UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

REINGENIERÍA DEL PRINCIPAL PORTAL WEB DE LA FACULTAD DE INGENIERÍA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERA EN COMPUTACIÓN

P R E S E N T A:

CLAUDIA NELLY ARRIAGA HERNÁNDEZ

DIRECTOR DE TESIS:

ING. NOÉ CRUZ MARÍN

CIUDAD UNIVERSITARIA, MÉXICO D. F. 2014

DEDICATORIAS

Primero que nada quiero dedicar este trabajo a dos personas muy importantes en mi vida y que gracias a ellos he logrado alcanzar todas las metas que me he propuesto; a mis padres, ***Marco Antonio Arriaga Arvizu y Isidora Hernández Bautista*** *gracias por confiar y creer siempre en mí, porque más que un logro mío , es suyo, gracias por todos los esfuerzos que hicieron los dos por sacarnos adelante a mi hermano y a mí , porque yo sé todo lo que les costó y les ha costado, brindarnos lo necesario, para darnos armas y que podamos salir adelante y valernos por nosotros mismos, no tengo palabras para decirles cuanto les agradezco todo lo que han hecho por mí, por apoyarme en cada momento y en cada decisión de mi vida, por aconsejarme y escucharme, por aguantar mis malos humores por desvelase conmigo cuando hacia tareas, y sobre todo por levantarme cada vez que sentía que ya no podía con la escuela , aún recuerdo cuando llegue frustrada del primer semestre de la carrera pero gracias a ustedes logre salir adelante y culminar ésta meta, los amo y me siento muy orgullosa y afortunada por tenerlos como padres , los amo con todo mi ser.*

También quiero agradecer a mi hermano ***Marco Antonio Arriaga Hernández****, por todo el apoyo brindado, por escucharme, porque más que mi hermano es mi mejor amigo, gracias por aguantarme, te amo hermanito.*

A mi perro ***Greñas****, mi amigo incondicional que siempre estaba a mi lado desvelándose conmigo y no se dormía hasta que yo no me levantara del escritorio jajaja, gracias amigo por tu gran compañía.*

A mis amigos de la carrera que más que amigos se hicieron como mis hermanos a los ***baaahh!, Areli , Tonathiu , Cinthia, Alberto, Luis, Alfredo, Cesar Vázquez*** *gracias por todo el apoyo comprensión y amistad que me han brindado no sólo en el transcurso de la carrera , sino hasta el momento, que sigamos cosechando muchos éxitos juntos como hasta ahora,* ***1,2,3 baaahhh!!.***

Gracias a ***la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería*** *mi alma mater, no puedo decir que es más que un orgullo y un honor pertenecer a ésta institución.*

A la ***Unidad de Servicios de Cómputo Académico (UNICA),*** *donde me la viví toda mi carrera desde que era apoyo en las salas hasta que fui becaria, donde además de brindarnos una formación profesional, aprendemos mucho como seres humanos, porque más que ser una institución educativa es una gran hermandad, donde conocí a muchas personitas importantes que siempre me apoyaron como el* ***Ing. Sergio****, gracias por todos tus consejos y por brindarme la*

primera oportunidad de pertenecer a UNICA, a las ***Ing. Chary y Bety*** *gracias por apoyarnos y escucharnos a Vik y a mí, a* ***Ibeth y a Nelida*** *gracias chicas por escucharme y aconsejarme y por el apoyo cuando fuimos becarias Neli. No puedo dejar de agradecer a todo el equipo del* ***"DROS"*** *donde conocí a grandes amigos y compañeros* ***Marcela, Faraday, Lety, Mike*** *porque con ustedes aprendí muchas cosas y me apoyaron siempre cuando no sabía qué hacer, si se me caía el server jajajaja, pero más que nada gracias por su amistad chicos y por todos los momentos compartidos ñ_ñ* ***y arriba el 28!!!!!.*** *También quiero agradecer a un gran persona, que más que mi jefe (en su momento) y director de tesis en un gran amigo de quien aprendí mucho, no sólo como profesional, sino también como ser humano, un gran hombre que admiro y quiero mucho, el* ***Ing. Noé Cruz Marín****, gracias por guiarme no sólo en el transcurso de mi tesis sino también en mi vida, gracias por la confianza y el apoyo, por todos tus consejos, y por escucharme cuando andaba toda depre jajajaja, de verdad que valoro mucho tus consejos y he aprendido mucho de ellos.*

A mis compañeros y amigos del trabajo. ***Miguel Manuel Meléndez*** *gracias por todo tu apoyo, comprensión, paciencia y sobre todo por brindarme tu amistad e impulsarme siempre a cumplir está meta. Gracias* ***Teresa Mondragón*** *y* ***Alfredo Domínguez*** *por brindarme su amistad incondicional y por apoyarme en cada momento para concluir está etapa.*

Gracias dios por darme ese empujoncito que siempre me hizo falta para terminar mis metas y superar mis caídas.

Ya por último a una personita muy importante y especial en mi vida, el corazón de mi alma, mi esposo ***Victor Dueñas Tello****, gracias por todo tu apoyo y comprensión por estar siempre conmigo en las buenas y en las malas desde que éramos novios, por apoyarme en el transcurso de mi tesis, por desvelarte a mi lado, por caminar juntos de la mano siempre en la misma dirección, por levantarme cuando siento que ya no puedo más, tu eres mi fuerza y mi luz, gracias por todas las palabras de aliento para tranquilizarme, no sabes cómo le agradezco a dios y a la vida por haberte puesto en mi camino, eres un gran hombre y un gran ser humano, te amo mi cielo y así va a ser toda la vida.*

INTRODUCCIÓN

Secretaría General

La Secretaría General coordina y apoya la debida ejecución de las actividades de carácter académico de la Facultad de Ingeniería, además de promover la participación coordinada de los directivos de ésta en la planeación y administración de la institución, así como su vinculación sistemática en decisiones cotidianas. También se encarga de elaborar, implementar y dar seguimiento a los planes y programas de trabajo necesarios para el buen funcionamiento de la Facultad.

Para cumplir con sus funciones la Secretaría General cuenta con una estructura orgánica dividida en varios departamentos encargados de tareas muy específicas, su organización es la siguiente: (Figura 1).

Figura 1. Organigrama de la Secretaría General.

Como podemos visualizar en la imágen anterior, la Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería (UNICA) se encuentra bajo la dirección de la Secretaría General, pero ¿Qué es UNICA? Es el resultado de la división del antiguo Centro de Cálculo (CECAFI), que en 1994, por la gran cantidad de labores que realizaba, fue separada en dos unidades independientes UNICA y la Unidad de Servicios de Cómputo Administrativo (USECAD). Ésta división se implementó para tener un mejor control de esas tareas académicas y administrativas de la Facultad de Ingeniería. UNICA quedó organizada en tres departamentos (Figura 2).

Figura 2. Organigrama de UNICA en el año 1994.

En el año 2000 el organigrama de UNICA fue reestructurado, y el Departamento de Cómputo Avanzado cambió de nombre por el Departamento de Redes y Operación de Servidores debido a las actividades más concretas que realiza en la actualidad. Para el año 2003, dado el trabajo que realizaba el Departamento de Redes y Operación de Servidores, fue creado otro departamento, en éste caso sería el Departamento de Seguridad en Cómputo, el cual fue puesto en marcha en 2004 (Figura 3).

Figura 3. Organigrama actual de UNICA.

Las funciones que desempeña la Unidad de Servicios de Cómputo Académico son:

- Mantener el liderazgo en cuanto a tópicos en cómputo.
- Proporcionar recursos de cómputo de calidad a la comunidad de la Facultad.
- Impulsar la creación de una política de cómputo definida.
- Lograr la capacitación cada vez más completa y actualizada para la formación de recursos humanos.
- Aplicar todos los conocimientos y las herramientas de cómputo con los que cuenta la unidad para realizar las actividades de forma más eficiente y segura.

La Unidad de Servicios de Cómputo Académico proporciona a la comunidad los siguientes servicios:

- Utilización de equipo de cómputo.
- Impresión de documentos.
- Asesoría especializada y personalizada.
- Correo electrónico.
- Internet.

- Cuentas personales de base de datos.
- Servicio de "hosting" de páginas Web.
- Administración del Sistema de Seguridad Informática.
- Programa de servicio social.
- Prácticas profesionales.
- Programa de Formación de Becarios.

Cabe destacar que el principal portal Web de la Facultad de Ingeniería está alojada en uno de los servidores de UNICA que se encuentra a cargo del Departamento de Redes y Operación de Servidores.

Redes y Operación de Servidores

Las funciones del Departamento de Redes y Operación de Servidores son las siguientes:

- Administrar, operar y dar mantenimiento a la red de comunicación de la Facultad, y de la intercomunicación con la red central de la UNAM.
- Desarrollar e implementar proyectos de la red para la expansión del servicio, así como también coordinar, colaborar y actualizar el diseño del proyecto de la red en la Facultad de Ingeniería.
- Administrar y dar mantenimiento a las cuentas de servicios de correo electrónico.
- Administrar el servidor de Web de la Facultad de Ingeniería (actualmente aloja 230 páginas).
- Administrar y dar mantenimiento a 145 cuentas de bases de datos.
- Administrar el servidor de plataformas educativas y sala Linux Anexo.
- Administrar el servidor de monitoreo y proyectos de redes.
- Brindar soporte técnico a Secretarías y Divisiones, así como también a alumnos y profesores dentro de la Facultad de Ingeniería.

En el desarrollo de esta tesis veremos varios aspectos importantes, en el capítulo 1 abordaremos todos los antecedentes (desde qué es un servicio Web, hasta sus ventajas y desventajas), definiremos el proyecto y su alcance, trataremos la problemática actual del servicio, y finalmente haremos una breve descripción sobre la historia del portal de la Facultad de Ingeniería y la evolución que ha tenido con el paso del tiempo.

En el capítulo 2 abordaremos los temas de Protocolos y software que intervienen en el servicio Web. Ofreceremos una breve descripción del Protocolo de Transferencia de Hipertexto (HTTP), de cómo funciona y cómo interviene en el servicio Web, y del Protocolo de Transferencia de Hipertexto Seguro (HTTPS), su funcionamiento, sus ventajas y desventajas en torno a HTTP. Más adelante, en éste mismo capítulo, hablaremos de todo el software que interviene para brindar el servicio Web: los diferentes sistemas operativos, los servidores Web con los que se cuentan, los lenguajes de programación en torno a las necesidades de la Facultad de Ingeniería (como los son PHP y JSP), y las herramientas de administración y creación de estadísticas de los sitios. Por último, trataremos los temas de seguridad y herramientas necesarios para fortalecer el servidor Web, y de algunas pruebas de *benchmarking* para servidores tipo Linux.

En el siguiente capítulo, el tercero, tocaremos el tema de la reingeniería: qué implica, la metodología a seguir para hacer reingeniería y los componentes esenciales de la misma.

En el capítulo 4 hablaremos sobre la virtualización de servidores: las diferentes arquitecturas de virtualización, como son la de tipo Hosted y la del tipo Hipervisor, las diversas tecnologías necesarias para dichas arquitecturas, sus ventajas y desventajas. Además, haremos una breve descripción de los diferentes proveedores de virtualización de servidores y referiremos las principales características de VMWare ESX, que es el proveedor con el cual trabajamos para realizar este proyecto.

En el capítulo 5 empezaremos a tratar la problemática del proyecto del portal Web de la Facultad de Ingeniería. En él detallaremos cómo se implementaron las diferentes etapas de la metodología de reingeniería para el desarrollo del proyecto, hablaremos de las buenas prácticas para la seguridad de nuestro sistema operativo y de la aplicación de *hardening* en sistemas Linux. Asimismo, enunciaremos las políticas de la Facultad de Ingeniería, con base en las cuales haremos recomendaciones de buenas prácticas para aumentar su ranking Web en relación con las diferentes universidades del mundo.

Finalmente, en el capítulo 6 abordaremos tanto la implementación como las pruebas que se realizaron en el servidor, y haremos un recuento de las herramientas que se utilizaron, las cuales ya referimos en el capítulo 2.

CAPÍTULO 1

ANTECEDENTES

1.1. Definiciones

Internet, o simplemente "la Red" como algunas veces solemos llamarlo, es un sistema mundial de redes de computadoras de todo el mundo, donde cualquier usuario que cuente con los permisos, o bien con los privilegios necesarios, puede acceder a información de otra computadora e incluso tener comunicación con otros usuarios conectados en otras computadoras, siempre y cuando éstas estén conectadas a Internet (la red mundial).

En la actualidad Internet es un medio de comunicación tan grande que se ha vuelto público y accesible para millones de personas en todo el mundo, ya que de cierta manera, podemos intercambiar nuestra información y con ello crear aportaciones a la misma, volviéndose cooperativo y autosuficiente. Visto de una manera física, Internet usa parte del total de recursos existentes en las redes de telecomunicaciones. Técnicamente, lo que distingue a Internet es el

uso del protocolo de comunicación llamado Transmission Control Protocol/Internet Protocol (TCP/IP).

a) World Wide Web (WWW).

La World Wide Web es un sistema de hipertexto que funciona sobre Internet, donde encontramos una fuente de conocimiento humano a la cual podemos acceder desde cualquier lugar del mundo. La exploración en la Web se realiza por medio de un software especial denominado *Browser* (navegador) o Explorador, con el cual se pueden extraer elementos de información para mostrarlos al usuario que las solicite. La apariencia de un sitio Web puede variar ligeramente dependiendo del explorador que use.

b) Página Web.

Este documento electrónico que contiene información específica de un tema en especial es la unidad básica y significativa de información accesible en la *WWW*, y se almacena en un sistema de cómputo que se encuentra conectado a la red mundial Internet.

Una página Web se caracteriza por combinar un texto con imágenes con el objeto de que el documento sea dinámico, permitiendo ejecutar diferentes acciones, una tras otra, a través de la selección de texto remarcado o de las imágenes. Ésta acción dinámica nos puede conducir a otra sección dentro del documento, abrir otra página Web, iniciar un mensaje de correo electrónico o transportarnos a otro sitio Web totalmente distinto a través de hipervínculos o enlaces.

c) Sitio Web.

Éste es un conjunto de archivos o documentos electrónicos y páginas Web que hacen referencia a un tema en específico. Está conformado generalmente por un conjunto de páginas organizadas a partir de una página inicial de bienvenida o mejor conocida como “home page”. Es posible acceder a un sitio Web mediante una URL única, con un nombre de dominio y dirección de Internet específica.

Un sitio Web no necesariamente debe alojarse en el sistema de cómputo de su negocio, los documentos que lo integran pueden ubicarse en un equipo de otra localidad, inclusive en otro país, el único requisito es que el equipo en el que residan los documentos esté conectado a la red mundial de Internet. Éste equipo de cómputo o servidor Web, como se le denomina técnicamente, puede contener más de un sitio Web y atender concurrentemente a los visitantes de cada uno de los diferentes sitios.

d) Portal.

Portal es un sinónimo de puente para referirse a un sitio Web, el cual sirve de sitio principal a las personas que se conectan a la *WWW*. Su objetivo principal es ofrecer a los usuarios una serie de recursos y servicios, entre los cuales podemos encontrar: buscadores, foros, compras electrónicas y más; es decir, los portales son empleados para localizar la información y los sitios de interés, y de ahí comenzar nuestra actividad en Internet. Un portal está encaminado especialmente a resolver necesidades específicas de una comunidad, o dar acceso a la información y servicios de una institución pública o privada.

e) Hospedaje Web.

También conocido como *hosting*, puede definirse como un servicio de almacenamiento, acceso y mantenimiento de los archivos que conforman y le dan forma a un sitio Web. Éste servicio de almacenamiento reserva un espacio en disco duro donde se alojaran todos los archivos, imágenes y videos que darán forma al sitio; por ello es importante tomar en cuenta los recursos del equipo donde albergaremos nuestros sitios Web, así como del espacio disponible con el que cuenta. Si un sitio Web empieza a ser muy robusto, es probable que el servidor Web en el que se encuentren instalados los archivos electrónicos tenga que dedicarse única y exclusivamente a atender a este sitio. Este servicio se conoce como "hospedaje Web dedicado".

1.1. Definición de Servicio Web

Un servicio Web se define como un sistema de software diseñado para permitir la interoperabilidad máquina a máquina en una red. En general, un servicio Web es un conjunto de Interfaces de Programación de Aplicaciones (API's) Web, las cuales pueden ser accedidas en una red, siendo éstas ejecutadas en un sistema de *hosting* remoto.

El servicio más popular de Internet es conocido como *World Wide Web* (*WWW*), o simplemente Web. Este servicio consiste básicamente en el uso del protocolo HTTP para que desde un cliente (navegador o *browser*), se solicite un documento dentro de la red, y un servidor Web le sirva una página en formato HTML, abarcando múltiples y diferentes sistemas; es decir, llevan una comunicación entre diferentes máquinas, con diferentes plataformas y entre programas distintos; enfocándonos hacia lo que es un cliente y un servidor comunicándose por medio de mensajes entre sí, usando el estándar SOAP (Simple Object Access Control Protocol, o bien Protocolo Simple de Acceso a Objetos).

En general, podemos decir que un servico Web, nos permite el acceso a la información por medio de documentos de hipertexto (páginas Web), que incluyen datos en cualquier tipo de formato (texto, fotos, video, audio, etcétera) referenciados entre sí.

El servicio Web sigue el modelo cliente-servidor; donde los servidores Web se encuentran conectados a Internet, contienen las páginas Web (*hosting*) y esperan permanentemente las peticiones de los clientes. Los clientes Web, que son los navegadores, se encargan de llevar a cabo éstas peticiones.

1.2. Historia del servicio Web

Una de las primeras ideas de la Web se remonta a la propuesta de Vannevar Bush, quien en los años 40 planteó un sistema similar a lo que hoy en día es la Web; a grandes rasgos, este sistema consistía en un conjunto de información distribuida a través de una interfaz operativa que permitía el acceso a la misma, como a otros artículos relevantes determinados por claves. Este proyecto nunca fue materializado, quedando relegado al plano teórico bajo el nombre de Memex. Es en los años 50 cuando Ted Nelson realizó la primera referencia a un sistema de hipertexto, donde la información es enlazada de forma libre.

A principios de los 80, cuando comenzaba el Internet, los servicios más utilizados eran el correo electrónico y la transferencia de archivos a través de un Protocolo de transferencia de archivos (FTP); así como el servicio *Gopher*, que permitía el acceso a diversos recursos de Internet a través de un menú de texto. Por tanto, fue hasta principios de los años 90 cuando apareció la *World Wide Web,* mejor conocida como *WWW* o simplemente el servicio Web, lo cual marco un antes y un después en el desarrollo de la sociedad de la información. Su implementación hizo posible acceder a una gran cantidad de información desde cualquier lugar del mundo por medio de un navegador o *Web browser*, una aplicación que nos permite conectarnos a un servidor Web para descargar y visualizar las páginas almacenadas en éste.

No obstante, es hasta estas fechas que, mediante un soporte operativo tecnológico para la distribución de información en redes informáticas, Tim Berners-Lee propondría *Enquire Within Upon Everything* (Preguntando de todo sobre todo) al *Conseil Européen pour la Recherche Nucléaire* (CERN, siglas utilizadas en 1952 de la que después se transformaría en la Organización Europea para la Investigación Nuclear), con lo cual se materializa este concepto de incipientes nociones de la Web.

En marzo de 1989 Tim Berners-Lee, ya como miembro de la división del CERN, redactó la propuesta que hacía referencia a ENQUIRE y describió un sistema de gestión de información

más elaborado. Sin embargo, el World Wide Web ya había nacido. Meses después, el 12 de noviembre de 1990, Berners-Lee publicó una propuesta más formal para la World Wide Web en coautoría con Robert Cailliau; en ese mismo año ambos investigadores del Laboratorio de Física de Altas Energías del CERN, en Ginebra, Suiza, desarrollaron el primer navegador Web, mismo que publicaron en 1992. Desde entonces, Berners-Lee ha jugado un papel activo guiando el desarrollo de estándares Web, éste navegador era capaz de visualizar páginas que tuvieran distintos tipos y estilos de letras; sin embargo, aún no era capaz de visualizar imágenes, además su uso era bastante complicado y estaba restringido a un pequeño número de usuarios, entre los cuales se encontraban académicos e investigadores del mundo y empresas en EEUU. El primer servidor Web aparece en mayo de 1991 en el centro de Aceleración Lineal de Stanford, a principios de 1992 ya existían 26 servidores Web, y siguieron creciendo durante los siguientes años.

En febrero de 1993 la Web experimentó un notable avance gracias a la aparición del navegador gráfico *Mosaic* para sistemas Unix, el cual fue desarrollado por Marc Anderssen y Eric Bina del Centro Nacional de Aplicaciones para Supercomputadoras (NCSA), de la Universidad de Illinois. Entre las principales características de este navegador estaban, por una parte, la distribución libre y gratuita para entorno UNIX, y su facilidad de uso; por otra parte, a diferencia de los navegadores de la época, éste permitía agregar texto con formato incorporando imágenes, así como agregar enlaces de hipertexto para acceder a otras páginas con un sólo "clic", incluso se podían integrar pequeños archivos multimedia con sonido e imágenes en movimiento.

En el transcurso de ese año el uso de la *WWW* fue creciendo de una manera exponencial, no sólo en el número de clientes, sino también en el número de servidores. Éste crecimiento aumentó conforme se fueron viendo las ventajas que podía ofrecer el servicio Web a las organizaciones e instituciones que estuviesen conectadas a Internet, ya que con sus propios servidores, éstas podrían intercambiar información (documentos de trabajo, artículos, proyectos, software, imágenes, etc.) teniéndola siempre disponible al resto de los usuarios. Dado el éxito que tuvo el navegador *Mosaic*, el NCSA vendió la licencia a una empresa denominada Spyglass y meses después a Microsoft. El 30 de abril de 1993, el CERN anunció que la Web sería gratuita para todos, sin ningún tipo de honorarios.

Para 1994, Jim Clark desarrolla Netscape, un nuevo navegador Web mucho más potente ya que éste incorpora el protocolo Secure Socket Layer (SSL), o bien capa de conexión segura, lo cual permitió realizar transacciones seguras a través de Internet, pues se basaba en un mecanismo de

encriptación de clave pública. Posteriormente se desarrollan versiones de este navegador para UNIX y Microsoft para Windows, teniendo un éxito rotundo. Con la creación de éste nuevo navegador la tecnología Web alcanzó un alto grado de madurez, permitiendo que se abriera el paso a la creación de los comercios electrónicos, lo cual era una forma más de ofrecer servicios, no sólo de información sino también de comercio para las empresas, dando pie a un campo de competencia entre las mismas.

1.3. Ventajas y desventajas del servicio Web

El componente más usado en el Internet es definitivamente el Web. Hoy en día el servicio Web es uno de los medios más utilizados para el intercambio de información e ideas de cada uno de los participantes, quienes hacen sus propias aportaciones al gran universo de información. Los que acceden a éste servicio puede consultar información ofrecida por una infinidad de personas de todas partes del mundo, siendo ésta una de las principales ventajas de hacer uso del servicio. Aunque también tiene sus desventajas, una de ellas es que no podemos afirmar que toda la información alojada en la Web sea veraz y fiable, es por ello que debemos tener mucho cuidado con los contenidos que podemos encontrar en éste medio.

1.3.1. Ventajas

Entre las ventajas que nos ofrece el servicio Web podemos encontrar las siguientes:

a) **Interoperabilidad.**

La interoperabilidad entre diferentes plataformas nos permite que la interacción entre el proveedor y el solicitante del servicio sea independiente de la plataforma y del lenguaje que utilice. Ésta interacción la realiza por medio de un documento WSDL (Lenguaje de Descripción de Servicios Web), especifica la sintaxis y los mecanismos de intercambio de mensajes, para definir la interfaz y el servicio, junto con un protocolo de red (generalmente el HTTP). Todo esto es posible gracias al uso de protocolos y estándares abiertos, garantizándonos la plena interoperabilidad entre las aplicaciones.

b) **Accesibilidad.**

Hablar de accesibilidad Web es referir a una actualidad en la que contamos con un acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica, así como de las capacidades diferentes de cada uno de los usuarios; además, con la tecnología de dispositivos móviles podemos acceder a la Web no,

sólo desde computadoras de escritorio, también podemos acceder a éste por medio de laptops, PDAS, teléfonos móviles, e incluso desde juegos de video como PSP o Play station.

c) **Facilidad de uso.**

Gracias a las interfaces con las que contamos hoy en día, es muy fácil tener acceso a la Web y obtener una respuesta rápida y eficiente a nuestras dudas; podemos obtener tanto información documental como de medios audiovisuales (televisión y videos) para complementar la información solicitada.

d) **Difusión y colaboración de contenidos.**

A partir del servicio Web podemos difundir información y al mismo tiempo colaborar y aportar diferentes puntos de vista a la misma, un ejemplo de esto son los blogs y los foros.

Oportunidades de negocio. Abre nuevas oportunidades de negocio, ahora podemos efectuar compras online, transacciones bancarias, pagos de impuestos y más, todo ello sin salir de nuestros hogares o de trasladarnos de un lugar a otro.

Creación de redes sociales. Ofrece la oportunidad de socializar con personas de diferentes partes del mundo y de la propia localidad o lugar de trabajo de sin necesidad de salir de sus hogares.

Centralización de Información. Para las empresas es de gran ayuda, puesto que pueden consultar los contenidos de la misma desde cualquier lugar del mundo y actualizarlos en tiempo real, siempre y cuando cuenten con los privilegios necesarios para tener acceso a ella, así como para poder modificarla.

1.3.2. Desventajas

Como todo medio de comunicación, el sistema Web también tiene desventajas, las cuales son las siguientes:

a) **Seguridad.** Éste es uno de los más importantes, ya que es un servicio basado en el protocolo de transferencia de hipertexto sobre TCP, en éste rubro mencionaremos algunas:

- La realización de las transacciones no es muy confiable puesto que depende del desarrollador de la aplicación.
- Al apoyarse en HTTP, se pueden esquivar medidas de seguridad basadas en *firewall,* cuyas reglas tratan de bloquear o auditar la comunicación entre programas en ambos lados de la barrera.

- Robo de información, pero de igual manera dependerá del responsable que suba su información al sistema Web, colocando los derechos reservados de la misma o bien los permisos adecuados para que puedan hacer uso de ella.
- Es responsabilidad del usuario de las redes sociales subir información sensible que pueda ser utilizada para lucrar con su persona, como por ejemplo: direcciones, teléfonos personales, posición socioeconómica, fotografías personales y familiares, etc. Puesto que hoy en día secuestradores y asaltantes hacen uso de esta información para seleccionar a sus víctimas.

b) Rendimiento.

Su rendimiento es muchísimo menor en comparación con otros sistemas de computación distribuida ya que depende de la aplicación, la eficiencia del procesamiento de la misma sea porque se cuenta con animaciones o imágenes muy grandes en tamaño, o bien hacen uso de audio.

1.4. Definición del Proyecto

Uno de los principales objetivos de UNICA es brindar el servicio Web en conjunto con el de bases de datos para algunas aplicaciones del mismo, brindar información a todo aquel interesado o bien que esté involucrado con la Facultad de Ingeniería. Estas aplicaciones Web se encuentran alojadas dentro de uno de los servidores de UNICA, a cargo del departamento de Redes y Operación de Servidores (DROS), en esos servidores también alojamos el principal portal Web de la institución (*www.ingenieria.unam.mx).* La información que se ofrece en este portal es de suma importancia por lo que es vital tenerla siempre disponible al público en general.

1.5. Objetivo

1.5.1. Objetivo General

Mejorar los servicios de Web que ofrece UNICA para los diferentes usuarios que tienen acceso al servicio.

1.5.2. Objetivos Particulares

1. Mejorar los servicios de *hosting*, en cuanto al espacio disponible que se le dará a cada uno de los usuarios, así como también hacer un monitoreo de los mismos para verificar que en realidad están aprovechando dicho servicio.

2. Obtener estadísticas sobre el rendimiento del servidor y su flujo de conexiones, sobre las peticiones realizadas y rechazadas, y en relación con el uso de las cuentas de usuarios para la depuración de las mismas. Una vez conociendo estos datos estadísticos podemos realizar varias modificaciones en cuanto a la calidad del servicio que brinda nuestro servidor Web y repercutir en los sitios Web que contiene.
3. Mejorar el sistema de tolerancia a fallos para que en cualquier contingencia sufrida en el servidor principal podamos contar con otro que contenga la información actualizada y de éste modo los sitios contenidos se encuentren siempre disponibles.
4. Implantar herramientas para la automatización de las tareas y los procesos de administración del servidor, como son: la creación y desactivación de cuentas de usuarios, y la realización de respaldos tanto de las bases de datos como de la información de los sitios alojados.
5. Utilizar nuevas herramientas de seguridad para aumentar su robustez.
6. Aumentar la alta disponibilidad del servicio considerando que un sitio Web institucional debe estar disponible los 7 días de la semana, las 24 horas del día, incluyendo los períodos vacacionales y los fines de semana, en resumen los 365 días del año.
7. Recomendar buenas prácticas para los desarrolladores de estos sitios a fin de que las tomen en cuenta. Así como algunos criterios mínimos que deben seguir quienes hacen uso y acceden a sus sitios, y quienes hacen uso del servicio de alojamiento de sus sitios en el servidor.
8. Realizar las actualizaciones pertinentes al manual del administrador del servidor, con el objeto de facilitar el manejo de los futuros administradores.

1.6. Alcance

El alcance de esta tesis consiste en mejorar de manera radical el servicio Web, de tal manera que cambiemos los paradigmas utilizados anteriormente de viejos conceptos del servicio, olvidándonos de los procesos pasados para realizar nuevos procesos con nuevas ideas, tomando todas las partes que componen éste servicio, dándole una nueva forma que nos ayude a brindar un mejor servicio, y con lo anterior lograr una mejora significativa.

1.7. Problemática actual

1.7.1. Definición del Problema

1.7.1.1. Escenario

Como ya se mencionó, el portal de la Facultad de Ingeniería está alojado específicamente en uno de los servidores de UNICA, a cargo del Departamento de Redes y Operación de Servidores. Esos servidores albergan además otras páginas que tienen que ver directamente con la Facultad, asociadas a profesores, asociaciones o sociedades de la misma, así como de alumnos, exalumnos y tesistas de la Facultad de Ingeniería.

Lo que se pretende realizar, es mejorar los servicios sin perder de vista su utilidad y funcionamiento, con el fin de lograr un cambio significativo en un corto período de tiempo, en cuanto a la productividad, tiempo de respuesta, calidad y disponibilidad.

1.7.1.2. Justificación

Hace mucho tiempo que no se ha hecho una mejora significativa a los servicios, ni al sistema operativo, ni a la seguridad del mismo. Esto último es una parte muy importante ya que hoy en día existe una cantidad enorme de vulnerabilidades de los propios sistemas o bien de los servicios de los sistemas, de las cuales no nos damos cuenta fácilmente; todo esto previendo que suceda de una manera activa y no reactiva.

1.7.1.3. Identificación de los procesos y servicios

Para identificar los procesos específicos que agregan valor, mencionaremos los servicios con los que cuenta hasta el momento, así como los procesos que se llevan a cabo de manera muy general, los cuales son:

a) Servicios:

- Servidor apache.
- Sistema manejador de bases de datos de PostgreSQL.
- PHP.

b) Procesos:

- Administración de usuarios y procesos.
- Respaldo de la información.
- Seguridad del servidor.

En la actualidad todo esto se hace manualmente, por ello pretendemos automatizar dichas tareas, además de que no se cuentan con estadísticas de los usuarios del mismo, ni del rendimiento y conexiones del servidor, por tanto buscamos que la administración se automatice para facilitar la tarea a sus administradores.

1.8. Requerimientos

Como ya se dijo anteriormente, éste servicio Web es brindado a varios usuarios dentro y fuera de la Facultad, pero para esto debemos tomar en cuenta las necesidades que cada uno de ellos requiere y clasificarlos en diferentes tipos de usuarios. De acuerdo a lo anterior, las necesidades de los usuarios pueden describirse de la forma siguiente:

Mejorar los servicios de *hosting* en relación con el espacio disponible que se le da a cada uno de los usuarios, así como también hacer un monitoreo de los mismos para verificar que en realidad están aprovechando dicho servicio y no se esté desperdiciando espacio disponible para algún otro usuario. Por otra parte, también se cuenta con el servicio de bases de datos, el cual se pretende que en cualquier contingencia que sufra el servidor principal y tenga que cambiarse por el servidor *mirror* se encuentre dicha base de datos siempre actualizada y sincronizada con la principal, para que no haya la necesidad de recargar dichos datos; es decir, hacer más autónoma ésta función.

Todo lo anterior va dirigido a un usuario en específico, el cual lo nombraremos "usuario del sistema", ya que estos usuarios mantienen el desarrollo de sus sitios dentro del servidor y, por lo tanto, tienen un contacto directo con los mismos (ya sea vía *Secure Shell*, un navegador o bien con conexiones a las bases de datos). Por ello es de gran importancia mantener un especial cuidado en la administración de sus recursos.

Además, tenemos a otros usuarios que también son beneficiados por nuestro servidor, los cuales denominaremos "usuarios www"*,* siendo éstos los usuarios que sólo realizan conexiones mediante el navegador. A diferencia de los usuarios del sistema, los "usuarios www" no cuentan con un usuario dentro del servidor. En éste sentido es necesario recordar que alojamos sitios Web institucionales y debemos contemplar las características de la comunidad a la que están dirigidos. Por lo cual se realizarán recomendaciones y buenas prácticas a los desarrolladores de éstos sitios, para que tomen en cuenta algunos aspectos mínimos para aquellos que hacen uso y acceden a sus sitios, los cuales son:

- Características promedio de las computadoras que posee su comunidad (en éste caso la Facultad de Ingeniería).

- Ancho de banda de Facultad de Ingeniería.

Por último, tenemos que tomar en cuenta algunas mejoras en la administración del servidor y sus servicios, tratando de automatizar dichas tareas y procesos, o bien haciéndolas más fáciles y sencillas para quien se encargue de éstas. Dentro de estas tareas debemos considerar: la administración de cuentas de usuarios y creación de las mismas, así como su desactivación; la creación de bases de datos; la obtención de estadísticas del servidor en cuanto a rendimiento; cantidad de tráfico relacionado a nuestro dominio; así como los respaldos de la información contenida en el mismo.

Otro punto muy importante es aumentar la disponibilidad y la robustez de la seguridad del servidor, además de la sincronización con un servidor *mirror,* el cual contendrá la información más actualizada del servidor principal. Todo lo anterior va enfocado a los "usuarios administradores" del servidor, quienes son los encargados de llevar a cabo las tareas de administración de los usuarios, los procesos y el mantenimiento de los mismos.

1.8.1. Infraestructura

Es importante tener en cuenta las características actuales del servidor, para tomar las consideraciones necesarias en la instalación del sistema, y de los servicios que proporcionará. Las características del servidor con el que contamos son (Tabla 1.8.1):

Tabla 1.8.1. Características del Servidor Hp Proliant ML350.

Características del Servidor Hp Proliant ML350	
Procesador	Intel(R) Xeon(TM) CPU 3.00GHz
Número de procesadores	2 Procesadores
Bus del sistema	Front Side Bus a 800 MHz
Memoria caché Interna	1 MB de caché de segundo nivel
Memoria estándar	1GB de memoria
Tipo de memoria	SDRAM DDR PC2700 (333 MHz))

Unidad de disco duro Interno	Dos unidades ultra320 SCSI de 150GB
Velocidad de la unidad de disco duro	10,000 rpm
Controlador de disco duro	Adaptador Ultra320 SCSI de dos canales; controlador Smart Array 641
Unidad de discos flexibles	Floppy de 1,44 MB
CD-ROM/DVD	Unidad de CD-ROM IDE (ATAPI) 48x
Tipo de unidad óptica	CD-ROM

1.8.2. Beneficios

Los beneficios que obtendremos al realizar el rediseño del servicio serán los siguientes:

1. Mayor disponibilidad del servicio Web.
2. Contar con información estadística del uso y abuso del servidor, datos con los que anteriormente no se contaba.
3. La automatización de tareas administrativas permitirá ahorrar tiempo y esfuerzo, que se podrá utilizar para otras tareas dentro del departamento.
4. Mejor tiempo de respuesta y procesamiento.

1.9. Historia del portal de la Facultad de Ingeniería

El portal de la Facultad de Ingeniería surgió de la necesidad de tener un sitio alterno en donde los académicos, alumnos y personal que tuviera que ver de alguna manera con la Facultad, pudiera tener acceso a información referente a ésta. Este portal fue uno de los primeros en implementarse después de la creación del portal de la UNAM en el año de 1993.

El primer sitio Web de nuestra Facultad, desarrollado por un ingeniero geofísico de la Facultad en 1994, estuvo montado bajo un servidor HP-UX bajo UNIX Enterprise de UNICA. Este portal era muy rústico en sus inicios, ya que consistía únicamente en un texto plano parecido a una terminal de Secure Shell, y se accedía a ésta con los primeros navegadores que se tenían como lo son Mosaic y Netscape. En octubre de 1999, por órdenes de la Secretaría General de la UNAM, el portal fue alojado en un servidor SUN con sistema operativo Solaris; ésta decisión fue tomada mientras la Facultad atravesaba por una huelga estudiantil para que se tuviera un sitio alternativo de información para todos los alumnos y profesores.

El segundo sitio Web ya contaba con algunos gráficos, y un año después quedó bajo la administración de UNICA, a cargo del Webmaster Víctor Manuel Duran, Claudia Cordero y Gustavo Sibaja. Una vez que las dependencias de la misma Facultad vieron que era un medio muy fiable para compartir información, también comenzaron a realizar sus propias páginas a partir del año 1997, las cuales fueron montadas en el mismo servidor.

Para 2003 se cambió el sistema Operativo de Solaris a Red-Hat, y finalmente, debido a que se tenía que pagar por la licencia de Red-Hat, se cambió a un sistema operativo con licencia GPL, o libre, llamado Fedora.

A partir de 1994, el área responsable de la página Web de la Facultad de Ingeniería, es la Secretaría General. Como parte integrante de Secretaría General, la Unidad de Servicios de Cómputo Académico (UNICA), sería la responsable técnica de la operación, seguridad y respaldos; el Departamento de Información y Estadística sería la responsable de los contenidos, y la Coordinación de Comunicación de la imagen gráfica institucional. La página de la Facultad de Ingeniería sería el punto de partida para acceder a las demás páginas de Secretarías y Divisiones. Quedó estrictamente prohibido que las páginas de otras Divisiones o Secretarías que no fuera la de la Secretaría General, dieran la impresión de representar a la Facultad de Ingeniería en forma oficial.

El Departamento de Información y Estadística tiene las siguientes funciones en la Web principal de la Facultad de Ingeniería:

- Planear el contenido de la página principal de la Facultad de Ingeniería, y establecer la estructura de las páginas.
- Dar de alta las ligas de las páginas que tengan el visto bueno de la Secretaría General, entre las que se incluyen las de las Divisiones, profesores, asociaciones, tesístas, alumnos etc., y cualquier página que desee depender de la página principal.
- Dar la ubicación de las nuevas páginas dentro de la estructura de la página principal.
- Actualizar y mantener el contenido de las páginas en la Web principal de la Facultad de Ingeniería.
- Administrar las cuentas de correo y la Web de *fainge y webmaster* (fainge@cancun.fi-a.unam.mx y webmaster@cancun.fi-a.unam.mx).
- Deshabilitar las páginas que rompan con la normatividad de la Web.

La Unidad de Servicios de Cómputo Académico tiene las siguientes funciones en la Web principal de la Facultad de Ingeniería:

- Mantener la operación física del servidor de la Web.

- Instalar, mantener y actualizar el sistema operativo del servidor
- Instalar, mantener y actualizar el software de administración de la Web.
- Implementar la seguridad del sitio Web y el sistema operativo, así como la revisión de bitácoras.
- Instalar las aplicaciones requeridas para la operación del sitio Web.
- Realizar los respaldos del servidor Web.
- Asesorar técnicamente a las áreas de la Facultad en tópicos relacionados con la Web.
- Proponer innovaciones de tecnológicas para crear servicios de la Web en beneficio de la Facultad (como el caso de la Biblioteca Digital, Búsqueda, Bolsa de trabajo, etc.)
- Actualizar y mantener el área de la Web correspondiente a UNICA.

La Coordinación de Comunicación tiene las siguientes funciones en la Web principal de la Facultad de Ingeniería:

- Diseñar la imagen gráfica institucional.
- Crear, diseñar, y recabar imágenes (fotos, backgrounds, íconos etc.) que se usarán en el sitio Web.

Para el año 2003 el ingeniero Alejandro García Romero, entonces alumno que realizaba su servicio social y el ingeniero César Osvaldo Pereida Gómez crearon el penúltimo diseño, el cual perduró hasta a mediados del año 2009. Para realizar la última versión de la página, se formó un comité en 2009 para en nuevo díselo del sitio.

1.10. Evolución del Portal de la Facultad de Ingeniería

El Portal ha ido evolucionando constantemente tanto en su aspecto gráfico como en el sistema en el cual está montado. Actualmente, tiene un sistema operativo de libre licencia, la distribución de Linux Fedora 9, y la última liberación de la página con un nuevo diseño fue en agosto del 2009 (la cual fue diseñada en conjunto con la Coordinación de Comunicación con la ayuda de un diseñador gráfico) (Tabla 1.10.1, Figura 1.10.1 y Figura 1.10.2).

