

**PARAMETRIZACIÓN DE EISIWEB E IMPLANTACIÓN DEL PORTAL DE LA
ESCUELA DE INGENIERÍA CIVIL ECIWEB**

FREDY ANDRÉS GÉLVEZ DÍAZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2008**

**PARAMETRIZACIÓN DE EISIWEB E IMPLANTACIÓN DEL PORTAL DE LA
ESCUELA DE INGENIERÍA CIVIL ECIWEB**

FREDY ANDRÉS GÉLVEZ DÍAZ

Trabajo de grado para optar el título de Ingeniero de Sistemas

Director

Msc. LUIS IGNACIO GONZALEZ RAMIREZ

Magíster en Informática

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA**

2008

DEDICATORIA

A Dios,

Por ser mí guía, y permitirme culminar esta etapa tan importante en mi vida.

A mi Padre,

A mi Madre,

A mis hermanas,

Por su incondicional cariño, apoyo y confianza que me han brindado durante toda mi vida.

A todas las personas especiales que han pasado por mi vida, las que se han ido y a las que aún están para seguir creyendo en mí.

AGRADECIMIENTOS

Agradezco especialmente a Dios por ser mi guía durante el desarrollo de este proyecto, y permitir que todas las personas que son especiales para mí estén acá para compartir con ellas este inmenso triunfo.

Al Ingeniero Luis Ignacio González por su constante apoyo, dirección y aportes, que fueron posibles para sacar el proyecto adelante.

A la escuela de Ingeniería Civil por permitir el montaje de nuestro Sistema Web en su Escuela y permitir demostrar los alcances que tiene nuestro portal.

A todo el Grupo de Desarrollo CALUMET, por enseñar lo que es el trabajo y apoyo de grupo, solución de dudas, y entregar conocimiento.

A mis amigos y compañeros que siempre de alguna u otra forma creyeron y esperaron lo mejor de mí para sacar el proyecto adelante.

A todas las personas que hacen buen uso de los portales EISIWeb y ECIWeb, para su desempeño como Estudiantes, Docentes y Administrativos dentro de cada una de sus escuelas.

TABLA DE CONTENIDO

INTRODUCCION	1
1. PRESENTACIÓN DEL PROYECTO	3
1.1. ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME	3
1.2. ANTECEDENTES	4
1.3. DEFINICIÓN DEL PROBLEMA	6
1.4. OBJETIVOS	6
1.4.1. Objetivos generales	6
1.4.2. Objetivos específicos	7
1.5. JUSTIFICACIÓN	8
1.6. ALCANCES Y LIMITACIONES	9
2. MARCO TEÓRICO	11
2.1. INTERNET	11
2.1.1. Servicios de Internet	11
2.2. ARQUITECTURA CLIENTE-SERVIDOR	12
2.2.1. Características de la arquitectura Cliente-Servidor	13
2.2.2. Arquitecturas Cliente-Servidor	13
2.2.2.1. Arquitectura Cliente-Servidor de Dos Capas	14
2.2.2.2. Arquitectura Cliente-Servidor de Tres Capas	15

2.2.3.	Clasificación de los sistemas Cliente/Servidor	16
2.2.4.	Arquitectura Cliente-Servidor utilizada	17
2.3.	HERRAMIENTAS DE DESARROLLO DE PÁGINAS DINÁMICAS	17
2.3.1.	Clasificación de las herramientas de desarrollo de páginas dinámicas	18
2.3.1.1.	Client Side Scripts	18
2.3.1.2.	Server Side Scripts	18
2.3.2.	Herramienta utilizada	20
2.3.2.1.	Modelo de acceso a JSP	21
2.4.	BASES DE DATOS	22
2.4.1.	Componentes principales de una base de datos	23
2.4.2.	Tipos de modelos de datos	24
2.4.2.1.	El modelo jerárquico	24
2.4.2.2.	El modelo de red	24
2.4.2.3.	El modelo relacional	25
2.4.3.	Conectividad a base de datos	25
2.4.3.1.	Conectores más utilizados	26
2.4.4.	Conectividad utilizada	27
2.4.5.	Manejadores de bases de datos	27
2.4.5.1.	MySQL	28
2.4.5.1.1.	Características de MySQL	28
2.4.5.1.2.	Ventajas de usar MySQL	28
2.5.	PROGRAMACIÓN DE APLICACIONES	29
2.5.1.	Programación Estructurada (PE)	29
2.5.1.1.	Programación Orientada a Eventos	30
2.5.2.	Programación Genérica (PG)	30
2.5.3.	Programación Orientada a Objetos (POO)	30
2.5.3.1.	Clases y objetos	31

2.5.3.2.	Atributos.....	31
2.5.3.3.	Métodos	32
2.5.3.4.	Herencia.....	33
2.5.3.5.	Ventajas de la POO	33
2.6.	SERVIDOR WEB	34
2.6.1.	Tipos de servidores Web.....	39
2.6.2.	Servidor Web utilizado	39
2.6.2.1.	Servidor Apache	40
2.6.2.2.	Contenedor Tomcat.....	41
2.6.2.3.	Apache y Tomcat en trabajo conjunto:.....	41
3.	MARCO METODOLOGICO	43
3.1.	EFFECTIVIDAD DEL PROTOTIPADO EVOLUTIVO	47
3.2.	PROBLEMAS DEL PROTOTIPADO EVOLUTIVO	47
4.	MANUAL DE INSTALACION Y CONFIGURACION DEL SERVIDOR WEB.....	48
4.1.	INSTALACIÓN DEL SISTEMA OPERATIVO.....	48
4.2.	CONFIGURACION DEL SISTEMA OPERATIVO	67
4.2.1.	CONFIGURACION DE LA RED	68
4.2.2.	CONFIGURACIÓN DE LAS RÉPLICAS DE RED.....	71
4.3.	CONFIGURACION DEL EQUIPO COMO SERVIDOR WEB	72
4.3.1.	INSTALACION DE PAQUETES NECESARIOS	72
4.3.2.	PRUEBAS DE FUNCIONAMIENTO DE PAQUETES INSTALADOS.....	74
4.3.3.	CONFIGURACION DE LOS PAQUETES INSTALADOS	78
4.3.4.	INSTALACION Y CONFIGURACION DEL FIREWALL	86
5.	PROCESO DE PARAMETRIZACION.....	92

5.1. BUSQUEDA DE PATRONES	92
5.1.1. NAVEGACIÓN DEL SITIO EN FUNCIONAMIENTO.....	93
5.1.2. EXPLORACIÓN DE ARCHIVOS.....	95
5.2. PARAMETRIZACIÓN DE EISIWEB	97
5.3. PROCESO DE IMPLANTACIÓN DEL PORTAL EN OTRA ESCUELA ...	100
5.3.1. COMPATIBILIDAD CON EL SERVIDOR	100
5.3.2. ALOJAMIENTO DEL PORTAL EN EL NUEVO SERVIDOR	102
5.3.3. CARGANDO LAS BASES DE DATOS DEL SITIO EN MySQL	103
5.3.4. LIBRERIAS .JAR NECESARIAS	106
5.3.5. REINICIANDO SERVICIOS.....	107
5.3.6. CORRIENDO EL SITIO POR PRIMERA VEZ	108
6. PRUEBAS DEL SISTEMA	112
6.1. CONEXIÓN Y ACTUALIZACIÓN DE LAS BASES DE DATOS	112
6.2. NAVEGACION.....	112
6.2.1. USUARIO INVITADO	113
6.2.2. USUARIO REGISTRADO.....	114
6.2.3. USUARIO ADMINISTRATIVO.....	115
7. CONCLUSIONES.....	116
8. SUGERENCIAS Y RECOMENDACIONES.....	118
BIBLIOGRAFIA	119
TEXTOS BIBLIOGRÁFICOS.....	119
HERRAMIENTAS Y PÁGINAS WEB UTILIZADAS.....	121

LISTA DE TABLAS

Tabla 1 Arquitectura Utilizada	17
Tabla 2 Características Principales de Cada Tecnología	20
Tabla 3 Programas para instalación en el servidor	73
Tabla 4 Parámetros Externos	94
Tabla 5 Parámetros Internos.....	97
Tabla 6 Librerías necesarias para el funcionamiento del Portal.....	106
Tabla 7 Pruebas de Conexión y Actualización de las BD	112
Tabla 8 Pruebas de Usuario Invitado.....	114
Tabla 9 Pruebas de Usuario Registrado	115
Tabla 10 Pruebas de Usuario Administrativo	115

LISTA DE FIGURAS

Fig. 1 Arquitectura Cliente / Servidor	12
Fig. 2 Arquitectura Cliente / Servidor de dos capas	14
Fig. 3 Arquitectura Cliente / Servidor de tres capas	15
Fig. 4 Modelo de acceso a JavaServer Pages.....	21
Fig. 5 Funcionamiento de un servidor Web	35
Fig. 6 Arquitectura de un Servidor Web	36
Fig. 7 Prototipado Evolutivo	43
Fig. 8 Diagrama de Flujo del Prototipado Evolutivo	45
Fig. 9 Procedimiento del Prototipado Evolutivo	46
Fig. 10 Pantalla de bienvenida a la instalación	49
Fig. 11 Selección del lenguaje de Sistema Operativo.....	50
Fig. 12 Selección de la Ubicación	50
Fig. 13 Selección de distribución del Teclado	51
Fig. 14 Configuración de la Red 1.....	51
Fig. 15 Configuración de la Red 2.....	52
Fig. 16 Particionado de Disco 1	52
Fig. 17 Particionado de Disco 2	53
Fig. 18 Particionado de Disco 3	53
Fig. 19 Particionado de Disco 4	54
Fig. 20 Particionado de Disco 5	54
Fig. 21 Particionado de Disco 6	55
Fig. 22 Particionado de Disco 7	55
Fig. 23 Particionado de Disco 8	56
Fig. 24 Particionado de Disco 9	56
Fig. 25 Particionado de Disco 10	57

Fig. 26 Particionado de Disco 11	58
Fig. 27 Particionado de Disco 12	59
Fig. 28 Particionado de Disco 13	59
Fig. 29 Particionado de Disco 14	60
Fig. 30 Configuración de Usuarios y Contraseñas 1	60
Fig. 31 Configuración de Usuarios y Contraseñas 2	61
Fig. 32 Configuración de Usuarios y Contraseñas 3	61
Fig. 33 Configuración de Usuarios y Contraseñas 4	62
Fig. 34 Instalando el Sistema Base	62
Fig. 35 Configurar el Gestor de Paquetes 1	62
Fig. 36 Configurar el Gestor de Paquetes 2	63
Fig. 37 Selección de Programas	63
Fig. 38 Seleccionar e Instalar Programas	64
Fig. 39 Instalación del cargador de arranque GRUB	64
Fig. 40 Gestor de Arranque GRUB	65
Fig. 41 Pantalla de Registro de Usuario	66
Fig. 42 Entorno Gráfico Linux Debian 4	67
Fig. 43 Apache corriendo en el Navegador	75
Fig. 44 Configuración de Firewall 1	88
Fig. 45 Configuración de Firewall 2	88
Fig. 46 Configuración de Firewall 3	89
Fig. 47 Configuración de Firewall 4	89
Fig. 48 Configuración de Firewall 5	90
Fig. 49 Configuración de Firewall 6	90
Fig. 50 Configuración de Firewall 7	91
Fig. 51 Primera Vista Preliminar del Sitio Implantado	109
Fig. 52 Vista Preliminar del Sitio implantado en la Escuela de Ingeniería Civil....	110
Fig. 53 Vista Prelim Sitio implantado en la Escuela de Ingeniería Industrial.....	111

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).

LINUX: Sistema operativo. Es una implementación de libre distribución UNIX para computadores personales, servidores y estaciones de trabajo. Consta de componentes GNU y el kernell desarrollado por Linux Torvalds.

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.

RÉPLICA DE RED: Dirección de un servidor donde se encuentran alojados los repositorios para actualizaciones de un sistema operativo.

REPOSITORIO: Programas para realizar actualizaciones de funcionamiento y de seguridad de un sistema operativo.

RESUMEN

TÍTULO: PARAMETRIZACIÓN DE EISIWEB E IMPLANTACIÓN DEL PORTAL DE LA ESCUELA DE INGENIERÍA CIVIL ECIWEB. *

AUTOR: FREDY ANDRÉS GÉLVEZ DÍAZ **

PALABRAS CLAVE

Sitio Web, EISI (Escuela de Ingeniería de Sistemas), Portal Web, EISIWeb, Implantación, EICI (Escuela de Ingeniería Civil), ECIWeb.

DESCRIPCIÓN

Ante la necesidad de una herramienta que permita la interacción entre los miembros de la Escuela de Ingeniería de Sistemas (Estudiantes, Egresados, Profesores, Administrativos) se desarrolló el portal Web de la Escuela EISIWeb.

Este se encuentra en funcionamiento y ha obtenido muy buenos resultados a la comunidad gracias a los servicios que ofrecen sus módulos. Estos últimos se encuentran en constante mejora y desarrollo, y se trabaja en la investigación de nuevos módulos que puedan ofrecer nuevos y mejores servicios.

Actualmente, el Portal Web de la Escuela EISIWeb, 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 de esta índole, 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.

Cada día, con el crecimiento de las Escuelas, se hace más necesario herramientas que permitan la interacción e integración de la comunidad.

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 como esta.

* Trabajo de grado. Modalidad: Práctica Empresarial

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: Luis Ignacio González Ramírez

ABSTRACT

TITLE: PARAMETERIZATION OF EISIWEB AND IMPLEMENTATION OF THE WEB SITE OF THE SCHOOL OF CIVIL ENGINEERING ECIWEB. *

AUTHOR: FREDY ANDRES GÉLVEZ DIAZ **

KEY WORDS

Sitio Web, EISI (School of Systems Engineering), Portal Web, EISIWeb, Implantación, EICI (School of Civil Engineering), ECIWeb, UIS (Industrial University of Santander)

DESCRIPTION

Faced with the need for a tool that interact among the members of the School of Engineering Systems (Students, teachers, administrators) are developing the web portal of the School EISIWeb, which is in full operation, it has obtained very good results with modules that comprise it, while these modules are in constant development and is working in the research of new modules that can offer new and better services.

Currently, the Website of the College EISIWeb 3.0, offers users comprehensive information on programs, curricula, calendars, billboards, public notices, schedules, teachers, groups, among other services, which enable the community to be aware of all events and developments that are planned or are offering this form of a convenient tool for interaction, information and integration.

The need for this type of tool is not unique to the school of engineering systems, it can be said with certainty that all the Schools of the UIS urgently require a tool that allows them to easily publish information that consistently reaches the directions related to business practices, job opportunities, academic and administrative processes.

Likewise, for many academic units is of vital importance to publish their corporate information, such as vision, mission, boards, calendars of courses, graduates, skills, technical weeks and other.

Every day, with the growth of the schools, it becomes more necessary tools for interaction and integration of the community.

It is for this that from the Calumet web development group has taken the decision to create a website template, parameterized, scalable and easy-to-deploy at any other school that is interested in having a tool like this.

* Modality: Business Practice

** Physical Mechanical Engineering Faculty, the Systems Engineering School.
Director: Msc. Luis Ignacio González Ramírez.

INTRODUCCION

Durante el desarrollo del proyecto se aprendieron nuevas tecnologías de desarrollo Web, así como nuevos sistemas operativos. Hay que tener en cuenta que el desarrollo Web es multiplataforma, y por tanto, es muy importante aprender diferentes entornos de desarrollo, lo que contribuye al desarrollo profesional y personal.

El estudio de estas nuevas tecnologías y de los Sistemas Operativos ha sido pieza fundamental en el Grupo de Desarrollo CALUMET, que mediante la investigación y la implementación, ha logrado un gran avance a través de muchos años de trabajo y mejoramiento continuo. Este trabajo ha sido reconocido por otras escuelas de la Universidad Industrial de Santander, que han querido implementar en sus escuelas el Portal Web del que hace uso actualmente la escuela de Ingeniería de Sistemas. Para esto, es necesario que el Portal sea un software parametrizado de manera que permita una implantación rápida, sencilla y 100% compatible con los servidores y lenguajes de programación con los que se implemente.

Para el desarrollo de este proyecto y la lectura de este documento, es necesario tener conocimientos de desarrollo Java, JSP, y muy importante, en el Sistema Operativo Linux Debian, ya que los manuales aquí contenidos explican los procesos de desarrollo del proyecto, pero no se hace explicación muy puntual de algunas cosas ya que son procesos que el lector debe saber previamente como son el uso de editores, navegadores y comando de consola de Linux Debian. Dado que estos procesos se realizan de esta manera, son muy pocas las cosas que se realizan en el entorno gráfico.

Es fundamental que el lector, antes de utilizar estos manuales, tenga un conocimiento amplio del funcionamiento del sitio ya que durante el proceso de implantación del Portal parametrizado se están obviando muchos pasos que se supone ya se debe tener como conocimiento previo.

Este es uno de los objetivos primordiales de este documento, mostrar como se llevo acabo ese proceso de Parametrización e implantación del portal EISIWeb en otras escuelas.

Se espera entonces que sea de total agrado para el lector y que sirva para nuevas ideas que permitan el desarrollo de nuevos proyectos, con el objetivo de encontrar un sistema que sea cada vez mejor.

1. PRESENTACIÓN DEL PROYECTO

1.1. ORIENTACIÓN SOBRE EL CONTENIDO DEL INFORME

Este documento contiene un informe detallado de cada una de las etapas llevadas a cabo durante el desarrollo del proyecto: PARAMETRIZACIÓN DE EISIWEB E IMPLANTACIÓN DEL PORTAL WEB DE LA ESCUELA DE INGENIERÍA CIVIL ECIWEB

Su distribución es la siguiente:

CAPITULO 1. PRESENTACIÓN DEL PROYECTO: Se hace un análisis de los antecedentes del proyecto, definición del problema, objetivos generales, específicos, justificación, alcances y limitaciones del proyecto.

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

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

CAPITULO 4. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR WEB: Presenta paso a paso la Instalación y configuración del servidor Web de Ingeniería Civil, Albatros, y en general para cualquier otro servidor.

CAPITULO 5. PROCESO DE PARAMETRIZACION: Este capítulo describe el proceso de Parametrización del portal Web de la Escuela de Ingeniería de Sistemas EISIWeb y la implantación del Portal parametrizado en las otras escuelas.

CAPITULO 6. DOCUMENTO DE PRUEBAS DEL SISTEMA: En este capítulo se presentan los resultados de las pruebas realizadas al portal implantado en las diferentes escuelas.

CAPITULO 7. CONCLUSIONES: Se dan las conclusiones del proyecto.

CAPÍTULO 8. SUGERENCIAS Y RECOMENDACIONES: Finaliza con las sugerencias y/o recomendaciones por parte del autor del proyecto.

