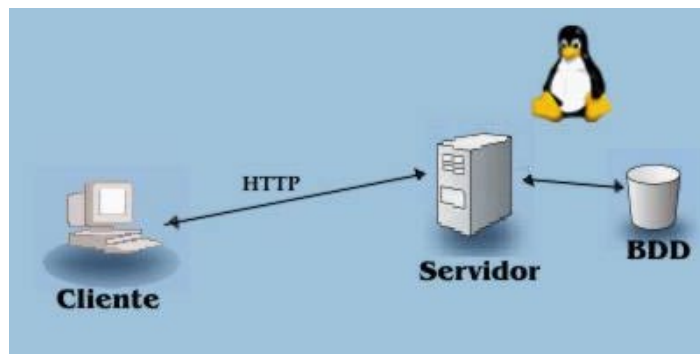


# Capítulo V

## Desarrollo de aplicativos web en la plataforma Linux



**Aplicaciones web**

**Sistema operativo Linux**

**Servidor Web Apache**

**Bases de Datos freeware en Linux**

**Servidor Jakarta-Tomcat**

**Estructura de publicación Cocoon**

**Software freeware XML en Linux**

En el desarrollo de aplicaciones con software libre, existe una idea fundamental, ésta consiste en que los desarrolladores puedan adoptar la plataforma Linux como una alternativa muy económica para sus sistemas de información, aprovechando las características importantes de Linux, que consisten en proveer sus servicios en forma libre, desde bases de datos, web, correo electrónico, paredes de fuego (seguridad o firewall), entre otros.

Entre las aplicaciones que se podrían implementar en la plataforma Linux, se encuentran los aplicativos web que son una herramienta de Internet que está en constante desarrollo, el avance de las tecnologías han permitido que los usuarios del Internet se comuniquen de manera tal que la interacción sea más real a través de la Web. De este modo, no sólo se ha extendido el uso de la misma, sino que nuevas funcionalidades han poblado los sitios web.

Cuando se plantea la realización de una aplicación web uno de los factores fundamentales a tener en cuenta son los recursos financieros disponibles para hacer frente al proyecto. En el caso de que dichos recursos sean escasos (o nulos), la opción respecto al software está clara: hay que decidirse por utilizar software libre.

Tradicionalmente se ha asociado software libre con software de baja calidad, pensando que no ofrece ninguna garantía porque no ha pasado por controles de calidad, no ofrece soporte, etc. Sin embargo, hoy en día hay muchas razones que obligan a replantearse la certeza de tal afirmación.

Este capítulo se enfoca en el desarrollo de aplicativos web en el sistema operativo Linux, centrándonos de manera específica en herramientas software necesarias para desarrollar una aplicación web basada en XML/XSL.

## **5.1. Aplicaciones web**

La Web en sus orígenes fue pensada como un medio para desplegar información, ésta reposa de manera estática en los servidores la cual es accedida a través de una consulta hecha por un navegador valiéndose del protocolo HTTP. Actualmente se maneja el mismo concepto en la comunicación cliente-servidor (navegador-servidor web) sólo que no necesariamente el resultado de la comunicación debe provenir de la carga de una página estática, esta puede ser el resultado de la ejecución en el servidor de alguna lógica de programación.

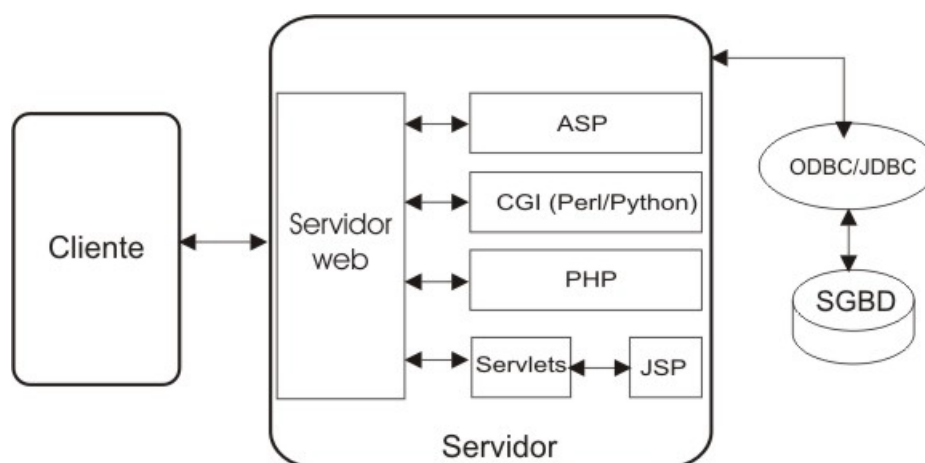
Una aplicación web es una aplicación que utiliza el Protocolo de Transferencia de hipertexto o HTTP como principal protocolo de comunicación y de intercambio de información entre un cliente y un servidor. Cuando un navegador web envía una petición a un servidor web, éste último se encarga de buscar la página solicitada en su sistema de archivos y devolverla al navegador. Sin embargo, lo que devuelve el servidor web no tiene por qué ser siempre una página HTML estática almacenada en el servidor, ya que puede tratarse de la salida de un programa que se ejecuta en el entorno del servidor web.

### 5.1.1. Recuento de las tecnologías para el desarrollo de aplicaciones web <sup>[WWW015]</sup>

Para el desarrollo de aplicaciones web se han generado múltiples tecnologías entre ellas se encuentran:

- **CGI.** Common Gateway Interface: Fue la primera técnica utilizada para que el contenido de las páginas web se generará de manera dinámica, es común encontrar en los diferentes servidores web el módulo que soporta la ejecución de CGIs. CGI es un mecanismo de comunicación entre el servidor web y una aplicación externa, ésta aplicación puede estar desarrollada en casi cualquier lenguaje, éste sólo debe cumplir la condición de ser soportado por el servidor HTTP. Este mecanismo tiene falencias que evita su uso a gran escala, la más conocida es en cuanto a rendimiento, ya que por cada petición que se realice en el servidor se crea un nuevo proceso, lo cual tiene un costo muy alto en lo que a recursos del sistema se refiere.
- **Fast-CGI.** Esta es una solución similar al CGI mencionado anteriormente, sólo que propone la creación de un sólo proceso persistente por cada programa FastCGI en lugar de por cada solicitud del cliente. Es una solución viable pero también tiene inconvenientes de proliferación de procesos en el caso de peticiones concurrentes.
- **Páginas dinámicas en servidor.** Con la aparición de esta tecnología se entra a una nueva forma de trabajo, la cual está orientada al trabajo del diseñador web, quien no necesariamente conoce de lenguajes de programación. Este nuevo enfoque consiste en insertar pequeños fragmentos de lógica de programación en la estructura HTML de la página, al contrario de lo que se hacía en los CGIs, que era en el lenguaje de programación utilizar sentencias de impresión para generar salidas HTML. En este sentido se conocen diferentes alternativas, entre ellas podemos mencionar PHP, ASP, JSP, entre otros.