Tabla 1.10.1. Línea de tiempo del portal de la Facultad de Ingeniería.

Evolución del Portal de la Facultad de Ingeniería	
Año	***Evolución***
1994	Primera página del portal de la Facultad de Ingeniería; S.O del Servidor HP-UX de UNICA.
1995	Se delega la administración del portal a la Unidad de Servicios de Computo Académico (UNICA).
1997	Las Divisiones de la Facultad de Ingeniería comienzan a crear sus propios sitos Web.
1999	El Portal es alojado en un equipo Sun con S.O Solaris.
2000	La página de la Facultad de Ingeniería pasa a manos del Departamento de Información y Estadísticas, mientras que la administración del servicio sigue bajo la responsabilidad de UNICA.
2003	Se crea la penúltima versión de la página, la misma estructura se mantuvo hasta mediados del 2009, y se cambia de S.O Solaris a Red-Hat.
2005	Se cambia S.O de Red-Hat a Linux Fedora.
2009	10 de agosto, se libera la última versión de la página.

Figura 1.10.1. Línea del Tiempo de la evolución del Portal de la Facultad de Ingeniería

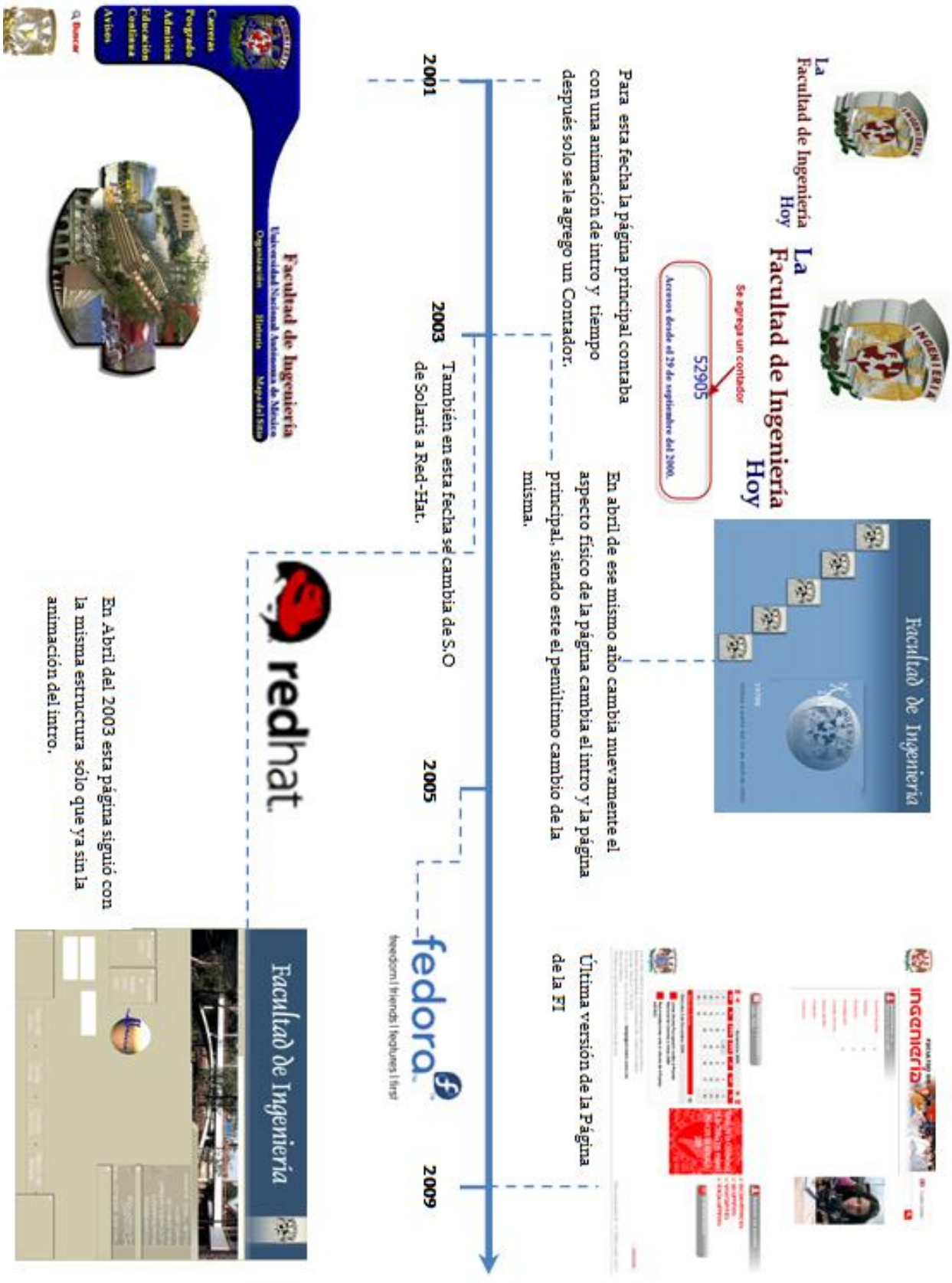


Figura 1.10.2. Línea del Tiempo de la evolución del Portal de la Facultad de Ingeniería

CAPÍTULO 2

PROTOCOLOS Y SOFTWARE EN EL SERVICIO WEB

Las primeras redes de computadoras fueron diseñadas, para tener al hardware como punto principal y al software para tenerlo como secundario para su comunicación. Hoy día es todo lo contrario, es decir, para que llevemos una correcta comunicación entre una máquina y otra sin tener problemas de compatibilidad, entre sus diferentes dispositivos interconectados en la red se hace uso de un software como elemento secundario, el cual es el encargado de llevar a cabo dicha comunicación, sin que el usuario se dé cuenta de cómo lo hace.

2.1. Protocolo de comunicación

Un protocolo es un conjunto de reglas y convenciones entre las partes para comunicarse unas con otras a través de una red. Un protocolo es un método estándar que permite la comunicación entre procesos, es decir, es un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red, entre los diferentes dispositivos. Los protocolos pueden ser implementados por hardware, software, o una combinación de ambos. A su más bajo nivel, un protocolo define el comportamiento de una conexión de hardware.

Un protocolo debe aportar las siguientes funcionalidades:

- Permiten localizar una computadora o cualquier dispositivo de red de forma inequívoca.
- Permitir intercambiar información entre computadoras de forma segura, independientemente del tipo de máquinas a las que estén conectadas (PC, Mac,AS-400 y más).
- Detección de la conexión física subyacente (con cable o inalámbrica), o la existencia de otro punto final o nodo para el intercambio de información.
- Handshaking o apretón de manos, el cual es la parte inicial del protocolo en el que dos máquinas se ponen de acuerdo sobre el formato, velocidad y secuencia que seguirán en el resto de la comunicación.
- Como iniciar y finalizar un mensaje.
- Procedimientos en el formateo de un mensaje.
- Que hacer con mensajes corruptos o formateados incorrectamente (corrección de errores).
- Detectar una pérdida inesperada de la conexión, y qué hacer entonces.
- Encapsulamiento.
- Ruteo.
- Permitir liberar la conexión de forma ordenada.

En la actualidad estos protocolos son regulados y aprobados por comités quienes se encargan de evaluar dichos protocolos, para aceptarlos como un estándar y así todas las telecomunicaciones puedan hablar un mismo lenguaje y entenderse de manera independiente y controlada, de tal manera que lleva acabo todo un control en la transmisión y emisión de la información entre los entes participantes de dicha comunicación.

2.1.1. Jerarquía de protocolos

Para llevar a cabo el intercambio de información entre diferentes equipos o dispositivos debemos tomar en cuenta varios aspectos:

- Conexiones físicas como los aspectos eléctricos: los tipo de cable como UTP, coaxial, fibra óptica, las señales, los conectores y más.
- La forma de agrupar los bits para formar paquetes y controlar que no se produzcan errores de transmisión en el intercambio de la información.
- Cómo identificar los equipos o dispositivos dentro de la red.
- Conseguir que la información que genere un equipo llegue a quien se pretende.

Como ya vimos, lo anterior es parte de lo que controlan los protocolos, y atacar todos éstos aspectos de manera global es muy difícil, es por eso que se desarrollaron modelos estructurados en niveles o capas donde cada uno de éstos niveles se ocupan de tareas muy específicas, además que la cooperación de todos los niveles proporciona la conectividad que desea el usuario.

Debido a la gran complejidad que conlleva la interconexión de computadoras y dispositivos, se han tenido que dividir todos los procesos necesarios para realizar las conexiones, en diferentes capas o niveles para reducir la complejidad del diseño de estas redes. Cada capa se ha creado para dar una solución a un tipo de problema particular dentro de la conexión. Cada capa tendrá asociado un protocolo, el cual entenderá todos los elementos que formen parte de la conexión, así como cada una de estas capas estará construida a partir de la que está debajo de ella. El propósito de cada capa, es ofrecer un cierto servicio a las capas superiores, sin que éstas se den cuenta a detalle de cómo es que les llega éste servicio, es decir les oculta información del proceso que lleva para ofrecer dicho servicio. Cuando se lleva a cabo una conversación entre dos PC o bien dos dispositivos, tenemos que la "*capa n"* de una máquina o dispositivo, mantiene dicha conversación con la *"capa n"* de la otra máquina, comunicándose mediante sus propias reglas entre éste par de capas abarcadas por entidades que participan en la comunicación, dónde dichas entidades podrían ser procesos, dispositivos de hardware o incluso seres humanos los cuales se están comunicando mediante un protocolo.

Debemos tomar muy en cuenta, que en la vida real, cuando se realiza la comunicación entre dos dispositivos, la información no pasa de manera directa de la capa n de una máquina a la capa n de la otra, ya que éstas van pasando la información a las capas inmediatamente

inferiores, hasta que llegan a la capa más baja, llegando el momento en que encontramos el medio físico, siendo éste el momento en que encontramos la comunicación real entre una máquina y la otra, pues como ya mencionamos se pasa la información hacia abajo de un nivel a otro hasta que llega al nivel que es el medio físico, como se muestra adelante en la Figura 2.1.1.

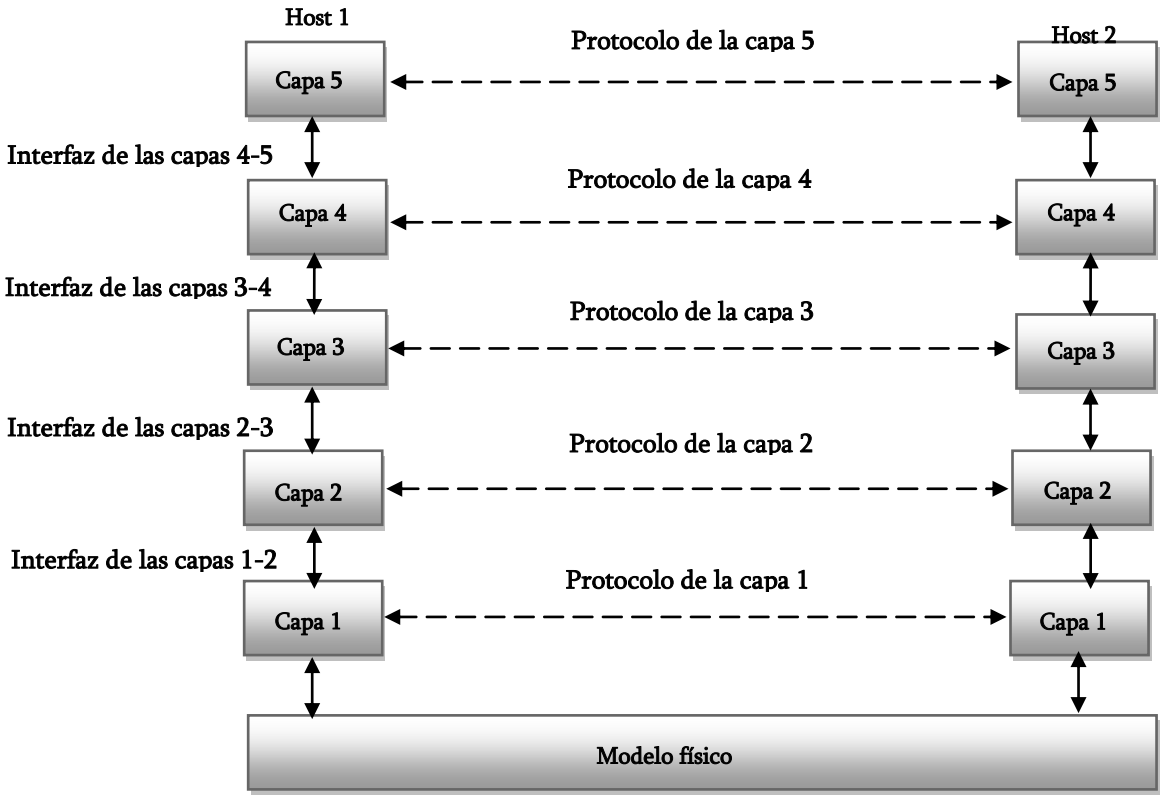


Figura 2.1.1. Jerarquía de protocolos.

Entre los niveles están las interfaces, estas interfaces definen que operaciones y servicios pone la capa inferior a la capa superior inmediata. Las interfaces limpias permiten cambios en la implementación de un nivel, sin afectar el nivel superior, minimizando la cantidad de información que pase entre las capa, además, el uso de estas interfaces, nos simplifican la implementación de una mayor cantidad de capas en nuestra red, haciéndola menos compleja teniendo la versatilidad de que cada quien pueda tener implementaciones diferentes de red, siempre y cuando conservemos que se ofrezca el mismo servicio entre su vecino de arriba como al de la implementación anterior.

2.2. Modelo de referencia OSI

En la actualidad las redes de información han ido creciendo de una manera exponencial, siendo estas implementadas con una gran variedad de infraestructura, tanto de hardware como de software, esto dependiendo de las necesidades de cada una de las diferentes redes, con lo que surgen problemas con la interoperabilidad entre estas redes, siendo incompatibles, es por eso que lo solucionaron creando sistemas abiertos (**open system**). Un sistema abierto debe utilizar componentes informáticos basados en estándares independientes del proveedor, los cuales nos permiten portabilidad para pasar de un sistema a otro sin necesidad de hacer cambios, así como también nos permiten la interoperabilidad entre las distintas redes y puedan trabajar entre sí sin cambios.

En el diseño de un sistema abierto se deben considerar varios aspectos, entre los que encontramos los siguientes:

- Contemplar estándares del sistema operativo utilizado.
- Contemplar estándares en el sistema de comunicaciones utilizado.
- Bases de datos.
- Sistemas de administración.
- Herramientas de desarrollo
- Interfaz de usuario.

Para solucionar todos estos problemas la Organización Internacional de Estándares (ISO) realizó varias investigaciones en cuanto a los diferentes esquemas de red. La ISO reconoció que era necesario crear un modelo de red, que pudiera ayudar a los diseñadores de red a implementar redes que pudieran comunicarse y trabajar en conjunto (interoperabilidad) y por lo tanto, elaboraron el modelo de referencia OSI (Interconexión de Sistemas Abiertos) en 1984 el cual consta de 7 capas que se muestra en la Figura 2.2.1 donde cada una de éstas se encarga de tareas muy específicas dentro de la emisión y recepción de la información en la red. El objetivo principal del modelo OSI, es que las compañías o bien todo aquel que cuente o que esté conectado a una red no se encuentren obligados a ligarse con ningún dispositivo de red en específico, así como también las compañías que crean los dispositivos de red, desarrollen productos para cada nivel del modelo OSI, para que no se trabaje sólo con una marca en particular y las redes sean compatibles independientemente del dispositivo que se esté usando.

Los principales principios que se aplicaron para construir estas capas son:

1. Una capa se debe crear donde se necesite una abstracción diferente.
2. Cada capa debe realizar una acción bien definida.
3. La función de cada capa se debe elegir con la intención de definir protocolos estandarizados internacionalmente.
4. Los límites de las capas, se deben elegir a fin de minimizar el flujo de información a través de las interfaces.
5. La cantidad de capas debe ser lo suficientemente grande para no tener que agrupar funciones distintas en la misma capa y lo bastante pequeña para que la arquitectura no se vuelva inmanejable.

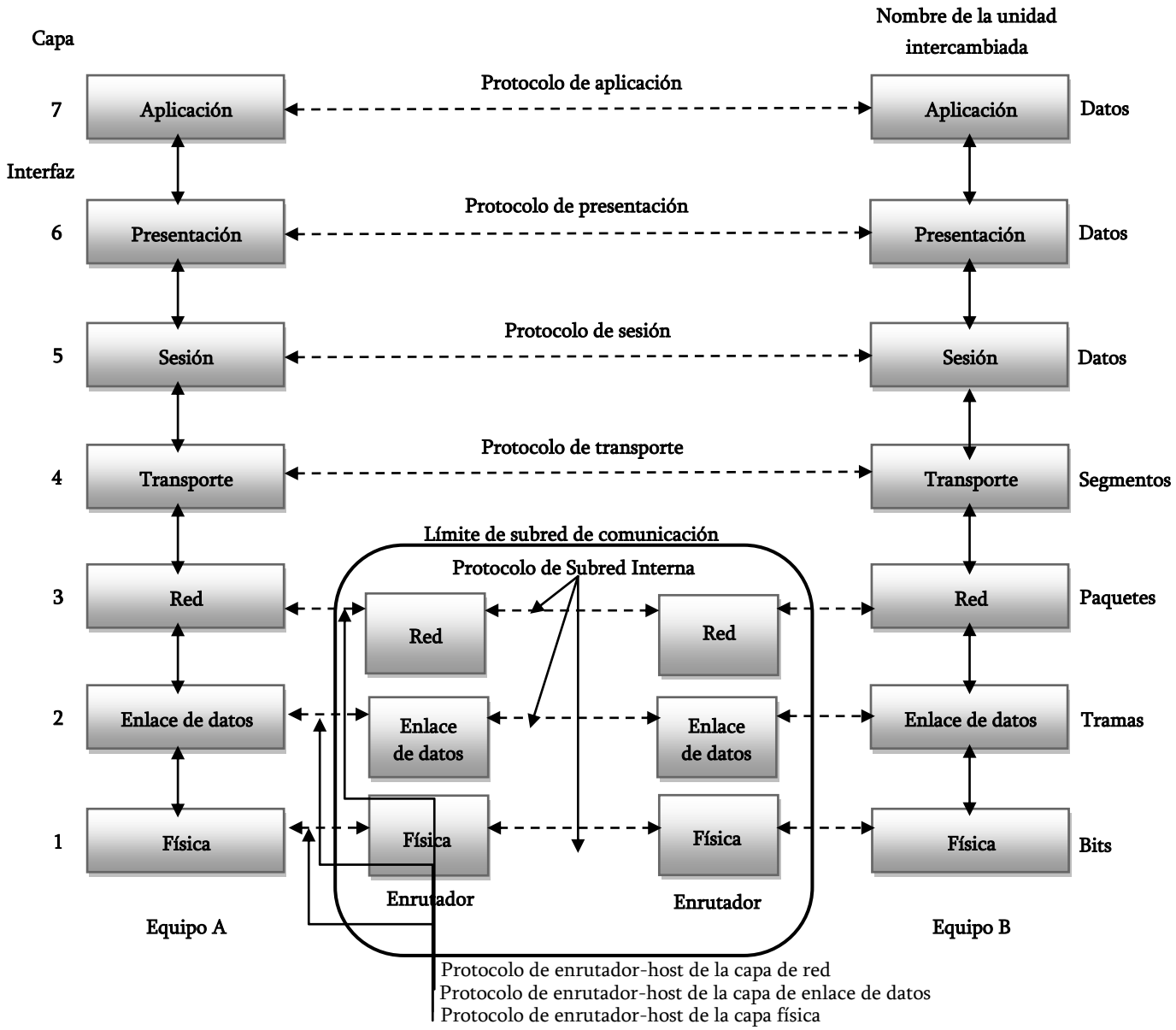


Figura 2.2.1 Modelo OSI.

2.2.1. Niveles del Modelo OSI

Como ya observamos en la Figura 2.2.1, el modelo OSI cuenta con siete capas o niveles donde cada uno de ellos se encarga de tareas y funciones específicas dentro de la comunicación, pasando la información por todas y cada una de estas capas. Cada nivel del modelo OSI es independiente, pero cabe mencionar que el siguiente nivel no funcionara si el nivel anterior jamás arranca, entre cada capa encontramos una interfaz la cual es la que nos define las operaciones y servicios con los que cuenta el nivel inferior, para ofrecerle dichos servicios al nivel superior.

Mencionaremos las actividades principales que definen a cada capa del modelo OSI:

1. La capa física

Esta capa se encarga de controlar las interfaces físicas como las mecánicas, eléctricas y de velocidades de datos físicos así como del medio físico por donde es enviada la información, como los conectores, tarjeta de red, el medio de trasmisión ya sea guiados como cable coaxial, fibra óptica, UTP, STP o medios no guiados como satélite, microondas, infrarrojo, ondas de radio, rayo láser y más. Esta capa es el nivel más bajo dentro del modelo OSI, por que como ya se menciono tiene que ver con las interfaces físicas entre dispositivos y las reglas bajo las cuales las cadenas de bits son transmitidas de un dispositivo a otro en un canal de comunicación, en ésta capa es donde se lleva a cabo la trasmisión real de los datos a través de un medio físico. Entre los dispositivos de capa 1 encontramos el Hub ó Repetidor.

2. La capa de Enlace de Datos

Esta es la capa lógica que se encarga de llevar a cabo una comunicación efectiva, confiable y libre de errores, controlando y gestionando el intercambio de datos entre los entes participantes, llevado mediante un control lógico fragmentando la información en tramas de manera secuencial. Entre sus principales tareas de control encontramos:

a) *Sincronización de tramas.*- Las entidades que se comunican se sincronizan para recibir y enviar las tramas de una manera ordenada.
b) *Control de flujo.*- Antes de llevar a cabo el intercambio de información los entes que participan en dicha comunicación deben ponerse de acuerdo en el tamaño y la señalización del flujo de las tramas.
c) *Control de errores.*- Se encarga de verificar que la información haya llegado de manera correcta y completa es decir, que la información enviada sea igual a la información recibida de manera intacta.
d) *Direccionamiento.*- Debe controlar y de saber quién es el ente que recibe los datos y quien los está enviando.
e) *Verificación.*- Verificar que los datos más la información de control se encuentren en el mismo enlace.
f) *Gestión de enlace.*- Se encarga de administrar la forma en la que se está llevando a cabo el enlace.

Entre los dispositivos de capa 2 tenemos:

a) **Switch.** La red deja de presentar grandes retardos y aumenta su productividad.
b) **Puentes (Bridges).** Sirven para unir dos segmentos de red de una LAN, determinan la dirección fuente y la dirección destino de la información.
c) **Tarjetas de Red (NIC).** Permite interconectar a las máquinas con el medio de transmisión correspondiente, reconoce direcciones MAC, traduce la señal producida por la computadora en el formato que se envía.

3. La capa de Red

La capa de red se encarga de encaminar los paquetes desde su origen hasta su destino, seleccionando la mejor ruta posible entre host que pueden estar ubicados en redes geográficamente distintas. Este encaminamiento lo hacen con ayuda de algoritmos de enrutamiento estáticos o dinámicos, los cuales se encargan de seleccionar la mejor ruta, queriendo decir con mejor ruta con que ésta se encuentre disponible, que sea confiable y rápida, esto con el propósito de evitar los cuellos de botella y ofrecer un mejor servicio.

En esta capa tenemos el protocolo de transmisión como el IP (Protocolo de Internet), para llevar a cabo la trasmisión de los datos a través de paquetes conmutados.

Dentro de los dispositivos de capa 3 encontramos al Router, que es un dispositivo que hace uso de los algoritmos de enrutamiento para encaminar los paquetes, haciendo uso de tablas de ruteo, además de ser un dispositivo inteligente que aprende conforme va pasando la información a través de él, aprendiendo las direcciones de origen y destino de donde desea enviar o recibir la información.

4. La capa de Transporte

Esta capa se encarga de aceptar los datos de las capas superiores, segmentarlos y dividirlos en unidades más pequeñas, pasarlas a la capa de red con una secuencia y asegurándose de que lleguen a su destino de manera ordenada, para poder reconstruir nuevamente de manera correcta la información.

Esta capa también de encarga de determinar qué servicio va a ofrecer a las capas de sesión y a los usuarios de red. Entre los protocolos de red encontramos en esta capa a el protocolo TCP

(Protocolo de Control de Trasmisión), el cual es orientado a la conexión y por lo tanto asegura que llegue la información en el mismo orden en que fueron enviados y sin errores y UDP (Protocolo de Datagrama de Usuarios) que es no orientado a la conexión y es lo contrario al TCP.

5. La capa de Sesión

Esta capa permite el establecimiento de sesiones entre usuarios que se encuentran en máquinas diferentes sincronizando el diálogo y el intercambio de información; las sesiones ofrecen varios servicios, entre los que encontramos el *control del diálogo* para saber a quién le corresponde trasmitir y saber si es una trasmisión en los dos sentidos o sólo en uno (full-duplex o half-duplex) y en qué momento activar la *administración del token,* para que no traten de realizar la misma acción u operación al mismo tiempo y la *sincronización,* en general se encarga de controlar el diálogo entre los dispositivos.

6. La capa de Presentación

Esta capa se encarga de garantizar que la información que viene de la capa de aplicación de un equipo, sea comprendida por la capa de aplicación de otro, es decir esta capa se encarga de la sintaxis y la semántica de la información transmitida independientemente de las diferentes representaciones de datos con los que cuenten los dispositivos de red, tiene su analogía con un traductor ya que traduce los datos de un equipo y otro, para que éstos hablen un mismo lenguaje, ya que codifica y le da formato a los datos.

7. La capa de Aplicación

Es la capa que se encuentra más cercana al usuario, pues no hace contacto directo con esta capa si no con las aplicaciones o programas, que a su vez interactúan con la capa de aplicación, pues ésta se encarga de las aplicaciones de red que se requieren usar para transportar las aplicaciones del usuario. En esta capa encontramos protocolos de aplicación con los cuales el usuario interactúa para acceder a la información, entre los que encontramos:

- HTTP (HyperText Transfer Protocol).
- FTP (File Transfer Protocol).
- SMTP (Simple Mail Transfer Protocol): envió y distribución de correo electrónico.
- POP (Post Office Protocol): reparto de correo al usuario final.

- SSH (Secure SHell): terminal remoto, cifra cualquier tipo de transmisión.
- Telnet: terminal remota.
- SNMP (Simple Network Management Protocol).
- DNS (Domain Name System).

2.3. Modelo de referencia TCP/IP

Internet es un conjunto de redes de manera mundial, para que ésta logre funcionar y comunicarse de manera correcta con las demás redes hace uso del modelo TCP/IP (Protocolo de Control de Transmisión/Protocolo de Internet), el cual es el idioma que hablan los dispositivos cuando se están comunicando a través de internet.

El nacimiento de TCP/IP comienza en los años 60 cuando el Departamento de la Defensa de Estados Unidos (DoD, U.S) se empezó a preocupar porque su sistema de red sólo contaba con un sistema de telefonía o conmutación de circuitos, el cual era demasiado frágil y poco fiable, pues el enemigo podría tirar dichas centrales telefónicas. De tal manera que en 1969 Bolt, Beranek y Newman se encargaron de diseñar una red experimental llamada ARPANET la cual estaba respaldada por el DoD y cuyo propósito era construir una red, de tal manera que aseguraran que la información llegara a su destino sin importar si en algún momento parte de la red quedara destruida. Éstos lo realizaron haciendo uso de la tecnología de conmutación de paquetes, donde más tarde no sólo se conectarían instituciones militares, sino también universidades para compartir información con investigadores permitiéndoles trabajar en lugares geográficos muy distantes entre sí.

Ya que los investigadores necesitaban trabajar en lugares distantes y compartir información para un mismo proyecto, en el año de 1972 con el protocolo NCP (Network Control Protocol, que busca el intercambio entre equipos) se presenta públicamente un nuevo servicio que posteriormente se conocería como correo electrónico, con éstos haciendo que la red creciera de una manera considerable donde tiempo después y a partir de las nuevas necesidades de los investigadores de compartir información mediante el correo electrónico, nace un protocolo llamado TCP o Protocolo de Control de Transferencia, al cual aún le faltaba que madurara un poco más, dividiendo sus tareas y funciones en dos protocolos IP (Protocolo de Internet) el cual sería el protocolo de encaminamiento dinámico entre nodos y TCP el cual se encargaría de garantizar los servicios, y éstos dieron origen a un nuevo protocolo UDP (Protocolo de Datagrama de Usuario) ofreciendo un servicio sin garantía. Con la unión de estos protocolos a

principios de los 80's dieron origen a TCP/IP, dejando atrás a NCP y con ello en enero de 1983, el DoD de los Estados Unidos decidió usar el protocolo TCP/IP en la red de ARPANET, convirtiéndose en una red civil, donde tiempo más tarde, con el crecimiento de la misma de manera exponencial, evolucionaría con el nombre conocido comúnmente por la sociedad civil de la red como **"Internet"**.

Cabe destacar que el crecimiento de internet vino de la mano de la creación de los sistemas abiertos como lo es TCP/IP y de la creación de un nuevo sistema operativo creado por la empresa de AT&T llamado Unix el cual implementaba este protocolo, dando un mayor acceso a internet a todo aquel que contara con una conexión a la misma, ampliando la participación de más y más usuarios abriendo las puertas a diferentes sistemas operativos compartiendo información de manera fácil y sencilla sin necesidad de hacer uso exclusivo de ciertos dispositivos, sino sólo de hablar un mismo lenguaje para comunicarse a través de internet.

2.3.1. Niveles del modelo TCP/IP

TCP/IP se basa en software utilizado en redes. Aunque el nombre TCP/IP es la combinación de dos protocolos: Protocolo de Control de Transmisión y Protocolo Internet. El término TCP/IP no es una entidad única que combina dos protocolos, sino una familia de programas de software más grande que proporciona servicios de red, como registro de entrada remota, transferencia de archivo remoto y correo electrónico, etc., siendo TCP/IP un método para transferir información de una máquina a otra. Además TCP/IP maneja los errores en la transmisión, administra el enrutamiento y entrega de los datos, así como controlar la transmisión real mediante el uso de señales de estado predeterminado. Para solucionar todos estos problemas al igual que el modelo OSI que es el punto de partida para todos los sistemas de comunicaciones, el cual consta de siete capas para controlar todo lo anterior TCP/ IP consta de cuatro capas (Figura2.3.1):

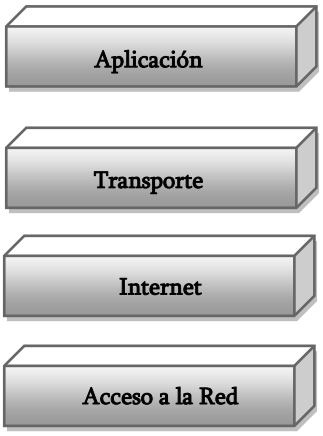


Figura 2.3.1. Modelo TCP/IP.

Cada una de estas capas al igual que en el modelo OSI se encarga de tareas específicas para mantener una comunicación.

1. Capa de Acceso a la Red

Esta capa es la más baja y se encarga de especificar información detallada, de cómo se envían físicamente los datos a través de la red, donde incluye cómo se realiza la señalización eléctrica de los bits mediante los dispositivos de hardware que conectan directamente con un medio de red, como un cable coaxial, un cable de fibra óptica o un cable de cobre de par trenzado. Por lo tanto de manera general, se encarga de ofrecer la capacidad de acceder a cualquier red física, es decir, brinda los recursos que se deben implementar para transmitir datos a través de la red. Algunas de las tareas de esta capa son las siguientes:

- ✓ Enrutamiento de datos por la conexión.
- ✓ Coordinación de la transmisión de datos (sincronización).
- ✓ Dar Formato de datos.
- ✓ Conversión de señal (análoga/digital).

Cabe destacar que esta capa también se encarga de proporcionarnos el control de errores para los datos entregados en la red física.

2. Capa de Internet

Esta es la capa más impórtate, ya que la tarea principal de esta capa es permitir que los hosts inyecten paquetes de datos (o datagramas definidos por esta capa) dentro de cualquier red, viajando a su destino de manera independiente, ya que la conmutación de paquetes que se lleva en esta capa es orientada a la conexión, estos paquetes pueden llegar en diferente orden del que fueron enviados, donde las capas superiores serán quienes se encarguen de ordenar estos paquetes, para su posterior reconstrucción. Entre sus tareas también tenemos el uso de la capa de acceso a la red, ésta relaciona las direcciones físicas con las lógicas, así como también se encarga de proporcionar enrutamiento de tráfico, para reducir la entrega y el apoyo a través de la red interna, evitando que ocurran congestiones.

Los protocolos más importantes que tenemos en esta capa son los siguientes:

- **IP** (Protocolo de Internet): Es un protocolo no orientado a la conexión, es uno de los más conocidos en esta capa, con mensajes o datagramas definiéndolos de un tamaño máximo. Cada datagrama se gestiona de forma independiente, por lo que dos datagramas pueden utilizar diferentes caminos para llegar al mismo destino, provocando que lleguen en diferente orden o bien duplicados, de tal manera que si enviamos una información éste se encargara que llegue a su destino de cualquier manera, pero éstos no nos garantiza que así sea. Es un protocolo no fiable ya que es no orientado a la conexión. Este protocolo recibe información del nivel superior y le añade la información necesaria para su gestión, como cabeceras como direcciones IP origen y destino, detección de errores o checksum, tiempo de vida o TTL, así como define el datagrama y su longitud, la versión IP, el protocolo usado, si un paquete fue fragmentado y la posición que le corresponde al fragmento.

- **ICMP** (Protocolo de Control de Mensajes de Internet): Proporciona un mecanismo de comunicación de información de control y de errores entre máquinas intermedias por las que viajaran los paquetes de datos. Estos datagramas los suelen emplear las máquinas (gateways, host, y más) para informarse de condiciones especiales en la red, como la existencia de una congestión, la existencia de errores y las posibles peticiones de cambios de ruta. Los mensajes de ICMP están encapsulados en datagramas IP, es por eso que ICMP es un compañero necesario de IP, por lo tanto no garantiza la entrega al destinatario y no puede estar seguro que va a ser informado de los problemas que puedan

aparecer en el tránsito de sus paquetes, mejorando el rendimiento. Entre los mensajes que nos proporciona ICMP podemos encontrar:

- Tamaño excesivo de paquetes.
- Si ha terminado su período de vida en los saltos autorizados por TTL.
- Si no tiene ni idea de cómo llegar al destino solicitado.
- Si existe una ruta mejor para llegar al destino.
- Si algún nodo está saturado para que se transmita con más calma.

- **ARP** (Protocolo de Resolución de Direcciones): Cuando una máquina desea ponerse en contacto con otra conoce su dirección IP, entonces necesita un mecanismo dinámico que le permita conocer su dirección física, entonces envía una petición ARP por broadcast (es decir a todas las máquinas). El protocolo establece que sólo contestara a la petición, si ésta lleva su dirección IP, por lo tanto sólo contestara la máquina que corresponde a la dirección IP buscada, con un mensaje que incluya la dirección física. El software de comunicaciones debe mantener una caché con los pares IP-dirección física, de éste modo la siguiente vez que se necesite hacer una transmisión a esa dirección IP, ya conoceremos la dirección física.

- **RARP** (Protocolo de Resolución de Direcciones Inverso): A veces el problema es al revés, o sea, una máquina sólo conoce su dirección física, y desea conocer su dirección lógica. Éste ocurre, por ejemplo, cuando se accede a Internet con una dirección diferente, en el caso de PC que acceden por módem a Internet y se le asigna una dirección diferente de las que tiene el proveedor sin utilizar. Para solucionar éste, se envía por broadcast una petición RARP con su dirección física, para que un servidor pueda darle su correspondencia IP.

3. Capa de Transporte

Este nivel proporciona una comunicación extremo a extremo entre programas de aplicación. La máquina remota recibe exactamente lo mismo que le envió la máquina origen. En este nivel el emisor divide la información que recibe del nivel de aplicación en paquetes, le añade los datos necesarios para el control de flujo y control de errores, y se los pasa al nivel de red junto con la dirección de destino.

En el receptor este nivel se encarga de ordenar y unir las tramas para generar de nuevo la información original.

Para implementar el nivel de transporte se utilizan dos protocolos:

- **UDP (Protocolo de Datagramas de Usuario)**: Proporciona un nivel de transporte no fiable de datagramas, pues es no orientado a la conexión, ya que apenas añade información al paquete que se envía al nivel inferior, sólo envía la necesaria para la comunicación extremo a extremo. UDP no cuenta con mecanismos para confirmar que los paquetes has sido enviados a su destinatario, pues cabe la posibilidad que lleguen o no a su destino, así como puede haber la duplicación de los datos transmitidos .UDP es más útil cuando requerimos de velocidad en la transmisión, como son los servicios de transmisión de voz y de video, pues UDP puede ser mucho más rápido a diferencia de TCP.

- **TCP (Protocolo de Control de Transmisión)**: Es el protocolo que proporciona un transporte fiable de flujo de bits en los dos sentidos de la conexión. Está pensado para poder enviar grandes cantidades de información de forma fiable, liberando al programador de aplicaciones, de la dificultad de gestionar la fiabilidad de la conexión (retransmisiones, pérdidas de paquete, orden en que llegan los paquetes, duplicados de paquetes y más) que administra el propio protocolo. Pero la complejidad de la gestión de la fiabilidad tiene un costo en eficiencia, ya que para llevar a cabo las gestiones anteriores se tiene que añadir bastante información a los paquetes a enviar. Debido a que los paquetes a enviar tienen un tamaño máximo, cuanta más información añada el protocolo para su gestión, menos información que proviene de la aplicación podrá contener ese paquete. Por eso, cuando es más importante la velocidad que la fiabilidad, se utiliza UDP, en cambio TCP, asegura la recepción en destino de la información a transmitir ya que es un protocolo orientado a la conexión.

4. Capa de Aplicación

Constituye el nivel más alto de la torre TCP/IP. A diferencia del modelo OSI, se trata de un nivel simple, en el que se encuentran las aplicaciones que acceden a servicios disponibles a través de Internet. Por lo tanto, el software en esta capa se comunica mediante uno o dos protocolos de la capa inferior (la capa de transporte), es decir, TCP o UDP. Estos servicios están sustentados por una serie de protocolos que proporcionan dichos servicios, como TELNET, FTP, SNMP, POP3, HTTP, DNS, SNMP, NFS y más, donde el protocolo HTTP o protocolo de transferencia de hipertexto es el más utilizado, pues éste es la base de la World Wide Web.

2.4. Modelo de referencia OSI vs. TCP/IP

Como hemos visto, tanto el modelo OSI como el modelo TCP/IP es muy importante en las comunicaciones, como nos pudimos dar cuenta ambos tienen mucho en común, como ser protocolos por capas así como tener un gran parecido entre el funcionamiento de las mismas.

Cabe notar que además de ser muy parecidos los modelos también cuentan con diferencias notables, como que en la arquitectura OSI y de TCP / IP se refieren a las capas por encima de la capa de transporte (capa 4) y los de la capa de red (capa 3). OSI tiene dos, la capa de sesión y la capa de presentación, mientras que TCP / IP combina en una sola capa la de aplicación la cual ofrece servicios muy diferentes en ambas. Por otra parte debido a que TCP / IP requiere del uso de protocolos de conexión, hacen que combine la capa física de la OSI y la capa de enlace de datos en un nivel de acceso a la red (Figura 2.4.1).

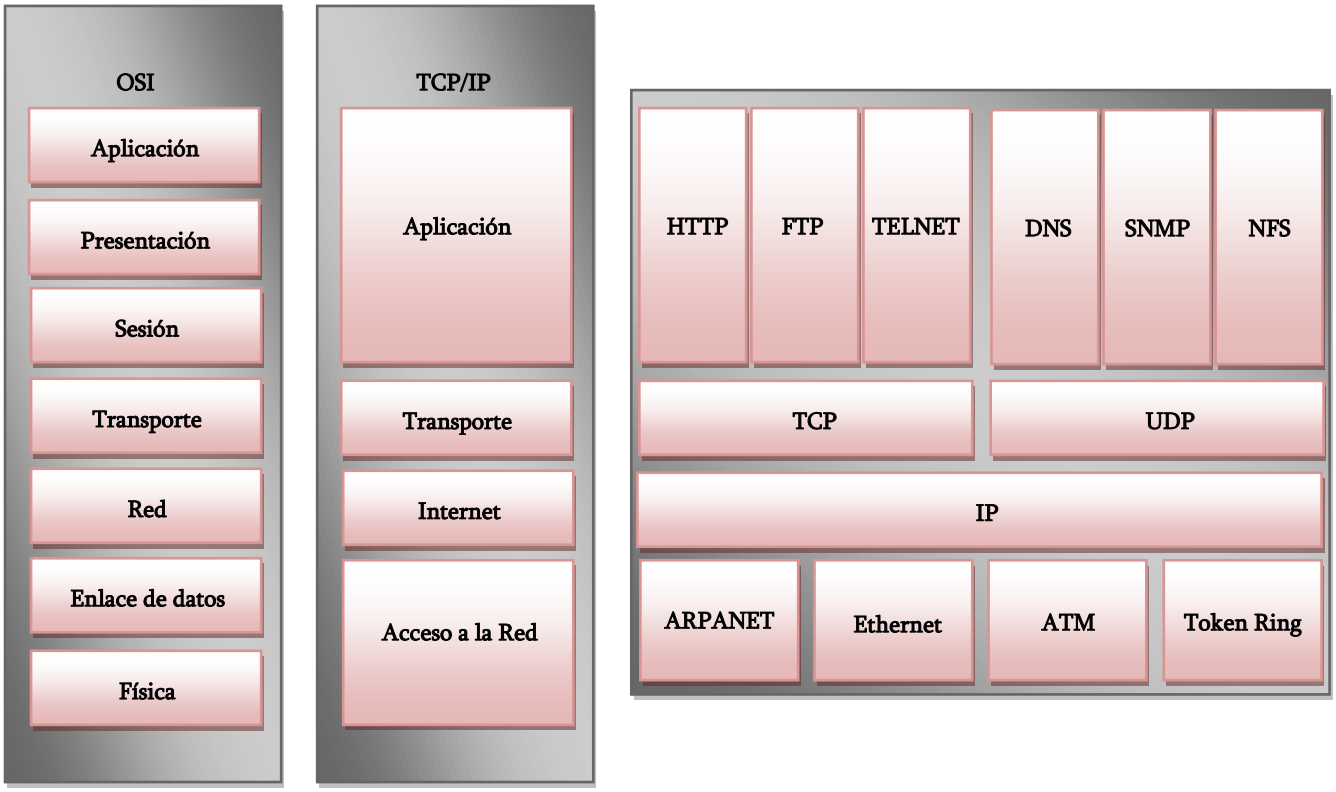


Figura 2.4.1. Comparación entre los Modelos OSI vs. TCP/IP.