1.2. ANTECEDENTES

“CONSTRUIR COMUNIDAD” es el lema bajo el cual se desarrolló EISIWeb. Para esto decidimos que era necesario tener publicada toda la información institucional como la misión, visión, estructura organizacional, ubicación geográfica, directivos, profesores, grupos, etc. Pero también vimos 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.

La primera implantación del sitio EISIWeb se realizará para la escuela de Ingeniería Civil con el nombre ECIWeb, el cual prestará los mismos servicios que ofrece actualmente el sitio de la escuela de Ingeniería de Sistemas. Pero no solo se quiere implantar en esta escuela, sino en todas las escuelas de la Universidad Industrial de Santander.

El Portal 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.

1.3. DEFINICIÓN DEL PROBLEMA

El portal Web de la Escuela de Ingeniería de sistemas ha tomado gran auge dentro de la misma escuela, ya que los nuevos usuarios (estudiantes de primeros niveles) han adoptado este paradigma nuevo en la escuela y se ha convertido en una necesidad gracias a los servicios que presta, permitiendo que interactúen los miembros que la componen (Estudiantes, Egresados, Docentes, Administrativos). Esta necesidad no es única de la escuela de Ingeniería de Sistemas, ella la presentan un buen número de escuelas que han visto crecer el número de sus miembros considerablemente. De lo anterior, nace la idea de implantar el portal Web de la Escuela de Ingeniería de Sistemas en otras escuelas.

¿Pero como se logra? Parametrizando el portal EISIWeb de manera que sea de fácil implantación en otros servidores.

1.4. OBJETIVOS

1.4.1. Objetivos generales

- Parametrizar el Portal de la Escuela EISIWeb para permitir de una manera sistemática la implementación del sitio en otras escuelas.
- Implantar el portal de la escuela de Ingeniería Civil ECIWeb.
- Capacitación a los nuevos integrantes del grupo de desarrollo en el desarrollo de nuevos módulos adoptando el esquema de sistema parametrizado.

1.4.2. Objetivos específicos

- Instalar y configurar el Sistema Operativo Linux Debian 4 en el equipo asignado como servidor “ALBATROS” para la escuela de Ingeniería Civil.
- 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 Civil ECIWeb y accederlo vía Internet.
- Realizar las instalaciones y configuraciones necesarias para acceder las Bases de Datos de la División de Servicios de Información con la información necesaria de la Escuela de Ingeniería Civil para el correcto funcionamiento del Portal.
- 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 ECIWeb.
- Realizar la Implantación del Sitio Web de la Escuela de Ingeniería de Sistemas EISIWeb al sitio Web de la Escuela de Ingeniería Civil ECIWeb.
- Depurar el Sitio de la Escuela de Ingeniería de Sistemas EISIWeb para eliminar archivos que contiene que ya no son usados por el sistema.
- Identificar puntos críticos en los archivos contenidos en el sistema para ser adecuado a otras escuelas.

- Crear un interfaz que contenga los parámetros con los valores que varían de una escuela a otra.
- Editar cada uno de los archivos que contiene el sitio y realizar los cambios respectivos en el código fuente para que el sitio quede completamente parametrizado.
- Realizar pruebas minuciosas de todos y cada uno de los módulos del sitio en busca de posibles fallos.
- Recoger las sugerencias y opiniones que los usuarios del portal EISI tengan, debido a que ellos están en constante interacción con el portal, y las cuales se tendrán en cuenta para proponerlas como mejoramiento y ofrecimiento de nuevos servicios en la versión 4.0 del portal EISI WEB.
- 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.

1.5. JUSTIFICACIÓN

El Portal Web de la Escuela, EISIWeb 3.0, 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 Parametrizar todas las variables de EISIWeb, de tal forma que hacia el futuro podemos de una manera sistemática implantar el sitio de otras escuelas. Para empezar este proceso, se ha decidido junto con la escuela de Ingeniería Civil iniciar el proceso de implantación del sitio Web partiendo del que ya se tiene en producción en nuestra escuela.

1.6. ALCANCES Y LIMITACIONES

- **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.

- **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 recurso de hardware, lo que se puede lograr son proyectos mucho más robustos, desarrollados en un tiempo menor.

2. MARCO TEÓRICO

2.1. INTERNET

Internet es una gran cantidad de redes de computadoras interconectadas entre sí que están distribuidas por todo el mundo, y en ellas se puede encontrar información y servicios de todo tipo.

2.1.1. Servicios de Internet

Internet ofrece una gran variedad de servicios consistentes en métodos de comunicación y entretenimiento, entre los cuales podemos destacar:

- **World Wide Web:** Este servicio permite conectarse con un ordenador remoto y acceder a la información que proporciona, ya sea texto, imágenes, sonido o programas.
- **Correo Electrónico:** Uno de los recursos más importantes de Internet. A diario gran cantidad de mensajes son enviados a algún lugar del mundo y estos encuentran su destino en unos pocos segundos o minutos. Para que esto sea posible, solo necesitamos una cuenta de acceso a Internet, una dirección de correo electrónico, un computador, y el programa o sistema que gestionará nuestro correo electrónico.
- **FTP: (File Transfer Protocol)** Es la forma de acceder a información en computadoras remotas. Esta información puede consistir en imágenes o programas de computador.

· **Sesión remota: (TELNET)** Este servicio nos permite conectarnos a cualquier computador remoto conectado a Internet, aunque es necesario contar con una cuenta de acceso a éste.

2.2. ARQUITECTURA CLIENTE-SERVIDOR

Clientes y servidores son entidades lógicas independientes que operan en conjunto a través de una red para realizar una tarea. En un esquema cliente servidor, se denomina cliente la máquina que solicita un determinado servicio y servidor la máquina que lo proporciona. Una vez que el servidor acepta el pedido, la información requerida es suministrada al cliente que efectuó la petición, siendo este último el responsable de proporcionar los datos al usuario con el formato adecuado. Finalmente debemos precisar que cliente y servidor no tienen que estar necesariamente en ordenadores separados, sino que pueden ser programas diferentes que se ejecuten en el mismo ordenador.



Fig. 1 Arquitectura Cliente / Servidor

2.2.1. Características de la arquitectura Cliente-Servidor

- Orientado a servicios: El servidor los ofrece y el cliente los consume.
- Compartir recursos: Servicios ofrecidos a muchos clientes. Transparencia de ubicación: El servidor es un proceso que puede residir en el mismo aparato que el cliente o en un aparato distinto a lo largo de una red.
- Mezcla e igualdad: Una aplicación cliente/servidor, idealmente es independiente Del hardware y de los sistemas operativos.

2.2.2. 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:** Tiene que ver con la presentación al usuario de un conjunto de objetos visuales y llevar a cabo el procesamiento de los datos producidos por el mismo y los devueltos por el servidor.
- **Lógica de aplicación:** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos:** Esta compuesta de los archivos que contienen los datos de la aplicación.

2.2.2.1. Arquitectura Cliente-Servidor de Dos Capas

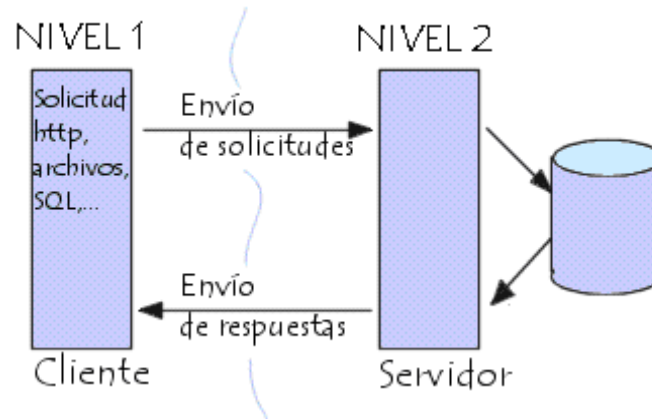


Fig. 2 Arquitectura Cliente / Servidor de dos capas

Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

- Cuando se requiera poco procesamiento de datos en la organización.
- Cuando se tiene una base de datos centralizada en un solo servidor.
- Cuando la base de datos es relativamente estática.
- Cuando se requiere un mantenimiento mínimo.

2.2.2.2. Arquitectura Cliente-Servidor de Tres Capas

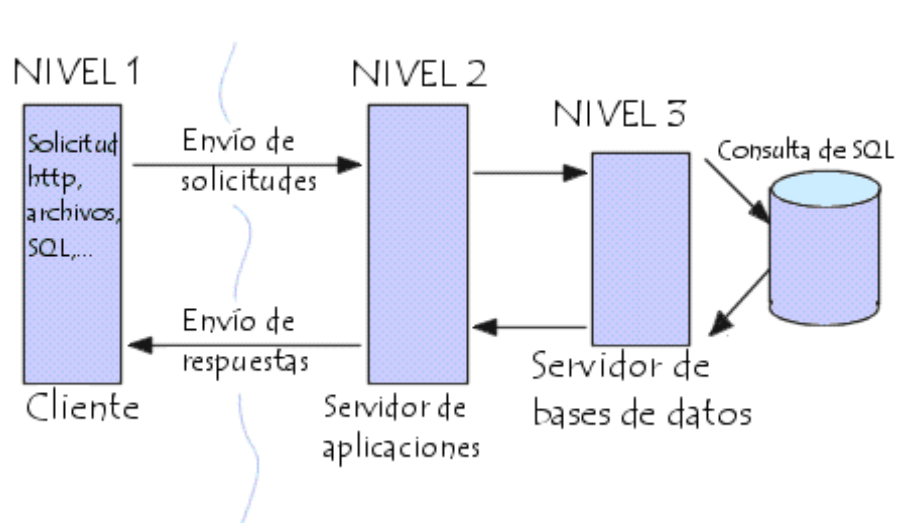


Fig. 3 Arquitectura Cliente / Servidor de tres capas

Consiste en una capa de presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

- Cuando se requiera mucho procesamiento de datos en la aplicación.
- En aplicaciones donde la funcionalidad esté en constante cambio.
- Cuando los procesos no están relativamente muy relacionados con los datos.
- Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.
- Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.
- Es muy adecuada para utilizarla con la tecnología orientada a objetos.

2.2.3. Clasificación de los sistemas Cliente/Servidor

A continuación mostramos la clasificación de los sistemas cliente/servidor de acuerdo al nivel de abstracción del servicio que ofrecen:

- **Representación distribuida:** La interacción con el usuario se realiza en el servidor, el cliente hace de pasarela entre el usuario y el servidor.
- **Representación Remota:** La lógica de la aplicación y la base de datos se encuentran en el servidor. El cliente recibe y formatea los datos para interactuar con el usuario.
- **Lógica Distribuida:** El cliente se encarga de la interacción con el usuario y de algunas funciones triviales de la aplicación. Por ejemplo controles de rango de campos, campos obligatorios, etc. Mientras que el resto de la aplicación, junto con la base de datos, están en el servidor.
- **Gestión Remota de Datos:** El cliente realiza la interacción con el usuario y ejecuta la aplicación y el servidor es quien maneja los datos.
- **Base de Datos Distribuidas:** El cliente realiza la interacción con el usuario, ejecuta la aplicación, debe conocer la topología de la red, así como la disposición y ubicación de los datos. Se delega parte de la gestión de la base de datos al cliente.
- **Cliente servidor a tres niveles:** El cliente se encarga de la interacción con el usuario, el servidor de la lógica de aplicación y la base de datos puede estar en otro servidor.

2.2.4. Arquitectura Cliente-Servidor utilizada

NIVEL 1 PRESENTACION	NIVEL 2 NEGOCIO	NIVEL 3 DATOS
Interfaz con el usuario (Navegador Web)	Servidor Web, en el cual se almacena las páginas JSP y los JavaBeans. Estos definen las operaciones propias de la lógica del negocio.	Motor de base de datos MySQL residente en el mismo servidor Web

Tabla 1 Arquitectura Utilizada

Para el desarrollo de este proyecto, se utilizó arquitectura de tres capas: de presentación, de lógica de la aplicación y de la base de datos. Para la capa de presentación se utilizó el lenguaje JSP, la parte de lógica de la aplicación se maneja con el lenguaje Java y la base de datos es manejada con el motor de bases de datos MySQL.

En este modelo se deja en el PC del usuario un procesamiento mínimo (la presentación de los datos y su validación previa), y se incorpora una capa denominada servidor de aplicaciones, donde se realizan los procesamientos complejos, y se solicitan los servicios del servidor de datos cuando es necesario acceder a la información almacenada en la base de datos.

2.3. HERRAMIENTAS DE DESARROLLO DE PÁGINAS DINÁMICAS

Las páginas estáticas presentan un problema que consiste en que no permiten la interactividad del usuario con la Web. En este sentido las páginas dinámicas son de gran utilidad ya que permiten atender a peticiones de información personalizada, o cuando queremos que el cliente nos envíe un formulario o queremos almacenar sus datos de forma automatizada en una Base de Datos.

2.3.1. Clasificación de las herramientas de desarrollo de páginas dinámicas

2.3.1.1. Client Side Scripts

Son scripts que se ejecutan en el navegador del cliente. Los Scripts Client Side al ejecutarse en el navegador, no hacen necesario que el servidor Web cumpla determinados requisitos.

Los Client Side Scripts más comunes son:

- **JavaScript.** Desarrollado por Netscape, permite, entre otras cosas, crear ventanas, mostrar texto en movimiento y verificar entradas de un formulario antes de enviarlo al servidor.
- **Controles Activex.** Tecnología Microsoft. Los usuarios de Netscape requieren de plug-ins especiales para soportarlos. Es la respuesta de Microsoft a los Applets de Java.
- **Java Applets.** Desarrollado por Sun Microsystems. Los applets son programas escritos en lenguaje de programación Java, se incrustan en el HTML y se ejecutan en el navegador gracias a la Java Virtual Machine (JVM) que lleva éste incorporado. Pueden lograr interesantes efectos para el texto, sonido y las imágenes.

2.3.1.2. Server Side Scripts

Los Scripts Server Side 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 Terminal, y el servidor muestra los resultados del programa en una página HTML que el usuario verá normalmente en su navegador.

Los más usados son:

- **ASP** (Active Server Pages). Tecnología de propiedad de Microsoft. Se utiliza a menudo para la gestión de Bases de Datos ya que puede conectarse a SQL, Access, Oracle u otras. Requiere por parte del servidor un Microsoft Web Server, el navegador es indiferente pues el trabajo se realiza del lado del Servidor.
- **PHP**. Lenguaje similar al ASP pero código abierto y gratuito. Su gran potencia se encuentra en la interacción con las bases de datos más utilizadas: Oracle, Sybase, MySQL.
- **JSP** (JavaServer Pages). Tecnología que nos permite la generación dinámica de páginas Web combinando código JAVA (scriplets) con un lenguaje de marcas como HTML o XML.

	JavaServer Pages	ActiveServer 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.
MANEJO DE TAGS PERSONALIZADOS	Se pueden utilizar librerías de Tags.	No se pueden utilizar Tags personalizados	Soporta el uso de librerías de Tags.

Tabla 2 Características Principales de Cada Tecnología

2.3.2. Herramienta utilizada

La herramienta utilizada en la implementación del sitio Web EISI es JavaServer Pages, tecnología para desarrollar aplicaciones Web independientes de la plataforma, portables a otros sistemas operativos y servidores Web. Una característica importante de JSP es que permite separar la interfaz del usuario de la generación del contenido dinámico, dando lugar a procesos de desarrollo más rápidos y eficientes. Adicionalmente, las páginas JSP pueden acceder directamente a componentes Javabeans, instanciándolos y estableciendo sus propiedades e invocando sus métodos directamente desde la página JSP. Esto

permite desarrollar aplicaciones n-capas donde se separan en lo posible los datos, la lógica del negocio y la lógica de presentación, encapsulando, generalmente en beans, el acceso a los datos.

Cada vez que un cliente solicita al servidor Web una página JSP, este pasa la petición al motor de JSP, el cual verifica si la página no se ha ejecutado antes o fue modificada después de la última compilación, tras lo cual la compila, convirtiéndola en Servlet, la ejecuta y devuelve los resultados al cliente en formato HTML.

2.3.2.1. Modelo de acceso a JSP

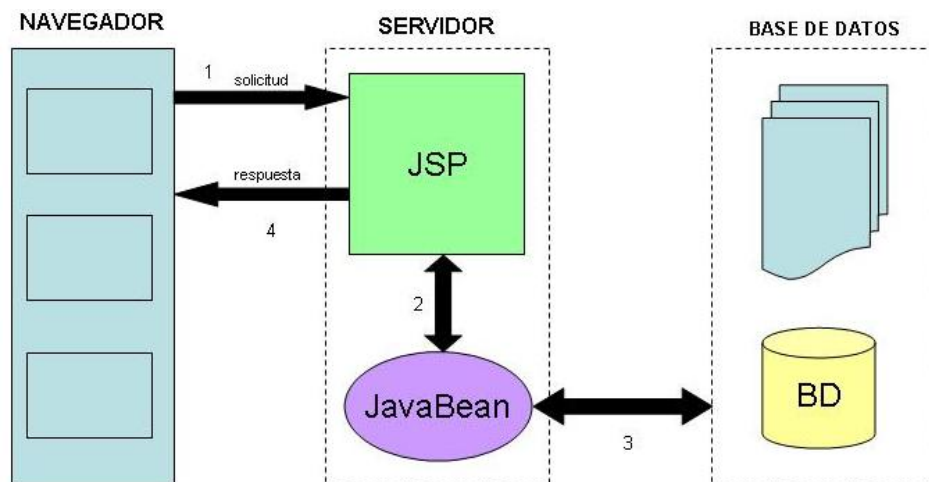


Fig. 4 Modelo de acceso a JavaServer Pages

- Un usuario trabajando en un navegador Web cliente hace una petición que es enviada a un archivo JSP. Este archivo accesa componentes del servidor que generan contenido dinámico y lo presentan en el navegador (1).
- Después de recibir la petición del cliente, el archivo JSP pide información de un Javabean (2).

- El Javabeen puede, en turnos, pedir información de un Javabeen o de una base de datos (3).
- Una vez el Javabeen genera el contenido, el archivo JSP puede consultar y presentar el contenido del bean (4).

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.

2.4. BASES DE DATOS

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

2.4.1. 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.

2.4.2. 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.

2.4.2.1. 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.

2.4.2.2. El modelo de red

El modelo de red evita redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector. La dificultad surge al manejar las conexiones o ligas entre los registros y sus correspondientes registros conectores.

2.4.2.3. 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.

2.4.3. Conectividad 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. La conectividad a bases de datos es una interfaz estándar del acceso a base de datos, que proporciona acceso uniforme a una amplia gama de bases de datos relacionales.

Para conectarse a un motor de bases de datos determinado, se necesita un driver o una interfaz estándar que medie entre la aplicación y la base de datos.

