

**INSTALACIÓN, CONFIGURACIÓN, IMPLANTACIÓN Y
MANTENIMIENTO DEL PORTAL WEB DE LA ESCUELA DE
INGENIERÍA MECÁNICA EIMWEB DE LA UNIVERSIDAD
INDUSTRIAL DE SANTANDER**

JONATHAN JULIÁN NAVAS AGUILAR

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2009**

**INSTALACIÓN, CONFIGURACIÓN, IMPLANTACIÓN Y
MANTENIMIENTO DEL PORTAL WEB DE LA ESCUELA DE
INGENIERÍA MECÁNICA EIMWEB DE LA UNIVERSIDAD
INDUSTRIAL DE SANTANDER**

JONATHAN JULIÁN NAVAS AGUILAR

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Msc. LUIS IGNACIO GONZÁLEZ RAMÍREZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2009

DEDICATORIA

A DIOS por darme vida, familia, capacidad y la oportunidad de conseguir este logro tan importante.

A Fanny, quien es mi madre, mi ejemplo y mi líder. Sin su extraordinario apoyo, sacrificio y amor no lo habría conseguido.

A mi padre Sivel, quien ha estado siempre pendiente, atento, orgulloso y quién en los momentos difíciles me ha apoyado.

A Dianita y Anso, quienes me han servido de motivación y ánimo para superar los momentos difíciles.

A mi abuelo Saúl, quien desde el cielo me protege y ayuda.

A Lyda, quien recorrió conmigo este camino brindandome su incondicional y valioso apoyo.

AGRADECIMIENTOS

A mis tias Eli, Marina, Elsa, Ferney, Claudia, a mi tío Luis Gabriel, y a toda mi familia quienes con sus buenos deseos, consejos y ayuda permitieron la consecución de este objetivo.

A la Escuela de Ingeniería de Sistemas y todo el cuerpo docente quienes, compartieron todo su conocimiento y experiencias ayudando así a mi formación como profesional

A todos los integrantes grupo de desarrollo de software CALUMET y en especial al profesor Luis Ignacio González Ramírez quien, como director de proyecto me motivó a no dejar de “pedalear” para llegar satisfactoriamente a la meta propuesta.

A mis compañeros y colegas con quienes viví, crecí, aprendí y luche por este logro.

A las escuelas de Ingeniería Eléctrica y Electrónica, Civil y Mecánica, las cuales me abrieron sus puertas y me permitieron aprender y obtener una experiencia valiosa en el campo laboral.

A todas las personas, quienes de una u otra manera me acompañaron y contribuyeron para conseguir este gran triunfo.

Índice general

1. INTRODUCCIÓN	1
1.1. PRESENTACIÓN DEL PROYECTO	2
1.1.1. Descripción del contenido del informe	2
1.1.2. Antecedentes	3
1.1.3. Definición del problema	4
1.1.4. Objetivos	4
1.1.4.1. Objetivos generales.	4
1.1.4.2. Objetivos específicos.	4
1.1.5. Justificación	5
1.1.6. Alcances y limitaciones	6
1.1.6.1. Alcances	6
1.1.6.2. Limitaciones	7
2. MARCO TEÓRICO	8
2.1. ARQUITECTURA CLIENTE-SERVIDOR	8
2.1.1. Características de la arquitectura Cliente-Servidor	9
2.1.2. Clasificación de las arquitecturas cliente-servidor	9
2.1.2.1. Arquitectura Cliente-Servidor de Dos Capas.	10
2.1.2.2. Arquitectura Cliente-Servidor de Tres Capas.	10
2.1.3. Arquitectura cliente-servidor usada	11

<i>ÍNDICE GENERAL</i>	II
2.1.4. Ventajas del esquema cliente-servidor	12
2.1.5. Desventajas del esquema Cliente-Servidor	12
2.2. TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS	13
2.2.1. Código del Lado del Cliente (Client Side Scripts).	13
2.2.2. Código del Lado del Servidor (Server Side Scripts).	13
2.2.3. Tecnología Utilizada.	15
2.2.3.1. Modelo de Acceso a JSP.	16
2.3. BASES DE DATOS	17
2.3.1. Componentes principales de una base de datos	17
2.3.2. Tipos de modelos de datos	18
2.3.2.1. El modelo jerárquico.	18
2.3.2.2. El modelo de red.	18
2.3.2.3. El modelo relacional.	18
2.3.3. Acceso a Base de Datos.	19
2.3.3.1. Conectores más Utilizados.	19
2.3.3.2. Conectividad utilizada.	20
2.3.4. Manejadores o Gestores de Bases de Datos.	20
2.3.4.1. MySQL.	22
2.3.4.2. Ventajas de MySQL	22
2.4. PROGRAMACIÓN DE APLICACIONES	23
2.4.1. Programación Estructurada (PE)	23
2.4.1.1. Programación Orientada a Eventos	23
2.4.2. Programación Genérica (PG)	23
2.4.3. Programación Orientada a Objetos (POO)	23
2.4.3.1. Clases y objetos	24
2.4.3.2. Atributos	24
2.4.3.3. Métodos	24

<i>ÍNDICE GENERAL</i>	III
2.4.3.4. Herencia	25
2.4.3.5. Ventajas de la POO	25
2.4.3.6. JAVA.	26
2.5. SERVIDOR.	26
2.5.1. Tipos de Servidores.	27
2.5.1.1. Servidores Web.	28
2.5.1.2. Servidor Web Apache	31
3. MARCO METODOLOGICO.	33
3.1. PROTOTIPADO EVOLUTIVO.	33
3.1.1. Procedimiento a seguir para la metodología planteada.	35
3.2. EFECTIVIDAD DEL PROTOTIPADO EVOLUTIVO.	36
3.3. PROBLEMAS DEL PROTOTIPADO EVOLUTIVO.	36
4. INSTALACIÓN Y CONFIGURACIÓN DEL <i>SERVIDOR</i>.	37
4.1. HARDWARE.	37
4.2. INSTALACIÓN DEL SISTEMA OPERATIVO.	38
4.2.1. Pantalla de Bienvenida.	39
4.2.2. Configuración de Discos Duros.	39
4.2.2.1. Sistema de Archivos.	39
4.2.2.2. Estructura de Directorios.	40
4.2.2.3. Sistema de Almacenamiento.	42
4.2.3. Paquetes de Instalación adicionales.	46
4.2.4. Instalación.	46
4.3. CONFIGURACIÓN INICIAL.	48
4.3.1. Cuenta de Administración.	48

<i>ÍNDICE GENERAL</i>	IV
4.3.2. Nombres de Red.	49
4.3.2.1. Nombre Unico de Máquina o Hostname.	50
4.3.3. Configuración de Red.	50
4.3.4. Servicios Básicos.	51
4.3.5. Zona Horaria.	52
4.3.6. Finalización.	53
5. CONFIGURACIÓN AVANZADA E IMPLANTACIÓN	55
5.1. APLICACIONES DEL SISTEMA OPERATIVO TIPO SERVIDOR.	55
5.1.1. Un Ejemplo: Mac OS X vs Mac OS X Server.	55
5.1.2. Algunos Servicios Pre-Instalados en Mac OS X Server Tiger.	56
5.1.3. Herramientas de Administración de Servidor.	58
5.2. CONFIGURACIÓN DEL SERVIDOR WEB.	60
5.2.1. Java Development Kit (JDK).	60
5.2.2. Configuración Apache - Tomcat.	61
5.2.2.1. Tomcat.	61
5.2.2.2. Apache.	65
5.2.2.3. Configuración del conector “mod_jk”.	70
5.3. CONFIGURACIÓN BÁSICA DE MYSQL	74
5.3.1. Mysql Manager.	74
5.3.2. Creación de Usuarios	75
5.4. CONFIGURACIÓN DEL FIREWALL	75
5.4.1. Administración del Firewall	76
5.5. PROCESO DE IMPLANTACIÓN DEL PORTAL EIMWEB	78
5.5.1. Compatibilidad con el Servidor	78
5.5.2. Alojamiento del Portal en el Nuevo Servidor	79
5.5.3. Cargando las Bases de Datos en Mysql	80
5.5.4. Librerías Necesarias	82
5.5.5. Finalización.	82

<i>ÍNDICE GENERAL</i>	v
6. PRUEBAS DEL SISTEMA	85
6.1. CONEXIÓN Y ACTUALIZACIÓN DE LAS BASES DE DATOS. . . .	85
6.2. NAVEGACIÓN EN EIMWEB	85
6.2.1. Usuario Invitado	86
6.2.2. Usuario Registrado	87
6.2.3. Usuario Administrador	88
7. CONCLUSIONES Y RECOMENDACIONES	89
7.1. CONCLUSIONES	89
7.2. RECOMENDACIONES	90

Índice de figuras

2.1. Modelo Cliente - Servidor	9
2.2. Esquema de Arquitectura Cliente-Servidor de Dos Capas	10
2.3. Esquema de Arquitectura Cliente-Servidor de Tres Capas	11
2.4. Modelo de acceso a JavaServer Pages	16
2.5. Función del DBMS.	21
2.6. Funcionamiento de una servidor web.	29
3.1. Prototipado Evolutivo.	34
3.2. Diagrama de flujo del Prototipo Evolutivo	35
4.1. Máquina servidora para el portal web de la Escuela de Ingeniería Mecánica.	38
4.2. Pantalla de bienvenida.	39
4.3. Utilidad de Discos.	43
4.4. Sistema RAID 1.	44
4.5. Utilidad de Discos despues de creación de RAID nivel 1.	45
4.6. Paquetes Adicionales.	46
4.7. Destino de Instalación.	47
4.8. Ventana de instalación.	47
4.9. Finalización de la instalación.	48
4.10. Creación de Usuario Administrador.	49
4.11. Asistente de Configuración de Red	51

4.12. Servicios Básicos	52
4.13. Zona Horaria	53
4.14. Pantalla de Bienvenida Despues de Instalación.	54
5.1. Server Admin.	59
5.2. Server Monitor	59
5.3. Mysql Manager	60
5.4. Tomcat en Server Admin.	62
5.5. Estructura de Tomcat.	63
5.6. Arranque y Parada de Tomcat Gráficamente	64
5.7. Archivos de Configuración de Apache	66
5.8. Servicio Web en Server Admin	67
5.9. Ventana Sites	68
5.10. Ficha General.	69
5.11. Pestaña Options	70
5.12. Activación Modulo mod_jk	73
5.13. Mysql Manager	74
5.14. Firewall en Server Admin	77
5.15. Firewall Services.	78
5.16. Primera Imagen EIMWeb	83
5.17. Portal EIMWeb Implantado	84

Índice de tablas

2.1. Características Principales de Cada Tecnología.	14
5.1. Librerías EIMWeb	82
6.1. Conexión y Actualización de Base de Datos.	85
6.2. Usuario Invitado	86
6.3. Usuario Registrado	87
6.4. Usuario Administrador	88

RESUMEN

TÍTULO:

INSTALACIÓN, CONFIGURACIÓN, IMPLANTACIÓN Y MANTENIMIENTO DEL PORTAL WEB DE LA ESCUELA DE INGENIERÍA MECÁNICA EIMWEB DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER¹

AUTOR:

JONATHAN JULIÁN NAVAS AGUILAR ²

PALABRAS CLAVE:

EISiWeb, EIMWeb, Tomcat, Apache, Mysql, Portal web, Servidor.

DESCRIPCIÓN:

El presente documento de grado contiene la forma como se desarrollaron los procesos de instalación, configuración e implantación del portal web de la Escuela de Ingeniería Mecánica EIMWeb, respondiendo a las necesidades de una herramineta que permita la interacción entre los miembros de la Escuela (Estudiantes, Egresados, Profesores y Administrativos).

El portal web EIMWeb está basado principalmente en los servicios ofrecidos por el portal web de la Escuela de Ingeniería de Sistemas, el cual ha demostrado ser útil para mantener en constante comunicación e interacción a los miembros de la comunidad educativa.

Se describen paso a paso los procesos de instalación y configuración básica del sistema operativo Mac OS X Tiger 10.4.11. Así mismo, la configuración del servidor web *Apache 1.3* trabajando en conjunto con *Tomcat 4.11* a través del módulo *mod_jk 1.2* y la administración de la base de datos utilizando el motor *Mysql 4*.

Por último se presenta un informe de pruebas del sistema basado en el “registro de usuario” en el portal EIMWeb de los estudiantes de primer nivel de Ingeniería Mecánica, lo cual permitió evaluar la eficiencia y confiabilidad de los servicios más importantes del portal en un ambiente real y masivo.

¹Proyecto de Grado

²Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías de Sistemas. Msc. Luis Ignacio González Ramírez.

ABSTRACT

TITLE:

INSTALLATION, CONFIGURATION, IMPLEMENTATION AND MAINTENANCE OF MECHANICAL ENGINEERING SCHOOL WEB SITE EIMWEB³

AUTHOR:

JONATHAN JULIAN NAVAS AGUILAR⁴

KEY WORDS:

EISiWeb, EIMWeb, Tomcat, Apache, Mysql, Web Site, Server.

DESCRIPTION:

This degree document contains the form of how the processes of installation, configuration and implementation of Mechanical Engineering School web site EIMWeb is developed, responding to the needs of a tool that allows interaction of all school members (Students, Graduates, Professors and Administrative Personal).

The web site EIMWeb is mainly based on services offered by the web site of the Engineering School Systems, which has proved to be useful to maintain constant communication and interaction of members of the educational community.

The processes of basic configuration and installation of Mac OS X Tiger 10.4.11 are described step by step. Likewise, the web server configuration Apache 1.3 work together with Tomcat 4.11 through the module mod_jk 1.2 and the administration of database engine using MySQL 4.

Finally a system test report based on "user record" is presented for first level Mechanical Engineering students on the website EIMWeb, which allows to evaluate the efficiency and reliability for the most important services in a real and massive context.

³Degree Project

⁴Faculty of Physical-Mechanic Engineering. School of Systems Engineering. Msc. Luis Ignacio González Ramírez.

Capítulo 1

INTRODUCCIÓN

El Portal EISIWeb de la Escuela de ingeniería de Sistemas e Informática (EISI), con el transcurrir del tiempo se ha convertido en el principal canal de comunicación e integración entre sus usuarios; y es debido a la gran aceptación y evidente utilidad que este ha mostrado, que día a día se considera de vital importancia el fortalecimiento y mejora de cada uno de los servicios que ofrece y módulos que lo conforman.

Calumet ha sido el grupo de desarrollo software encargado de desarrollar, administrar y mantener el sitio EISIWeb, el cual desde sus inicios hasta hoy ha ido evolucionando, hasta hacer de este medio un instrumento cada vez más útil, agradable y de fácil uso para sus usuarios. Con el objeto de llevar a cabo esta labor se ha contado con herramientas software de libre distribución como lo son JSP, Java, Javascript y MySQL, permitiendo así que el portal EISIWeb proporcione páginas con contenido dinámico y fácil de usar.

Esta creciente popularidad, ha dejado nuestras fronteras y se ha abierto paso a otras escuelas que ven con buenos ojos el trabajo que se ha venido realizando, y que concluyen que el portal web EISIWeb propone soluciones a problemas que se vienen presentando a partir del crecimiento institucional. Gracias al trabajo y a los buenos resultados hoy, estamos trabajando con la Escuela de Ingeniería Civil y con la Escuela de Estudios Empresariales e Industriales, quienes están satisfechos y utilizan masivamente nuestros servicios.

En este documento se presenta la instalación, configuración e implantación de EISIWeb para la Escuela de Ingeniería Mecánica llamado “EIMWeb”. Se muestra paso a paso los procesos que culminan con un producto que pretende cumplir las expectativas, como lo ha hecho en otras escuelas, y seguir siendo, no solo, importante para la escuela de

Ingeniería Mecánica, sino también para toda la comunidad universitaria.

1.1. PRESENTACIÓN DEL PROYECTO

1.1.1. Descripción del contenido del informe

Este documento contiene un informe detallado de cada una de las etapas llevadas a cabo durante el desarrollo del proyecto: INSTALACIÓN, CONFIGURACIÓN, IMPLANTACIÓN Y MANTENIMIENTO DEL PORTAL WEB DE LA ESCUELA DE INGENIERÍA MECÁNICA EIMWEB DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.

Su distribución es la siguiente:

CAPÍTULO 1. INTRODUCCIÓN: Se presenta el proyecto y se hace un análisis de los antecedentes, definición del problema, objetivos generales, específicos, justificación, alcances y limitaciones del proyecto.

CAPÍTULO 2. MARCO TEÓRICO: Se muestran los conceptos aplicados en el desarrollo técnico del proyecto: lenguajes de programación, servidores y arquitectura.

CAPÍTULO 3. MARCO METODOLÓGICO: Explica la metodología utilizada como guía de este proyecto y la explicación del por qué fue escogida.

CAPÍTULO 4. INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR : Presenta paso a paso la Instalación y configuración básica de la máquina servidora de Ingeniería Mecánica.

CAPÍTULO 5. CONFIGURACIÓN AVANZADA E IMPLANTACIÓN: Presenta el proceso de configuración del servidor web y detalla los pasos de la implantación del portal web EIMWeb.

CAPÍTULO 6. PRUEBAS DEL SISTEMA: En este capítulo se presentan los resultados de las pruebas realizadas al portal implantado en la escuela de Ingeniería Mecánica..

CAPÍTULO 7. CONCLUSIONES Y RECOMENDACIONES: Se dan las conclusiones del proyecto y recomendaciones por parte del autor.

1.1.2. Antecedentes

“CONSTRUIR COMUNIDAD” es el lema bajo el cual se desarrolló EISIWeb. Para esto se decidió que era necesario tener publicada toda la información institucional como la misión, visión, estructura organizacional, ubicación geográfica, directivos, profesores, grupos, etc. Pero también se notó la necesidad de publicar aquella información pasajera concerniente a prácticas empresariales, oportunidades de empleo, procesos académicos y administrativos, noticias de la disciplina, miembros que se destacan, calendarios, y demás.

Tener toda la información de la escuela no era suficiente para CONSTRUIR COMUNIDAD; no basta con ingresar a un Sitio Web y encontrar toda la información y poder acceder a ella de manera pasiva. Para “construir comunidad” es necesario generar los espacios de comunicación, integración, debate, participación, publicación para todos los miembros de la comunidad.

El grupo de desarrollo de software CALUMET dio inicio a este proyecto con el fin de beneficiar a los estudiantes, egresados, administrativos, profesores y directivos de la EISI; tal beneficio se refleja en el uso de los servicios desarrollados y ofrecidos por el portal EISIWeb. Estos servicios han nacido de los aportes de diferentes generaciones de estudiantes y con la ayuda de las últimas tecnologías de desarrollo. Algunas de ellas son Java, JSP, MySQL.

El Portal EISIWeb ha mejorado la comunicación entre los miembros de la Escuela, y día a día incrementa el número de sus usuarios, quienes encuentran la información que necesitan e interactúan con el resto de la comunidad. La buena acogida que ha tenido, y los buenos resultados de aceptación por parte de los usuarios, hacen que un portal como este, sea una necesidad no solamente para la escuela de Ingeniería de Sistemas, sino para otras escuelas que presenten características muy similares a la nuestra.

Gracias al continuo trabajo, el desarrollo de servicios de interés común, y al fortalecimiento de nuestro portal EISIWeb, exitosamente se ha llegado a otras escuelas, las cuales han visto que nuestra filosofía “CONTRUIR COMUNIDAD” ha tenido resultado; se empezó con la Escuela de Ingeniería Civil (ECIWeb) y se continuó con la Escuela de Estudios Industriales y Empresariales (EEIEWeb). Ahora se suma la Escuela de Ingeniería Mecánica (EIMWeb) quien teniendo como base nuestro portal web (EISIWeb), mejorará el flujo de información hacia fuera de las escuelas, entre los miembros de esta, permitirá la integración y participación de todos en las decisiones importantes, y se

convertirá en una herramienta imprescindible para la comunidad.

1.1.3. Definición del problema

Debido al crecimiento institucional y al manejo de un gran volumen de personas e información, los procesos internos y externos de cada escuela se hacen difíciles, largos y complejos. Por otro lado, se necesita que los estudiantes, egresados, docentes y personal administrativo, tengan una comunicación fácil y clara entre ellos, que permita, el desarrollo de estrategias para la resolución de problemas tanto académicos como administrativos.

Esta situación no es ajena a la Escuela de Ingeniería Mecánica, razón por la cual, es necesario contar con un sistema de información que permita agilizar estos procesos y resolver de manera más eficiente algunas de estas problemáticas.

1.1.4. Objetivos

1.1.4.1. Objetivos generales.

- Instalar y configurar el servidor de la Escuela de Ingeniería Mecánica.
- Implantar el portal de la escuela de Ingeniería Mecánica EIMWEB.
- Capacitar a los nuevos integrantes del grupo de desarrollo CALUMET en el desarrollo de las tareas de configuración e implantación.

1.1.4.2. Objetivos específicos.

- Instalar y configurar el Sistema Operativo MAC OSX SERVER TIGER 10.4.11 en el equipo asignado como servidor “mecanica.uis.edu.co” para la escuela de Ingeniería Mecánica.
- Instalar y Configurar los paquetes necesarios para ejecutar el equipo como Servidor Web con el fin de implementar el Portal de la Escuela de Ingeniería Mecánica EIMWEB y accederlo vía Internet.

- Realizar las instalaciones y configuraciones necesarias para acceder las Bases de Datos de la División de Servicios de Información y de esta forma obtener la información de la Escuela de Ingeniería Mecánica necesaria para el correcto funcionamiento del sitio Web EIMWEB.
- Crear las bases de datos en MySQL Server para la Carga de las Bases de Datos SQL en MySQL Server.
- Crear la Base de Datos Diamante y adecuarla para el funcionamiento del sitio EIMWEB.
- Realizar la migración del Sitio Web de la Escuela de Ingeniería de Sistemas EISIWeb al sitio Web de la Escuela de Ingeniería Mecánica EIMWEB.
- Depurar el Sitio de la Escuela de Ingeniería de Sistemas EISIWeb preparándolo para la implantación en el sitio de la escuela de ingeniería mecánica EIMWEB.
- Identificar puntos críticos en los archivos contenidos en el sistema para ser adecuado a la Escuela de Ingeniería Mecánica.
- Editar cada uno de los archivos que contiene el sitio y de ser necesario, realizar los cambios respectivos en el código fuente para que el sitio quede completamente funcional para la Escuela de Ingeniería Mecánica.
- Realizar pruebas minuciosas de todos y cada uno de los módulos del sitio en busca de posibles fallos.
- Realizar capacitaciones a los nuevos integrantes del grupo CALUMET para asegurar que el trabajo realizado en el Grupo de desarrollo siga con las mismas políticas y metodologías realizadas hasta el momento.

1.1.5. Justificación

El Portal Web de la Escuela, EISIWeb 5.0, ofrece a sus usuarios información completa de programas, planes de estudio, agendas, cartelera, comunicados, horarios, docentes, grupos, entre otros servicios, que permiten a la comunidad estar enterada de todos los eventos y novedades que se programan o se presentan, ofreciendo de esta forma una herramienta cómoda de interacción, información e integración.

La necesidad de una herramienta de este tipo no es exclusiva de la escuela de Ingeniería de Sistemas, se puede afirmar con seguridad que todas las Escuelas de la UIS requieren con urgencia una herramienta que les permita de manera sencilla publicar la información que constantemente llega a las direcciones relacionadas con prácticas empresariales, oportunidades laborales, procesos académicos y administrativos.

Así mismo, para muchas unidades académicas es de vital importancia publicar su información institucional, como la visión, la misión, órganos directivos, calendarios de cursos, diplomados, especializaciones, semanas técnicas y otros. Es por lo anterior, que desde el grupo de desarrollo Web CALUMET se ha tomado la decisión de crear un sitio Web plantilla, parametrizado, escalable, y de fácil implantación en cualquier otra escuela que se interese en tener una herramienta de este tipo.

1.1.6. Alcances y limitaciones

1.1.6.1. Alcances

Con las tecnologías de desarrollo con las que se cuenta hoy, muchas de libre distribución, se pueden lograr proyectos como el desarrollado por el grupo de software CALUMET, el cual maneja tecnologías de desarrollo java que permiten investigar, desarrollar y compartir con la comunidad.

Hay que tener en cuenta que estas tecnologías permiten que el alcance de los proyectos cada vez sea mayor; esto se puede aplicar a el proyecto desarrollado en el grupo ya que se puede investigar sobre nuevas metodologías de desarrollo y aplicar para que el proyecto cada vez sea mas robusto, y a la vez permita resolver problemas de vida cotidiana respecto a los que corresponden al tema del proyecto.

Con la Parametrización de EISIWeb, y la implantación de este sistema en otras escuelas se puede ver que las nuevas tecnologías de desarrollo permiten que los proyectos desarrollados sean escalables y que tengan un estándar reconocido, de manera que al momento que ingresen nuevas personas al proyecto, asimilen fácilmente lo realizado y entiendan la metodología de desarrollo con la que se está trabajando. De esta manera, podrán continuar trabajando de forma similar y con las metodologías que se vienen aplicando.

1.1.6.2. Limitaciones

Las metodologías nombradas anteriormente, exigen que cada vez sean necesarios más recursos de hardware. Por lo anterior, es que cada vez que se investigue sobre una nueva tecnología se debe tener en cuenta los recursos de software que exige la tecnología de desarrollo. Esta es, tal vez, la limitación más importante para utilizarlas. Al tener suficientes recursos de hardware, lo que se puede lograr son proyectos mucho más robustos, desarrollados en un tiempo menor.

Capítulo 2

MARCO TEÓRICO

2.1. ARQUITECTURA CLIENTE-SERVIDOR

En el esquema Cliente-Servidor dos o más procesos actúan autónomamente, pero en una forma coordinada y cooperativa; de este modo una aplicación solicita datos a otra e inmediatamente se recibe la petición, se procede a elaborar la respuesta y se devuelve a la aplicación demandante. Los principales componentes de esta arquitectura son los Clientes, los Servidores y la infraestructura de comunicaciones.