- **Servlets.** Un servlet se puede definir como un programa independiente de plataforma que aporta la misma funcionalidad a la programación en el lado del servidor que tradicionalmente ha realizado la interfaz CGI. El servlet puede considerarse como una evolución de los CGIs desarrollada por SUN Microsystems como parte de la tecnología JAVA. De forma general consiste en la ejecución de aplicaciones Java en el motor de servlets (Servlet engine) el cual hace parte del servidor web. Esta tecnología hace parte de la arquitectura propuesta por SUN en su plataforma J2EE (Java 2 Enterprise Edition).
- **Servicios Web.** La arquitectura de servicios web plantea algo más que una técnica para el desarrollo de aplicaciones web, representa un modelo de computación distribuida para Internet basado en XML. Bajo este concepto ya no sólo se trata la comunicación usuario-aplicación, sino que de manera adicional se maneja la interacción aplicación-aplicación. Para aclarar un poco más el concepto se puede tomar como ejemplo una rutina de programación, una rutina es como una caja negra, la cual encierra un proceso y cumple una función claramente definida, luego para construir una aplicación se llama a dichas rutinas enviando parámetros y recibiendo la respuesta respectiva. Un servicio web se puede considerar como una rutina a la cual se le envían los parámetros utilizando XML encapsulados en el protocolo HTTP.



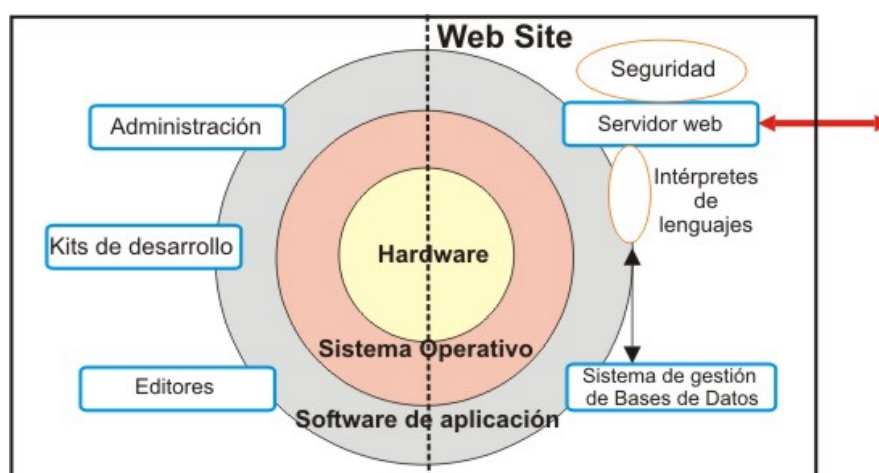
**Figura 5.1.** Distintas posibilidades tecnológicas para el desarrollo de una aplicación web.

### 5.1.2. Elementos software para la construcción de una aplicación Web

Para la implantación de una aplicación web deben considerarse los distintos elementos software que permitirán configurarla con todas las funcionalidades requeridas. Estos elementos software pueden formar parte del sistema operativo o del software de aplicaciones que se debe añadir al mismo para alcanzar las funcionalidades requeridas.

Normalmente se piensa en una aplicación web, desde el punto de vista arquitectónico, como un programa que se encarga de servir páginas HTML. Pero con esta simple funcionalidad (como ocurría en los inicios del web) no sería posible alcanzar los objetivos que requiere hoy en día un sitio web (adaptación de contenidos al usuario, intercambio de información con el cliente, computación en la parte del cliente o del servidor, etc). Para ello se debe disponer de una plataforma más compleja, proporcionada bien por el propio sistema operativo, el servidor web o aplicaciones adicionales que interactúen con ambos proporcionando los servicios adicionales (acceso a datos, desarrollo de lógica de control, generación dinámica de contenidos, etc.). <sup>[www016]</sup>

Las prestaciones del sistema operativo, del servidor web, o de módulos adicionales, la integración entre ellos, su rendimiento en conjunto, así como el costo de todos los elementos software necesarios son criterios a considerar a la hora de seleccionar la plataforma. <sup>[www016]</sup>



**Figura 5.2.** Elementos software necesarios para el desarrollo de una aplicación web.

A continuación se describirán los elementos necesarios para el desarrollo de una aplicación web, considerando la plataforma Linux.

## 5.2. Sistema Operativo Linux

Linux es un sistema operativo gratuito y de libre distribución inspirado en el sistema Unix, escrito por Linus Torvalds con la ayuda de miles de programadores en Internet. Unix es un sistema operativo desarrollado en 1970, una de cuyas mayores ventajas es que es fácilmente portable a diferentes tipos de ordenadores, por lo que existen versiones de Unix para casi todos los tipos de ordenadores, desde PC y Mac hasta estaciones de trabajo y supercomputadores. Al contrario de otros sistemas operativos, Unix no está pensado para ser fácil de emplear, sino para ser sumamente flexible. Por lo tanto Linux no es en general tan sencillo de emplear como otros sistemas operativos, aunque, se están realizando grandes esfuerzos para facilitar su uso. Pese a todo la enorme flexibilidad de Linux y su gran estabilidad (y el bajo coste) han hecho de este sistema operativo una opción muy a tener en cuenta por aquellos usuarios que se dediquen a trabajar a través de redes, naveguen por Internet, o se dediquen a la programación. Además el futuro de Linux es brillante y cada vez más y más gente y más y más empresas (entre otras IBM, Intel, Corel) están apoyando este proyecto, con lo que el sistema será cada vez más sencillo de emplear y los programas serán cada vez mejores.

### 5.2.1. Características y Ventajas de Linux

Linux es un sistema operativo robusto y suficientemente completo para manejar grandes tareas, a continuación listaremos algunas de sus características y ventajas más relevantes <sup>[LIB008]</sup>:

- Linux es un sistema operativo versátil, los sistemas Linux pueden constituir una alternativa para las empresas con necesidades informáticas específicas. Un servidor Linux puede reducir costos de hardware y software en comparación con el costo licencia de muchos sistemas operativos comerciales.
- La disponibilidad del código fuente del núcleo de Linux, permite a los programadores localizar los errores y proporcionar parches al núcleo, en consecuencia el núcleo de Linux es robusto desde su desarrollo inicial y con frecuencia se considera más sólido que muchos sistemas operativos comerciales.
- Linux es de fácil configuración, las distribuciones Linux se suministran completamente preconfiguradas e incorporan utilidades e instaladores de configuración gráfica.