Mostraremos una tabla comparativa de éstos dos modelos (Tabla 2.4.1):

Tabla 2.4.1. OSI vs. TCP/IP.

OSI vs. TCP/IP	
OSI	TCP/IP
Hace una clara separación y énfasis en tres aspectos importantes para que funcione el modelo: ✓ Servicio: Lo que un nivel o la capa hace. ✓ Interfaz: Cómo se pueden accesar a los servicios. ✓ Protocolo: La implementación de los servicios.	TCP/IP no tiene ésta clara separación.
OSI fue definido antes de implementar los protocolos, los diseñadores no tenían mucha experiencia sobre dónde se debieran ubicar las funcionalidades.	El modelo de TCP/IP fue definido después de los protocolos y se adecúan perfectamente. Pero no otras pilas de protocolos.
OSI es un buen modelo (no los protocolos), es por eso que los demás modelos lo toman como un modelo de referencia.	TCP/IP es un buen conjunto de protocolos, pero el modelo no es general, además de que éste fue el predecesor de OSI.
Algunas de las actividades de las capas son demasiado redundantes, como el control de errores en todas las capas, haciendo crecer el código de la implantación del mismo, volviéndose ineficiente y de mala calidad.	Se logró la implantación en sistemas UNIX sin la necesidad de tomar muchos recursos del sistema de una manera muy eficiente.
OSI no tuvo éxito debido a : ✓ Mal momento de introducción: insuficiente tiempo entre las investigaciones y el desarrollo del mercado a gran escala para lograr la estandarización. ✓ Mala tecnología: OSI resultó ser complejo para entender e implementar, es dominado por una	TCP/IP se ha ido acrecentando gracias a que: ✓ Los protocolos TCP/IP, para el momento en que se quería introducir los protocolos de OSI, los de TCP/IP ya eran ampliamente utilizados por las universidades. ✓ Ya que el mercado de las universidades era bastante amplio

mentalidad de telecomunicaciones sin pensar en las computadoras, carece de servicios sin conexión. ✓ Malas implementaciones debido a su complejidad. ✓ Malas políticas: investigadores y programadores contra los ministerios de telecomunicación	crecieron los proveedores de productos TCP/IP, así como su amplia gama de protocolos. ✓ Se implementó en UNIX de manera satisfactoria, incrementándose los usuarios que hacían uso de éste pues UNIX era gratis, además que con el paso del tiempo fue mejorando gracias a la propia comunidad de UNIX.

2.5. HTTP (Protocolo de Transferencia de Hipertexto)

El protocolo de transferencia de Hipertexto o mejor conocido como HTTP es uno de los más importantes y utilizados dentro del WWW, pues éste es capaz de proporcionar información que se encuentre almacenada en lugares extremos a cualquier usuario que la solicite, independientemente que se encuentren en lugares geográficos diferentes.

HTTP es un protocolo del nivel de aplicación, el cual trabaja en conjunto de hipermedia, no sólo nos ayuda acceder a información simple de texto plano, sino que también se puede acceder a imágenes como animaciones, sonidos, videos y más.

En la práctica los sistemas de información requieren de más funcionalidad de recuperación simple, incluyendo búsquedas y para esto requiere de una notación. HTTP hace uso de un conjunto de métodos y cabeceras, que indican el propósito de una solicitud, para esto se construyeron las referencia proporcionadas por el Uniform Resource Identifier (URI , Identificador de Recursos Uniforme), como una dirección (URL) o el nombre (URN, Nombre de Recursos Uniforme), para indicar el recurso al que el método es aplicable. Los mensajes son pasados a un formato similar, al utilizado por el correo de Internet, definido por el Multipurpose Internet Mail Extensions (MIME, Extensiones Multipropósito de Correo Internet).

HTTP se utiliza también como un protocolo genérico para la comunicación entre los agentes de usuario y los servidores proxy o puertas de acceso a los sistemas de Internet, incluyendo soporte para protocolos como el SMTP, NTP, FTP, Gopher, y WAIS.

Las principales características de este protocolo son:

- Responde al modelo cliente/servidor, el cual se encuentra orientado a transacciones de tipo solicitud/respuesta entre un cliente y un servidor. A la información transmitida se le llama recurso y se identifica mediante una URL.
- Es un protocolo sin estado, donde cada transacción que se realiza se lleva a cabo aisladamente, no guarda ninguna información sobre conexiones anteriores. En la mayor parte de los casos, el desarrollo de aplicaciones Web requiere mantener el estado. Debido a ésta necesidad, en el *protocolo HTTP* se utilizan las cookies (galletitas).
- El modelo de capas dentro del modelo TCP/IP, HTTP se encuentra en el nivel de aplicación, operando sobre las conexiones de transporte de tipo TCP, generalmente en el puerto 80.
- Una implementación típica, creará una conexión nueva entre cliente y servidor para cada transacción y después de esto la cerrará. También es posible establecer conexiones de transporte persistente, que se mantienen abiertas durante varias transacciones HTTP, de tal manera que se reduce el número de paquetes en la red que se hubieran generado por la apertura y cierre de conexiones TCP; permitiéndonos enviar múltiples solicitudes al servidor, sin la necesidad de esperar a recibir previamente la respuesta correspondiente a la anterior solicitud, devolviendo el servidor las respuestas en el mismo orden en que fueron llegando las solicitudes.

2.5.1. Funcionamiento

El protocolo HTTP es un protocolo de solicitud/respuesta. Un cliente envía una solicitud al servidor en forma de una solicitud al método, un URI y la versión del protocolo, seguido de un mensaje MIME-like contiene información relevante como información del cliente, y un posible contenido del cuerpo del mensaje mediante una conexión. El Servidor responde con una línea de estado, incluyendo la versión del protocolo, mensajes de código de error o de éxito, seguido de un mensaje MIME-like que contiene información del servidor, metainformación de la entidad, y un posible contenido del cuerpo de la entidad. Tenemos que se hacen peticiones entre un cliente y un Servidor Web quienes no hablan el protocolo HTTP, por lo que a menudo son llamados servidores HTTP. Estos servidores HTTP almacenan de datos de Internet y suministran los datos cuando son solicitados por los clientes HTTP. Los clientes envían las peticiones HTTP a los servidores y éstos a su vez se encargan de devolver los datos solicitados en las respuestas HTTP. De tal manera que en éste caso los clientes hacen uso de un

software el cual es un navegador Web como Internet explorer, Nestcape, Mozila, y más. El navegador Web es quien se encarga de mostrar los objetos en pantalla al cliente donde dichos objetos se encuentran alojados en un servidor HTTP.

La mayoría de las comunicaciones vía HTTP son iniciadas mediante un usuario agente, donde ese realiza una solicitud para hacer uso de algún recurso que se encuentra dentro de algún servidor origen, mediante el uso de una única conexión(v) entre el Usuario Agente (UA) y el servidor origen (O) (Figura 2.5.1).

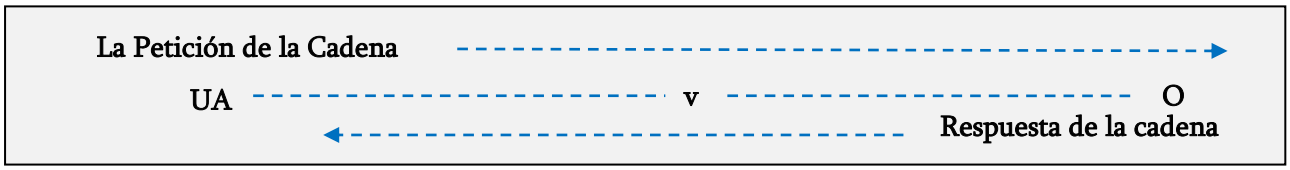


Figura 2.5.1. Caso simple de comunicación HTTP.

2.5.1.1. Componentes de la arquitectura Web

Dentro del funcionamiento de la interacción entre una petición de un cliente, mediante un navegador a un servidor Web, para obtener alguna información la forma sencilla de describir ésta comunicación es la que se muestra en la Figura 2.5.1; pero podemos encontrar más intermediarios que son capaces de interactuar con Internet entre las que encontramos:

a) Proxies.

Éste es un intermediario de HTTP que se encuentra entre los clientes y los servidores. Los servidores proxy son de gran importancia para la seguridad Web, así como para la integración, optimización y rendimiento de las aplicaciones Web. Un proxy se encarga de recibir todas las peticiones HTTP, retransmitiendo todas estas peticiones al servidor.

Generalmente los servidores proxy son usados por cuestiones de seguridad como intermediarios de confianza, encargándose de recibir todo el flujo de tráfico Web por él, antes de llegar al servidor origen, de tal forma que éstos son capaces de filtrar las solicitudes y las respuestas en entre el cliente y el servidor, un ejemplo de ello es cuando se detectan virus en las descargas de aplicaciones o filtran el contenido de sitios no aptos para niños en una escuela (Figura 2.5.2).

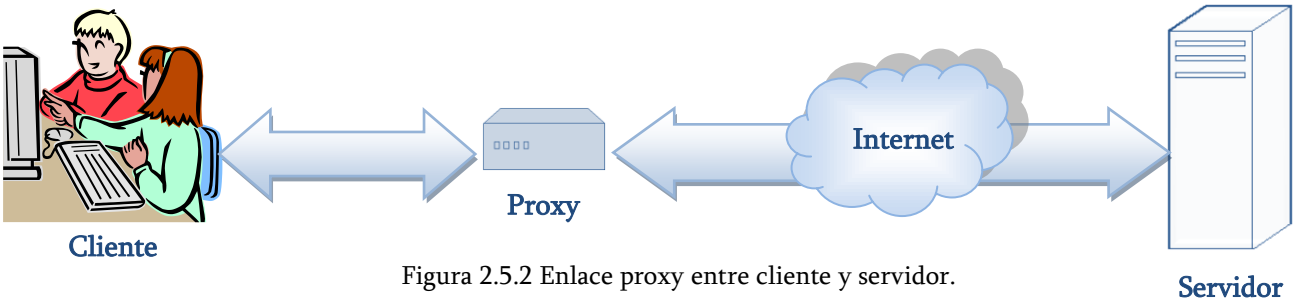


Figura 2.5.2 Enlace proxy entre cliente y servidor.

b) Cachés

Éste es un almacén de HTTP y es un tipo especial de servidor proxy HTTP, el cual se encarga de guardar copias de los sitios Web más populares que pasan a través del proxy. Esto lo hace con el fin de reducir el tiempo de respuesta y el consumo de ancho de banda.

Un cliente puede descargar un documento mucho más rápido de una caché, puesto que se encuentra más cerca, que de un servidor Web que está mucho más lejano (Figura 2.5.3).

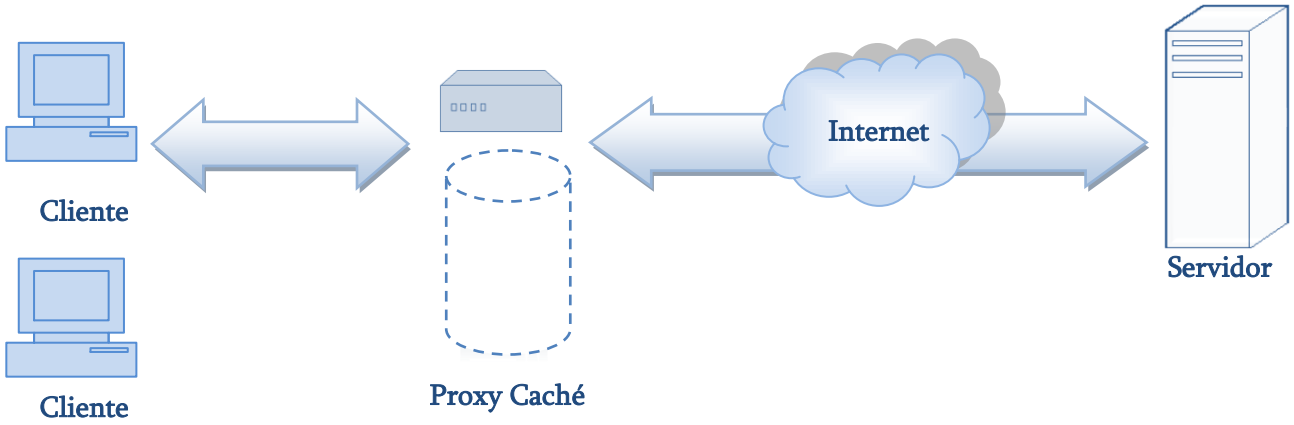


Figura 2.5.3. Proxy Caché que aumenta el rendimiento de la red
Guardando copias locales de los documentos que más utiliza el usuario.

c) Puerta de enlace (Gateway)

También conocido como puerta de enlace, son servidores Web que se conectan con otras aplicaciones actuando como intermediarios de otros servidores. Su función es permitir interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación. Su propósito es traducir la información del protocolo utilizado en una red, al

protocolo usado en la red de destino. También son utilizados para convertir tráfico HTTP en otros protocolos. Una puerta de enlace se encarga de recibir peticiones http como si fuera el servidor origen al que se le solicita algún recurso.

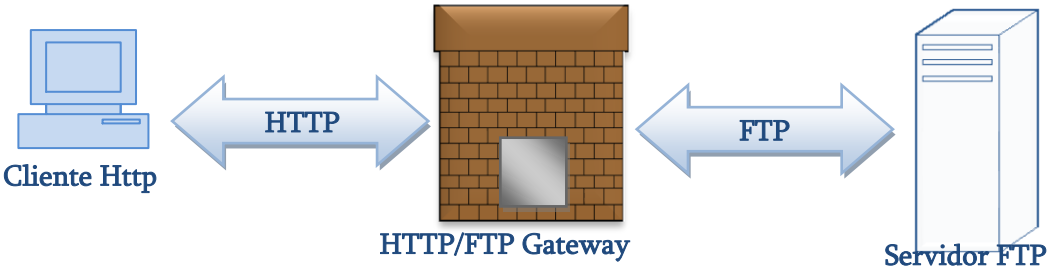


Figura 2.5.4. Gateway HTTP/FTP.

Un ejemplo del funcionamiento del Gateway es cuando éste recibe solicitudes de FTP a través de URIs mediante el uso de peticiones HTTP, Esto para solicitar alguna información como algún documento, recuperando dicho documentos utilizando el protocolo FTP, como se muestra en la Figura 2.5.4. El documento resultante se encapsula en un mensaje HTTP y envía al cliente que lo solicita.

d) Túneles (Tunnels)

Los túneles son un tipo de proxy especial, donde las comunicaciones HTTP las transmite a ciegas. Los túneles son solicitudes HTTP, que después de la instalación, se encargan de retransmitir los datos ciegamente entre dos conexiones. Los túneles HTTP se utilizan con frecuencia para el transporte de datos que no son del tipo HTTP, a través de una o más conexiones HTTP, sin fijarse si los datos son o no del tipo HTTP.

El uso más común de los túneles de HTTP es cuando se lleva cifrado un tráfico, mediante SSL (Secure Soket Layer) en una conexión de HTTP, permitiendo pasar dicho tráfico sin ningún impedimento, pasando por los firewalls, que por sus reglas sólo permiten el tráfico de internet.

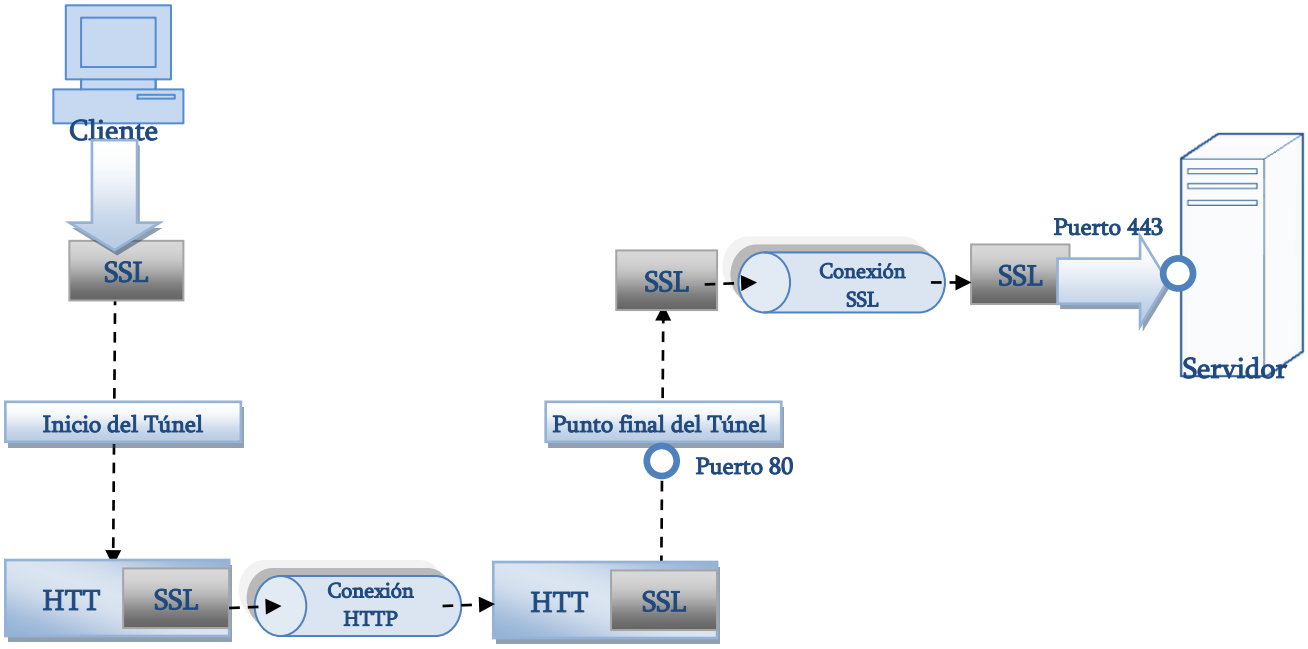


Figura 2.5.5. Envío de datos a través de un túnel HTTP/SSL.

Un ejemplo claro, lo podemos observar en la Figura 2.5.5. Donde un túnel de HTTP / SSL recibe una petición del tipo HTTP, para establecer una conexión de salida o socket hacia una dirección de destino y puerto, de ahí se dirige el tráfico encriptado SSH al túnel pasando por un canal HTTP, de forma que pueda ser transmitida a ciegas al servidor de destino.

e) Agentes (Agents)

Loa agentes son semi clientes Web, que se encargan de realizar peticiones Web de manera automática. Cualquier aplicación que emita una solicitud Web es un agente HTTP. Los agentes son programas de cliente que realizan solicitudes Web en nombre del usuario, un ejemplo sencillo son los navegadores Web, pero también podemos encontrar equipos o máquinas que navegan de forma autónoma en la Web, también son conocidos como *"robots Web" o "arañas"*, las cuales trabajan sin la intervención de un usuario, donde dichas máquinas van buscando páginas de todo el mundo, para crear archivos que les proporcionen información útil, como crear sus catálogos de productos que se encuentren a la venta y encontrar los mejores precios (Figura 2.5.6).

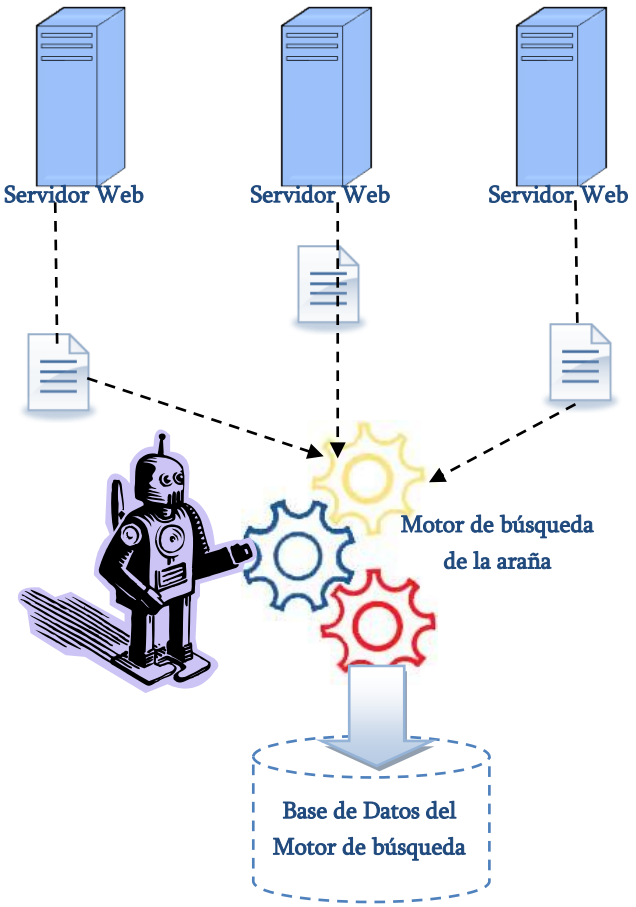


Figura 2.5.6. Forma automática de navegación en la Web.

2.5.2. Tipos de Intermediarios en el funcionamiento

Dentro del funcionamiento de HTTP podemos encontrar más de un intermediario, dentro de las peticiones/respuestas. Encontramos tres formas comunes de intermediario, proxy, puerta de enlace y túnel.

Si contamos con tres intermediarios (A, B, C) entre el usuario agente (UA) y el servidor de origen (O), podemos encontrar éste tipo de conexión HTTP, cuando se envía una solicitud o un

mensaje de respuesta, que viaja a través de una cadena que atraviesa cuatro conexiones diferentes, como se muestra en la Figura 2.5.7.

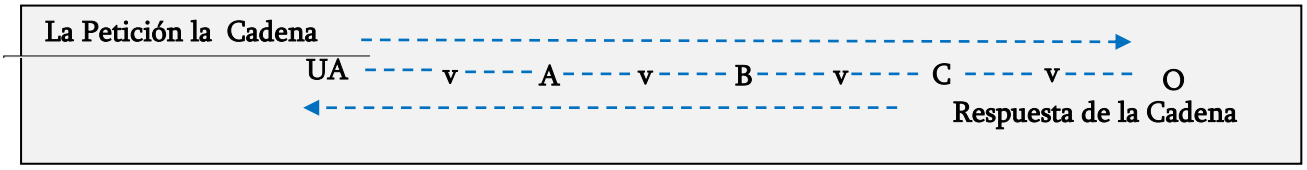


Figura 2.5.7. Cadena de comunicación entre 3 intermediarios.

Éste tipo de conexión HTTP sólo aplica, a vecinos que no sea un túnel, sólo para los puntos extremos de la cadena o para todas las conexiones de la cadena. Cualquier parte de la comunicación que no esté actuando como un túnel, donde cada uno de los participantes podrá recibir múltiples comunicaciones simultáneamente.

Cualquier parte de la comunicación que no esté actuando como un túnel, puede usar su caché interna para transmitir las solicitudes, haciendo que la solicitud/respuesta de la cadena sea más corta si uno de los participantes, a lo largo de la cadena, tiene esa respuesta en su caché interna aplicable a esa solicitud. Un ejemplo se muestra en la Figura 2.5.8. Donde B tiene una copia en caché de una respuesta anterior de O vía C de una solicitud que no se ha almacenado en cache por UA o A.

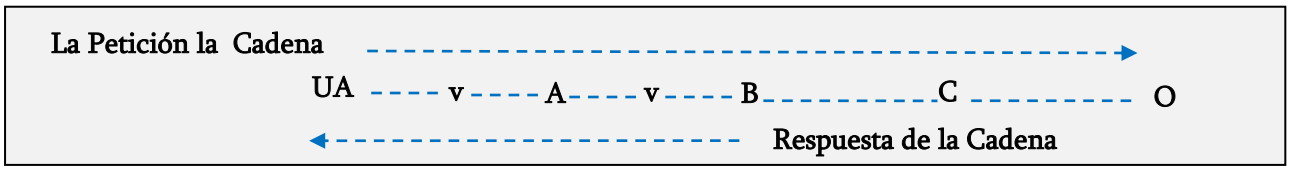


Figura 2.5.8. Uso de la caché de B para responder una solicitud de UA ó A.

2.6. Parámetros del Protocolo HTTP

2.6.1. Versión del protocolo

El protocolo HTTP hace uso de un esquema de numeración "<major>.<minor>". Ésta política del control de versiones del protocolo, le permite al remitente darle formato a sus mensajes, así como también le ayuda a comprender la comunicación HTTP. No ocurre ningún cambio en el número de la versión, si se agregan componentes a los mensajes que no afecten la conducta en la comunicación, sólo se incrementa el número <minor> si los cambios que se le agregan al protocolo, no cambian el algoritmo que se encarga de analizar el mensaje sintácticamente, pero

que si puede aportar a la semántica del mensaje, así como también agrega nuevas capacidades al remitente. El número <major> sólo se incrementa cuando se cambia el formato de un mensaje dentro del protocolo.

Los proxy y los Gateway de aplicación deben tener cuidado en enviar mensajes con versiones del protocolo diferentes a las de su aplicación. Puesto que la versión del protocolo indica la capacidad del remitente, un proxy o Gateway, no debe enviar mensajes que indiquen que su versión del protocolo es mayor a su versión actual. En el caso de que el proxy/Gateway reciba alguna petición con una versión del protocolo mayor a la suya, deben bajar la versión de la solicitud, o bien responder con un mensaje de error, o cambiar el comportamiento de los túneles; es por éstos problemas de compatibilidad que los túneles no deben actualizar las solicitudes de las versiones más altas que soportan.

2.6.2. Codificación del Contenido

La codificación de los contenidos se usa principalmente para permitir la compresión de documentos o bien para transformarlos de tal manera que no se pierda información del documento. Este documento se almacena de una manera codificada, se transmite directamente, y sólo es decodificada por el destinatario.

Con lo anterior podemos afirmar que:

La codificación del contenido = token

Donde un token es un conjunto de bits que viaja a través de la red, siendo único en la red donde viaja.

La IANA o la Autoridad de Asignación de Números de Internet, es la encargada de registrar los valores de los tokens, donde dicho registro contiene los siguientes tokens:

- Un formato de codificación gzip, es generado por el programa de codificación "gzip" (GNU zip), siendo éste un algoritmo llamado Lempep-Ziv, que es acompañado por un CRC (Código de Redundancia Cíclica) de 32 bits.

2.6.3. Codificación de la Transferencia

La codificación de la transferencia de mensajes se lleva a cabo en el cuerpo de dicha entidad, para garantizar un "transporte seguro", a través de la red, esto es diferente a la codificación del contenido, ya que la codificación de la transferencia es una propiedad exclusiva del mensaje, más no de la entidad original.

De lo cual podemos decir que:

La codificación de transferencia = "chunked" o "fragmento"

Siempre que exista un codificación de transferencia se aplica al cuerpo del mensaje, donde en cada codificación se incluye un "chunked", con la excepción que dicho mensaje sea el cierre de conexión, sólo se aplica un "chunked" al cuerpo del mensaje, ya que ésta regla le permite al receptor determinar la longitud de la transferencia del mensaje.

2.6.4. Mensajes HTTP

El protocolo HTTP cuenta con dos tipos de mensajes de *"solicitud y de respuesta",* donde los mensajes de solicitud van del cliente hasta el servidor y los de respuesta del servidor al cliente.

Los mensajes de solicitud y de respuesta hacen uso de un formato especial para la transferencia de los mismos, dicho formato consta de una línea con cero o más cabeceras (donde cada una de ellas contiene sus propios campos), seguido de una línea vacía (CRLF) para posteriormente comenzar con el cuerpo del mensaje.

2.6.4.1. Cabeceras de los mensajes

Los encabezados de los mensajes HTTP pueden ser de 4 diferentes tipos:

- Cabeceras Generales.
- Cabeceras de Solicitud.
- Cabeceras de Respuesta.
- Cabeceras de Entidades.

Como ya mencionamos anteriormente cada cabecera consta de sus propios campos, dichos campos también cuentan con su formato, el cual es el siguiente:

<Nombre del campo> ":" <valor de campo>

El orden en que los campos de las cabeceras con diferente nombre, son recibidos no es de importancia, sin embargo, por "buenas prácticas", es recomendable que se envíen primero los campos de las cabeceras generales, después los campos de las cabeceras de solicitud o de respuesta y por último los campos de las cabeceras de entidad.

2.6.4.2. Cabeceras generales

Los campos de las cabeceras generales, se llaman así, porque tienen una aplicación general para los mensajes de solicitud y de respuesta, pero que no aplica a la entidad que es transferida. Estos campos sólo se aplican al mensaje que se transmite (Tabla 2.6.1).

Tabla 2.6.1. Cabeceras Generales del Mensaje.

Nombre de la Cabecera	Descripción de la Cabecera
Cache-Control	Se utiliza para especificar directivas, las cuales deben ser obedecidas por todos los mecanismos de almacenamiento en caché, a lo largo de las peticiones de solicitud/respuesta para almacenamiento en caché, por medio de algoritmos predeterminados por las mismas directivas.
Connection	Esta cabecera le permite al remitente realizar especificaciones para conexiones especiales, sin necesidad de comunicarles a los proxies con conexiones adicionales.
Date	Se encarga de especificar la fecha y la hora en que se originó el mensaje. Un ejemplo es : Date: Tue, 15 Nov 1994 08:12:31 GMT
Pragma	Es usado para incluir directivas específicas de implementación que podrían aplicarse a lo largo de la cadena de solicitud/respuesta. Estas generalmente, especifican el punto de vista del protocolo.
Trailer	Éste indica el conjunto de campos de cabeceras, que está presente en el trailer de un mensaje codificado, con una

	codificación de transferencia fragmentada. De tal manera que permite que el destinatario sepa los campos que debe esperar. Éste no debe incluir encabezados de campos como Transfer_Encoding, Content-Length y Trailer.
Transfer-Encoding	Indican los tipos de transformaciones que ha sufrido el cuerpo del mensaje, con el fin de llevar a cabo una transferencia segura entre el remitente y el destinatario.
Upgrade	Se encarga de que el cliente especifique qué protocolos de comunicación está utilizando, para apoyarse en la comunicación, así como que protocolo desea utilizar si el servidor considera adecuado cambiar de protocolo.
Via	Generalmente es utilizado por los Gateway y los proxies, para indicar los protocolos intermedios y los beneficiarios entre el usuario agente y el servidor de peticiones, y entre el servidor de origen y las respuestas del cliente.
Warning	Es un campo de alerta que se utiliza para transportar información, acerca de la situación o de la transformación del mensaje, advirtiendo de posibles fallas de las operaciones o transformaciones que se le han aplicado el cuerpo del mensaje.

2.6.4.3. Mensajes de Solicitud

Antes de hablar de las cabeceras de solicitud, hablaremos primero de los mensajes de solicitud. Un mensaje de solicitud del cliente a un servidor, debe incluir en la primera línea, el método que se le aplicara al recurso, el identificador de dicho recurso, así como la versión del protocolo que se está utilizando.

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

La línea comienza con un método de razón, seguida de la URI de la petición, la versión del protocolo y terminando con CRLF. Los elementos están separados por caracteres de SP, N, CR o LF los cuales están permitidos, excepto en la secuencia CRLF al final.

Este mensaje de solicitud, comienza con la línea Request-Line, seguido por sus cabeceras generales, de petición y entidad, y finalmente comenzando el cuerpo del mensaje.

2.6.4.3.1. Métodos

Como ya mencionamos cuando se realiza un mensaje de solicitud se requiere de un método, al cual se le aplicara al recurso solicitado, identificado por el URI de la petición, entre los cuales encontramos los siguientes (Tabla 2.6.2):

Tabla 2.6.2. Métodos de solicitud de mensaje.

Método	Descripción del Método
OPTIONS	Solicita información sobre las opciones de comunicación en la cadena de petición/respuesta identificados por el URI de la petición.
GET	Solicita el recurso ubicado en la URI especificada.
HEAD	Solicita el encabezado del recurso ubicado en la URL especificada.
POST	Envía datos al programa ubicado en la URI especificada.
PUT	Envía datos a la URI especificada.
DELETE	Borra el recurso ubicado en la URI especificada.
TRACE	Este método le permite al cliente ver que es lo que está recibiendo el otro extremo de la cadena de solicitud, de tal manera que usa dichos datos para pruebas de información de diagnóstico.
CONNECT	Se reserva para el uso de un proxy que pueda cambiar de forma dinámica a ser un túnel.

2.6.4.3.2. Cabeceras de Solicitud

Las cabeceras de solicitud brindan información adicional acerca de la solicitud así como del propio cliente (Tabla 2.6.3).

Tabla 2.6.3. Cabeceras de solicitud.

Nombre de la Cabecera	Descripción de la Cabecera
Accept	Especifica el tipo de datos que se aceptan como respuesta.
Accept- Charset	Indica el juego de caracteres aceptables para la respuesta.
Accept-Encoding	Restringe la codificación del contenido que es aceptable para la respuesta.

Accept-Languages	Restringe el conjunto de lenguas que son preferidas para la respuesta de una solicitud.
Authorization	Esta cabecera nos sirve cuando el usuario agente, desea autenticarse con el servidor, dicha cabecera contiene información (credenciales) de autenticación del usuario agente, para que éste mediante la autorización pueda acceder a los recursos solicitados.
Expect	El campo de expectativa, nos indica el comportamiento de los servidores que son solicitados por el cliente, como por ejemplo si se encuentra en espera una solicitud o bien mandar un mensaje de error de la expectativa del servidor.
From	Esta cabecera contiene una dirección de correo electrónico, del cliente o usuario agente, el cual es usado con propósitos de registro para identificar las fuentes de solicitudes no válidas o no deseadas. Este campo interpreta que el nombre de la persona que realiza la solicitud, acepta la responsabilidad por el método realizado.
Host	Se encarga de identificar el Host y número de puerto del recurso que es solicitado mediante la URI.
If-Match	Es un método condicional. Su principal función es permitir las actualizaciones de la información almacenada en la caché, con un gasto mínimo en las transacciones. También se utiliza en la actualización de las solicitudes.
If- Modified-Since	Se utiliza cuando un método se hace condicional, si la variante que es solicitada no se modifica desde la fecha indicada en éste campo, regresa con código de respuesta 304 sin cuerpo del mensaje. Ejem: If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT
If- None-Match	Se utiliza cuando un método se hace condicional pero de manera negada.
If-Range	Permite realizar una copia parcial de la entidad en caché, siempre y cuando cumpla una condición de algún método.
If-Unmodified-Since	Es utilizado para que el servidor realice una operación o

	una serie de operaciones, si el recurso solicitado no se ha modificado desde la fecha indicada en este campo.
Max-Forwards	Hace mano del método TRACE para limitar el número de servidores proxy o gateways que puedan enviar solicitudes de entrada al siguiente servidor.
Proxy-Authorization	Le permite al cliente identificarse así mismo, así como al servidor proxy al cual requiere autenticarse, mediante el uso de credenciales de autenticación.
Range	Hace referencia al rango de bytes entre las operaciones que realizan el cliente y el servidor.
Referer	Permite que el cliente especifique el elemento de donde se obtuvo el enlace mediante la URI del recurso que se obtuvo de la petición.
TE	Indica que es lo que está dispuesto a aceptar, y que no en la codificación de la trasferencia fragmentada. Éste sólo aplica a la conexión inmediata.
User-Agent	Se encarga de identificar el software del cliente.

2.6.4.4. Mensajes de Respuesta

Como ya vimos, cuando se envía un mensaje de solicitud, siempre debe haber una respuesta, la cual es dada por el servidor a quién se le hizo dicha solicitud, mediante un mensaje de respuesta HTTP.

Dicho mensaje está formado por una línea de estado, la cual nos ayuda a identificar que es un mensaje de respuesta, donde se indica la versión del protocolo, seguido de un código de estado numérico que está asociado a una frase textual separados por caracteres .

Frase Textual

Dicha frase, pretende dar una descripción textual del Código de estado donde dicha frase está destinada para que el usuario sepa que es lo que está sucediendo con su petición.

2.6.4.4.1. Código de estado

El código de estado, se encuentra formado por 3 dígitos enteros los cuales intentan comprender y satisfacer las peticiones, de tal forma que éstos no están destinados para la comprensión de los usuarios, sino para el uso de los autómatas.

El primer dígito nos define de manera general la clase de respuesta que se brindará y los siguientes números no son de gran relevancia, simplemente son valores individuales del tipo de respuesta. Para el primer dígito, encontramos 5 valores los cuales son (Tabla 2.6.4):

Tabla 2.6.4. Código de estado del mensaje.

Código	Descripción
1XX	Informativo.- Petición recibida, proceso continuo. Éste no es utilizado y se reserva para usos futuros.
2XX	Éxito.- La acción fue recibida, entendido y aceptado con éxito.
3XX	Redirección.- Medidas adoptadas con la finalidad de que se lleve a cabo completa la solicitud.
4XX	Error del Cliente.-Nos indica que la solicitud contiene una sintaxis incorrecta o que no se puede cumplir.
5XX	Error del Servidor.- Indica que el servidor no puede cumplir solicitud aparentemente valida.

Los valores de los códigos de estado individuales son los siguientes (Tabla 2.6.5):

Tabla 2.6.5. Código de estado individual del mensaje.

Código	Descripción
100	Continuar
101	Cambiando de protocolo
200	OK
201	Creado
202	Aceptado
203	Información no autorizada
204	No Contenido
205	Restablecer contenido
206	Contenido Parcial

300	Opciones múltiples
301	Traslado permanente
302	Establecer
303	Ver otros
304	No se modificó
305	Utilice un proxy
307	Redirección temporal
400	Solicitud errónea
401	No autorizado
402	Pago requerido
403	Prohibido
404	No se encontró
405	Método no permitido
406	No aceptable
407	Autenticación requerida al proxy
408	Hora de la solicitud de salida
409	Conflicto
410	Quitado
411	Requiere de longitud
412	Precondición fracasada
413	Entidad de solicitud demasiado larga
414	Request-URI demasiado grande
415	Tipo de soporte incompatible
416	Rango solicitado no valido
417	Error de expectativa
500	Error interno del servidor
501	No implementado
502	Gateway erróneo
503	Servicio no disponible
504	Gateway time-out
505	Versión del protocolo HTTP no compatible

2.6.4.4.2. Cabeceras de Respuesta.

Las cabeceras de respuesta se encargan de brindarnos información acerca del servidor, proporcionándonos un mayor acceso a los recursos, identificados por el URI de la petición realizada. Las cabeceras de Respuesta son las siguientes (Tabla 2.6.6):

Tabla 2.6.6. Cabeceras de respuesta.

Nombre de la Cabecera	Descripción de la Cabecera
Accept-Ranges	Le permite al servidor indicar la aceptación de solicitudes para un rango de recursos en bytes.
Age	Calcula el tiempo estimado desde que la respuesta fue generada en el servidor origen.
ETag	Indica la etiqueta de la entidad, la cual sirve para la comparación contra otras entidades de un mismo recurso.
Location	Se encarga de redirigir al receptor a una ubicación diferente a la de la URI de la petición, para completar la solicitud o bien identificar un nuevo recurso creado por otra petición, indicando la ubicación URI del servidor deseado para el redireccionamiento automático a los recursos.
Proxy-Authenticate	Este campo se incluye cuando hay una respuesta con código 407 (Requiere una autenticación de proxy).El valor del campo contiene la información de las credenciales para que el usuario se pueda autenticar ante el proxy.
Retry-After	Éste se debe incluir cuando existe una respuesta con código 503 (Servicio no disponible), para indicar el tiempo que hay que esperar para que dicho servicio esté disponible.
Server	Identifica el software del servidor, para gestionar la petición.
Vary	Identifica que la entidad de respuesta sea seleccionada entre las diferentes respuestas disponibles, utilizando sus propios criterios mediante algoritmos de selección.
WWW-Authenticate	Éste se incluye cuando existe mensaje de respuesta con un

	código 401 (no autorizado), indica los parámetros del sistema de autenticación y los parámetros aplicables a la petición URI.