Las aplicaciones del lado del cliente interactúan con el usuario, normalmente usando una interfaz gráfica. Con frecuencia se comunican con procesos auxiliares que establecen una conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Las aplicaciones del lado del servidor no tienen interfaz gráfica, sin embargo proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente.

Para que las aplicaciones del lado del cliente y del servidor se comuniquen, se hace necesaria una infraestructura de comunicaciones que proporciona los mecanismos básicos de direccionamiento y transporte (Interfaz de comunicaciones).

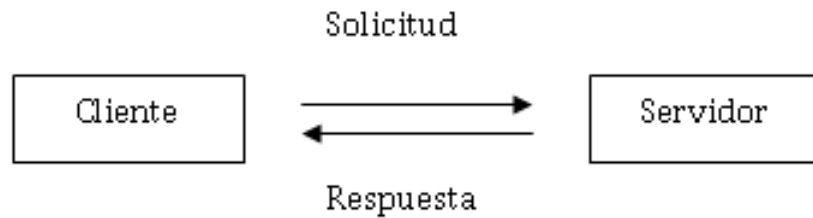


Figura 2.1: Modelo Cliente - Servidor

2.1.1. Características de la arquitectura Cliente-Servidor

- Las tareas de las aplicaciones de lado del cliente y del servidor tienen diferentes exigencias en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco.
- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Las aplicaciones del lado del cliente corresponden a procesos con carácter activo porque hacen peticiones de servicios a los servidores, que tienen un carácter pasivo ya que esperan las peticiones de las aplicaciones del lado del cliente.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente-Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

2.1.2. Clasificación de las arquitecturas cliente-servidor

Los sistemas cliente servidor se clasifican de acuerdo al nivel de abstracción del servicio que se ofrece. Se distinguen tres componentes básicos de software:

- **Presentación:** Muestra al usuario un conjunto de objetos visuales y realiza el procesamiento de datos producidos por el mismo y los que son devueltos por el servidor.
- **Lógica de aplicación:** Es responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos:** Esta compuesta por los archivos que contienen los datos de la aplicación.

2.1.2.1. Arquitectura Cliente-Servidor de Dos Capas.

- El sistema se separa en dos partes fijas: Las aplicaciones del lado del cliente y las aplicaciones del lado del servidor.
- La lógica de las aplicaciones debe estar en el cliente o en el servidor.
- La comunicación con el servidor es transparente para el usuario: El cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos.

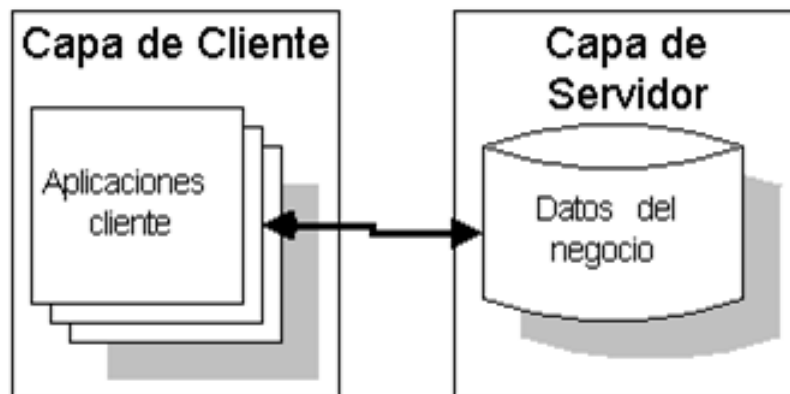


Figura 2.2: Esquema de Arquitectura Cliente-Servidor de Dos Capas

2.1.2.2. Arquitectura Cliente-Servidor de Tres Capas.

Esta compuesta de:

- Un equipo cliente con una interfaz de usuario (normalmente se utiliza un navegador Web), que solicita los recursos.
- El servidor de aplicaciones (también es llamado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos, que almacena y proporciona, al servidor de aplicaciones, los datos que requiere.

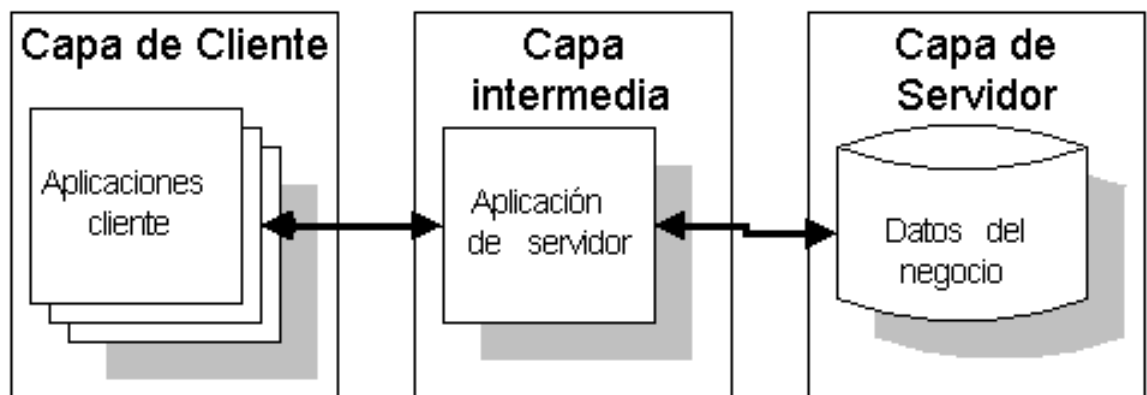


Figura 2.3: Esquema de Arquitectura Cliente-Servidor de Tres Capas

2.1.3. Arquitectura cliente-servidor usada

Para el desarrollo de este proyecto, se utiliza arquitectura de tres capas, debido a las ventajas que ofrece como escalabilidad, facilidad de mantenimiento y el manejo de un mayor número de usuarios que la arquitectura Cliente-Servidor de dos capas. La arquitectura es aplicada de la siguiente forma:

- Capa de Cliente: Interfaz con el usuario, en este caso se usa un navegador Web.
- Capa Intermedia: Para los servicios del negocio se utiliza una computadora configurada como servidor Web, en el cual se almacena el sitio Web conformado por páginas JSP y JavaBeans. Allí se realizan los procesos complejos, y se solicitan los servicios del servidor de datos cuando es necesario acceder a la información almacenada en la base de datos.

- Capa de Servidor: Se utiliza el motor de bases de datos MySQL, el cual se encuentra en el mismo servidor Web.

2.1.4. Ventajas del esquema cliente-servidor

- La arquitectura Cliente-Servidor facilita la integración entre sistemas heterogéneos y comparte información permitiendo, por ejemplo, que las máquinas ya existentes puedan ser usadas con interfaces más amigables al usuario.
- Al favorecer el uso de interfaces gráficas interactivas, los sistemas construidos bajo este esquema son más intuitivas para el usuario.
- Proporciona, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información principal globalmente.

2.1.5. Desventajas del esquema Cliente-Servidor

- El mantenimiento de los sistemas es algo complicado sin la debida documentación, pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con pocas herramientas para la administración y ajuste del desempeño de los sistemas, además, se debe tener estrategias para el manejo de errores y para mantener la consistencia de los datos.
- La seguridad de un esquema Cliente-Servidor es un factor importante a tener en cuenta. Por ejemplo, se deben hacer validaciones y verificaciones tanto en el cliente como en el servidor.
- Un inadecuado desempeño en una arquitectura de este tipo puede ocasionar congestión en la red, dificultad de tráfico de datos, etc.

2.2. TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS

2.2.1. Código del Lado del Cliente (Client Side Scripts).

Se refiere al código que se ejecutan en los navegadores que las computadoras clientes tienen instalados. Estos códigos, no hacen necesario que el servidor Web cumpla determinados requisitos. Las tecnologías más comunes de este tipo son:

- JavaScript: es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Permite la creación de ventanas, mostrar y cambiar texto e imágenes en movimiento, validar entradas de un determinado formulario antes de enviarlo al servidor.
- Java Applets: Desarrollado por Sun Microsystems. Los applets son programas escritos en lenguaje de programación Java, se incrustan en el código fuente de la página Web y se ejecutan en el navegador del cliente gracias a la Máquina Virtual de Java (Java Virtual Machine, JVM) que éste lleva incorporado. Pueden lograr efectos para el texto, sonido e imágenes.
- Controles Activos: Tecnología Microsoft. Los usuarios de Netscape requieren de determinados plug-ins para soportarlos. Es la propuesta de Microsoft frente a los Applets de Java.

2.2.2. Código del Lado del Servidor (Server Side Scripts).

Estos códigos se ejecutan en el servidor. Para su funcionamiento, el programa se ejecutará en el servidor con los datos o peticiones que el usuario envía desde su navegador y el servidor muestra los resultados del programa en una página HTML que el usuario verá normalmente en su navegador.

Los más usados son:

	Java Server Pages	Active Server Pages	PHP
SERVIDORES WEB QUE SOPORTAN LA METODOLOGÍA	Esta tecnología se puede implementar en los servidores Web más Populares como Apache, Netscape, Xitami, IIS.	Solo es soportado por Microsoft IIS y Microsoft Personal Web Server	Al igual que JSP, es instalable en Apache, IIS, Netscape, etc.
PLATAFORMA QUE SOPORTA LA TECNOLOGÍA	Independiente de alguna plataforma en específico	Dependiente de la plataforma Windows debido a la dependencia del componente win32.	Independiente de alguna plataforma en específico.
COMPONENTES	Se basa principalmente en la reusabilidad de componentes. Por ejemplo JavaBeans Enterprise, JavaBeans, Tag Libraries.	Basada Principalmente en la arquitectura COM de Win32.	Basado en el motor de interpretación creado por Zend.
SCRIPTING	Se utiliza el lenguaje de programación Java	Se utiliza VBScript o JScript	Utiliza una sintaxis similar C++ y Java.
SEGURIDAD	Trabaja con el modelo de seguridad de Java.	Funciona con el modelo de Seguridad de Windows NT.	Depende de la manera de instalar, si es por modo CGI o como módulo del servidor Web.
ACCESO A BASE DE DATOS	Acceso por medio de JDBC	Acceso por medio de los objetos ADO.	Funciones incorporadas para los diferentes DBMS que PHP soporta.

Cuadro 2.1: Características Principales de Cada Tecnología.

- ASP (Active Server Pages). Se utiliza mucho en la gestión de Bases de Datos ya que puede conectarse a SQL, Access, Oracle u otras. Requiere de una computadora configurada como Servidor Web de Microsoft (Microsoft Web Server), en este caso, el navegador del cliente es indiferente pues el trabajo se realiza del lado del Servidor.
- PHP. Es un lenguaje similar al usado en la tecnología ASP pero de código abierto (Open Source) y gratuito. Su gran potencia se encuentra en la interacción con los motores de bases de datos más usados: Oracle, Sybase, MySQL.
- JSP (Java Server Pages). Es una tecnología que permite la generación dinámica de páginas Web combinando código JAVA (scriptlets) con un lenguaje marcado como HTML o XML.

Para complementar esta información ver Cuadro 2.1

2.2.3. Tecnología Utilizada.

La tecnología usada para la creación del sitio Web es JSP, de la misma manera los nuevos servicios son desarrollados con esta misma tecnología ya que permite desarrollar aplicaciones independientes de la plataforma y portables a otros sistemas operativos y servidores Web.

Los paginas JSP y servlets se ejecutan en una máquina virtual de Java, lo cual permite que se puedan usar en cualquier tipo de computadora, siempre que exista una máquina virtual de Java para ella. Cada JSP se ejecuta en su propio contexto (llamado también hilo o hebra); pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa e interpretarlo). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

Un JSP se compila a una aplicación en Java la primera vez que se invoca, y de esta aplicación en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página Web.

2.2.3.1. Modelo de Acceso a JSP.

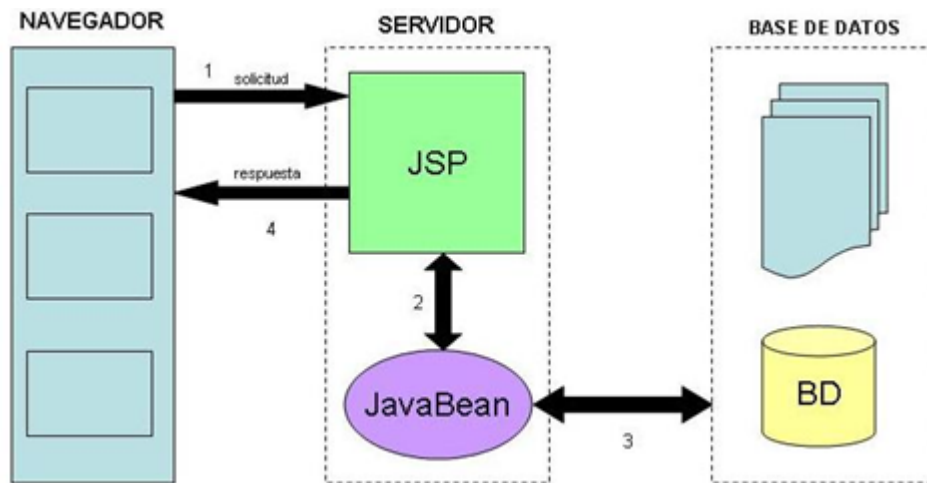


Figura 2.4: Modelo de acceso a JavaServer Pages

Teniendo en cuenta la Figura 2.1 podemos explicar el funcionamiento del modelo de acceso a JSP a través de los siguientes pasos:

1. Un usuario desde un navegador Web cliente hace una petición que es enviada a un archivo JSP. Este archivo accede a componentes del servidor que generan contenido dinámico y lo presentan en el navegador.
2. Después de recibir la petición del cliente, el archivo JSP pide información de un Javabeen si es necesario.
3. El Javabeen puede, en turnos, pedir información de otro Javabeen o de una base de datos.
4. Una vez el Javabeen genera el contenido, el archivo JSP puede consultar y presentar el contenido del Javabeen al navegador.

En este modelo, el archivo JSP está identificado por una extensión `.jsp`, lo cual indica al servidor que este archivo requiere de un manejo especial. La primera vez que este archivo es solicitado, bien sea desde un navegador Web o desde un servlet, el archivo `.jsp` es compilado en un objeto. La respuesta del objeto es HTML estándar, el cual es interpretado por el navegador para ser presentado al usuario.

Después de la compilación, el objeto de la página compilada es almacenado en la memoria del servidor. En las peticiones posteriores a esta página, el servidor revisa si el archivo .jsp ha cambiado. Si no ha cambiado, el servidor utiliza el objeto de la página compilada guardado en memoria para generar la respuesta al cliente. Si el archivo ha cambiado, el servidor automáticamente recompila el archivo de la página y reemplaza el objeto en la memoria.

2.3. BASES DE DATOS

Una base de datos es un conjunto de datos que pertenecen a un mismo contexto y que son almacenados porque se consideran necesarios para una determinada organización o negocio.

2.3.1. Componentes principales de una base de datos

- Datos. Los datos son la base de datos propiamente dicha.
- Hardware. El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- Software. Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DMBS: Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.
- Usuarios. Existen tres clases de usuarios relacionados con una Base de Datos:
 1. El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
 2. El usuario final, quien accesa la base de datos por medio de un lenguaje de consulta o de programas de aplicación.
 3. El administrador de la base de datos (DBA: Data Base Administrator), quien se encarga del control general del sistema de base de datos.

2.3.2. Tipos de modelos de datos

Existen fundamentalmente tres alternativas disponibles para diseñar las bases de datos: El modelo jerárquico, el modelo de red y el modelo relacional.

2.3.2.1. El modelo jerárquico.

La forma de esquematizar la información se realiza a través de representaciones jerárquicas o relaciones de padre/hijo, de manera similar a la estructura de un árbol. Así, el modelo jerárquico puede representar dos tipos de relaciones entre los datos: Relaciones uno a uno y relaciones uno a muchos.

En el primer tipo se dice que existe una relación de uno a uno si el padre de la estructura de información tiene un solo hijo y viceversa, si el hijo tiene solamente un padre. En el segundo tipo se dice que la relación es de uno a muchos si el padre tiene más de un hijo, aunque cada hijo tenga un solo padre.

2.3.2.2. El modelo de red.

En este modelo se permite que un mismo nodo tenga varios padres. Ofrece una solución eficiente al problema de redundancia de datos; sin embargo, la dificultad para administrar los datos en una base de datos de red ha conllevado a que sea un modelo usado más por programadores que por usuarios finales.

2.3.2.3. El modelo relacional.

En este modelo toda la información se representa a través de arreglos bidimensionales o tablas. Para el manejo y consulta de la información que se almacena en estas tablas es posible efectuarles varias operaciones básicas.

Estas operaciones son:

- Seleccionar renglones de alguna tabla (SELECT)
- Seleccionar columnas de alguna tabla (PROJECT)
- Unir o juntar información de varias tablas (JOIN)

El modelo relacional se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden. El hecho de que el modelo relacional esté basado en la teoría de las matemáticas es lo que lo hace tan seguro y robusto. Al mismo tiempo, estas ramas de las matemáticas proporcionan los elementos básicos necesarios para crear una base de datos relacional con una buena estructura, y proporcionan las líneas que se utilizan para formular buenas metodologías de diseño. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica.

2.3.3. Acceso a Base de Datos.

Para desarrollar aplicaciones que conecten bases de datos, se utilizan interfaces y programas estándar que envían demandas escritas en SQL, y procesan los resultados. Para conectarse a un motor de bases de datos determinado, se necesita una interfaz estándar o controlador (en inglés: driver) que medie entre la aplicación y la base de datos.

2.3.3.1. Conectores más Utilizados.

- ODBC. Es un programa de interfaz de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacionales, utilizando para ello el lenguaje de consulta estructurado (SQL). Se administran a través de la ventana ODBC del Panel de Control, En computadoras con sistema operativo Microsoft Windows.
- MDB. Servidor de bases de datos casi profesional. Esta aplicación permite trabajar con tablas de base de datos creadas en Microsoft Access 97/2000. Es posible abrir tablas en Lenguaje de consulta estructurado, visualizarlas, navegar, crear y borrar índices, fijar relaciones, copiar, etc.
- JDBC. La conectividad de bases de datos Java (Java Database Connectivity, JDBC) es una especificación de la interfaz de aplicación de programa (Application Programming Interface, API) para conectar los programas escritos en Java a los datos en bases de datos de mayor uso.

2.3.3.2. Conectividad utilizada.

Para la implantación del sitio Web EIMWeb y cada uno de sus módulos se empleó el conector JDBC. Uno de los mayores beneficios de usar el API JDBC es la capacidad para crear aplicaciones cuya programación sea independiente de la base de datos, es decir, la mayoría de las aplicaciones que usan JDBC pueden ser migradas a otro servidor de bases de datos sin mayores complicaciones. Sin embargo, dos elementos siguen estando ligados a una base de datos en particular, el nombre de la clase que se usa para cargar el controlador (driver) JDBC y la dirección (Universal Resource Locator, URL) para acceder a la base de datos.

Los servlets y las páginas JSP usan JDBC prácticamente de la misma manera que cualquier otra aplicación en Java, típicamente los datos del controlador JDBC, la cadena de conexión, el nombre de usuario y la contraseña para conectarse a la base de datos son codificados dentro del programa.

Las operaciones básicas realizadas durante la ejecución de un controlador JDBC son:

- Cargar un controlador JDBC.
- Utilizar ese controlador para abrir una conexión con la base de datos.
- Emitir instrucciones SQL a través de la conexión.
- Procesar los conjuntos de resultados devueltos por las operaciones SQL

2.3.4. Manejadores o Gestores de Bases de Datos.

Son un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, para almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las funciones principales de un gestor de bases de datos (DataBase Manager System, DBMS) son:

- Crear y organizar la Base de datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos se puedan acceder rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.

- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos. Esto a través de las sentencias en Lenguaje Manipulador de Datos (Data Manipulation Language, DML) al comando del sistema de archivos. Así el Manejador de base de datos es el responsable del verdadero almacenamiento de los datos.
- Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- Seguridad e integridad. Consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

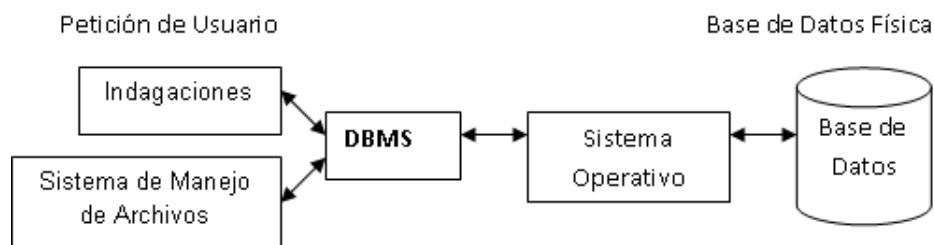


Figura 2.5: Función del DBMS.

La figura 2.2 muestra el DBMS como interfaz entre la base de datos física y las peticiones del usuario. El DBMS interpreta las peticiones de entrada-salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

Un sistema manejador de base de datos es como el cerebro de la base de datos porque se encarga del control total de los posibles aspectos que la puedan afectar.

Existen diferentes manejadores de bases de datos como MySQL, ORACLE, FoxPro, Microsoft Access y PowerBuilder.

Para la implantación y configuración del portal Web EIMWeb se utiliza MySQL.

2.3.4.1. MySQL.

MySQL es el SQL más popular en la plataforma UNIX. Es rápido y eficiente, aunque no es tan fácil de usar como otros productos similares. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandarizado para acceder a bases de datos relacionales. Sus principales características son:

- Consume pocos recursos tanto de procesador como de memoria principal en una computadora.
- Su principal objetivo de diseño fue la velocidad.
- Tiene gran disponibilidad en varias plataformas y sistemas.
- Soporta gran cantidad de datos. • Es de código abierto, puede ser usado y modificado.

2.3.4.2. Ventajas de MySQL

- Es posible manipular bases de datos enormes.
- Permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es normal que servidores Linux o Unix, usando MySQL, sirvan datos para computadoras con otros sistemas operativos.
- Permite manejar multitud de tipos para columnas.
- Permite manejar registros de longitud fija o variable.
- Acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- Seguridad, en forma de permisos y privilegios, determinados usuarios tienen permiso para consulta o modificación de determinadas tablas.
- Potencia: SQL es un lenguaje muy potente para consulta de bases de datos, usar un motor ahorra mucho trabajo. • Portabilidad: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL pueden hacerse fácilmente en otros sistemas y plataformas.

2.4. PROGRAMACIÓN DE APLICACIONES

2.4.1. Programación Estructurada (PE)

La programación estructurada es una teoría de programación que consiste en construir programas de fácil comprensión. Esta programación es especialmente útil, cuando se necesitan realizar correcciones o modificaciones después de haber concluido un programa o aplicación ya que se hace mucho más sencillo entender la codificación del programa.

2.4.1.1. Programación Orientada a Eventos

Consiste en la ejecución de código cuando se lleva a cabo una acción sobre un control. Las porciones de código a ejecutarse no son controladas por la aplicación en si, sino por el código asociado a los eventos de determinado control.

2.4.2. Programación Genérica (PG)

Cuando se diseña un algoritmo, no siempre se pone especial atención en el tipo de dato que dicho algoritmo maneja. Se dice que el algoritmo es genérico (independiente del tipo de dato) y se puede usar con cualquier tipo de dato abstracto que cumpla con ciertos requerimientos. En este tipo de programación, la atención se enfoca en la implementación de modelos definidos por conceptos.

2.4.3. Programación Orientada a Objetos (POO)

Este tipo de programación fue empleado en el desarrollo del sitio Web EISI.

En Programación orientada a objetos, el conocimiento se descentraliza en todos los objetos que lo componen, cada objeto sabe hacer lo suyo y no le interesa saber como el vecino hace su trabajo. La intención general de la POO es abstraer algunas características de sistemas naturales complejos como son:

- Estado del objeto (Atributos).
- Comportamiento del objeto (Métodos).

- Comportamientos comunes entre objetos relacionados para hallar relaciones.
- De especialización y generalización de comportamientos (Herencia).

2.4.3.1. Clases y objetos

Un objeto es cualquier cosa, real o abstracta, que posee atributos y un conjunto de operaciones que manipulan esos atributos; atributos y métodos que le dan al objeto un comportamiento particular.

Para realizar la abstracción de sistemas naturales, observamos y analizamos un grupo de cosas que tengan características comunes, el resultado de esta abstracción será válido para todas y cada una de estas cosas, y al conjunto de todas ellas lo llamamos clase. Un objeto hace referencia a una sola de estas cosas; se dice que un objeto es una instancia de la clase.

El estado del objeto se determina por el estado (valor) de sus propiedades o características (atributos), por ejemplo, si observamos el estado de un vehículo en movimiento, uno de sus atributos es la velocidad actual de desplazamiento.

2.4.3.2. Atributos

Los atributos son las características de un objeto. Son un conjunto de datos (valores) y calificadores para aquellos datos. Estos atributos pueden ser desde tipos de datos simples (enteros, caracteres, cadenas de texto) hasta otros objetos. Cuando una clase tiene como atributo otra u otras clases, se denomina a esta una clase compuesta.

2.4.3.3. Métodos

Los métodos de una clase son funciones o procedimientos propios de la clase que pueden tener acceso a los atributos de la misma para realizar las operaciones para los que son programados.

Podemos clasificar los métodos de la siguiente forma:

- Constructores: Son aquellos métodos encargados de dar un estado inicial válido a la clase; inicializan variables, reservan memoria para el proceso, abren y leen

archivos, etc. Este método se ejecuta automáticamente al instanciarse un objeto de la clase.

- Analizadores: Se puede decir que son métodos de consulta, sirven para obtener el valor actual almacenado en los atributos de la clase pero no modifican estos atributos.
- Modificadores: Estos métodos se encargan de modificar los atributos de la clase, su principal función es la de validar los datos que se pretenden asignar a las variables, evitando así la asignación de valores no válidos que puedan llevar a un estado de inconsistencia del objeto instanciado.
- Destruidores: Los métodos destructores se ejecutan automáticamente al terminarse el ciclo de vida del objeto y se encargan de devolver al sistema recursos que utilizó el objeto durante su existencia; liberan memoria asignada dinámicamente, salvan datos en los archivos abiertos y modificados, etc. Una clase solo puede tener un destructor.