- Linux dispone de algunas interfaces gráficas configurables (XWindows), además dispone de dos administradores de ventanas Gnome y Kde que toman diferentes aspectos pero tiene la misma funcionalidad de enseñanza.
- El sistema operativo Linux tiene una gran reputación por ser muy rápido, es eficaz en la gestión de memoria, potencia de CPU y espacio en disco.
- Linux como servidor web resulta mejor y más fiable, ya que en la actualidad cientos de proveedores de servicios de Internet (ISP), laboratorios informáticos universitarios y algunas grandes empresas usan servidores Linux.
- Linux es un sistema operativo multitarea y multiusuario, pueden trabajar varios usuarios simultáneamente en él y cada uno de ellos pueden tener varios programas en ejecución.
- La compatibilidad de redes Linux es superior a algunos de los sistemas operativos comerciales, Linux permite conexión a Internet y a otras redes que usan TCP/IP o IPX vía Ethernet, ATM, MODEM, protocolo X25, RDSI o tokenring.
- Linux se integra fácilmente con redes de área local LAN independientemente del sistema que se ejecute, Linux es compatible y se integra con Macintosh, DOS, Windows 3x, 9x, NT/2000, Novell y OS/2 mediante protocolos nativos de cada sistema operativo.
- El sistema de gestión de memoria virtual en Linux posee un mecanismo de seguridad interno, ningún programa tiene acceso al espacio de memoria asignado a otro proceso, así el sistema es más seguro y estable ya que un proceso no interfiere en el segmento de memoria de otro proceso y algún problema en un proceso no afectará a ningún otro.

### **5.3. Servidor Web Apache**

Una parte importante de cualquier aplicación web es, con toda seguridad, el servidor web; el software encargado de atender las peticiones de los clientes y enviarles las páginas web solicitadas. Existen multitud de paquetes software para montar servidores web, muchos de ellos distribuidos como software libre y siendo, sin lugar a dudas, el más popular de todos Apache.

El servidor Apache es el producto de un equipo de desarrolladores que, de forma desinteresada, construyen un servidor HTTP robusto, de alta calidad y disponible de

forma libre. Apache es el servidor web más popular en el Internet debido a que es altamente configurable, extensible, gratuito y su código fuente es libre. Apache también es un servidor relativamente seguro, ya que si se descubre un agujero los parches para solucionarlos han estado en Internet casi inmediatamente. Apache viene incluido en casi todas las distribuciones Linux. <sup>[WWW017]</sup>

### 5.3.1. Características de Apache

Apache es un servidor web flexible, rápido y eficiente. Entre sus características destacan:

- Multiplataforma
- Es un servidor web conforme al protocolo HTTP/1.1
- Modular: Con los diferentes módulos de apoyo que proporciona puede adaptarse a diferentes entornos y necesidades, y con la API de programación de módulos desarrolla módulos específicos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: por ser modular se han desarrollado diversas extensiones (añadir módulos) entre las que destaca PHP, un lenguaje de programación del lado del servidor.

### 5.3.2. Ficheros de Configuración de Apache

Los ficheros de configuración de Apache están por defecto en el directorio `/usr/local/apache/conf` aunque esto es algo configurable, y estos son:

- `httpd.conf`: fichero principal de configuración de Apache.
- `srm.conf`: fichero de definición del espacio de nombres, en este fichero se especifica donde se encuentran los `cgi-bin`, los iconos, el tipo de documento por defecto, como se responde ante los errores, que fichero es el índice dentro de un directorio, donde está la página personal de los usuarios del sistema.
- `access.conf`: fichero de control de acceso global a los datos del servidor web. En él se especifica los permisos de accesos a directorios, ficheros y URLs dentro del servidor, así como diferentes configuraciones.



El servidor Apache es relativamente seguro, pero hay métodos para configurarlo de la forma más segura, la configuración de Apache se establece en el fichero `httpd.conf`, este fichero contiene una serie de directivas que rigen el comportamiento de Apache.

Siempre que se instale Linux, vendrá con una configuración predeterminada, los hackers conocen estas configuraciones y saben como obtener información del servidor web y explotar sus vulnerabilidades, entonces lo primero que hay que hacer es revisar la configuración predeterminada y desactivar todas aquellas opciones que no sean necesarias.

Restringir al servidor web para que sólo acceda a los ficheros que están dentro del árbol de documentos web es la mejor estrategia en lo que a seguridad respecta, pero puede configurarse para que pueda seguir enlaces simbólicos que apuntan hacia el exterior del árbol de documentos.

El servidor web Apache maneja un demonio denominado `httpd`, hay varias copias de `httpd` ejecutándose, lo que garantiza que pueda procesarse más de una conexión concurrentemente. Apache tiene que ser ejecutado por el usuario `root`, porque se engancha al puerto 80 (*puerto HTTP*), pero una vez arrancado puede cambiar el usuario de ejecución, normalmente es el usuario `nobody` el que ejecuta los procesos `httpd`, aunque puede ser cualquier usuario creado para este propósito.

## **5.4. Bases de Datos en Linux**

### **5.4.1. Acceso a base de datos**

Gran parte de las aplicaciones web sirven como interfaz hacia un almacén de datos, ya que permiten obtener y escribir datos de una manera intuitiva. Para aplicaciones que requieren manipulación compleja de los datos, control de acceso concurrente de usuarios, mayor estabilidad, control de integridad de datos, entre otros, es imprescindible el uso de un Sistema de administración de bases de datos o DBMS con soporte para SQL.

Un DBMS “es un sistema de software de propósito general que facilita el proceso de definir, construir, y manipular bases de datos para diferentes aplicaciones”. <sup>[WWW018]</sup>

Independientemente de qué DBMS se elija y de cómo trabaje, lo que se busca es que el manejador provea una API independiente del manejador. Entre las APIs independientes de manejador más comunes están ODBC, JDBC y DBI.

El papel de una API independiente del manejador es crear una interfaz común para los diferentes manejadores. De esta manera, una aplicación programada que usa una API independiente para comunicarse con la base de datos, puede migrarse fácilmente a otro manejador, siempre y cuando éste cuente con soporte para la API independiente para la que fue creada la aplicación.

#### **5.4.2. PostgreSQL**

PostgreSQL es un Sistema de Gestión de Base de Datos objeto-relacional e incluye conceptos como clases, herencia, tipos y funciones, y tiene características de servidor que aportan potencia y flexibilidad como: Triggers, subselects, Constrains, Foreigns Keys, Replicación, reglas e Integridad transaccional y referencial.