2.4.3.1. Conectores más utilizados

ODBC. Es un programa de interfase de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacionales, utilizando para ello SQL (lenguaje de consulta estructurado). Se administran a través de la ventana ODBC del *Panel de Control*.

Se pueden configurar tres diferentes fuentes de datos ODBC, la forma más interesante es la de DSN del Sistema, que presenta la ventaja de poder ser accedida por cualquier usuario, siendo el tipo usado en aplicaciones ASP.

MDB. Servidor de bases de datos casi profesional. Esta aplicación permite trabajar con tablas de base de datos creadas en Access 97/2000. Es posible abrir tablas en SQL, visualizarlas, navegar, crear y borrar índices, fijar relaciones, copiar, etc.

La aplicación requiere algún conocimiento de SQL aunque realizar las operaciones es más fácil que tener que escribirlas en SQL. Los usuarios también pueden crear, borrar y ejecutar consultas e importar tablas Foxpro.

JDBC. La conectividad de bases de datos Java (Java DataBase Connectivity, JDBC) es una especificación de la interfaz de aplicación de programa (application program interface, API) para conectar los programas escritos en Java a los datos en bases de datos populares.

2.4.4. Conectividad utilizada

Para el desarrollo del sitio Web EISI 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 demasiadas 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 driver JDBC y el 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 driver JDBC, la cadena de conexión, y el nombre de usuario y contraseña para conectarse a la base de datos son codificados dentro del programa.

2.4.5. Manejadores de bases de datos

Un Sistema Manejador de Base de Datos está constituido por un conjunto de programas que manejan todas las solicitudes formuladas por los usuarios a la base de datos. Algunos manejadores de bases de datos son: SQL, ORACLE, FOXPRO, FOXBASE, PARADOS, ACCESS y APPROACH. Un Manejador de Bases de Datos facilita las funciones de:

- Almacenar físicamente,
- Garantizar consistencia,
- Garantizar integridad,
- Atomicidad transaccional,
- Manejar vistas a la información.

2.4.5.1. 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.

2.4.5.1.1. Características de MySQL

- Su principal objetivo de diseño fue la velocidad.
- Consume muy pocos recursos, tanto de CPU como de memoria.
- MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. Una base de datos relacional que almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar.
- MySQL es Open Source. Open Source significa que la persona que quiera puede usar y modificar MySQL. Usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones.

2.4.5.1.2. Ventajas de usar MySQL

- El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar.

- El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes. Su conectividad, velocidad, y seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet.
- Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- Mejores utilidades de administración (backup, recuperación de errores, etc).
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.

2.5. PROGRAMACIÓN DE APLICACIONES

2.5.1. 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.

2.5.1.1. 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.

2.5.2. 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*.

2.5.3. Programación Orientada a Objetos (POO)

Este tipo de programación fue el 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).

2.5.3.1. 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.

2.5.3.2. 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.

2.5.3.3. 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.
- **Destruyores:** 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.

2.5.3.4. 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.

2.5.3.5. 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.

2.6. SERVIDOR WEB

¿Qué es un 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 ordenador 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.

Hay que destacar el hecho de que la palabra servidor identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término.

¿Y un servidor 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://albatros.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.

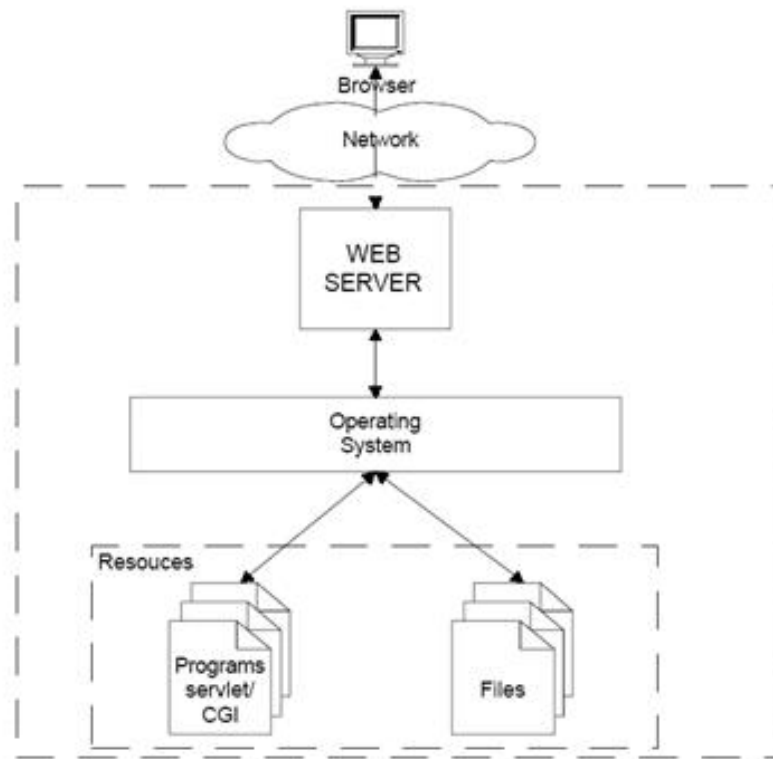


Fig. 5 Funcionamiento de un servidor Web

FUNCIONAMIENTO DE UN SERVIDOR WEB

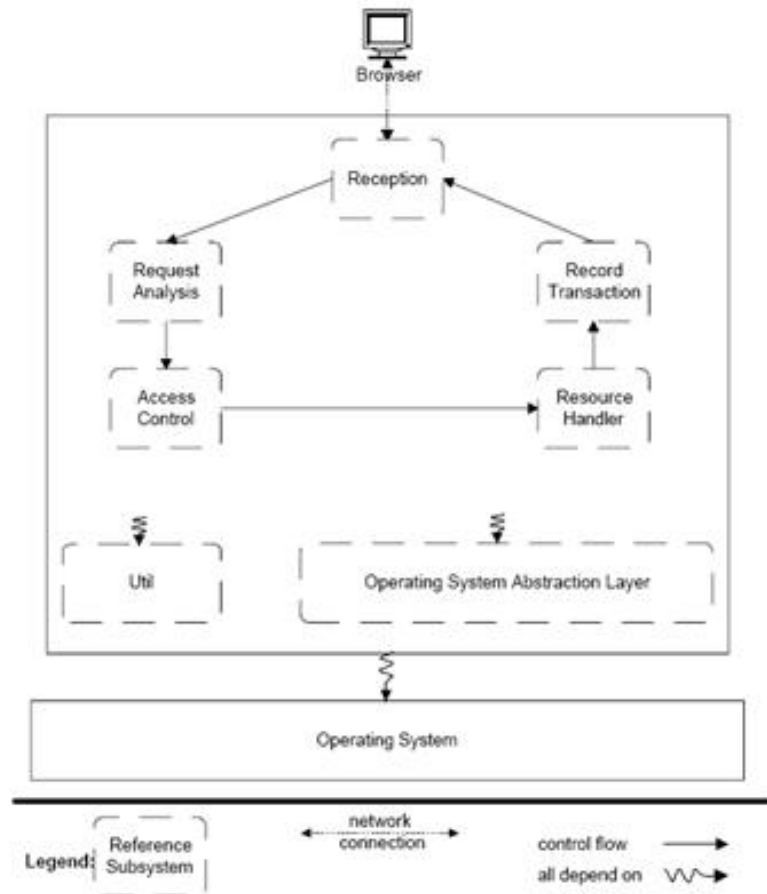


Fig. 6 Arquitectura de un Servidor Web

Vemos en el gráfico superior una arquitectura habitual de un servidor Web, dividido en dos capas:

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

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://albatros.uis.edu.co> se traduce al fichero local \$CATALINA_HOME/webapps/eisi/index.jsp.

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.

Capa soporte. Esta capa actúa como una interfase entre el sistema operativo y el servidor Web y, entre los propios subsistemas de la capa superior. 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.

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.

2.6.1. Tipos de servidores Web

- Resin, de Caucho Technologies, es un motor especialmente enfocado al servicio de páginas XML, con una licencia libre para desarrolladores.
- BEA Weblogic, es un servidor de aplicaciones de alto nivel. Está escrito en Java, y se combina con otra serie de productos, tales como Tuxedo, un servidor de bases de datos para XML.
- JRun, de Macromedia, un servidor de aplicaciones de Java, de prestaciones medias.
- Lutris Hendirá, servidor gratuito y open source. También está enfocado a servir XML, y para plataformas móviles.
- El más popular, open source, y continuamente en desarrollo, es el Tomcat , del consorcio Apache, un contenedor de servlets con muchos desarrollos adicionales alrededor; por ejemplo, Cocoon para servir páginas XML. Puede servir páginas sólo o como un añadido al servidor Apache. Es relativamente rápido, y fácil de instalar.

2.6.2. Servidor Web utilizado

Para el desarrollo y funcionamiento de los módulos del portal ECIWeb, se utilizó el servidor Tomcat en conjunto con el servidor Web Apache.

2.6.2.1. Servidor 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

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.

· 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.

Respuestas personalizadas ante errores del servidor: Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

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.

2.6.2.2. 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.

2.6.2.3. 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.

3. MARCO METODOLOGICO

Para realizar la Parametrización del Sitio EISIWeb y la implantación del sitio Web en la escuela de Ingeniería Civil, la creación de un sitio plantilla, completamente parametrizado, se propone como metodología de desarrollo el Prototipado Evolutivo.

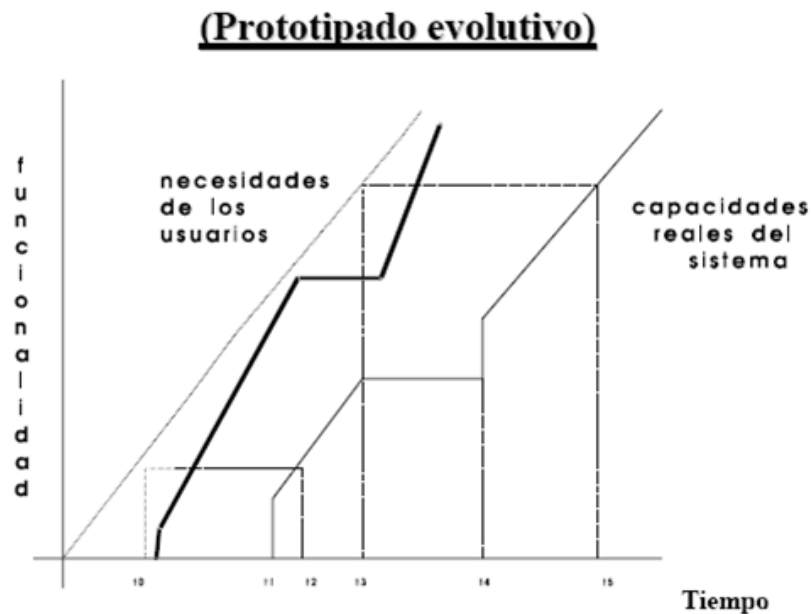


Fig. 7 Prototipado Evolutivo

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 Ingeniera 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.

Diagrama de flujo del prototipado evolutivo

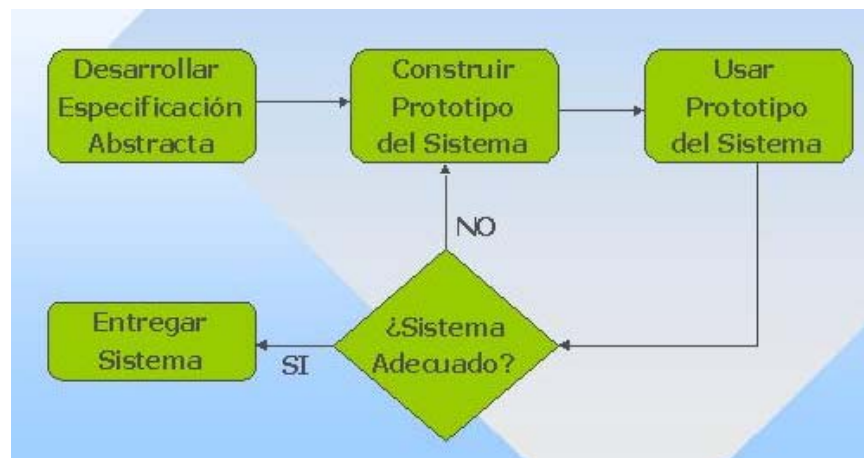


Fig. 8 Diagrama de Flujo del Prototipado Evolutivo

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.

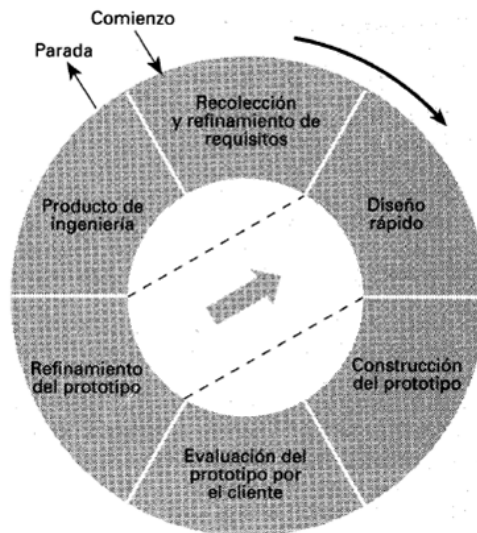


Fig. 9 Procedimiento del Prototipado Evolutivo

3.1. 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

3.2. 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

4. MANUAL DE INSTALACION Y CONFIGURACION DEL SERVIDOR WEB

En este capitulo se describirá paso a paso la instalación y configuración del Servidor Web, el proceso de instalación se realiza en forma general, pero si se quiere información mas detallada del proceso de instalación puede visitar la guía oficial de instalación de Debian en español.

<http://www.debian.org/releases/stable/i386/index.html.es>

4.1. INSTALACIÓN DEL SISTEMA OPERATIVO

A continuación se mostrara el manual de instalación de un sistema operativo Debian 4 Etch.

Se puede conseguir una distribución del sistema operativo en la siguiente URL:

http://cdimage.debian.org/debian-cd/4.0_r4a/i386/iso-cd/

Son archivos de imágenes .iso, se puede descargar el CD1, que tiene los requerimientos mínimos para instalar el sistema operativo en el equipo.

PANTALLA DE BIENVENIDA

El CD es completamente autoejecutable al iniciar el sistema, lo cual permite que al introducir el CD en el equipo e iniciarlo, automáticamente entra al programa de instalación de Linux Debian 4 Etch en el equipo. Al insertar el CD, aparece una imagen como la de la Fig. 10, para iniciar la instalación se tienen varias opciones, presionar Enter, escribir `install`, o si se quiere un entorno gráfico de instalación se escribe `installgui`.

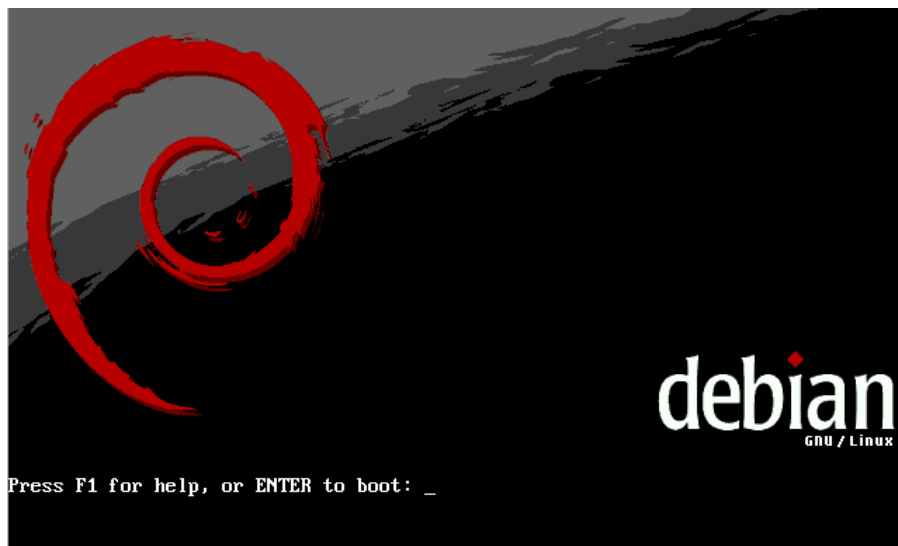


Fig. 10 Pantalla de bienvenida a la instalación

Se selecciona el lenguaje con el se va a trabajar el sistema operativo, Debian y todas las distribuciones de Linux permiten la instalación multilingüaje.

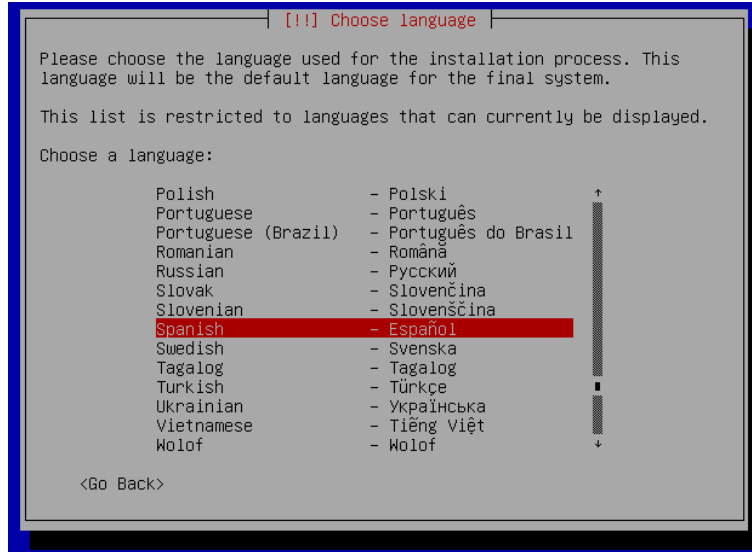


Fig. 11 Selección del lenguaje de Sistema Operativo

Se selecciona la ubicación donde se encuentra el equipo al cual se le va a instalar el sistema operativo.

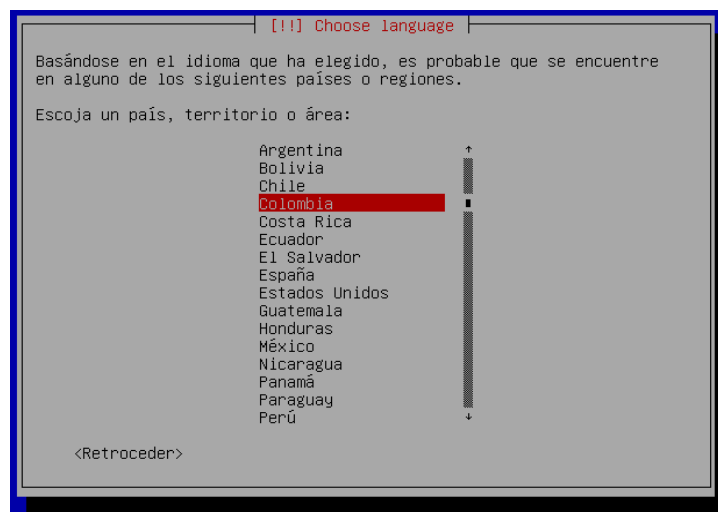


Fig. 12 Selección de la Ubicación

Se selecciona la distribución del teclado, en este caso es **español**.