2.4.3.4. Herencia

La herencia consiste en utilizar una clase ya creada para tomar sus características en clases más especializadas o derivadas de ésta para reutilizar el código que sea común con la clase base y solamente definir nuevos métodos o redefinir algunos de los existentes para ajustarse al comportamiento particular de esta sub-clase.

2.4.3.5. Ventajas de la POO

Las ventajas que presenta este tipo de programación son:

- Uniformidad: La representación de los objetos lleva implícita tanto el análisis como el diseño y la codificación de los mismos.
- Flexibilidad: Las relaciones entre los procedimientos que manipulan los datos, cualquier cambio se ve reflejado automáticamente en cualquier lugar donde estos datos aparezcan.

- Estabilidad: Dado que permite un tratamiento diferenciado de aquellos objetos que permanecen constantes en el tiempo sobre aquellos que cambian con frecuencia, permite aislar las partes del programa que permanecen inalterables en el tiempo.
- Reusabilidad: Los programas que poseen las mismas estructuras de información reutilizan las definiciones de objetos empleadas en otros programas e incluso los procedimientos que los manipulan.

2.4.3.6. JAVA.

Java es un lenguaje desarrollado por Sun Microsystems que permite el desarrollo de aplicaciones que pueden ejecutarse en casi cualquier plataforma. Java cuenta con una característica denominada “recolección de basura”, este programa examina la memoria y libera cualquier variable u objeto que no se esté usando, esto es de gran ayuda para los programadores aunque no le quita la responsabilidad de hacer programas limpios. El JDK es el entorno de desarrollo de JAVA.

Java Development Kit (JDK) Para trabajar con Java se necesita un equipo (kit) de desarrollo que proporciona:

- Un compilador: javac
- Un intérprete: java
- Un generador de documentación: javadoc
- Otras herramientas complementarias.

2.5. SERVIDOR.

Un servidor es un tipo de software que suministra servicios a los usuarios o terminales que lo solicitan. Por ejemplo, en una típica arquitectura cliente-servidor, el cliente podría ser un computador que realiza peticiones de información a través de un programa de correo (Outlook Express por ejemplo) y, el servidor le entrega los datos en forma de correos electrónicos en respuesta a su solicitud.

Un servidor sirve información a los computadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

Los servidores se conectan a la red mediante una interfaz que puede ser una red verdadera o mediante conexión vía línea telefónica o digital.

2.5.1. Tipos de Servidores.

Esta lista categoriza los diversos tipos de servidores del mercado actual:

- **Plataformas de Servidor (Server Platforms):** Un término usado a menudo como sinónimo de sistema operativo, la plataforma es el hardware o software subyacentes para un sistema, es decir, el motor que dirige el servidor.
- **Servidores de Aplicaciones (Application Servers):** Designados a veces como un tipo de middleware (software que conecta dos aplicaciones), los servidores de aplicaciones ocupan una gran parte del territorio entre los servidores de bases de datos y el usuario, y a menudo los conectan.
- **Servidores de Audio/Video (Audio/Video Servers):** Los servidores de Audio/Video añaden capacidades multimedia a los sitios web permitiéndoles mostrar contenido multimedia en forma de flujo continuo (streaming) desde el servidor.
- **Servidores de Chat (Chat Servers):** Los servidores de chat permiten intercambiar información a una gran cantidad de usuarios ofreciendo la posibilidad de llevar a cabo discusiones en tiempo real.
- **Servidores de Fax (Fax Servers):** Un servidor de fax es una solución ideal para organizaciones que tratan de reducir el uso del teléfono pero necesitan enviar documentos por fax.
- **Servidores FTP (FTP Servers):** Uno de los servicios más antiguos de Internet, File Transfer Protocol permite mover uno o más archivos.
- **Servidores Groupware (Groupware Servers):** Un servidor groupware es un software diseñado para permitir colaborar a los usuarios, sin importar la localización, vía Internet o vía Intranet corporativo y trabajar juntos en una atmósfera virtual.

- **Servidores IRC (IRC Servers):** Otra opción para usuarios que buscan la discusión en tiempo real, Internet Relay Chat consiste en varias redes de servidores separadas que permiten que los usuarios conecten el uno al otro vía una red IRC.
- **Servidores de Listas (List Servers):** Los servidores de listas ofrecen una manera mejor de manejar listas de correo electrónico, bien sean discusiones interactivas abiertas al público o listas unidireccionales de anuncios, boletines de noticias o publicidad.
- **Servidores de Correo (Mail Servers):** Casi tan ubicuos y cruciales como los servidores web, los servidores de correo mueven y almacenan el correo electrónico a través de las redes corporativas (vía LANs y WANs) y a través de Internet.
- **Servidores de Noticias (News Servers):** Los servidores de noticias actúan como fuente de distribución y entrega para los millares de grupos de noticias públicos actualmente accesibles a través de la red de noticias USENET.
- **Servidores Proxy (Proxy Servers):** Los servidores proxy se sitúan entre un programa del cliente (típicamente un navegador) y un servidor externo (típicamente otro servidor web) para filtrar peticiones, mejorar el funcionamiento y compartir conexiones.
- **Servidores Telnet (Telnet Servers):** Un servidor telnet permite a los usuarios entrar en un ordenador huésped y realizar tareas como si estuviera trabajando directamente en ese ordenador.

2.5.1.1. Servidores Web.

Un servidor Web es un programa que sirve datos en forma de páginas Web, hipertextos o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP.

Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que solemos conocer como un navegador Web.

A modo de ejemplo: al teclear `http://mecanica.uis.edu.co` en un navegador, éste realizará una petición HTTP al servidor que tiene asociada dicha URL. El servidor responde al cliente enviando el código HTML de la página; el navegador cuando recibe el código, lo interpreta y lo muestra en pantalla.

El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página. El servidor se encarga de transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

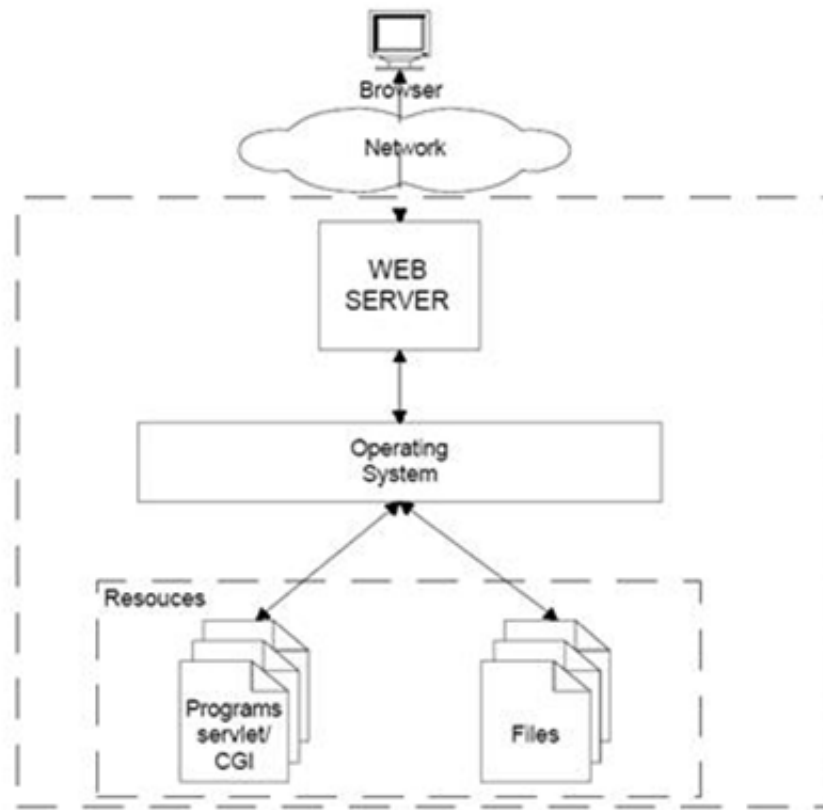


Figura 2.6: Funcionamiento de una servidor web.

Arquitectura de un Servidor Web. La arquitectura habitual de un servidor web se divide en dos (2) capas:

1. **Capa servidor.**

Esta capa contiene cinco subsistemas, que son los responsables de implementar la funcionalidad de un servidor Web:

- **Subsistema de recepción:** representa la primera “línea de ataque” y su labor consiste en esperar las peticiones HTTP de los clientes que llegan por la red. También, analiza las peticiones y determina las capacidades de los navegadores (tipo de navegador, compatibilidad, etc.). Este subsistema contiene la lógica necesaria para manejar múltiples peticiones.
- **Analizador de peticiones:** encargado de traducir la localización del recurso de la red al nombre del archivo local. Por ejemplo, la solicitud del recurso `http://mecanica.uis.edu.co` se traduce al fichero local `$CATALINA_HOME/webapps/eisi/index.jsp`.¹
- **Control de acceso:** sirve para autenticar y permitir el acceso.
- **Manejador de recursos:** este subsistema es el responsable de determinar el tipo de recurso solicitado; lo ejecuta y genera la respuesta.
- **Registro de transacción:** se encarga de registrar todas las peticiones y su resultado.

2. Capa de Soporte.

Esta capa actúa como una interfase entre el sistema operativo y el servidor Web y, entre los propios subsistemas de la capa superior. Se compone de los siguientes subsistemas:

- **Util:** contiene funciones que son utilizadas por el resto de subsistemas.
- **Capa abstracta del Sistema Operativo (OSAL):** Este subsistema encapsula el funcionamiento específico del sistema operativo para facilitar la portabilidad del servidor Web a diferentes plataformas.

Los servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas Web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros lenguajes como PHP, los CGIs (common Gateway interface), o ASP (Active Server Pages). Sin embargo, se diferencian de ellos en otras cosas.

¹`$CATALINA_HOME` es una variable del sistema servidor que apunta al directorio de instalación del contenedor de servlets TOMCAT.

Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet se ejecuta en su propio contexto pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página Web. Ambos necesitan un programa que los contenga, y sea el que envíe efectivamente páginas Web al servidor, reciba las peticiones, las distribuya entre los servlets, y lleve a cabo todas las tareas de gestión propias de un servidor Web. Mientras que servidores como el Apache están especialmente pensados para páginas Web estáticas, CGIs, y programas ejecutados por el servidor, tales como PHP, hay otros servidores específicos para servlets y JSPs llamados contenedores de servlets (servlet containers) o servlet engines.

2.5.1.2. Servidor Web Apache

Apache es un proyecto nacido para crear un servidor Web estable, fiable y veloz para plataformas Unix. Está disponible para una gran multitud de plataformas:

- GNU/Linux
- Mac OS y Mac OS X Server
- Netware
- OpenStep/Match
- UNIX comerciales como AIX (R), Digital UNIX (R), HP-UX (R), IRIX (R),
- SCO(R), Solaris (R), SunOS (R), UnixWare (R)
- Windows (R)

Características del servidor Apache

1. Independencia de plataforma: Apache funciona en casi todas las Plataformas actuales; debido a esto, podemos escoger la plataforma que más se adapte a nuestras características, produciendo una independencia tecnológica del fabricante de hardware lo que hace que se ofrezcan productos de calidad a los clientes.
2. Autenticación de diferentes tipos: Apache permite el uso de bases de datos DBM para la autenticación de usuarios. De esta forma se puede restringir el acceso a determinadas páginas de un sitio Web de una forma sencilla y de fácil mantenimiento.
3. Respuestas personalizadas ante errores del servidor: Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
4. Creación de contenidos dinámicos: Apache permite la creación de sitios Web dinámicos mediante el uso de CGI's, lenguajes de Scripting como PHP y javascript, y el uso de Java y páginas jsp.

Contenedor Tomcat Tomcat es un subproyecto de Yakarta que provee un poderoso servidor Web con soporte a Java Servlets y JSP. Se entrega bajo licencia de software para Apache y es open source.

Apache y Tomcat en trabajo conjunto Para que Apache procese la peticiones JSP, se necesita conectarlos por medio de un modulo llamado JK. El conector mod_jk está desarrollado por jakarta, que son los desarrolladores de Apache y Tomcat. El JK es un conector que permite al contenedor de Java Server Pages (JSP) Jakarta Tomcat interactuar con servidores web como Apache, Netscape, iPlanet, SunOne e incluso IIS usando el protocolo AJP.

La principal funcionalidad de este módulo es permitir a servidores de aplicaciones o al servidor Jakarta Tomcat enlazarse con un servidor web. Este servidor web, típicamente el servidor HTTP Apache, introduce una mayor gestión en las conexiones de los clientes y mayor la seguridad en las transacciones del sistema. Así mismo se puede enlazar varias instancias al servidor web permitiendo así una mayor tolerancia a errores y aligerar la carga en los servidores Java.

Capítulo 3

MARCO METODOLOGICO.

3.1. PROTOTIPADO EVOLUTIVO.

Para realizar la implantación y configuración del sitio Web en la escuela de Ingeniería Mecánica, la creación de un sitio plantilla, completamente parametrizado, se propone como metodología de desarrollo el Prototipado Evolutivo. La elección de la metodología se debe a las siguientes razones:

- La Escuela de Ingeniería de Sistemas debe estar actualizando su sitio Web y para ello nace la necesidad de desarrollar nuevas versiones de ella en un tiempo considerablemente corto debido a que el sistema no es un producto final sino que al contrario es sometido a una permanente reconstrucción.
- Es deseable tener un esbozo de lo que se desee mejorar o crear para poder incorporar sugerencias de cambio por parte de la Escuela de Ingeniería de Sistemas en etapas tempranas del desarrollo.
- Por otra parte, es necesario saber lo antes posible si hemos interpretado correctamente las especificaciones y las necesidades de la Escuela.
- En muchos casos los usuarios no tienen una idea acabada de lo que desean, por lo tanto debemos tomar decisiones y suponer qué es lo que el usuario quiere. Por este motivo, la emisión de los prototipos brinda la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado.

- La decisión se fundamenta en la ventaja de la realización de los cambios en etapas tempranas y la posibilidad de emisión de varios prototipos evaluables durante el desarrollo, obteniéndose de este modo paralelamente una metodología integral también para el proceso de evaluación del programa.
- Esta metodología propicia un intercambio de conocimientos y de autocrítica al sistema, lo que conlleva a que se produzcan muchas pruebas antes de liberar una nueva versión así como mejoras rápidas a problemas que puedan surgir durante su uso.
- Para la implantación del sistema en cualquier otra escuela, se tiene la certeza de que el sistema está completamente parametrizado, y esto facilita que nuevos módulos desarrollados pasen por una fase de prueba antes de ser implantados en el sistema que este funcionando en otra escuela, solo es realizar los cambios en la interfaz de parámetros que se adapte a cada una de las escuelas.

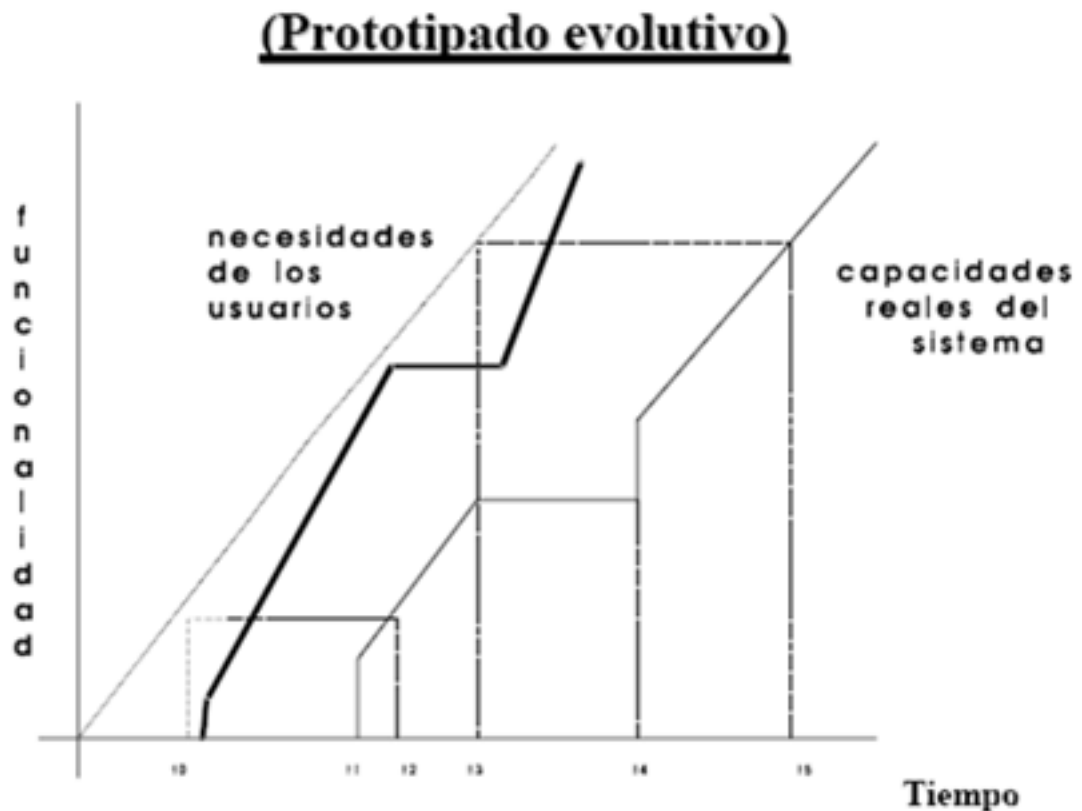


Figura 3.1: Prototipado Evolutivo.

3.1.1. Procedimiento a seguir para la metodología planteada.

La construcción de prototipos comienza con la Recolección de los Requisitos.

- El desarrollador y usuario se reúnen y definen los objetivos globales para el software, identifican todos los requisitos conocidos y perfilan las áreas en donde será necesaria una mayor definición.
- Luego se produce el Diseño del Prototipo que se enfoca sobre la representación de los aspectos del software visibles al usuario (por ejemplo, métodos de entrada y formatos de salida) y se prosigue a su construcción.
- El prototipo es evaluado por el usuario y se utiliza para refinar los requisitos del software a desarrollar.
- Se produce un proceso interactivo en el que el prototipo es “afinado” (Refinamiento del prototipo) para que satisfaga las necesidades del usuario, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer y poder entregar el producto final requerido o Producto de Ingeniería.

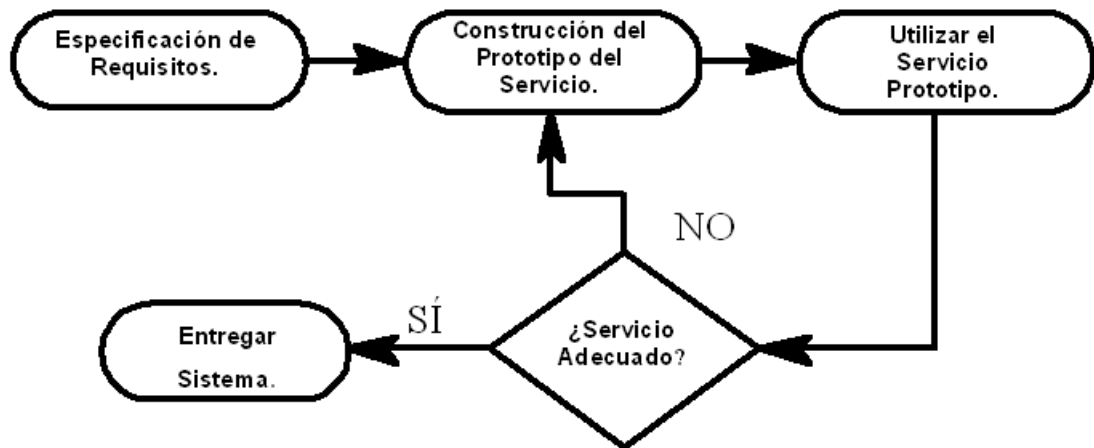


Figura 3.2: Diagrama de flujo del Prototipo Evolutivo

3.2. EFECTIVIDAD DEL PROTOTIPADO EVOLUTIVO.

Para que el prototipo sea efectivo debe cumplir las siguientes características:

- Debe ser un sistema con el que se pueda experimentar.
- Debe ser comparativamente barato ($< 10\%$).
- Debe desarrollarse rápidamente.
- Énfasis en la interfaz de usuario.
- Equipo de desarrollo reducido.
- Herramientas y lenguajes adecuados.

3.3. PROBLEMAS DEL PROTOTIPADO EVOLUTIVO.

El prototipado evolutivo no es perfecto y puede presentar los siguientes problemas:

- Los procesos de gestión existentes presuponen que el modelo de desarrollo es en cascada
- Los cambios continuos provocan la destrucción de la estructura del sistema, por lo que lleva a un encarecimiento del posterior mantenimiento
- El ciclo de vida de los sistemas desarrollados con esta técnica es necesariamente corto
- Se debe contar con especialistas.

Capítulo 4

INSTALACIÓN Y CONFIGURACIÓN DEL *SERVIDOR.*

En este capítulo se presentará un manual paso a paso de la instalación y configuración básica del sistema operativo Mac OS X server Tiger para la el *servidor* de la Escuela de Ingeniería Mecánica EIMWeb.

4.1. HARDWARE.

A continuación se presentan algunas especificaciones referentes a la máquina en la cual esta alojado el servidor web. En adelante nos referiremos a él como *servidor*.

- Procesamiento: Dos (2) Intel Xeon Doble Nucleo, 2.0GHz.
- Bus de sistema: Bus frontal 1.33GHz por procesador.
- L2 caché: 4MB compartida por procesador.
- Memoria de acceso aleatorio (RAM): Cuatro (4) tarjetas de 512MB a 667MHz, DDR2 FB-DIMM.
- Almacenamiento: Dos discos duros de 80GB SATA, 7200rpm, 8MB Caché, SAS configurable a un sistema Mirrored RAID 1 o sistema Stripped RAID 0.

- Conectividad: Dos (2) puertos externos FireWire 800, Un (1) puerto externo FireWire 400, Serial DB-9, Dos (2) puertos seriales universales USB 2.0.
- Video: Tarjeta ATI Radeon X1300 64MB, PCI-E GDDR3 SDRAM.
- Red: Dos (2) Tarjetas de Red Ethernet 10/100/1000BASE-T Gigabit y conectores RJ-45.
- Marca: Apple Xserver.



Figura 4.1: Máquina servidora para el portal web de la Escuela de Ingeniería Mecánica.

4.2. INSTALACIÓN DEL SISTEMA OPERATIVO.

El sistema operativo correspondiente a esta versión del *servidor* es Mac OS X Server Tiger 10.4.7. actualizado a la versión 10.4.11.

Debido a que este sistema operativo esta diseñado exclusivamente para este *servidor*, no es posible instalarlo en una máquina diferente, como por ejemplo, un equipo de escritorio habitual o incluso un computador de escritorio o portátil fabricado por Apple. El sistema detecta automáticamente el hardware donde va ser instalado, validando que corresponda a una versión válida de un Xserver.¹

¹Xserver, Mac OS X Server Tiger son una marca registrada de Apple Inc. registrada en E.U y otros países.

4.2.1. Pantalla de Bienvenida.



Figura 4.2: Pantalla de bienvenida.

Esta primer pantalla nos muestra los idiomas disponibles para la instalación; vemos que solo incluye los idiomas inglés, francés, alemán y japonés. Escogemos el idioma inglés como idioma principal tanto de la instalación como de la interfaz gráfica del sistema operativo.

4.2.2. Configuración de Discos Duros.

4.2.2.1. Sistema de Archivos.

A diferencia de Linux que utiliza para la mayoría de sus particiones, incluyendo la raíz (/), el sistema de archivos ext3² (tercer sistema de archivos extendido) y Windows que

²Actualmente en algunas distros con sistemas operativos Linux se está manejando el ext4 (cuarto sistema de archivos extendido).

utiliza el NTFS (Sistema de Archivos NT), Mac OS X Server Tiger utiliza el HFS plus (sistema jerárquico de archivos extendido).

HFS plus es un sistema de archivos desarrollado por Apple Inc. para reemplazar al HFS (Sistema jerárquico de archivos). También es el formato usado por el iPod al ser formateado desde un Mac. HFS Plus también es conocido como HFS Extended y Mac OS Extended. Durante el desarrollo, Apple se refirió a él con el nombre clave Sequoia.

HFS Plus es una versión mejorada de HFS, soportando archivos mucho más grandes (Bloques direccionables de 32 bits en vez de 16) y usando Unicode (En vez de Mac OS Roman) para el nombre de los archivos, lo que además permitió nombres de archivo de hasta 255 letras.

HFS Plus permite nombres de fichero de hasta 255 caracteres de longitud UTF-16, y archivos n-bifurcados similares a NTFS, aunque casi ningún software se aprovecha de bifurcaciones con excepción de la bifurcación de los datos y de la bifurcación del recurso. HFS Plus también utiliza tabla de asignación de 32 bits, en lugar de los 16 bits de HFS. Ésta era una limitación seria de HFS, significando que ningún disco podría apoyar más de 65.536 bloques de la asignación sobre de HFS.

4.2.2.2. Estructura de Directorios.

El sistema operativo Mac OS X server Tiger esta basado en UNIX®³ y tiene una estructura de directorios definida a partir de la raíz (/) que en el proceso de instalación es inmodificable; Se describe a continuación:

- /Applications: Directorio donde, teóricamente, debemos organizar nuestras aplicaciones; allí es dónde están las aplicaciones del sistema recién instalado. Mencionamos especialmente al subdirectorio /Applications/Utilities donde podemos encontrar utilidades del sistema como la utilidad de discos o el monitor del sistema.
- /Developer: Aparece si se tiene instalado la herramientas de desarrollo de Apple (Apple's Developer Tools) y contiene las herramientas y entornos como XCode, documentación y archivos relacionados.

³Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

- */Library*: Librerías compartidas, archivos necesarios para que el sistema y las aplicaciones funcionen correctamente, también incluye preferencias, configuraciones y otros archivos de personalización. También existe en el directorio *home*, entonces recoge dichos ficheros relacionados con el usuario activo.
- */Network*: Archivos de red, conexiones con servidores, librerías de conexión, etc.
- */System*: Archivos de sistema, librerías preferencias y configuración críticas para el arranque y el funcionamiento correcto de Mac OS X.
- */Users*: Las cuentas de usuario del *servidor* y todas las configuraciones únicas, bastante parecido al directorio */home* de Linux.
- */Volumes*: Dispositivos y volúmenes que se han montado en el sistema ya sean virtuales o reales como discos físicos, CDs, DVDs, discos de imagen DMG, etc.
- */* la raíz: presente en todos los sistemas derivados de UNIX. es el directorio padre de todo el árbol de directorios.
- */bin*: Directorio de binarios comunes, mantiene archivos relacionados con el arranque y las operaciones básicas del sistema, es necesario para que éste funcione correctamente.
- */etc*: Configuración local del sistema, mantiene datos de administración y configuración de los archivos del sistema.
- */dev*: Ficheros de dispositivo, representan periféricos como teclado, ratón, etc.
- */usr*: El segundo en nivel de importancia incluye subdirectorios que continene información sobre la configuración del sistema.
- */sbin*: Otro directorio que contiene binarios esenciales para la administración del sistema.
- */tmp*: Archivos temporales, cache, etc.
- */var*: Directorio de datos variables, contiene archivos cuyo contenido varía mientras el sistema se ejecuta.

4.2.2.3. Sistema de Almacenamiento.

El instalador del sistema operativo incluye un menú en la parte superior de la ventana con algunas herramientas para modificar algunos parámetros antes de empezar la instalación. Entre ellas encontramos la herramienta “Utilidad de Discos” que nos permite gestionar el particionamiento y manejo de los discos duros que estén instalados en el *servidor* (Ver Figura 4.3). A través de esta aplicación procedemos a configurar un sistema de Discos en RAID. A continuación se muestra una breve explicación acerca de este sistema:

El acrónimo RAID (del inglés Redundant Array of Independent Disks, «conjunto redundante de discos independientes», originalmente era conocido como Redundant Array of Inexpensive Disks, «conjunto redundante de discos baratos») hace referencia a un sistema de almacenamiento que usa múltiples discos duros entre los que distribuye o replica los datos. Dependiendo de su configuración (a la que suele llamarse «nivel»), los beneficios de un RAID respecto a un único disco son uno o varios de los siguientes: mayor integridad, mayor tolerancia a fallos, mayor throughput (rendimiento) y mayor capacidad. En sus implementaciones originales, su ventaja clave era la habilidad de combinar varios dispositivos de bajo coste y tecnología más antigua en un conjunto que ofrecía mayor capacidad, fiabilidad, velocidad o una combinación de éstas que un solo dispositivo de última generación y coste más alto.

En el nivel más simple, un RAID combina varios discos duros en una sola unidad lógica. Así, en lugar de ver varios discos duros diferentes, el sistema operativo ve uno solo. Los RAID suelen usarse en servidores y normalmente (aunque no es necesario) se implementan con unidades de disco de la misma capacidad. Debido al decremento en el precio de los discos duros y la mayor disponibilidad de las opciones RAID incluidas en los chipsets de las placas base, los RAID se encuentran también como opción en los ordenadores personales más avanzados. Esto es especialmente frecuente en los computadores dedicados a tareas intensivas de almacenamiento, como edición de audio y vídeo.

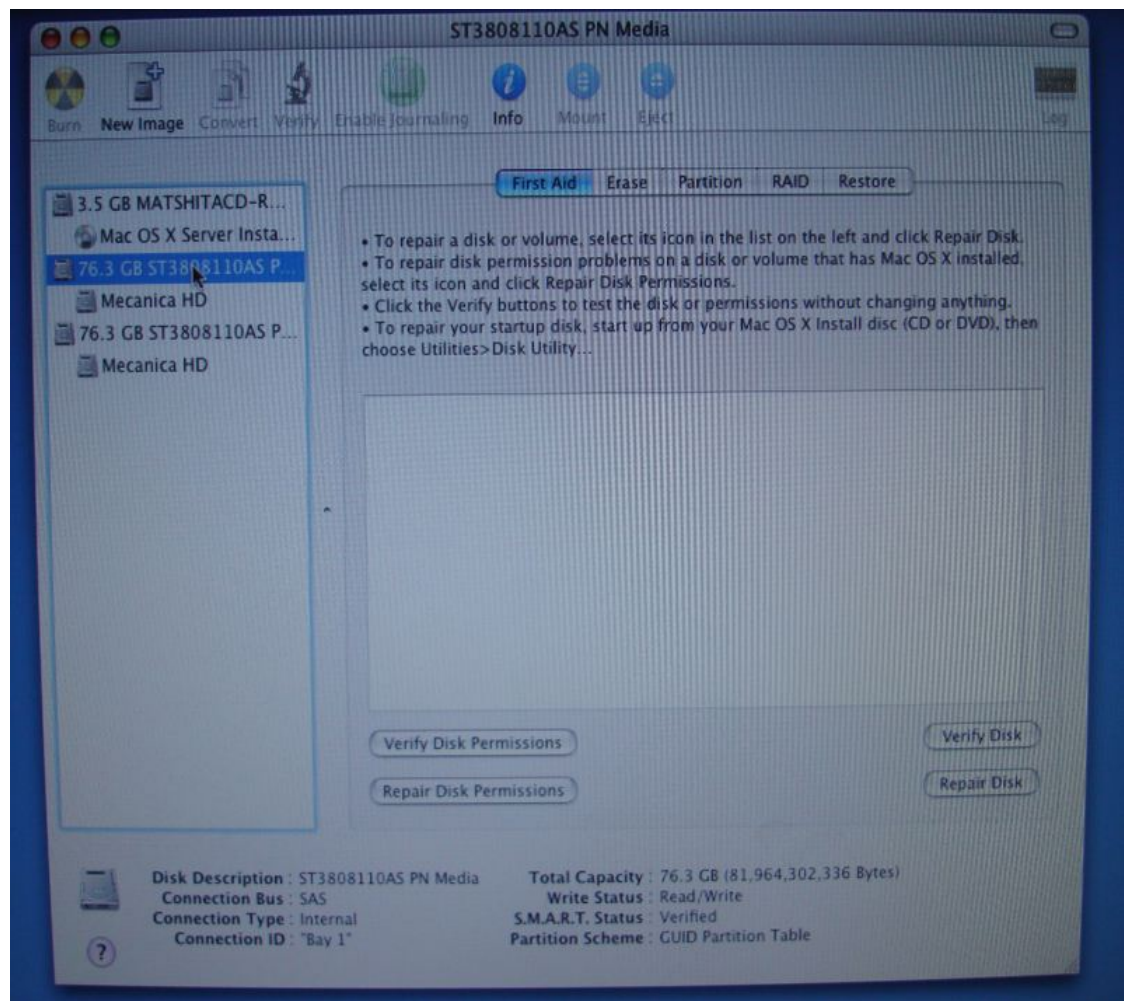


Figura 4.3: Utilidad de Discos.

Para crear un conjunto RAID, como se ha explicado, debemos disponer de dos o más discos duros SCSI o IDE. Al crear el grupo, debemos especificar el tipo de grupo RAID que corresponde a los requisitos de las aplicaciones con las que se utilizará. El sistema operativo Mac OSX Server Tiger permite crear:

- "Grupo RAID con misma información en cada disco" duplica los archivos en cada disco del grupo de tal modo que, si un disco falla, siempre tenga una copia de seguridad actualizada de toda su información. Además, este tipo de grupo le permite continuar procesando transacciones.
- "Grupo RAID striped (tratado como un solo disco)" almacena los archivos en segmentos ("stripes") de todos los discos del grupo. Esta opción de grabación

alternada aumenta el rendimiento de las aplicaciones que emplean grandes cantidades de datos. También puede usar varios discos o volúmenes para crear un grupo "concatenado" que utilice los discos del grupo como si fueran un solo disco. Pese a que los grupos concatenados no son exactamente del tipo RAID, puede utilizarlos con un grupo RAID.

Si se ha arrancado el *servidor* desde un disco con varias particiones, no podrá crear un grupo RAID que contenga particiones en el disco de arranque. Para poder crear un disco RAID que contenga particiones en este último disco, debemos arrancar el servidor desde otro disco.

Debido a que el Sistema de información del portal web de la Escuela de Ingeniería Mecánica, requiere enormemente que la información contenida sea fiable, este disponible en cualquier momento y siempre este protegida frente algún fallo, escogemos la opción de Grupo de RAID con misma información en cada disco o también conocido como "RAID 1".

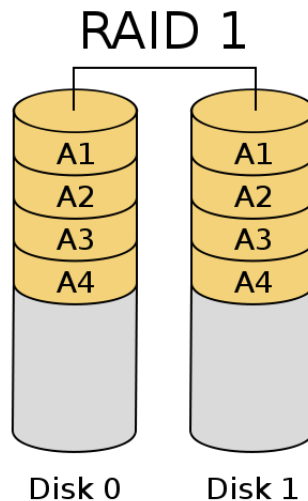


Figura 4.4: Sistema RAID 1.

Lo que haremos a continuación será proceder a crear el sistema RAID 1, para esto seguiremos los siguientes pasos⁴:

- Abrimos "Utilidad de Discos", seleccionamos uno de los discos de la lista que deseamos utilizar en el grupo y hacemos clic en RAID.

⁴Algunos términos utilizados en los pasos de creación del sistema RAID se han traducido al español. Recordemos que el instalador está en idioma inglés.

- Asignamos un nombre al grupo RAID y seleccionamos un formato en el menú local "Formato del volumen".
- Seleccionamos un tipo de RAID en el menú local.
- Arrastramos los discos que deseamos integrar en el grupo hasta la lista de la derecha. Si creamos un grupo RAID duplicado podemos elegir un tipo para cada disco en el menú local "Tipo de RAID". Seleccionamos "fragmento" para convertir el disco en parte integral del grupo o "reserva" para que el grupo RAID únicamente utilice el disco si falla alguno de los otros discos.
- Hacemos clic en el botón Opciones para especificar un tamaño de bloque óptimo para los datos almacenados en el grupo. Si decidimos crear un grupo RAID duplicado, marque la casilla "Autorreconstrucción del mirror RAID". Hacemos clic en OK cuando haya terminado.

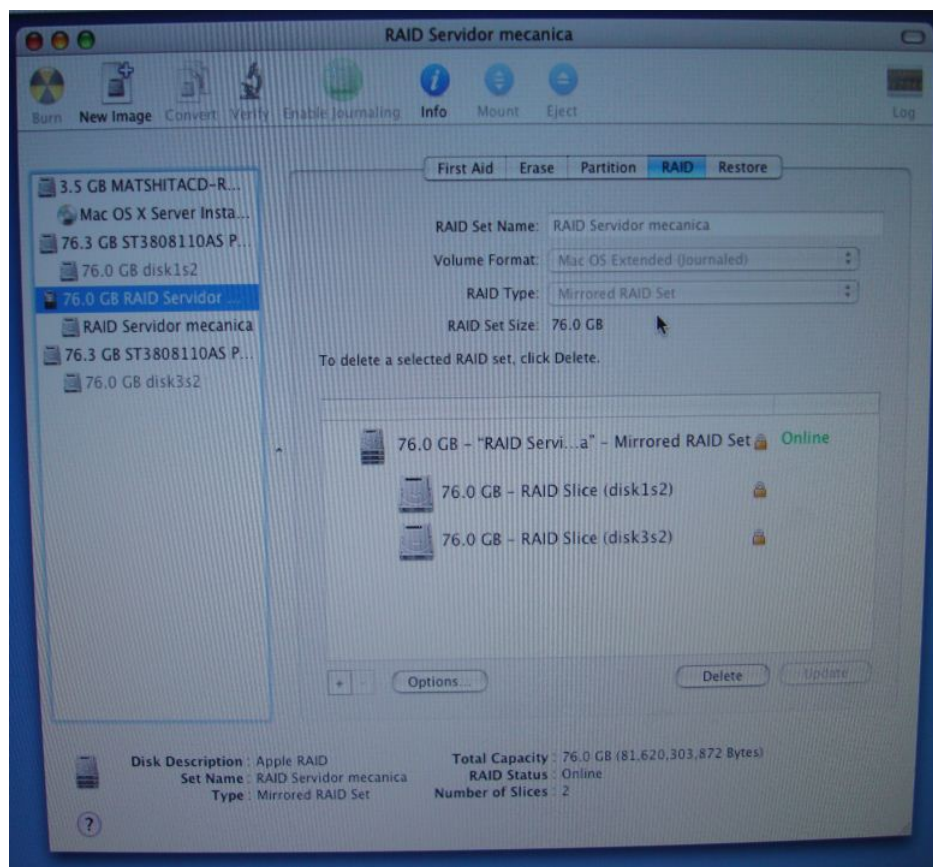


Figura 4.5: Utilidad de Discos después de creación de RAID nivel 1.

Tras crear un grupo RAID, el software RAID gestiona el almacenamiento de archivos en los discos.

4.2.3. Paquetes de Instalación adicionales.

En caso de ser necesario instalar algunos otros paquetes adicionales, que no se incluyen en una instalación por defecto, tenemos la posibilidad de hacerlo a través del botón “Personalizar” de la ventana principal del instalador.

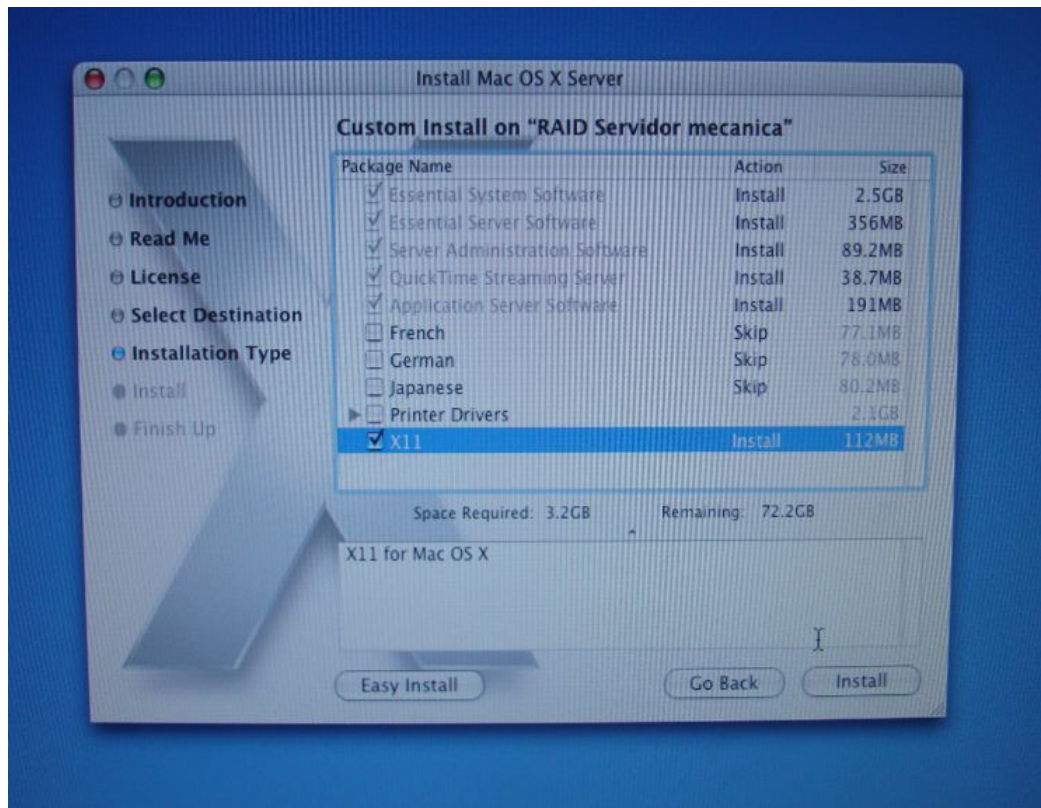


Figura 4.6: Paquetes Adicionales.

4.2.4. Instalación.

Teniendo listo todo lo descrito anteriormente, procedemos a seleccionar el destino de la instalación. Ver Figura 4.7.

A continuación empezará el proceso de instalación. Tomará un tiempo que oscila entre los 20 y 25 minutos, dependiendo de los paquetes adicionales que se quieran en el sistema

operativo. Ver Figura 4.8.

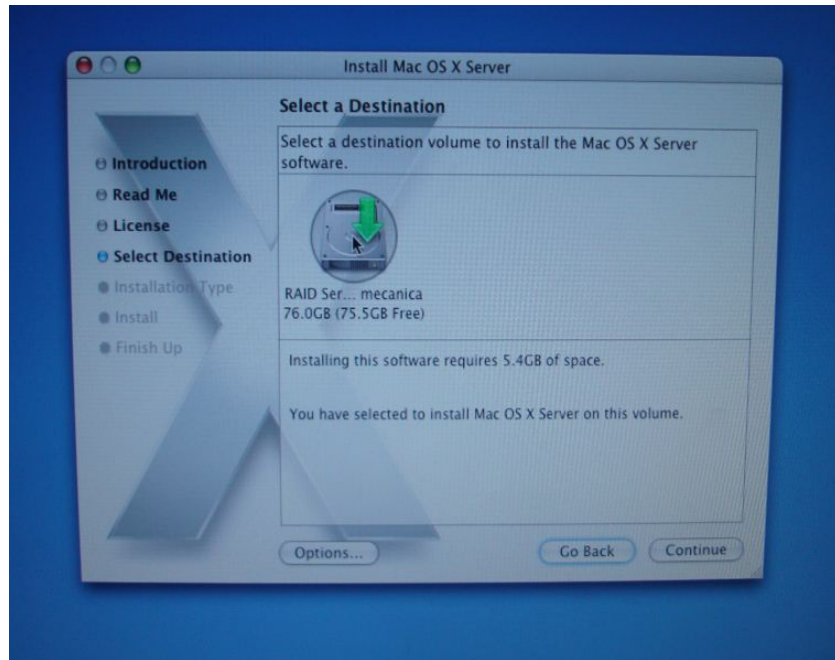


Figura 4.7: Destino de Instalación.

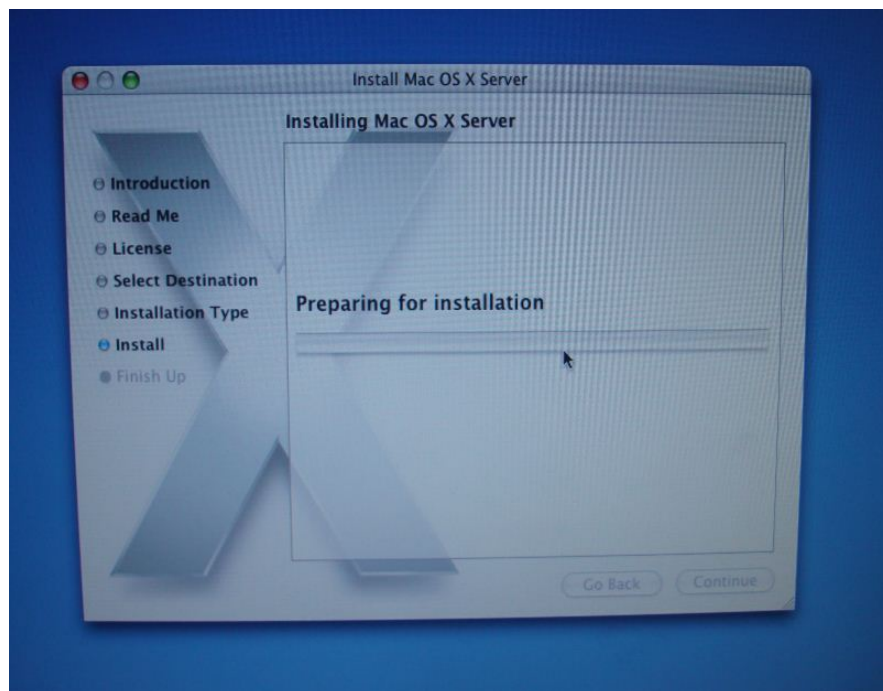


Figura 4.8: Ventana de instalación.

Al terminar este proceso el sistema reiniciará automáticamente.

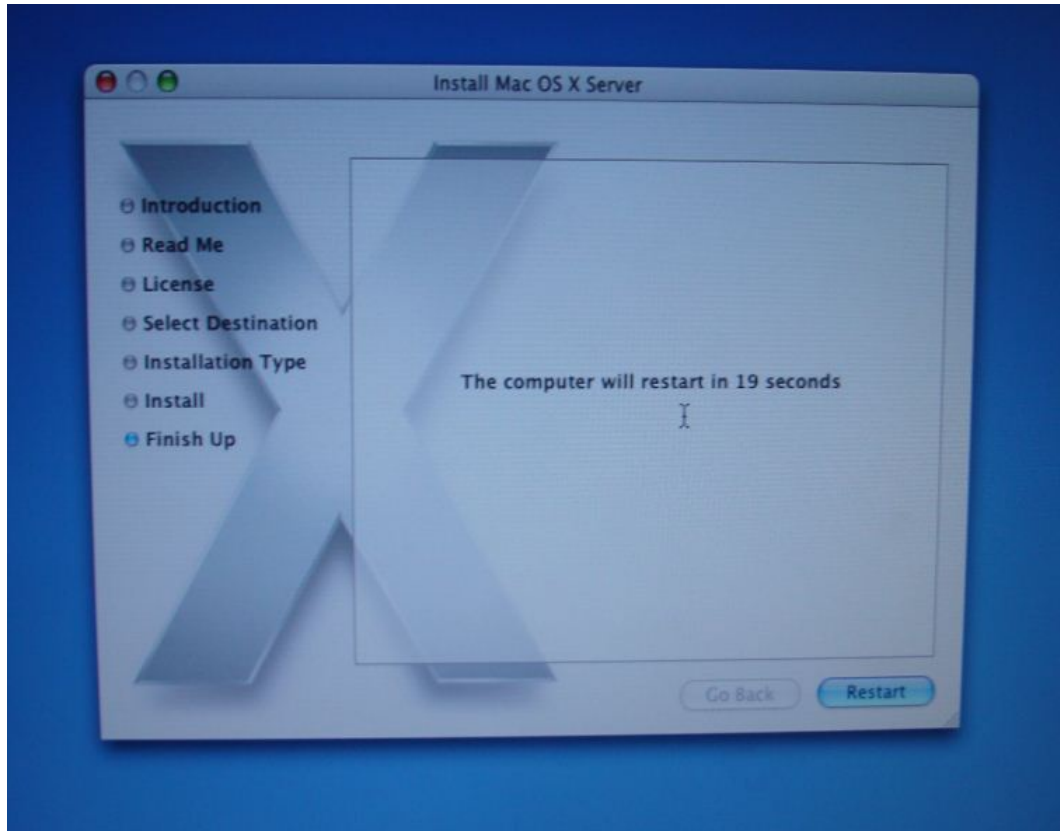


Figura 4.9: Finalización de la instalación.

4.3. CONFIGURACIÓN INICIAL.

En este punto presentaremos el proceso de inserción de información necesaria para la creación de una configuración inicial del servidor, activación de servicios básicos y preparación del *servidor* para el proceso de configuración avanzada que veremos más adelante.

4.3.1. Cuenta de Administración.

Aquí se refiere al usuario con el cual podemos administrar todo el sistema operativo. Este usuario no es el usuario ROOT ⁵; nombre convencional de la cuenta de usuario que

⁵En sistemas operativos de tipo Unix.

posee todos los derechos en todos los modos (mono o multi usuario). ROOT es también llamado superusuario. Normalmente esta es la cuenta de administrador. El usuario ROOT puede hacer muchas cosas que un usuario común no, tales como cambiar el dueño de archivos y enlazar a puertos de numeración pequeña, entre muchas otras. En Mac OS X server Tiger el usuario ROOT está deshabilitado por defecto, lo cual esta definido de esa manera por cuestiones de seguridad.

El sistema operativo nos pedirá el nombre completo de la cuenta, un nombre corto para logueo (aunque con los dos nombres es posible entrar al sistema) y una clave de acceso.

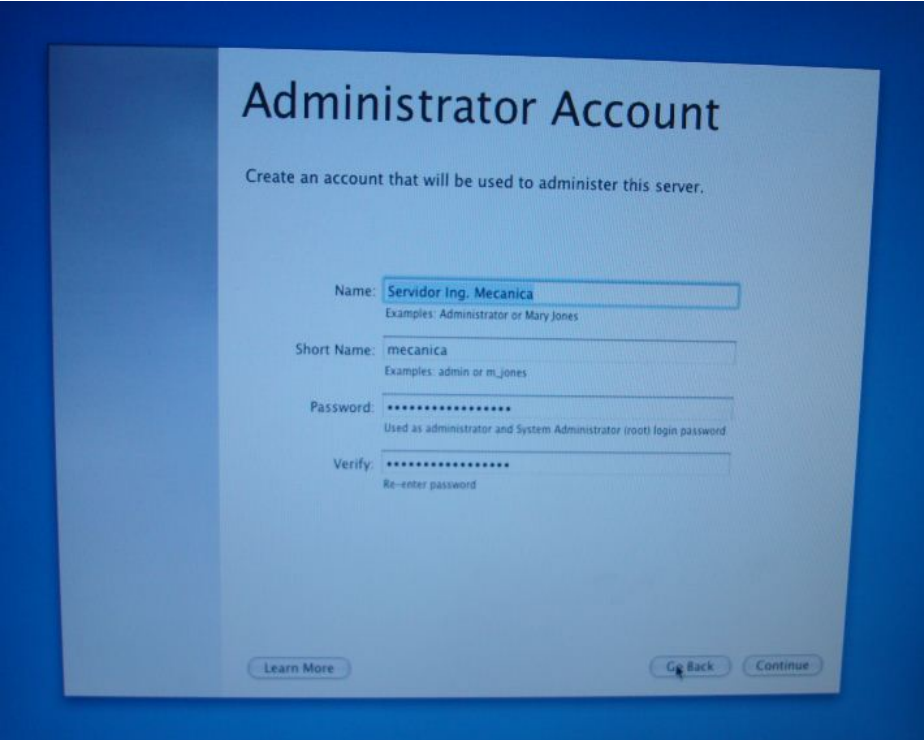


Figura 4.10: Creación de Usuario Administrador.