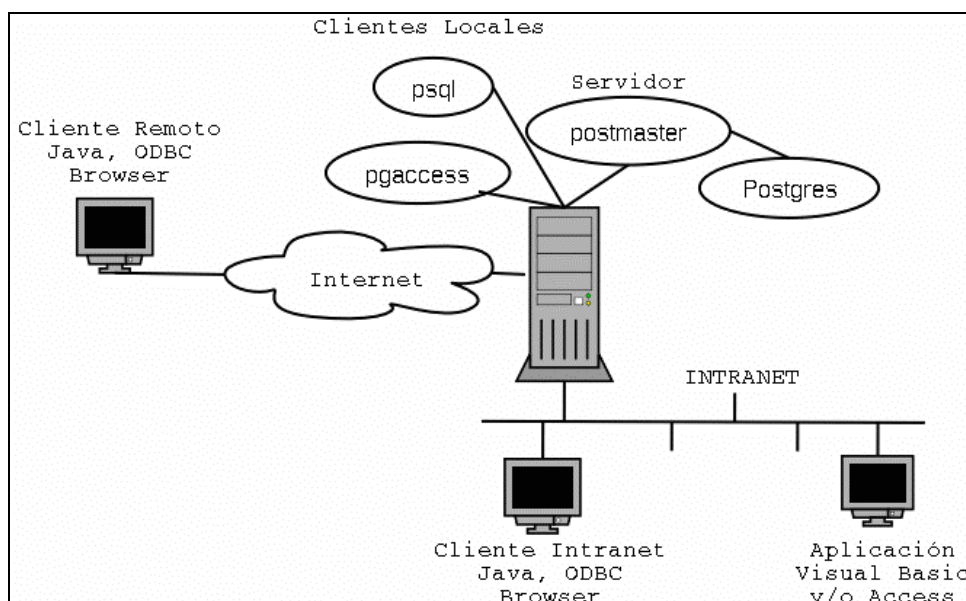
Los tipos de datos que soporta son: datos de carácter, datos de amplio rango, datos tipo date/time, float, serial, money y boolean. Los datos almacenados en la base de datos están protegidos contra escritura para cualquier cuenta que no sea el superusuario de postgres.

PostgreSQL es un Motor de Bases de Datos con todas las características y ventajas de un motor comercial. PostgreSQL soporta conectividad ODBC para ser utilizado con aplicaciones Windows y JDBC para aplicativos Java. PostgreSQL ya viene incluido en las versiones de Linux Red Hat 7.1 en adelante, debido a que ofrece estabilidad, eficiencia y facilidad de uso.

##### **5.4.2.1. Arquitectura de Postgresql**

PostgreSQL implementa una arquitectura cliente/servidor lo que facilita su administración y configuración y mejora su desempeño.

Las conexiones de los clientes al servidor de bases de datos se realizan mediante sockets Unix locales, sockets TCP/IP; y para conexiones no locales se debe inicializar el demonio potmaster con la opción -i para conexiones a Internet.



**Figura 5.3.** Arquitectura de PostgreSQL [WWW019]

En el servidor, PostgreSQL cuenta con un proceso postmaster, que se encarga de recibir las conexiones y cada vez que recibe una, crea una instancia del motor postgres, para procesar todos los requerimientos del cliente. A cada usuario de postgres se le asigna un nombre de usuario y contraseña, por defecto los usuarios no tiene permiso de escritura a bdd que no hayan creado.

Postgres cuenta con una interfaz de comandos llamada psql, que le permite al usuario ejecutar cualquier sentencia SQL, así como administrar la Base de Datos y las tablas. El lenguaje de consulta que soporta está basado en el lenguaje estándar SQL3.

Para el desarrollo de aplicativos cliente ofrece una interfaz gráfica para el desarrollo de consultas como es pgaccess, que permite realizar las funciones de administración, crear funciones, formas y reportes. PostgreSQL cuenta con interfaces de programación para los lenguajes: C, C++, Perl, PHP, Python y Java.

#### 5.4.3. MySQL [WWW020]

MySQL es un Sistema de Gestión de Base de Datos, es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños (siempre, claro está, comparada con las de su categoría). Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hacen de MySQL altamente conveniente para acceder a bases de datos en Internet.

Entre sus principales características se puede mencionar:

- El principal objetivo de MySQL es velocidad y robustez.
- Escrito en C y C++, testado con GCC 2.7.2.1.
- Clientes C, C++, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y S.O. distintos.
- Sistema de contraseñas y privilegios muy flexible y seguro.
- Todas las palabras de paso viajan encriptadas en la red.
- Registros de longitud fija y variable.
- Todas las columnas pueden tener valores por defecto.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintas lenguas.
- Todos los comandos tienen -help o -? Para las ayudas.
- Diversos tipos de columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ODBC para Windows 95 (con fuentes), se puede utilizar ACCESS para conectar con el servidor.

#### **5.4.4. Evaluación de SGBDD freeware** <sup>[LIB009]</sup>

Para el uso de SGBDD (Sistemas de Gestión de Base de Datos) freeware, se debe seguir las siguientes actividades.

1. Localización del SGBDD freeware, estableciendo sitios donde se encuentra disponible el software, manuales, etc.
2. Selección del SGBDD freeware, haciendo una revisión de contexto global y necesidades particulares del proyecto a desarrollar.
3. Obtener los SGBDD freeware, el software se puede obtener principalmente de Internet o de CDRoms.
4. Conocimiento del Funcionamiento de los SGBDD freeware, mediante el autoaprendizaje.

5. Evaluación de los SGDBB freeware, obtener información para tomar decisiones, considerando prestaciones, fácil instalación y uso.

6. Documentación de los SGBDD freeware, Manuales, libros, etc.

Tomar una decisión de uso de un SGBDD es algo crítico, primero se debe asegurar que el SGBDD que se elija, permita hacer lo que se desea durante toda la vida del producto; esto significa que no sólo va almacenar datos sino que disponga de recursos necesarios para resolver problemas complejos de actualización, seguridad, escalabilidad y portabilidad que exige la tecnología de información actual.

Medir el rendimiento no es algo simple, para tomar la decisión correcta se necesita comprender lo que se está midiendo. Para seleccionar cual SGBDD es el que mejor se acoge a nuestras necesidades será necesario primeramente establecer parámetros o factores de evaluación como:

- Prestaciones del producto
- Prestaciones del Motor
- Soporte Técnico
- Interfaz de Usuario
- Desarrollo, Control y Administración de BDD
- Compatibilidad SQL
- Soporte de Programación
- Capacidad de respuesta.

Los SGBDD Linux tienen características como transacciones, subselecciones, triggers, listas y bloques sofisticados y algunos ofrecen Integridad Referencial de claves foráneas, pero día a día se están mejorando las versiones subsiguientes, reparando fallas y adicionando nuevas características.

La falencia de los SGBDD freeware para Linux es la no existencia de herramientas visuales para el desarrollo de aplicaciones cliente y herramientas case para la construcción de Base de Datos.