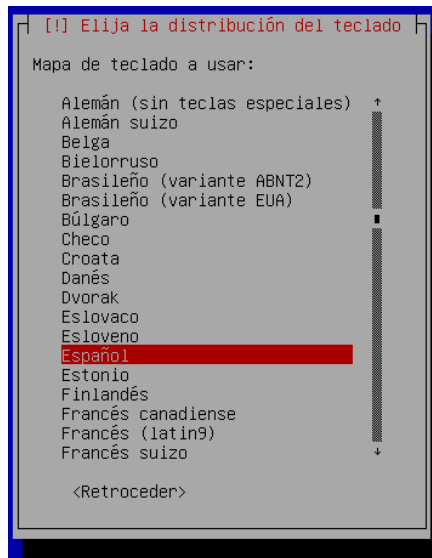


Fig. 13 Selección de distribución del Teclado

Luego de hacer la anterior selección, el instalador comienza a hacer una detección del hardware, esto con el fin de encontrar las unidades de CD-ROM que tiene el equipo, y detectar unidades adicionales.

A continuación se inicia la configuración de la red, se introduce el nombre de la máquina, esto se realiza para que el equipo pueda ser reconocido en el sistema de red en el que va a trabajar.

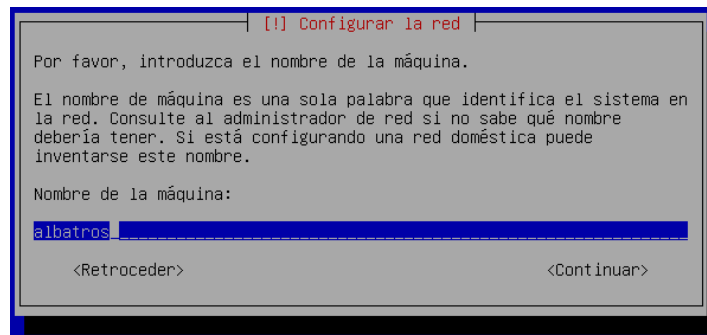


Fig. 14 Configuración de la Red 1

El nombre de dominio, para el caso de este servidor se deja en blanco.

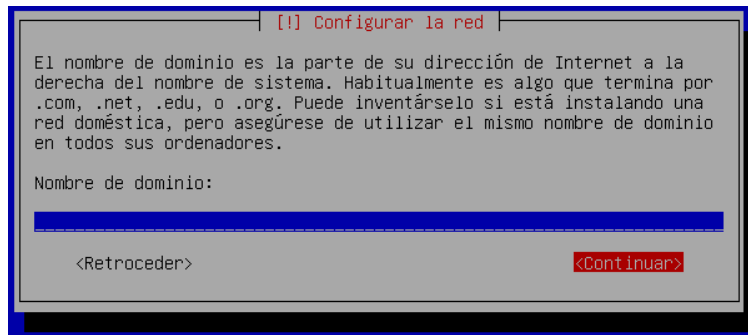


Fig. 15 Configuración de la Red 2

Esta es la parte más delicada de la instalación, en esta parte se va a escoger particionado Manual para establecer las particiones. Para información más detallada de como deberían quedar las particiones puede visitar:

<http://www.debian.org/releases/stable/i386/apc.html.es>

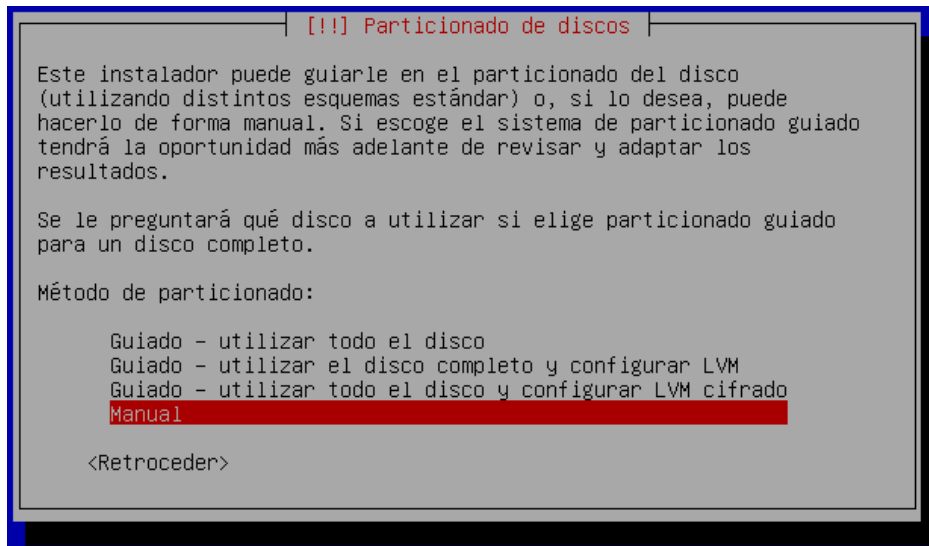


Fig. 16 Particionado de Disco 1

En esta parte se realiza las particiones que se requieran para el funcionamiento del sistema operativo. Por seguridad del servidor no se muestra el estado de particionamiento del disco duro del servidor Albatros, pero se va a realizar un ejemplo de particionamiento.

Como se puede observar, se tiene un disco duro de 8.6 GB, completamente en blanco, no se han realizado particiones anteriormente, se selecciona el espacio libre, en este caso, los 8.6 GB.

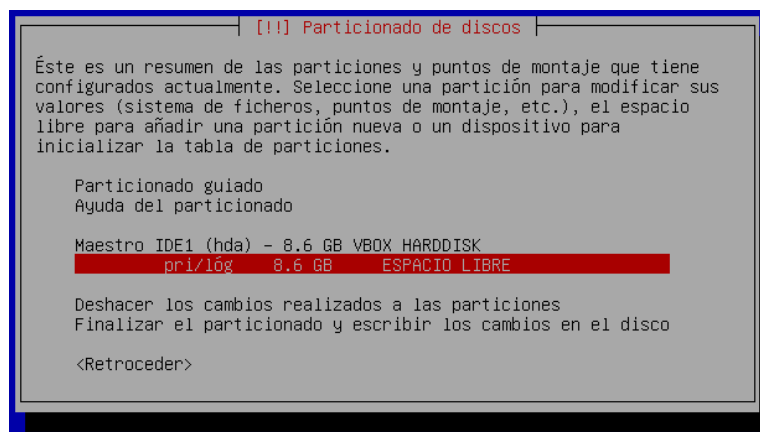


Fig. 17 Particionado de Disco 2

El sistema da la opción de particionamiento, como el proceso se esta realizando de forma manual, se escoge la opción de crear una partición nueva.

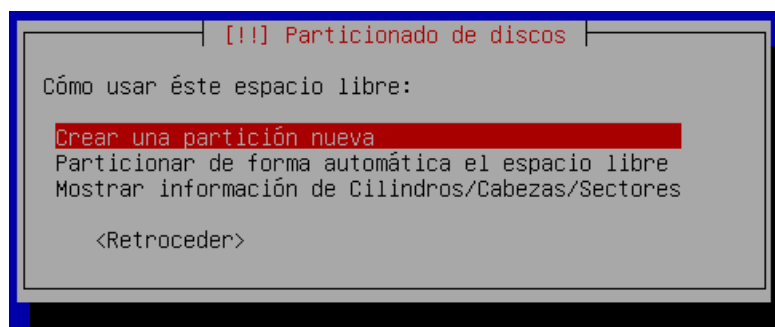


Fig. 18 Particionado de Disco 3

En este paso, se entra la cantidad de espacio de disco que se quiera para la partición, la primera partición que se realiza es la de los archivos del sistema, en este caso la de los ficheros raíz /. Como consejo, se puede crear una partición para los archivos de booteo del sistema la partición /boot, proceso que se va a realizar a continuación, para la partición de /boot se pueden asignar unos 150 MB.

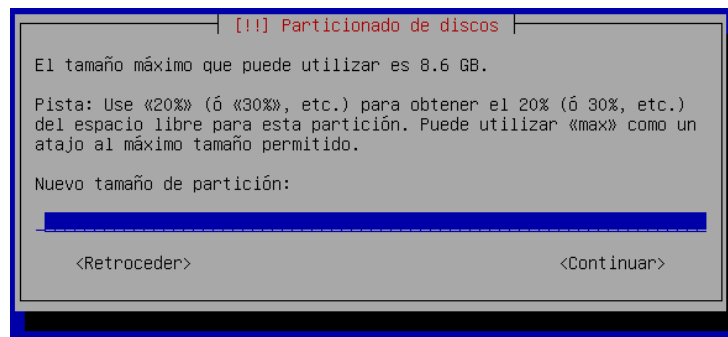


Fig. 19 Particionado de Disco 4

Debe ser de tipo Primaria

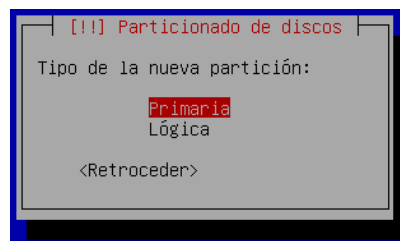


Fig. 20 Particionado de Disco 5

La partición se ubica al principio del Disco, recuerde que se esta instalando la partición de los archivos de inicio de ejecución del sistema operativo.

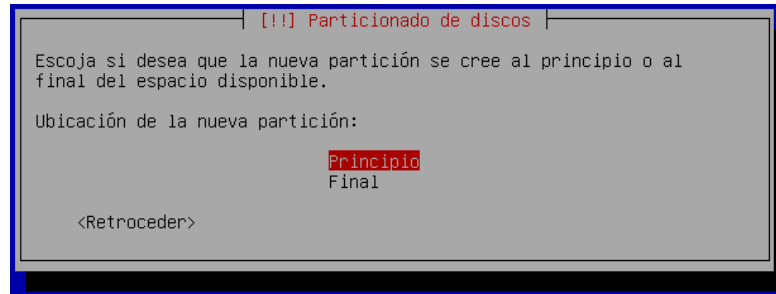


Fig. 21 Particionado de Disco 6

Se selecciona el punto de Montaje, acá se va a asignar el punto de montaje `/boot` a la partición que se acabo de definir, recuerde que en "Utilizar como:" debe ser de tipo sistema ext3 transaccional, este es el sistema de ficheros que utiliza Linux Debian.

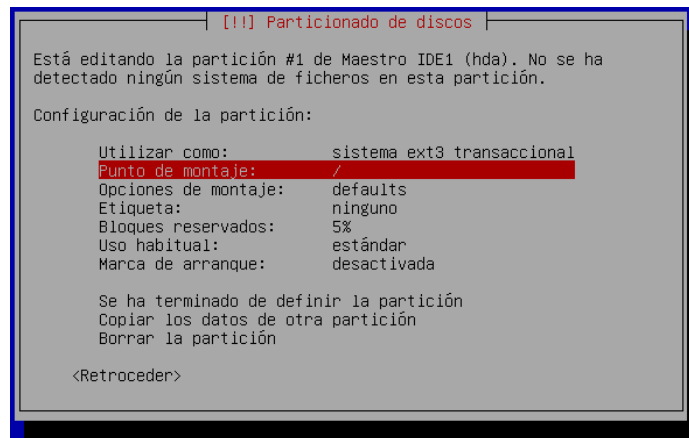


Fig. 22 Particionado de Disco 7

Se selecciona el punto de montaje, en este caso “/boot – ficheros estáticos del cargador de arranque”.

```
[!!] Particionado de discos

Punto de montaje para esta partición:

/ - sistema de ficheros raíz
/boot - ficheros estáticos del cargador de arranque
/home - directorios personales de los usuarios
/tmp - ficheros temporales
/usr - datos estáticos
/var - datos variables
/srv - datos de los servicios que ofrece el sistema
/opt - paquetes de aplicaciones añadidas
/usr/local - jerarquía local
Introducir manualmente
No montarla

<Retroceder>
```

Fig. 23 Particionado de Disco 8

Al tener ya todo listo para crear la partición se verifican que los datos estén correctos, de tipo ext3, el punto de montaje /boot, el resto de información no se modifica, se selecciona la opción de “Se ha terminado de definir la partición”

```
[!!] Particionado de discos

Está editando la partición #1 de Maestro IDE1 (hda). No se ha
detectado ningún sistema de ficheros en esta partición.

Configuración de la partición:

Utilizar como:          sistema ext3 transaccional
Punto de montaje:      /boot
Opciones de montaje:   defaults
Etiqueta:              ninguno
Bloques reservados:    5%
Uso habitual:          estándar
Marca de arranque:     desactivada

Se ha terminado de definir la partición
Copiar los datos de otra partición
Borrar la partición

<Retroceder>
```

Fig. 24 Particionado de Disco 9

Se ha creado una nueva partición, el mismo proceso de particionamiento se realiza para crear las otras particiones, tenga en cuenta que la partición que sigue después de `/boot` es la partición de ficheros raíz `/`, esta también es de tipo primaria y se coloca al principio, la cantidad de disco que puede utilizar depende del tamaño del disco.

```
[!!] Particionado de discos

Éste es un resumen de las particiones y puntos de montaje que tiene
configurados actualmente. Seleccione una partición para modificar sus
valores (sistema de ficheros, puntos de montaje, etc.), el espacio
libre para añadir una partición nueva o un dispositivo para
inicializar la tabla de particiones.

Particionado guiado
Ayuda del particionado

Maestro IDE1 (hda) - 8.6 GB VBOX HARDDISK
#1 primaria 148.0 MB f ext3 /boot
pri/lóg 8.4 GB ESPACIO LIBRE

Deshacer los cambios realizados a las particiones
Finalizar el particionado y escribir los cambios en el disco

<Retroceder>
```

Fig. 25 Particionado de Disco 10

Ahora se va a definir una partición de intercambio, la partición SWAP. El espacio de intercambio es una zona del disco que se usa para guardar las imágenes de los procesos que no han de mantenerse en memoria física, Normalmente se asigna tamaño de partición al espacio de intercambio el doble del tamaño de la RAM, es decir, si el equipo tiene un memoria RAM de 1 GB, el espacio de particionamiento para la SWAP será de 2 GB.

La mayoría de los sistemas operativos modernos poseen un mecanismo llamado memoria virtual, que permite hacer creer a los programas que tienen más memoria que la disponible realmente; por ejemplo, 4 GB en un ordenador de 32 bits. Como en realidad no se tiene físicamente toda esa memoria, algunos procesos no podrán ser ubicados en la memoria RAM.

En este caso es cuando es útil el espacio de intercambio: el sistema operativo puede buscar un proceso poco activo, y moverlo al área de intercambio (el disco duro) y de esa forma liberar la memoria principal para cargar otros procesos. Mientras no haga falta, el proceso extraído de memoria puede quedarse en el disco, ya que ahí no gasta memoria física. Cuando sea necesario, el sistema vuelve a hacer un *intercambio*, pasándolo del disco a memoria RAM. Es un proceso lento (comparado con usar sólo la memoria RAM), pero permite dar la impresión de que hay más memoria disponible.

Se selecciona “Utilizar como:”

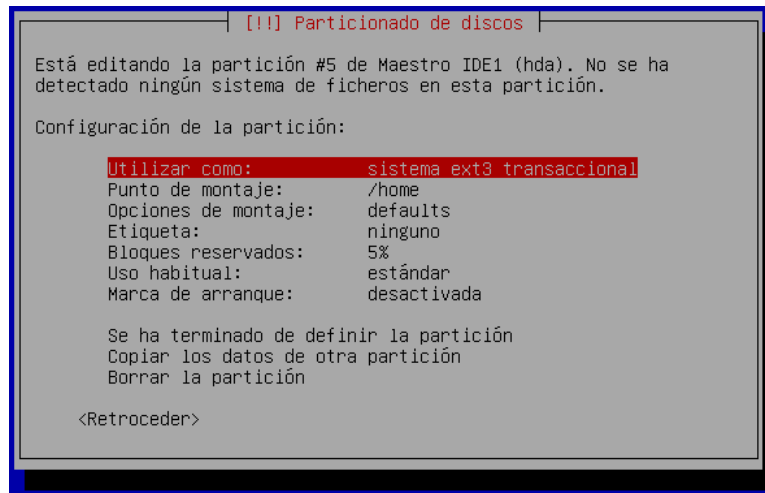


Fig. 26 Particionado de Disco 11

Se selecciona “área de intercambio”:

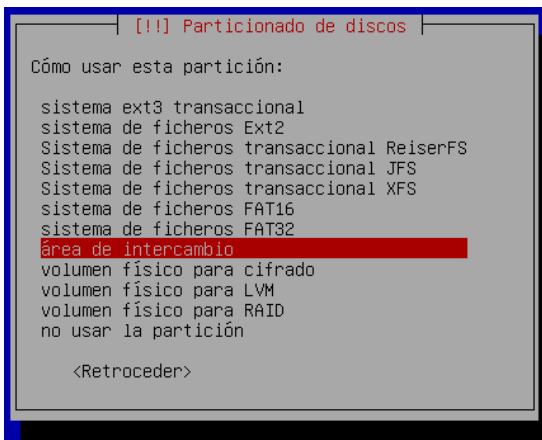


Fig. 27 Particionado de Disco 12

Se selecciona “Se ha terminado de definir la partición”, ahora se puede continuar

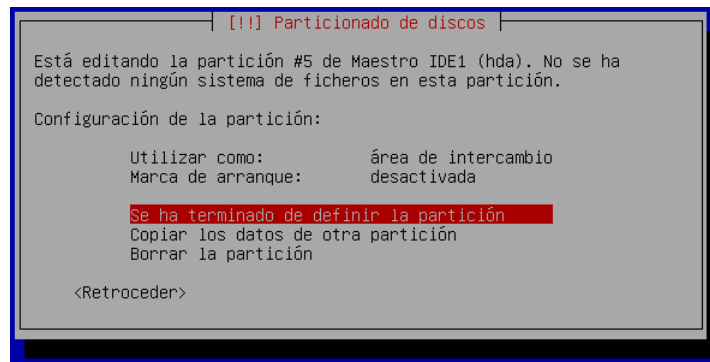


Fig. 28 Particionado de Disco 13

Se muestra un resumen con las particiones que se van a formatear, en este caso se muestran las tres particiones que se crearon, se selecciona la opción <Sí> y el programa de instalación realiza las procesos correspondientes.