2.6.4.5. Entidad

Tanto los mensajes de solicitud, como los de respuesta, son capaces de transferir una entidad. Una entidad se encuentra compuesta de sus propios campos de cabeceras así como del cuerpo de la entidad, aunque suele ocurrir que algunas respuestas sólo incluyan las cabeceras de la entidad.

2.6.4.5.1. Cabeceras de Entidad

Las cabeceras de Entidad son las siguientes (Tabla 2.6.7):

Tabla 2.6.7. Cabeceras de entidad.

Nombre de la Cabecera	Descripción de la Cabecera
Allow	Se encarga de informar al destinatario de los métodos que son válidos, que están relacionados con el recurso solicitado.
Content-Encoding	Se encarga de indicar el tipo de codificación que se le ha aplicado al cuerpo de la entidad, así como los mecanismos de decodificación que se deben aplicar para obtener los datos.
Content-Language	De fine el lenguaje natural de la información, así como el preferido por el propio usuario.
Content-Location	Provee la ubicación del recurso solicitado para la entidad, cuando la entidad se encuentra en una ubicación separada de la URI.
Content-MD5	Se encarga de verificar que el mensaje no haya sufrido ninguna alteración, es decir que lo mismo que envió el emisor sea igual a lo que recibe el receptor.
Content-Range	Indica la duración total del cuerpo de la entidad.
Content-Type	Indica el tipo de medio usado entre el cuerpo de la

	entidad enviada al destinatario, o bien tipo de contenido del cuerpo de la solicitud como por ejemplo un testo plano como el HTML.
Expires	Indica la fecha límite del uso de los datos.
Last-Modified	Indica la última vez en que se realizó una modificación a algún elemento.

2.7. Ventajas y desventajas del protocolo HTTP

- **Ventajas**
  - Es multiplataforma y por tanto es utilizado globalmente por cualquier aplicación Web y a su vez es escalable en cualquier plataforma.
  - No sobrecarga la Red para crear, mantener el estado de sesión y la información.
  - Es mucho más rápido que HTTPS, ya que no pierde tiempo y recursos de procesamiento en hacer uso de algoritmos criptográficos y de compresión.

- **Desventajas**
  - Pasa por encima de los firewalls, carece de seguridad.
  - Integridad.- No usa métodos de cifrado y está sujeto a muchos ataques como lo son el ***"Man in the middle"*** el cual es un tercero que esté a la escucha de nuestra información.
  - Privacidad.- Tiene muchas limitaciones de seguridad ya que cualquier persona puede ver el contenido de la información en caso de ser interceptada la conexión, ya que los datos viajan en texto plano sin ser cifrado.
  - Autenticación.- No cuenta con un mecanismo de autenticación y por lo tanto no sabe con quién o quienes se está comunicando.

2.8. HTTPS (Protocolo de Transferencia de Hipertexto Seguro)

El Protocolo de Transferencia de Hipertexto Seguro es un protocolo de red, basado en HTTP, el cual se encarga de realizar transferencias de datos de hipertexto de una manera segura. HTTPS nace de la necesidad de la creación de páginas dinámicas, en donde se empiezan a crear sitios Web donde las diferentes instituciones como compañías, comercios y bancos empezaron a utilizarlas para transacciones financieras, en donde bien se compran productos, por medio de las tarjetas de crédito, o bien como hoy en día se pagan diferentes servicios públicos, en donde se realizan operaciones bancarias en línea; por lo tanto estamos haciendo uso de datos muy

sensibles donde se transfieren contraseñas con números de cuentas bancarias, lo cual exigió una demanda de conexiones seguras, creando canales seguros sobre una red insegura.

Su funcionamiento básico de HTTPS consiste en crear un canal seguro, donde pueda transferir información sensible, haciendo uso de un canal cifrado el cual se encuentre basado en SSL v.3 (Secure Sockets Layer/Capa de Sockets Seguro) o por su predecesor TLS (Transport Layer Security /Seguridad de la Capa de Transporte), también conocido como SSL versión 3.1, de tal forma que los datos enviados pasen a través de este canal seguro cifrado, para que en caso de que dicha conexión haya sido interceptada por algún tercero, sólo obtenga un flujo de datos cifrados el cual le sea imposible de descifrar, sin obtener dicha información sensible y vital de cualquier usuario, como serían las cuentas bancarias con contraseñas y por consiguiente los nombres, direcciones de domicilio, teléfonos e información sensible.

2.8.1. Protocolo SSL (Secure Sockets Layer) versión 3.0

El protocolo SSL de manera muy general es un protocolo de seguridad, el cual se encarga de proporcionar que una comunicación del tipo Cliente/Servidor vía internet, sea privada, de tal forma que evite la escucha, manipulación o falsificación del mensaje entre los entes participantes en la comunicación.

La creación e inicios de este protocolo datan de 1995, donde Netscape Communications Corp, entonces líder en navegadores, creó un paquete de seguridad SSL que ha ido creciendo y mejorando, creando diferentes versiones, pero la más utilizada para navegadores Web es la versión de SSL versión 3.0.

Dentro de la pila de los protocolos más comunes se añade una nueva capa de seguridad SSL entre la capa de aplicación y de transporte. Dicha capa se encarga de aceptar las solicitudes del navegador y enviarlas al protocolo TCP para transmitir al servidor (Figura 2.8.1).

Figura 2.8.1. Uso de la capa de seguridad SSL para los navegadores de usuarios domésticos.

La encomienda principal de SSL es que una vez establecida la conexión segura éste se encarga de dos aspectos muy importantes los cuales son la compresión y la encriptación de los datos transmitidos. Generalmente SSL es usado en navegadores Web donde el puerto utilizado cuando se hace uso de HTTP por encima de SSL, o mejor conocido como HTTPS es el Puerto 443 a diferencia del estándar (o puerto 80) con HTTP.

2.8.1.1. Funcionamiento básico de SSL

El protocolo SSL consta de dos subprotocolos uno se utiliza para crear la conexión mediante un canal seguro y el segundo para utilizar dicha conexión. Suponiendo que contamos con dos usuarios, el usuario A y usuario B, donde el usuario A quiere establecer una conexión segura con el usuario B.

I. El establecimiento de la conexión se realiza primero con el envío de un mensaje 1, el cual es la solicitud de ***"A"*** a ***"B"*** para que establezca la conexión; este mensaje de solicitud debe indicar la versión de SSL con la que cuenta el usuario A, así como sus predilecciones en cuestión de los algoritmos criptográficos y los métodos de comprensión a utilizar. Por otra parte se le añade a dicho mensaje una marca aleatoria, que en éste caso la llamaremos M_A la cual se usara posteriormente.
II. Más tarde el usuario B envía un mensaje 2, donde el usuario B realiza una elección entre los diferentes algoritmos criptográficos y de compresión que el usuario A puede soportar así como también envía su propia marca aleatoria M_B.
III. Una vez hecho lo anterior se envía un mensaje 3, que se encarga de enviar un certificado que contiene la llave pública, el cual puede o no estar firmado por una autoridad certificadora bien conocida y en caso de no ser así, éste envía una cadena de certificados que pueden seguirse hasta encontrar alguno firmado. Este certificado sólo se crea para un puerto y una dirección IP en particular.
IV. Llegando a éste punto B podría enviar muchos otros mensajes como la solicitud del certificado de clave pública de A, pero en éste punto termina con un mensaje 4 para indicarle al usuario A, que es su turno.
V. El usuario A le responde a B enviándole un mensaje 5, que contiene una clave premaestra aleatoria de 384 bits y se la encripta con la clave pública de B.
VI. El usuario A y B calculan la clave de sesión para encriptar los datos, donde dicha clave se deriva de la clave premaestra en conjunto de la combinación de las marcas aleatorias M_A y M_B. Una vez calculada la clave de sesión A le indica a B que cambie el cifrado.

VII. El mensaje 7 indica que ha terminado con el establecimiento del subprotocolo.

VIII. Y por último se envía el mensaje 8 y 9 que muestran que B confirma que ha recibido la indicación.

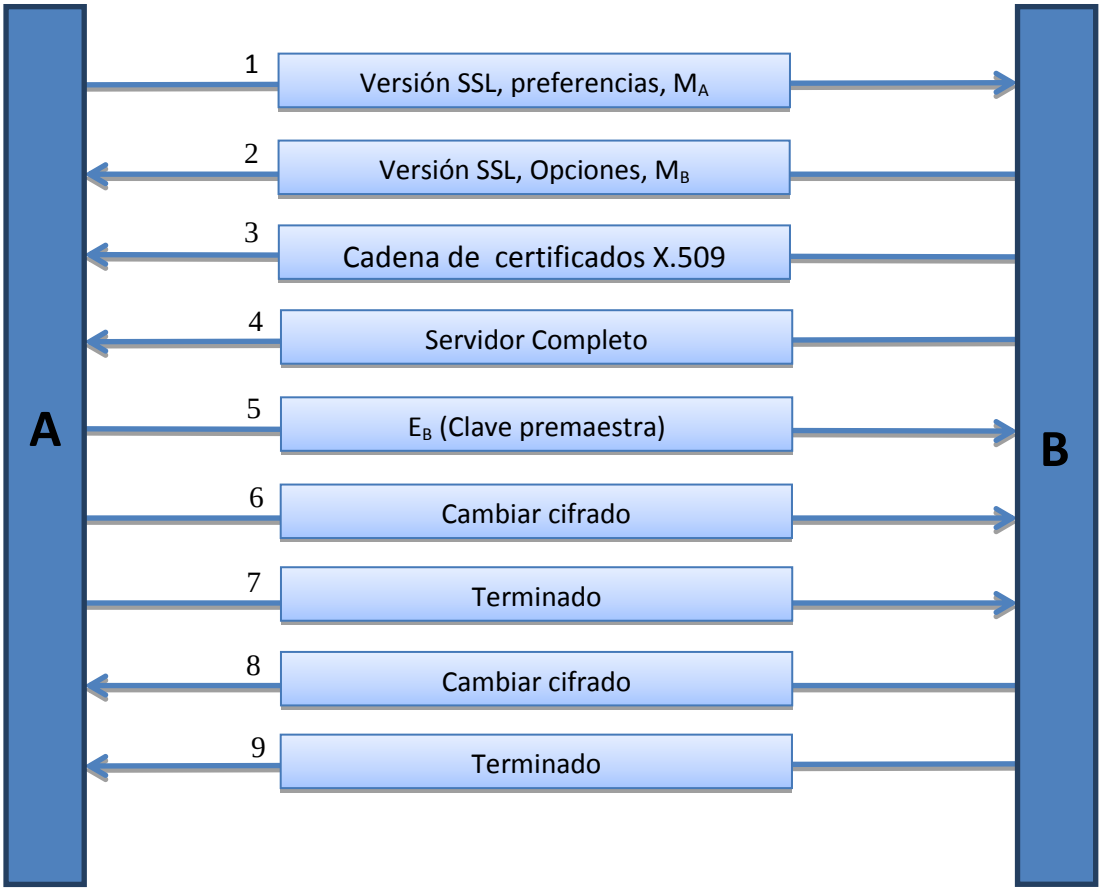


Figura 2.8.1. Funcionamiento Básico SSL.

El Protocolo SSL se encarga de proporcionar la seguridad en la conexión teniendo tres propiedades fundamentales, las cuales son:

- **Conexión Privada.** El cifrado se lleva a cabo después de un handshake inicial para definir su clave secreta; esto haciendo uso de la criptografía simétrica para el cifrado de los datos, usando por ejemplo DES, RC4 y más.
- **Autenticación.** La identidad de los compañeros puede ser autenticada mediante el uso de criptografía asimétrica o de clave pública, como por ejemplo el uso de RDA, DSS y más.

- **Conexión Fiable.** El transporte del mensaje debe incluir una comprobación de la integridad del mismo haciendo uso de una clave MAC; realizando una función hash (como SHA y MD5) para el cálculo de la MAC.

Los principales objetivos de este protocolo en orden de prioridad son:

I. ***Seguridad criptográfica.*** Para establecer una conexión segura entre ambas partes de la comunicación.
II. ***Interoperabilidad.*** Independientemente de los programadores, éstos deben tener la capacidad de desarrollar aplicaciones que hagan uso de SSL 3.0, y éstas a su vez intercambiar con éxito los parámetros criptográficos, sin el conocimiento de los respectivos códigos.
III. ***Extensibilidad.*** Proporciona nuevos métodos de cifrado de clave pública incorporándolos necesariamente, de tal manera que previene la necesidad de crear nuevos protocolos, aumentando riesgos posibles de vulnerabilidades en el mismo.
IV. ***Eficiencia relativa.*** Las operaciones criptográficas, suelen consumir muchos recursos a nuestros CPU, y mucho más las operaciones de clave pública y por dicha razón el protocolo SSL incorpora un tiempo opcional de almacenamiento en memoria caché, para con éstos reducir el número de conexiones que deben establecerse desde el principio y con ello reducir la actividad de la red haciéndola menos pesada.

2.8.2. HTTP sobre TLS (Transport Layer Security /Seguridad de la Capa de Transporte)

Como ya habíamos mencionado anteriormente el soporte de HTTP sobre TLS, es el predecesor de SSL y que también es conocido como SSL versión 3.1. Su origen comienza en 1996 cuando Netscape Communications Corp. mando a SSL a la IETF (Internet Engineering Task Force, o Grupo de Trabajo en Ingeniería de Internet) para su estandarización, dando como resultado TLS siendo éste la versión estándar de SSL, donde se le realizaron ciertos cambios para que fuera más seguro, aunque fueron pequeños cambios, fueron los suficientes para que SSL y TLS no pudieran interoperar aunque ambos protocolos permanecen sustancialmente iguales. Entre los cambios que se le realizaron, encontramos que cambia la manera de generar la clave de sesión que como ya sabíamos se generaba a partir de la clave pre maestra y las marcas aleatorias M_A y M_B, ésta forma se cambió de tal manera que dicha clave de sesión fuera más fuerte y mucho más difícil de criptoanalizar.

2.9. Ventajas y desventajas de HTTPS

- **Ventajas**
  - Privacidad.- La creación de los certificados permiten el cifrado, éste contiene su clave pública la cual permite que no se muestre información sensible a personas no autorizadas.
  - Autenticación.- Cada certificado que se genera es único ya que guarda la información de autenticidad de los usuarios.
  - Integridad.- La Autoridad de Certificación es una organización fiable que acepta solicitudes de certificados de entidades, las valida, genera certificados y mantiene la información de su estado, una entidad certificadora es VeriSing; donde el certificado ssl más sencillo para proteger la transferencia de datos confidenciales en sitios web, intranets y extranets cuenta por un año $399, por dos años $695 y por tres $995 todo éstos en dólares. Se encarga de verificar la identidad del propietario del certificado en el momento de su expedición del mismo.
- **Desventajas**
  - Es un poco más lento que HTTP, ya que necesita procesamiento para el uso de sus algoritmos criptográficos y de compresión, haciendo una sobrecarga en el equipo así como en la red.
  - El nivel de seguridad depende mucho del software de la implementación del navegador así como de los algoritmos de cifrado soportados.
  - HTTPS no puede evitar el robo de información confidencial de las páginas en caché en el navegador, puesto que la encriptación de los datos SSL es sólo durante la transmisión en la red y en el navegador de memoria es en texto claro.

En el caso de SSL en cual opera bajo HTTP, éste no tiene conocimiento de protocolos de nivel más alto, además de que SSL puede proporcionar un sólo certificado para un Puerto y una dirección IP de tal forma que no se recomienda el uso de los Hosts virtuales, ya que ese certificado se crea para un único servidor.

2.10. Software que interviene en el servicio Web

2.10.1. Sistema Operativo

Software libre.

El software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

El software libre se encuentra disponible gratuitamente o al precio de costo de la distribución a través de otros medios; por lo cual no hay que asociar software libre a "software gratuito", ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, éste tipo de software *no es libre* en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquel software que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual.

La distribución de un software requiere de una licencia; y es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Una de las más utilizadas es la *Licencia Pública General de GNU* (GNU GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL, el conjunto tiene que ser GPL.

Es decir, la licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia.

Distribución Linux.

Una distribución Linux es son conjunto de paquetes de software basadas en un núcleo del S.O. Linux, generalmente estas distribuciones incluyen las bibliotecas y herramientas del proyecto GNU/Linux. Existe una gran diversidad en cuanto a las distribuciones Linux (Figura 2.10.1).

Existe una gran cantidad de distribuciones Linux como las siguientes:

- Debian, una distribución mantenida por una red de desarrolladores voluntarios con un gran compromiso por los principios del software libre.
- Fedora, una distribución lanzada por Red Hat para la comunidad.
- Gentoo, una distribución orientada a usuarios avanzados, conocida por la similitud en su sistema de paquetes con el FreeBSD Ports, un sistema que automatiza la compilación de aplicaciones desde su código fuente.
- gOS, una distribución basada en Ubuntu para netbooks.
- Knoppix, la primera distribución live en correr completamente desde un medio extraíble. Está basada en Debian.
- Kubuntu, la versión en KDE de Ubuntu.
- Red Hat Enterprise Linux es la base de una estrategia de TI a largo plazo, compatible con las principales arquitecturas de hardware, incluyendo soporte y actualización de siete años (con opción para extenderlo a diez años). Una arquitectura modular, flexible, firme y las herramientas de administración ofrecen un control y escalabilidad mayores, y una gama de opciones de extensiones aumentan la disponibilidad de infraestructura y aplicaciones.

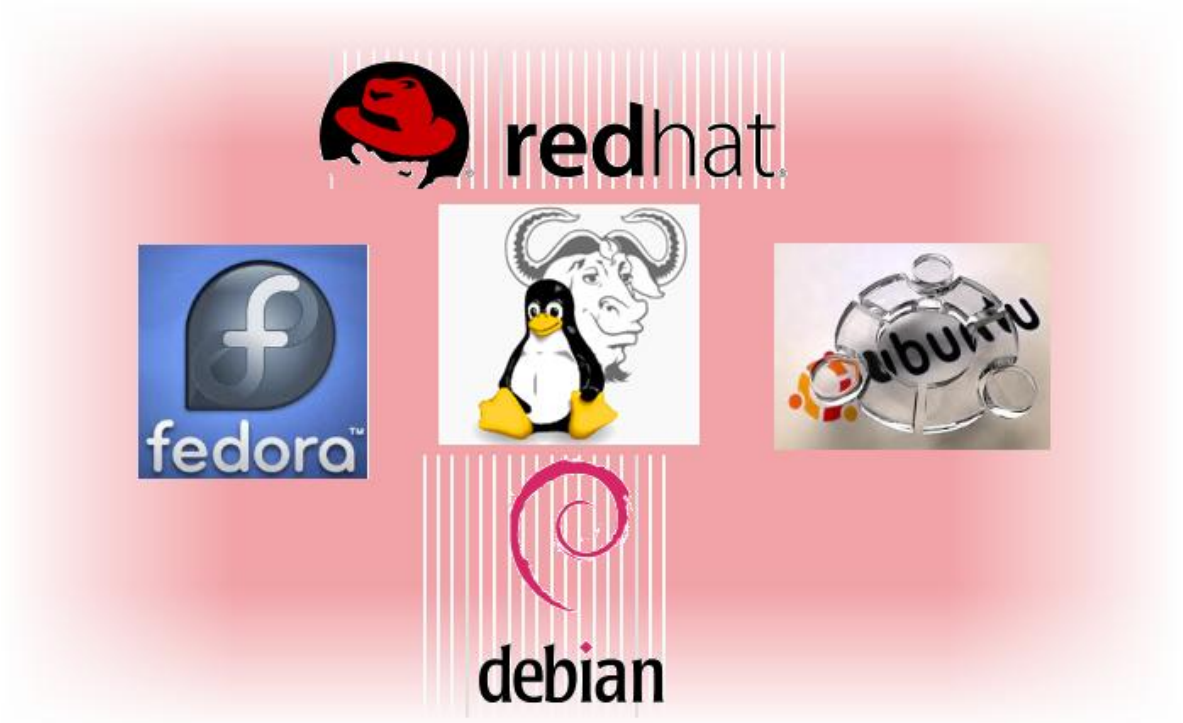


Figura 2.10.1. Algunas distribuciones Linux.

CentOS (Sistema Operativo de la Comunidad Empresarial /Community ENTerprise Operating System).

CentOS es una distribución de Linux que a diferencia de Fedora es de clase empresarial y orientada para servidores, basada en los fuentes libres y disponibles de Red-Hat Enterprise Linux.(***RHEL***), es un clon a nivel binario del mismo. Red-Hat libera de manera gratuita el código fuente de su producto debido a que la mayoría de su código es GPL, por tal motivo los desarrolladores de CentOS usan éste código, el cual es compilado y empaquetado, quitándole todas las marcas de Red-Hat creando un producto muy similar a RHEL, pero con la diferencia que éste es libre para ser usado por el público. Una de sus principales características es que cuenta con un ciclo de vida de 7 años y soporta casi todas las plataformas que soporta RHEL, es mucho más estable que Fedora ya que éste está diseñado para un entorno de escritorio a diferencia de CentOs que está más orientado a servicios de Red.

Por todo lo anterior para nuestro proyecto de tesis es que se decidió utilizar como sistema operativo a CentOS (Figura 2.10.2).

Figura 2.10.2. CentOS.

2.10.2. Servidor Web

Un Servidor Web es un programa que está diseñado para transferir hipertexto o bien para atender y responder a las diferentes peticiones de los navegadores Web, proporcionando los recursos solicitados mediante la implementación el protocolo HTTP, perteneciente a la capa de aplicación del modelo OSI.

El esquema básico del funcionamiento de un servidor Web es ejecutar infinitamente los siguientes procesos:

1. Esperar peticiones en el puerto TCP indicando (el estándar para HTTP es el 80).
2. Recibe una petición.
3. Busca el recurso.
4. Envía el recurso utilizando la misma conexión por la que recibió la petición.
5. Regresa al punto número dos.

El anterior esquema sólo sirve para manejar archivos estáticos, pero en base a éste se han diseñado los servidores HTTP, sólo variando el tipo de peticiones, ya sea sirviendo páginas estáticas, Servlets, CGIs (Common Gateway Interface) y más. Entre los servidores Web más importantes tenemos los siguientes:

- Apache.- Es el más común y utilizado en todo el mundo, es gratuito y de código abierto (configurable) y por lo tanto es multiplataforma.
- IIS de Microsoft.- Sólo funciona sobre plataformas Windows.
- Google.- Es un servidor Web propietario de google.
- Cherokee.- Es un servidor multiplataforma, su objetivo es ser rápido y completamente funcional siendo éste muy liviano.
- Nginx.- Es muy ligero y corre sobre plataformas Unix y Windows y se encuentra entre los primeros más populares.

- Lighhttpd.- Es uno de los más ligeros que hay en el mercado y está pensado para hacer cargas pesadas sin perder balance usando poca RAM y CPU además de ser gratuito.
- Thttpd.- Es de código libre, disponible para las variantes de UNIX, se caracteriza por ser simple, pequeño, portátil, rápido y seguro utilizando sólo los requerimientos mínimos de un servidor HTTP.
- SUN.- Sun Java System Web Server, pertenece a la casa de Sun y se emplea en entornos de este sistema aunque es multiplataforma al igual que apache.

Ranking de Servidores.

La clasificación mensual de Junio del 2012 realizada por Netcraft que ha recibido respuestas de 697,089,482 sitios, muestra que de los servidores Web más importantes, tres de ellos (Apache,Microsoft y nginx) han registrado un incremento de hostnames mientras que Google ha sufrido algunas pérdidas.

Las mayores ganancias las obtuvo Apache, el cual registró un aumento de más de 22 millones de hostnames, mientras que IIS presento algunas pérdidas; sin embargo tuvo un crecimiento de 3.5 millones de host (Figura 2.10.3 a Figura 2.10.6).

Ranking Servidores Julio 2012:

1. Apache con 448,452,703 de sites y el 64.33% del mercado
2. Microsoft con 95,891,537 de sites y el 13.76% del mercado
3. nginx con 72,881,755 de sites y el 10.46% del mercado
4. Google con 22,464,345 de sites y el 3.22% del mercado

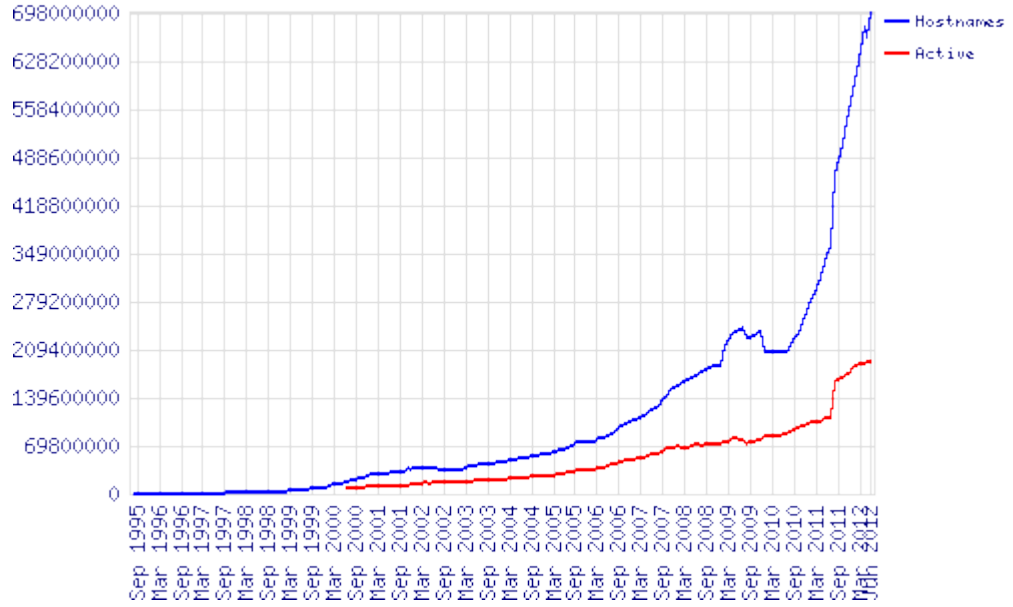


Figura 2.10.3. Total de sitios de todos los dominios[1].
Agosto 1995-Junio 2012.

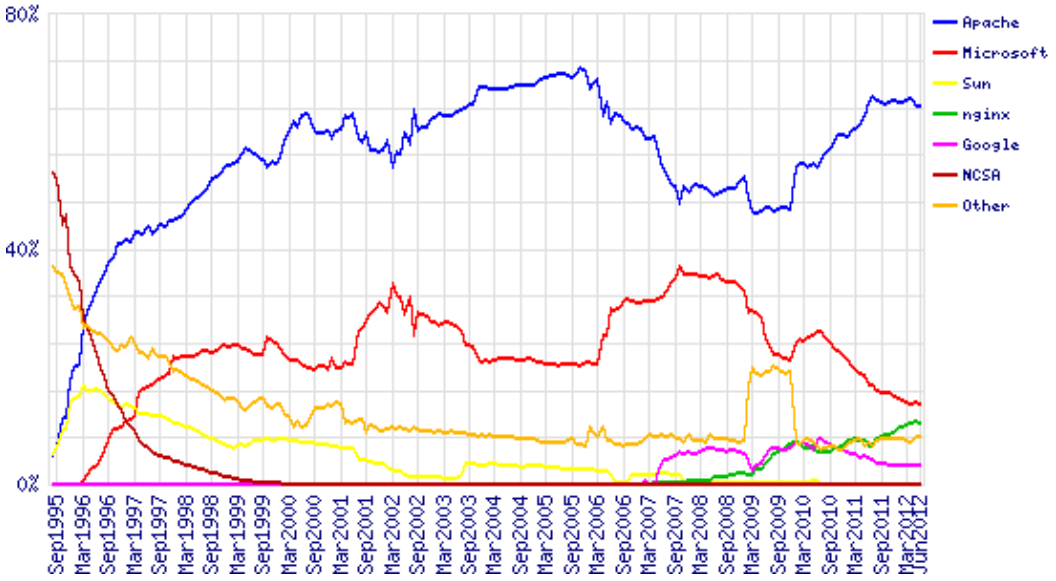


Figura 2.10.4. Mercado del top de servidores de todos los dominios.
Agosto 1995-Junio 2012

1) http://news.netcraft.com/archives/2010/07/16/july-2010-web-server.html

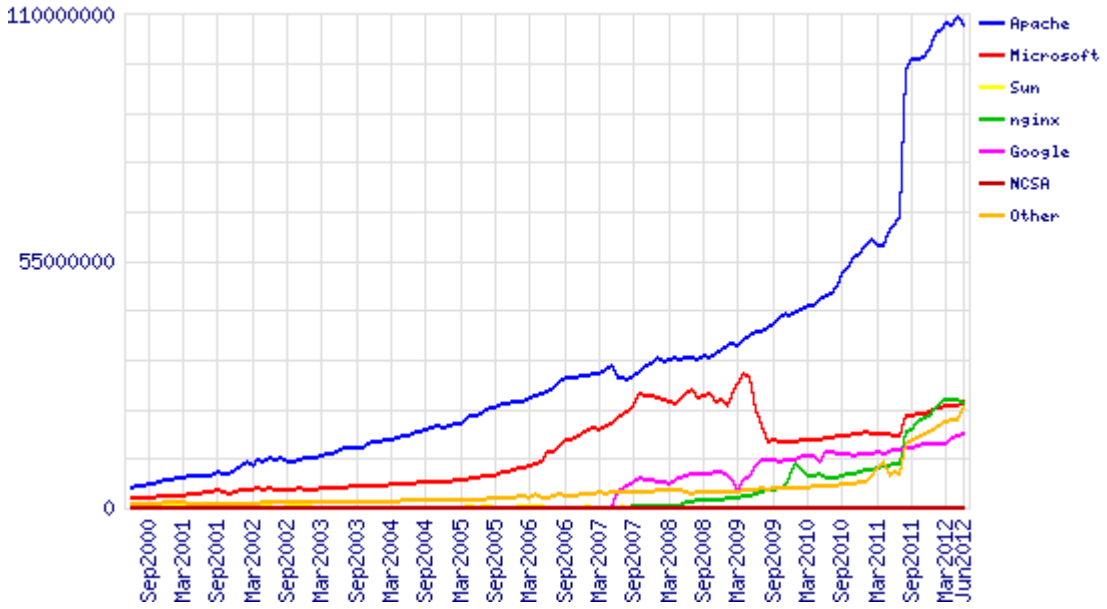


Figura 2.10.5. Total del top servidores activos en todos los dominios.
Septiembre 2000-Junio 2012

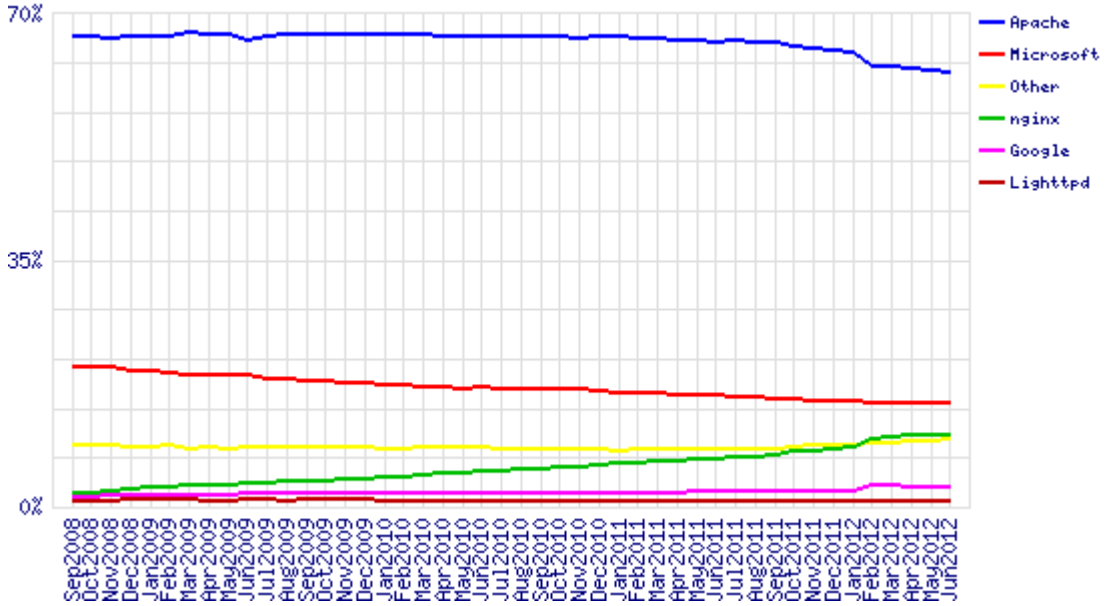


Figura 2.10.6. Mercado del top de servidores con sitios de mayor tráfico.
Septiembre 2008-Junio 2012

Como nos dimos cuenta el servidor apache es uno de los mejores de libre distribución y es de los más utilizados alrededor del mundo.

Ventajas:

- Fue desarrollado dentro del proyecto HTTP.
- Altamente configurable.
- Es modular lo cual ayuda a que la plataforma Apache sea competitiva incluso frente a rivales de alto precio.
- Posee código abierto. Este diseño abierto permite a cualquier programador crear una solución personalizada basada en el programa núcleo de Apache, o ampliar las funciones del software.
- Multiplataforma; es capaz de ejecutarse en todas las versiones del sistema operativo UNIX. Linux es compatible, así como los sistemas operativos Windows NT y MacOS.
- Cuenta con un gran soporte. Apache ha incorporado en su soporte a una amplia gama de lenguajes de programación web, como Perl, PHP y Python, JSP, Servlets.
- Posee licencia freeware (distribución sin costo).El servidor web apache es completamente gratuito y puede ser descargado por cualquier persona en el mundo, por el contrario del servidor Windows Server en cualquiera de sus versiones la cual si tiene un amplio costo por sí mismo, más aparte las configuraciones más avanzados.

Desventajas:

- Posee formatos de configuración no estándar.
- No cuenta con una buena administración.

Dado que nuestra línea va más ligada al software libre y por todas las ventajas con las que cuanta el servidor Web Apache es que seguiremos trabajando con éste (Figura 2.10.7).

Figura 2.10.7. Servidor Apache

2.11. Lenguajes de programación

Un Lenguaje de programación es un lenguaje artificial que puede ser usado para poder controlar el comportamiento de una máquina, generalmente de una computadora. Un lenguaje programación se compone de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que después serán interpretadas.

2.11.1. PHP (Preprocesador de Hipertexto /Hypertext Pre-processor)

PHP es un lenguaje de programación de código abierto y gratuito influido por C y PERL, ampliamente usado para el desarrollo Web dinámico pues puede ser incrustado dentro de código HTML. Está diseñado para ser ejecutado por medio de un intérprete del lado del servidor (sever-side scripting). Cuando el cliente hace una petición al servidor para que le envíe una página Web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.
Se puede hacer cualquier cosa en PHP que se pueda hacer con un script de CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, enviar y recibir cookies además cuenta con la capacidades de crear imágenes, archivos PDF, películas flash e incluso soporta un gran cantidad de bases de datos como: IBM DB2, MySQL, Oracle, PostgreSQL, Sybase, ODBC y más. Entre lo más destacable con lo que cuenta PHP es que cuenta con soporte para comunicarse con otros servicios como LDAP, IMAP, SNMP, HTTP, POP3 y muchos otros; además de contar con características para el procesamiento de texto con expresiones regulares extendidas o tipo PERL hasta procesadores de texto XML.

PHP puede ser ejecutado en la mayoría de los sistemas operativos como Linux, UNIX y sus variantes, Windows, RISCOS, Solaris; de igual forma es capaz de soportar la mayoría de los servidores Web como Apache, IIS de Microsoft, Netscape y muchos más.

La mayoría de los sitios que se encuentran alojados en el servidor principal que contiene el portal de la Facultad de Ingeniería se encuentran realizados en este lenguaje de programación, incluso el principal portal Web de la Facultad de Ingeniería ésta hecho en PHP (Figura 2.11.1).

Figura 2.11.1. Lenguaje PHP.

2.11.2. PERL (Lenguaje Práctico para la Extracción e Informe / Practical Extraction and Report Language)

PERL es un lenguaje de programación interpretado, el cual toma características del lenguaje C, Shell y AWK, es un lenguaje de propósito general. Perl es uno de los lenguajes que hereda ciertas estructuras de los intérpretes de comandos UNIX.

Muchas de las tareas de administración de UNIX se pueden simplificar con un programa en PERL, también se puede usar para crear archivos de texto. Una de las características más importantes que tiene y que se está usando con mayor frecuencia hoy en día es que ha encontrado su aplicación en escritura de CGI (Common Gateway Interface), o scripts ejecutados desde páginas de la Word Wide Web. Los programas PERL se ejecutan en el servidor como todos los programas CGI; otra de las características muy útiles e importantes de PERL (Figura 2.11.2) es que puede acceder a las bases de datos mediante el uso de algunos módulos adicionales.

Aunque fue desarrollado para un entorno UNIX, éste corre sobre casi todos los sistemas operativos, como Windows, Mac OS y las diferentes distribuciones de Linux.

Figura 2.11.2. Lenguaje PERL.

2.11.3. JSP (Páginas del Servidor java / Java Server Pages)

JSP son las iniciales de Java Server Pages, en español significa Páginas de Servidor Java. En sí es una tecnología orientada a crear páginas Web con programación en Java (Figura 2.11.3).

Las Java Server Pages (JSP) nos permiten separar la parte dinámica de nuestras páginas Web del código estático. Para ello:

- Simplemente escribimos el HTML (o XML) regular de la forma normal, usando cualquier herramienta de construcción de páginas Web.
- Encerramos el código de las partes dinámicas en unas etiquetas especiales, la mayoría de las cuales empiezan con "<%" y terminan con "%>.

Figura 2.11.3. Lenguaje JAVA.

- **Ventajas de los JSP's.**

Podemos crear aplicaciones Web que se ejecuten en varios servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma.

Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts ejecutables en el servidor en sintaxis Java. Por lo tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.

JSP tiene bastantes ventajas frente a otras orientaciones, como ASP o PHP. Se puede elegir entre diversas implementaciones, comerciales o gratuitas, sin tener que depender de un proveedor en particular. La ventaja fundamental es que tenemos toda la potencia del lenguaje Java a nuestro alcance, con sus ventajas como reusabilidad, robustez, multiplataforma, etc.

Como se puede leer anteriormente es software libre, porque no se necesita de ningún costo para el mismo en cuanto al mismo y cumple con los requerimientos del sistema.

- **El Motor JSP.**

El motor de las páginas JSP está basado en los servlets, que son programas en Java destinados a ejecutarse en el servidor, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, debido a que la programación de los servlets es más complicada.

En JSP creamos páginas de manera parecida a como se crean en ASP o PHP otras dos tecnologías ejecutables en el servidor. Generamos archivos con extensión .jsp que incluyen,

dentro de la estructura de etiquetas HTML las sentencias Java a ejecutar en el servidor. Antes de que sean funcionales los archivos, el motor JSP lleva a cabo una fase de traducción de esa página en un servlet, implementado en un archivo class (Byte codes de Java). Ésta fase de traducción se lleva a cabo habitualmente cuando se recibe la primera solicitud de la página jsp, |aunque existe la opción de precompilar el código para evitar ese tiempo de espera la primera vez que un cliente solicita la página.

- MÁQUINA VIRTUAL JAVA.

Para poder hacer uso de los JSP's requerimos de la máquina virtual de java.

Una **Máquina virtual Java** (en inglés *Java Virtual Machine*, **JVM**) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable incluso como lenguaje de entrada para un microprocesador físico. Java fue desarrollado originalmente por Sun Microsystems.

La JVM es una de las piezas fundamentales de la plataforma Java. Básicamente se sitúa en un nivel superior al Hardware del sistema sobre el que se pretende ejecutar la aplicación, y ésta actúa como un puente que entiende tanto el bytecode, como el sistema sobre el que se pretende ejecutar. Así, cuando se escribe una aplicación Java, se hace pensando que será ejecutada en una máquina virtual Java en concreto, siendo ésta la que en última instancia convierte de código bytecode a código nativo del dispositivo final.

La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera, que desde Sun Microsystems se han creado diferentes máquinas virtuales java para diferentes arquitecturas , así un programa .class escrito en un Windows puede ser interpretado en un entorno Linux. Tan sólo es necesario disponer de dicha máquina virtual para dichos entornos. De ahí el famoso axioma que sigue a Java, "escríbelo una vez, ejecútalo en cualquier parte", o "Write once, run any where".

La máquina virtual de Java puede estar implementada en software, hardware, una herramienta de desarrollo o un Web browser; lee y ejecuta código precompilado bytecode que es

independiente de la plataforma multiplataforma. La JVM provee definiciones para un conjunto de instrucciones, un conjunto de registros, un formato para archivos de clases, la pila, un heap con recolector de basura y un área de memoria. Cualquier implementación de la JVM que sea aprobada por SUN debe ser capaz de ejecutar cualquier clase que cumpla con la especificación (Figura 2.11.4).

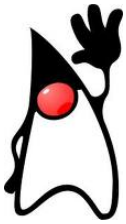


Figura 2.11.4. Mascota de JAVA DUKE.

- JAKARTA-TOMCAT.

Para el uso de los JSP's requerimos de un servidor Web que soporte los JSP's por lo mismo usaremos Tomcat. Primero necesitamos saber que Jakarta es un proyecto que fue creado y mantenido bajo soluciones open source en la plataforma java. Éste implica que el servidor donde se instale tenga previamente instalada la plataforma Java (JSDK).