4.3.2. Nombres de Red.

En este apartado debemos configurar el Nombre de Host (Hostname), el nombre del computador y el nombre local del servidor (Localhost):

- Nombre de Servidor (Computer Name): Este concepto nada tiene que ver con el mencionado Nombre de host o Hostname. Se usa para que los clientes que usan el

modo de conexión AFP⁶ y servicios de impresión, los cuales, estén administrados por el *servidor*.

- Nombre Local del Servidor (Local Hostname): Es el nombre con el cual los usuarios de la red interna del *servidor*, pueden identificar al mismo. En general se requiere para poder observar a la máquina servidora en una red interna, ser usuario de un sistema operativo Mac OS X, sin embargo es posible que un usuario con el sistema operativo Windows, también pueda hacerlo utilizando el protocolo de red SMB (samba).

4.3.2.1. Nombre Único de Máquina o Hostname.

Un nombre de host es un nombre único para el *servidor* que se refiere históricamente a el nombre de máquina de UNIX. El hostname en Mac OS X Server por defecto el hostname es configurado dinámicamente, siendo controlado por el archivo `/etc/hostconfig`.

Para poder determinar automáticamente el hostname, el sistema operativo valida que un nombre sea correcto en los siguientes casos, en orden de aparición:

1. El nombre que se provee por DHCP para la dirección IP primaria.
2. El primer nombre que es retornado por una consulta a la dirección IP primaria a través de DNS inverso (dirección a IP).
3. El nombre local del *servidor* o también llamado “local hostname”.
4. El nombre “localhost”.

En nuestro caso no es posible cambiar este nombre “hostname” en este asistente. Por defecto el valor es llevado a automático.

4.3.3. Configuración de Red.

El *servidor* cuenta con dos (2) tarjetas de red Ethernet. Para nuestro propósito solo será necesario configurar una de ellas. A través de este asistente es muy fácil poder hacerlo, solo debemos suministrar la información que nos solicita⁷.

⁶El Apple Filing Protocol (AFP) es un protocolo de capa de presentación (según el modelo OSI) que ofrece servicios de archivos para Mac OS X y Mac OS Classic..

⁷Por razones de seguridad y confidencialidad no se mostrarán en este documento.

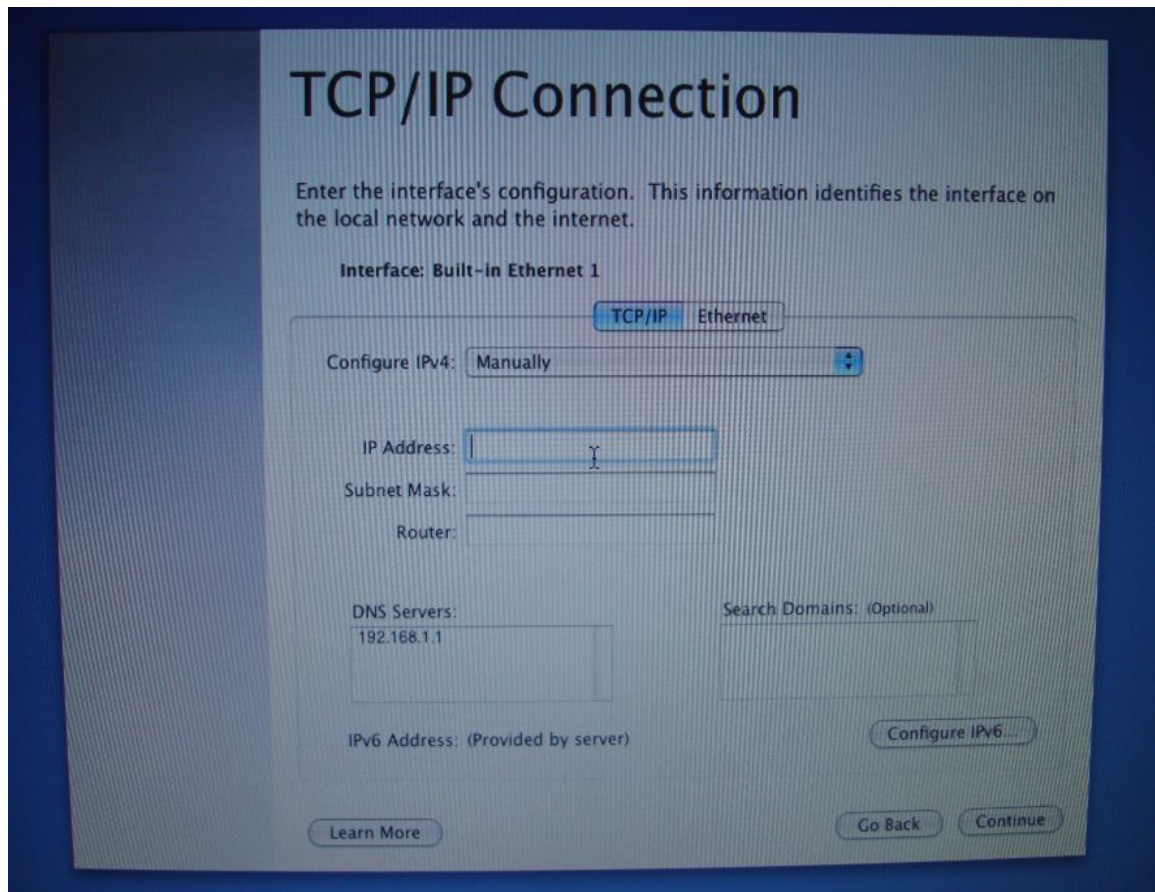


Figura 4.11: Asistente de Configuración de Red

4.3.4. Servicios Básicos.

Los servicios que se muestran en este apartado, corresponde a aquellos que no necesitan una configuración mas detallada y que desde este momento pueden ser activados y funcionar sin ningún problema. Para el *servidor* solo será necesario activar el servicio de Actualización horario por red (Network Time Services) quien nos permitirá tener actualizada la hora y fecha del *servidor* a través de internet.

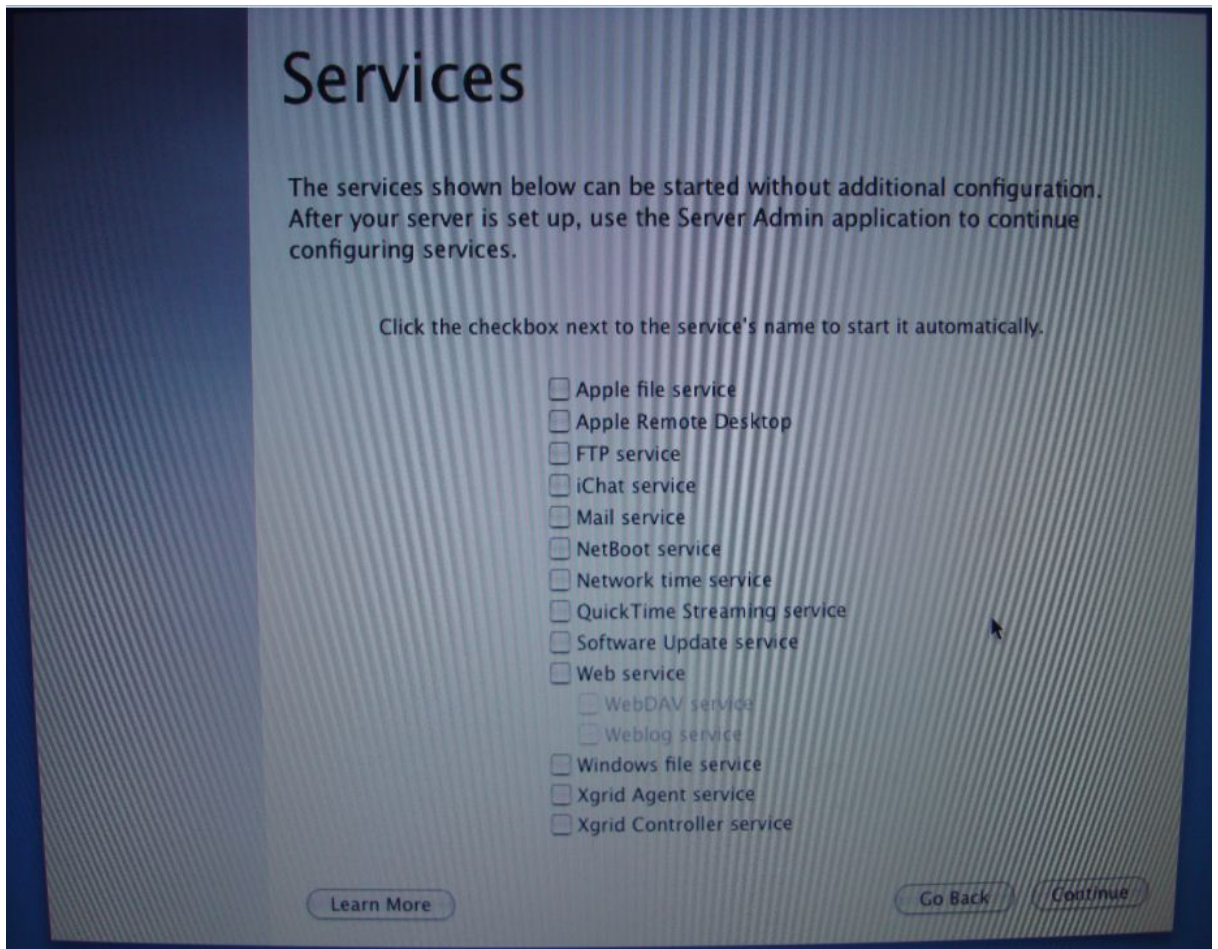


Figura 4.12: Servicios Básicos

4.3.5. Zona Horaria.

Es muy importante configurar este aspecto debido a que el *servidor* debe mantenerse constantemente actualizado. Muchos servicios que ofrece el portal web EIMWeb requieren de la fecha y hora del sistema.

Seleccionamos en el mapa la zona geográfica de ubicación del servidor. En nuestro caso será Bogotá - Colombia.

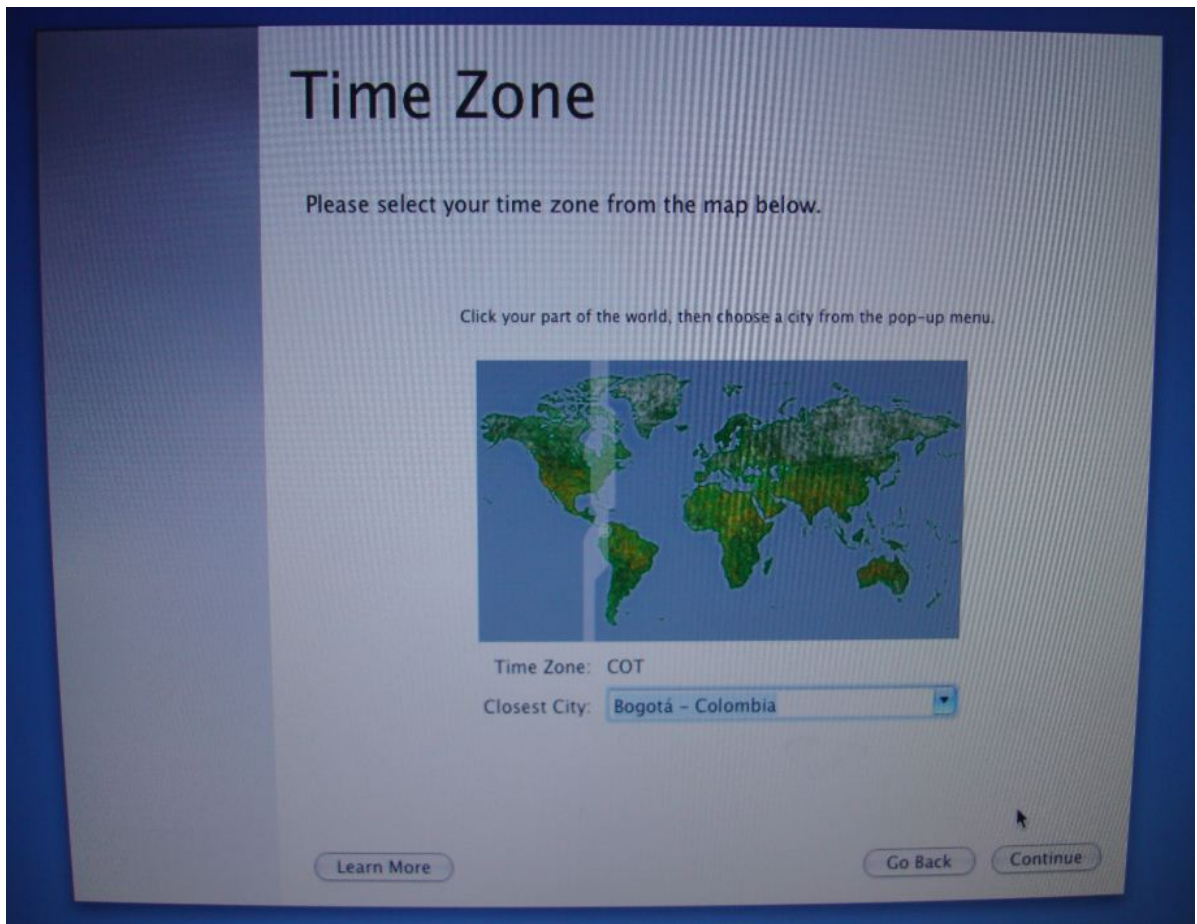


Figura 4.13: Zona Horaria

4.3.6. Finalización.

En este punto se ha configurado inicialmente el sistema operativo del *servidor*, dejándolo listo para empezar a trabajar en el montaje de nuestro servidor web, que alojará el portal web de la Escuela de Ingeniería Mecánica EIMWeb.

Cabe destacar que la configuración mostrada durante este capítulo puede ser modificada en cualquier momento. También es importante saber que el sistema de administración del *servidor* se mostrará por defecto de manera gráfica, sin embargo es posible configurarlos a través de la línea de comandos de consola o terminal UNIX, lo cual expondremos mas adelante.

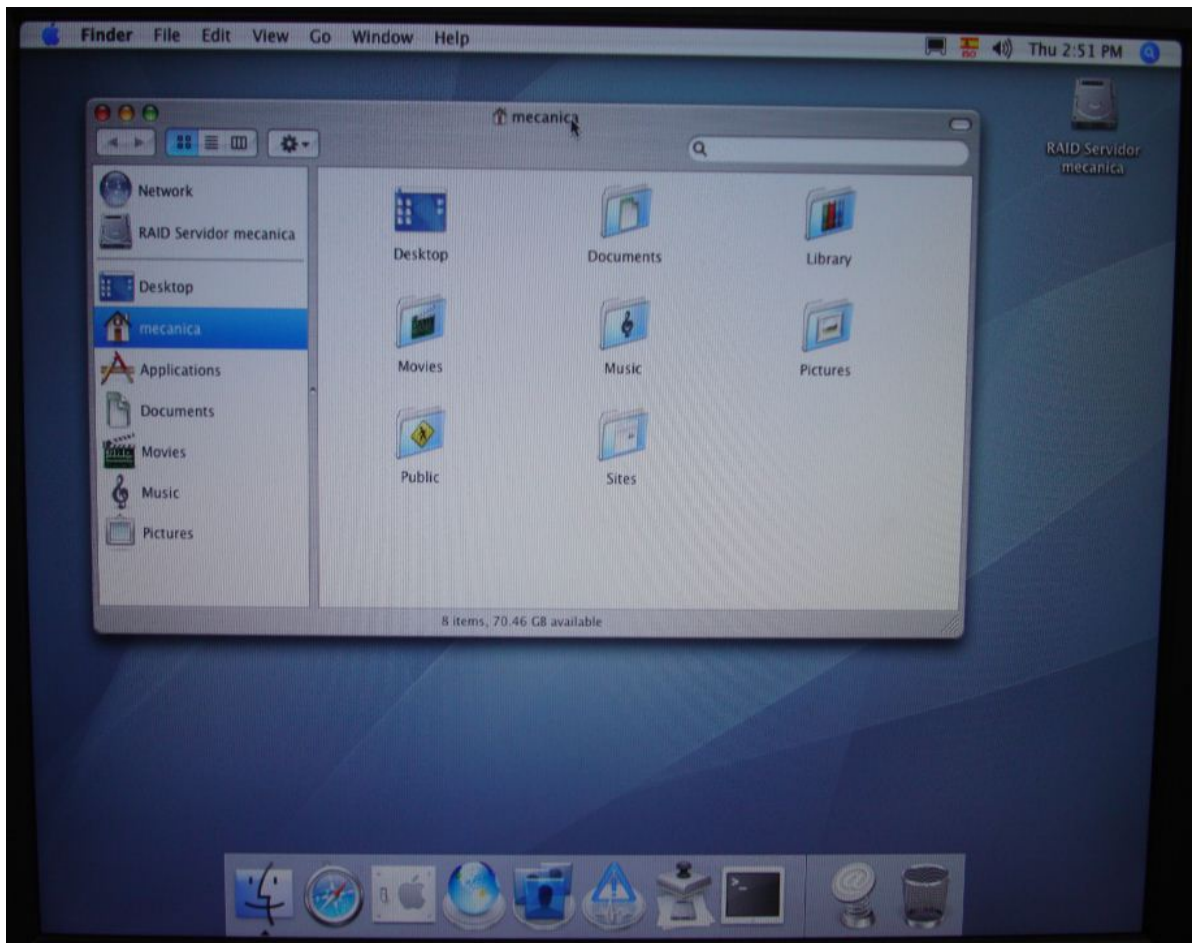


Figura 4.14: Pantalla de Bienvenida Despues de Instalación.

Capítulo 5

CONFIGURACIÓN AVANZADA E IMPLANTACIÓN

Este capítulo mostrará la configuración del servidor web dentro del *servidor* de la Escuela de Ingeniería Mecánica, mostrando detalladamente todos los cambios y procesos realizados para la consecución de este objetivo. Además resaltaremos los puntos más importantes en el proceso de implantación del sitio EIMWeb dentro del servidor web.

5.1. APLICACIONES DEL SISTEMA OPERATIVO TIPO SERVIDOR.

5.1.1. Un Ejemplo: Mac OS X vs Mac OS X Server.

En este momento, hemos configurado nuestro sistema operativo para poder empezar a configurar los servicios que nos puede prestar. Sin embargo es necesario hacer un alto en el camino y entender un poco la diferencia de este sistema tipo servidor¹ con un sistema operativo común.

En general, cualquier computador con un mínimo de requerimientos, que dependen del tipo de servicio que queremos prestar, es apto para alojar algún tipo de programa servidor y ofrecer un servicio, como puede ser el alojamiento de páginas web para que

¹Tipo servidor se refiere a que el sistema operativo es principalmente creado para trabajar con máquinas servidoras.

un cliente pueda observarlas. Este comportamiento se obtiene gracias a que en nuestra máquina hemos instalado ciertos programas que permiten esta funcionalidad.

El sistema operativo Mac OS X Server incorpora los programas mencionados en su instalación por defecto y crea toda una estructura funcional a partir de ellos, lo que nos permite activar y configurar rápidamente los servicios disponibles. Esta es la principal diferencia con un sistema convencional en donde estos paquetes deben ser instalados manualmente. En ambos casos es posible ofrecer la misma funcionalidad, lo que esta en juego en cada una de ellas es el rendimiento y la confiabilidad de los servicios

En este caso hemos mencionado los sistemas operativos fabricados por Apple, pero también se extiende a otros como Linux o Windows.

5.1.2. Algunos Servicios Pre-Instalados en Mac OS X Server Tiger.

Una vez finalizada la instalación del sistema operativo estamos en capacidad de configurar cualquiera de los siguientes servicios:

- AFP(Apple Filling Protocol): El Apple Filing Protocol (AFP) es un protocolo de capa de presentación (según el modelo OSI) que ofrece servicios de archivos para Mac OS X y Mac OS Classic. En Mac OS X, AFP es uno de los varios servicios de apoyo a disposición incluida Bloque de mensajes de servidor (SMB), Sistema de archivos de red (NFS), el Protocolo de transferencia de archivos (FTP), y WebDAV.
- Application Server (Servidor de Aplicaciones): Usualmente se trata de un software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. El sistema operativo cuenta con el servidor de aplicaciones JBOSS 3.2 para trabajar con J2EE en modo stand-alone o en cluster. También en esta sección encontraremos el servidor Tomcat 4.1.3 del cual hablaremos mas adelante.

- DHCP (Protocolo de Configuración Dinámica de IP): Es utilizado para distribuir dinámicamente las direcciones IP a equipos clientes que estén conectados al servidor. Cada vez que un equipo cliente se inicia, solicita una dirección IP al Servidor DHCP que encuentre primero. Este servicio DHCP controla la dirección IP disponible y la envía a la computadora cliente junto con un período de arrendamiento, el período de tiempo que el cliente equipo puede utilizar la dirección IP.
- DNS (Domain Name System): Se conoce también como servidor de nombres, guarda en una base de datos distribuida una lista de nombres de host o hostnames y las asociaciones que estos tengan con una IP determinada, para relacionarlos y resolverlos, a quien los solicite (normalmente otro cliente).
- Firewall: Este servicio protege las aplicaciones de nuestro sistema operativo que interactúen a través de la de red. Vigila los paquetes que entran y salen a través de la red, rechazándolos o permitiéndolos dependiendo de algunas reglas establecidas.
- FTP (File Transfer Protocol): Este servicio permite a nuestro sistema poder manejar transferencia de archivos entre equipos clientes y el sistema servidor, dependiendo de algunos parámetros y de permisos de acceso.
- iChat: A través de este servicio es posible manejar un sistema de mensajería instantánea de clientes, protegiendo la integridad de la información y su confidencialidad.
- Mysql: Por defecto instala la versión 4.1 del software. Es un manejador de base de datos relacionales que nos permitirá almacenar toda la información referente al portal web EIMWeb (explicado con detalle más adelante).
- NAT (Network Address Translation): Este servicio permite conectar múltiples clientes a internet usando solo una IP válida de navegación.
- Software Update: Permite mantener actualizado nuestro sistema operativo, descargando de internet e instalando paquetes de seguridad y de aplicaciones nuevos.
- Open Directory: Con este servicio podemos acceder a la información de usuarios, de red y de directorios usando la red.
- Web: Básicamente a través de este servicio tendremos nuestro servidor web. Se incluye por defecto la versión 1.3 de Apache.

Dependiendo de la configuración y de la funcionalidad que deba brindar nuestro sistema operativo, tendremos múltiples servidores dentro de nuestra máquina servidora. Para nuestra implantación del Portal Web EIMWeb será necesario configurar los servicios de servidor de aplicaciones, web, firewall, software update y mysql².

5.1.3. Herramientas de Administración de Servidor.

Además del sistema operativo, el *servidor* incluye un paquete adicional de programas que nos ayudan en la administración de los diferentes servicios y funcionalidades que incluye el sistema operativo. Debemos destacar, que este sistema Algunos de estos programas son:

- Administración de Servidor (Server Admin): Con este programa podremos manejar de forma gráfica, clara e intuitiva la configuración de los diferentes servicios ofrecidos por sistema operativo. A través de esta aplicación también es posible modificar información como, nombre local del servidor, nombre de máquina y además de esto acceder a información clave como, el número serial de producto, logs del sistema, gráficas de rendimiento y carga, configuración horaria, configuración de red activa, etc. Ver Figura 5.1.
- Monitoreo de Servidor (Server Monitor): Esta aplicación nos muestra información acerca del estado del sistema operativo, discos duros, sistemas de potencia, sistema de procesamiento, ventilación, memoria RAM y configuración de red. Cuando algunos de estos parámetros presenta fallos envía un mensaje de notificación al administrador. Ver Figura 5.2.
- Administración de MYSQL (MYSQL Manager): Nos permite configurar algunas características importantes del manejador de base de datos MYSQL, como son, cambio de clave para el usuario administrador o ROOT, reinicio del servicio e instalación de archivos de configuración básica. Ver Figura 5.3.

²Debido a que es objeto del documento es presentar la configuración e implantación del portal EIMWeb, se han omitido algunos servicios. Para más información remitase a www.apple.com

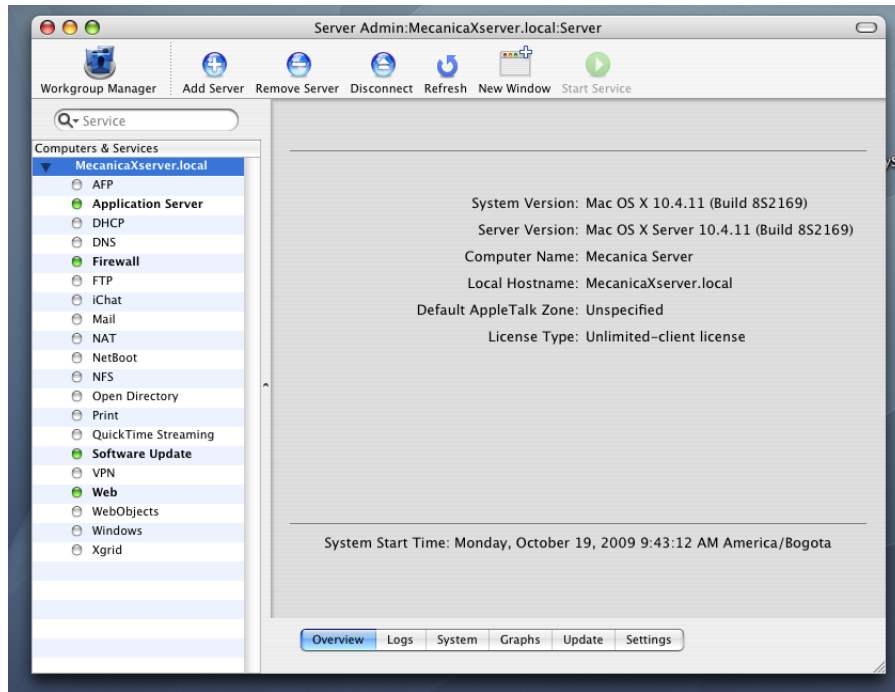


Figura 5.1: Server Admin.