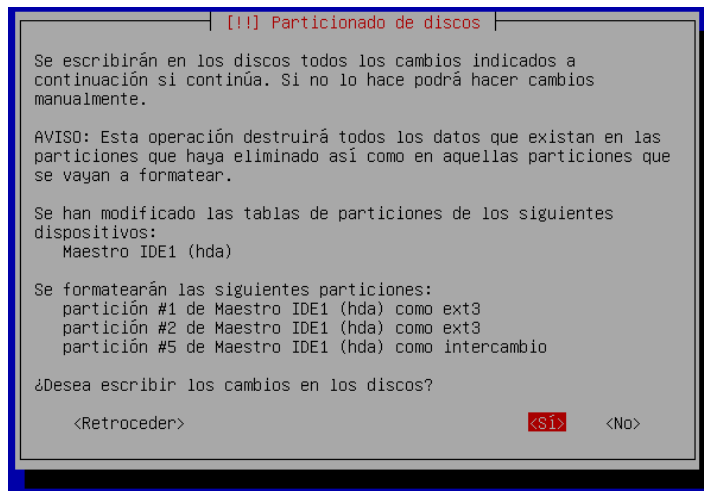


Fig. 29 Particionado de Disco 14

Una vez que se escriban los datos en el disco y el proceso sea correctamente finalizado pasamos al siguiente paso, el cual es crear una nueva contraseña para el usuario Root o Super Administrador. El usuario Root es el que tiene permisos de modificar y acceder a todo el sistema, se recomienda colocar una contraseña compleja y solo usar esta cuenta para hacer modificaciones al sistema.

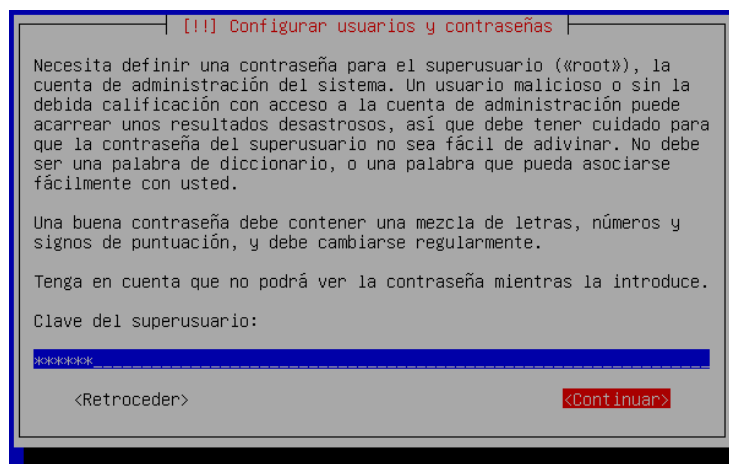


Fig. 30 Configuración de Usuarios y Contraseñas 1

En este paso se debe colocar el nombre para un usuario, como en Windows existe el usuario Administrador y no se recomienda usarlo como sesión principal, en el caso de Linux se aplica la misma técnica así que Debian obliga a crear otro usuario a parte de Root.

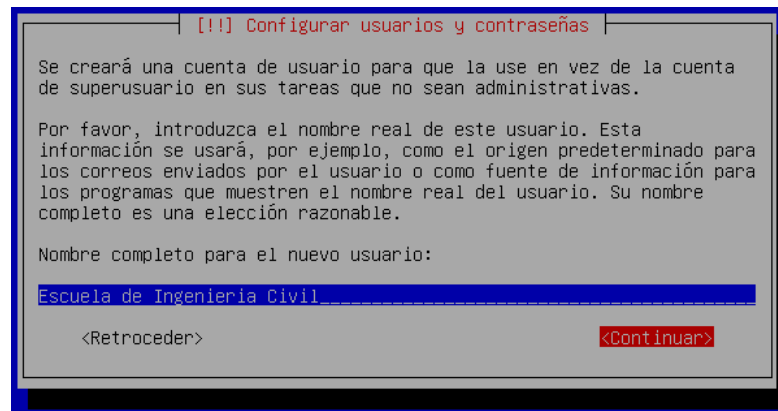


Fig. 31 Configuración de Usuarios y Contraseñas 2

Se crea el nombre corto de usuario, el que se va a usar durante las sesiones en Debian.

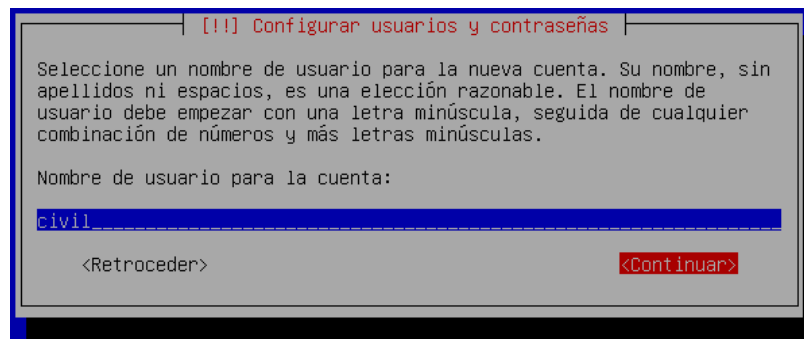


Fig. 32 Configuración de Usuarios y Contraseñas 3

Se asigna una contraseña para la cuenta de usuario de Debian, el siguiente paso es conformar la contraseña.

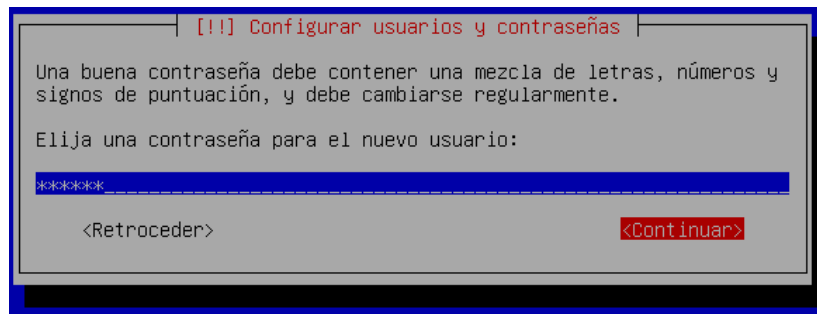


Fig. 33 Configuración de Usuarios y Contraseñas 4

Y empieza la instalación de Debian copiando los paquetes principales para iniciar el sistema, solo lo básico

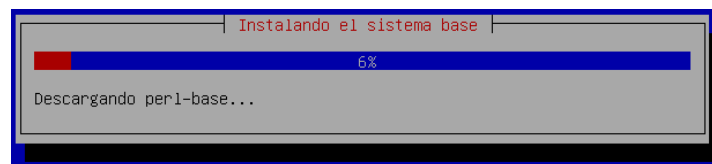


Fig. 34 Instalando el Sistema Base

Durante la instalación se preguntará si deseamos descargar una réplica de Red al sistema, a esto se responde <No>, la idea es personalizar el sistema, si selecciona <Si> se escogerán los paquetes a instalar automáticamente.

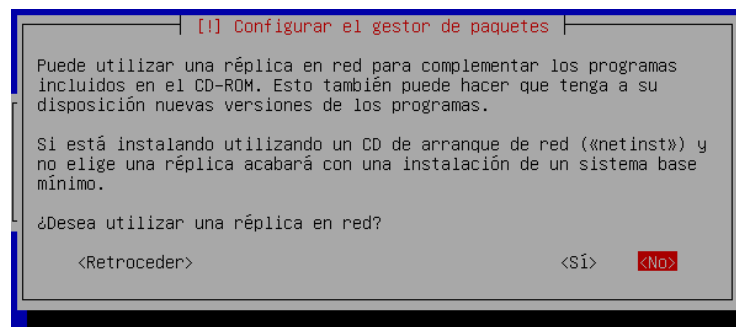


Fig. 35 Configurar el Gestor de Paquetes 1

Como no se selecciono ninguna Réplica de Red, el asistente mostrará un mensaje como el siguiente, y muestra un ruta “/etc/apt/sources.list”, este fichero se modificará más adelante, por ahora solo se selecciona <Continuar>.

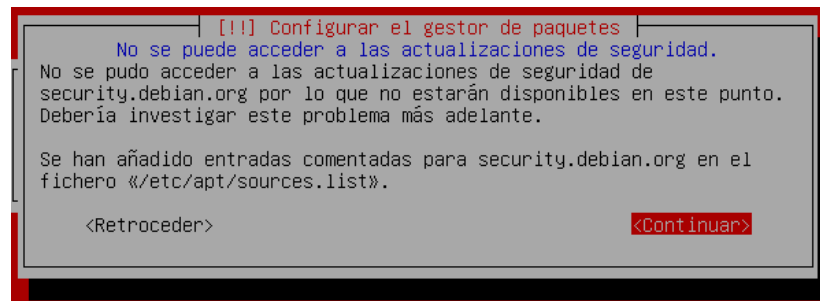


Fig. 36 Configurar el Gestor de Paquetes 2

Se esta instalando un sistema básico, por tanto se selecciona “Entorno de escritorio”, para que instale una interfaz gráfica de el sistema operativo, por defecto la opción “Sistema estándar” esta opción siempre debe estar seleccionada ya que contiene los paquetes mínimos para que el sistema funcione.



Fig. 37 Selección de Programas

Comienza la descarga de ficheros.

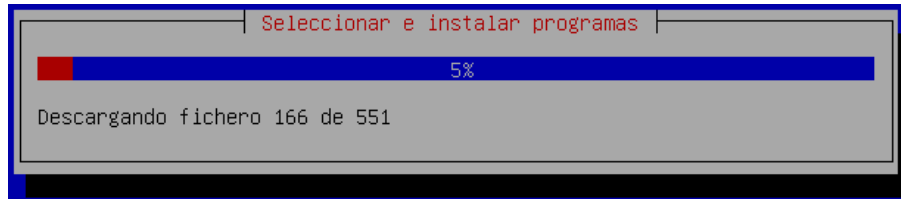


Fig. 38 Seleccionar e Instalar Programas

En este paso se pide la instalación del GRUB, lo que significa **GRand Unified Bootloader**, es un gestor de arranque múltiple que se usa comúnmente para iniciar dos o más sistemas operativos instalados en un mismo ordenador, GRUB se carga siempre que se inicia el equipo.

Para continuar con la instalación, se selecciona la opción **<Sí>**, ya que es necesario tener un GRUB instalado, en el caso de Debian, si no se tiene otro sistema instalado en el disco, permite escoger los diferentes kernel con los que cuenta el sistema operativo.

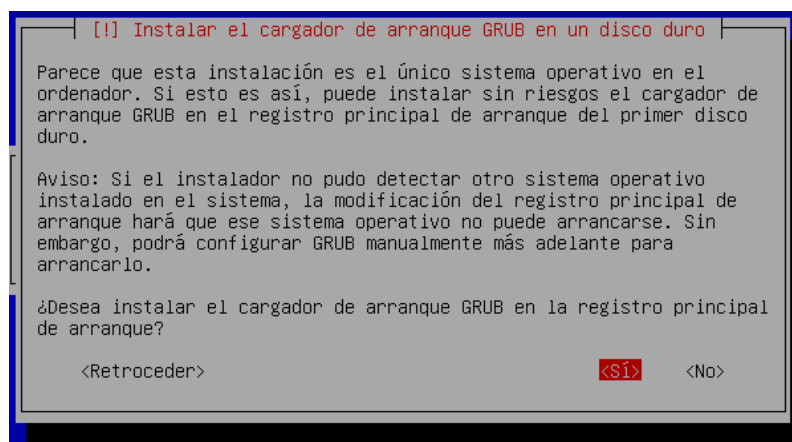


Fig. 39 Instalación del cargador de arranque GRUB

Luego de terminar el proceso de instalación, el sistema automáticamente reiniciará y entrará al nuevo sistema operativo instalado, pero antes de entrar pasa primero por el GRUB, en este caso, como solo se tiene instalado un sistema operativo, se muestra solo los registros de arranque de Debian, pero si se tuviera otro sistema operativo instalado en el equipo, se mostraría la opción con el nombre del otro sistema operativo instalado.

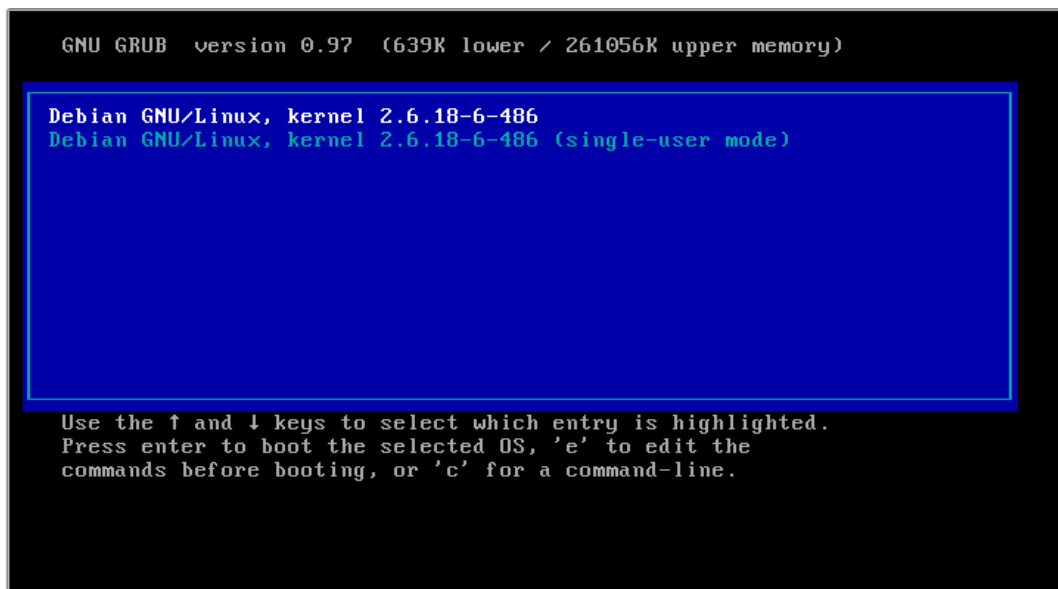


Fig. 40 Gestor de Arranque GRUB

Esta es la caja de registro de usuario, hay que tener en cuenta que el inicio de sesión se hace con el usuario que se creó durante la instalación, por defecto, Debian no permite iniciar la sesión como usuario ROOT, se escribe el nombre de usuario, a continuación se pedirá la contraseña,



Fig. 41 Pantalla de Registro de Usuario

Y listo, la instalación ha concluido y ya se está en el entorno gráfico gnome de Linux Debian 4, ya está listo para iniciar la configuración del sistema.

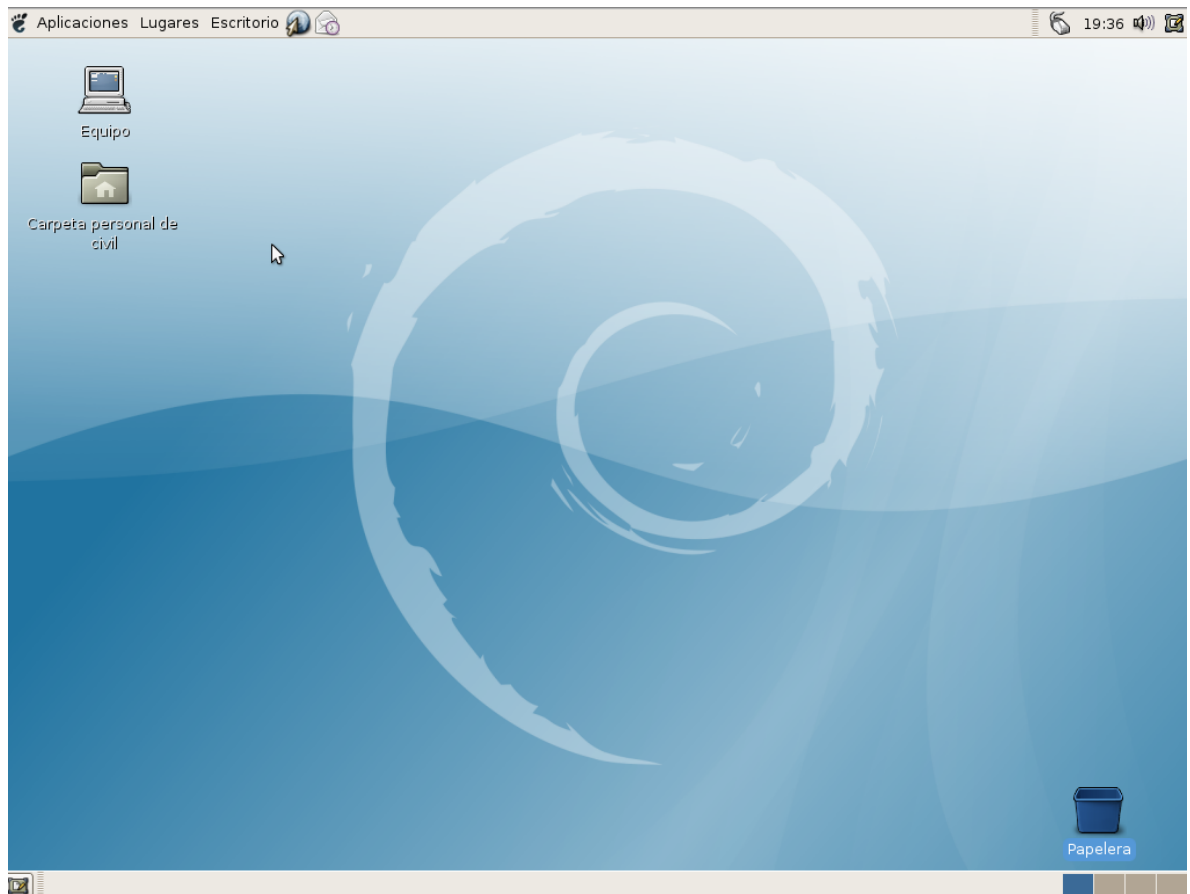


Fig. 42 Entorno Gráfico Linux Debian 4

4.2. CONFIGURACION DEL SISTEMA OPERATIVO

La configuración del sistema operativo se va a realizar toda en modo consola, esto para ayudar al usuario a que en caso de algún problema con el entorno gráfico, el cual es muy común, pueda solucionarlos desde el entorno de consola de Linux Debian.

Para esto se expande el Menú Aplicaciones → Accesorios → Terminal, por defecto la sesión se abre con el usuario, se procede a iniciar la sesión como superusuario o root.

: ~\$ su ← al presionar enter, se solicitará la contraseña de ROOT.

Contraseña: ← Se introduce la contraseña de ROOT.

← Prompt de usuario ROOT, estamos en la sesión como usuario ROOT.