Los productos Jakarta son desarrollados y distribuidos a través de varios subproyectos.

ENTORNO

Tomcat es un servidor de aplicaciones con soporte de Servlets y JSP's. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache haciendo uso de conectores entre apache y Tomcat.

Tomcat puede funcionar como servidor Web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java (Figura 2.11.5).

Figura 2.11.5. Mascota de Apache-Tomcat.

2.12. Software de Administración

Este software será de gran utilidad, pues nos ayudara a realizar una administración tanto de usuarios como los servicios de nuestro servidor así como también nos ayudara a automatizar algunas tareas que realiza nuestro servidor.

2.12.1. Webmin

Webmin es una interfaz basada en Web que nos ayuda a simplificar la administración de sistemas Linux o UNIX. Generalmente se han tenido que editar de manera manual los archivos de configuración y ejecutar comandos para crear cuentas en los diferentes sistemas de nuestros servidores; ahora todas estas tareas que realizábamos o ejecutábamos de manera manual por medio de Webmin las podemos realizar por medio de una interfaz Web de manera fácil y sencilla haciendo mucho más fácil el trabajo de administración del sistema.

Algunas de las cosas que podemos hacer con Webmin son:

- Crear, editar, suprimir las cuentas de entrada de Linux o Unix.
- Exportar archivos y directorios a otros sistemas.
- Establecer cuotas en disco, para controlar el espacio y a archivos de los usuarios.
- Instalación, visualización y eliminación de software en diferentes formatos.
- Cambio de dirección IP del sistema, configuración de DNS y configuración de enrutamiento.
- Crear y configurar sitios WEB virtuales para el servidor Web Apache.
- Administración de bases de datos, tablas y campos de un servidor MySQL o PostgreSQL.

Estos son algunas de las funciones que se pueden hacer con Webmin aunque no son los únicos, pues también nos permite configurar todos los servicios más comunes en servidores Linux o Unix.

El desarrollo de Webmin fue casi completamente realizado por Jamie Cameron, recibiendo sólo algunas contribuciones de parches y traducciones a otros idiomas, así como la realización de algunos módulos realizados por otras personas.

Webmin se encuentra bajo la licencia BSD, significando éste que puede ser de libre distribución y modificado para uso comercial y no comercial. Dado que Webmin apoya el concepto de módulos, cualquier persona puede desarrollar y distribuir sus propios módulos de Webmin así como también puede instalar o desinstalar de forma independiente dichos módulos del resto del programa, pues cada módulo se encarga de algún servicio o servidor (Figura 2.12.1).

Figura 2.12.1. Administración de Linux mediante Webmin.

2.12.2. Google Analytics

Google Analytics es un servicio de tipo gratuito que se encarga de obtener estadísticas de sitios Web dependiendo del interés de cada usuario, se puede obtener informes como.

- Cuál es contenido más visitado
- Cuál es el promedio de visitas de la página y la hora local de las visitas
- Qué anuncios son los que más atraen visitas a tu sitio.
- Cuenta con una sencilla herramienta de filtros para controlar los datos qué deseamos incluir en nuestros informes.
- Obtiene datos de los usuarios, como que navegadores utilizan, si acceden directamente o mediante sitios de referencia, que tipo de navegadores, tipo de sistemas operativos que usan, las resoluciones de pantalla que utiliza, hasta si nos visitan mediante un dispositivo de telefonía móvil, entre otros más.

Gracias a la recopilación de estos datos, se podrán realizar recomendaciones para la creación de nuestros sitios Web institucionales.

El funcionamiento de Google Analytics se realiza por medio de un código JavaScript que es colocado en las páginas de su sitio Web para recopilar información de los usuarios que lo visitan, efectuando un seguimiento anónimo de la forma en que los usuarios interactúan con su sitio Web (Figura 2.12.2).

Figura 2.12.2. Análisis Web con Google Analytics

2.12.3. AWSTATS

AWSTATS es un software de código abierto (GPL, Licencia Pública General) escrito en perl, que sirve para darnos un informe de análisis Web detallado y muy completo en cuanto a un servidor Web, uno de correo o bien un servidor ftp, estos informes los crea a partir del análisis de los logs de nuestro sistema, presentándonos de manera gráfica en forma de barras y tablas de gráficos en un formato html pudiéndolos visualizar a través de cualquier navegador (Figura 2.12.3).

Entre los informes principales que nos puede brindar awstats encontramos los siguientes:

- Número de visitas y visitantes únicos.
- usuarios autenticados en el sistema y última autenticación.
- Días de la semana y horas pico, páginas, hits, visitas de otros países, dominios.
- Sitos más vistos.
- Tipos de archivos usados.
- Sistemas operativos utilizados.
- Navegadores utilizados.
- Errores de HTTP.
- Tamaño de pantalla de los usuarios.
- Motores Web, etc.

Figura 2.12.3. Obtención de estadísticas del servidor mediante AWStat.

2.13. Seguridad

Hoy en día los problemas de seguridad en los sistemas informáticos no son nada nuevos y a medida que han crecido éstos, de igual forma se han ido incrementando. Primero que nada definiremos que es "Seguridad". Seguridad es el conjunto de protecciones como controles, herramientas reglas, buenas prácticas recomendaciones, etc. que nos permiten resguardar algo o a alguien ante un comportamiento inesperado.

2.13.1. Seguridad en un Servidor Web

Muchas empresas e instituciones hacen el uso de sitios Web, alojados en servidores dedicados a brindar este servicio, para la divulgación de información en el caso de instituciones o bien para realizar transacciones comerciales para las empresas. Cómo podemos darnos cuenta ya no sólo hablamos de información sino también de dinero, es por eso que se le debe tomar gran importancia a la seguridad de nuestros Servidores Web.

Como ya se mencionó anteriormente la información que circula a través del Word Wide Web puede ser muy valiosa para otras personas quienes se aprovecharan de los puntos débiles de quienes alojan nuestros sitios, que en éste caso son los servidores Web.

En la actualidad dada la competencia de mercado, la demanda de éste, así como la necesidad de compartir información a grandes distancias, ha obligado a que tanto empresas como instituciones utilicen cada vez más en Word Wide Web, creando sus propios portales Web para compartir dicha información y para el comercio electrónico. Siendo dichos portales Web la imagen propia de una empresa o una institución; de tal manera que en el momento en que es violada la seguridad del servidor que aloja dicho portal, pueda impactar en dañar la reputación de dicha empresa o institución, teniendo como consecuencia la pérdida del prestigio, el cambio completo de la página Web, la modificación de datos de los usuarios, intrusión en el servidor Web, la pérdida de información valiosa y confidencial de alguna empresa e incluso dinero como

es el caso de los bancos, comercio electrónico y los propios usuarios que hacen uso de dichos portales realizando transacciones.

Como podemos observar los servidores Web deben contar con prevenir todo éste tipo de incidentes que causen un comportamiento anormal del servicio que se está brindando, es por eso que debemos tener mucho cuidado en las posibles fallas de seguridad que pueda explotar algún transgresor, pues es mucho más engorroso y tardado tratar de recuperarse de un incidente de seguridad que tratar de implementar medidas preventivas.

La mayor parte de éstos incidentes de seguridad son exitosos debido a malas configuraciones en el servidor o bien a errores de diseño o simplemente el uso de entornos nada serios.

Si bien definimos anteriormente que es la seguridad, ahora definiremos lo que es seguridad informática, la cual será nuestro punto de partida para prevenir cualquier tipo de incidente en nuestro servidor Web. *"SEGURIDAD INFORMÁTICA"* es el conjunto de protecciones como controles, herramientas reglas, buenas prácticas recomendaciones, estrategias etc. que nos permiten resguardar algo o a alguien ante un comportamiento inesperado, garantizando la integridad, disponibilidad y la confidencialidad de la información de una entidad.

- ***Integridad:*** Ésta se refiere a la seguridad de que una información no ha sido alterada, borrada, reordenada, copiada por alguien no autorizado en el momento de la transmisión o en el propio equipo de origen. La pérdida de la integridad puede tener como consecuencias la toma de decisiones erróneas e incluso hasta el fraude.
- ***Disponibilidad:*** Es la seguridad que tenemos para tener acceso a la información en el momento en que se necesite y a la hora que se necesite para evitar pérdidas o bloqueos en la productividad de ciertos procesos de alguna entidad. Está nos garantiza que los usuarios autorizados tengan acceso a la información y a los recursos relacionados con la misma todas la veces que se requiera.
- ***Confidencialidad:*** Se refiere a que la información que es difundida sólo pueda ser vista por personal autorizado. Pues al perder la confidencialidad nos puede ocasionar grandes problemas, como es el caso de las empresas privadas que manejan una enorme cantidad de información confidencial y si se llega a caer esta información en manos no deseadas podemos llegar a tener hasta problemas legales y la pérdida de credibilidad como institución seria incluso la pérdida de nuestro negocio o empresa.

Gracias a que el Web ha ido tomando cada día más fuerza con el paso de los años además que es de muy fácil usos para promover información sino que también como medio de comercio para las empresas, los servidores Web son un blanco perfecto para recibir múltiples ataques informáticos.

Ejemplos de algunos incidentes de seguridad del Web:

- Cuando tratamos de entrar a nuestro portal y vemos que nuestro sitio está completamente cambiado queriendo decir, que tuvieron acceso a nuestra información o simplemente insertaron código mediante algún script, en ésta parte no sólo es culpa del administrador de servidor Web sino también de las buenas prácticas o técnicas de programación de los Webmaster.
- Ataques de denegación de servicio (DoS) los cuales pueden ser dirigidos a servidores Web dejando inaccesible el servicio para aquellos usuarios que deseen utilizarlo.
- Ataques de fuerza bruta para conseguir cuentas de usuarios y comprometer algunas cuentas de usuario incluso todo el servidor.
- Implantación en el servidor comprometiendo el sistema incluso la red, teniendo acceso a todo el sistema como carpetas archivos, ejecutando comandos e incluso instalando software malicioso.
- SQL inyection, insertando código malicioso en sitios modificando información valiosa de los propios sitios, e incluso con ello accediendo a una sesión si autentificarse en el sistema.
- Escaneos de puertos, método de descubrir canales de comunicación susceptibles de ser explotados.
- Implantación de Back Doors o puertas traseras las cuales son piezas de código en un programa que permite a quien las conoce saltarse los métodos usuales de autenticación para realizar ciertas tareas.

2.13.2. Estrategias de seguridad

Cuando se desea brindar de protección y tener un cierto nivel de confianza, las organizaciones deben implementar algunas acciones para mantener seguro su servidor Web contestándose las siguientes preguntas:

- ¿Qué quiero proteger?

- ¿De qué se quiere proteger?
- ¿Cómo lo voy a proteger?

a) ¿Qué quiero proteger?

Es vital e indispensable identificar de manera puntual los activos, bines tangible e intangibles que son importantes y que como ente, no estamos dispuestos a perder o que de alguna manera sufran algún tipo de modificación.

b) ¿De qué se quiere proteger?

Se tratar de identificar todas las acciones que pueden dañar los activos que son importantes para un usuario u organización, es decir hacer conciencia de los peligros que pueden correr todos los bienes que no deseo que los dañen debido a amenazas y vulnerabilidades. ¿Pero que es una amenaza y una vulnerabilidad?

1) **Amenaza:** es todo aquello que puede dañar incluso destruir o no, que siempre se encuentra de manera latente con un alto riesgo y puede llegar a culminar o no. Las amenazas provienen de 5 fuentes.

   a) **Desastres.-** Tiene que ver con el comportamiento de la naturaleza y el hombre no puede controlarlas ni predecirlas.
   b) **Errores de Hardware.-** Se refiere a las posibles fallas físicas totales o parciales, en cualquiera de los dispositivos de nuestro esquema.
   c) **Problemas de Software.**- Como tratar las posibles fallas debido a incorrectas configuraciones de implementación en el sistema.
   d) **Errores de red.-** Con éstos nos referimos a problemas debido al mal uso o implementación del diseño de la red.
   e) **Humana.-** Ésta es una de las más importantes pues el hombre es el eslabón más grande, generalmente los incidentes de seguridad se dan debido a la ignorancia y la falta de capacitación hacia los usuarios, por descuido de los mismos, debido a ingeniería social o simplemente personas que lo hacen por diversión.

2) **Vulnerabilidades:** Una vulnerabilidad es debilidad que puede ser explotada por una amenaza. Estas se identifican considerando algo que no se ha hecho hasta el momento o todos los puntos débiles que no se tomaron en cuenta.

Si hablamos de amenazas y de vulnerabilidades también tenemos que hablar de lo que es un ataque.

Un Ataque consiste en aprovechar alguna debilidad o falla (vulnerabilidad) en el software, hardware, o personal que forma parte de algún ambiente informático; con el fin de dañar algo o a alguien. En pocas palabras un ataque es la culminación de una amenaza (Figura 2.13.1).

Figura 2.13.1. Cuando las amenazas explotan las vulnerabilidades nos ocasionan ataques.

c) ¿Cómo lo voy a proteger?

Una vez identificando las amenazas y vulnerabilidades se planea una estrategia para proteger la disponibilidad, la integridad y la confidencialidad de los datos de nuestro sistema, con el fin de

implantar herramientas, desarrollar directivas y controles de seguridad apropiados como los siguientes:

- Deshabilitar servicios y cuentas no autorizadas.
- Actualización del Sistema Operativo y aplicaciones (parches.)
- Uso de buenas contraseñas.
- Chequeo de integridad de aplicaciones y S.O.
- Análisis periódico de logs.
- Utilización de firewalls.
- Realización de respaldo de manera periódica.
- Verificación periódica de servicios activos y vulnerabilidades presentes.
- Encriptación de tráfico.
- Desarrollo seguro de aplicaciones Web.
- Concientización de los usuarios.
- Crear planes de contingencia y recuperación.
- Crear mecanismos de seguridad.
- Definición y uso de Políticas, Procedimientos y buenas prácticas.

2.13.3. Mecanismos de seguridad

Los mecanismos de seguridad son el conjunto de controles que permiten implementar servicios de seguridad para así disminuir las vulnerabilidades mejorando la seguridad en un sistema de información; los mecanismos pueden ser físicos o lógicos, buenas práctica, estándares o recomendaciones. Dichos servicios hacen uso de uno o varios mecanismos de seguridad y se clasifican en:

a) **Control de acceso.-** Se refiere a que el acceso a los activos ya sea información, entidades físicas, recursos sean controlados y limitados por algún ente ya sea de software o hardware o ambos, de tal manera que nos proteja de uso y manipulación de activos no autorizados. Un ejemplo de éstos serían los candados, credenciales, contraseñas, lector de huella digital o de caras, tarjetas digitales, firewalls, etc.
b) **Integridad.-** Nos dice que sólo las entidades autorizadas puedan modificar cierta información como escribir, cambiar, borrar, crear o restaurar. El uso de este mecanismo de seguridad nos asegura que los datos recibidos no han sido modificados,

por ejemplo mediante el uso de resúmenes Hash criptográficos, por medio de la asignación de permisos de lectura, escritura y ejecución, creación de respaldos y más.

c) **No repudio.-** Éste protege a un usuario de forma que otro usuario niegue posteriormente que realizo algo y que alguno de los dos lo niegue, por ejemplo que establecieran una comunicación entre ambos o que se haya quedado en algún acuerdo. Éste tipo de protección se realiza por medio de una colección de evidencias irrefutables que permitan la resolución de cualquier desacuerdo. Por ejemplo el uso de cámaras de vigilancia, bitácoras y el más usado el uso de firmas digitales.

d) **Confidencialidad.-** Permite que la información como los datos intercambiados o sólo segmentos de información, sólo sea accesible únicamente por las entidades autorizadas. Por ejemplo mediante el uso de cifrado, el uso de canales seguros, no comentar las contraseñas propias etc.

e) **Autenticación.-** Solicita de una identificación del origen del mensaje, asegurando que la entidad no es falsa. Por ejemplo el uso de sensores biométricos, servidores de autenticación, sensor de huella digital, tarjetas de banda magnética, el uso de contraseñas, firmas digitales.

f) **Disponibilidad.-** Nos dice que los recursos del sistema informático estén siempre disponibles a las entidades autorizadas cuando los necesiten. Por ejemplo redundancia de redes y servicios, replicación de la información, horarios de servicios etc.

2.14. Manejo de Seguridad en Servidores Web

Observamos con más frecuencia que los sitios Web maneja una gran cantidad de información de carácter confidencial; requiriendo de mecanismos de seguridad que nos garanticen que nuestra información no caiga en manos mal intencionadas o que no deberían por qué tener nuestra información por ningún motivo. En un sistema que brinda el servicio Web debe cumplir con la integridad, disponibilidad y confidencialidad (Figura 2.14.1) de toda la información que se maneja en el sistema, por ello que debe planearse bien la instalación y configuración de éste servicio considerando todos y cada uno de los servicios que brindara, implantando medidas de seguridad, cumpliendo las políticas de la organización, llevando a cabo

planes de contingencia y sobre todo y la más importantes la capacitación de las personas que están a cargo de éste servicio pues generalmente los problemas de seguridad en un servidor Web se dan más que nada por la mala planeación, implementación y prevención de incidentes de seguridad de nuestros equipos.

Figura 2.14.1. Átomo de seguridad.

Lo que debemos considerar en la planeación de un servidor Web es lo siguiente:

- Identificación y evaluación de los activos; cuales son los bienes importantes dándoles una escala de prioridad. En ésta parte de identificación veremos, si es un servidor Web dedicado que es lo más recomendable o se encargara de brindar algún otro servicio dependiendo de las característica de hardware del mismo, identificar qué tipo de información será almacenada y transmitida a través del servidor Web.
- Qué tipo de software como el sistema operativo qué aplicaciones manejarán, transformarán y administrarán nuestros datos.
- Ser conscientes del tipo de hardware con el que contamos para brindar el servicio Web y que éste sea soportado por el mismo, como las instalaciones eléctricas, si cuenta con las

instalaciones eléctricas adecuadas, ininterrumpidas y si cuenta con sistemas de alimentación alternativas (no-break).

- Empezar por definir qué tipo de host vamos a construir, si es del tipo privado o público.
- Si cuenta con soportes dispositivos de almacenamiento externo al equipo como CD, DVD, discos duros, para la realización de respaldos.
- Verificar si las instalaciones son adecuadas y seguras para albergar nuestro sistema de información como controles ambientales de humedad y temperatura, si cuenta con controles de acceso sólo para el personal autorizado, como candados, cámaras de seguridad, etc.
- Identificar qué tipos de usuarios existirán en nuestro sistema así como los privilegios que contara cada tipo de usuario.
- Identificar qué tipo de personas serán las que interactúen con nuestro sistema, si son programadores, si son usuarios internos o externos a nuestra organización.
- Identificar y valorar las amenazas que puedan afectar la seguridad de los activos.
- Identificar y valorar las medidas de seguridad existentes.
- Identificar y valorar las vulnerabilidades de los activos y que amenazas son las que las pueden afectar.
- Determinar el impacto que pudiera provocar un ataque, es decir el resultado de un ataque, como lo es una denegación de servicio, la modificación, suplantación etc.
- Identificar los riesgos residuales los cuales son los mínimos riesgos que estoy dispuesta a aceptar como los que no puedo eliminar o los que no se encuentran dentro de nuestro control.
- Implantar controles como :
    - Buenas prácticas
    - Estándares
    - Políticas de seguridad
    - Implantación de controles de hardware o de software que proporcione de seguridad, como firewall, IDS, contraseñas, uso de cifrado y canales seguros, sensores biométricos etc.
    - Crear un plan de contingencia en caso de desastres.
    - Crear planes de redundancia de recuperación y respaldo.
- Realizar revisiones periódicas en el sistema en busca de anomalías y realizar pruebas de seguridad.
- Preparar informes de la evaluación de nuestro sistema.

- Por último capacitar a nuestro personal que administrara nuestro sistema así como los que harán uso de él, pues las personas son el eslabón más débil en cuestiones de seguridad.

2.14.1. Seguridad en la transmisión del Servidor Web

Como ya se mencionó anteriormente, la información en internet viaja en un medio totalmente inseguro; por tal motivo es necesario establecer protocolos de comunicación que nos proporcionen la seguridad de que nuestra información viaja segura a través del medio de comunicación de tal forma que viaje cifrada la información. Entre los protocolos de seguridad más utilizados, tenemos los siguientes: SSH, SSL, TLS, HTTPS, y PGP.

2.14.1.1. Protocolo SSH (Intérprete de órdenes seguras/ Secure Shell)

SSH es un protocolo para crear una conexión de forma segura entre dos sistemas utilizando al arquitectura cliente/servidor, generalmente es utilizado común remplazo del comando telnet para establecer sesiones remotas a través de una red, permitiéndonos administrar equipos mediante un intérprete de comandos, otra de sus características en que nos brinda la posibilidad de hacer copias de datos de manera segura ya que a diferencia de Telnet y de FTP, SSH encripta la sesión de conexión evitando que puedan obtener la contraseñas no encriptadas.

Sus características principales del protocolo son:

- Después que realiza una conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente durante sus siguientes sesiones.
- El cliente trasmite su información de autenticación al servidor usando una encriptación de 128 bits.
- Todos los datos enviados y recibidos durante la sesión se transfieren por medio de una encriptación de 128 bits, haciéndolos difíciles de descifrar y leer.
- El cliente transmite su información para autenticación con el servidor, tanto usuario y contraseña de manera cifrada.
- El cliente cuenta con la posibilidad de reenviar aplicaciones X11 desde el servidor.

Como vemos SSH cifra todo lo que envía y recibe de tal manera que lo podemos usar para proteger y asegurar otros protocolos inseguros mediante la técnica de reenvío por puertos o post forwarding .

2.14.1.2. Protocolo SSL (Protocolo de capa de conexión segura/Secure Sockets Layer)

El principal objetivo de este protocolo es brindar autenticación, privacidad y confidencialidad entre extremos sobre internet que se comuniquen, generalmente sólo el servidor es autenticado, mientras que el cliente no.

SSL es un protocolo dividido en capas, y en cada capa los mensajes pueden incluir campos como, la longitud, descripción y el contenido; una vez que sean enviado los mensajes toma los que serán transmitidos, los fragmente en bloques de datos manejables si es necesario los comprime, aplica una MAC cifra y transmite el resultado. Ya que se reciben los datos, éstos son descifrados, verificados, descomprimidos y ensamblados, entregándolos a los clientes.

SSL cuenta con una serie de fases básicas:

- Negociar entre el cliente y el servidor el algoritmo que será usado en la comunicación éntrelos cuales se encuentran los siguientes:
    - Criptografía de clave pública: RSA, Diffie – Hellman, DSA o Fortezza.
    - Cifrado Simétrico: IDEA, DES, 3DES, AES, RC2 y RC4.
    - Con funciones Hash: MD5 o de la familia SHA.
- Realiza un intercambio de claves públicas y una autenticación basada en la creación de certificados digitales.
- Realizar cifrado del tráfico basado en cifrado simétrico.

Este protocolo se ejecuta en una capa entre los protocolos de aplicación como los son el HTTP, SMTP, NTP y sobre los protocolos de transporte TCP, generalmente éste se usa para proteger HTTP para formar HTTPS.

2.14.1.3. Protocolo TLS (Seguridad en la Capa de Transporte/Transport Layer Security).

TLS fue el sucesor de SSL; éste es un protocolo para establecer una conexión segura entre cliente y servidor ya que a diferencia de SSL, TLS es capaz de autenticar tanto en cliente como en servidor, creando una conexión cifrada entre ambas partes, ofreciendo privacidad e integridad de los datos entre ambas partes en la comunicación.

TLS al igual que SSL hace uso de las mismas fases básicas, otra de sus características es que es interoperable ya que aplicaciones distintas deben ser capaces de intercambiar parámetro criptográficos sin la necesidad que ninguno de los dos conozca el código de la otra, también es extensibles pues permite incorporar nuevos algoritmos criptográficos y algo muy importante es que cuenta con un esquema de cache de sesiones que permite reducir el número de sesiones que deben inicializarse desde cero usando criptografía de clave pública.

El protocolo TLS está compuesto por dos capas:

- **TLS Record (De Registro).-** Este protocolo ofrece seguridad en un transporte fiable como el TCP en una conexión y sus propiedades básicas son:
    - **Conexión privada:** hace uso de criptografía simétrica (DES, AES, RC4, etc.) para el cifrado de los datos, generando claves sólo una vez por cada conexión basándose en un secreto negociado por otro protocolo (TLS Handshake).
    - **Conexión confiable:** el transporte de los mensajes incluye una verificación de integridad de mensaje, por medio del uso de MAC con llave.
- **TLS Handshake (De mutuo acuerdo).-** Éste ofrece seguridad en la conexión y tiene tres propiedades básicas:
    - La identidad el interlocutor puede ser autentificada haciendo uso de criptografía de clave pública y puede ser de manera opcional. Pero es necesaria al menos para uno de los interlocutores.
    - La negociación de un secreto compartido es segura.
    - La negociación es confiable, nadie puede modificar la negociación sin ser detectado por los interlocutores.

2.14.1.4. Protocolo HTTPS (Protocolo de Transferencia de Hipertexto Seguro)

Este protocolo es una versión segura del protocolo HTTP, y éste usa un cifrado basado en SSL, de tal manera que forma un canal seguro donde el nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente, este protocolo es mucho más recomendado para el uso de aplicaciones que manejan el uso de información sensible como por ejemplo entidades bancarias donde su información es de suma confidencialidad.

Como ya habías mencionado anteriormente, el protocolo HTTP puede ser usado sobre algún otro protocolo de seguridad ya sea SSL o TLS, de la misma forma que se utiliza en el protocolo TCP.

A diferencia de HTTP, HTTPS trabaja por defecto por el puerto 443 TCP, y antes de enviar los datos realiza algunas acciones previas.

Para hacer ésta negociación, el cliente, envía al servidor las opciones de cifrado, compresión y versión de SSL junto con algunos bytes aleatorios llamados *Challenge de Cliente*. El servidor, escoge las opciones de cifrado, compresión y versión de SSL entre las que ha ofertado el cliente y le envía su decisión y su certificado. Ambos negocian la clave secreta llamada *master secret* y usando ésta clave, la *Challenge de Cliente* y las opciones pactadas se envían la información encriptada de tal manera que de ser interceptada no se puede descifrar.

2.14.1.5. PGP (Buena Privacidad/Pretty Good Privacy).

Su objetivo principal de este protocolo es proteger la información distribuida a través de Internet, haciendo uso de criptográfica asimétrica y firmas digitales; PGP también puede ser utilizado para proteger la información que se encuentra almacenada en nuestros discos duros brindándonos un alto nivel de seguridad. Otra de las características con las que cuenta PGP es que combina criptografía simétrica como asimétrica siendo éste un criptosistema híbrido.

El Funcionamiento de PGP es el siguiente:

a) **Cifrando con PGP.-** PGP comienza creando una llave de sesión, la cual es una llave secreta que utiliza una sola vez, esta llave es un número aleatorio generado utilizando como función generadora, los movimientos del ratón, esta llave de sesión trabaja con un algoritmo de cifrado simétrico seguro y muy rápido, para cifrar el texto plano, el resultado es el texto cifrado. Una vez que los datos son cifrados, la llave de sesión es entonces cifrada con la llave pública del que recibe el mensaje, junto con esta llave cifrada, se envía también el texto cifrado.

b) **Descifrando con PGP.-** para el proceso de descifrar, el receptor del mensaje utiliza su llave privada para descifrar la llave de sesión que viene acompañando al mensaje cifrado, entonces utilizando la llave de sesión, se descifra el texto cifrado. La combinación de los dos métodos de criptografía, combina la practicidad de la asimétrica con la velocidad de la criptografía simétrica, la cual es en proporción mil veces más rápida. La criptografía de llave pública, ofrece una solución al problema de distribución de llaves y transmisión de datos, y utilizados ambos sistemas, se obtiene una solución mejorada, sin sacrificar en seguridad.

2.15. Herramientas de Seguridad

Una vez que hemos protegido el sistema operativo, debemos contar con algunas otras herramientas de seguridad adicionales, para darle mayor robustez a nuestro sistema con algunos otros mecanismos de seguridad.

2.15.1. OSECC.(Es un Código Abierto de un Sistema de Detección de Intrusos basado en Host /Open Source Host-based Intrusion Detection System)

Un sistema detector de intrusos o IDS es una herramienta que nos permite detectar accesos no autorizados a nuestros equipos de cómputo o a una red. Un IDS cuanta con sensores virtuales que le permiten obtener datos externos permitiéndole detectar comportamientos inadecuados que pudieran causar alguna anomalía en nuestros equipos, poniéndonos en alerta de un posible ataque.

Hay 3 tipos de detectores de intrusos:

a) ***Host IDS (HIDS):*** los sensores se encuentran en cada máquina y por tanto vigilan únicamente dicha máquina.

b) ***Network IDS (NIDS):*** los sensores se encuentran en segmentos de red o bien en toda la red y por tanto vigilan el tráfico de la misma detectando ataques a todo el segmento de red; donde su interfaz debe actuar de modo promiscuo permitiéndole capturando todo el tráfico de la red.

c) ***Distributed IDS (DIDS):*** en la práctica se trata de una serie de NIDS que se comunican con un sensor central, ideal para VPN's.

OSSEC es un detector de intrusos en host que proporciona análisis de logs, verificación de integridad de ficheros, monitorización del registro de Windows, detección de rootkits en tiempo real, etc. y que permite configurar respuestas activas. Funciona en multitud de sistemas operativos como Linux, OpenBSD, FreeBSD, MacOS, Solaris y Windows (Figura 2.15.1).

Figura 2.15.1. Átomo de seguridad.

2.15.2. OpenSSL

Consiste en un robusto paquete de herramientas de administración y librerías relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como OpenSSH y navegadores Web (para acceso seguro a sitios HTTPS), siendo éste de grado comercial y con la ventaja que es de código abierto. Estas herramientas ayudan al sistema a implementar la capa de sockets seguro o Secure Sockets Layer (SSL) por sus siglas en inglés, así como otros protocolos relacionados con la seguridad, como el Transport Layer Security (TLS). Este paquete de software es importante para la seguridad en nuestro servidor. OpenSSL también nos permite crear certificados digitales que podremos aplicar a nuestro servidor (Figura 2.15.2).

OpenSSL nos brinda de un entorno adecuado para encriptar los datos que mandamos a otra máquina en una red y a su vez desencriptarlos adecuadamente por el receptor evitando así el crackeado de la información. Éste puede ser usado en cualquier protocolo ftp, telnet, http, etc.

El proyecto OpenSSL es un esfuerzo de colaboración de voluntarios para desarrollar la aplicación de herramientas de código abierto el protocolo Secure Sockets Layer (SSL v2/v3) y Transport Layer Security (TLS v1). El proyecto está dirigido por una comunidad mundial de usuarios que usan Internet para comunicarse, planear y desarrollar tanto el conjunto de herramientas OpenSSL como su documentación.

OpenSSL se basa en la excelente biblioteca SSLeay desarrollada por Eric A. Young y Tim J. Hudson. El conjunto de herramientas OpenSSL está licenciado bajo una licencia de tipo Apache, que básicamente significa que usted es libre de conseguir y utilizar con fines comerciales y no comerciales sujetos a unas simples condiciones de la licencia.

Figura 2.15.2. Open SSL herramienta criptográfica.

2.15.3. ModSecurity

ModSecurity es un módulo para Apache que se encarga de brindarle in nivel de seguridad adicional a nuestro servidor Web, funcionando como una barrera entre la red y el servidor web funcionando como un firewall de aplicaciones Web permitiéndonos monitorear el trafico HTTP. Este módulo está disponible como software libre bajo la licencia GNU Licencia publica general y de igual forma la podemos encontrar bajo el uso de licencias comerciales (Figura 2.15.3).

Entre las funciones con las que cuenta este módulo, tenemos las siguientes:

- **Filtrado de Peticiones:** los pedidos HTTP entrantes son analizados por el módulo mod_security antes de pasarlos al servidor Web Apache, a su vez, éstos pedidos son comparados contra un conjunto de reglas predefinidas para realizar las acciones correspondientes. Para realizar este filtrado se pueden utilizar expresiones regulares, permitiendo que el proceso sea flexible.
- **Técnicas antievasión:** las rutas y los parámetros son normalizados antes del análisis para evitar técnicas de evasión.
  - Elimina múltiple barras (**//**)
  - Elimina directorios referenciados por si mismos (**./**)
  - Se trata de igual manera la **** y la **/** en Windows.
  - Decodificación de URL
  - Reemplazo de bytes nulos por espacios (**%00**)
- **Comprensión del protocolo HTTP:** al comprender el protocolo HTTP, ModSecurity™ puede realizar filtrados específicos y granulares.
- **Análisis Post Payload:** intercepta y analiza el contenido transmitido a través del método POST.
- **Log de Auditoría:** es posible dejar traza de auditoría para un posterior análisis forense.
- **Filtrado HTTPS:** al estar embebido como módulo, tiene acceso a los datos después de que éstos hayan sido descifrados.
- **Verificación de rango de Byte:** permite detectar y bloquear shellcodes, limitando el rango de los bytes.

Figura 2.15.3. Open SSL herramienta criptográfica.

2.16. Benchmarking

Benchmarking es un proceso sistemático y continuo para evaluar los productos, servicios y procesos de trabajo de las organizaciones que son reconocidas como representantes de las mejore prácticas, con el propósito de realizar mejoras organizacionales. *Michael J. Spendolini.*

En cómputo el benchmarking se refiere a al conjunto de procedimientos y pruebas para evaluar el rendimiento de nuestro equipo o computadora, como medir la velocidad con la que ejecuta una tarea o bien analizar los componentes de la misma como el CPU, memoria RAM, tarjeta gráfica o de video, etc. realizando diferentes combinaciones de componentes o trabajo realizado en un momento dado y analizar los resultados con equipos similares.

2.16.1. Tipos de Benchmarking

Dependiendo de nuestras necesidades o retos a cumplir sobre la calidad de los servicios que decíamos brindar podemos elegir entre tres tipos de benchmarking, los cuales son los siguientes:

- **Interno.-** Éste tipo de benchmarking es el que empieza por la casa, cuando se buscan las mejores prácticas dentro de sus límites de la propia organización para así transferirlas a otras partes de la organización y con ello realizar una solución en conjunto de los problemas.

- **Competitivo.-** Éste se identifica información de manera muy específica referente a los productos, procesos y servicios de sus competidores comparándolos con los de su organización, aunque debemos tomar en cuenta que puede ser muy difícil recaudar información sobre las operaciones de los competidores e incluso podrías llegar a no conseguir dicha información.

- **Funcional (genérico).-** Consiste en identificar las mejores prácticas de cualquier organización que cuente con una reputación de excelencia en el área especificada que se encuentre sometida a benchmarking y se refiere a ella como

funcional por que comprende actividades comerciales específicas de un área funcional determinada como la manufactura, ingeniería, recursos humanos.

También lo podemos encontrar como genérico por que realmente se enfoca en los presos de excelencia del trabajo que en las prácticas comerciales de una organización en particular.

2.16.2. Cinco etapas del proceso de Benchmarking

Para tener éxito al aplicar benchmarking cuenta con cinco etapas que nos ayudaran a facilitar dicho proceso, éstas son las siguientes:

Etapa 1. Determinar a qué se le va a hacer benchmarking.

El objetivo principal de ésta primera etapa es identificar los objetivos a quienes le va aplicar el benchmarking así como también la captura de información del benchmarking, teniendo como base algún tipo de necesidad crítica, no analice sólo por analizar.

Etapa 2. Formar un equipo de benchmarking.

En ésta etapa de trata de definir quiénes son los involucrados en el proceso de benchmarking y ver el benchmarking como trabajo en equipo, fijando unos objetivos comunes, para lograr la meta de obtener las mejores prácticas para nuestros procesos; por otra parte, también se identificara quienes son los expertos internos y externos y una vez identificados éstos se tratara de definir funciones y responsabilidades del equipo de benchmarking, dar la capacitación pertinentes a los miembros del equipo y calendarizar el trabajo para obtener resultados a tiempo.

Etapa 3. Identificar los socios del benchmarking.

Identificar los aliados que formaran parte de nuestra asociación que nos ayudaran a obtener los mejores resultados del benchmarking gracias a su cooperación para establecer una red de información propia buscando las mejores prácticas con fuentes de información internas o externas.

Etapa 4. Recopilar y analizar información de benchmarking.

En ésta fase del proceso de benchmarking se trata de recopilar toda al a información obtenida y analizarla en base a nuestros propios procesos internos, con el objetivo de conocernos mejor y ser objetivos en cuanto a nuestros resultados.

Etapa 5. Actuar.

Crear el informe del benchmarking y de nuestro análisis de las posibles mejoras de los productos, servicios y procesos. Independientemente de que el proceso de benchmarking es un proceso de investigación, hay que actuar para proponer mejor sin tener miedo al cambio.

2.17. Elección de herramientas para pruebas de Benchmarking

En nuestro caso haremos uso de herramientas de benchmarking libres para sistemas GNU/Linux, ya que nuestro servidor al cual se le aplico la reingeniería y el servidor actual se encuentran basados bajo éste tipo de distribuciones. Con la finalidad de medir diferentes aspectos de rendimiento de nuestros servidores haremos uso de dos herramientas de propósito general las cuales son Unixbench, Lmbench y una específica para medir el rendimiento de las redes Iperf.

2.17.1. Unixbench

Unixbench es una suite de benchmark para equipos tipo UNIX actualizada y revisada en los últimos años, por lo cual es muy utilizado para realizar benchmark en equipos tipo UNIX. Éste inicio por primera vez en 1983 en la Universidad de Monash como una simple aplicación de comparativa de síntesis, después fue tomado y ampliado por la revista Byte (Figura 2.17.1).

David C. Niemi mantuvo por mucho tiempo el programa, incluso realizo modificaciones y actualizaciones produciendo Unixbench 4, y más tarde Ian Smith realizo cambios muy importantes a la versión 4 y a la 5.

El propósito principal de Unixbenchmark es medir el rendimiento global, proporcionándonos un indicador de la actuación de un sistema operativo tipo Unix, por medio de múltiples pruebas de rendimiento del sistema, ejercitando el rendimiento de E/S de archivos y multitarea del núcleo. Estas son las pruebas con las que cuenta:

- **Dhrystone.-** Ésta referencia se utiliza para medir el rendimiento de los equipos y se centra en el manejo de cadenas, está fuertemente influido por el hardware y el software de diseño, la optimización de código, el compilador, la memoria cache, los estados de espera y los tipos de datos enteros.
- **Whetstone.-** Mide la velocidad y eficiencia de las operaciones punto flotante.
- **Rendimiento Execl.-** Mide la cantidad de llamadas execl que se pueden hacer por segundo. Execl es de la familia de las funciones Exec que sustituye el proceso actual por un nuevo proceso.
- **Copia de archivos.-** Mide la velocidad a la cual los datos pueden ser transferidos de un archivo a otro con diferencia de tamaño de buffer. El archivo de lectura, escritura y copia de las pruebas de captura y el número de caracteres que puede escribir, leer y copiar en un tiempo de terminado (por defecto son 10 seg.).
- **Tubo de rendimiento.-** Esta se encarga de medir el número de veces por segundo que un proceso es capaz de escribir 512 bytes a una línea de tubería nueva, pues una tubería es la forma más simple de comunicación entre procesos.
- **Tubería basada en cambio de contexto.-** Mide la cantidad de veces que dos procesos pueden intercambiar un número entero cada vez mayor a través de un tubo, ésta prueba es más una aplicación del mundo real. El programa se encarga de crear un proceso hijo con el cual de lleva a cabo una conversación en tubo del tipo bidireccional.
- **Creación de procesos.-** Mide el número de veces que un proceso puede bifurcarse o dividirse en varias partes y recoger un hijo que inmediatamente sale. Se refiere más que nada a la creación de bloques de control de procesos y la reasignación de memoria para nuevos procesos y se utiliza para comparar implementaciones del sistema operativo y llamadas a creación de procesos.
- **System Call-Overhead.-** Se encarga de medir el costo de entrada y salida del núcleo del sistema operativo, es decir los gastos generales para realizar una llamada al sistema.
- **Shell Script.-** Ésta prueba mide el número de veces por minuto en las que un proceso puede iniciar y obtener un conjunto de copias de uno, dos cuatro y ocho Shell Scripts concurrentes.
- **Pruebas gráfica.-** Su propósito es proporcionar una idea aproximada del rendimiento gráfico del sistema operativo.

Figura 2.17.1. Suite Unixbench.

2.17.2. Lmbench.

Es una herramienta de software libre GPL (Licencia Publica General), Lmbench es una herramienta que compara el rendimiento de diferentes sistemas operativos tipo Unix. Empezó como una herramienta para identificar y valorar los cuellos de botella que ocurrían de manera concurrente en los equipos y desde entonces esta herramienta ha sido utilizada por los diseñadores de equipos (Figura 2.17.2).

El principal objetivo de esta herramienta es comprobar factores como el rendimiento de la comunicación entre los procesos de las llamadas al sistema del sistema de archivos de E/S y del funcionamiento de la red.

