- d) Usuario por Departamento (UD): son los miembros de la organización que tienen acceso a la información no pública del departamento al cual pertenecen.



Figura 4.1. Roles para la aplicación³⁴

4.1.4 Modelo de requisitos

Si bien la fase de recolección de requisitos está dividida en la realización de los modelos de casos de uso, de tareas y de usuario, y con el objetivo de simplificar la visión general de la aplicación, se opta por realizar el diagrama de casos de uso inicial y una tabla-esquema con las tareas identificadas (algunas se dividen en sub-tareas), los usuarios implicados y los casos de uso asociados, para posteriormente desarrollar aquellos casos de uso y/o tareas que resulten de especial interés. La numeración de los casos de uso se realiza con el objetivo de simplificar el diagrama inicial de casos de uso mostrado en el Anexo II.

	Caso de Uso	Tarea/s	Roles
1	Activar el Portal	Activar el Portal	A
2	Navegación pública	Navegación de la parte pública	A,U,AD,UD
3	Validación en el sistema	Validación de usuario en el sistema	A,AD,UD
4	Desconexión del sistema	Desconexión del sistema	A
5	Cambiar contraseñas	Cambiar contraseñas	A
6	Configuración presentación	Agregar Logotipos	A
		Agregar diseño de página inicio	
		Personalizar contenido público	

³⁴ Merseguer, J.; Campos, J.: Exploring Roles for the UML Diagrams in Software Performance Engineering, 2003

		Personalizar la interfaz de usuario	
7	Visualizar contenido público	Visualizar el contenido público	A,U,AD,UD
8	Agregar contenido público	Publicar información compañía	A
		Publicar documentación	
		Publicar noticias generales	
9	Establecer parámetros funcionamiento	Establecer parámetros de funcionamiento	A
10	Definir usuarios por área	Agregar un usuario a un área	A
		Establecer roles por área	
		Quitar roles por área	
11	Agregar imágenes compañía	Agregar imágenes de la compañía	A
12	Eliminar imágenes compañía	Eliminar imágenes de la compañía	A
13	Agregar enlaces de interés	Agregar enlaces de interés	A,AD
14	Agregar documentos adjuntos	Agregar documentos adjuntos	A,AD
15	Visualizar lista de noticias	Visualizar lista de noticias	A,U,AD,UD
16	Visualizar noticias	Visualizar noticias	A,U,AD,UD
17	Gestionar lista de noticias	Publicar noticias en el portal	A,AD
		Modificar noticia publicada	
		Eliminar noticia publicada	
18	Visualizar información usuarios	Visualizar la información de los usuarios	A,AD
19	Gestionar lista de usuarios	Registrar usuario en el portal	A
		Modificar información del usuario	
		Eliminar usuarios del portal	
20	Eliminar información obsoleta	Eliminar mensajes antiguos leídos	A
		Eliminar archivos antiguos	
21	Gestión de mantenimiento	Agregar nueva información al portal	A
		Modificar información del portal	
		Eliminar información antigua del portal	
22	Gestionar Agenda Compartida	Agregar Cita a la Agenda Compartida	A,U,AD,UD
		Modificar noticia publicada	
		Eliminar noticia publicada	
		Visualizar la Agenda Compartida	
23	Gestionar Recursos Compartidos	Agregar Recurso Compartido	A
		Modificar Recurso Compartido	
		Eliminar Recurso Compartido	
24	Accionar sobre Reservas de Recursos Compartidos	Reservar Recurso Compartido	A,U,AD,UD
		Editar Reserva de Recurso Compartido	
		Eliminar Reserva de Recurso Compartido	
		Buscar Recurso Compartido	
		Visualizar Información de Recurso Compartido	
		Listar Recursos Compartidos	

Tabla 4.1. Tabla de Casos de Uso³⁵

El siguiente paso consiste en la realización de una ficha para cada tarea de las anteriores.