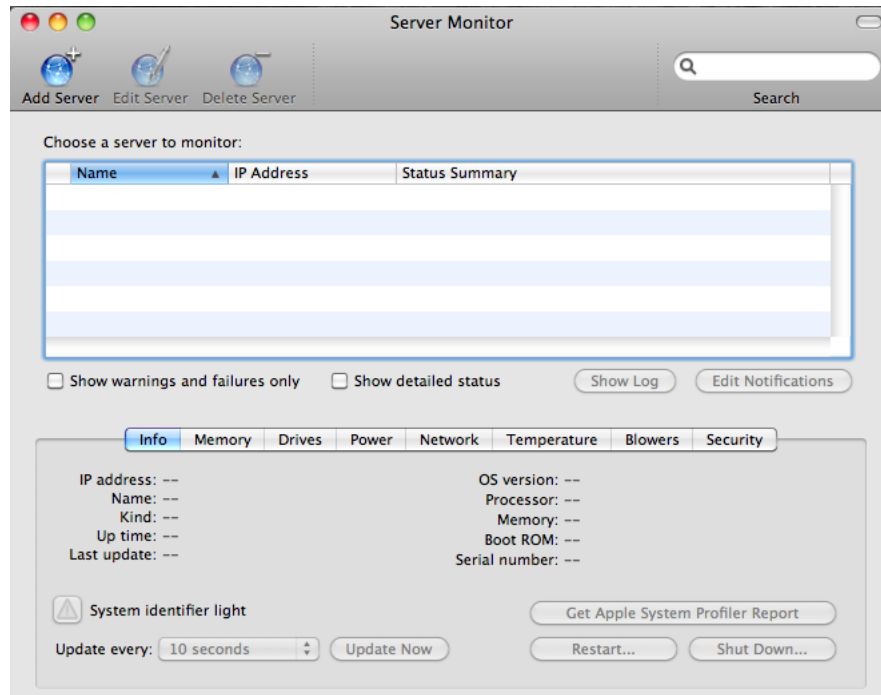


Figura 5.2: Server Monitor

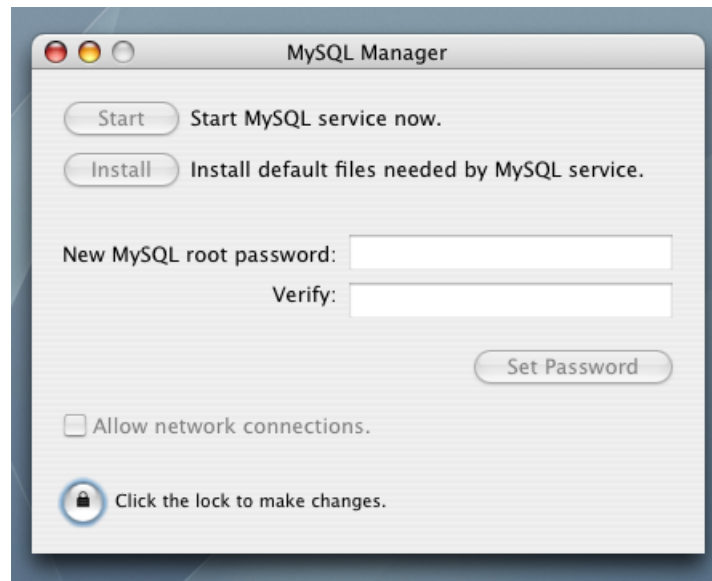


Figura 5.3: Mysql Manager

5.2. CONFIGURACIÓN DEL SERVIDOR WEB.

En esta sección mostraremos la configuración realizada a los servicios necesarios para la configuración del servidor web donde se hará la implantación del portal web de la Escuela de Ingeniería Mecánica EIMWeb.

5.2.1. Java Development Kit (JDK).

Todas las versiones del sistema operativo Mac OS X (clientes y servidoras), traen alguna versión de java instalada por defecto. En nuestro caso, tenemos instalada la version 1.4 del java development kit o JDK.

En el momento que finaliza la instalación del sistema operativo, automáticamente se despliega el servicio de actualización de apple (software update) quien nos permite descargar e instalar actualizaciones que liberen los desarrolladores de apple y asociados para nuestros servidor. Luego de este proceso nuestra versión del JDK es la 1.5.

5.2.2. Configuración Apache - Tomcat.

En resumen, el funcionamiento del portal web EIMWeb se centra en la recepción de peticiones que llegan de los clientes (estudiantes, docentes, administrativos, etc) y el envío de información a través de la red, que responda a dichos requerimientos.

Para poder lograr esto utilizando la tecnología JSP (Java Server Pages) es necesario integrar el servidor Tomcat y el Apache. En nuestro caso el servidor Apache recibirá las peticiones que provienen de la red y las enviará al servidor Tomcat, quién las procesará, enviando una respuesta de nuevo al servidor Apache, que a su vez, mostrará esta información al usuario final.

En esta subsección explicaremos brevemente cada uno de ellos y cómo podemos hacerlos trabajar en equipo.

5.2.2.1. Tomcat.

Para configurar nuestro servidor tomcat entramos a server admin y vamos al submenú “Application Server”. En la parte inferior de la ventana principal encontramos el botón “Settings”, damos click en el, y nos mostrará tres opciones:

- Load Remote Configuration: Nos permite cargar una archivo de configuración del servidor de aplicaciones JBoss³, que se encuentre en un equipo remoto.
- Use Local Configuration: Con esta opción habilitaremos un tipo de configuración local para el servidor de aplicaciones JBoss. Podemos seleccionar por defecto,
- Tomcat only: Configuramos nuestro servicio “Application Server” para que solo trabaje con Tomcat. Recordemos que Tomcat no es un servidor de aplicaciones, pero a través de esta herramienta podemos administrarlo.

³JBoss 3.2.2 es el servidor de aplicaciones incluido en Mac OS X Server Tiger 10.4.

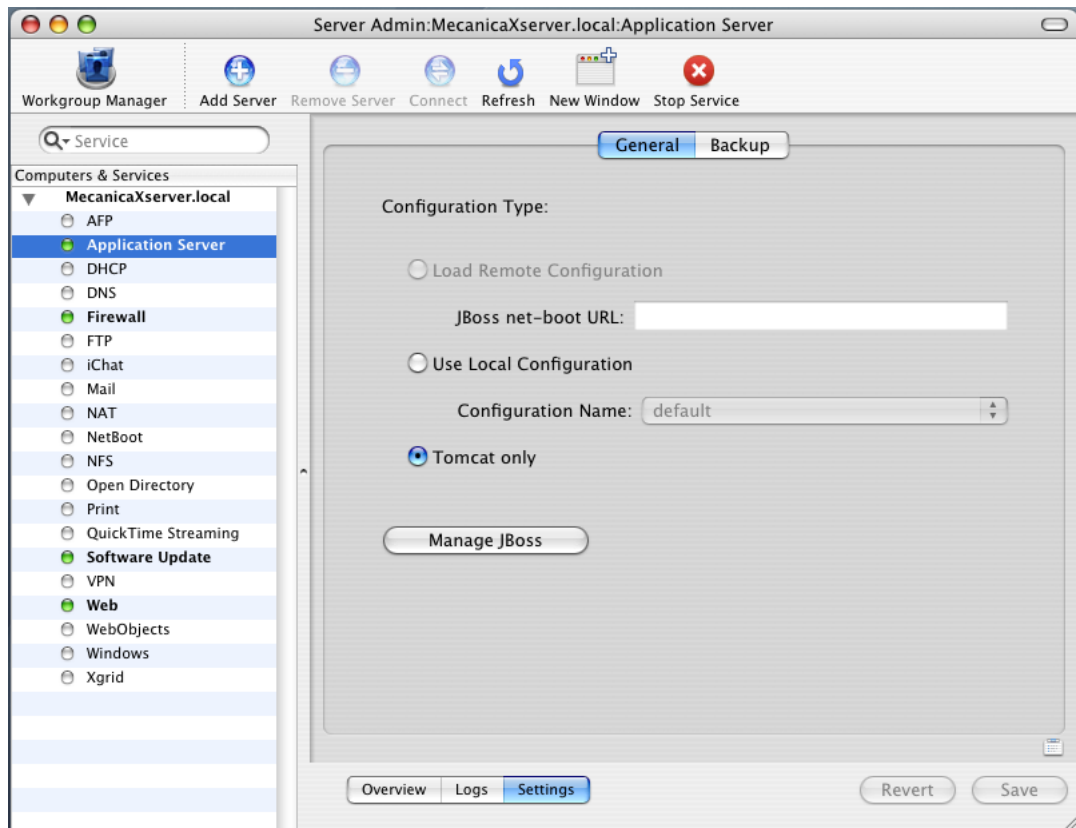


Figura 5.4: Tomcat en Server Admin.

Tomcat queda habilitado para trabajar con el puerto 9006 por defecto.

Estructura y Archivos de Configuración. Tomcat esta alojado en la carpeta /Library/Tomcat de nuestro servidor.

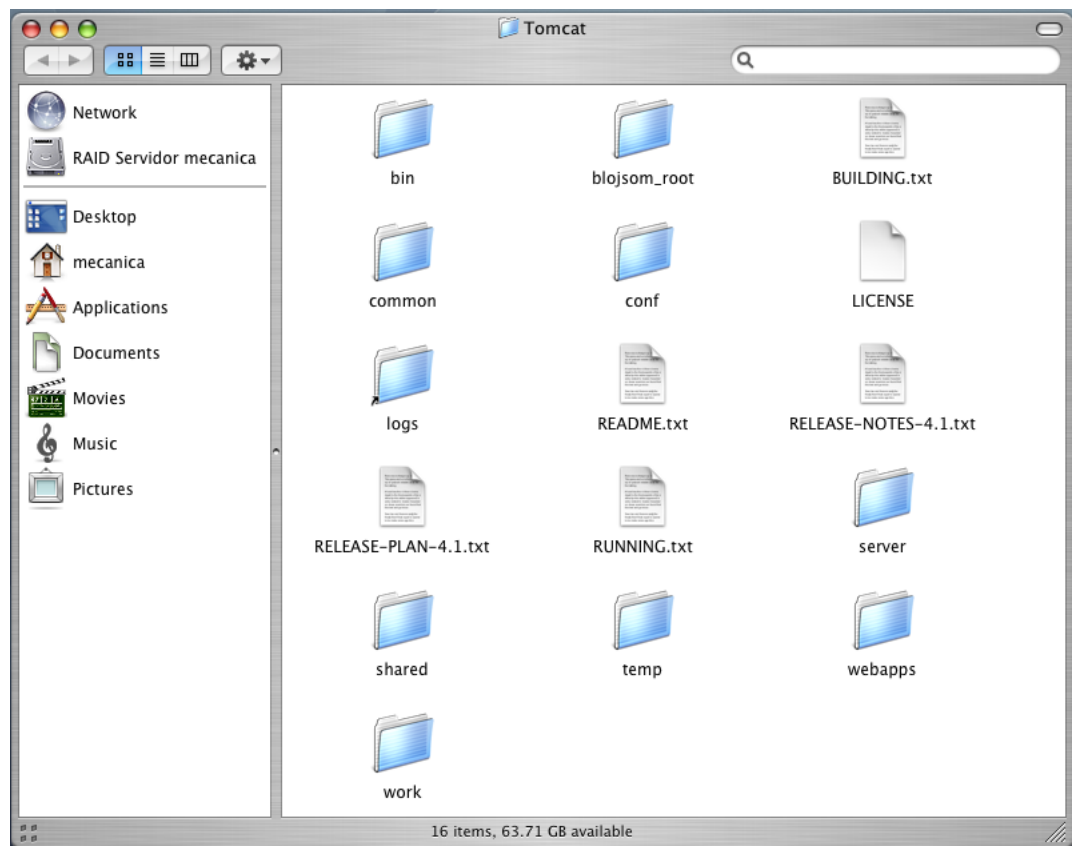


Figura 5.5: Estructura de Tomcat.

Las carpetas mas importantes de esta estructura son:

- webapps: Donde se alojan las aplicaciones que querremos desplegar con Tomcat. Son carpetas con la estructura de la aplicación (jsp's, xml, librerías, etc).

- bin: Allí están los archivos de arranque y parametrización de Tomcat.

Se destacan los archivos `startup.sh` y `shutdown.sh` con los que podemos a través de consola iniciar o parar el servidor tomcat.

- conf: Se alojan los archivos de configuración, entre los que se destacan los archivos `server.xml` y `web.xml`.

`Server.xml` contiene todos los parámetros de configuración del tomcat, pudiéramos decir que es el "corazón" de tomcat (puertos, seguridad, tipo de aplicaciones a desplegar, tipos de archivos, etc.).

Web.xml proporciona información sobre configuración y despliegue de los componentes Web que componen una aplicación Web. Los ejemplos de los componentes Web son los parámetros de servlets, las definiciones de servlets y JSP (JavaServer Pages) y las correlaciones de URL (Uniform Resource Locators).

- work: Aquí se guardan archivos temporales de las aplicaciones que han sido desplegadas en el servidor tomcat.

Inicio y Parada del Servidor Tomcat

1. A través del ambiente gráfico:

Vamos a la herramienta “admin server”, luego a “Application Server” y en la parte superior de la ventana principal hacemos click en “Start Service”. Si queremos pararlo pulsamos en “Stop Service”

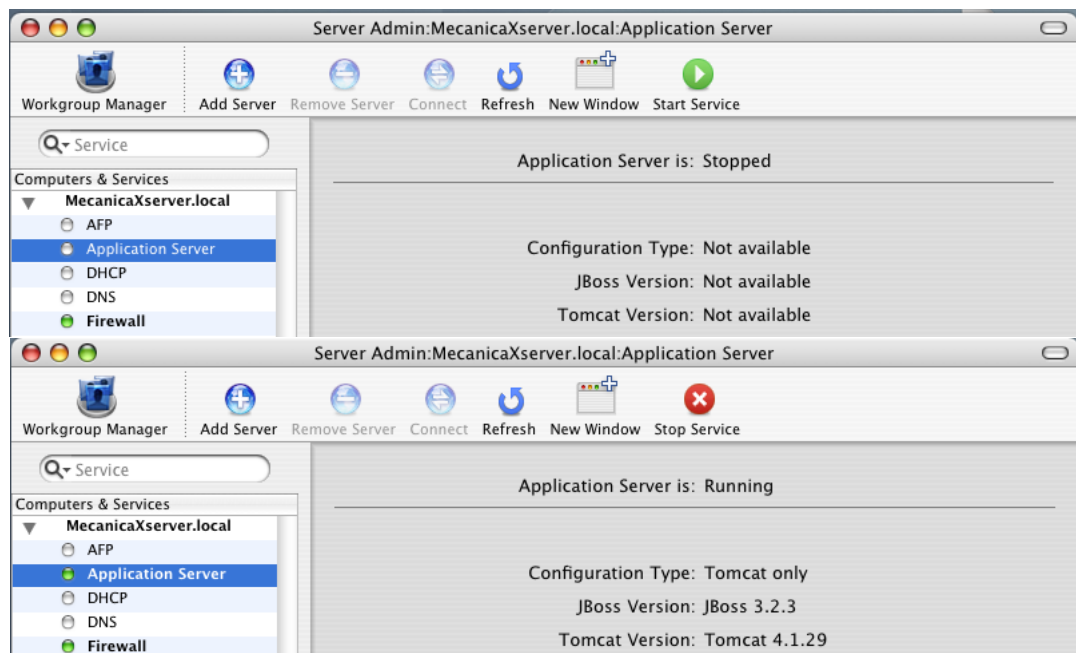


Figura 5.6: Arranque y Parada de Tomcat Gráficamente

2. Por consola:

Vamos a la carpeta conf de tomcat. Ejecutamos el archivo startup.sh para iniciar el servicio y el archivo shutdown.sh para detenerlo.

Escribimos en consola: “`sh /Library/Tomcat/bin/startup.sh`” para iniciar ó “`sh /Library/Tomcat/bin/shutdown.sh`” para detenerlo.

En otros sistemas operativos en donde se instala el tomcat manualmente, es necesario configurar lo que se conoce como variables de entorno, las cuales permiten agregar al sistema información correspondiente a la ubicación de archivos de configuración de Tomcat y de Java, necesarios para que pueda arrancar el servicio Tomcat. Algunas de estas variables son: `JAVA_HOME` que es la ruta donde encontramos los archivos de arranque del JDK y `CATALINA_HOME` que es la ruta de instalación de Tomcat. En nuestro caso por ejemplo, `CATALINA_HOME` sería `/Library/Tomcat`.

Para este sistema servidor no será necesario definir estos parámetros ya que el sistema por defecto a configurado estas variables actualizandolas dinámicamente cuando se presentan cambios de versiones.

5.2.2.2. Apache.

El servidor web Apache es un software que esta estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- Módulos Base: Módulo con las funciones básicas del Apache
- Módulos Multiproceso: son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiproceso para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. El resto de funcionalidades del servidor se consiguen por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Archivos de Configuración. Los archivos de configuración de apache se encuentran en `/etc/httpd/`. Aquí encontramos los archivos de configuración tanto del servidor web, como de otros servicios tales como el servicio de correos y la autenticación SSL. Los más importantes son:

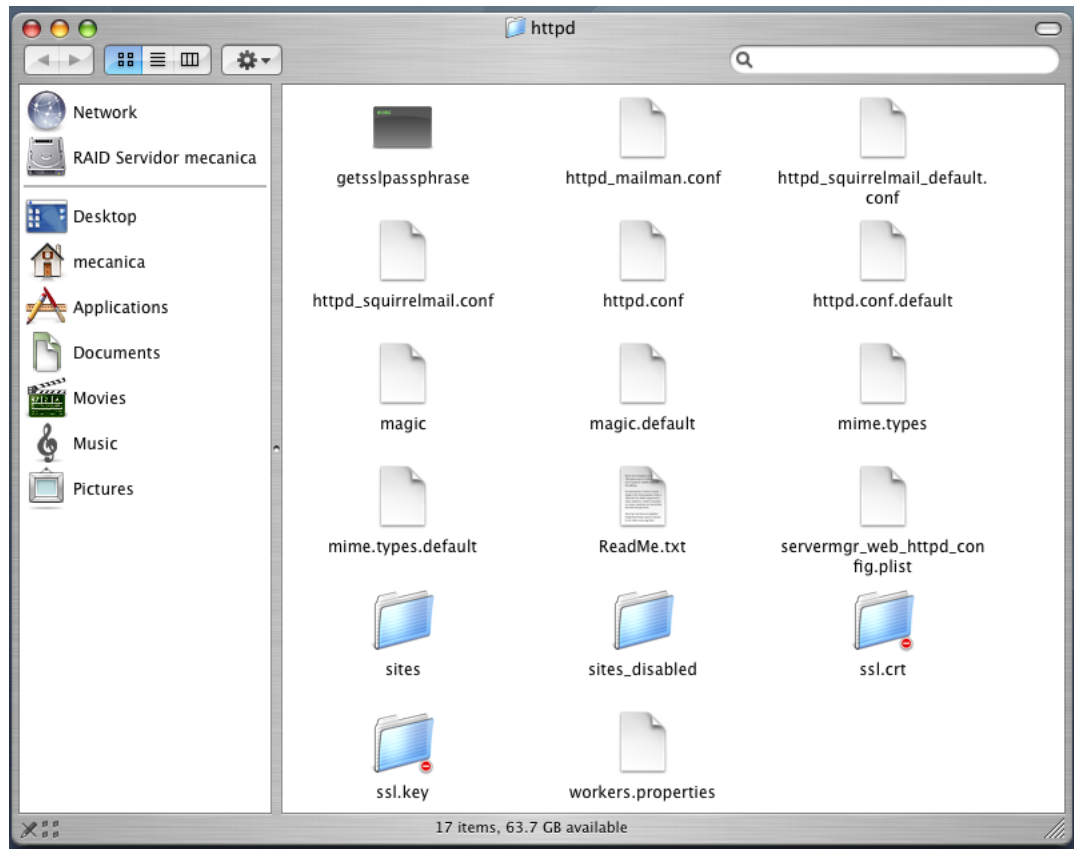


Figura 5.7: Archivos de Configuración de Apache

- `Httpd.conf`: Es el archivo principal de configuración del Apache. En primer lugar hay que destacar que el archivo está dividido en tres secciones, que son:
 1. Parámetros globales
 2. Directivas de Funcionamiento
 3. Host Virtuales

En el archivo se encuentran todos los parámetros de funcionamiento del Apache. Algunos parámetros son generales para la instalación y funcionamiento del Apache. Muchos otros de los parámetros se pueden configurar independientes para

un conjunto de directorios y/o archivos . En estos casos los parámetros se encuentran ubicados dentro de secciones donde se indica el ámbito de aplicación del parámetro.

- **Workers.properties:** A través de este archivo podemos configurar lo que se conoce como un worker, en nuestro caso un Tomcat worker, el cual es una instancia de Tomcat que está a la espera ejecutar Servlets o cualquier otro contenido que proviene del servidor web. Por ejemplo, podemos tener un servidor web, como Apache, que reenvie solicitudes (servlets) a un proceso de Tomcat (worker) corriendo detrás de él.

Configuración de Apache. Para poder configurar el servidor Apache iniciamos la aplicación “server admin” y seleccionamos en la lista de servicios seleccionamos “Web”. En esta ventana vemos cuatro (4) opciones en la parte inferior, seleccionamos “Settings”. Aquí la ventana principal nos muestra seis (6) pestañas. Para configurar nuestro servidor web solo necesitamos entrar en “Sites” y “Modules”.

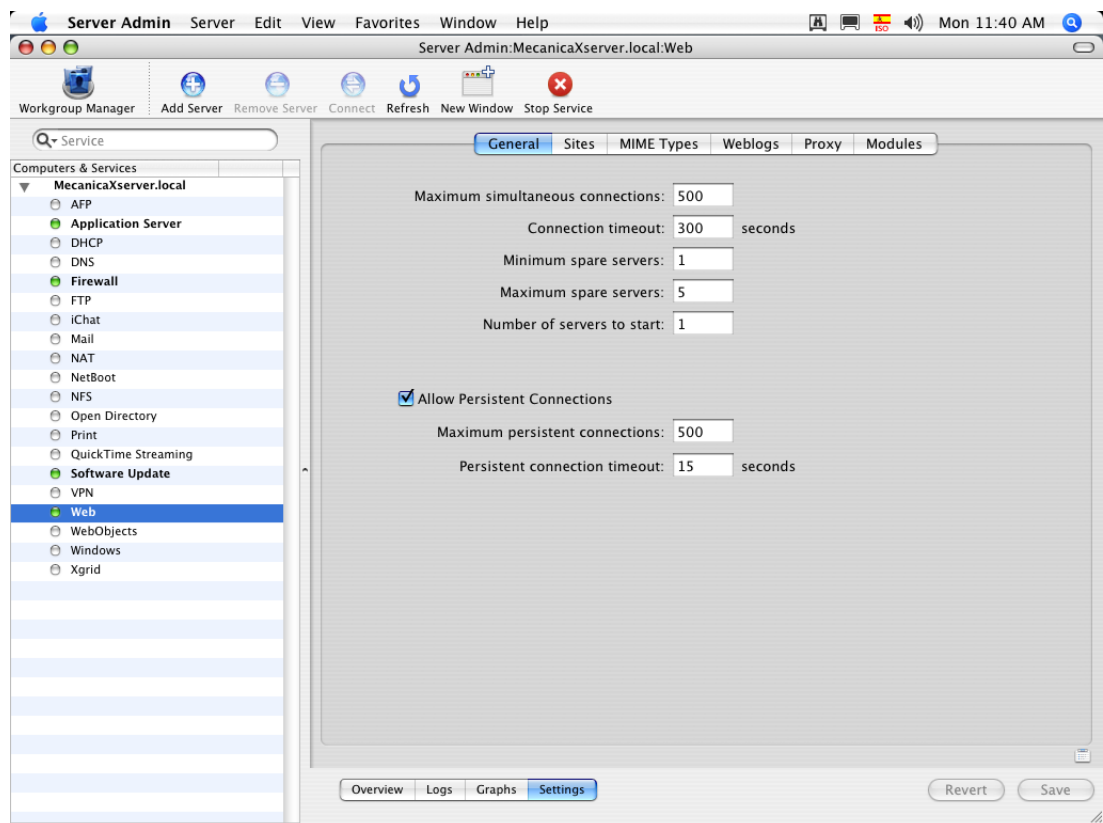


Figura 5.8: Servicio Web en Server Admin

Prácticamente en esta interfaz se muestra todo el archivo de configuración httpd.conf de una manera mas clara y de fácil administración, siendo posible rápidamente abarcar y modificar casi todos los parámetros del archivo.

En la ventana “Sites” observamos los sitios que estan alojados en el servidor Apache. En este caso observamos que ya esta el sitio `mecanica.uis.edu.co`.