4.2.1. CONFIGURACION DE LA RED

Para la configuración de la red se van a editar dos archivos, el archivo interfaces, que es el que va a contener la información de la interfase de red por la cual se va a conectar a la red el servidor, y el archivo resolv.conf para añadir los dominios que proveen el servicio de internet

Edición del archivo interfaces:

Para editar el archivo de interfaces, tecleamos por consola:

```
# vim /etc/network/interfaces
```

Se añaden las siguientes líneas, por defecto si el servidor solo tiene una tarjeta de red el sistema la va reconocer como eth0, en el caso de que tenga varias, el sistema las va a reconocer como eth0, eth1, eth2...

Para este caso se tiene solo una tarjeta de red, por tanto se añaden las siguientes líneas:

```
auto lo
iface lo inet loopback

iface eth0 inet static
#direccion IP
address xxx.xxx.xxx.xxx
#Mascara de Red, en la mayoría de los casos es:
netmask 255.255.255.0
#Puerta de enlace predeterminada
gateway xxx.xxx.xxx.xxx

auto eth0
```

Se asume que se conocen los anteriores datos suministrados, la IP, la Máscara de Red y la puerta de Enlace predeterminada.

Edición del archivo interfaces:

Para editar el archivo de interfaces, tecleamos por consola:

```
# vim /etc/resolv.conf
```

Se debe tener conocimiento de las direcciones IP que proveen los servicios de Internet, puede haber varios dominios, por tanto se añaden las siguientes líneas:

```
#nombre dominio 1
nameserver xxx.xxx.xxx.xxx
#nombre dominio 2
nameserver xxx.xxx.xxx.xxx
#...
#...
#... asi sucesivamente

#domain nombre.dominio
```

Si se tiene conocimiento del nombre del dominio, se descomenta la línea de domain y se coloca el dominio nombre de dominio, la mayoría de las veces son terminados en .com, .net, .edu, .org.

Terminados de editar esot archivos se reicinia la red:

```
# /etc/init.d/networking restart
```

Se verifica que se haya tomado bien la configuración de la red haciendo ping a una dirección cualquiera, en este caso:

```
# ping www.google.com
```

```
PING www.l.google.com (74.125.45.103) 56(84) bytes of data.
64 bytes from yx-in-f103.google.com (74.125.45.103): icmp_seq=1 ttl=241 time=129 ms
64 bytes from yx-in-f103.google.com (74.125.45.103): icmp_seq=2 ttl=241 time=98.9 ms
64 bytes from yx-in-f103.google.com (74.125.45.103): icmp_seq=3 ttl=241 time=109 ms
64 bytes from yx-in-f103.google.com (74.125.45.103): icmp_seq=4 ttl=241 time=99.6 ms
64 bytes from yx-in-f103.google.com (74.125.45.103): icmp_seq=5 ttl=241 time=99.5 ms
```

```
--- www.l.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 8150ms
rtt min/avg/max/mdev = 98.901/107.546/129.820/11.847 ms
```

Hasta acá todo va bien, ya se tiene acceso a la red.

4.2.2. CONFIGURACIÓN DE LAS RÉPLICAS DE RED

Durante el proceso de instalación se tomó la opción de no seleccionar ninguna réplica de red por tanto ahora se van a configurar estos repositorios, recordemos que las replicas de red son los sitios donde se encuentran alojados los paquetes de actualizaciones y seguridad para esta distribución de Linux.

Editamos el archivo `sources.list`, el cual contiene la lista de repositorios a los cuales se va a conectar el sistema operativo para actualizarse. Por consola se teclea

```
# vim /etc/apt/sources.list
```

Al editar el archivo por primera vez, se debe ver algo muy parecido al contenido que se muestra a continuación, las líneas que no aparezcan, se añaden, tenga en cuenta que se debe comentar la línea donde el archivo llama los repositorios del CD de instalación:

```
#
# deb cdrom:[Debian GNU/Linux 4.0 r0 _Etch_ - Oficial i386 DVD Binary-1 20070407-11:40]/ etch
contrib main

#deb cdrom:[Debian GNU/Linux 4.0 r0 _Etch_ - Oficial i386 DVD Binary-1 20070407-11:40]/ etch
contrib main ← Esta línea se comenta para no tener que estar introduciendo el CD de instalación
cada vez que se vaya a realizar una actualización.

#REPOSITORIOS DE PAQUETES DE ACTUALIZACIONES
deb http://ftp.us.debian.org/debian/ etch main contrib non-free
deb-src http://ftp.us.debian.org/debian/ etch main contrib non-free

#REPOSITORIOS DE PAQUETES DE SEGURIDAD
deb http://security.debian.org/ etch/updates main contrib non-free
deb-src http://security.debian.org/ etch/updates main contrib non-free
```

Una vez editado y guardado el archivo se procede a actualizar la el sistema operativo:

`# aptitude update` ← Guarda en cache la lista de paquetes nuevos.

`# aptitude upgrade` ← Instala los nuevos paquetes.

`# aptitude dist-upgrade` ← Actualiza la distribución.

Como es de saberse, las replicas de red con las que se están trabajando son para la distribución estable de Linux Debian 4 etch, pero en cualquier momento se puede cambiar de distribución, a (testing o inestable), este proceso se puede realizar editando el fichero anterior y cambiando los repositorios.

Después de realizar las actualizaciones y tener Debian a punto se procede a instalar los programas para que Debian funcione como servidor Web.

4.3. CONFIGURACION DEL EQUIPO COMO SERVIDOR WEB

4.3.1. INSTALACION DE PAQUETES NECESARIOS

En este caso, se va a instalar y configurar los paquetes que interesan para este proyecto, Java, Servidor Apache, Servidor Tomcat, Modulo de conexión JK, MySQL, afortunadamente estos paquetes se encuentran en los repositorios de Debian en sus versiones estables.

Los paquetes que se van a instalar se encuentran con los siguientes nombres en los repositorios

Nombre de Programa	Nombre de Paquete	Descripción
Java 1.5	sun-java5-jdk	Software que provee herramientas de desarrollo para la creación de programas en Java.
Servidor Apache	apache2	Servidor HTTP de código abierto para plataformas que implementan el protocolo HTTP/1.1 y la noción de sitio virtual.
Servidor Tomcat	tomcat5.5	Servidor web con soporte de servlets y JSPs
Conector JK	libapache2-mod-jk	Conector de Tomcat con Apache
MySQL	mysql-server-5.0	Servidor de Bases de Datos
Otros Programas para instalar		
NTP	ntp	Sincroniza la hora y la fecha del equipo con un servidor de hora y fecha.
NMAP	nmap	Hace un mapeo de los puertos abiertos en la maquina local.
SSH	ssh	Protocolo seguro para conexiones remotas, permite conectarse a otros equipos y que otros equipos se conecten al local.
Firewall	arno-iptables-firewall	Firewall para bloquear algunos puertos.

Tabla 3 Programas para instalación en el servidor

Para instalar los paquetes se puede hacer todo en un solo comando en una consola de root, excepto el firewall, ya que este necesita una configuración, al final de la instalación, por tanto el comando para instalar todo a la vez sería:

```
# aptitude install sun-java5-jdk apache2 tomcat5.5 libapache2-mod-jk mysql-server-5.0
```

```
# aptitude install ntp nmap ssh
```

4.3.2. PRUEBAS DE FUNCIONAMIENTO DE PAQUETES INSTALADOS

Ahora se va a comprobar que los paquetes instalados estén funcionando bien.

- **Servidor Apache:** Para comprobar el funcionamiento del servidor apache se puede realizar de dos maneras:

- Con el navegador w3m por consola:

```
# w3m localhost
```

Aparecerá algo como esto



- En un navegador cualquiera, por defecto Debian trae el Epiphany e Iceweasel, en la barra de navegación se escribe `localhost` y debe aparecer algo como esto:



Fig. 43 Apache corriendo en el Navegador

- **Servidor Tomcat 5.5:** Por defecto, al instalar Tomcat desde los repositorios de Debian, se instala y habilita el puerto 8180 para enviar las peticiones JSP, por tanto, para comprobar el funcionamiento del servidor tomcat, basta con escribir en un navegador: `localhost:8180`, y listo, si el servidor está corriendo, aparecerá en blanco el contenido del navegador, esto sucede porque no se instalaron los webapps de tomcat, si se hubiesen instalado este complemento, al escribir `localhost:8180`, en el navegador aparecería la página de bienvenida de tomcat.
- **Java 1.5:** Para comprobar que java esté funcionando correctamente, se escribe por consola:

```
# java -version
```

Aparece esto:

```
java version "1.5.0_15"  
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_15-b04)  
Java HotSpot(TM) Client VM (build 1.5.0_15-b04, mixed mode, sharing)
```

En ocasiones, después de instalar java, puede quedar por defecto seleccionado otro proveedor de java, por ejemplo:

```
# java -version
```

```
java version "1.4.2" gij (GNU libgcj) version 4.1.2 20061115 (prerelease)
(Debian 4.1.1-20)
```

Copyright (C) 2006 Free Software Foundation, Inc. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

En caso de que no este seleccionado este proveedor de java, se utiliza el comando:

```
# update-alternatives --config java
```

Hay 3 alternativas que proveen `java`.

Selección Alternativa

```
-----
      1      /usr/lib/jvm/java-1.5.0-sun/jre/bin/java
      2      /usr/bin/gij-wrapper-4.1
*+    3      /usr/lib/jvm/java-gcj/jre/bin/java
```

Pulse <Intro> para mantener el valor por omisión [*] o pulse un número de selección:

Se selecciona el valor 1, que es el proveedor de java que se necesita para este caso, al seleccionar 1 y presionar enter, aparecerá algo como esto:

Se utiliza `/usr/lib/jvm/java-1.5.0-sun/jre/bin/java` para proveer `java`.

Este mismo procedimiento se utiliza para establecer el compilador de java, javac, por tanto se realizan los mismos pasos anteriores, solo que esta vez se realiza la búsqueda con javac.

```
# javac -version
```

Realizar la comprobación del proveedor de javac es importante si se van a compilar archivos java en el servidor.

- **Conector JK:** El Conector JK, es un módulo que se instala para el trabajo conjunto de apache y tomcat, por tanto, para comprobar el funcionamiento del mismo se escribe en consola:

```
# a2enmod jk
```

Obteniéndose como resultado lo siguiente:

```
This module is already enabled!
```

- **MySQL:** Para comprobar el funcionamiento del servidor de bases de datos basta escribir por consola `mysql`. Cuando se instala mysql, por defecto, no contiene ninguna contraseña de root y ningun usuario creado, este proceso se realiza mas adelante en la configuración de los paquetes.

```
# mysql
```

Aparece algo como esto:

```
# mysql
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 6  
Server version: 5.0.32-Debian_7etch6-log Debian etch distribution
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql>
```

- **NTP:** La única prueba de funcionamiento que se realiza al paquete ntp, es corroborar que la fecha y la hora del servidor coincidan con las actuales.
- **NMAP:** Para comprobar el funcionamiento de nmap, se realiza un mapeo a los puertos del equipo local:

```
# nmap localhost
```

Aparecerá algo como esto:

```
Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2008-10-13 20:05
Interesting ports on localhost (127.0.0.1):
Not shown: 1672 closed ports
```

PORT	STATE	SERVICE
22/tcp	open	ssh
25/tcp	open	smtp
80/tcp	open	http
111/tcp	open	rpcbind
113/tcp	open	auth
631/tcp	open	ipp
3306/tcp	open	mysql
8009/tcp	open	ajp13

```
Nmap finished: 1 IP address (1 host up) scanned in 0.212 seconds
```

- **SSH:** Se puede realizar una conexión remota a cualquier otro equipo con sistema operativo linux, escribir por consola:

```
# ssh usuario@dominio.usuario
```

```
usuario@dominio.usuario's password: *****
```

4.3.3. CONFIGURACION DE LOS PAQUETES INSTALADOS

Una vez instalados y hecha la comprobación de que los servicios están funcionando correctamente, se realiza la configuración de cada uno de los paquetes, esta parte es la principal ya que se muestra paso a paso como se establece que el servidor pueda funcionar correctamente:

- **JAVA 1.5:** Al instalar java desde los repositorios de Debian, por defecto java queda instalado en `/usr/lib/jvm/`, se siguen los siguientes pasos:

- **Crear enlace simbólico**, el enlace simbólico permite tener un acceso más rápido a la carpeta donde se encuentra instalado JAVA.

```
# cd /usr/lib/jvm
```

```
:/usr/lib/jvm# ln -s java-1.5.0-sun-1.5.0.14/ jdk
```

Se hace un listado de los archivos que contiene la carpeta, y debe aparecer una línea mostrando el enlace simbólico

```
:/usr/lib/jvm# ls -l
```

```
lrwxrwxrwx 1 root root 24 2008-10-13 20:33 jdk -> java-1.5.0-sun-1.5.0.14/
```

Por ahora esa es la configuración básica de JAVA, más adelante se definirán las variables de entorno.

- **SERVIDOR APACHE:** Por defecto, los archivos de configuración de apache, se instalan en [/etc/apache2](#), para configurar hay que editar algunos archivos:

- **apache2.conf:** Se edita este archivo para agregarle el nombre del servidor.

```
# cd /etc/apache2
```

```
:/etc/apache2# vim apache2.conf
```

Se busca la línea donde dice `ServerRoot "/etc/apache2"` e inmediatamente después agregue lo siguiente `ServerName dominio.servidor`

```
#
# Do NOT add a slash at the end of the directory path.
#
ServerRoot "/etc/apache2"
ServerName dominio.servidor
#Si esta instalando para un servidor local puededejarlo
#ServerName localhost
#
```

- **Hosts:** Se edita el archivo hosts, ubicado en la carpeta `/etc`: para agregar la ip del servidor, y el nombre del dominio, si se esta realizando la configuración para un servidor local, no se modifica el archivo.

Debe aparecer algo como lo siguiente:

```
127.0.0.1    localhost
127.0.1.1    albatros
#Se agrega la siguiente linea con la IP del servidor y el nombre del
dominio, el mismo que se añadio en apache2.conf
xxx.xxx.xxx.xxx dominio.servidor

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

- **SERVIDOR TOMCAT 5.5:** La configuración del servidor Tomcat se hace realizando un enlace simbólico a la ubicación donde se encuentra la carpeta weapps, la cual va a contener las aplicaciones que se quieren correr en el servidor, por defecto tomcat queda instalado, algunas carpetas

en `/var/lib/tomcat5.5` y otras en `/usr/share/tomcat5.5` , por tanto necesitamos reunir todo en un solo lugar sin mover ninguna carpeta, para esto se crea un enlace simbólico a la carpeta webapps en `/usr/share/tomcat5.5`:

```
# cd /usr/share/tomcat5.5
```

```
: /usr/share/tomcat5.5# ln -s /var/lib/tomcat5.5/webapps/ webapps
```

```
: /usr/share/tomcat5.5# ls -l
```

Se debe encontrar una línea como esta:

```
lrwxrwxrwx 1 root root 27 2008-10-13 18:15 webapps -> /var/lib/tomcat5.5/webapps
```

Por ahora esta es la configuración del servidor tomcat 5.5, mas adelante se hará la configuración de las variables de entorno que implican al servidor Tomcat.

- **CONECTOR JK:** Esta es la configuración más importante de todas, ya que con este módulo se va a hacer que las aplicaciones JSP funcionen por el puerto 80 y puedan ser accesibles desde redes externas. Por defecto, las carpetas se instalan en `/etc/libapache2-mod-jk/`, en esta carpeta se edita un archivo llamado `workers.properties`.
 - **Workers.properties:** Este archivo contiene el worker, un worker Tomcat es un ejemplar Tomcat que está esperando para ejecutar servlets por cuenta de algún servidor web. Por ejemplo, podemos tener un servidor web como Apache reenviando peticiones servlets a un proceso Tomcat

(el worker que se ejecuta detrás de él). En este archivo se van a modificar algunas líneas:

```
# primera linea que se modifica
workers.tomcat_home=$CATALINA_HOME

#Se Busca esta otra linea y se modifica como está acá:
workers.java_home=$JAVA_HOME

#ahora se busca esta linea, este es el worker con el que se esta trabajando
# Si se quiere se puede cambiar el nombre, pero se debe cambiar en todas
las otras propiedades que esta usando el worker.

worker.list=ajp13_worker

#Por ahora se va a dejar con ese nombre
```

Más adelante se dará una explicación de lo que significa \$CATALINA_HOME y \$JAVA_HOME, variables de entorno.

- **jk.conf:** Este archivo, no existe, toca crearlo manualmente, en este archivo se ubican las aplicaciones que van a trabajar con el worker definido anteriormente. Se va a crear el archivo:

```
# cd /etc/apache2/mods-available
:/etc/apache2/mods-available#
```

Se verifica que dentro de esta carpeta se encuentre el archivo jk.load y que contenga la siguiente información:

```
LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so
```

```
:/etc/apache2/mods-available# vim jk.conf
```

Se debe recordar que el archivo no existe, se edita el archivo y se agregan las siguientes líneas:

```
#<IfModule !mod_jk.c>
    loadModule jk_module /usr/lib/apache2/modules/mod_jk.so
#</IfModule>

# Where to find workers.properties
JkWorkersFile /etc/libapache2-mod-jk/workers.properties

# Where to put jk logs
JkLogFile /var/log/apache2/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
#JkLogStampFormat ~@~[[%a %b %d %H:%M:%S %Y] ~@~]

# JkOptions indicate to send SSL KEY SIZE,
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat set the request format
#JkRequestLogFormat ~@~\%w %V %T~@~]

# Send servlet for context /jsp-examples to worker named ajp13_worker
JkMount /*.jsp ajp13_worker

# Send JSPs for context /jsp-examples/* to worker named ajp13_worker
JkMount /*.jsp/* ajp13_worker

JkMount /prueba ajp13_worker
JkMount /prueba/* ajp13_worker
```

- **Creación de Enlace simbólico a jk.conf:** Para que el worker funcione debe estar habilitado en los módulos disponibles, por tanto se crea un enlace simbólico al archivo jk.conf dentro de la carpeta mods-enabled ubicada dentro de el directorio de apache:

```
# cd /etc/apache2/mods-enabled
```

```
:/etc/apache2/mods-enabled# ls -l
```

Al listar los archivos, debe aparecer una línea con el enlace simbólico a jk.load:

```
jk.load -> ../mods-available/jk.load
```

Ahora se crea el enlace simbólico a jk.conf

```
:/etc/apache2/mods-enabled# ln -s ../mods-available/jk.conf jk.conf
```

Al listar los archivos aparecerán los dos enlaces simbólicos seguidos:

```
jk.conf -> ../mods-available/jk.conf
```

```
jk.load -> ../mods-available/jk.load
```

Listo, hasta acá está configurado el módulo JK, esta era la parte más complicada de la configuración.

- **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:

Se va a crear contraseña para el usuario root:

```
# mysql
```

```
mysql> use mysql;
```

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

```
mysql> exit
```

Para que los efectos surtan efecto es necesario reiniciar mysql.