³⁵ Merseguer, J.; Campos, J.: Exploring Roles for the UML Diagrams in Software Performance Engineering, 2003

TAREA : 1.Activar el portal	
Caso de uso:	Activar el portal
Objetivo:	Establecer el sitio Web como operativo
Tareas asociadas:	Validación de usuario en el sistema

TAREA : 2.Navegación de la parte pública	
Caso de uso:	Navegación pública
Objetivo:	Operar en el sitio Web
Tareas asociadas:	Validación de usuario en el sistema Visualizar el contenido público Establecer parámetros de funcionamiento

TAREA : 3.Validación de usuario en el sistema	
Caso de uso:	Validación en el sistema
Objetivo:	Validar el rol de usuario para permitir la navegación de acuerdo con los permisos del rol
Tareas asociadas:	Establecer parámetros de funcionamiento

TAREA : 4.Desconexión del sistema	
Caso de uso:	Desconexión del sistema
Objetivo:	Cerrar el servicio del sitio Web
Tareas asociadas:	

TAREA : 5.Cambiar contraseñas	
Caso de uso:	Cambiar contraseñas
Objetivo:	Cambiar las contraseñas de los usuarios
Tareas asociadas:	

TAREA : 6.1.Agregar logotipos	
Caso de uso:	Configuración presentación
Objetivo:	Agregar al diseño los diferentes logotipos que caracterizan a la compañía
Tareas asociadas:	Agregar nueva información al portal Agregar imágenes de la compañía Agregar diseño de página de inicio Personalizar contenido público Personalizar la interfaz de usuario

TAREA : 6.2.Agregar diseño de página de inicio	
Caso de uso:	Configuración presentación
Objetivo:	Desarrollar el diseño de la página de inicio del sitio Web
Tareas asociadas:	Agregar logotipos Agregar imágenes de la compañía Personalizar contenido público Personalizar la interfaz de usuario Establecer parámetros de funcionamiento

TAREA : 6.3.Personalizar contenido público	
Caso de uso:	Configuración presentación
Objetivo:	Determinar qué contenido del sitio Web puede ser accedido por un usuario público
Tareas asociadas:	Validación de usuario en el sistema Personalizar la interfaz de usuario Visualizar el contenido público Establecer parámetros de funcionamiento

TAREA : 6.4.Personalizar la interfaz de usuario	
Caso de uso:	Configuración presentación
Objetivo:	Diseñar la interfaz de usuario de acuerdo con el contenido
Tareas asociadas:	Agregar logotipos Agregar imágenes de la compañía

TAREA : 7.Visualizar el contenido público	
Caso de uso:	Visualizar contenido público
Objetivo:	Ver la información catalogada de tipo público
Tareas asociadas:	Navegación parte pública

4.2 Fase de diseño.

La fase de diseño se ha dividido en los aspectos más importantes: el diagrama de clases y el diseño del sitio Web, además de contemplar el acoplamiento con las herramientas de Software libre.

4.2.1 Diseño del sitio Web.

Teniendo en cuenta los diagramas anteriormente realizados, en el diseño del sitio Web tendremos en cuenta:

a) Página inicial.

- La parte superior incluirá logotipos, enlaces a los diferentes Departamentos de la organización y un enlace a la página Web de la Organización.
- El resto de la pantalla quedará dividido en dos partes. La parte izquierda estará dedicada a la navegación. Incluirá la fecha y una parte donde se validará el ingreso al área restringida de cada Departamento.

- La parte derecha se dedicará a visualizar las páginas. Inicialmente, los datos del Departamento con la información más relevante y representativa de cada uno.
- En la portada aparecerá un mensaje de bienvenida, además de información de carácter general que irá cambiando periódicamente.

b) Navegación. Decisiones tomadas:

- En la parte superior de la página de inicio se dará la bienvenida al usuario.
- El menú de navegación se dividirá en dos partes: el contenido del sitio Web y una parte restringida para cada Departamento.
- Las opciones de contenido varían de acuerdo con el usuario. Únicamente tendrá acceso al contenido asignado a su perfil.
- Siempre existirá un enlace para validarse o desconectarse de la parte restringida del portal.
- Siempre existirá un enlace en la parte superior de la navegación y en el menú de contenidos que facilite el acceso a la página de inicio.
- Las opciones que aparezcan en el menú de navegación del contenido del portal estarán determinadas por el rol del usuario conectado.
- La Figura del Anexo III muestra los menús de navegación personales para los distintos tipos de roles de usuario.

c) Interfaz de usuario. Se toman las siguientes decisiones:

- Para guardar la homogeneidad en las fuentes, colores y estilos, todas las hojas de la intranet harán referencia a las hojas de estilo (CSS) de navegación.
- Sin olvidar las hojas de estilo, cada Departamento de la Organización podrá hacer uso de las imágenes que desee para diseñar su propio espacio dentro de la Intranet.