#### **5.4.4.1. Selección del SGBDD freeware**

Los Sistemas de Gestión de BDD que ofrecen mayores prestaciones para trabajar eficientemente son PostgreSQL y MySQL por lo que ofrecen más soporte técnico, documentación y facilidad de instalación; por esto los compararemos:

- Postgresql tiene características como subselects, disparadores, vistas y bloques que MySQL no posee.
- Postgresql y Mysql ofrecen Integridad Referencial de Claves.
- Postgresql y Mysql generan código estable y probado que tiene un mínimo de errores.
- Ambos SGBDD implementan algunas funciones y tipos del estándar ANSI SQL.
- Ofrecen APIs que son transparentes durante la conexión, estos son: C/C++, TCL, PHYTON, PERL, ODBC, JDBC y soporte PHP. Los drivers JDBC permiten la conexión desde aplicativos con Java-XML.
- Mysql tiene mayor capacidad de almacenamiento para bdd, y Postgresql tiene un límite de 8k por registro, reduciendo el rendimiento.
- Postgresql esta tomando auge en Internet como servidor de datos para la web debido a las múltiples conexiones simultáneas que soporta.
- Ambas tienen ayudas y documentación en Internet y un site donde se puede encontrar cualquier información referente a estos SGBDD.

Aquí conviene resaltar que la elección de uno u otro SGBDD vendrá dada por las necesidades específicas del web que se quiera implementar, y en muchos casos será una decisión “subjetiva”.

Las opciones principales, en Linux, son *MySQL* y *PostgreSQL*; la diferencia fundamental entre ambos SGBD's es la implementación de SQL y la velocidad, si se necesita rapidez y características sencillas la elección es *MySQL* mientras que en el caso de sistemas complejos en los que el rendimiento es una cuestión secundaria debería optarse por *PostgreSQL*.

En las últimas versiones de ambos SGBDD, se está tendiendo hacia un punto común. *MySQL* está incorporando transacciones e intentando ser 100% ANSI 92 compatible, mientras que *PostgreSQL*, además de ciertas mejoras, como el aumento del tamaño máximo de una fila, está poniendo especial énfasis en mejorar el rendimiento, habiendo conseguido en su versión 7.1.x grandes mejoras en este terreno.

El software libre ofrece muchas posibilidades dentro de los SGBDD's, pudiendo encontrarse sistemas completos que cumplan todos los requisitos de cualquier proyecto. Además, las perspectivas de futuros desarrollos no hacen sino anunciar una mayor cantidad de prestaciones de todos estos sistemas, así como mejores rendimientos. Por

todo ello, se puede afirmar que es posible optar por un SGBDD libre en lugar de uno propietario sin temor a ninguna merma de funcionalidad.

### **5.5. Servidor de Servlets Jakarta Tomcat**

En un modelo de co-servidor, otro proceso servidor, además del servidor HTTP, se levanta al mismo tiempo que éste. El conjunto de peticiones definidas por el servidor HTTP son transferidas al otro proceso servidor, que actúa como un mecanismo capaz de abstraer la información HTTP en una serie de facilidades computacionales, tales como manejo automático de sesiones, validación y corrección de datos de entrada y salida, etc.

Estas aplicaciones que se comunican con la capa de software intermedia se les conocen como Servlets, en contraposición a los Applets que se ejecutan en el software cliente. Los Servlets son módulos de código normalmente en lenguaje Java, pero pueden encontrarse en algún otro, que se ejecutan dentro de un servidor de aplicaciones (motor de Servlets) para responder a las peticiones de los sistemas cliente.

Por lo tanto, y debido a su diseño, este modelo tiene las siguientes ventajas:

- El Servlet no corre como un proceso separado, corre dentro del proceso “motor”. Esto elimina el sobreprocesamiento y disminuye el consumo de recursos.
- El Servlet permanece en memoria y solamente existe una instancia de él. Esto ahorra memoria y permite el manejo de datos persistentes.
- El “motor de Servlets” puede actuar como una caja de arena (aplicación que restringe las posibles operaciones que un módulo del programa puede ejecutar), aumentando la seguridad del sistema.

Un servlet es un programa en Java usado para extender las capacidades de un servidor cuyas aplicaciones son accedidas vía un modelo petición-respuesta (request-response). A pesar de que un servlet podría ser usado en cualquier servidor de este tipo, actualmente su uso se ha extendido básicamente a las aplicaciones que se ejecutan bajo un servidor web. En este caso, los servlets se ejecutan dentro del entorno del servidor web, y por lo tanto amplían la funcionalidad del mismo al permitir la generación de páginas HTML de manera dinámica. Estos servlets reciben y responden a peticiones de clientes web a través del protocolo HTTP.

Para que se puedan ejecutar los servlets es preciso tener un contenedor de servlets que trabaje en conjunción con el servidor web. Este contenedor se encarga entre otras cosas de administrar la carga y descarga de los servlets dentro del servidor web.

Tomcat proporciona el contenedor de servlets necesario para el trabajo con los Java Servlets, y además puede ser usado en conjunción con algunos otros servidores web, tales como Apache.

Tomcat es un contenedor de servlets con un entorno JSP, un contenedor de servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario. Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (actualmente se soportan los servidores Apache, IIS y Netscape). Esto significa que siempre que se despliegue Tomcat se tendrá que decidir cómo usarlo, y si se necesita que se instale un servidor web.

### 5.5.1. La Estructura de Directorios de Tomcat

Esta es la estructura de directorios de Tomcat:

Directorio	Descripción
bin/	Incluye los archivos binarios y scripts de Tomcat. Dentro del directorio bin de Tomcat se tienen dos scripts con los cuales se puede iniciar y detener el servidor en el momento que se desee.
common/	Para clases disponibles tanto para Catalina (el nombre de Catalina se refiere al contenedor de servlets en sí que se incluye con Tomcat) como para las aplicaciones web.
conf/	Contiene varios ficheros de configuración como server.xml (fichero de configuración principal de Tomcat) y web.xml (configura los valores por defecto para las distintas aplicaciones desplegadas en Tomcat).
server/	Contiene clases para uso interno y exclusivo de Catalina.
Lib	Contiene varios ficheros jar que son utilizados por Tomcat.
logs/	Guarda los archivos de registros (logs).
work/	Directorio de trabajo para archivos y directorios temporales usado por Tomcat.
webapps/	Directorio base para las aplicaciones web.

**Tabla 5.1.** Directorios de Tomcat <sup>[WWW021]</sup>



### 5.5.2. Los Scripts de Tomcat

Tomcat es un programa Java, y por lo tanto es posible ejecutarlo desde la línea de comandos, después de configurar varias variables de entorno. Pero configurar las variables de entorno es tedioso y propenso a errores. En su lugar, el equipo de desarrollo de Tomcat proporciona unos pocos scripts para arrancar y parar Tomcat fácilmente.

Los scripts son sólo una forma conveniente de arrancar/parar se puede modificarlos para personalizar: el CLASSPATH, variables de entorno, etc.; mientras que se genera la línea de comandos correcta para Tomcat.