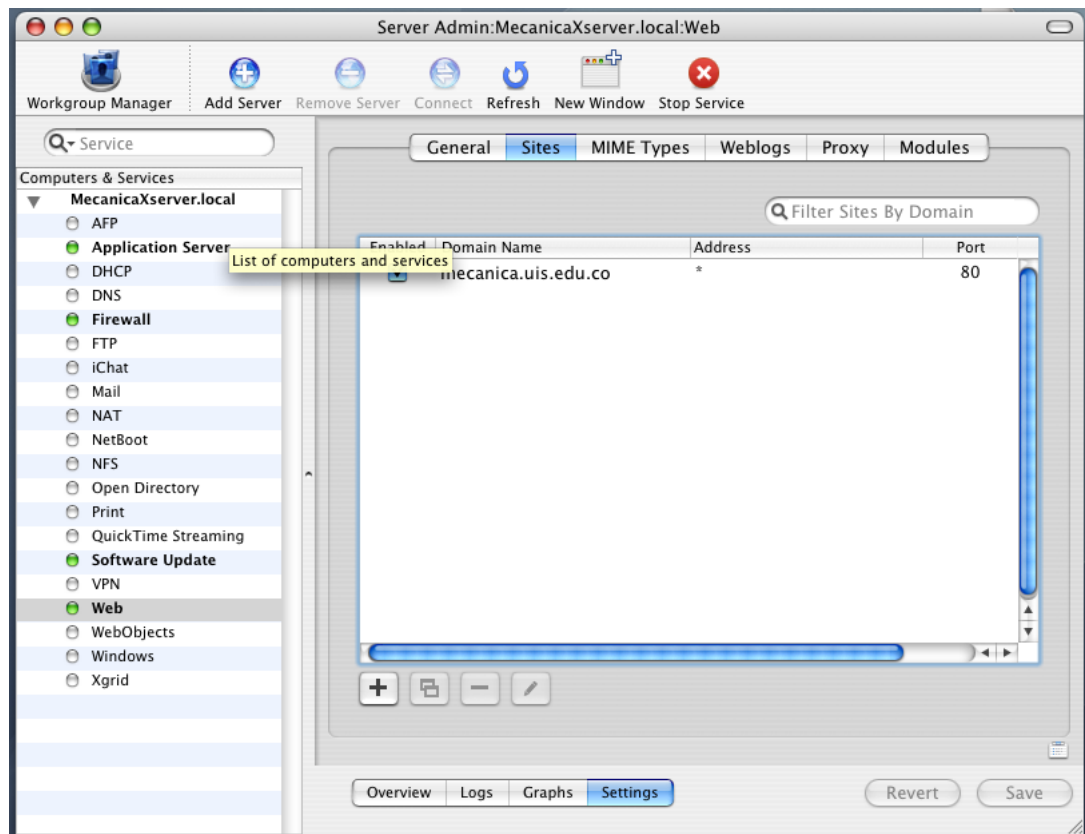


Figura 5.9: Ventana Sites

El proceso para agregar esta aplicación se muestra a continuación:

- Hacemos click en el botón + de la parte inferior lo cual mostrará unas pestañas con los parámetros que debemos editar. La primera es la “General”, donde configuramos el dominio o nombre del sitio (domain name), la ip que tendrá, el puerto de escucha para esa aplicación, el directorio donde se encontrarán los archivos de la aplicación (llamada Web Folder)⁴, archivos por defecto que deberá

⁴En el archivo http.conf es el mismo parámetro DocumentRoot.

buscar el Apache cuando se llame a la aplicación, archivo de error y correo de administración.

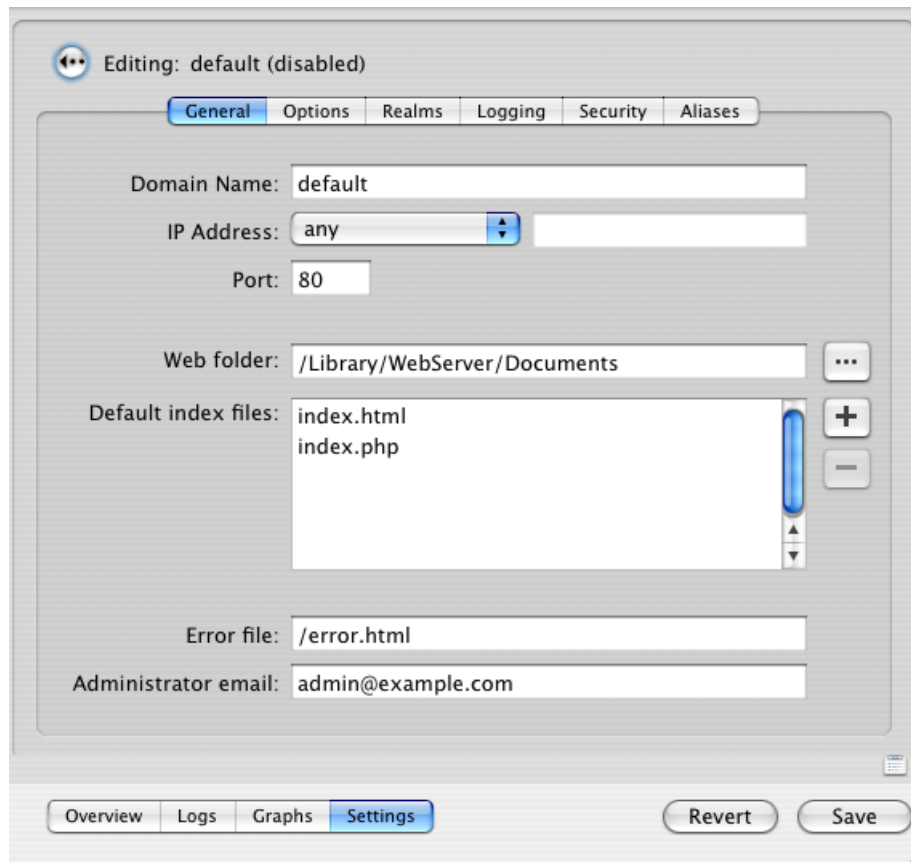


Figura 5.10: Ficha General.

- Vamos a la pestaña “Options” y allí deseleccionamos la casilla “Performance Cache”, ya que nuestro servidor Apache va funcionar en conjunto con el servidor Tomcat a través del modulo JK que veremos mas adelante.

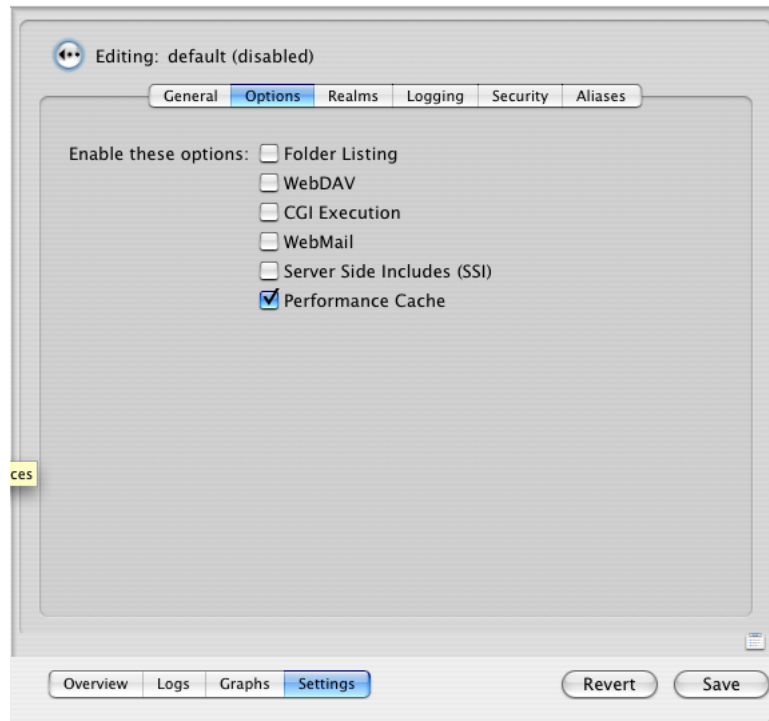


Figura 5.11: Pestaña Options

Inicio y Parada del Servidor Apache. De forma similar a Tomcat, en la ventana principal pulsamos “Start Service” para iniciar el servicio o “Stop Service” para detenerlos. Si queremos hacerlo por consola debemos escribir:

“serveradmin start web” para iniciar el servicio ó *“serveradmin stop web”* para pararlo.

5.2.2.3. Configuración del conector “mod_jk”.

Mod_jk es un conector que permite al contenedor de Java Server Pages (JSP) Tomcat interactuar con servidores web como Apache, Netscape, iPlanet, SunOne e incluso IIS usando el protocolo AJP. La principal funcionalidad de este módulo es permitir a servidores de aplicaciones (JBoss en nuestro caso) o al servidor Tomcat enlazarse con un servidor web. Este servidor web, típicamente el servidor HTTP Apache, introduce una mayor gestión en las conexiones de los clientes y mayor la seguridad en las transacciones del sistema. Así mismo se puede enlazar varias instancias al servidor web permitiendo así una mayor tolerancia a errores y aligerar la carga en los servidores Java.

Normalmente cuando Apache es instalado manualmente el módulo que permite utilizar este conector no es incluido, por tanto es necesario seguir algunos pasos:

1. Descargar el archivo fuente (dependiendo de cual sistema operativo tengamos) del módulo `mod_jk`, que se encuentra disponible en la seccion de módulos de la página web de Apache.
2. Compilar este código con `apxs` (apache extension tools). Esta herramienta permite compilar, construir e instalar extensiones de módulos para el Apache, a partir del código fuente descargado en el paso anterior. Por defecto utilizamos el comando `./configure --with-apxs=/usr/sbin/apxs`. Al finalizar debemos tener creado el archivo `mod_jk.so` en la carpeta `httpd/modules` de nuestro sistema operativo.
3. Creamos el archivo `workers.properties` el cual contiene información referente al worker Tomcat quien resolverá las peticiones de Apache en tomcat.
4. Modificación del archivo `server.xml` de tomcat para habilitar el protocolo de conexión `ajp1.3`.
5. Modificamos el archivo `httpd.conf` para incluir el modulo `mod_jk`. Reiniciamos el servidor.

El servidor Apache que esta instalado en nuestro servidor ya incluye el modulo `mod_jk` (es necesario activarlo), por tanto solo será necesario realizar los pasos 3 y 4 anteriormente descritos, con la diferencia que el archivo `workers.properties` ya esta creado.

Configuración del archivo `Workers.properties` A continuación se muestra el contenido del archivo:

```
worker.list=tomcat1 // nombre del worker.  
workers.tomcat_home=/Library/Tomcat // dirección de instalación de tomcat  
workers.java_home=/Library/Java/Home // dirección del JDK  
worker.tomcat1.type=ajp13 //Tipo de worker ajp1.3  
worker.tomcat1.host=MecanicaXserver // Nombre de Host donde trabajará el worker.  
worker.tomcat1.port=9007 //Puerto por el cual entrarán y enviarán las peticiones y  
respuestas respectivamente.
```

Esta es una configuración básica que nos permite crear nuestro worker y darle algunas características.

Configuración del archivo `server.xml` Descomentar la información:

```
<!-- Define an AJP 1.3 Connector on port 9007 -->  
<Connector port="9007" enableLookups="false" redirectPort="8443" protocol="AJP/1.3"  
>
```

Configuración del archivo `httpd.conf` Agregar la siguiente información al archivo:

```
<IfModule mod_jk.c>  
  
JKWorkersFile /etc/httpd/workers.properties // ubicación del archivo workers.properties  
JKLogFile /var/log/httpd/mod_jk.log //ubicación del log  
JKLogLevel error  
  
JKMount /*.jsp tomcat1 //el worker tomcat1 resuelve peticiones de todos los archivos  
jsp.  
  
JKMount /servlet/* tomcat1 //el worker tomcat1 resuelve peticiones que se procesen  
dentro de la carpeta /servlet  
  
JKMount /examples/* tomcat1 //el worker tomcat1 resuelve peticiones que se procesen  
dentro de la carpeta /examples  
  
JKMount /eisi tomcat1  
JKMount /eisi/* tomcat1  
  
</IfModule>
```

Activación del módulo `mod_jk` Para activar este modulo y otros que se pudieran necesitar, lo hacemos de la siguiente manera:

- Abrimos la aplicación “server admin”.
- Seleccionamos el servicio Web.
- Vamos a la pestaña Settings. Luego a la opción “Modules”.

- Buscamos el modulo `mod_jk.so` y lo activamos.

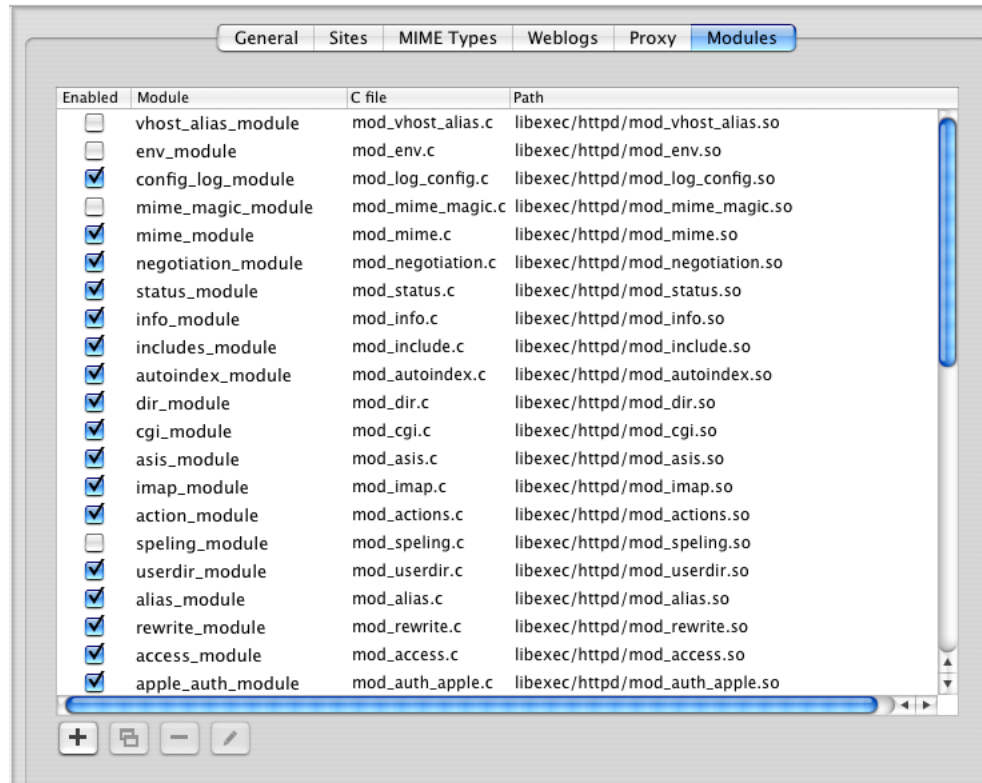


Figura 5.12: Activación Modulo `mod_jk`

Este proceso equivale a ir al archivo `httpd.conf` y descomentar la línea:

```
LoadModule jk_module libexec/httpd/mod_jk.so.
```

En estos momentos todas las peticiones configuradas en el archivo `httpd.conf` que lleguen al puerto 80 (Apache) serán contestadas por el puerto 9006 (Tomcat) gracias al conector `mod_jk`. Dicho de otra manera: al escribir en nuestro navegador web “`http://localhost/ejemplo`” saldría la página de inicio de la aplicación “eisi” que funciona a través del puerto 9006. Si no estuviera activado el modulo `mod_jk`, seguramente nos saldría la página de inicio de Apache y para poder acceder a la aplicación “ejemplo” sería necesario escribir “`http://localhost:9006/ejemplo`”.⁵

⁵En este caso la aplicación ejemplo debe estar balanceada con un worker definido en `workers.properties` y adicionado en `httpd.conf`.

5.3. CONFIGURACIÓN BÁSICA DE MYSQL

La configuración Básica de MySQL, es crear la contraseña para el usuario ROOT y crear un usuario externo para acceso a las bases de datos de las aplicaciones.

5.3.1. Mysql Manager.

A través de esta herramienta de administración podemos realizar cuatro (4) acciones:



Figura 5.13: Mysql Manager

1. Arrancar y detener el servicio de mysql en nuestro sistema.
2. Instalar los archivos de arranque de mysql, entre ellos estan, my.cnf y mysql.sock.
3. Cambiar la contraseña del usuario administrador ó Root. No tiene por defecto.
4. Permitir administración a través de la red.

Lógicamente es posible configurar mysql a través de la linea de comandos. Para crear la contraseña del usuario root escribimos en consola:

```
mysql
```

```
mysql> use mysql;
```

```
mysql> update user set Password=password('xxxxxx') where User='root';
```

```
mysql> exit
```

5.3.2. Creación de Usuarios

Ahora se va a crear un usuario que va a tener acceso solo a las bases de datos que necesita y realizar las operaciones sql que necesita, al contrario del usuario root que tiene acceso a todas las bases de datos, esta es una consideración de seguridad para las bases de datos:

Entramos por consola a mysql como el usuario Root:

```
“mysql -u root -p xxxxxxxxxxx”
```

En el prompt de MySQL escribimos:

```
“mysql> GRANT ALL PRIVILEGES ON BD.* TO empleado@"%" IDENTIFIED BY  
'xxxxxx'
```

```
WITH GRANT OPTION; “
```

Se está creando un usuario (empleado) con contraseña xxxxxx con acceso a la base de datos BD y todas sus tablas (BD.*) desde cualquier parte (empleado@"%") con todas las opciones de modificación a la base de datos BD. En la aplicación que se monta al servidor se usa la base de datos BD, las conexiones a BD se realizan con el usuario “empleado” y la contraseña “xxxxxx”.

Gracias a la configuración realizada hasta ahora tenemos listo nuestro servidor para empezar la implantación. Sin embargo nos hace falta un punto muy importante: Firewall.

5.4. CONFIGURACIÓN DEL FIREWALL

Un firewall es un dispositivo que funciona como cortafuegos entre redes, permitiendo o denegando las transmisiones de una red a la otra. Un uso típico es situarlo entre una

red local y la red Internet, como dispositivo de seguridad para evitar que los intrusos puedan acceder a información confidencial.

Un firewall es simplemente un filtro que controla todas las comunicaciones que pasan de una red a la otra y en función de lo que sean permite o deniega su paso. Para permitir o denegar una comunicación el firewall examina el tipo de servicio al que corresponde, como pueden ser el Web, el correo o el IRC. Dependiendo del servicio el firewall decide si lo permite o no. Además, el firewall examina si la comunicación es entrante o saliente y dependiendo de su dirección puede permitirla o no.

De este modo un firewall puede permitir desde una red local hacia Internet servicios de Web, correo y ftp, pero no a IRC que puede ser innecesario para el trabajo que se este realizando. También se pueden configurar los accesos que se hagan desde Internet hacia la red local y se puede denegar a todos o permitir algunos servicios como el de la Web, (si es que se posee un servidor Web y se quiere que sea accesible desde Internet).

Dependiendo del firewall que se tenga también se puede permitir algunos accesos a la red local desde Internet si el usuario se ha autenticado como usuario de la red local. Un firewall puede ser un dispositivo software o hardware, es decir, un dispositivo que se conecta entre la red y el cable de la conexión a Internet, o bien un programa que se instala en la máquina que tiene el módem que conecta con Internet. Incluso se puede encontrar ordenadores computadores muy potentes y con software específico que lo único que hacen es monitorizar las comunicaciones entre redes.

5.4.1. Administración del Firewall

Para poder administrar el firewall en Mac OS X Server Tiger, utilizamos nuevamente la herramienta “Server Admin”. Escogemos en la lista de servicios, “Firewall”.

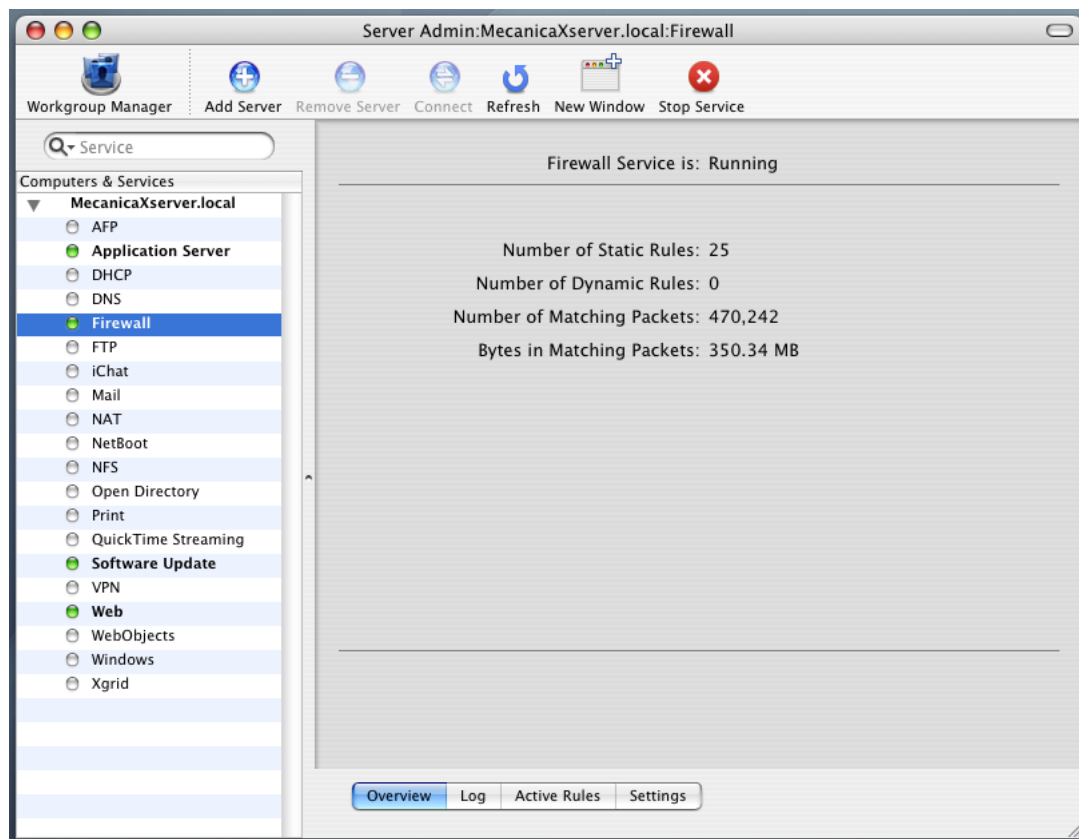


Figura 5.14: Firewall en Server Admin

Vamos a la pestaña “Settings”. En la ventana principal encontramos cuatro (4) opciones, de ellas nos interesa la opción “Services”. En este menú encontramos todos los puertos que están asignados para las diferentes aplicaciones y servicios instalados en el servidor, permitiéndonos activar o desactivar los puertos según nuestro criterio de seguridad. También nos da la opción de administrar los puertos dependiendo de la ip o grupos de ip’s que puedan conectarse al servidor. Por defecto estas reglas se aplican para todas las ip entrantes.

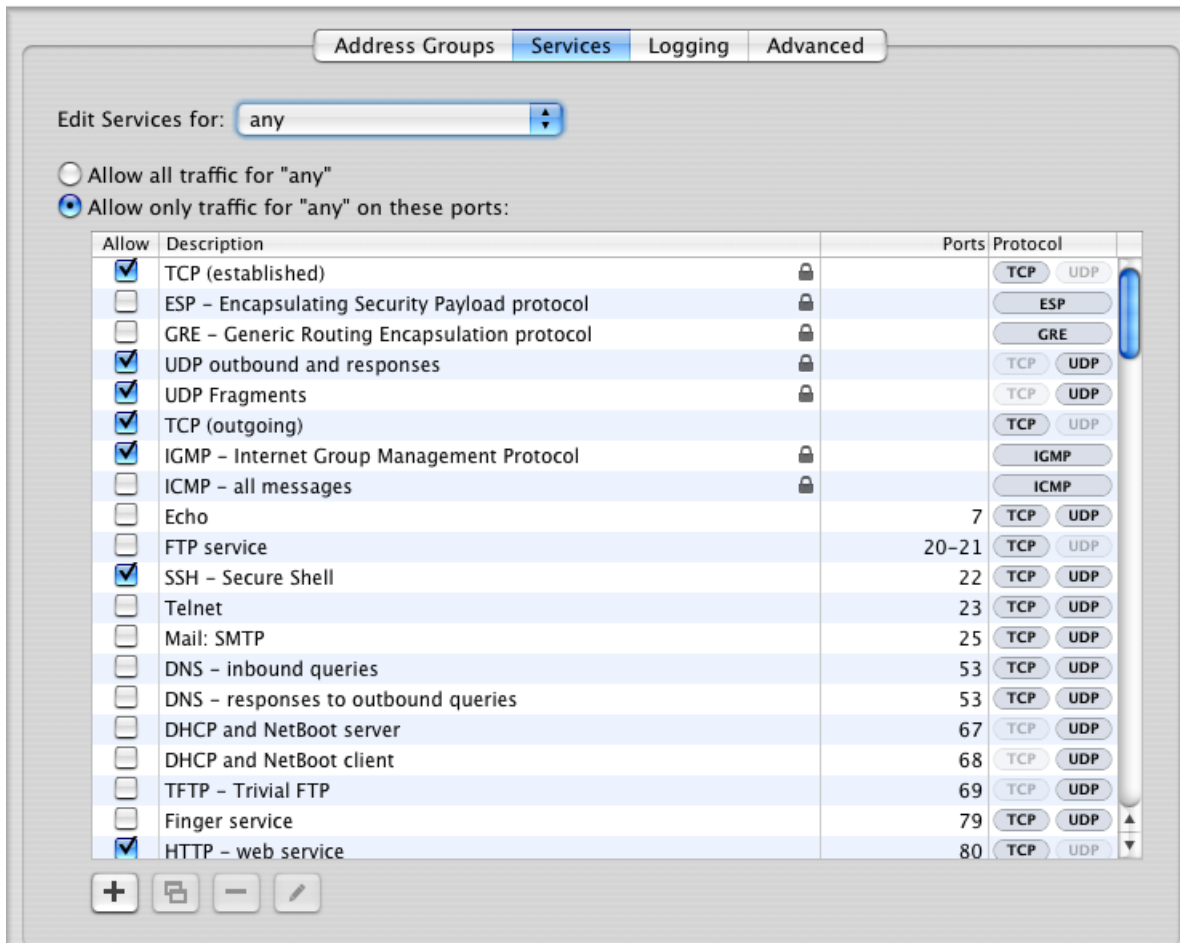


Figura 5.15: Firewall Services.