Las pruebas que realiza son las siguientes:

- Ancho de banda de los puntos de referencia
  - En caché de lectura de archivos.
  - Copia de la memoria (bcopy).
  - Lectura de memoria.
  - Escritura de memoria.
  - Tubería.
  - TCP.
- Latencia de los puntos de referencia
  - El cambio de contexto.
  - Redes: establecimiento de la conexión, pipe, pipe TCP, UDP y RPC caliente
  - Crea y borra sistemas de archivos.
  - Creación de Procesos.
  - Manejo de señales.
  - Sistema de llamadas generales.
  - Memoria latencia de lectura.
- Miscellanious Varios
  - Procesador de cálculo de la tasa de reloj.

Figura 2.17.2. Análisis de Performance LMBench.

2.17.3. Iperf

Es una herramienta del tipo cliente/servidor escrita en C++ y es apoyado por el Laboratorio Nacional de Red de Investigaciones Aplicadas de EUA (Figura 2.17.3).

Iperf nos permite medir ambos extremos de la conexión, como el ancho de banda del protocolo de internet IP y nos proporciona información relacionada con conexiones tanto TCP y UDP como la tasa de transferencia del datagrama de red, el retardo y la pérdida de paquetes.

Actualmente se encuentra bastante actualizada y es un software de código abierto y multiplataforma pues funciona ya sea Linux, Unix y Windows.

La salida general de Iperf contiene un informe con fecha y hora de la cantidad de datos transferidos y el rendimiento del medio.

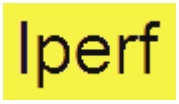


Figura 2.17.3. Iperf medición de conexiones.

CAPÍTULO 3

REINGENIERÍA

3.1. Concepto

REINGENIERÍA es la revisión fundamental y el **rediseño radical** de los **procesos**, con el fin de provocar mejoras ***espectaculares*** en los rendimientos y resultados (Michael Hammer ,Steven A. Stanton, Revolución de la Reingeniería).

Dicho de otra manera, la reingeniería constituye una recreación y reconfiguración de las actividades y procesos de una empresa, lo cual implica volver a crear y configurar de una manera radical el o los sistemas de tal manera que se logren mejoras dramáticas y significativas, dentro de medidas críticas y actuales de desempeño, en un corto período de tiempo en cuanto a la rentabilidad, productividad, tiempo de respuesta y calidad. Reingeniería es comenzar de cero, es un cambio de todo o nada, en base a la satisfacción del cliente.

Algo resaltable de la definición oficial, es que encontramos algunas palabras clave que definen más a detalle la reingeniería. Empecemos por ***<<mejoras espectaculares>>;*** la reingeniería no alcanza mejoras marginales, pues en una empresa ni en cualquier negocio que maneje procesos, dicha mejora no es gradual ni mínima, sino que trata de alcanzar grandes saltos cuantitativos en los resultados y rendimiento en medida de los costos, rapidez eficiencia y más.

La siguiente palabra es ***<< radical >>***, que significa llegar hasta la raíz de las cosas. La reingeniería no se refiere a trabajar mejorando lo que ya existe, si no es borrón y cuenta nueva, comenzar con una página en blanco y reinventar la forma en cómo se realizaran todos y cada uno de los procesos en un trabajo, es comenzar desde cero, abandonar los viejos pasos, en la búsqueda de otros nuevos rompiendo la estructura y cultura del trabajo anterior.

La tercera palabra es ***<< procesos >>***, es un conjunto de tareas interrelacionadas, que en conjunto crean y dan valor a los clientes de tal manera que ese conjunto de tareas llegan a un fin común.

Por último la cuarta palabra clave es ***<<rediseñar >>***; la reingeniería trata sobre el diseño de cómo se debe realizar el trabajo, se basa en el diseño de los procesos teniendo como objetivo principal, que dicho trabajo se realice de manera eficiente. El punto de partida para el éxito de una organización, radica en tener procesos bien diseñados.

3.2. Historia de la reingeniería.

Michael Hammer es el creador del término y concepto de reingeniería a finales de los 80's y es por ello que es considerado el padre de la reingeniería, definiéndola como el cambio fundamental para llegar a la base de los problemas de la organización; un cambio drástico que debe ocurrir para poder obtener los resultados espectaculares que la reingeniería promueve, por medio del estudio de los nuevos procesos productivos que harán a una organización mucho más productiva.

En el año de 1993 Michael Hammer escribió junto con James A. Champy el libro de Reingeniería de la Corporación: A Manifest for Business Revolution, el cual fue fundamental para captar la atención de la comunidad empresarial pues propuso uno de los cambios radicales dentro de ellas, de modo que, echaron a la basura la idea de que *"si algo no está descompuesto no hay que arreglarlo"*, el mencionaba que, porqué no hacer que funcione de lo mejor y que se explotaran totalmente sus funcionalidades, y preguntándose el cómo podíamos estancarnos en algo que simplemente es funcional, porque no mejorarlo, algo muy importante que él decía

sobre la reingeniería, es que ésta es "radical y es como comenzar una hoja en blanco" donde nosotros la iremos llenando de nuevas ideas.

3.2.1. Participantes en la reingeniería

Una parte muy importante y fundamental para que se logre la reingeniería, es el equipo de trabajo, el cual se debe encargar de convencer y convencerse, de hacer un cambio radical sin que el medio natural a ello impida la realización de dichos cambios aportando mejoras sustanciales a nuestra organización.

Los participantes que actúan en la Reingeniería son:

- **Líder**

  Empecemos por mencionar que un líder no es el jefe de un grupo de trabajo , sino más bien es aquella persona, que se encarga de guiar a su equipo de trabajo a una meta en común, mostrándoles por qué camino hay que ir; de tal forma que el líder dentro de la reingeniería, es quien nos va a motivar al cambio y quien se va a encargar de que éstos se acepten, de tal manera que todos los miembros del equipo hagan propios éstos cambios y que con seguridad se logren las metas propuestas. El líder debe ser una persona con visión a futuro, creando los nuevos procesos así como también es el encargado de ir designando los nuevos procesos, dándoles un dueño a cada uno de estos procesos siendo responsables de los mismos de principio a fin.

- **Dueño del Proceso**

  Éste es el responsable de principio a fin de un proceso en específico; por lo cual debe conocerlo perfectamente, permitiendo que el proceso se integre a su equipo de trabajo de tal forma que haya un compromiso entre las dos partes y se logre con éxito dicho proceso.

- **Equipo de Reingeniería**

  Grupo de individuos enfocados a la reingeniería de un proceso en particular, realizando un diagnóstico del proceso existente, lo rediseñan e implementan. En el equipo de reingeniería, se debe contar con personas "internas" que trabajan con el proceso actual, para que aporten al equipo su conocimiento, experiencia y credibilidad de dicho

proceso, así como también de personal "externo", que desconocerá por completo el proceso actual, lo cual les permitirá hacer diferentes aportaciones, plasmando su creatividad con una nueva visión fresca y objetiva.

3.3. El porqué de hacer reingeniería y que implica

En cualquier empresa u organización se requiere de reingeniería cuando:

- El rendimiento de la organización se encuentra por debajo de la competencia.
- Si la organización se encuentra en crisis, como la pérdida del mercado.
- Si las condiciones del mercado han cambiado, de tal forma que se tiene que visualizar una actualización, como es el constante cambio de la tecnología.
- Si se quiere ser líder de mercado.
- En caso de que nuestra área sea muy competitiva y se requiera de una competencia mucho más agresiva.
- Si ya se es líder, por consiguiente debemos no sólo de ser constantes, sino seguir mejorando para mantener nuestra posición.

Por los anteriores motivos, es que los altos ejecutivos toman la reingeniería como su forma de trabajo, para lograr sus metas de una manera estratégica, ya que la competencia, la rentabilidad y la participación del mercado existente, son los principales motivos del cambio en sus empresas o instituciones de trabajo, pues dado que requieren de resultados rápidos, puesto que éstos implica perdidas en dinero, así como la pérdida del propio mercado, evitando su expansión.

Por todo lo anterior es que hoy en día, muchos se suman a la reingeniería y por lo tanto es impulsada por tres factores muy importantes, los cuales fueron denominados como las tres C :

- CLIENTES.- Pues éstos debido al amplio mercado con el que cuentan, se vuelven mucho más exigentes, en cuanto a lo que se les ofrece y por lo mismo conocen mucho más lo que les interesa.
- COMPETENCIA.- Ésta ha crecido de una manera enorme, pues con el paso del tiempo y el crecimiento de las telecomunicaciones, hoy no sólo vemos carteles o anuncios en el

periódico, sino que ya también contamos con internet, creando nuevas formas de competencia, ofreciendo sus mejores productos, haciendo uso de la mejor mercadotecnia.

- CAMBIO.- Como todo estamos en un mundo cambiante, en continua evolución para mejorar nuestra calidad de vida, satisfaciendo nuestras necesidades de una manera sencilla, rápida y no por ello significa que por ser de una manera rápida tiene que carecer de calidad.

Ventajas de la reingeniería

Dentro de las ventajas que tiene la reingeniería tenemos las siguientes:

- Tiene un rápido reposicionamiento de la empresa en el mercado.
- Mejoras rápidas en la calidad de servicios, y tiempos de reacción.
- Reducción de costos.
- Mejoras en los niveles de satisfacción y tiempos de ciclos.
- Les permiten a las instituciones o negocios, ver incrementos en la rentabilidad.
- Mayor participación en el mercado.
- Obtiene mayores ingresos y rendimiento sobre la inversión que se haya hecho.
- Mejorar su posición dentro de la competencia posicionándose dentro de las líderes del mercado.

Principalmente lo que busca la reingeniería es tener avances de lo más decisivos, que impacten en el rendimiento de nuestras empresas o instituciones, en lugar de pequeñas mejoras incrementales, siempre y cuando éstos grandes avances se encuentren acompañados de calidad, reducción de costos, flexibilidad, rapidez, precisión y que satisfagan las necesidades de los clientes lo mejor posible.

3.4. Metodología de Reingeniería

Una metodología es una manera sistemática, o bien, es una forma de itinerario perfectamente definido que nos ayuda a llegar a donde queremos ir. La metodología que se usa para aplicar la Reingeniería, en diferentes organizaciones de trabajo es la metodología "***Rápida Re"*** o mejor conocida como ***"Rápida Reingeniería"***, la cual está diseñada para que se utilice sin tener que basarse de expertos fuera de nuestras organizaciones y a su vez ayuden a los equipos de reingeniería a encontrar el o los procesos necesarios para lograr ese cambio

radical necesario, para lograr el éxito frente a todas las demás competencias de su mercado de trabajo, sin la necesidad de valerse de muchos expertos externos.

Esta metodología se divide en 5 etapas las cuales son las siguientes:

a) Etapa 1 – Preparación

Definir las metas y los objetivos estratégicos que justifiquen la reingeniería, los vínculos entre los resultados de la reingeniería y los resultados de la organización hasta el momento. Realmente su propósito principal de ésta etapa es movilizar, organizar y estimular a las personas que realizaran la reingeniería, en ésta etapa es donde se produce el mandato de cambio, construyendo una nueva estructura organizacional.

En ésta etapa, es que la organización se encarga de declarar sus metas, para satisfacer mejor al cliente, dándonos la facilidad de recabar toda la información y que ésta sea lo más clara posible. Otra parte importante y que muy pocas veces se realiza, es la capacitación que se le debe brindar a nuestro equipo de trabajo en cuanto a la metodología a seguir, pues si no se les informa de la nueva forma de trabajo a todo aquel que esté involucrado, para que funcione mejor nuestras estrategias de trabajo, de tal forma que sepan de lo que estamos hablando; además que deben estar lo suficientemente motivados para incitarlos a entender la oportunidad de cambios decisivos.

Si bien en ésta etapa se crea el plan de cambio, es relevante realizar un análisis completo de cómo está funcionando nuestra organización, de tal manera que encontremos los puntos fuertes así como los débiles de la organización; pero no basta con sólo analizar los procesos internos, sino también todo el ambiente que nos rodea en el negocio y que además pueden afectar a la organización, o bien pueden estar a favor pudiendo aprovecharlas.

b) Etapa 2 – Identificación

El propósito de ésta etapa es el desarrollo de un modelo orientado al cliente, identificando procesos específicos que agregan valor, por lo cual mencionaremos los servicios con los que cuenta hasta el momento, así como los procesos que se llevan a cabo. Uno de los objetivos muy importantes de ésta etapa es lograr comprender de manera total al cliente, con su relación con la organización y sus expectativas, viendo los flujos del trabajo, para así contabilizar los costos de las actividades realizadas con los recursos disponibles. Por otra parte también en ésta etapa

es donde llevamos todo ese análisis, para así darles prioridades y mayor peso a los diferentes procesos.

c) Etapa 3 - Visión

El propósito de ésta etapa es desarrollar una visión del proceso capaz de producir un avance decisivo en rendimiento. La visión de los nuevos procesos deben describir claramente las características primarias así como también deben ser claras y fáciles de comprender para todos los miembros de la organización.

Dentro de ésta etapa se logra la identificación de todos los elementos participantes del proceso, los problemas, cuestiones actuales así como también las medidas comparativas en cuanto al rendimiento de los procesos actuales, también se definen los cambios que se requieren, ofreciendo una nueva visión de los nuevos procesos. La gestión del cambio, la administración del proyecto y la facilitación, son técnicas realizadas de manera continua en ésta etapa.

d) Etapa 4 – Solución

En ésta etapa se produce un diseño técnico y un diseño cultural-organizacional del trabajo o bien diseño social.

La etapa de diseño técnico busca realizar la visión (etapa 3), especificando las dimensiones técnicas de los nuevos procesos, permitiéndonos describir la tecnología, en particular de donde y cuando se aplicará, así como la capacitación para implementar las actividades, los procesos, las normas, los procedimientos, los sistemas y los controles utilizados, así como también producirá los diseños para la interacción entre los elementos sociales y técnicos.

El objetivo principal del diseño social es especificar las dimensiones sociales dentro de los nuevos procesos, permitiéndonos describir la organización con su personal reestructurándolo, así como definir nuevas responsabilidades en cuanto a la toma de decisiones, tomando en cuenta su capacitación, reorganización así como su reubicación de las nuevas posiciones; las cuales dependerán de las habilidades del personal, para reubicarlos en los nuevos equipos de proceso.

e) Etapa 5 – Transformación

El propósito de ésta etapa es realizar la visión del proceso, implementando el diseño de la etapa 4. En ésta etapa es donde se realiza todo tipo de pruebas en nuestros nuevos procesos que llevamos a cabo, generando versiones piloto, las cuales nos permiten mejor el funcionamiento para así liberar la versión de producción totalmente completa de los procesos rediseñados y los mecanismos de cambio continuo.

3.5. Diferencias entre la reingeniería y la mejora continua

Cabe mencionar que dentro de la reingeniería una de las cosas en las que hace hincapié es, la idea de reinventarse, lo cual no quiere decir que cambiemos lo que ya existe, sino que aún no existe. Por otra parte la reingeniería propone un cambio radical en los procesos que llevamos a cabo, haciendo todo un análisis de los mismos procesos y reinventarlos, obteniendo los resultados de los mismos, en un corto período de tiempo. A diferencia de la mejora continua, la cual trabaja sobre los procesos ya existentes, haciendo sólo cambios pequeños incrementales poco a poco a largo plazo. Entre las diferencias que se marcan entre la reingeniería y la mejora continua, tenemos las siguientes (Tabla 3.5.1):

Tabla 3.5.1. Reingeniería vs. Mejora Continua.

REINGENIERÍA	MEJORA CONTINUA
Verifica los procesos deficientes y obsoletos, para crear nuevos y eliminar viejos, haciendo todo un análisis de los mismos, para que bien éstos cumplan, excedan sus exigencias y éstos siempre sean líderes en el enorme mercado existente, teniendo ellos la ventaja ante sus competidores.	Los procesos existentes, se encuentran próximos a las exigencias de los clientes, incluso pueden ser obsoletos y aun así se usan.
Se cuestiona y analiza la base de los procesos, agregando más procesos o bien eliminándolos.	Acepta los procesos existentes y parte de ellos, para hacer las mejoras incrementales.
La tecnología es el motor de las transformaciones para una organización	Utiliza la tecnología dándole un enfoque incremental.
Los riesgos son mayores y por lo tanto el impacto que se da en una organización es	El impacto que tiene ésta sobre una organización es muy pequeño y poco riesgoso,

enorme, porque viene a cambiar toda una forma de pensar y trabajar.	pues son pequeños cambios que se van dando paulatinamente.
Esta impacta bastante los recursos económicos de las empresas, pues el costo llega a ser elevado, pero se justifican los mismos en un corto período de tiempo.	Los costos son mucho menores.

CAPÍTULO 4

VIRTUALIZACIÓN

4.1. Concepto

En el ámbito de Tecnologías de la información, la virtualización es la abstracción de los recursos con los que cuenta una computadora, la cual, es también conocida como monitor de máquina virtual o ***hipervisor*** (hypervisor), pues se encarga de crear una abstracción entre el hardware del dispositivo físico (host) y el sistema operativo de la máquina virtual, creando un medio virtual para el uso de algún dispositivo o recurso; ya sea un servidor, un dispositivo de almacenamiento, una red, un sistema operativo, donde dichos recursos pueden estar en varios entornos de ejecución. La virtualización es la separación de un recurso o solicitud de algún servicio subyacente físico y la prestación del mismo, donde dichos servicios o dispositivos virtuales dependen de los mismos dispositivos físicos o hardware, pero trabajando como modelos totalmente independientes de éste, compartiendo recursos locales físicos entre varios dispositivos virtuales.

La virtualización en las TIC's (Tecnologías de la Información), no es algo nuevo, es algo que se ha venido utilizando en los últimos años, en todos los centros de datos de las pequeñas y grandes empresas. La virtualización cuenta con muchas ventajas, a diferencia de los servidores físicos tradicionales, pues con el paso del tiempo ha ido evolucionando de tal manera, que sea más fácil tomar el paso de virtualizar nuestros servicios.

Definición de Hipervisor (Hypervisor)

Es una plataforma de virtualización compuesta por una capa de software, la cual nos permite utilizar simultáneamente, diferentes sistemas operativos en una misma computadora. El hipervisor es la parte principal de una máquina virtual, éste se encarga de manejar los recursos del sistema principal, exportándolos a la máquina virtual, presentando a cada una de estas máquinas virtuales, una interfaz de hardware que sea compatible con el sistema operativo elegido.

4.2. Arquitecturas de Virtualización

Software de virtualización

El software de virtualización se encarga de desmontar el hardware físico real de una máquina, desde el sistema operativo inferior, mediante un programa informático; permitiéndonos que varios ambientes virtuales, puedan funcionar de lado a lado en la parte superior de una máquina física, dónde gracias a éste proceso de abstracción, cada máquina virtual cuenta con su propio hardware (RAM, CPU, tarjeta de red, DD, etc.), el cual es independiente en cada máquina virtual.

Como podemos observar en la Figura 4.2.1, se muestra como un equipo normal presenta para su uso el sistema operativo y su hardware, a diferencia de la Figura 4.2.2 y 4.2.3 donde nos muestra una arquitectura de virtualización, la cual cuenta con una capa de virtualización que es dónde encontramos al hipervisor que se encarga de desacoplar el hardware, para asignarle a las diferentes máquinas virtuales la utilización del hardware físico. Un equipo que hospeda una plataforma de virtualización, puede o no tener un sistema operativo base, para instalar la plataforma de virtualización, o bien instalarla en bruto en un sistema completamente limpio, de tal forma que las dos maneras en que se instale la plataforma de virtualización, se puedan crear una o más máquinas virtuales, que actuaran de manera independiente una de la otra, con su propio hardware, S.O. y aplicaciones.

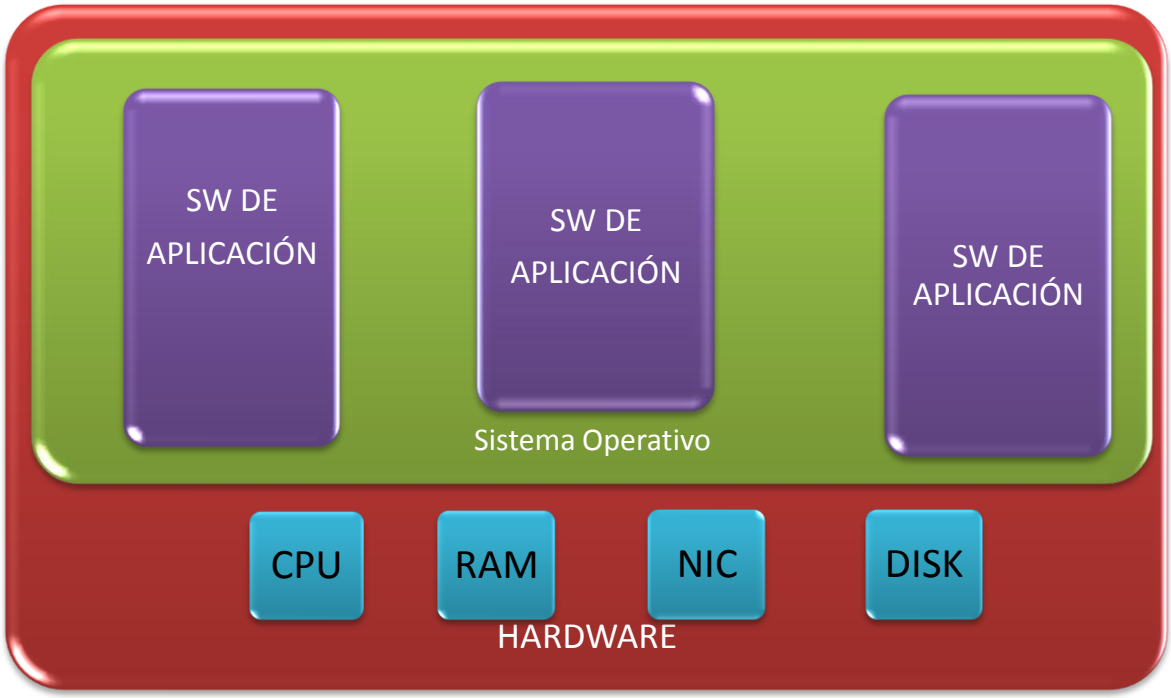


Figura 4.2.1. Máquina que presenta su hardware al sistema operativo para su uso.

Dentro de la virtualización tenemos dos tipos de arquitecturas, una que es la de tipo *Hosted* u hospedaje y la arquitectura del tipo *Hipervisor*.

4.2.1. Arquitectura de tipo Hospedaje (Hosted)

La arquitectura de tipo hospedaje es cuando el software de virtualización se encuentra instalado sobre un sistema operativo base denominado anfitrión, donde la capa de virtualización, se apoya del sistema operativo base para obtener el soporte necesario de los dispositivos, así como de la administración de los recursos de hardware, de tal forma que si el dispositivo no es admitido por el sistema operativo, las máquinas virtuales no podrán hacer uso del mismo, siendo éstos una desventaja del uso de éste tipo de arquitectura. Una de sus ventajas que encontramos es, que si el sistema operativo admite el hardware, contamos con mucho más soporte y variedad para el mismo.

El hecho de que el sistema operativo base sea un puente para acceder a los dispositivos físicos de nuestro equipo, implica que dicho S.O. base pueda sobrecargase, afectando el desempeño del mismo y por consiguiente a las máquinas virtuales.

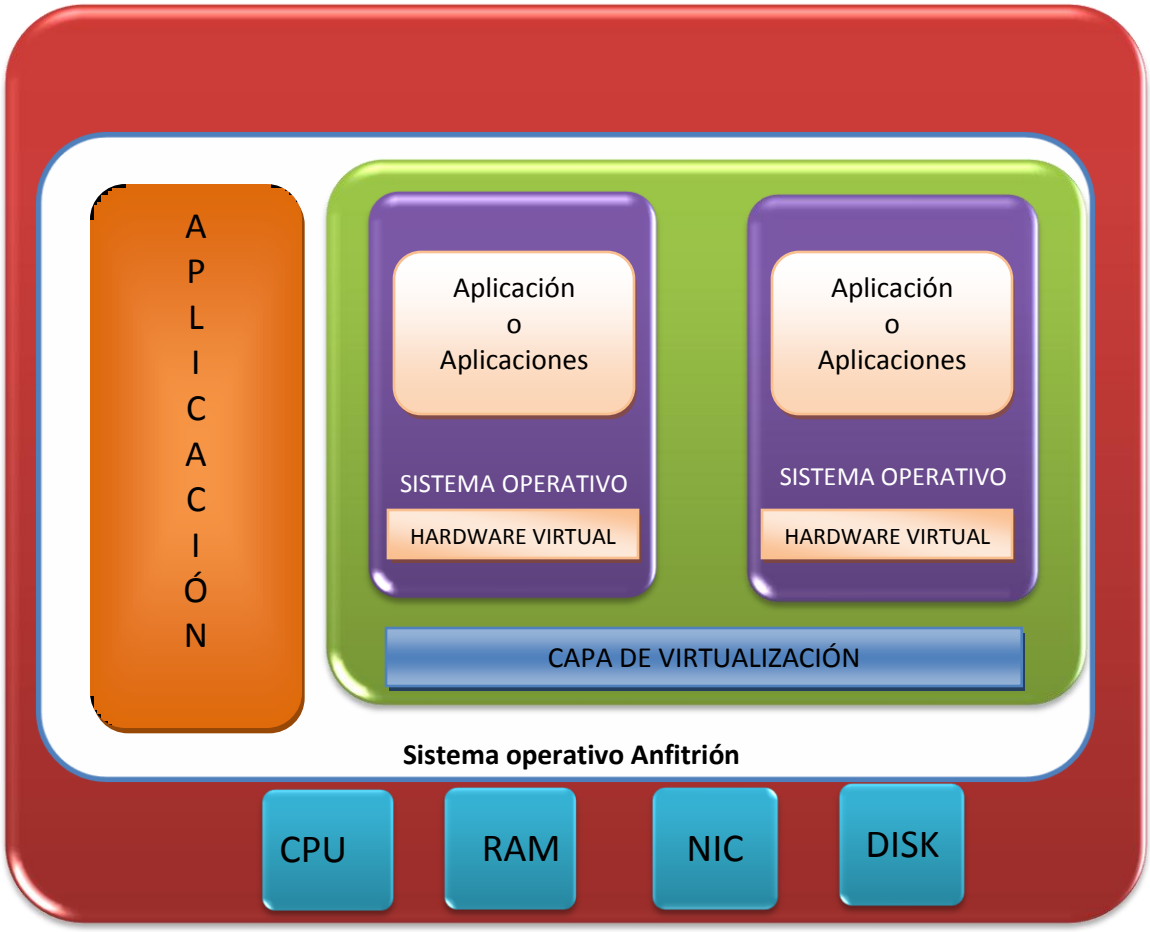


Figura 4.2.2. Arquitectura de Virtualización del tipo Hosted.

4.2.1.1. Tecnologías de virtualización de arquitectura tipo Hospedaje

Dentro del tipo de arquitectura tipo hospedaje, tenemos las siguientes tecnologías:

- **VMware Server.**

  VMware Server tiene un buen comportamiento en entornos domésticos. Las restricciones de su licencia, la computación distribuida, el alquiler y venta de servicios ejecutados en el servidor, probablemente no preocupen al usuario doméstico. Al igual que la versión Workstation, VMware Server está disponible como RPM y se puede instalar en cualquier Windows y Linux.

- **VMware Workstation.**
  La interfaz gráfica es bastante clara, y no deberíamos tener muchos problemas a la hora de encontrar funciones críticas. Aunque VMware tiene una exuberancia de posibles configuraciones, está tan bien organizado, que ningún usuario debería tener problemas para encontrarlas.
- **Microsoft Virtual PC.**
  Es una aplicación para sistemas Windows y cuenta ya con los sistemas operativos preconfigurados, entre los que encontramos a Windows 98, NT, 2000, 2003, XP, Vista y OS/2.
- **Microsoft Virtual Server 2005.**
  Microsoft Virtual Server 2005, es una plataforma de virtualización de servidor basada en la tecnología adquirida por Connectix Corporation. Virtual Server 2005 debe estar instalado en un Windows 2003 Server o un sistema Windows XP Professional (no es compatible con Microsoft Virtual Server 2005 en Windows XP Professional para su uso en producción).
- **VMware GESX.**
  VMware GSX Server, es una plataforma de virtualización de servidores ligera, originalmente su estación de trabajo se encuentra basada en productos de VMware. VMware GESX Server debe ser instalado en un sistema Linux o cualquier sistema operativo de Windows.

4.2.2. Arquitectura de tipo Hipervisor

Se dice que se emplea una arquitectura de tipo hipervisor cuando no existe un sistema operativo base, teniendo un enfoque conocido como ***"metal desnudo"***; pues el software de virtualización se instala en un sistema completamente limpio, de tal manera que el núcleo o kernel, nos proporcionará los controladores necesarios para soportar el hardware físico.

La diferencia de ésta arquitectura con la de hospedaje, es que no necesita de un S.O. puente para acceder a los recursos de HW y por lo tanto, no tiene esa sobrecarga en el procesamiento de la máquina física, disminuyendo los gastos generados en el procesamiento dentro del equipo físico, pero con la desventaja que como el software de virtualización se encarga de instalar su propio kernel, y éste nos proporciona los diferentes controladores que podemos soportar, nos reduce las opciones de los controladores de hardware compatible.

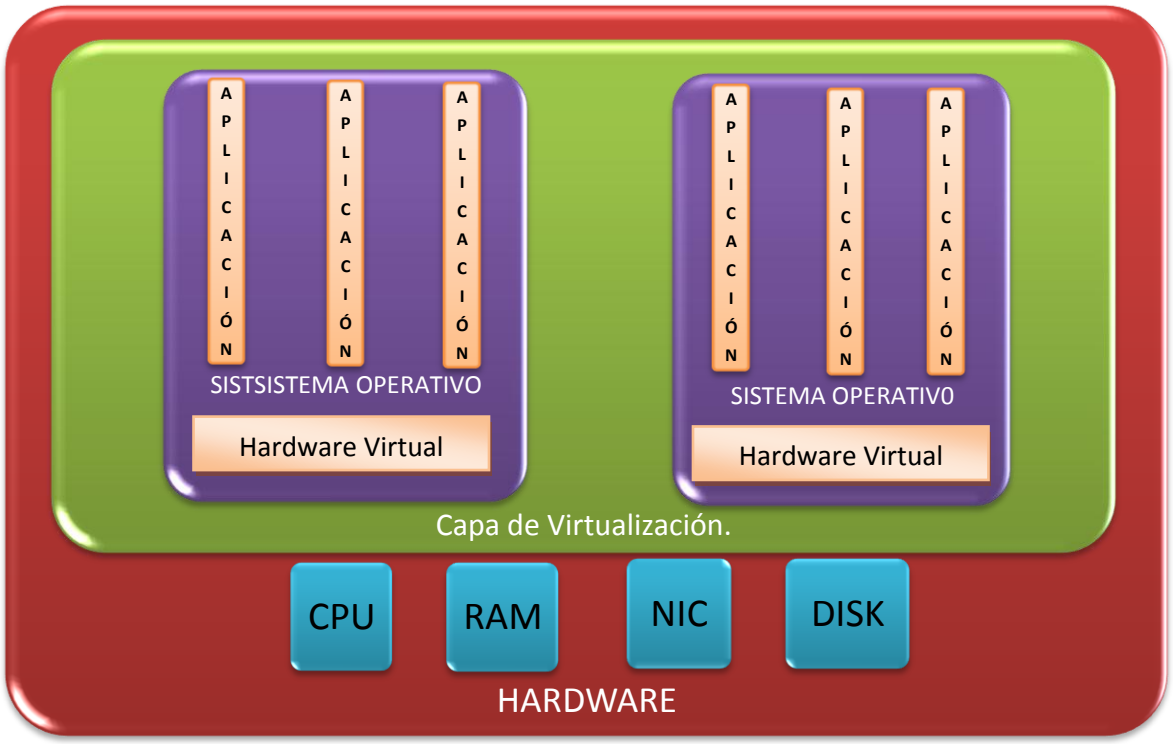


Figura 4.2.3. Arquitectura de Virtualización del tipo Hipervisor.

4.2.2.1. Tecnologías de virtualización de arquitectura tipo Hipervisor

Dentro del tipo de arquitectura tipo hipervisor, tenemos las siguientes tecnologías:

- **VMware ESX.**
  Éste es el más rápido de realizar y es la plataforma de virtualización de servidores más madura hasta la fecha. VMware ESX Server se ejecuta directamente sobre el hardware físico para maximizar la eficiencia.
- **Windows Server 2008.**
  Es la versión empresarial de Microsoft de tipo hipervisor y está incluida en Windows Server 2008 y sólo se ejecuta sobre plataformas x64.

4.3. Uso de Virtualización de servidores

Casi todos los servidores junto con su carga de trabajo, pueden ser virtualizados, pero en algunas ocasiones puede que no se desee que lo sean por alguna razón, es por eso que debe tomar en cuenta los siguientes aspectos a considerar para virtualizar un servidor.

- **Consolidación de servidores.**

La solidez y firmeza de los servidores físicos con los que contemos, permite que sean virtualizados en muy poco tiempo, siendo alojados en un mismo servidor físico, ocupando de cierta manera menos espacio de nuestro sito de servidores, además que se aprovecha mejor los recursos, pues muchos de los servidores de hoy en día son subutilizados, de tal forma que su uso está entre un 8% y 12% de su capacidad total.

- **Soporte de aplicaciones.**

Esta provee de mejores rutas de actualización, para mover todas las aplicaciones que se tengan heredadas, de tal forma que los sistemas que se encuentren operando dentro del nuevo hardware, puedan funcionar sin problema alguno, por cuestiones de incompatibilidades con las plataformas que cuente con hardware más reciente.

- **Soporte para múltiples sistemas operativos.**

Éste es de gran utilidad cuando se está trabajando en ambientes de desarrollo y puesta a prueba de sistemas operativos.

- **Software de demostración.**

Mediante el uso de la virtualización, las demostraciones de software beta pueden correr de una manera consistente.

- **Desarrollo, pruebas y depuración.**

Debido al amplio aislamiento entre el entorno y la plataforma de Virtualización, se vuelve fácil la realización de pruebas y depuración del software.

- **Formación técnica y aprendizaje.**

En la actualidad, en muchas de las clases que se imparten hoy en día, se requiere de varios equipos para cada uno de los estudiantes. Mediante el uso de la virtualización, se pueden reducir el número de equipos necesarios para enseñar una clase.

4.4. Donde no usar Virtualización

- **Pruebas de virtualización X86.**

No se pueden hacer pruebas de virtualización dentro de la virtualización, pues éstas pueden usar el doble de tiempo de procesamiento, y no podrán ser utilizables. Ésta limitación sólo se aplica a virtualización X86, pues la mayor parte de éste tipo de virtualización fue diseñada para que sólo IBM las soporte.

- **Utilización alta de recursos.**

No puede tenerse un servidor virtual que tenga un uso de recursos muy caro, ya que generalmente su porcentaje en CPU es muy alto, así como el uso de memoria, de disco y de red, y por lo tanto se debe tomar en cuenta que, si se tiene más de una máquina virtual corriendo y están compitiendo por el uso de los recursos del equipo físico, podría no funcionar de una manera óptima.

- **Juegos de computadoras.**

El rendimiento y los requisitos de desempeño de los mismos son demasiado grandes, como para ponerlos aprueba con éxito y jugar juegos de gama alta en entornos virtuales.

- **Modelos de licencia del proveedor.**

Se debe considerar, porque generalmente, las aplicaciones no cuentan con concesiones de licencias para la virtualización, pues su licencia del software se encuentra basada por el número de CPU físicos, y no con los virtuales que ejecutan la aplicación, un caso de éstos es ORACLE. Aunque hoy en día los proveedores de software ya están creando sus licencias de software basados en el número de CPU's asignados al servidor, pero aun así independientemente si son físicos o virtuales éstos siguen siendo una limitante.

- **Especialización de hardware y periféricos.**

No hay manera de personalizar el hardware de periféricos y tarjetas dentro de una plataforma de virtualización. Debido a que no hay una forma de emular éstos dispositivos en un entorno virtual.

- **Compatibilidad de aplicaciones.**

Ya que algunos de los vendedores de software no proporcionan el apoyo, si nuestras aplicaciones se encuentran ejecutándose dentro de un servidor virtual.

- **Pruebas de rendimiento.**

Pues la sobrecarga de virtualización causaría una visión errónea del funcionamiento.

- **Controladores de depuración del hardware.**

Todas las máquinas virtuales en cada versión de la virtualización, el hardware es fijo y emulado, no hay forma de probar o depurar controladores de hardware en la plataforma de virtualización.

4.5. Ventajas de la Virtualización

La virtualización cuenta con muchas ventajas, pero entre las más importantes, tenemos las siguientes:

- **Portabilidad.**

Cuenta con una gran posibilidad de tener una plataforma de hardware compatible, inclusive si el hardware real, lo produjeron diferentes fabricantes.

- **Administración.**

Los entornos virtuales pueden gestionarse de manera muy sencilla, sin que el administrador se encuentre físicamente frente a el servidor, pues pude tener una administración remota de una manera muy sencilla, le permite utilizar todos los recursos del hardware, sin ningún problema, siempre y cuando cuente con los permisos necesarios para poder hacerlo.

- **Mayor eficiencia.**

Cuando se tienen bien implementados los servidores virtuales, se le puede sacar mejor provecho al servidor físico, pues con las herramientas de virtualización que se utilizan, cuentan con una gran cantidad de mejoras para el aprovechamiento del servidor físico, sin sobrecargarlo y aprovechando todos sus recursos al máximo.

- **Alta disponibilidad.**

Actualmente, muchas de las herramientas de virtualización, ofrecen proporcionarnos alta disponibilidad en nuestros servidores, pues cuentan con potentes funcionalidades como lo son los *Snapshots* o fotografías en un instante del sistema operativo; nos sirve en caso de querer hacer algún cambio en nuestro sistema y en caso de no funcionar dichos cambios dañando él mismo, no nos impactara tanto en tiempo y costo, para volverlo a restablecer, ya que podemos regresar a un momento anterior del sistema con los snapshots.

- **Movilidad.**

Pues muchas de éstas aplicaciones de virtualización, nos permite hacer movimientos de máquinas virtuales de un lugar a otro o de un servidor a otro, claro que con sus respectivas limitaciones de hardware, sin la necesidad de que perdamos los servicios que se están ejecutando en ese momento, de tal forma que no afecta nuestra productividad.

- **Balanceo dinámico.**

Un balanceo dinámico de máquinas virtuales entre los servidores físicos que componen el pool de recursos, garantiza que cada máquina virtual se ejecute en el servidor físico más adecuado, proporcionando un consumo de recursos homogéneo y óptimo en toda la infraestructura.

- **Reducción de costo.**

Reducción de los costos de espacio, del hardware necesario, así como de sus costos asociados y consumo.

- **Aislamiento.**

En caso de un fallo general de sistema de una máquina virtual, no afecta al resto de máquinas virtuales.

- **Fácil incorporación de recursos.**

Rápida y fácil incorporación de nuevos recursos para los servidores virtualizados.

4.6. Principales proveedores de virtualización de servidores

Los principales proveedores de virtualización de servidores son tres, y son los que han tomado la gran mayoría del mercado existente, éstos son los siguientes:

- **CITRIX.**

  Es una de las empresas que nos ofrece una gran variedad de tecnologías de virtualización, el cual se ha encargado de ampliar todas sus ofertas en cuanto a los productos que nos ofrece contamos con ***XenServer*** en cuatro diferentes versiones, la **Expres Edition** la cual es una versión libre. **Standard Edition**, es una versión básica que soporta uno o dos sistemas virtuales a la vez. **Enterprise,** ésta cuenta con mayor recursos de hardware y puedes ejecutar de forma ilimitada sistemas virtuales, claro siempre y cuando se cuente con los recursos necesarios. Por último tenemos a la **Platinum Edition** que añade reparto dinámico entre los host. Citrix nos ofrece sus propios equipos que contiene alguna versión de sus propios hipervisores, que ya vienen incorporados en el hardware del servidor. Por otra parte también nos ofrece productos como XenDesktop para escritorios virtuales y XenApps para aplicaciones virtuales.

- **Microsoft.**

  Nos ofrece una serie de tecnologías de virtualización con productos adicionales. Actualmente cuenta con ***Virtual Server 2005 R2 SP1*** y ***Virtual PC 2007*** los cuales son gratis, pero son productos de software virtual. Su clase empresarial es del tipo hipervisor y forma parte del sistema operativo Windows Server 2008; sólo se ejecuta sobre plataformas x64. También nos ofrece Microsoft Application para aplicaciones virtuales y Servicios de Terminal Server para presentaciones virtuales.

- **VMWare.**

  Es considerada como la empresa que ofrece los productos más maduros en cuanto a virtualización, con una amplia y completa gama de servidores y herramientas de virtualización de escritorio. Entre lo que nos ofrece, encontramos a ***VMWare Server*** el cual es un producto de software libre de virtualización. ***VMWare Workstation y Virtual Infraestructure*** la cual es una versión muy completa de su servidor hipervisor ESX. VMWare fue el primero en crear un hipervisor que conociera el

hardware del servidor, con la versión de ESXi, el cual es gratuito. También nos ofrece *Virtual Desktop* para infraestructura de escritorios virtuales y *ThinAPP* para aplicaciones virtuales. Es por esto y mucho más, que hoy en día, **VMWare** ha ido creciendo como industria de la virtualización, por la amplia gama de productos, así como las herramientas de administración que ofrece para el manejo de los entornos virtuales.