La siguiente tabla presenta los scripts más importantes para el usuario común:

Nombre del Script	Descripción
catalina.sh	El script principal. Configura el entorno apropiado, incluyendo CLASSPATH, CATALINA_HOME y JAVA_HOME, y arranca Tomcat con los parámetros de la línea de comandos apropiados.
startup.sh	Arrancar tomcat en segundo plano. Acceso directo para tomcat Start.
shutdown.sh	Detiene tomcat (lo apaga). Acceso directo para tomcat stop.

**Tabla 5.2.** Scripts de Tomcat <sup>[WWW021]</sup>

### 5.5.3. Ficheros de Configuración de Tomcat

La configuración de Tomcat se basa en dos ficheros:

1. server.xml - El fichero de configuración global de Tomcat. Sirve para dos objetivos:
  - Proporcionar configuración inicial para los componentes de Tomcat.
  - Especifica la estructura de Tomcat, lo que significa, permitir que Tomcat arranque y se construya a sí mismo a través de los componentes especificados en server.xml.
2. web.xml - Configura los distintos contextos en Tomcat. Tomcat permite definir los valores por defecto de web.xml para todos los contextos poniendo un fichero web.xml por defecto en el directorio conf. Cuando se construye un nuevo contexto, Tomcat usa el fichero web.xml por defecto como configuración base, y el

fichero web.xml específico de la aplicación (WEB-INF/web.xml), sólo sobrescribe estos valores por defecto.

## 5.6. Cocoon - Estructura de Publicación basada en XML/XSL <sup>[WWW022]</sup>

Cocoon es un sistema de publicación web, basado en XML/XSL. Cuenta con desarrollo total en Java por lo cual se puede ejecutar desde cualquier servidor que pueda contener Servlets; y al ser un Servlet cuenta con las ventajas de éstos.

Cocoon es *Open Source*. Es bastante configurable y personalizable. Además adopta características para escribir páginas de servidor en XML (XSPs). Permite diferenciar el procesamiento del documento para tenerlo en distintos formatos, dependiendo del tipo de software que hace la petición y cuenta con un sistema de caché para tener un mejor rendimiento. Cocoon es una buena alternativa si se desea separar contenido, presentación y lógica en una aplicación.

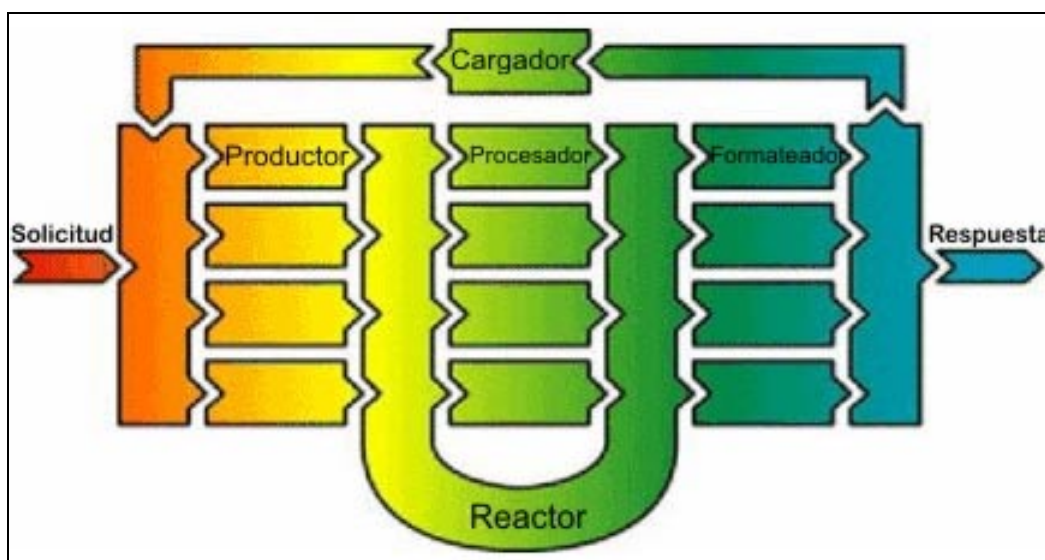
Un servidor web atiende peticiones de ficheros, mientras que un entorno de publicación web sirve *versiones publicadas de esos ficheros*. Una versión publicada de un fichero puede consistir en un fichero transformado con XSLT, o un fichero cuyo contenido se pasa a una aplicación de la que se obtiene un resultado, o en la conversión a otro formato, como PDF o WML. El usuario que realiza la petición, no tiene porque saber nada sobre lo que hay por debajo. Sólo tiene que teclear una determinada URL y recibir el resultado.

Cocoon puede servir archivos de los siguientes formatos:

- *XML*: Se pueden servir páginas XML para navegadores que entiendan XML, usando el parser (analizador XML) Xerces.
- *XSLT*: las transformaciones a documentos XML las puede aplicar el servidor, resultando un documento XML que se envía al cliente, usa el procesador Xalan.
- *FO*: mediante el módulo FOP, se pueden aplicar transformaciones a DTD finales, tales como PDF, pudiéndose servir documentos que no sean XML.
- *XSP*, son páginas de servidor en XML, tecnología equivalente a las ASP o JSP; son páginas "activas" que incluyen código Java.

### 5.6.1. Estructura de Cocoon

Desde el punto de vista estructural Cocoon se compone de los siguientes elementos:



**Figura 5.4.** Estructura de Cocoon.

- **Productores:** Son los ficheros fuentes de donde proviene el XML. Estos pueden ser estáticos o dinámicos (XSP). La operación de un productor se basa en transformar los datos del fichero en eventos SAX.
- **Procesadores:** Son los encargados de tratar el XML de los productores y someterlo a diversos procesos consecutivos. Por ejemplo las XSP son transformadas a clases Java, hacer conectividad a una base de datos o aplicar transformaciones XSL a documentos XML.
- **El Reactor:** Es la pieza central, extrae del XML generado por el productor las instrucciones de proceso que determinan que procesadores actuarán.
- **Formateadores:** Son los encargados de recoger la representación interna del XML resultante de los procesadores y prepararlo para enviarlo como respuesta al cliente en el formato adecuado (texto, HTML, XML, WML...).

### 5.6.2. Funcionamiento de Cocoon

Cocoon funciona como un *servlet*, o sea, en principio se podrá ejecutar desde cualquier servidor que pueda contener servlets. El procesamiento que hace Cocoon es en tres fases:

- La creación de contenidos XML la hace el usuario.
- El procesamiento del fichero XML lo lleva a cabo Cocoon, incluyendo la lógica incluida (si son XSP). La lógica está totalmente separada del contenido.
- Finalmente, se crea el documento aplicando las hojas de estilo, y se formatea al tipo requerido: HTML, WML, PDF.