5.5. PROCESO DE IMPLANTACIÓN DEL PORTAL EIMWEB

5.5.1. Compatibilidad con el Servidor

El proceso de implantación del sitio completamente parametrizado, se reduce a tener en cuenta solo algunos ítems de compatibilidad en el servidor es decir se necesita saber como se encuentra el servidor, es necesario adaptar el sitio al servidor, mas no el servidor al sitio, para implantar se deben tener en cuenta algunos aspectos importantes del servidor:

- Versión de Máquina virtual de java Instalada, este aspecto es importante ya que

si en el sitio plantilla, el servidor en el cual esta alojado trabaja con una versión de java 1.5 y el servidor destino al cual se va a alojar el sitio, trabaja con una máquina virtual java 1.6, es necesario recompilar todos los Beans con java 1.6 para que funcione, de otra manera no se obtendrá ningún resultado.

- Versión de Servidor Tomcat, el servidor tomcat es el que va a llevar toda la carga del sitio, por tanto es importante saber con que versión de tomcat se cuenta, de manera que sea compatible con la máquina virtual instalada, es el único requerimiento para tomcat, ya que la forma en la que esta desarrollado el sitio hace que los servlets sean soportados por cualquier versión de Tomcat, sea la 4, 5, 5.5 o 6.
- Versión de Conector JK, la versión del mod-jk debe ser la 1.2.
- Versión de MySQL, aunque las pruebas que se han realizado, el sitio ha funcionado muy bien con las versiones 4, 5 y 5.1 de MySQL.
- Nombre del Servidor.
- Dirección de IP del Servidor.
- Ubicación completa de la carpeta webapps de Tomcat: Se necesita esta información ya que, esta carpeta contiene los paquetes que se quieren correr con tomcat, siendo este el caso del Portal EISIWeb.

5.5.2. Alojamiento del Portal en el Nuevo Servidor

Para alojar el portal en el servidor destino es necesario seguir los siguientes pasos:

- Teniendo ya la información necesaria del servidor en el cual se va a alojar el portal, es decir el servidor destino, se adquiere una copia del sitio EISIWeb actualizada, esta copia se obtiene del servidor en el cual esta alojado, es decir el servidor origen, el servidor cormorán, en el servidor la aplicación se encuentra con un nombre de carpeta “xxxx”, esta carpeta se ubica en la carpeta de aplicaciones de tomcat (webapps), para que sea reconocida por el servidor.
- El siguiente paso es editar el archivo de Parametrización, Parametrizacion.java, dentro del cual se va a agregar la información necesaria, para esto es necesaria la información del servidor previamente solicitada, se varían los parámetros, anteriormente se hizo una descripción completa del archivo, el proceso solo es reemplazar los datos anteriores con

los datos obtenidos del servidor, tanto para los parámetros externos, como para los parámetros internos.

5.5.3. Cargando las Bases de Datos en Mysql

Las Bases de datos son parte fundamental para el funcionamiento del sitio, este consta de dos Bases de Datos, la base de datos que obtiene la información de los Usuarios desde la División de servicios de Información y la Base de Datos del Sitio, la cual contiene toda la información necesaria para que el sitio funcione.

Para la Parametrización, se crearon dos Bases de Datos plantilla, las cuales contienen los mínimos requerimientos para que el sitio funcione, hay que tener en cuenta que no se van a dar los nombres de las bases de datos ni el contenido de las tablas, se hará una descripción del objetivo de cada una y luego el proceso en MySQL para cargar las bases de datos del sitio:

- Base de Datos de División: Las tablas que se cargan con la información obtenida de la división de servicios de información están completamente vacías, ya que se van a agregar nuevos datos, estos datos son los referentes a la escuela para la cual se va a implantar el sistema. Hay que tener en cuenta que el sistema automáticamente borra cualquier contenido de las tablas en el momento en que se va a realizar una actualización de esta base de datos.
- Base de Datos del Sitio: Esta base de datos contiene las tablas necesarias para que el sitio cargue toda la información que ha ido desarrollando, información de usuarios, publicaciones, servicios, grupos, etc. Se tomó una base de datos actualizada del sitio, y se hizo un análisis de los requerimientos básicos que tendría que tener para que el sitio funcione:
 - No hay Usuarios Registrados, solamente deben tener el usuario Invitado y un Usuario Administrativo con acceso total.
 - No hay publicaciones en cartelera, noticias, comunicados, ni destacados.
 - No hay encuestas propuestas.
 - No hay grupos

- No hay Lista de Docentes, esto implica que no hayan horarios ni actividades extraclase.
- No hay grupos.
- No hay materias.
- No hay Eventos propuestos.
- No hay Archivos, ni Backups de las bases de Datos.
- No hay Foros.

Lo único que se conserva son los servicios ya que estos no cambian y se van a ofrecer los mismos servicios en todas las escuelas, se mantiene el esquema de entidad-relación de la base de datos.

El proceso para cargar las bases de datos en MySQL, para las dos bases de datos es el mismo y es el siguiente:

- En una consola entramos a mysql con el usuario que se ha creado para este propósito. No es necesario el usuario root.

```
“mysql -u usuario -p Password: clave”
```

- El proceso se realiza mediante el comando source de MySQL, se ubica la posición del archivo backup que se realizó de la base de datos del sitio, este archivo tiene la extensión .sql y se carga en MySQL de la siguiente manera:

```
mysql> source /ubicacion/de/la/base/de/datos/nombrebasededatos.sql
```

- Se realiza el proceso de creado de tablas y todo lo necesario para acceder a las tablas. Después de realizado el proceso, se puede seleccionar la base de datos creada y se pueden mostrar las tablas:

```
“mysql> use basededatos;”
```

```
“mysql> show tables;”
```

- A continuación se muestra el contenido de las tablas de la base de datos del sitio. El mismo proceso se realiza para cargar la base de datos de división.

5.5.4. Librerías Necesarias

Para que el sitio funcione adecuadamente son necesarias unas librerías .JAR que se ubican en algunas carpetas de java, tomcat y la aplicación, estas librerías se encuentran dentro de la carpeta /WEB-INF/lib de la aplicación, de allí se pueden tomar para ubicarlas en las carpetas respectivas como indica la siguiente tabla⁶:

LIBRERIA	UBICACIÓN	DESCRIPCION
commons-fileupload-1.1.1.jar	\$CATALINA_HOME/common/lib \$CATALINA_HOME/webapps/aplicacion/ WEB-INF/lib	Librería que da la capacidad de carga de archivos a los servlets y aplicaciones web.
commons-io-1.2.jar	\$CATALINA_HOME/common/lib \$CATALINA_HOME/webapps/aplicacion/ WEB-INF/lib	Librería que permite la funcionalidad en el Manejo de archivos.
mail.jar	\$JAVA_HOME/jre/lib/ext	Librería que permite el envío de mensajería Mediante la aplicación.
mysql-connector-java-3.0.11-stable-bin.jar	\$CATALINA_HOME/webapps/aplicacion/ WEB-INF/lib \$JAVA_HOME/jre/lib/ext	Connector/J para la creación de aplicaciones de bases de datos con Java.
ifxjdbc.jar	\$CATALINA_HOME/webapps/aplicacion/ WEB-INF/lib \$JAVA_HOME/jre/lib/ext	Conector a bases de datos Informix.
activation.jar	\$CATALINA_HOME/webapps/aplicacion/ WEB-INF/lib \$JAVA_HOME/jre/lib/ext	Librería utilizada para determinar el tipo de paquete de datos que se esta recibiendo y determina la operación necesaria que se deba Realizar con el.

Cuadro 5.1: Librerías EIMWeb

5.5.5. Finalización.

Después de realizar todos los pasos anteriores correctamente, estamos en condiciones de hacer nuestra primera prueba.

⁶Las variables CATALINA_HOME Y JAVA_HOME hacen referencia a los directorios de instalación de Tomcat y de la máquina virtual de java.

Desde nuestro navegador web escribimos: <http://mecanica.wis.edu.co> lo que nos debería mostrar será:



Figura 5.16: Primera Imagen EIMWeb

Por primera vez, siempre se muestra el sitio ejecutándose con los colores de la escuela de Ingeniería de Sistemas, pero cabe resaltar que son los colores los que le dan el aspecto. El último paso es cambiar los colores mediante las Hojas de Estilo en Cascada (CSS), este proceso se realiza con la ayuda de un Diseñador Gráfico que ha tomado previamente los requerimientos de colores para fondos, imágenes, banners, logos, imágenes, letras que se desean en la escuela donde se implantó el sistema.

Después de realizar el proceso de cambio de colores, el sitio se vé de la siguiente forma:



Figura 5.17: Portal EIMWeb Implantado

Capítulo 6

PRUEBAS DEL SISTEMA

Las pruebas de funcionamiento del sistema se hicieron de forma general, es decir, se tuvo en cuenta la parte funcional del sitio, comenzando desde las conexiones a las bases de datos hasta el funcionamiento de cada uno de los módulos.

6.1. CONEXIÓN Y ACTUALIZACIÓN DE LAS BASES DE DATOS.

PRUEBA	DESCRIPCION	RESULTADO
Conexión a la BD de División	La Base de Datos de la División de Servicios de Información muestra los Datos referentes a la Escuela.	OK
Actualización de la BD de División del Servidor	Se actualiza la Base de Datos de División del servidor con la información de Horarios, Matriculas, Estudiantes, Docentes y Egresados	OK
Actualización de la BD del Sitio	Se actualiza la BD del sitio con la información de Usuarios, Docentes, Horarios, Matriculas, Egresados.	OK

Cuadro 6.1: Conexión y Actualización de Base de Datos.

6.2. NAVEGACIÓN EN EIMWEB

La forma de acceso al sistema se realiza de tres maneras:

- Usuario Invitado: Tiene acceso a toda la información institucional de la escuela, puede ser cualquier usuario que navegue por el portal, no necesariamente es un usuario registrado.
- Usuario Registrado: Tiene acceso a varios servicios que permiten interaccionen con las tareas de desarrolladas en la escuela, son los usuarios que están registrados en el sistema, Estudiantes, Docentes, y Administrativos de la escuela.
- Usuario Administrador: Son los usuarios que tienen acceso a la modificación de servicios del sistema, publicaciones, pueden crear grupos, eventos, encuestas, etc., Desarrolladores, Docentes, Diseñadores.

Cada forma de acceso posee ciertos permisos sobre los servicios ofrecidos por el sistema, por tanto las pruebas se realizaron sobre los módulos sobre los cuales está actuando la Parametrización.

6.2.1. Usuario Invitado

PRUEBA	DESCRIPCION	RESULTADO
Mostrar Información Institucional	Se muestra toda la información Institucional de la Escuela, la información de cada uno de los Menús de primer Nivel.	OK
Listado de Docentes	Se muestra la lista completa de docentes	OK
Horarios de Docentes	Al hacer clic sobre el nombre de un Docente o al acceder por el Menú horarios Docentes se muestra el horario completo del Docente	OK
Menú Programas	Se muestran los planes de pregrado, Especialización, Maestría y Planes de Estudio	OK
Menú ECI-HOY	Se muestran los comunicados, Agenda, Cartelera, Cursos y eventos propuestos.	OK
Menú Grupos	Se Muestran los grupos y la información de cada uno de los grupos.	OK
Recordatorio Contraseña	Si se es un Usuario Registrado y se hace recordatorio de contraseña, se asigna una nueva contraseña, se modifica en la BD y se envía y llega el correo de Confirmación.	OK
Consultas y Sugerencias	Se realizan consultas y sugerencias del sitio, y da confirmación de que la sugerencia ha sido enviada.	OK
Aula Virtual	Permite Mirar el material docente disponible	
Registro	Se hace registro masivo de Usuarios para hacerlos usuarios del sistema, crea el usuario con toda la información contenida en la BD con los datos del Usuario.	OK

Cuadro 6.2: Usuario Invitado

6.2.2. Usuario Registrado

PRUEBA	DESCRIPCION	RESULTADO
Aula Virtual	Crea material docente, ve listados de clase, hace cartelera, ve cartelera e información Estudiantes y Egresados.	OK
Listado de Docentes	Se muestra la lista completa de docentes	OK
Horarios de Docentes	Al hacer clic sobre el nombre de un Docente o al acceder por el Menú horarios Docentes se muestra el horario completo del Docente	OK
Menú Programas	Se muestran los planes de pregrado, Especialización, Maestría y Planes de Estudio	OK
Menú EIM-HOY	Se muestran los comunicados, Agenda, Cartelera, Cursos y eventos propuestos.	OK
Menú Grupos	Se Muestran los grupos y la información de cada uno de los grupos.	OK
Recordatorio Contraseña	Si se es un Usuario Registrado y se hace recordatorio de contraseña, se asigna una nueva contraseña, se modifica en la BD y se envía y llega el correo de Confirmación.	OK
Consultas y Sugerencias	Se realizan consultas y sugerencias del sitio, y da confirmación de que la sugerencia ha sido enviada.	OK
Subir Archivos	Se suben archivos al servidor, por Materias, Docentes, Grupos.	OK
Ver Archivos	Se ven los archivos subidos al servidor, de las Asignaturas, Docentes, Grupos.	OK
Participación en Foros	Se hacen participaciones en los Foros.	OK
Mostrar Información Institucional	Se muestra toda la información Institucional de la Escuela, la información de cada uno de los Menús de primer Nivel.	OK
Envío de Correos	Envío de correos por parte de los Estudiantes, y por parte de los Docentes.	OK

Cuadro 6.3: Usuario Registrado

6.2.3. Usuario Administrador

PRUEBA	DESCRIPCION	RESULTADO
Creación Manual de Usuario	Se crea de forma manual un usuario, permite entrar la información y guardar los cambios.	OK
Envío de Correos	Se envía Correos por parte del Administrador.	OK
Creación de Backups	Se generan Backups de la BD del sitio.	OK
Administración Publicaciones	Se crean, editan, Eliminan, guardan cambios a los Comunicados, Cartelera y Cursos.	OK
Administración de Grupos	Se crean Grupos, se editan, se modifican, se guardan cambios en los Grupos.	OK
Administración de Servicios	Se crean Servicios, se editan, se modifican y se guardan cambios en los Servicios.	OK
Administración de Usuarios	Permite mantener categorías, perfiles y usuarios	OK
Actualizar BDs	Actualiza Matriculas en división en diamante, actualiza docentes y ve contadores	
Administración de Sugerencias	Se reciben sugerencias, se muestran, se atienden, se eliminan, se envía y llega correo de respuesta a la sugerencia	OK

Cuadro 6.4: Usuario Administrador

Capítulo 7

CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

- Las herramientas utilizadas en el grupo de Desarrollo CALUMET, son un claro ejemplo del alcance que pueden ofrecer estas tecnologías de desarrollo Web, con las cuales se ha logrado la construcción del portal Web de la Escuela de Ingeniería de sistemas EISIWeb, y que ha sido de gran utilidad para los usuarios del sistema dentro de la escuela cumpliéndose así el objetivo principal de EISIWeb, “Construir Comunidad”.
- Tanto el hardware como el sistema operativo con el que se trabajó en este proyecto, permitieron que el proceso de instalación y configuración fuera exitoso. Además nos asegura en gran medida, estabilidad y seguridad para todos los usuarios (Estudiantes, Docentes y Personal Administrativo) de EIMWeb.
- Se demostró que el trabajo que se ha venido realizando en el grupo de software CALUMET, esta siendo tenido en cuenta, no solo en nuestra escuela, sino también en toda la comunidad universitaria, lo que nos permite seguir motivados y avanzando en la mejora de servicios.
- Con el desarrollo de este proyecto es claro que el portal web EISIWeb, puede ser exitosamente parametrizado e implantado en los más importantes sistemas

servidores que existen actualmente; desplegando toda su funcionalidad al servicio de toda la comunidad de la Escuela de Ingeniería Mecánica.

7.2. RECOMENDACIONES

- Generar acciones de mantenimiento programadas en todas las escuelas que tienen implantado nuestro portal web EISIWeb, tratando de crear documentos históricos que ayuden en la resolución de problemas comunes.
- Actualización de algunos servicios del sistema operativo Mac OS X Server, a versiones mas nuevas que permitan estar al día y aprovechar los adelantos que nos ofrezcan las nuevas tecnologías.
- Mejorar constantemente los servicios ofrecidos por los portales web EISIWeb, EEIEWeb, EICWeb y EIMWeb, buscando solucionar problemas propios de cada escuela, permitiendo el aprovechamiento al máximo de las herramientas disponibles para todos los usuarios.
- Seguir con el proceso de registro masivo de los usuarios, con el objetivo de registrar al 100 % de los integrantes de la Escuela de Ingeniería Mecánica (Estudiantes, Egresado, Docentes y Personal Administrativo).

GLOSARIO

CGI: (Common Gateway Interface). Es una norma para establecer comunicación entre un servidor Web y un programa, de tal modo que este último pueda interactuar con Internet. También se usa la palabra CGI para referirse al programa mismo, que se ejecuta en tiempo real en un Web Server en respuesta a una solicitud de un navegador.

COOKIE: Es un pequeño documento en formato de texto que es grabado y acogido por el disco duro del computador del usuario. Se utiliza para mantener el estado de una aplicación o seguir la trayectoria del usuario en el sitio.

HIPERTEXTO: Cualquier texto disponible en el World Wide Web que contenga enlaces con otros documentos.

HTML: (HiperText Markup Lenguaje, Lenguaje de Marcado de Hipertexto). Lenguaje empleado para describir el interior de los documentos Web, basado en el uso de etiquetas. Permite describir hipertexto con enlaces (hiperlinks) que conducen a otros documentos o fuentes de información relacionadas y con inserciones multimedia (gráficos, sonido...).

HTTP: (HiperText Transfer Protocol, Protocolo de Transferencia de Hipertexto). Lenguaje empleado para describir cómo se envían los documentos HTML por Internet. HTTP proporciona las normas para que los navegadores hagan peticiones y los servidores entreguen respuestas.

INTERNET: Red global de comunicaciones que interconecta computadoras y bases de datos diseminadas por todo el planeta.

IP: (Internet Protocol). Protocolo que provee las funciones básicas de direccionamiento en Internet y en cualquier red TCP/IP (software de comunicación). El protocolo

de Internet se encarga de poner una etiqueta con la dirección adecuada a cada paquete, ya que cada computador conectado a la red tiene una dirección de Internet única que lo distingue de cualquier otro computador en el mundo.

JAVA: Lenguaje de programación orientado a objetos desarrollado por SUN Microsystems. Está diseñado para utilizarse en entorno distribuido de Internet.

JAVASCRIPT: Lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. La compilación es realizada en el cliente por parte del navegador.

JDBC: (Java Database Connectivity, Conectividad de Base de Datos Java). Es una especificación de la interfaz de programa de aplicación (API), para conectar los programas escritos en Java a los datos en bases de datos.

JSP: (Java Server Pages). Plantilla para una página Web que emplea código Java, para generar un documento HTML dinámicamente. Las páginas JSP se ejecutan en un componente del servidor conocido como contenedor de JSP, que las traduce a servlets (Java equivalentes).

Mac OS X: Es una línea de sistemas operativos gráficos desarrollados y vendidos por la compañía Apple Inc, especialmente para ser usados en computadoras Macintosh y/o dispositivos como el iPhone, el iPod y similares. Mac OS X es el sucesor del original Mac OS de 1984, primer sistema operativo de Apple. Pero, a diferencia de su predecesor, el Mac OS X está basado en los sistemas operativos Unix. Fue en el año 1997 cuando Steve Jobs, nombrado CEO de Apple, decidió terminar con la versión clásica y crear este nuevo sistema operativo usando tecnología del sistema operativo NEXTSTEP de la compañía NeXT (adquirida por Apple a principios de ese año). La primera versión fue lanzada en 1999 con el nombre de Mac OS X Server 1.0, seguida por una versión orientada a escritorio, la Mac OS X v10.0 en marzo de 2001. El Mac OS X utiliza el gestor de ventanas Aqua. Las versiones para dispositivos pequeños, como los iPhone y los iPod, son versiones reducidas del sistema operativo.

SERVLET: Son clases Java que amplían la funcionalidad de un servidor Web, mediante la generación dinámica de páginas Web.

MOTOR DE SERVLETS: Administra la carga y descarga del servlet y trabaja con el servidor Web para dirigir peticiones a los servlets y enviar la respuesta a los clientes.

PÁGINA WEB: Servicio de Internet que permite el hipertexto (permite ir de una página a otra enlazando el hipermedia). Presenta documentos con texto, imagen estática y en movimiento, sonido, video, etc. y utiliza el estándar HTML.

PÁGINA WEB DINÁMICA: Página Web cuyo contenido es calculado por el servidor en el momento en que el usuario accede a ella. Normalmente el contenido se obtiene desde una base de datos.

PÁGINA WEB ESTÁTICA: Página Web con textos y otro tipo de archivos (imágenes, multimedia, etc.) que contiene toda la información necesaria y se muestra al tiempo que es solicitada.

SERVIDOR WEB: Servidor que almacena las páginas de un sitio Web y envía páginas Web en respuesta a la peticiones HTTP hechas desde los navegadores de los clientes.

PROTOCOLO: conjunto de reglas que gobiernan el intercambio de datos entre entidades dentro de una red. Es el lenguaje común “que utilizan” los ordenadores para “hablar” y entenderse entre sí. Existen muchos tipos de protocolos cada uno con sus reglas bien definidas, como por ejemplo: FTP, POP3, SMTP, ICMP, etc.

NAT: (Network Address Translation - Traducción de Dirección de Red) es un mecanismo utilizado por routers IP para intercambiar paquetes entre dos redes que se asignan mutuamente direcciones incompatibles. Consiste en convertir en tiempo real las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes para permitir la operación de protocolos que incluyen información de direcciones dentro de la conversación del protocolo.

Bibliografía

- [1] BRUEGGE BY DUTOIT A. Ingeniería de Software Orientado a Objetos. Prentice Hall. 2002. Aquí encontramos todo lo relacionado con las metodologías de desarrollo software
- [2] PRESSMAN, R. Ingeniería de Software, Un enfoque práctico. Quinta Edición. McGraw-Hill. 2002. En este libro encontramos teoría de la metodología aplicada en el diseño y desarrollo de software.
- [3] IEEE Recommended Practice for Software Requirements Specifications (SRS). IEEE Std 830-1998. IEEE Computer Society. Especificaciones y normas de la IEEE para una buena especificación de requisitos.
- [4] IEEE Std 1016-1998 Recommended Practice for Software Design Descriptions. Especificaciones y normas de la IEEE para el diseño de software.
- [5] JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Editorial Addison Wesley. Madrid, España 2000. Nos brinda la teoría sobre el proceso unificado de desarrollo de software e ilustra técnicas que apoyan el ciclo de vida completo de desarrollo.
- [6] SWEBOK – Guide to the Software Engineering Body Of Knowledge, 2004 version. Metodologías para el desarrollo de software.
- [7] Mario G. Piattini, José A. Calvo-Manzano, Joaquín Cervera y Luis Fernández. Análisis y diseño detallado de Aplicaciones Informáticas de Gestión. Alfaomega. 2000. Nos proporciona técnicas para el buen modelamiento de las aplicaciones informáticas.

- [8] LEMAY, Laura; PERKINS, Charles. Aprendiendo JAVA en 21 días. Editorial Prentice Hall. México, México D.C. 1996. Manual de referencia para principiantes sobre JAVA.
- [9] KROENKE, David. Procesamiento de Bases de Datos Fundamentos, Diseño e Instrumentación. Editorial Prentice-Hall Hispanoamericana, S.A. México, México D.C. 1996. Nos brinda fundamentos de Bases de datos.
- [10] SILBERSCHTZ, Abraham; KORTH, Henry; SUDARSHAN, S. Fundamentos de Bases de Datos. Tercera Edición. McGraw-Hill. España, 1998. Este libro presenta los conceptos fundamentales de desarrollo y administración de bases de datos.
- [11] JOHNSON, James. Bases de datos: Modelos lenguajes y diseño. Primera edición. Oxford. 2000. Presenta temas de teoría de bases de datos: modelos y métodos de acceso, administración, diseño de aplicaciones.
- [12] HANSEN, Gary. Hansen James. Diseño y administración de bases de datos. Segunda edición. Prentice Hall. 1997. Se enfoca en los fundamentos teóricos de planificación estratégica, diseño, implementación, administración y gestión del entorno de bases de datos.
- [13] <http://www.apple.com/support/macosxtigerserver/>. Sitio oficial de apple que da soporte a los usuarios del Sistema Operativo Mac OS X Server Tiger.
- [14] www.tomcat.apache.org/connectors-doc/. En este sitio se encuentra la documentación para realizar la conexión entre el servidor apache y el servidor Tomcat para que funcionen conjuntamente por medio del conector JK.
- [15] PDF MySQL Reference Manual. Manual que contiene información completa y detallada sobre MySQL. Facilitado por los anteriores desarrolladores de grupo de desarrollo CALUMET.
- [16] PDF Hanna, Phil. JSP Manual de referencia. Manual completo de JSP. Facilitado por los anteriores desarrolladores de grupo de desarrollo CALUMET.
- [17] www.htmlgoodies.com/primers/jsp. Guía práctica para principiantes en jsp, ilustración de conceptos básicos de este lenguaje mediante ejemplos.

- [18] www.senavirtual.edu.co. Diversos cursos virtuales en el desarrollo de software basado en la Web: Desarrollo de aplicaciones con manejo simultaneo y uso menús java, curso de desarrollo de applets con interfaz gráfica, manejo de eventos, clases y objetos java, curso de creación de páginas Web con HTML y java script. Los anteriores cursos fueron realizados y aprobados por los integrantes del proyecto y nos van a facilitar un mejor desarrollo de las aplicaciones que se necesitarán en el mismo.
- [19] www.programacion.com/java/tutorial. En este sitio se encuentran una gran variedad de tutoriales relacionados con: Los APIS, las herramientas y los servicios que proporciona los Servlets y las Java Server Pages (JSP), los tópicos necesarios para la programación de acceso a bases de datos en Java.
- [20] www.astalaweb.com. Sitio Web que presenta una guía de todo lo relacionado con Java Script.
- [21] www.programacionfacil.com Sitio que nos brinda la facilidad de encontrar y consultar código fuente y tutoriales que nos servirá para la implementación de los módulos.
- [22] <http://es.wikipedia.org/wiki/Wikipedia:Portada>: Enciclopedia online, de acceso libre que permitió buscar definiciones como para Arquitectura Cliente/Servidor, Servidor Web, Conector JK, y otras Definiciones.