4.7. VMWare ESX

En la actualidad se cuentan con varias versiones de VMWare ESX, siendo la última la versión 4.0, pero para nuestro estudio hablaremos de manera muy general de todas las bondades que nos ofrece ESX.

ESX es una base sólida en la capa de virtualización, pues ha demostrado que es una de las aplicaciones más maduras en infraestructuras de virtualización. Los servidores ESX incrementan la utilización del hardware, reduciendo de una manera drástica el costo de operación por intercambio de los recursos de hardware a través del número de máquinas virtuales; haciendo todo esto mediante su gestión de recursos avanzados, la alta disponibilidad y sus características de seguridad. Permitiéndonos crear ambientes virtualizados en un corto periodo de tiempo, proporcionándonos más rápido el retorno de la inversión. ESX cuenta con una versión gratuita la cuál es "ESXi", siendo una versión más ligera de 32 MB y por lo tanto es mucho más limitada que la ESX.

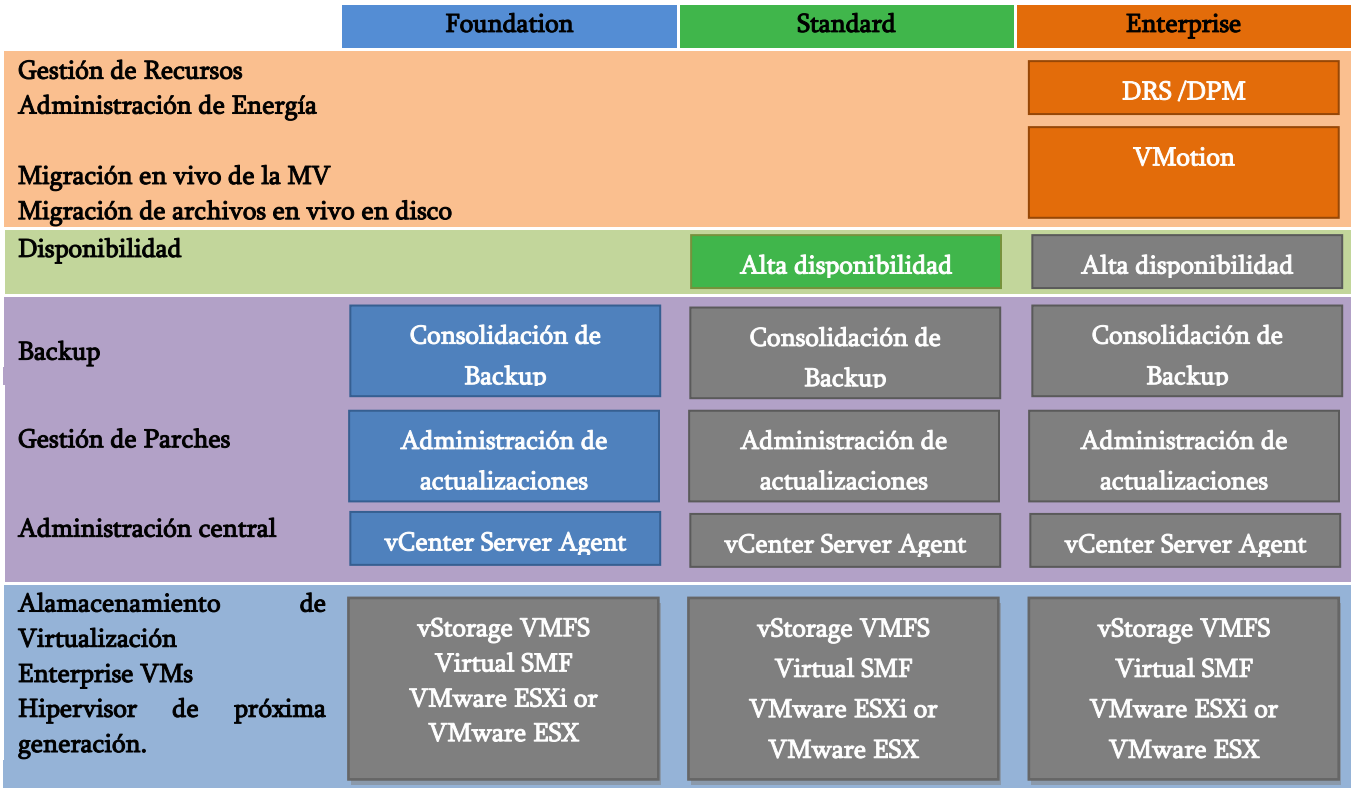


Figura 4.7.1. Ediciones de ESX.

Actualmente las bondades de ESX se encuentran en tres ediciones, para satisfacer según las necesidades y presupuestos del mercado, las cuales son la Fundation, Estándar y Enterprise (Figura 4.7.1.).

4.7.1. Núcleo VMkernel

El propósito principal de un servidor ESX, es proporcionar a las máquinas virtuales (VM) invitadas el acceso a los recursos físicos necesarios para su correcto funcionamiento de sus sistemas operativos de las mismas. Ésta capacidad de proporcionar y dotar de recursos de hardware a más de una VM, es gracias a su núcleo VMkernel. La diferencia de los núcleos de hoy en día, como los de Microsoft, Sun y Linux por nombrar a algunos, están diseñados para proporcionar el acceso al hardware a un solo sistema operativo; en cambio el VMkernel se encuentra diseñado de tal forma que pudiera programar el tiempo de CPU, el disco de E/S, acceso a memoria y la creación de redes de manera simultánea, para más de una máquina virtual, las cuales se encuentren ejecutando diferentes sistemas operativos.

ESX tiene una arquitectura de tipo hipervisor, pues es una aplicación que se instala en el metal desnudo de la máquina, de tal forma que no requiere de un sistema operativo base para su

funcionamiento. Con esto le brinda a las máquinas virtuales el acceso directo con el VMkernel para acceder al hardware, y tener un rendimiento mucho mejor, que si tuviera que interactuar con el software de virtualización, que a su vez, se encuentra instalado en el sistema operativo base.

Algo muy importante que debemos tomar en cuenta para el correcto funcionamiento del VMkernel, es que los fabricantes de servidores que deseen que se ejecute un servidor ESX sobre su hardware, deben encontrarse certificados por VMware, para que sus dispositivos sean compatibles. Pues el VMkernel viene precargado con los drivers de los dispositivos y asume que se ejecuta en un hardware soportado.

El VMkernel virtualiza una instancia de ejecución de Red-Hat Enterprise 3 durante el proceso de arranque. Dicha máquina virtual se denomina como la consola de servicio, siendo una consola del sistema operativo que se crea en la instalación del servidor ESX, utilizándola para la administración de las máquinas virtuales.

4.7.2. Principales características de ESX

Entre las principales características de ESX tenemos las siguientes:

- **VMWare VMFS (Sistema de archivos de la Máquina Virtual / VirtualMachine File Syetm).**
  Cuenta con un sistema de archivos de tipo clúster que le permiten que múltiples instalaciones de servidores ESX, tengan acceso al mismo almacenamiento de la máquina virtual al mismo tiempo. Permitiéndonos ofrecer servicios de infraestructura distribuida, gracias a los servicios de VirtualCenter, VMotion, DRS (Programador de recursos distribuidos/ Distributed Resource Scheduler) y HA (Alta Disponibilidad / High Availability).
- **La tecnología VMware Virtual SMP (Multi-Proceso Simétrico / Symmetric Multi-Processing).**
  Se encarga de mejora el rendimiento de la máquina virtual, al permitir que una sola máquina virtual haga uso físico de múltiples procesadores al mismo tiempo. Su principal característica es que permite la virtualización de la mayoría de los procesadores y aplicaciones empresariales con gran cantidad de recursos, como lo son las bases de datos.

- **VMware Virtual Center.**

  Se encarga de ofrecer una administración centralizada y automatizada, pues administrar los recursos disponibles y da una mayor disponibilidad de los mismos. Brinda comodidad, eficiencia y fiabilidad; pues VirtualCenter cuenta con un conjunto de interfaces Web, de tal forma que permiten la integración de otros productos de administración. El principal uso de VirtualCenter es brindarnos mayor comodidad en cuanto a la administración de las máquinas virtuales, pues es un punto central de control para la administración, sin la necesidad de estar interactuando con el servidor físico, trabajando en él desde cualquier lugar en donde nos encontremos, pues podemos monitorear, administrar y migrar máquinas virtuales.

- **VMware DRS (Programador de recursos Distribuidos / Distributed Resource Scheduler).**

  VMware DRS se encarga de alinear los recursos disponibles, predefiniendo las prioridades del negocio, racionalizando el trabajo y las actividades intensivas de los recursos. Por otra parte, también incluye ahora DPM (Administración y Distribución de Energía / Distributed Power Management), el cual se encarga de equilibrar la carga de trabajo entre las máquinas virtuales, con el propósito de reducir el consumo de energía en el centro de datos.

- **VMware VMotion.**

  VMWare VMotion es una tecnología que nos permite la migración en vivo de las máquinas virtuales, de un servidor físico a otro, para no interrumpir en mantenimiento de nuestros entornos de TIC´s. Esto se hace siempre y cuando la arquitectura de éstos sea la misma y se cuente con un arreglo de discos en iSCSI SAN o NAS.

- **VMware Storage VMotion.**

  VMWare Storage VMotion permite la migrar una máquina virtual en vivo de discos, para así compartir un lugar de almacenamiento, sin interrupción del funcionamiento de la máquina o tiempo de inactividad para los usuarios de la aplicación.

- **VMware HA (Alta Disponibilidad / High Availability).**

  VMware HA nos permite y nos da una tranquilidad de ofrecernos una alta disponibilidad de nuestras aplicaciones, independientemente del hardware y del sistema operativo.

- **VMware Administrador de Actualizaciones (Update Manager).**
  Éste se encarga de administrar los parches y actualizaciones correspondientes de nuestro servidor ESX, para su mejor funcionamiento y asegurar de la mejor manera nuestra infraestructura TI.
- **VMware Consolidación de Backup.**
  Proporciona una manera de instalación centralizada de una copia de seguridad de una máquina virtual. Permite el acceso a contenidos de una copia de seguridad, que se puede encontrar en una central de un servidor proxy Microsoft Windows 2003, en lugar de realizarlo directamente en el servidor ESX.

CAPÍTULO 5

CASO PRÁCTICO DEL PRINCIPAL PORTAL WEB DE LA FACULTAD DE INGENIERÍA

5.1. Identificación de componentes esenciales.

Una vez ya definida la problemática actual la cual es mejorar el servicio Web que ofrece la Unidad de Servicios de Cómputo Académico, en cuestiones de hospedaje, rendimiento, automatización de tareas, seguridad, estadísticas del sistema, rendimiento y disponibilidad del mismo, empezaremos ahora con las nuevas herramientas de infraestructura con las que contamos. Primero que nada se nos dio un espacio en el servidor de virtualización que se encuentra en la Unidad se Servicios de Cómputo académico de la Facultad de Ingeniería, dado nuestros requerimientos y necesidades de espacio como se muestra en la Tabla 5.1.1.

Tabla 5.1.1. Características del servidor virtual.

Características del Servidor Virtual	
Sistema Invitado	Linux de 64-bits
Número de Procesadores	2 Procesadores
Memoria Estándar	2GB de Memoria
Memoría de Overhead	212 MB
Unidad de Disco Duro Interno	70GB en Disco
CD-ROM/DVD	Unidad de CD-ROM

Ya con el servidor virtual, también haremos uso de otro servidor físico que actuará como servidor ***'mirror'*** del servidor virtual. Las características mínimas para este servidor mirror serán muy similares a la tabla anterior.

5.2. Metodología

5.2.1. Etapa 1 Preparación

Definir las metas y los objetivos estratégicos que justifiquen la reingeniería.

Como ya habíamos mencionado anteriormente el portal principal de la Facultad de Ingeniería se encuentra alojado en uno de los servidores de servicios de cómputo académico (UNICA) y por tal motivo es de gran importancia que el servicio Web se encuentre siempre disponible, pues este portal, es la entrada a todas las divisiones, asociaciones y en pocas palabras a todo aquel que se encuentre relacionado de alguna manera a nuestra Facultad de manera Institucional; así como siempre innovar para ofrecer el mejor servicio de hosting, en el que puedan estar manejando no sólo texto plano en sus sitios , si no también se creen contenidos dinámicos.

Nuestra principal meta es lograr un cambio total y significativo del servicio Web en cuanto a los procesos que se realizan en éste, dándonos así una renovación tecnológica en todos los aspectos, ya que día con día ésta va cambiando y no nos podemos quedar atrás o con la idea de que un cambio nunca es bueno si algo está funcionando; no debemos conformarnos, sino estar a la vanguardia ya que con éste cambio de mentalidad podremos lograr nuestros principales objetivos los cuales son:

- Revisión y modernización del sistema web para que sea rápido y eficiente.
- Automatización de la administración de los usuarios y procesos usando un modelo cliente-servidor.
- Aseguramiento adecuado del servicio.
- Visualización del rendimiento y carga del servidor.
- Migración a un entorno virtual.

Todo esto a través de la reingeniería de los procesos que se llevan a cabo dentro de la administración del servidor y el equipo de reingeniería, los cuales son los administradores del servicio Web de la unidad de servicios de cómputo académico (UNICA).

5.2.2. Etapa 2 Identificación

El propósito de ésta etapa es el desarrollo de un modelo orientado al cliente, identificando procesos específicos que agregan valor, por lo cual mencionaremos los servicios con los que cuenta hasta el momento, así como los procesos que se llevan a cabo, los cuales son:

SERVICIOS:

- Servidor apache
- Sistema Manejador de Bases de datos de PostgreSQL
- PHP

PROCESOS:

- Administración de usuarios y procesos
- Respaldo de la información
- Seguridad del Servidor

Cuando decimos que es un modelo orientado al cliente, hacemos referencia a los usuarios del servicio y también al equipo de reingeniería o administradores del servicio; dónde se tomarán en cuenta las necesidades de cada uno de los clientes, como la implantación de nuevos servicios dados los avances tecnológicos, necesidades de los mismos, así como la automatización de los procesos requeridos por los administradores, como lo son la administración de usuarios, visualización del rendimiento del sistema y recopilación de información para la creación de buenas prácticas orientados a sitios Web institucionales, para así aumentar el ranking Web de nuestra institución. Teniendo los servicios y procesos de la siguiente manera.

SERVICIOS:

- Servidor apache.
- Sistema Manejador de Bases de datos de PostgreSQL.
- PHP.
- Perl.
- Jakarta Tomcat(para el uso de jsp's y servlest).
- SNMP (Protocolo Simple de Administración de Red).

PROCESOS:

- Administración de usuarios y procesos(Mediante Web min)
- Respaldo de la información(Cluster)
- Seguridad del Servidor
- Visualización del Rendimiento(Estadísticas)
- Visualización de búsquedas y exploración del sitio.

5.2.3. Etapa 3 Visión

El propósito de ésta etapa es desarrollar una visión del proceso capaz de producir un avance decisivo en rendimiento. Con lo cual se espera ofrecer un servicio Web de calidad en el mejor tiempo de respuesta, además que siempre se encuentre disponible y que por otra parte sean más automatizadas las tareas para una fácil administración en un futuro. Teniendo como punto de partida toda una migración a un esquema de virtualización.

5.2.4. Etapa 4 Solución

En ésta etapa se produce un diseño técnico y un diseño cultural-organizacional del trabajo.

La etapa de diseño técnico busca realizar la visión (Etapa 3), especificando las dimensiones técnicas de los nuevos procesos. Todo esto se realizará de la manera siguiente:

- Migración a un entorno de virtualización (nueva infraestructura)
- Evaluando el Sistema Operativo actual si es funcional o no.
- Implementando herramientas de Administración remota.
- Automatizando la forma de hacer respaldos de información, tanto de las bases de datos como de la información propia del servidor, ya sea vía programación o bien con alguna herramienta de software.
- Incrementando las herramientas de Seguridad que automaticen dicha función, así como en la herramientas de programación.

Actualmente nuestro servidor cuenta con 264 páginas entre las cuales se encuentran Sitios Web Institucionales y sitios personales de aquéllos que tienen que ver con la facultad, como los sitios personales de los profesores e incluso alumnos y sitios de algunas divisiones de la facultad.

.

En la Figura 5.2.1 muestra cómo se encontraba en un inicio el servidor de la Facultad.

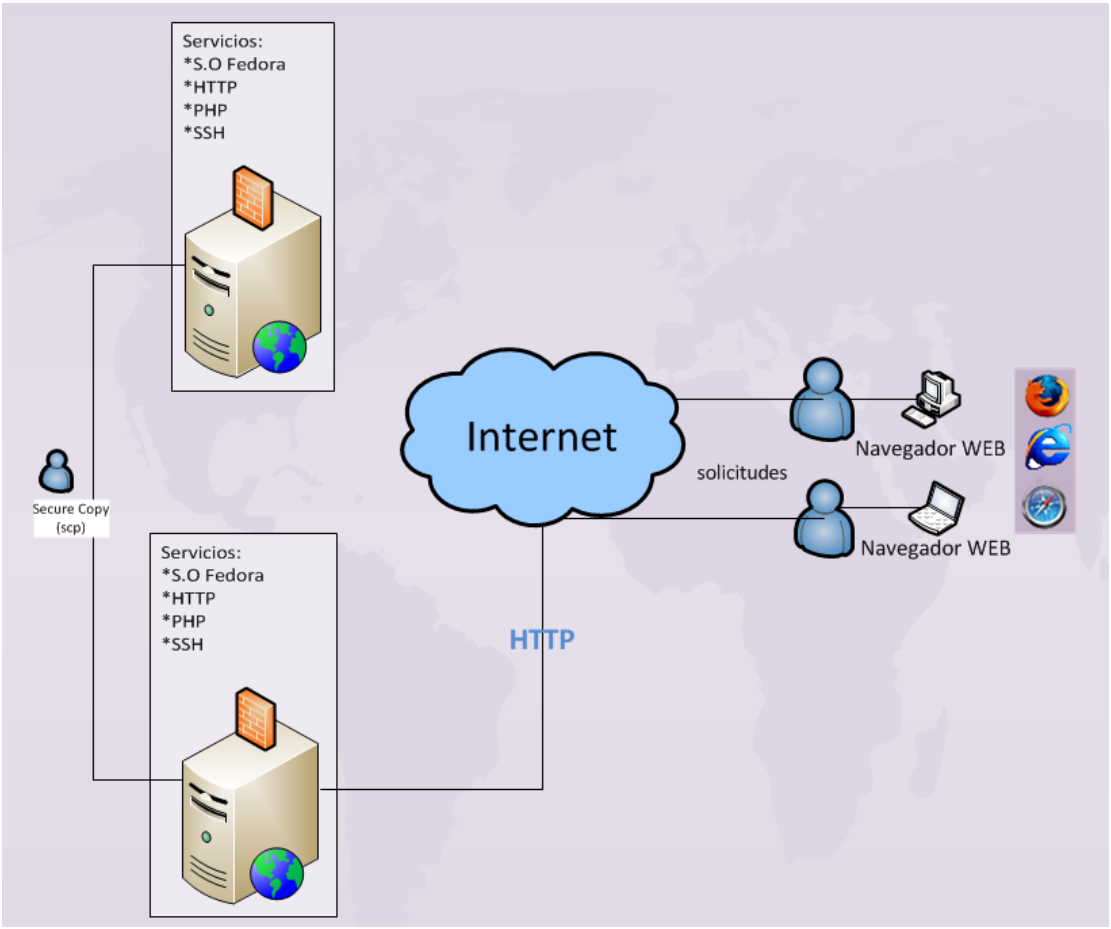


Figura 5.2.1.Esquema Básico del Servidor Web de la FI.

Una vez realizando todo un análisis se tomó la decisión de implementar el servidor de la manera siguiente como se muestra en la Figura 5.2.2.

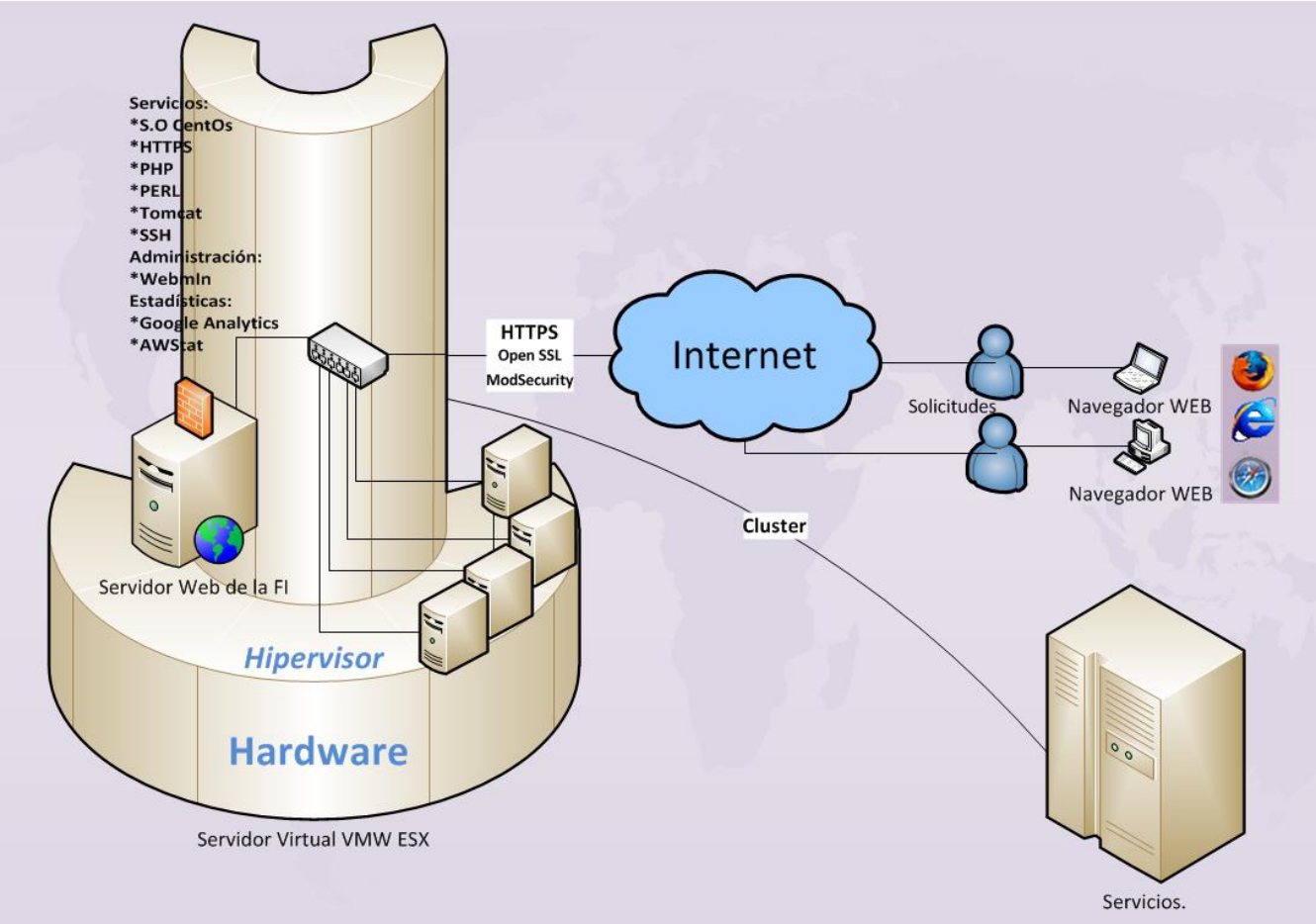


Figura 5.2.2. Esquema Virtualizado del Servidor Web de la FI y con más servicios.

Si observamos la figura 5.2.1. y la 5.2.2 cambiamos por completo el esquema de trabajo con dicho servidor, pues pasamos de trabajar en un esquema de servidor físico a un entorno completamente virtual, donde de primera instancia cambio por completo la manera de tener acceso a nuestros recursos y manipularlos de cierta forma, así como ampliamos las medidas de seguridad en la transmisión de información , como buscamos las necesidades a futuro de los usuarios en cuestión de lenguajes de programación para la realización de aplicaciones más dinámicas, así como también contamos con herramientas para un manejo más fácil de administración y se cuenta con información estadísticas de dicho servidor.

5.3. Manejo de Seguridad en Servidores Web

5.3.1. Seguridad en el sistema Operativo

El proceso de asegurar el servidor Web como ya lo mencionamos anteriormente comienza desde antes de la instalación eligiendo el tipo de sistema operativo que se va a instalar en el equipo, en nuestro caso elegimos una distribución Linux.

5.3.2. Hardening Linux

Hardening en sistemas operativos es la aplicación de acciones compuestas por un conjunto de buenas prácticas que son llevadas a cabo por el administrador de un sistema para reforzar al máximo la seguridad, con la finalidad de minimizar ataques removiendo servicios innecesarios y vulnerables, cerrando así huecos de seguridad protegiendo los controles de acceso para evitar las consecuencias de un entorno indeseable de seguridad para el sistema.

En la actualidad existen varias distribuciones Linux, todas ellas con características muy similares, como las mismas versiones de núcleo, las aplicaciones básicas etc. Sin embargo cada una tiene sus diferencias entre sí, como las herramientas de instalación, pues algunas de ellas especifican que servidores se activan durante el arranque, mientas que otras no hacen nada de esto y nos preguntan antes de la instalación de los mismos, o bien, especifican que paquetes queremos instalar con exactitud, mientras que otros no. Todas estas variables de instalación pueden incurrir en dejar huecos de seguridad dejando vulnerable nuestro sistema. Por tal motivo es muy importante que los administradores del sistema sepan con exactitud qué paquete se instalará; pues la seguridad de las diferentes distribuciones Linux dependerá de las configuraciones de los elementos con los que deseamos contar.

Cuando aplicamos el Hardening a un sistema no quiere decir que éste sea 100% seguro y que nunca le ocurrirá nada, pues con el paso del tiempo de un sistema se hace menos seguro; ya que en los paquetes y aplicaciones que se instalan en nuestro sistema se pueden descubrir amenazas con el paso del tiempo. Resguardar nuestro sistema es un curso constante realizando actualizaciones y cubriendo los parches de los mismos.

5.3.3. En cuanto el sistema Operativo

El proceso de Hardening en un sistema operativo empieza primero que nada teniendo claro que es lo que queremos instalar; pues las instalaciones por defecto incluyen muchos servicios, aplicaciones y paqueterías que para un servidor dedicado no sería de ninguna utilidad aquí entre menos instalemos mejor, lo más recomendable es que se instale sólo el sistema base.

5.3.3.1. Solamente instale lo mínimo necesario

Una vez teniendo claro que es lo que queremos instalar como mínimo para que funcione correctamente nuestro servidor, el encargado de instalar el sistema debe ser capaz de realizar una instalación personalizada y no realizar la que viene por defecto. En una instalación personalizada podemos eliminar las aplicaciones, servicios y paquetes innecesarios para nuestro sistema como lo son los juegos, servidores de red, herramientas Web, editores, gestión de documentos, inclusive eliminar el entorno gráfico del mismo (KDE y Gnome), pues siendo un servidor de aplicación no lo requiere.

Algo muy importante que debemos tomar en cuenta cuando se instala un sistema operativo es jamás instalarlo conectados a la red, pues mientras que nuestro sistema no haya recibido las pertinentes actualizaciones y parches su sistema es vulnerables a posibles ataques.

5.3.3.2. Instalación de parches y actualizaciones

La importancia de las actualizaciones y parches es que nuestro sistema es vulnerable a cualquier ataque pues cualquier vulnerabilidad debe ser corregida antes de conectarlo a la red pues un atacante puede darse cuenta e identificar que el sistema no tiene las protecciones pertinentes y puede penetrar en él, comprometiendo nuestro equipo.

Lo más recomendable es descargar las actualizaciones y parches necesarios en otro sistema con características similares a nuestro verificando su autenticidad con MD5 publicadas por el proveedor. Este sistema de prueba nos servirá para realizar pruebas de las nuevas versiones de actualizaciones antes de instalarlas en nuestro servidor de producción, pues muchas veces las actualizaciones y los parches pueden causar problemas en la operación de nuestro servidor. Una vez realizadas las pruebas satisfactoriamente, se crea un paquete de actualización para instalarlas en nuestro servidor de producción.

5.3.3.3. Gestor de arranque seguro

Después de una instalación en ocasiones dejamos servicios, demonios, comandos de una sola vez o herramientas que se inician en el arranque de nuestro equipo y que no son necesarias para nuestros propósitos, quedando expuestos a vulnerabilidades y es por eso que debemos de poner especial atención en la administración de ellos.

Los sistemas Linux usan uno de los dos gestores de arranque ya sea *LILO o Grub* quienes se encargan de controlar las imágenes de arranque de booteo, determinan que núcleo o kernel se inicia así como los servicios que deben iniciar cuando el sistema se inicia o reinicia. El gestor de arranque se carga después que entra el BIOS de nuestro sistema, da un periodo de entre 10 y 30 segundos para que nosotros seleccionemos que kernel es el que queremos arrancar en caso de tener más de un sistema, pero lo más recomendable es que sólo tengamos una sola versión de kernel para arrancar, eliminando las versiones más viejas limpiando al realizar actualizaciones de kernel, pues un atacante podría aprovecharse de eso e iniciar con una versión menos actualizada y dejando ahí huecos de seguridad en nuestro sistema, así como disminuir el tiempo en el menú del gestor de arranque para que el usuario no tenga la facilidad de iniciar con algún kernel anterior.

Tanto el gestor LILO como Grub son inseguros si el atacante tiene acceso físico a nuestro quipo, pues por defecto nos permitirá arrancar en modo usuario único (single-user) el cual tiene privilegios de root sin tener que ingresar una contraseña; por tal motivo debemos asegurar a los gestores de arranque para evitar que algún intruso acceda a nuestro sistema con todos los privilegios, esto lo podemos evitar colocando contraseña a nuestro gestor de arranque. En las versiones más actuales de algunas distribuciones Linux se puede colocar la contraseña al gestor de arranque desde el momento de la instalación, pero también lo podemos realizar posterior a la misma. Otra forma de proteger el equipo de éstos ataques es asegurar nuestro BIOS estableciéndole una contraseña y deshabilitar el arranque desde una unidad de CD/DVD o bien dependiendo el caso de disquete.

5.3.3.4. Init y secuencia de arranque.

Generalmente los sistemas operan con una gran cantidad servicios que se inician en el arranque como los indispensables para el funcionamiento básico del sistema, aplicaciones como apache, postgresql, etc. muchos de éstos servicios en ocasiones son realmente innecesarios ocasionando riesgos de seguridad, éstos los podemos habilitar o deshabilitar dependiendo de

nuestras necesidades mediante las utilidades de configuración o el setup del sistema en la parte de servicios del sistema (Figura 5.3.1 y 5.3.2).

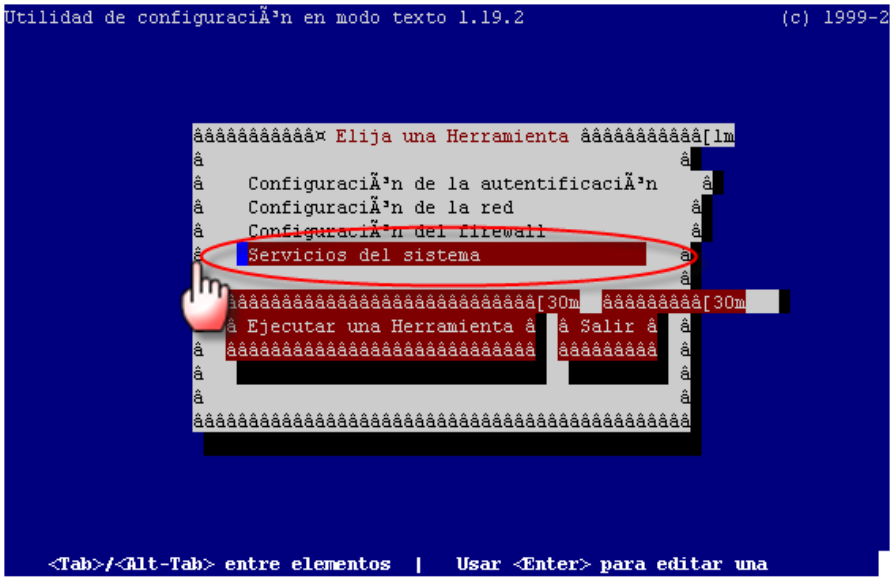


Figura 5.3.1. SetUp del sistema.

Figura 5.3.2. Servicios del sistema de inicio.

Otro lugar importante donde debemos asegurar a los scripts de inicio es el directorio /etc/rc.d/init.d, cada uno acompañado del comando chkconfig el cual nos especifica tres datos importantes, los niveles en los que se ejecuta, la secuencia para iniciar y detener el servicio; dato que podemos corroborar al inicio de cada script, usando el comando “chkconfig –list”,

podemos verificar para cada servicio su estado en cada nivel de ejecución como se muestra en la Figura 5.3.4.

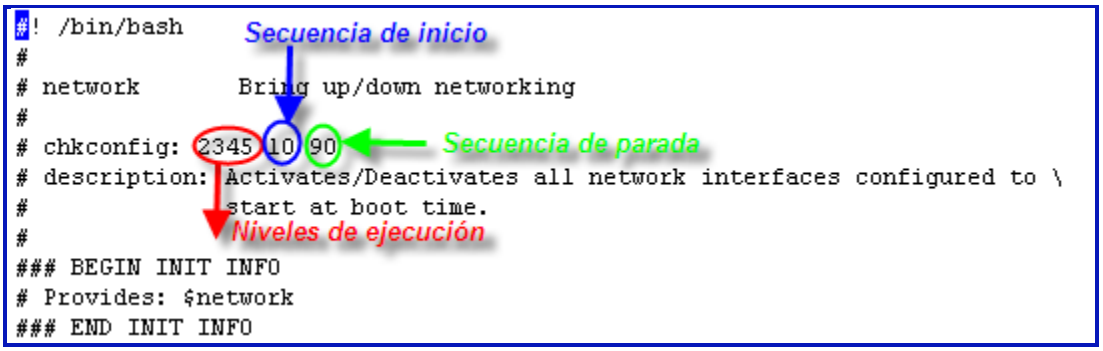


Figura 5.3.3. Ejemplo del servicio de red en el directorio initd.d.

firstboot	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:activo	6:desactivado
gpm	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
haldaemon	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:activo	5:activo	6:desactivado
hidd	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
httpd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
ibmasm	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
ip6tables	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
iptables	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
irda	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
irqbalance	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
isdn	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
jexec	0:activo	1:activo	2:activo	3:activo	4:activo	5:activo	6:activo
kudzu	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:activo	5:activo	6:desactivado
lvm2-monitor	0:desactivado	1:activo	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
mcstrans	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
mdmonitor	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
mdmpd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
messagebus	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:activo	5:activo	6:desactivado
microcode_ctl	0:desactivado	1:desactivado	2:activo	3:desactivado	4:activo	5:activo	6:desactivado
multipathd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
netconsole	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
netfs	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:activo	5:activo	6:desactivado
netplugd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
network	0:desactivado	1:desactivado	2:activo	3:activo	4:activo	5:activo	6:desactivado
nfs	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado
nfslock	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:activo	5:activo	6:desactivado
nscd	0:desactivado	1:desactivado	2:desactivado	3:desactivado	4:desactivado	5:desactivado	6:desactivado

Figura 5.3.4. .Salida del comando chkconfig –list.

Aquí y mediante el comando chkconfig se puede dar de alta o quitar servicios, es recomendable que si no se va a dar de alta algún script a la secuencia, lo mejor sería eliminarlo del directorio; después de haber hecho esto es recomendable también el asegurarlo para evitar que se pueda acceder o modificar algo en el directorio sin estar autorizado para tal fin (Figura 5.3.5).

```
chown root:root /etc/rc.d/init.d/*
chmod –R 700 /etc/ rc.d/init.d/*
```

Figura 5.3.5. Asegurando el directorio /etc/rc.d/init.d

5.3.3.5. Secuencia de Arranque (Boot)

Otro punto al que hay que poner especial atención es a la Secuencia de arranque, el orden en que se inicia y detienen los servicios en el sistema es importante; hay que asegurarse que se inicien los servicios como Firewall, (iptables) y el demonio syslog, antes de levantar el demonio de red, con esto nos aseguramos que no vamos a estar conectados sin protección, lo mismo durante el cierre, asegurándonos de parar los servicios de red antes de detener el firewall y el syslog. Asegurándonos que se inicien con la secuencia que se les asignó .Ver el ejemplo del al Figura 5.3.6.

```
#! /bin/bash
#
# network       Bring up/down networking
#
# chkconfig: 2345 10 90
# description: Activates/Deactivates all network interfaces configured to \
#              start at boot time.
```

```
#!/bin/sh
#
# iptables      Start iptables firewall
#
# chkconfig: 2345 08 92
# description:  Starts, stops and saves iptables firewall
#
# config: /etc/sysconfig/iptables
# config: /etc/sysconfig/iptables-config
```

```
[root@hardening rc3.d]# ln -s /etc/init.d/syslog /etc/rc3.d/S09syslog_

#!/bin/bash
#
# syslog        Starts syslogd/klogd.
#
#
# chkconfig: 2345 9 93
# description: Syslog is the facility by which many daemons use to log \
# messages to various system log files.  It is a good idea to always \
# run syslog.
```

Figura 5.3.6. Prioridad e inicio de servicios.

5.3.3.6. Asegurando consolas y terminales virtuales Pantallas de inicio de sesión

Primero que nada definiremos que es una consola, una terminal virtual y que es una pantalla de inicio de sesión:

- **Consola.-** Cuando nos referimos a consolas de un equipo de cómputo, nos referimos a la consola que nos aparece cuando nos conectamos físicamente a nuestro servidor Linux, donde un usuario que se conecta a éste, puede realizar operaciones que no sería posible realizar si se conectara desde otros lugares como por ejemplo una conexión segura a través de SSH(Secure Shell).
- **Terminal Virtual.**- Las terminales virtuales son las distintas sesiones que se arrancan en Linux en modo consola y a las que accedemos con CTRL+ATL+Fn donde n es un número del 1 al 6.
- **Pantalla de inicio de sesión.**- Son aquellas que se presentan cuando cualquier usuario del sistema se conecta al mismo iniciando una sesión en el equipo. En muchas ocasiones cuando no ponemos cuidado en ellas puede aparecer información sobre nuestro sistema siendo aprovechada por algún atacante.
-

5.3.3.7. Seguridad en la Consola

Nosotros podemos limitar donde puede iniciar sesión el súper usuario root en el sistema limitándolo a un conjunto especifico de terminales virtuales y consolas esto es posible en el archivo /etc/securetty, lo único que hay que modificar es comentar las terminales y dejar las que consideremos necesarias (Figura 5.3.7).

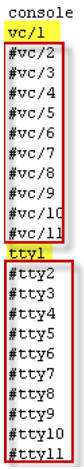


Figura 5.3.7. Asegurando las consolas en /etc/securetty.

El fichero /etc/securetty le permite especificar desde qué dispositivos tty (terminales) tiene root permitido el inicio de sesión. Le sugerimos que comente todas las líneas excepto vc/1 si está usando devfs y todas las líneas excepto tty1 si está usando udev. Esto le asegurará que root sólo puede hacer un login en sólo una terminal. Una vez realizado esto debemos asegurara dicho archivo para que sólo root pueda modificarlo como se muestra en la Figura 5.3.8.

```
chown root:root /etc/securetty
chmod 0600 /etc/securetty
```

Figura 5.3.8. Asegurando las consolas en /etc/securetty dándole permisos sólo a root.

Por otra parte debemos asegurar el directorio /etc/security/consoleapps, pues este directorio contiene programas adicionales, donde cualquier usuario que inicia sesión en una consola los puede ejecutar y que no nos conviene que ejecute. Como los siguientes (Figura 5.3.9):

```
authconfig        halt          pm-suspend         system-config-network
authconfig-tui    kbdrate       pm-suspend-hybrid  system-config-network-cmd
cpufreq-selector  neat          poweroff
eject             pm-hibernate  reboot
gnome-system-log  pm-powersave  setup
```

Figura 5.3.9. Contenido del directorio /etc/security/consoleapps.

Como ven son un gran riesgo estos comandos para ser ejecutados desde consola una vez que se ha iniciado sesión por tal motivo es recomendable eliminar el contenido de este directorio de la siguiente manera(Figura 5.3.10):

```
rm –f /etc/security/console.apps/*
```

Figura 5.3.10. Eliminando comandos de inicio de sesión en /etc/security/consoleapps.

5.3.3.8. Seguridad en Terminales Virtuales

Las terminales virtuales nos son útiles cuando nos conectamos físicamente y abrimos varias sesiones para realizar algún conjunto de procesos en cada terminal, pero éstas pueden ser muy peligrosas si dejamos las sesiones abiertas sin vigilancia. La forma más sencilla de bloquear las terminales es mediante el programa **vlock** es un programa para bloquear una o más sesiones en la consola de Linux. Esto es especialmente útil para las máquinas Linux que tienen varios usuarios con acceso a la consola. Un usuario puede bloquear su período de sesiones al mismo tiempo que permite que los usuarios utilicen el sistema en otras consolas virtuales, además que

permite bloquear todas las terminales virtuales y desactivar el cambio entre terminales virtuales.