A nivel funcional, el trabajo de Cocoon desde una petición del usuario a la devolución del documento final, pasa por las siguientes fases:

1. El usuario solicita un documento de cualquier tipo al servidor.
2. La solicitud se analiza para concluir si se puede atender o no. Si no se puede atender se produce un mensaje de error.
3. Si se puede atender se analiza a qué productor XML corresponde. Se genera un documento XML con el cual se trabajará.
4. Se extraen las instrucciones del XML generado en el paso anterior y éstas se le pasan al procesador apropiado para que se le apliquen al XML. Al procesar el XML podría salir un XML con más instrucciones que serán tratadas en algún otro ciclo.
5. El XML procesado se le pasa al elemento que aplica el formato. Si el documento es un documento final, XML aplica el formato y le envía el documento formateado al cliente. En el caso que el documento XML procesado, sea código que deba ejecutarse (como XSP compilada), éste se pasa como productor de XML y se vuelve a procesar hasta que se llega a un documento XML final.

### 5.6.3. Cocoon y XSP (XML Server Pages)

Las XSPs manejan la misma idea de las JSPs, es decir, páginas de servidor, con lo cual se tiene dinamismo con posibilidad de conectividad a bases de datos y con las ventajas del XML. Una XSP es simplemente un documento XML en donde se puede incluir contenido tanto estático como dinámico para generar XML de forma dinámica.

Una XSP debe estar bien formada y además debe tener ciertas características propias:

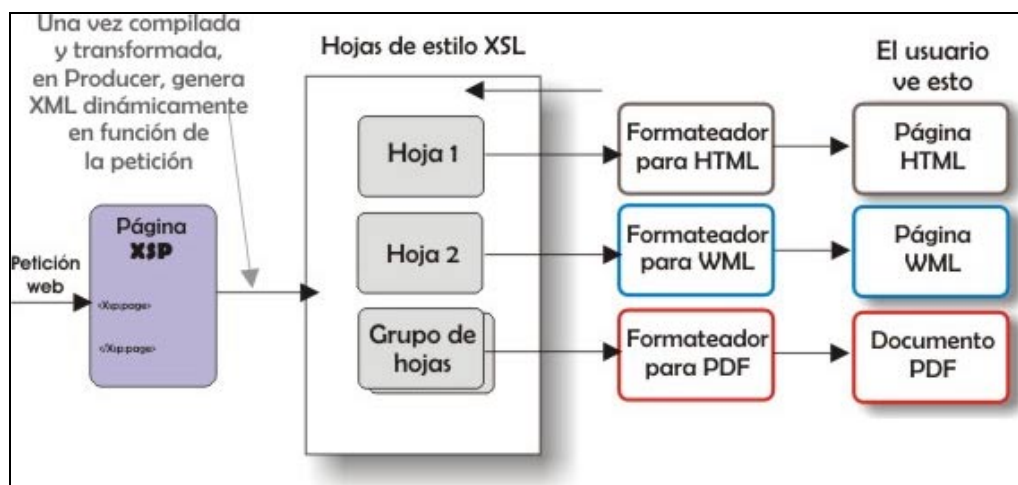
- Como es un documento XML, su primera línea es la de un documento XML:

```
<?xml version="1.0"?>
```

- El elemento raíz del documento debe ser `<xsp:page>`. Tiene un atributo en el que se especifica el lenguaje de programación usado para la lógica embebida, Cocoon

sólo soporta Java. Además en el elemento raíz se declaran los espacios de nombres usados y como mínimo hay que declarar el espacio de nombres xsp.

Dentro del modelo de Cocoon las XSP serían productores, la siguiente gráfica muestra el flujo de operación en una solicitud XSP.



**Figura 5.5.** Flujo en XSP

En la figura anterior vemos cómo a partir de un contenido en XSP se obtienen diversas vistas para distintos formatos como HTML, WML y PDF.

#### 5.6.4. Tipos de Páginas XSP

De acuerdo a la forma como se programan, las XSPs se pueden dividir en tres grupos:

1. Con la lógica embebida en la presentación
2. Con hojas de estilos
3. Con bibliotecas de etiquetas

##### 5.6.4.1. XSP con la lógica embebida en la presentación

En este tipo de páginas se escribe el código en la propia página XML. Para hacer el tratamiento de este tipo de XSP, el procesador XSP analiza el XML y convierte la página en un Servlet compilado. Esto lo hace llevando el XML a Java mediante un árbol DOM. Una vez llevado a código Java, se procesa y al resultado se le aplica una transformación XSLT para llevar el resultado final a una página HTML.

#### **5.6.4.2. XSP con hojas de estilos**

En esta forma de programar la página XSP original sería vista como un documento XML que se vale de hojas de estilos para aplicar la lógica de la programación. Cuando el documento XML original es procesado, se le aplica la transformación y como resultado se tiene una página XSP con el código embebido.

#### **5.6.4.3. XSP con bibliotecas de etiquetas**

La idea de esta forma de implementar XSP es tener en bibliotecas especializadas, etiquetas que se encarguen de ejecutar cierto proceso, cierta función o procedimiento escrito en un lenguaje de programación, para que dichas bibliotecas puedan ser incluidas mediante espacios de nombres en los ficheros XML que las necesitan y así mismo se puedan utilizar las funciones que proveen dichas bibliotecas.

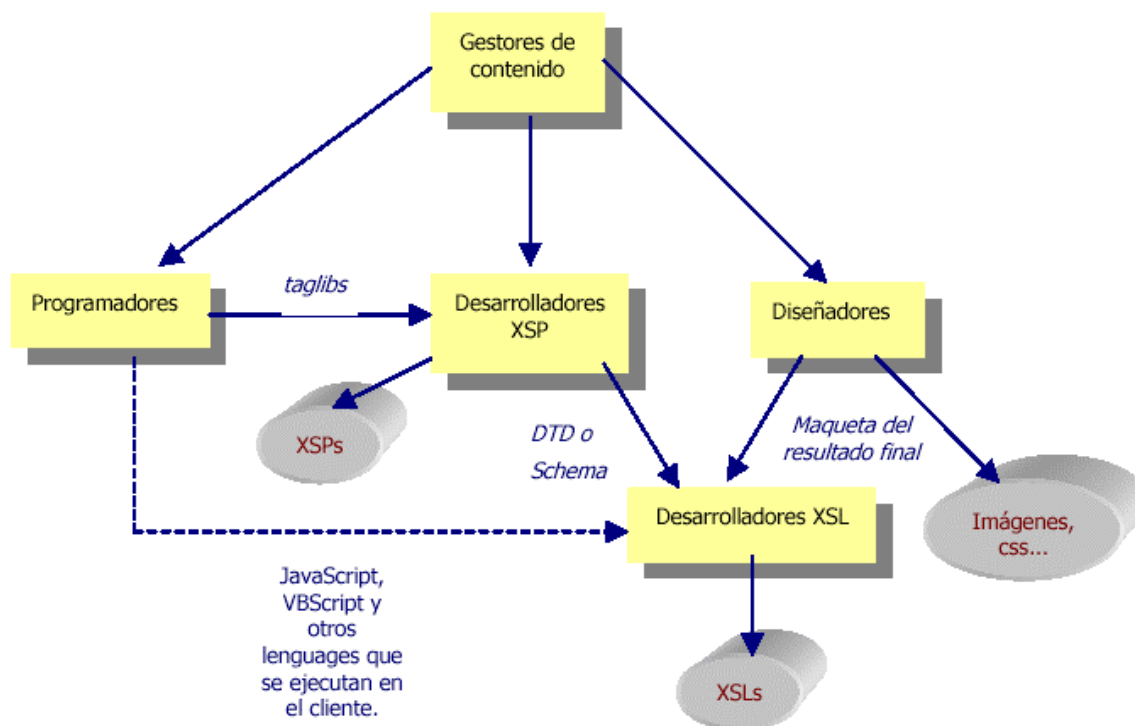
#### **5.6.5. Conectividad a bases de datos**