```
# /etc/init.d/mysql restart
```

Para volver a entrar a mysql ya se pide confirmación de la contraseña:

```
# mysql -u root -p  
Enter password:xxxxxx
```

```
mysql>
```

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:

En el prompt de MySQL como root:

```
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 "xxxxx".

Esta es la configuración básica para MySQL.

- **Creación de Variables de entorno:** Una variable de entorno es un valor dinámico cargado en la memoria, que puede ser utilizado por varios procesos que funcionan simultáneamente, en este caso se van a editar dos archivos para agregar las variables de entorno:

```
# /etc/profile
```

Se agrega lo siguiente, al final del archivo:

```
export JAVA_HOME=/usr/lib/jvm/jdk
export CATALINA_HOME=/usr/share/tomcat5.5
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export JRE_HOME=$JAVA_HOME/jre
export BASEDIR=$CATALINA_HOME
```

```
# vim /etc/bash.bashrc
```

Se agrega la misma información agregada al archivo profile.

Después de realizar el proceso de configuración de los paquetes y de agregar las variables de entorno, es necesario reiniciar el equipo para que todos los cambios a las configuraciones surtan efecto y se tomen las variables de entorno.

4.3.4. INSTALACION Y CONFIGURACION 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.

Para el caso del servidor que se esta configurando, se utilizará el arno-iptables, este se encuentra en los repositorios de Debian, el proceso para la instalación y configuración es el siguiente:

En una consola:

```
# aptitude install arno-iptables-firewall
```

El proceso de instalación es automático, los repositorios de Debian realizan la instalación, luego de terminar la instalación, el programa de configuración inicia automáticamente:

El asistente de configuración pregunta si se quiere crear el administrador de firewall para que sea reconfigurado mas adelante, ya que los repositorios del firewall pueden variar, se selecciona <Si>

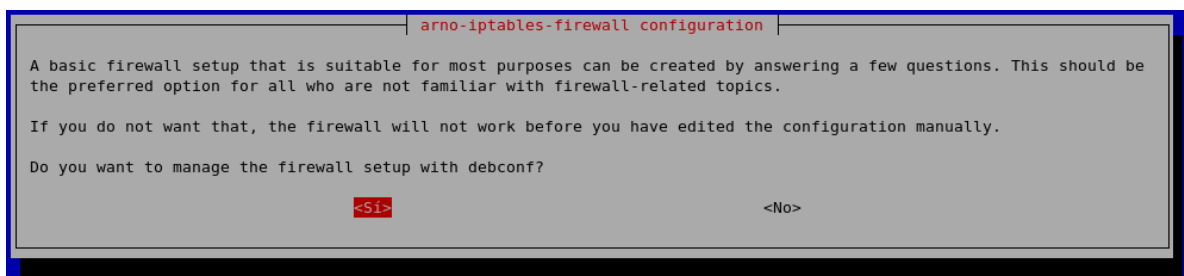


Fig. 44 Configuración de Firewall 1

El asistente de configuración pide las interfaces a las cuales se les van a aplicar las reglas del firewall, en este caso se escribe `eth0` ya que es la única tarjeta de red por la cual se tiene acceso a la red y se selecciona <Aceptar>

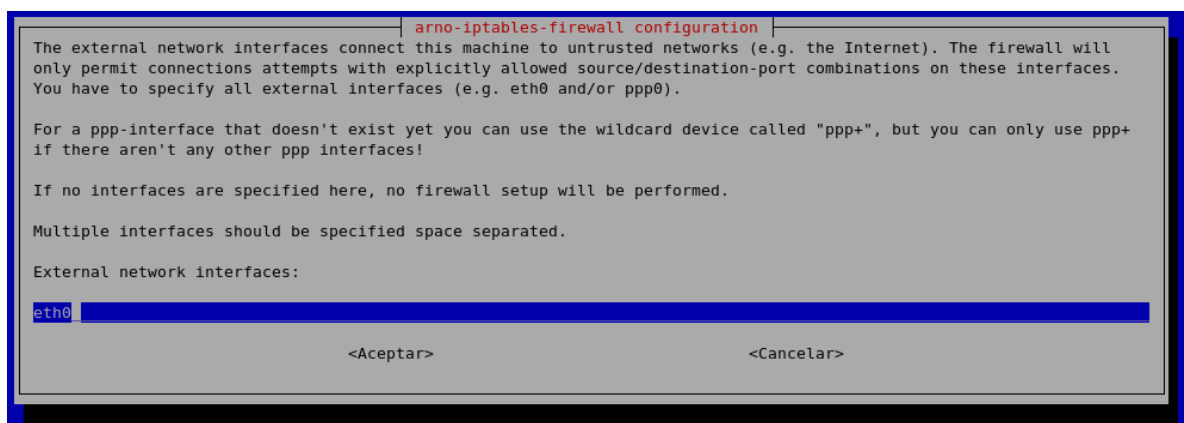


Fig. 45 Configuración de Firewall 2

En este paso, se dejan abiertos únicamente los puertos 80 (http), el puerto de apache, por el cual las peticiones van a estar transitando hacia y desde Internet, y el puerto 22 (ssh), este puerto permiten conectarse al servidor en forma remota, para labores de administración.

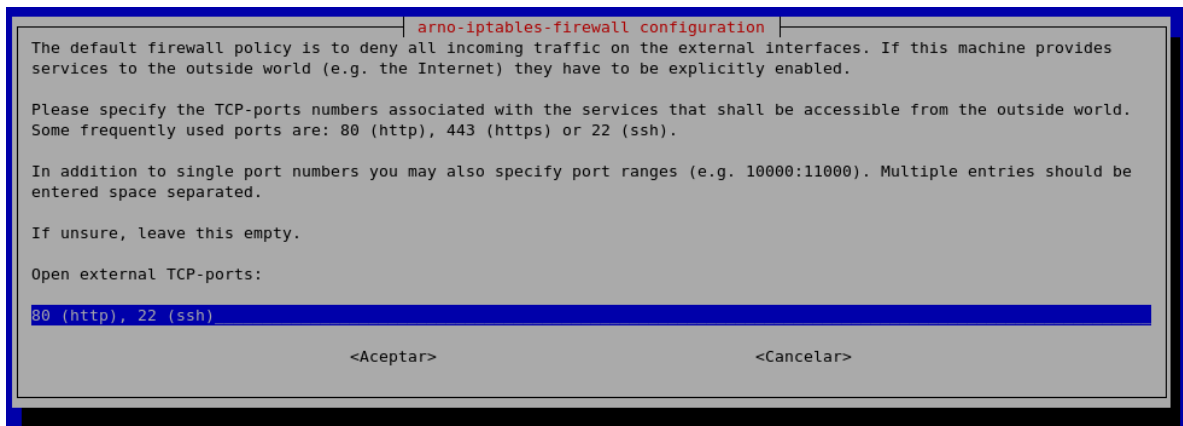


Fig. 46 Configuración de Firewall 3

EL puerto 25 es asignado para el protocolo simple de transferencia de correo. Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos, este puerto se deja abierto para permitir el intercambio de correo electrónico, se escribe 25 (smtp) y se selecciona <Aceptar>

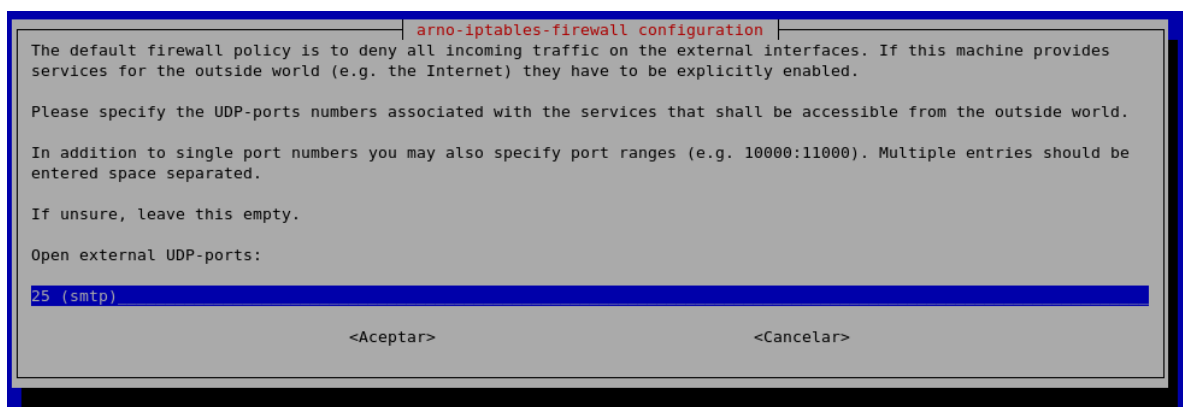


Fig. 47 Configuración de Firewall 4

En este paso, el asistente pregunta si se quiere que se pueda hacer ping al servidor desde afuera, acá se selecciona **<No>**, para que no pueda ser accedido desde afuera.

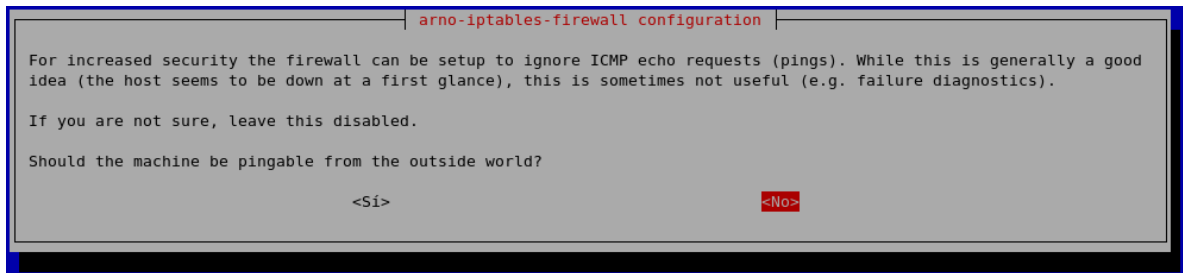


Fig. 48 Configuración de Firewall 5

En este paso de configuración se deja en blanco ya que el servidor no tiene sino una interfaz de red, por tanto no hay otras interfaces por las cuales se permite conectar a Internet otros equipos. Se selecciona **<Aceptar>**.

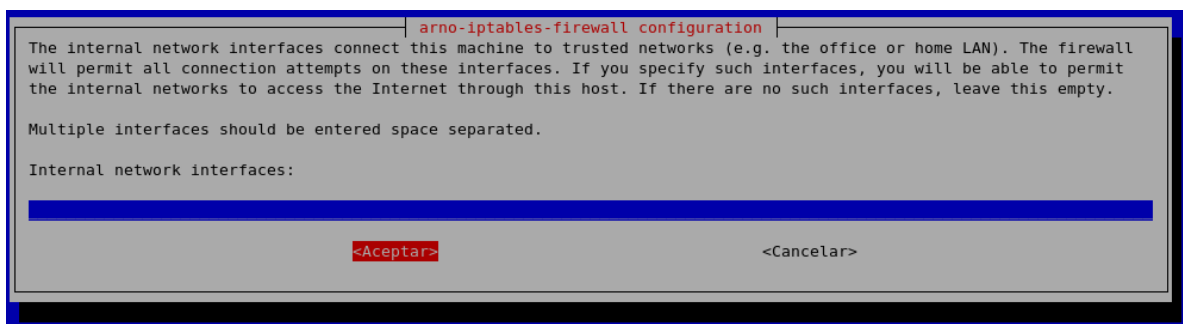


Fig. 49 Configuración de Firewall 6

Este es el paso final de la instalación, en el cual el asistente pregunta si se quiere reiniciar el firewall ahora para que se puedan aplicar las reglas, a lo cual se responde **<Sí>**

```
arno-iptables-firewall configuration

For security reasons the new firewall setup is not applied automatically. You might want to perform a manual
inspection of the firewall configuration in /etc/arno-iptables-firewall/firewall.conf, especially when upgrading to a
new version, as configuration variables might have changed.

In order to later manually apply the new firewall settings before the next reboot, invoke 'invoke-rc.d
arno-iptables-firewall start'.

If you do not need manual inspection, the firewall-setup can be applied now.

Should the firewall be (re)started now?
<Si>                                     <No>
```

Fig. 50 Configuración de Firewall 7

Listo, el firewall esta activo, se muestra el proceso de aplicación de las reglas de firewall. Para poder reconfigurar el firewall en futuras ocasiones se puede realizar mediante el comando por consola:

```
# dpkg-reconfigure arno-iptables-firewall
```

5. PROCESO DE PARAMETRIZACION

5.1. BUSQUEDA DE PATRONES

Al inicio del Proyecto EISIWeb, no se tuvo en cuenta tener un sitio parametrizado, extensible a otras escuelas, este portal era de uso exclusivo de la Escuela de Ingeniería de Sistemas, el proyecto siguió su evolución y bajo el concepto de exclusividad de la escuela. La creación de nuevos servicios de utilidad, despertaron el interés de otras escuelas por adquirir estos servicios, y de aquí, surge la idea de Parametrización.

La primera pregunta que surgió al momento de iniciar el proceso de Parametrización era la de ¿Cómo se podría realizar este proceso?, necesariamente tenía que haber una exploración exhaustiva, tanto en navegación del sitio en funcionamiento, como exploración completa de todos y cada uno de los archivos del sitio, hay que tener en cuenta que el sitio hasta ese momento contenía alrededor de 230 archivos que deberían ser explorados.

El objetivo final de la navegación y exploración de los archivos del sitio era el de buscar patrones, es decir valores que se repitieran en el sitio y que tuvieran que ver con información de cada una de las escuelas, un ejemplo, nombre completo de la escuela, nombre corto de la escuela.

Estos patrones se convertirían en los puntos críticos donde podría variar el sitio de una escuela a otra, y por esto debían ser convertidos en variables, las que serían los parámetros que se necesitaban para el sitio.

El proceso de obtención de los parámetros se realizó de la siguiente manera:

5.1.1. NAVEGACIÓN DEL SITIO EN FUNCIONAMIENTO

Se realizó una exploración completa del sitio en modo ejecución, es decir entrar a navegar por todos y cada uno de los módulos para tomar nota de los patrones que pudieran afectar la Parametrización.

Con esta exploración, se podían ir notando algunos valores que podrían ser variables en cada una de las escuelas. Como el sitio maneja una plantilla externa, no se podía saber exactamente el nombre de la página que contenía el patrón, pero hay herramientas que permiten encontrar estos patrones en los archivos.

Al momento de la navegación se iban tomando nota de los valores que posiblemente serían parametrizables, estos valores serían patrones externos, los que podrían afectar la parte gráfica del sitio en cada una de las escuelas.

Co la finalización del proceso de navegación y un análisis de cada uno de los patrones encontrados, se obtuvo una lista definitiva que era necesario que variaran en el proceso de implantación, estos patrones se denominaron Parámetros Externos. Acá se tiene la lista de los parámetros externos:

PARAMETROS EXTERNOS		
No.	NOMBRE PARAMETRO	DESCRIPCION
1	Nombre Completo Escuela	Es el nombre completo de la Escuela, normalmente la escuela de Ingeniería de Sistemas se llama "Ingeniería de Sistemas", cuando el nombre es completo es "Escuela de Ingeniería de Sistemas e Informática".
2	Nombre abreviado Escuela	Es la abreviación del nombre de la escuela, la abreviación del nombre de la Escuela de Ingeniería de Sistemas sería: "EISI".
3	Nombre Sistema	Es el nombre del sistema que se implementa en la escuela, en la Escuela de Ingeniería de Sistemas es "EISIWeb".
4	Nombre Largo Universidad	El nombre completo "Universidad Industrial de Santander"
5	Nombre Corto Universidad	La abreviación de Universidad Industrial de Santander "UIS".
6	Correo CALUMET	Se tomó como parámetro externo, teniendo en cuenta que puede cambiar en cualquier momento y no afecte el envío de correos del sitio
7	Nombre Servidor	El nombre del servidor de cada escuela, en la Escuela de Ingeniería de Sistemas "CORMORAN".
8	Versión	Versión actual en la que se encuentra el sistema, en la Escuela de Ingeniería de Sistemas la "3.0"
9	Ciudad	Ciudad donde se encuentra la escuela donde fue implantado el Sistema.

Tabla 4 Parámetros Externos

Estos parámetros son los que se muestran en las páginas, son visibles al usuario.

5.1.2. EXPLORACIÓN DE ARCHIVOS

El siguiente avance en el proceso de búsqueda de patrones, era la exploración de cada uno de los archivos, como se resalto anteriormente el sitio posee alrededor de 230 archivos, realizar de forma manual el proceso de búsqueda de patrones en cada uno de los archivos sería un trabajo extensivo. Afortunadamente se cuenta con un comando en Linux que permite buscar patrones contenidos dentro de un archivo, se quiso primero que todo mostrar el script ya que se recomienda que sea tenido en cuenta ya que puede ayudar a ahorrar un poco de tiempo si se esta realizando un trabajo de este tipo. El script es el siguiente:

En una consola de Linux Debian, se escribe:

```
: /carpeta/de/archivos# for i in *; do echo "Archivo: $i"; grep -n patron_a_buscar $i; echo"-----  
-----"; done
```

grep: Comando linux que permite buscar patrones requeridos dentro de los archivos. Para más información del comando grep, consulte la guía de Linux, grep - -help.

-n: Parámetro que se agrega al grep para que al encontrar un parámetro, muestre la línea exacta donde se encuentra el parámetro buscado.

patron_a_buscar: Puede ser un número, un string, es el patrón se desea buscar.

echo "-----": Es simplemente una división entre cada uno de los archivos, ya que al correr el script, se van a mostrar todos los archivos, por tanto para darle un poco de orden a los resultados se agrega esta línea para que divida los archivos.