4.2.2 Acoplamiento con otras herramientas.

Teniendo en cuenta lo mencionado anteriormente con respecto a las diferentes herramientas utilizadas para el desarrollo de la Intranet, en el siguiente capítulo se detallará la instalación y configuración de cada una de las herramientas para el correcto acoplamiento y funcionamiento de las mismas.

4.3 Fase de implementación.

En los procesos de desarrollo de software, la fase de implementación es una fase en la que fundamentalmente se traslada la fase de diseño al lenguaje de programación elegido. En este caso, la fase de implementación lleva asociadas determinadas particularidades que condicionan la fase. En primer lugar, es una aplicación cliente-servidor desarrollada para la Web en lenguaje PHP y en segundo lugar, la aplicación se implementa sobre un servidor de aplicaciones Linux.

Las características que se consideran más importantes para destacar en esta fase son: la persistencia de la información, la estructura del sitio y la seguridad.

4.3.1 Persistencia de la información.

Este punto es uno de los más delicados de la aplicación ya que, en principio, podría parecer razonable que el servidor de aplicaciones asumiera toda la responsabilidad de almacenamiento de la información. En cambio, existen unas premisas obtenidas en la fase de captura de requisitos que obliga a tomar una decisión alternativa. Como se expone en los apartados 3.3.2, 3.3.3 y 4.1.2, el acoplamiento de la Intranet con algunas herramientas de E-Groupware, las cuales permitirán aprovechar el uso de servicios como el Chat y el Foro, nos centran a almacenar la información en una base de datos MySQL, y el uso del esquema LDAP, a almacenar la información de los usuarios.

Anteriormente, hemos visto la evolución de la creación de páginas HTML dinámicas a lo largo de la historia. Inicialmente se usó programas C o Shell-Scripts que devolvían información en hipertexto por su salida estándar. Posteriormente se descubrió que el lenguaje interpretado PERL era excelente para la labor de interpretación necesaria para la programación dinámica. Pero la aparición de soluciones más adecuadas y sencillas hace que PHP se convierta en la mejor opción actual para multitud de necesidades. Por esta razón se permite citar algunas de sus ventajas:

- Muy sencillo de aprender.
- Similar en sintaxis a C.
- Soporta en cierta medida la orientación a objetos, clases y herencia.
- El análisis léxico para recoger las variables que se pasan en la barra de dirección lo hace PHP de forma automática, librándose el usuario de tener que separar las variables y sus valores.

- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente (como se hace con Java Script) evitando así, el chequeo de no recibir solicitudes adulteradas.
- Viene equipado con un conjunto de funciones de seguridad, que previenen la inserción de órdenes dentro de una solicitud de datos.
- Se puede hacer de todo lo que se pueda transmitir por vía HTTP.

Cabe agregar que PHP no soporta directamente punteros, como C, de forma que no existen los problemas de depuración provocados por éstos. Se pueden hacer grandes cosas con pocas líneas de código, lo que hace que merezca la pena aprenderlo. El código PHP es mucho más legible que el de otros lenguajes. Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-Commerce, XML, creación de PDF). Al poderse encapsular dentro de código HTML se puede recoger el trabajo del diseñador gráfico e incrustar el código PHP posteriormente. Es multiplataforma, ya que funciona en todas las plataformas que soporten apache, y quizás la más importante de las ventajas, es software libre. Se puede obtener en la web y su código está disponible bajo la licencia GPL³⁶.

Este hecho, ha resultado determinante para decidir qué parte de la persistencia se delegará en la base de datos MySQL compatible en diseño con la base de datos de la aplicación, qué parte en el directorio LDAP y qué parte en el servidor como tal. La justificación se basa en dos razones fundamentalmente: por un lado, se evitará que determinada información se tenga que introducir por varias veces (datos de los usuarios, Departamentos de la Compañía, números telefónicos, entre otros), ya que podrían generar inconsistencias; y por otro lado, se facilitará el desarrollo de páginas complementarias o alternativas en lenguajes de scripting como PHP o ASP.

En el Anexo IV se muestra el diseño de la base de datos para ambos casos. Es necesario agregar que la información generada por la publicación de información en el portal (noticias, boletines, materiales de trabajo, avisos, etc.) se almacenan como objetos del sistema en el servidor, como se explica en el siguiente apartado.