Con las XSP y Cocoon se puede tener acceso a una base de datos de cualquier tipo, con lo cual se puede tener la persistencia de la aplicación en un sistema manejador de bases de datos y aprovechar las ventajas tanto del manejador como de las XSP.

Para poder utilizar una XSP para conectarse a una base de datos, se tiene que cargar el *driver* de la base de datos, definir las variables típicas de conexión a base de datos, tales como url, nombre de usuario, contraseña, etc.

#### **5.6.6. Paralelismo en Cocoon**

El paralelismo con Cocoon se incrementa de forma enorme. Esto mejora tanto la calidad como los tiempos de desarrollo de trabajo y de respuesta de los mismos. Aquí se explica como se puede hacer *Workflow* con Cocoon.



**Figura 5.6.** Workflow en Cocoon

Cocoon apuesta por un desarrollo web que involucre diferentes tipos de personal o grupos de trabajo: creadores de contenidos, diseñadores gráficos y programadores, cada uno hace lo que mejor sabe y no influye en el trabajo de los demás. En el workflow ideal para el desarrollo de XSPs intervendrían 5 perfiles de desarrolladores:

- **Gestores de contenido:** Estudian el tipo de contenido al cual se quieren o deben referir. Una vez que identifican los contenidos, notifican a los programadores para la construcción de taglibs (librerías de etiquetas). La especificación de contenidos la reciben también los desarrolladores XSP y los diseñadores.
- **Programadores:** Proporcionan taglibs que al ejecutarse generan el contenido acordado con los gestores.
- **Desarrolladores de XML (XSPs):** Estructuran los contenidos en los XMLs que crean según lo acordado con los gestores. Introducen contenido estático en las páginas y las etiquetas taglibs donde corresponda para el contenido dinámico.
- **Diseñadores:** Se encargan de construir el esquema de la interfaz. Generan entonces la forma de presentación de los datos de cada vista, de cada formato y cada uno de los elementos estáticos.

- **Desarrolladores de XSL:** Se dedican a elaborar documentos XSL para obtener las vistas hechas por los diseñadores a partir de los XMLs hechos por los desarrolladores XML.

## 5.7. Software Freeware XML en Linux

Las herramientas que ofrece el Software Libre en Linux para implementar aplicativos web, están construidas para interactuar con otras herramientas a través de interfaces basadas en texto, sin embargo esta interacción no siempre es clara para el programador, y cuando se busca documentación al respecto, casi siempre es escasa, motivo por el cual el programador tiene que recurrir a los foros de discusión, deteniendo así el ciclo de desarrollo.

El desarrollo de aplicaciones web esta tomando una nueva forma, la cual implica retos no enfrentados previamente por los programadores, administradores de proyectos, analistas de sistemas, etc. Las aplicaciones web cambian profundamente los conceptos preestablecidos, cuestiones de seguridad, multiprocesamiento, presentación de datos, etc.

Existen en el mercado herramientas para la plataforma Linux que permiten editar, procesar y dar estilo a documentos XML/XSL que están empezando a distribuirse, pero algunas de ellas no son estables o no están listas para funcionar como se requiere; por este motivo se ha decidido proceder a la programación XML, XSL y XSP a través de Kwrite (semejante a Bloc de Notas en Windows), digitando todo el código para generar los resultados deseados.

En los siguientes cuadros vamos a listar los diferentes tipos de Software Freeware para Linux que existe en Internet para el manejo de documentos XML, XSL, entre otros [www023].



<b>Producto</b>	<b>Plataforma</b>
Amaya	Win32, Linux, Sun/Solaris, Unix
XED	Win32, Linux, FreeBSD and Solaris
Visual XML:	Java 1.1
Merlott	Java 1.2
Xml operator	Java 1.2
PECEL	Java 1.2.1
Protégé	Java
S-Link-S Editor	Java 1.1

**Tabla 5.3.** Editores XML

<b>Producto</b>	<b>Plataforma</b>
FOA	Java
XPath Tester	Java 1.2
XSL Editor	Java
Xslide	Emacs
XSLT-process	Emacs

**Tabla 5.4.** Editores XSLT

<b>Producto</b>	<b>Plataforma</b>
DTD explorer	Java 1.1
DTD2 relax	Java
XML man	Linux
EZDTD	sgml

**Tabla 5.5.** Editores DTD

Además de analizar XML, pueden también aplicar las transformaciones XSLT de una página de estilo.

<b>Producto</b>	<b>Plataforma</b>
Sablotron (sabcmd)	Linux
JAXP	Java
Libxslt (xsltproc; proyecto GNOME)	Linux
Xalan	C++, GNU

**Tabla 5.6.** Analizadores de XML/XSLT

### **Software para aplicativos web en Linux**

#### **Xara WebStyle 3 versión completa 35.7 MB**

Esta aplicación trae diseños, botones, banners animados, headers, barras de separación, logos y muchas cosas más ya prediseñadas, sólo se agrega texto y en segundos se tendrá una web atractiva, el programa es de gran calidad.

#### **Adobe Golive 6 104 MB**

Un editor web es la competencia directa de Dreamweaver, Golive6 puede crear poderosas aplicaciones en PHP, ASP, JSP... y todo sin saber nada de programación. Tiene el mejor editor de DHTML y Javascript y un sin fin de cosas más.

#### **Macromedia Fireworks MX 25 MB**

Fireworks desde la versión 3 nos ofrece un editor gráfico dedicado a hacer gráficos para web, es decir, crear buenas imágenes pero sin descuidar el peso de los archivos, cosa que es importantísima para que una web cargue rápido. En Fireworks MX no solo se puede hacer menús desplegables, se pueden cargar los menús desde XML, esto revoluciona el significado de gráfico dinámico en Internet.