Haciendo un estudio detallado de que otros patrones no serían visibles al usuario pero que serían necesarios que variaran en un proceso de implantación se llegaron a reconocer ciertos patrones, los cuales se convertirían en parámetros internos, estos parámetros funcionan dentro del sistema, no son visibles al usuario. Acá se tiene la lista de los parámetros externos:

PARAMETROS INTERNOS		
No.	NOMBRE PARAMETRO	DESCRIPCION
1	Ubicación de Tomcat	Este parámetro toma la ubicación exacta de los archivos de Tomcat, se dijo anteriormente que por medio de enlaces simbólicos se reunirían todas las carpetas de Tomcat en solo sitio, este sitio es el parámetro, el path completo de tomcat, por ejemplo "/carpetas/ubicacion/tomcat5.5"
2	IP Servidor	Es la IP propia del servidor, es una dirección IP dada por la universidad, "xxx.xxx.xxx.xxx"
3	URL	Es la dirección completa del sitio, por ejemplo en Sistemas: https://cormoran.uis.edu.co
4	Nombre Servidor	El nombre del servidor de cada escuela, en la Escuela de Ingeniería de Sistemas "CORMORAN". También está catalogado como un parámetro externo.
5	Nombre Corto Escuela	Este patrón se utiliza para la clasificación de las escuelas, para el caso de la Escuela de Ingeniería de Sistemas es "INGENIERIA DE SISTEMAS"
6	Usuario MySQL	Es el usuario utilizado en las conexiones a las bases de datos.
7	Contraseña MySQL	La contraseña utilizada para conexiones a las bases de datos.
8	Código Escuela	El código de cada escuela, para Sistemas es el Código "11"

9	Código Unidad Escuela	Código Centro de Costo, para la escuela de sistemas es el código "6570"
10	Código Maestría	Este código se utiliza para las materias de Maestría de cada Escuela, en Sistemas son los códigos "302" y "224"

Tabla 5 Parámetros Internos

Con la definición de los 18 patrones externos e internos, se procede a buscar estos patrones en cada uno de los archivos del sitio, el sitio esta dividido en módulos, cada módulo esta contenido en carpetas, por tanto se realiza el proceso de búsqueda en cada una de las carpetas contenidas en el sitio.

Como se dijo anteriormente, el script de búsqueda de patrones se ejecutaría 18 veces en cada carpeta, una vez por cada patrón, cada vez que se ejecutaba el patrón, se tomaba anotación de la ubicación del patrón tomando el nombre del archivo y la línea exacta donde se encontraba el patrón.

5.2. PARAMETRIZACIÓN DE EISIWEB

Teniendo establecidos los patrones, se procede a convertir estos patrones a variables, creando un JavaBean el cual iba a contener toda la información de la escuela establecida en los patrones.

El proceso fue crear una clase llamada Parametrización, la cual contenía la lista de parámetros, todos de tipo String, cada uno de los parámetros estaba contenido en un elemento de la lista. Se utilizó una lista de parámetros ya que era mas sencillo al momento de invocar el bean en los archivos JSP, o en las otras clases, se pudo haber utilizado nombres comunes para cada parámetro por ejemplo:

Para la IP se pudo haber utilizado la variable IPServidor, o para el nombre del sistema se pudo haber utilizado NombreSistema, pero al momento de invocar el bean había que crear cada una de las variables, por tanto se pensó en un vector que almacenara cada uno de los parámetros en cada uno de sus elementos respectivamente.

La clase quedaría de esta manera

```
public class Parametrizacion {

    public String [] Parametros() {

        String [] parametro = new String[19];

        parametro[0]="/carpeta/ubicacion/tomcat5.5"; //UBICACION DE TOMCAT
        parametro[1]="xxx.xxx.xxx.xxx"; //IP SERVIDOR
        parametro[2]="http://xxxxxxx.uis.edu.co"; //URL
        parametro[3]="XXXXWeb"; //NOMBRE ABREVIADO ESCUELA
        parametro[4]="Escuela XXXXXX"; //NOMBRE LARGO ESCUELA
        parametro[5]="XXXXXX"; //NOMBRE SISTEMA
        parametro[6]="UIS"; //NOMBRE CORTO UNIVERSIDAD
        parametro[7]="Universidad Industrial de Santander"; //NOMBRE LARGO UNIVERSIDAD
        parametro[8]="calumet@uis.edu.co"; //CORREO CALUMET
        parametro[9]="xxxxxxx"; //NOMBRE SERVIDOR
        parametro[10]="X.X"; //VERSION
        parametro[11]="Bucaramanga"; //CIUDAD
        parametro[12]="INGENIERIA XXXXXX"; //NOMBRE CORTO ESCUELA
        parametro[13]="xxxxxxx"; //USUARIO MySQL
        parametro[14]="*****"; //CONTRASEÑA MySQL

        // SISTEMAS 11,302,6570 , 224
        // CIVIL 21,6540
        // INDUSTRIAL 23,6530

        parametro[15]="xx"; //CODIGO ESCUELA
        parametro[16]="xxxx"; //CODIGO UNIDAD ESCUELA
        parametro[17]="xxx"; //CODIGO MAESTRIA
        parametro[18]="xxx"; //CODIGO MAESTRIA

        return parametro;

    }

}
```

En las páginas JSP, donde sea necesario utilizar Parametrización, es decir donde hayan parámetros que pueden variar de una escuela a otra, ya sean externos o internos, la invocación del JavaBean se hace de forma similar a cualquier otro bean, a continuación se muestra un ejemplo de cómo se realizó en las páginas JSP, donde fuera necesario:

```
<jsp:useBean id="Parametrizacion" class="beans.Parametrizacion" scope="page"/>  
<%  
String [ ] parametro = Parametrizacion.Parametros();  
%>
```

La utilización de las variables se vería de la siguiente manera

Si se necesita mostrar el nombre completo de la escuela en una página JSP, y este nombre varía de una escuela a otra se hace de la siguiente manera:

```
<%=parametro[4]%>
```

Ahora, si se necesita el nombre completo de la escuela en una variable de un javascript, se tomaría de la siguiente manera:

```
escuela=parametro[4];
```

La variable escuela debe ser de tipo String.

En los archivos .java, de donde se generan las clases que utiliza el sitio, también se parametrizaron, el proceso de identificación de patrones se realizó de la misma manera como se llevó a cabo con los archivos JSP, buscando los mismos patrones, al tener identificados los archivos .java, el proceso de Parametrización se realizó de la siguiente manera:

Se invoca el bean de Parametrización

```
Parametrizacion objeto = new Parametrizacion();  
String [ ] parametro = objeto.Parametros();
```

En las dos líneas anteriores se crea una instancia de tipo parametrización, la cual se va a cargar en una lista de tipo string llamada parámetro, es decir, acá se cargan los parámetros.

Para utilizar cualquier parámetro por ejemplo para llenar una variable con el nombre del servidor, se haría de la siguiente manera:

```
Servidor = parametro[9];
```

Hay que tener en cuenta que la variable Servidor debe ser de tipo String.

Este proceso se realizó en los archivos donde fuera necesario realizar Parametrización, tanto en los archivos JSP, como los archivos .java que generan las clases utilizadas por el Sitio.

De esta manera se parametrizó El portal de la Escuela de Ingeniería de Sistemas EISIWeb, ahora se verá a continuación como se implanta este portal parametrizado en un servidor de otra escuela.

5.3. PROCESO DE IMPLANTACIÓN DEL PORTAL EN OTRA ESCUELA

5.3.1. 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:

- **Sistema Operativo Linux**, el sitio se diseño completamente para correr sobre servidores Linux, aunque se podría hacer un estudio más profundo en la Parametrización para que este pueda correr en cualquier tipo de servidor, ya sea Linux o Windows.
- **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 APACHE**, aunque no es relevante, es importante saber con que servidor apache se cuenta.
- **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.

5.3.2. 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 "XXXXX", esta carpeta se ubica en la carpeta de aplicaciones de tomcat (webapps), para que sea reconocida por el servidor, hay que recordar la carpeta de aplicaciones de tomcat [/usr/share/tomcat5.5/webapps](#), hay que tener en cuenta que esta ubicación se creó cuando se instaló tomcat desde los repositorios de Debian 4 Etch, y la carpeta webapps es un enlace simbólico a la carpeta webapps ubicada en [/var/lib/tomcat5.5/](#)
- 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.

5.3.3. CARGANDO LAS BASES DE DATOS DEL SITIO 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, es el siguiente:

En una consola de Linux ya que no es necesario el usuario root porque en MySQL se ha creado el usuario propio de las base de datos.

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

```
:$ mysql -u xxxxxx -p  
Password: *****
```

```
mysql> source /ubicacion/de/la/base/de/datos/nombrededatos.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.

Para más información sobre el manejo de comandos MySQL, puede consultar la guía general de referencia de MySQL que se dará al final en la bibliografía de este documento.

5.3.4. LIBRERIAS .JAR 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 tabla,

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.
lfxjdbc.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.

Tabla 6 Librerías necesarias para el funcionamiento del Portal

5.3.5. REINICIANDO SERVICIOS

Para que los cambios realizados surtan efecto, es necesario reiniciar varios de los servicios en el siguiente orden:

En una consola como root:

```
:# /etc/init.d/mysql restart
```

```
:# /etc/init.d/apache2 force-reload
```

El servidor tomcat se inicia desde el script de arranque de tomcat, por defecto el servidor se inicia desde el script de Debian ubicado en la carpeta [/etc/init.d/](#), pero para hacer un arranque efectivo del servidor se inicia desde el script propio de tomcat de la siguiente manera:

```
:# /etc/init.d/tomcat5.5 stop
```

```
:# /usr/share/tomcat5.5/bin/startup.sh
```

Se obtendrá la siguiente información:

```
Using CATALINA_BASE: /usr/share/tomcat5.5
Using CATALINA_HOME: /usr/share/tomcat5.5
Using CATALINA_TMPDIR: /usr/share/tomcat5.5/temp
Using JRE_HOME: /usr/lib/jvm/jdk/jre
```

Para verificar que los servicios estén todos arriba y funcionando se hace un mapeo de los puertos, se recuerda el uso del comando nmap.

```
:# nmap localhost
```

El cual arrojará la siguiente información:

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2008-10-19 00:58 COT
Interesting ports on localhost (127.0.0.1):
Not shown: 1672 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
3306/tcp  open  mysql
8009/tcp  open  ajp13
```

De esta manera se observa que el Puerto 8009/tcp esta abierto, esto quiere decir que el servicio de tomcat esta arriba y funcionando.

5.3.6. CORRIENDO EL SITIO POR PRIMERA VEZ

Si se ha ejecutado el proceso de implantación correctamente, para correr el sitio por primera vez ejecutándolo desde el mismo servidor donde fue alojado, se abre un navegador, se escribe en la barra de navegación localhost. Si se esta ejecutando desde cualquier otro equipo, se escribe el dominio del servidor seguido de la aplicación, por ejemplo, el sitio implantado en la escuela de ingeniería civil

<http://albatros.uis.edu.co/eisi> o en la escuela de Ingeniería Industrial <http://carpintero.uis.edu.co/eisi>

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:



Fig. 51 Primera Vista Preliminar del Sitio Implantado

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 verá de la siguiente forma:

Para la escuela de Ingeniería Civil:



Fig. 52 Vista Preliminar del Sitio implantado en la Escuela de Ingeniería Civil

Sitio Implantado en la Escuela de Ingeniería Industrial



Se puede observar como varía el aspecto de cada uno de los sitios, pero el funcionamiento es el mismo, se ofrecen los mismos servicios que ofrece EISIWeb.

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.	
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	
Actualización de la BD del Sitio	Se actualiza la BD del sitio con la información de Usuarios, Docentes, Horarios, Matriculas, Egresados.	

Tabla 7 Pruebas de Conexión y Actualización de las BD

6.2. NAVEGACION

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.	✓
Listado de Docentes	Se muestra la lista completa de docentes	✓
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	✓
Menú Programas	Se muestran los planes de pregrado, Especialización, Maestría y Planes de Estudio	✓
Menú ECI-HOY	Se muestran los comunicados, Agenda, Cartelera, Cursos y eventos propuestos.	✓
Menú Grupos	Se Muestran los grupos y la información de cada uno de los grupos.	✓

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.	✓
Consultas y Sugerencias	Se realizan consultas y sugerencias del sitio, y da confirmación de que la sugerencia ha sido enviada.	✓
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.	✓

Tabla 8 Pruebas de Usuario Invitado

6.2.2. USUARIO REGISTRADO

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,	✓
Listado de Docentes	Se muestra la lista completa de docentes	✓
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	✓
Menú Programas	Se muestran los planes de pregrado, Especialización, Maestría y Planes de Estudio	✓
Menú ECI-HOY	Se muestran los comunicados, Agenda, Cartelera, Cursos y eventos propuestos.	✓
Menú Grupos	Se Muestran los grupos y la información de cada uno de los grupos.	✓
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.	✓
Consultas y Sugerencias	Se realizan consultas y sugerencias del sitio, y da confirmación de que la sugerencia ha sido enviada.	✓
Subir Archivos	Se suben archivos al servidor, por Materias, Docentes, Grupos.	✓

Ver Archivos	Se ven los archivos subidos al servidor, de las Asignaturas, Docentes, Grupos.	✓
Participación en Foros	Se hacen participaciones en los Foros.	✓
Envío de Correos	Envío de correos por parte de los Estudiantes, y por parte de los Docentes.	✓

Tabla 9 Pruebas de Usuario Registrado

6.2.3. USUARIO ADMINISTRATIVO

PRUEBA	DESCRIPCION	RESULTADO
Información Institucional	La información Institucional se edita, se modifica y se guardan los cambios.	✓
Creación Manual de Usuario	Se crea de Forma Manual un usuario, permite entrar la información y guardar los cambios.	✓
Envío de Correos	Se envía Correos por parte del Administrador.	✓
Creación de Backups	Se generan Backups de la BD del sitio.	✓
Administración Publicaciones	Se crean, editan, Eliminan, guardan cambios a los Comunicados, Cartelera y Cursos.	✓
Administración de Grupos	Se crean Grupos, se editan, se modifican, se guardan cambios en los Grupos.	✓
Administración de Servicios	Se crean Servicios, se editan, se modifican y se guardan cambios en los Servicios.	✓
Administración de Sugerencias	Se reciben sugerencias, se muestran, se atienden, se eliminan, se envía y llega correo de respuesta a la sugerencia	✓

Tabla 10 Pruebas de Usuario Administrativo

7. CONCLUSIONES

- Con el desarrollo de este proyecto se puede demostrar que la Ingeniería de Sistemas abarca un amplio campo de investigación desde el cual se pueden realizar experimentos e implementaciones de los productos investigados y desarrollados como en cualquier otra área de la ciencia, obteniéndose resultados que pueden ir mejorando cada vez más los sistemas que se han desarrollado, los que se encuentran en desarrollo y los que serán desarrollados.
- 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 el desarrollo 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”.
- La Escuela de Ingeniería de Sistemas e Informática, al contar con un sistema de información tan completo como lo es EISIWeb, ha despertado un interés por parte de otras escuelas debido a los excelentes servicios que el portal ofrece a la comunidad EISI. Es importante destacar el diseño de las interfaces, las cuales están diseñadas de tal manera que cualquier usuario de la página pueda hacer uso de los servicios que se ofrecen ya que son de fácil y comprensible manejo.
- La Parametrización era una fase por la cual necesariamente tenía que pasar EISIWeb, ya que, era fundamental que el sitio estuviera en su

mayoría parametrizado, ya que facilitó la implantación del sistema en otras escuelas, se podría decir que EISIWeb ya cuenta con dos hijos, ECIWeb en la escuela de Ingeniería Civil y EEIEWeb en la Escuela de Estudios Industriales y Empresariales.

- Actualmente el portal de la escuela de Ingeniería Civil se encuentra en total funcionamiento y al igual que en EISIWeb, y gracias al grupo CALUMET, se encuentra en constante investigación y desarrollo de nuevos módulos y servicios que permitan cada vez mas una mejor interacción, y comodidad en el desempeño de labores académicas de los usuarios de la escuela.

8. SUGERENCIAS Y RECOMENDACIONES

- Se recomienda realizar mejoras, mantenimientos preventivos y de seguridad a los servidores en los cuales se encuentran alojados los portales, teniendo así, los sistemas operativos actualizados evitando problemas de seguridad que afecten el nombre del Grupo CALUMET y de la escuela de Ingeniería de Sistemas.
- Cada vez mas surgen necesidades propias de cada escuela que se puedan resolver por medio del portal, no para todas las escuelas son las mismas necesidades, por tanto se recomienda hacer un proceso de investigación, análisis y diseño de las necesidades que podría tener cada escuela y desarrollar módulos y servicios propios de cada escuela.
- Se recomienda seguir investigando que otros patrones se podrían convertir en parámetros para conseguir que el portal sean 100% parametrizado.
- Se recomienda al Grupo de Desarrollo adoptar por completo el paradigma de Sistema Parametrizado, esto asegura que los nuevos módulos y servicios desarrollados sean 100% compatibles y de fácil implantación en cada uno de los portales implantados en cada una de las escuelas.
- Se recomienda la investigación de nuevas tecnologías en las cuales se podría desarrollar el sitio, asegurando una mejor calidad en el ofrecimiento de los servicios y la facilidad de desarrollo de nuevos módulos y servicios, además de promover el conocimiento de estas nuevas tecnologías, aprenderlas y utilizarlas en proyectos futuros.

BIBLIOGRAFIA

TEXTOS BIBLIOGRÁFICOS

- **BRUEGGE BY DUTOIT A. Ingeniería de Software Orientado a Objetos. Prentice Hall. 2002.** Aquí encontramos todo lo relacionado con las metodologías de desarrollo software
- **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.
- **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.
- **IEEE Std 1016-1998 Recommended Practice for Software Design Descriptions.** Especificaciones y normas de la IEEE para el diseño de software.
- **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.

- **SWEBOK – Guide to the Software Engineering Body Of Knowledge, 2004 version.** Metodologías para el desarrollo de software.
- **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.
- **LEMAY, Laura; PERKINS, Charles. Aprendiendo JAVA en 21 días. Editorial Prentice Hall. México, México D.C. 1996.** Manual de referencia para principiantes sobre JAVA.
- **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.
- **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.
- **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.
- **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.

HERRAMIENTAS Y PÁGINAS WEB UTILIZADAS

- **www.debian.org.** Sitio oficial donde se puede descargar la versión del sistema operativo Debian 4, versión que va a ser instalada en el servidor Web de la Escuela de Ingeniería Civil (ALBATROS).
- Para la Instalación de Java, Apache, Tomcat, MySQL, y el conector JK, se utilizarán los repositorios oficiales del sistema operativo Debian:
 - deb `http://ftp.us.debian.org/debian/etch main contrib non free`
 - deb-src deb `http://ftp.us.debian.org/debian/etch main contrib non free`
- **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.
- **PDF MySQL Reference Manual.** Manual que contiene información completa y detallada sobre MySQL. Facilitado por los anteriores desarrolladores de grupo de desarrollo CALUMET.
- **PDF Hanna, Phil. JSP Manual de referencia.** Manual completo de JSP. Facilitado por los anteriores desarrolladores de grupo de desarrollo CALUMET.
- **www.htmlgoodies.com/primers/jsp.** Guía práctica para principiantes en jsp, ilustración de conceptos básicos de este lenguaje mediante ejemplos.
- **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.

- **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.
- **www.astalaweb.com**. Sitio Web que presenta una guía de todo lo relacionado con Java Script.
- **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.
- **<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.