³⁶ GNU GPL, *General Public License – Free Software Foundation*

4.3.2 Estructura del sitio Web.

La estructura del sitio ha sido diseñada teniendo en cuenta las características del servidor donde se ha instalado. Las claves que se han tenido en cuenta para el diseño de esta estructura han sido las siguientes:

- Con objeto de facilitar el sistema de navegación y, sobre todo, facilitar la inclusión de nuevos elementos sin tener que modificar las páginas que presentarán los menús de contenido, se decide considerar cada enlace en la navegación del Web como un objeto carpeta.
- Tras tomar la decisión en la fase de diseño que cada página del sitio estaría dividida en tres secciones (superior, izquierda y derecha), se implementa utilizando marcos (frames) que facilitan la presentación del contenido. En cambio, la utilización de marcos lleva asociado un problema: es posible que la tarea asociada a un enlace necesite la recarga de otro(s) marco(s) diferente(s) del destino del enlace. Para solucionarlo, y en aras de facilitar la reutilización, el mantenimiento y dar genericidad a los enlaces, se ha tomado la decisión de que cada enlace en la navegación apunte a la página raíz y se le pase el parámetro de lo que se visualizará: el menú de dónde se encuentra en el marco izquierdo y el contenido que se visualizará en el marco derecho. Para conseguir esta reutilización y genericidad, a cada objeto susceptible de ser navegado se le incluye estas dos propiedades: índice y contenido, que representan la página de navegación asociada y la página de contenido respectivamente. En realidad, no todos los incluyen.
- En las aplicaciones Web que utilizan lenguajes de scripting de lado servidor tradicionales, los diseñadores suelen crear una carpeta donde incluyen todos los scripts que se usarán. En este caso, no resulta conveniente, ya que, una carpeta es un objeto que representa una parte del diseño, los objetos tienen propiedades (atributos) y métodos (operaciones) y los métodos (scripts) deben estar dentro de su objeto. Este es el motivo de que los scripts no estén incluidos en una única carpeta, sino situados estratégicamente en el lugar que les corresponde (su objeto propietario). Por ejemplo, existe un método `visualizar_noticia` genérico y luego existe uno para visualizar recursos, materiales, citas, etc., cada uno situado en su objeto correspondiente.
- También ha resultado determinante la seguridad. La navegación privada se ha situado en un lugar jerárquicamente independiente de la pública. Así, con

definir los permisos apropiados en la raíz de parte privada, queda garantizada la seguridad de acceso a esa parte. Ver apartado siguiente.

- Los objetos del diagrama de clases que se decide que su persistencia se encuentre en el servidor, son, como se explicó, las publicaciones. Se decide que cada publicación se almacene en su contexto natural (por ejemplo, los materiales de trabajo publicado por cada Departamento, se almacenan en el objeto correspondiente a su página propia). Los manuales de procedimientos se almacenan en una carpeta llamada Manuales para facilitar su presentación. El Anexo V muestra la estructura del sitio (sin publicaciones, departamentos, páginas personales, etc.).

4.3.3 Seguridad.

Las aplicaciones Web necesitan seguridad. Es necesario controlar quién accede a la aplicación y saber qué hace. La seguridad es un elemento imprescindible en el diseño de las aplicaciones Web. Debe proporcionar un mecanismo de privacidad de la información y de protección contra acciones malintencionadas que vulneren la integridad de la aplicación. El método de seguridad para controlar los accesos a una carpeta es el mismo que el que asegura la administración de cualquier aplicación del servidor; esto da una idea de la potencia de la seguridad en las aplicaciones acopladas con el servidor HTTP Apache.

Cuando un usuario intenta acceder a un recurso protegido, Apache obliga al usuario a facilitar un nombre de usuario y una contraseña que permita el acceso al recurso.

La seguridad, en general, lleva asociados dos conceptos: autenticación y autorización. Autenticación significa saber quién es el usuario (evaluación en Apache y LDAP) y autorización significa determinar que está haciendo (evaluación en E-Groupware). E-Groupware proporciona facilidades separadas para manejar el proceso de la administración de los permisos que les permiten o impiden realizar acciones.

Debido a que la seguridad se ha implantado basándose íntegramente en las características del servidor HTTP Apache, en la configuración individual del E-Groupware y en las características de autenticación del LDAP, en el siguiente capítulo se hará reseña de las configuraciones necesarias para certificar la Intranet-Auteco como una Aplicación Web segura.