Para bloquear la terminal virtual actual lo puede hacer como se muestra en la Figura 5.3.11.

```
# vlock -c
Este TTY está bloqueado.
Por favor, introduzca la contraseña para desbloquear.
root Contraseña:
```

Figura 5.3.11. Bloqueando terminal virtual actual.

Y para desbloquear la terminal tenemos que introducir la contraseña de root. Para deshabilitar todas las terminales virtuales y el cambio entre ellas lo hacemos como se muestra en la Figura 5.3.12.

```
# vlock –a
Toda la pantalla de la consola está bloqueada.
Usted no será capaz de cambiar a otra consola virtual.
Por favor, introduzca la contraseña para desbloquear:
root Contraseña:
```

Figura 5.3.12. Bloqueando terminales virtuales.

5.3.3.9. Seguridad en las Pantallas de Inicio de Sesión

Estas pantallas es lo primero que ve cuando un usuario se firma a un sistema, además puede revelar mucha información del mismo, como el sistema operativo usado, la versión del núcleo y más, si un atacante tiene esta información es mucho más fácil para ellos descubrir nuestros huecos de seguridad, así que nunca hay que mostrar éste tipo de información en una pantalla de inicio. Éste tipo de información que se muestra en un inicio se encuentra en los siguientes archivos.

/etc/issue y /etc/issue.net como se muestra en la siguiente Figura 5.3.13.

```
[root@localhost ~]# cat /etc/issue
CentOS release 5.5 (Final)
Kernel \r on an \m

[root@localhost ~]# cat /etc/issue.net
CentOS release 5.5 (Final)
Kernel \r on an \m
[root@localhost ~]#
```

Figura 5.3.13. Ejemplo de lo que podría aparecer.

Lo que hay que hacer es limpiar estos archivos y cambiar los permisos para que sólo root los pueda editar de la siguiente manera (Figura 5.3.14):

```
clear > /etc/issue
clear > /etc/issue.net
chown root: root / etc / issue / etc / issue.net
chmod 0600 / etc / issue / etc / issue.net
```

Figura 5.3.14. Limpiando y cambiando permisos a root de /etc/issue y /etc/issue.net.

5.3.3.10. Borrando Grupos y Usuarios innecesarios

El sistema crea cuentas de usuarios y grupos por default, que no son requeridas y para mejorar la seguridad del sistema deben removerse de igual forma que se eliminó del sistema los paquetes que eran innecesarios; algunos los podemos ver por puro sentido común, pero ésta decisión de eliminar usuarios se tiene que ver de acuerdo con las necesidades del equipo y de los servicios que brinde el sistema haciendo un análisis con los administradores del mismos, entre los que podemos eliminar se encuentran los siguientes (Tabla 5.3.1).

Tabla 5.3.1. Tabla de usuarios necesarios e innecesarios en el sistema.

Usuario	Descripción	Removerlo?
adm	Herramientas de diagnóstico y contabilidad	Sí
backup	Para realizar copias de seguridad de archivos críticos	No
bin	Ejecutables para usuarios y comandos	No
daemon	Posee y administra los procesos	No
desktop	Usuario KDE	Sí
ftp	Usuario default de FTP	Sí
games	Usuario de juegos	Sí
gdm	Usuario GDM	Sí
gnats	Usuario gnats para seguimiento de errores	Sí
gopher	Usuario gopher	Si
halt	Usuario /sbin/halt	No
identd	Para el demonio identd	Sí
irc	Usuario de internet realy chat	Sí
list	Usuario cartero	Sí
lp	Usuario de impresión	Sí
Lpd	Usuario de impresión	Sí
mail	Usuario predeterminado para el agente de transferencia de mail	Tal vez

mailnull	Usuario de Sendmail	Sí
man	Usuario de la base de datos del manual	No
news	Usuario predeterminado de news	Sí
nfsnobody	Usuario NFS	Sí
nobody	Usuario predeterminado de Apache NFS	Tal vez
nscd	Usuario para el demonio del cache de nombre de servicios	Sí
Ntp	Usuario del protocolo de red tiempo	No
operator	Usuario de operaciones	Sí
proxy	Usuario predeterminado de proxy	Sí
root	Súper usuario root	No
rpc	Usuario RPC	Sí
rpcuser	Usuario predeterminado de RPC	Sí
rpm	Usuario RPM	No
Shutdown	Usuario de apagado	No
sshd	Usuario de sshd	No
sync	Usuario Sync	Sí
Sys	Usuario de sincronización	No
telnetd	Usuario predeterminado de telnetd	Sí
uucp	Usuario predeterminado de uucp	Sí
vcsa	Memoria de consola virtual	No
www-data	Datos www	Sí
Xfs	Fuente X server	Sí

Eliminaremos los usuarios con el comando **userdel** y si desea eliminar su directorios hacemos uso de la **opción –r**.

Por otra parte los usuarios que se encuentran en la tabla anterior también cuentan con su respectivo grupo y de igual forma hay que eliminarlos, generalmente el grupo se llama igual que el usuario y lo eliminaremos con el comando **groupdel.**

Una vez que hemos eliminado los usuarios y grupos innecesarios debemos verificar que los usuarios del sistema no hagan login al mismo, porque en caso de hacer eso significa que algún usuario mal intencionado logró tener acceso al sistema, para esto tenemos que poner especial atención a el archivo de configuración de usuarios **/et c/passwd** que tiene el siguiente formato:

Nombre_de_usuario:contraseña:UID:GID:Descripcion del usuario:Directorio:Shell

Si ponemos atención el último campo es el perteneciente al Shell que se le brindará a cada uno de los usuarios; este campo es de gran importancia pues nos asigna el intérprete por el cual nos podernos conectar al sistema o negarnos esa posibilidad con /sbin/nologin, esto con sus desventajas ya que no sólo nos impide ingresar al sistema, sino que también nos genera entradas al syslog quien se encarga de llevar todos los registros del sistema; el hecho de que /sbin/nologin sea un script lo hace peligroso porque en ocasiones se puede vulnerar el shellscript. El hecho de que no puedan iniciar sesión no garantiza que no pueda ser usada la cuenta para otros fines. Una posible solución es que se manden estos shells a /dev/null o se use un binario de noshell aunque con la desventaja de que no se generan entradas en el syslog. Una solución a esto es utilizar el binario de noshell de Titan hardening Tools, compilarlo y utilizarlo, con el beneficio de que si se generan entradas en bitácoras.

Procedimiento:

Conseguir el archivo fuente noshell.c y corroborar la integridad del mismo, esto se puede hacer con la ayuda de md5sum .

```
# md5sum noshell.c
d4909448e968e60091e0b28c149dc712 noshell.c
```

Elaborar al MakeFile para compilar el archivo noshell (Figura 5.3.15).

```
CC = gcc
CPPFLAGS =
CFLAGS   = -static
LDFLAGS  = -dn
LIBS     = -static /usr/lib/libc.a -static /usr/lib/libnsl.a
noshell: noshell.o
        $(CC) $(CFLAGS) -o noshell $(LIBS) $(LDFLAGS) noshell.o
```

Figura 5.3.15. MakeFile de noshell.

Hecho esto se introducen los siguientes comandos.

```
#make noshell
```

Copiamos el noshell resultante a /sbin y eliminamos los archivos fuente, salidas y binarios.

```
#cp noshell /sbin
#rm –f noshell.c noshell.o noshell
```

Ahora podemos utilizar /sbin/noshell para evitar que los usuarios se firmen en el servidor por ejemplo el usuario daemon.

daemon:x:2:2:daemon:/sbin:/sbin/noshell

Y para la siguiente vez que intente acceder al sistema generará una entrada en bitácora de advertencia.

5.3.3.11. Administración Remota

En la mayoría de los casos, la administración remota juega un papel fundamental en la gestión de un sistema, ya sea por comodidad, por facilidad o porque no es posible el acceder físicamente a la zona donde se encuentra nuestro sistema, tomando en cuenta que por lo general no nos encontramos físicamente cerca de ellos todo el tiempo, ya sea por cuestiones de acceso a los centros de dato, por reglas de seguridad, por cuestiones de que el centro de datos se encuentre fuera de la estación de trabajo , etc. resultando una ventaja importante aunque también un riesgo; esto porque en algunos casos la comunicación con el sistema se hace mediante internet, con la posibilidad de que alguien esté a la escucha o monitoreando lo que se transmite y el uso de software que no permite comunicaciones seguras (telnet, ftp, rlogin, etc.) incrementa este riesgo, refiriéndonos a comunicaciones seguras el que la información no viaje como texto plano, sin ningún tipo de consideración para el ocultamiento de la misma (cifrado).

De aquí el incremento y uso cada vez más extendido del SSH del paquete OpenSSH que por lo general viene incluido en cualquier sistema tipo linux, el cual reemplaza a las herramientas que manejan el texto en claro y que adicionan seguridad en otras herramientas; SSH (Secure Shell) es el nombre de un protocolo y del programa que lo implementa, nos sirve para acceder a maquinas (hosts) de manera remota a través de la red, permitiéndonos manejar la computadora mientras se tenga el intérprete, también permite copiar datos de manera segura, administrar claves RSA o DSA para el establecimiento de llaves de confianza. Trabaja de forma similar a como lo hace telnet con la principal diferencia de que se usan técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya sin poder ser legible a ningún tercero, ni lo que se escribe durante la sesión.

SSH cuenta con dos versiones del protocolo que pueden ser utilizadas, siendo la versión 2 la que se considera como más segura que la versión 1.Para permitir las conexiones remotas al

servidor es necesario tener levantado el demonio sshd que escucha por default el puerto 22, para levantar el servicio podemos teclear lo siguiente:

sshd –p 22 -p le dice que se levante en el puerto 22

o directamente desde el demonio /etc/init.d/sshd (start|restart)

Veamos más sobre la configuración de ssh y sshd, del lado del cliente la configuración es controlada por el archivo ssh_config , para el lado del servidor tenemos el archivo sshd_config

Como observamos en la Figura 5.3.16. , vincula el servicio con el puerto 22, como ya lo habíamos mencionado se selecciona únicamente el protocolo versión 2. El siguiente paso es seleccionar el tipo de control de cómo el sshd manejara los archivos log mediante el daemon syslog, aquí se especifican los parámetros para el registro de eventos, SyslogFacility especifica el tipo de registros que hará, en éste caso es AUTHPRIV.

```
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

# Disable legacy (protocol version 1) support in the server for new
# installations. In future the default will change to require explicit
# activation of protocol 1
Protocol 2

# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 1h
#ServerKeyBits 1024
```

Figura 5.3.16. Archivo sshd_config seleccionando el protocolo en su versión 2.

En LogLevel INFO es el valor predeterminado, otros parámetros están especificados en la página del manual de sshd_config (man sshd_config), los parámetros DEBUG2 y DEBUG3 cada uno especifican el nivel más alto de logueo, guardar registros con el nivel DEBUG viola la privacidad de los usuarios y por lo tanto no es recomendada.

Ahora sigue la parte de los métodos de autenticación.
Por default se encuentra de ésta manera (Figura 5.3.17):

```
# obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
SyslogFacility AUTHPRIV
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6

#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile     .ssh/authorized_keys

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
```

Figura 5.3.17. Archivo sshd_config LogLevel INFO por default.

La primera opción es: #LoginGraceTime 2m, esto le dice al servidor sshd el tiempo en el que desconectara al usuario después de que no ha podido iniciar sesión satisfactoriamente, si el valor es 0 no hay límite de tiempo para que un usuario se autentique, lo cual no es recomendable ya que de ésta manera podrían hacer ataques de fuerza bruta, o usando métodos de diccionario y así adivinar la contraseña por lo tanto no es recomendable dejar este parámetro a 0, el valor predeterminado es: 2m, es decir 120 segundos, en nuestro caso permitimos sólo 60 segundos

Se descomentara y se dejara por default haciendo explícito:

LoginGraceTime 1m

El siguiente parámetro es uno de los más importantes en cuanto a seguridad en el servidor sshd se refiere:

#PermitRootLogin yes

Este parámetro por default le dice que acepte conexiones con el usuario "root" lo cual no es nada recomendable, porque alguien podría identificarse como tal usuario y podría adivinar la contraseña, y tendría privilegios de "root" y así podría hacer cualquier cosa, por lo tanto se desactivará poniendo:

PermitRootLogin no

Luego sigue:

StrictModes

Esto significa que sshd revisará los modos y permisos de los archivos de los usuarios y el directorio $HOME del usuario antes de aceptar la sesión. Esto es normalmente deseable porque algunos dejan sus directorios accidentalmente con permiso de escritura para cualquiera, el valor predeterminado es 'yes'. Por lo tanto lo dejaremos con su valor predeterminado y sólo lo des comentaremos para hacerlo explícito:

StrictModes yes

Y el último de estas opciones es:

MaxAuthTries 6,

Esta opción especifica el máximo número de intentos de autenticación permitidos por conexión. Una vez que el intento alcanza la mitad de éste valor, las conexiones fallidas siguientes serán registradas. El valor predeterminado es 6. En éste caso lo dejaremos con su valor predeterminado, pero lo descomentaremos para hacerlo explícito, así:

MaxAuthTries 6

5.3.3.12. Seguridad en el montaje del sistema de archivos

Durante el proceso de instalación del sistema operativo, se nos pide que creemos alguna partición en el disco duro del equipo; cuando nos referimos a partición nos referimos a áreas de disco duro que se reservan para los sistemas de archivos. Algo muy importante que debemos hacer en el momento de la instalación es que no se deben colocar los sistemas de archivos raíz y de un usuario en la misma partición de Linux pues es muy peligroso, pues dejamos un hueco de seguridad aumentando la riesgo que los usuarios puedan explorar los programas para acceder a áreas restringidas. Si dejamos todo nuestro Linux en una sola partición, en el momento en que se dañen unos cuantos archivos ocasionaría problemas, incluso llevándonos a la reinstalación de todo el sistema.

Por lo anterior debemos hacer un análisis de cómo distribuiremos nuestro sistema de archivos en base a nuestras necesidades; una vez teniendo esto recurriremos a crear particiones

independientes para cada uno de nuestros sistemas principales, lo cual nos permitirá administrar de forma más óptima obtenido copias de seguridad de nuestro sistema con mayor facilidad.

Cuando el sistema arranca cada archivo del sistema es montado para permitir el acceso a la información contenida en ellos (cuando decimos montar, nos referimos a la manera en que linux permite utilizar los distintos sistemas de archivos), puede ser montado con diferentes opciones, estas van desde la capacidad de escribir a un sistema de archivos para especificar qué tipo de archivos se pueden ejecutar en ese sistema de archivos. Estas opciones le permiten bloquear la capacidad y funcionalidad de cada uno de sus archivos del sistema. Estas opciones son controlados por el archivo /etc /fstab. El archivo /etc/fstab es un simple archivo de textos sin formato en el cual se especifican las opciones de montaje de los sistemas de archivos, donde cada línea escrita en él se encargara de administrar un sistema de archivos en específico (Figura 5.3.18).

```
[root@localhost ~]# cat /etc/fstab
/dev/VolGroup00/LogVol00 /                       ext3    defaults        1 1
LABEL=/boot              /boot                   ext3    defaults        1 2
tmpfs                    /dev/shm                tmpfs   defaults        0 0
devpts                   /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                    /sys                    sysfs   defaults        0 0
proc                     /proc                   proc    defaults        0 0
/dev/VolGroup00/LogVol01 swap                    swap    defaults        0 0
```

Figura 5.3.18. Archivo /etc/fstab por default.

El archivo /etc/fstab cuenta con seis campos:

- La especificación del sistema de archivos, es decir el dispositivo de bloque o el sistema de archivos que se va a montar.
- La ubicación del sistema de archivos, se especifica el punto de montaje.
- El tipo de sistema de archivos, ya sea minix, extendido (ext#), dos, iso9660, swap, NFS etc.
- Las opciones de montaje del sistema de archivos, en ésta parte se especifica el nivel de acceso que tendrán tanto los usuarios como el sistema en este sistema. Ésta parte es muy importante, pues es de gran importancia en la seguridad de nuestro sistema de archivos; aquí podemos encontrar las siguientes opciones (Tabla 5.3.2).

Tabla 5.3.2. Opciones de montaje de las particiones de un S.O. Linux.

Opción	Descripción
auto	El sistema de archivos se montará en el arranque del sistema
noauto	El sistema de archivos NO se montará en el arranque del sistema
dev	Permite la interpretación de bloques o de carácter especial en los dispositivos de este sistema de archivos.
nodev	No interpreta bloques ni caracteres especiales
exec	La ejecución de binarios es permitida en el sistema de archivos
noexec	La ejecución de binarios es NO permitida en el sistema de archivos
suid	setuid está permitida para tomar efecto en el sistema de archivos
nosuid	setuid NO está permitida para tomar efecto en el sistema de archivos
user	Usuarios normales pueden montar el dispositivo
nouser	Sólo root pueden montar el dispositivo
owner	Permite que el propietario del dispositivo monte el sistema de archivos
ro	El sistema de archivos será montado en modo read-only
rw	El sistema de archivos será montado en modo read-write
defaults	Aplica las opciones rw, suid, exec, auto, nouser, async al sistema de archivos

Aquí se define qué sistema de archivos puede montarse, así como varias características y facetas de los mismos; nosotros nos dirigiremos a la 4 columna que es donde se puede especificar opciones que definan como será montado por ejemplo: como sólo lectura-escritura. Además de cómo el usuario interactuara con la partición montada.

Hay que poner especial cuidado con la opción noexec, ya que no puede ser aplicada a cualquier sistema de archivos, como en las particiones /, /boot ya que pueden evitar que se inicie el sistema operativo pues éste requiere de ejecutar algunos binarios. Las opciones de montaje deben ser usadas con precaución para no caer en problemas con el sistema por ser usadas incorrectamente.

- El siguiente parámetro es el parámetro de volcado (hacer copias de seguridad) del sistema de archivos; éste es un valor numérico.

- Por último tenemos el número de secuencia de verificación del sistema de archivos. Aquí se especifica la prioridad del sistema de archivos para las verificaciones de integridad que realiza el comando fsck.

5.4. Políticas

Una política es aquella actividad que tiende gobernar o dirigir en beneficio de algo mediante el uso de leyes, reglas y prácticas que me permitan llegar hacia la toma de decisiones para la resolución de objetivos de un determinado grupo, en éste caso sería en el uso del Web.

Una política de seguridad es el conjunto de leyes, reglas que se crean y buenas prácticas que permiten salvaguardar los activos de una organización, brindando seguridad dentro de ésta. Estas políticas sirven para cumplir los objetivos de seguridad de una organización. Las políticas de seguridad son los documentos que describen la forma adecuada del uso de los recursos de un sistema de cómputo, las responsabilidades y derechos que tanto usuarios como administradores tienen que hacer durante un incidente de seguridad. Una política de seguridad siempre indica ¿Qué? se va a realizar, pero no el ¿cómo? Pues los procedimientos son los encargados de esté.

Existen dos tipos de filosofías de políticas:

A. **Prohibitiva:** Absolutamente "todo está prohibido a excepción de lo que está permitido", y ésta es la más usada por las instituciones y empresas.
B. **Permisiva:** "Todo está permitido a excepción de lo que está prohibido".

Cabe señalar que la ***Facultad de Ingeniería*** hace el uso de políticas de seguridad del tipo prohibitivas.

5.4.1. Principios Fundamentales

Un conjunto de políticas de seguridad debe representar fielmente los siete principios fundamentales.

a) Responsabilidad Individual: Las personas deben estar conscientes de sus actos y de sus responsabilidades.
b) Autorización: Quién, cuándo y cómo puede acceder a ciertos recursos.
c) Mínimos privilegios: el personal debe contar con la autorización correspondiente para realizar su trabajo.

d) Separación de obligaciones: Deben separarse las actividades entre varias personas, disminuyendo la dependencia y la posible confabulación.
e) Auditoria: Plantear revisiones periódicas, ayudara al personal a llevar un control en las actividades de la organización así como mejorarlas.
f) Redundancia: Debe indicarse la creación de respaldos y evitar tener información no autorizada o incompleta de manera repetida, lo cual puede ocasionar problemas en las actividades de la organización.
g) Reducción de riesgos: Se busca disminuir el riesgo de los activos considerando las herramientas necesarias aceptables para los mismos.

5.4.2. Políticas de la Facultad de Ingeniería.

En la actualidad, la Facultad de Ingeniería cuenta con sus propias políticas de seguridad en cómputo, las cuales fueron revisadas por última vez en marzo de 2003. Los rubros más importantes que nos interesan son los siguientes:

- Políticas de seguridad física
- Políticas de cuentas
- Políticas de contraseñas
- Políticas de control de acceso
- Políticas de respaldos
- ***Políticas de Web***

Pondremos especial atención en las políticas de Web pues en nuestro caso son las que más nos interesan. Tomando como base estas políticas realizaremos un análisis de las mismas, para actualizarlas o bien aportar algo a las mismas para el uso del servicio Web.

5.4.2.1. Políticas de Seguridad Física.

Las políticas de seguridad física nos permiten proteger contra amenazas y vulnerabilidades que puedan sufrir, todos nuestros recursos físicos.

Dentro de estas políticas sugiero agregar las siguientes políticas:

- El lugar donde se encuentren los servidores deben contar con equipos de control de medio ambiente.

- Para acceder a los centros de datos se debe contar al menos con dos tipos de controles (chapas, sensor biométrico, huella digital, registro).
- Sólo el personal autorizado puede tener acceso a los medios (llaves, candados, etc.) para acceder a los centros de cómputo y es responsabilidad de los mismo, resguardar dichos medios.
- Todo el personal que ingrese a las instalaciones y que no pertenezca a la institución debe ingresar a las instalaciones con previa autorización del jefe inmediato y con una identificación que los identifique como tal, además de ser vigilados y acompañados por personal autorizado durante todo el tiempo de su estancia.
- Contar con un seguro contra robo y pérdidas.
- Está prohibido mover o reubicar sin previa aprobación cualquier tipo de equipo de cómputo, ya sea PCs, servidores, equipo de comunicaciones.
- Almacenar equipos redundantes y la información de respaldos en un lugar seguro y distante del sitio principal.

5.4.2.2. Políticas de Cuentas

Establecen qué es una cuenta de usuario de un sistema de cómputo, cómo está conformada, a quién puede serle otorgada, quién es el encargado de asignarlas, cómo deben ser creadas y comunicadas.

Agregaremos las siguientes políticas:

- Los administradores poden eliminar cuentas que se encuentren en desuso o se encuentren inactivas después de un determinado tiempo.
- Si dicha cuenta de usuario cambia de responsable, es responsabilidad de los responsables directos notificarlo de manera inmediata a los administradores para que realice un cambio inmediato de la contraseña y del responsable de la misma.

5.4.2.3. Políticas de Contraseñas

Las políticas de contraseñas son las más importantes, porque son lo primero que tratan de vulnerar en un sistema y por tal motivo debemos tener especial cuidado en el momento de construirlas.

Se sugieren las siguientes:

- Todas las contraseñas deben contar con al menos 8 caracteres y máximo entre 12 y 14 caracteres, debe estar conformada por:
    - Números, letras y símbolos.
    - Letras mayúsculas y minúsculas.
    - Cuanto más larga, más robusta es.
- Está prohibido usar palabras escritas al revés ni que sean escritas en otro idioma.
- Está prohibido usar dicha contraseña en alguna otra cuenta de usuario, como correo electrónico o algún otro sistema.
- Los usuarios son responsables de aprender dicha contraseña de memoria para evitar el robo o pérdida de la misma.
- Por ningún motivo se debe compartir la contraseña con alguna otra persona, pues es confidencial y personal.

5.4.2.4. Políticas de Control de Acceso

Su función principal es controlar el acceso a áreas restringidas evitando accesos no autorizados y no sólo hablamos de acceso físico, sino también en la entrada a los sistemas como desde donde y de qué manera se van autentificar ante ellos.

Se sugieren las siguientes:

- Los usuarios deben asegurarse de cerrar todas las sesiones que abra, en caso de dejar sesiones inactivas durante un periodo de tiempo largo, el administrador puede eliminar estas sesiones.
- Se debe definir los perfiles de acceso, los usuarios estándar y los administradores.

5.4.2.5. Políticas de Respaldos

Contar con respaldos de nuestra información, pues en caso de algún desastre podríamos volver a un punto anterior por lo menos, por lo que es de vital importancia realizar respaldos de manera periódica.

Esta política cuenta con dos ramas una para el usuario y otra para el administrador sugeriremos las siguientes:

PARA LOS USUARIO.

- Contar con al menos dos copias de respaldos en diferentes lugares y que sean seguros.
- Contar con un control de versiones de sus respaldos.

PARA LOS ADMINISTRADORES.

- Contar con al menos dos copias de respaldos en diferentes lugares y que sean seguros.
- Realizar los respaldos con un medio seguro.
- Etiquetar los respaldos para evitar duplicidad.

5.5. Definición de Ranking Web

Un ranking es una relación entre un conjunto de elementos, con los cuales para uno o varios criterios, el de primero de ellos siendo el de mayor relevancia al segundo, esté a su vez mayor que el tercero y así de manera sucesiva con cada uno de estos criterios, de tal forma que se puede tener que dos o más elementos diferentes puedan tener la misma posición.

El ranking Web es una marera de comparar distintos aspectos y características que permiten que un sitio Web pueda tener un mejor posicionamiento dentro de internet, donde el objetivo principal de éste es aplicar medidas de mejora en los sitios Web para así promover los sitios Web.

Por lo general casi todos los ranking Web su objetivo principal son más de carácter comercial, los motores de búsqueda no ofrecen resultados estables, pues muchos de ellos se centran generalmente en muy pocos aspectos relevantes como para promover un sitio Web de carácter Institucional.

El ranking mundial de universidades en la Web, cumples otros parámetros con mucha mayor cobertura que otros rankings, toma en cuenta algunas métricas como la visibilidad, la proyección, contenidos académicos, el uso de redes sociales, entre otros; con el objetivo de motivar a docentes e investigadores de las universidades del mundo a tener presencia en la Web.

5.5.1. Ranking Web de Universidades del mundo

El "Ranking Mundial de Universidades en la Web", elaborado por el Centro de Información y Documentación (CIDOC) es una iniciativa del Laboratorio de Cibermetría, que pertenece al Consejo Superior de Investigaciones Científicas (CSIC), el mayor centro nacional de investigación de España, el CSIC, forma un papel muy importante en la formación de investigad ores y técnicos en diferentes ramas en ciencia y tecnología. Este estudio de ranking Web para las universidades es muy completo, pues éste mide la penetración y presencia en internet a más de 10, 000 centros de educación superior de todo el mundo, donde son ordenados de acuerdo al volumen de información publicad, la visibilidad y el impacto de estas páginas según en número de enlaces que reciben.

Se diseñaron cuatro indicadores a partir de los resultados cuantitativos obtenidos de los principales motores de búsqueda como se detalla a continuación:

a) **Tamaño (S)**. Número de páginas recuperadas desde los siguientes motores de búsqueda: Google, Yahoo, y Bing.
b) **Visibilidad (V)**. El número total de enlaces externos únicos recibidos (inlinks) obtenidos de Yahoo Site Explorer.
c) **Ficheros ricos (R)**. Los siguientes formatos de archivo fueron seleccionados tras considerar su relevancia en las actividades académicas y de publicación, y teniendo en cuenta su volumen de uso: Adobe Acrobat (.pdf), Adobe PostScript (.ps), Microsoft Word (.doc) y Microsoft Powerpoint (.ppt). Estos datos fueron extraidos a través de Google, Yahoo Search, y Bing.
d) **Académico (Sc).** Los datos de Google académico incluyendo el número de artículos publicados entre 2006 y 2010 y los de Scimago IR para el periodo 2004-2008.

ranking	University	Det.	Country	Presence Rank*	Impact Rank*	Openness Rank*	Excellence Rank*
38	Ohio State University			98	69	23	33
39	Arizona State University			131	32	84	117
40	National Taiwan University			2	114	22	104
41	Washington University Saint Louis			247	40	178	36
42	Northwestern University			115	57	187	27
43	University of Wisconsin Madison			5288	14	18	22
44	University of Tokyo / 東京大学			125	115	31	24
45	University of Colorado Boulder			169	50	123	85
46	University of California Santa Barbara			106	53	141	90
47	University of Arizona			2180	36	12	64
48	University of Rochester			9	161	24	107
49	University College London			306	71	216	11
50	(2) Universidad Nacional Autónoma de México			48	49	41	302
51	North Carolina State University			789	27	71	161
52	University of Edinburgh			260	61	152	53
53	University of California Irvine			395	48	161	65
54	University of Virginia			3482	9	132	91
55	University of Pittsburgh			942	63	120	19
56	Tsinghua University China / 清华大学			290	41	475	54
57	Boston University			68	82	215	55
58	University of Kentucky			149	129	4	170
59	University of California Davis			1483	51	80	29

Figura 5.5.1. Ranking Web de Universidades del Mundo. Top 12000 Universidades.[2]

A nivel mundial en este ranking, la ***Universidad Nacional Autónoma de México*** se encuentra en el lugar ***número 50 de 12,000 universidades*** superados por la universidad de Sao Paulo Brasil, dónde el primer lugar está ocupado por la Universidad de Harvard, el segundo el Instituto tecnológico de Massachusetts y tercer lugar lo ocupan la Universidad de Stanford. Por otra parte en el top Latino América de 100 universidades, la ***Universidad Nacional Autónoma de México*** se encuentra ***en el segundo lugar,*** superados por la universidad de Sao Paulo[2].

2) http://www.webometrics.info/

Latin America

ranking	World Rank	University	Det.	Country	Presence Rank*	Impact Rank*	Openness Rank*	Excellence Rank*
1	29	(2) Universidade de São Paulo USP	»		33	54	10	78
2	50	(2) Universidad Nacional Autónoma de México	»		48	49	41	302
3	206	Universidade Federal do Rio Grande do Sul UFRGS	»		24	487	48	420
4	235	Universidade Federal de Santa Catarina UFSC	»		254	180	133	705
5	240	Universidade Federal do Rio de Janeiro	»		355	274	175	378
6	248	Universidad de Chile	»		124	299	242	462
7	276	Universidad de Buenos Aires	»		489	355	195	362
8	335	Universidade Estadual de Campinas UNICAMP	»		1424	433	128	324
9	354	Universidade Federal de Minas Gerais UFMG	»		499	507	222	441
10	368	Universidad Nacional de Colombia	»		236	335	181	977
11	373	Universidade Estadual Paulista Júlio de Mesquita Filho	»		587	724	94	393
12	418	Universidade Federal do Paraná	»		181	629	140	868
13	474	Universidade de Brasília UNB	»		612	531	296	841
14	480	Universidade Federal da Bahia	»		459	466	412	958
15	503	Universidad Autónoma Metropolitana	»		915	352	442	1052
16	526	Universidade Federal Fluminense	»		885	486	325	983
17	530	Universidade Federal do Ceará	»		929	394	647	896
18	539	Universidad Nacional de la Plata	»		67	1135	321	700
19	615	Universidad Nacional de Córdoba	»		447	745	602	960
20	646	Universidad de Costa Rica	»		791	569	484	1329
21	665	Universidade Federal de Pernambuco	»		954	871	476	869
22	678	Centro de Investigación y de Estudios Avanzados del IPN CINVESTAV	»		1839	620	1712	573
23	693	Instituto Politécnico Nacional	»		934	1044	323	940
24	757	Pontificia Universidad Católica de Chile	»		1747	1005	1330	519
25	764	Universidad de Guadalajara	»		724	839	293	1670

Figura 5.5.2. Ranking Web de Universidades del Latinoamérica.

5.5.2. Buenas Prácticas para el Servicio Web de la Facultad de Ingeniería

Cuando hablamos de políticas Web nos referimos a los estándares y buenas prácticas que son importantes adoptar en nuestro sitio principal de la Facultad de Ingeniería como sitio Web institucional de tal forma que sea homogéneo e identificado mundialmente como parte de nuestra Universidad Nacional Autónoma de México, su quehacer, la importancia de su gente y todas las contribuciones que tenemos ante la sociedad. Por tal motivo el Consejo Asesor en Tecnologías de la Información y comunicación se dio a la tarea de crear lineamientos, para la creación de sitios Web institucionales, con la finalidad de unificarnos como universidad, así como también realizar las mejores prácticas para la creación de nuestros sitios Web, obteniendo como beneficio, mayor visibilidad en buscadores, usuarios así como en diversos navegadores, menor tiempo de espera en respuesta para los usuarios y la facilidad de uso y acceso a la información y de los servicios que se ofrecen.

5.5.2.1. Facilidad de Uso

La facilidad de uso, es un factor muy importante, pues no sólo hace más fácil la navegación de un sitio para los diferentes tipos de usuarios, sino que también impacta en arquitectura con la que se cuente así como en el desarrollo de contenidos.

- **Navegación**
  - Se debe mostrar de manera clara, la ubicación actual en el árbol de navegación, por ejemplo: /paginas/Carreras/planes2010/ingComputo_Plan.htm
  - El enlace a la página principal del sitio debe estar claramente identificado y todas las páginas deberán estar enlazadas a la de inicio de su entidad y a la de la UNAM.
  - Las secciones más importantes del sitio deberán ser accesibles directamente desde la página principal.
  - Se deberá incluir un mapa de sitio en formato html plano.
  - Incorporar alguna opción de búsquedas por ejemplo Bing, google, joomla o el que sea de su presencia.
  - La estructura de los sitios Web, no debe exceder los cuatro niveles de consulta. Se recomienda incluir, en todo su sitio el regreso a la página inicial y conservar el menú principal en las páginas alternas.
  - Los sitios deberán estar orientados a mostrar información y los servicios que la institución ofrece a la comunidad.

- Verificar que no existan páginas sin enlaces a la página principal del sitio.

Funcionalidad

- El sito deber ser sencillo para todo tipo de usuario desde el principiante hasta el más experto.
- Los servicios ofrecidos deben estar claramente señalados.
- Los contenidos que requieran incorporar extensiones del navegador (*plug-ins*) deberán usarse sólo si agregan alguna funcionalidad que no pueda ser implementada de otra forma, con objeto de evitar que la página no pueda visualizarse en algunos dispositivos o que el usuario requiera instalar software adicional.
- Evitar el uso de frames, en el sitio Web, pues puede causar confusiones cuando se requiere guardar una liga o bien imprimir documentos; se recomienda usar el elemento "div".
- Toda aquella sección relevante, como instrucciones de uso, bases, convocatorias o bien textos relevantes deben contemplar versiones imprimibles.
- El cierre de sesión de un usuario autenticado, debe ser claro en cada página.
- Se sugiere que se agregue al usuario la opción de cambio de tamaño de letra para su mejor lectura.
- Informar a los usuarios si se requiere alguna extensión ("plug-in") para ver cierto contenido siempre y cuando éstas agreguen valor al sitio.

Lenguaje y contenido.

- El diseño gráfico debe brindarle mayor importancia visual a la información y servicios brindados.
- Se recomienda que el contenido del sitio Web siempre se encuentre en la misma página, en el mismo menú y en la misma zona dentro de una página, sin que se cambie el contexto y que el usuario no se sienta fuera del sitio.
- Se recomienda un lenguaje sencillo, claro y directo que permita entender el mensaje de manera fácil y rápidamente. Si es conveniente agregar un glosario de términos.

Claridad arquitectónica.

- El sitio debe diseñarse de acuerdo a los servicios que ofrece a la comunidad universitaria, quedando en segundo término la estructura organizacional.

- El diseño debe ser sencillo, utilizando de forma moderada elementos decorativos, como evitar a las animaciones innecesarias, pues éstas podrían ser molestas, pesadas y pueden ser un distractor para el usuario.
- Los sitios Web Institucionales, no deberán ser desarrollados en flash.
- Evitar que las páginas Web se vean saturadas.
- Utilizar colores que como usuarios permita distinguir fácilmente los enlaces ya visitados. Los enlaces deben ser visibles para evitar se pierdan con el texto.
- Organizar la información incluyendo espacios, colores, bullets, imágenes, textos en negritas, mayúsculas, cursivas de manera que no sea excesiva y además podamos facilitar el entendimiento del contenido al ojo del usuario.
- Con el objetivo de agilizar las lecturas de las páginas Web median la redacción de párrafos breves de manera concreta y concisa; si es el caso de que el texto sea muy extenso, se recomienda que se organicé por medio de un índice ligando a diferentes secciones para que el usuario sepa de manera rápida el contenido del tema y si en verdad es de su interés.
- Cuidar el uso de colores en las páginas Web, donde el fondo sea claro y las letras obscuras, obteniendo mejor legibilidad y eliminando problemas de impresión.
- Los sitios principales deberán evitar el uso de barras de scroll horizontal y vertical, procurando que sea visible toda la página en primera instancia.
- Los sitios Web deben conservar la misma apariencia y funcionalidad en los diferentes navegadores y plataformas.
- La página principal, así como los servicios que puedan tener un público extranjero deben contar, *por lo menos con una versión en idioma Inglés* para facilitar el intercambio de colaboraciones académicas.
- Cuando el sitio Web cuente con versiones en lenguaje diferentes al español, se deberá colocar un enlace de cambio de idioma mediante texto o una imagen para acceder a esté.
- Las traducciones a otros idiomas, deberán tener mucho cuidado en las en contexto, redacción y ortografía.

Ayuda en línea y guías de usuario

- La ayuda e instrucciones, en caso de necesitarse para la navegación del sitio deberán ser fácilmente localizables.

Retroalimentación de sitio Web

- Permitir al usuario proporcionar información de sugerencias mediante correo electrónico o mediante un formulario de comentarios hacia nuestro sitio Web.
- Incluir pantallas de confirmación después que el usuario envía información a través de formularios, de tal forma que validen la recepción o el resultado del envío.

Coherencia

- Utilizar la misma palabra o frase de manera sistemática para describir un tema.
- Cada sitio Web deberá contener la fecha de última actualización.
- El título de la página (etiqueta title) estará acorde con su contenido, por lo tanto debe cuidarse cuando se utilice una plantilla, modificar el título para cada página creada en el sitio Web.

Prevención y corrección de errores

- Los sitios Web que incorporan servicios, deberán incluir mensajes de error de manera visible.
- Los mensajes de error deben redactarse en un lenguaje sencillo y describir que es lo que debe hacer el usuario para solucionarlo o bien proporcionar información de contacto para la obtención de ayuda.

5.5.2.2. Visibilidad

Cuando hablamos de visibilidad, no nos referimos a la apariencia física que podría tener el sitio Web, sino a la visibilidad que el sitio Web puede tener ante los diferentes navegadores Web y su posicionamiento en los mismos.

Entre las recomendaciones que se hacen, tenemos las siguientes:

- Incluir meta-tag con las palabras claves más importantes que den una r4eferencia al contenido del sitio Web, para que los buscadores encuentren el contenido de la página.
- Manejar nombres de archivos cortos y significativos, que describan su contenido, como: https://www.ingenieria.unam.mx/Inscripciones/horarios.html
- Se deben manejar textos alternos o significativos en las imágenes de los sitios Web (atributo “alt” de la etiqueta <img>).En caso de usar mapas de imagen, emplear el elemento map y texto alternativo para las zonas activas en el cliente.

- Utilizar títulos cortos, diferentes y acordes al contenido de cada página del sitio Web (etiquete <title>).
  - Si la información es de alto contenido académico, se recomienda publicarlos en PDF para su consulta fuera de línea.
  - No se recomienda cambiar la dirección URL de las páginas. Los buscadores no efectúan la labor de indexación en forma diaria, arrojando errores de tipo NOT FOUND cuando los usuarios encuentran la referencia anterior en el buscador.
  - Es de gran importancia que siempre exista un mapa de sitio.
  - No deben usarse técnicas computacionales que "inflen" falsamente la presencia de las páginas ante los buscadores, con la finalidad de evitar sanciones que envíen a la página al fondo de las listas de preferencias de los buscadores.

5.5.2.3. Estadísticas

Es de gran importancia, que se cuente con estadísticas de acceso al sitio Web ya que éstas nos permitirán realizar un análisis de las necesidades de nuestros usuarios en la navegación del sitio, permitiéndonos mejorar la comunicación de los servicios que se ofrecen. Para la obtención de éstas estadísticas nos podemos apoyar de las herramientas de Google Analytics, pues podemos obtener la siguiente información de nuestro sitio.

- Número de accesos únicos diarios/al mes/al año.
- Número de archivos html.
- Número de archivos pdf, doc, ppt.
- Número de sitios que hacen link hacia el sitio institucional.
- Número de sitios fuera del dominio UNAM.mx que hacen link hacia el sitio institucional.
- Tiempo de permanencia del usuario en el sitio Web.
- Origen del tráfico.
- Horas de mayor tráfico.
- Meses o temporadas de mayor tráfico.
- Archivos más visitados o descargados.

Un sitio Web institucional deberá contar con estadísticas de acceso. Éstas deberán revisarse de forma continua para modificar las estrategias de comunicación.

5.5.2.4. Lineamientos Estructurales

Se refiere a los elementos o componentes que forman parte de un sitio Web institucional abarcando dos aspectos:

- **Imagen Institucional**

- Todos los sitios Web pertenecientes a la UNAM deberán incorporar el **encabezado institucional,** el cual se encuentra integrado por dos elementos (Figura 5.5.3):
  - La imagen con el escudo de la UNAM, ala Izquierda del encabezado.
  - La imagen a la derecha del encabezado, que puede ser un collage con fotos relativas al servició del que se trate la página Web.

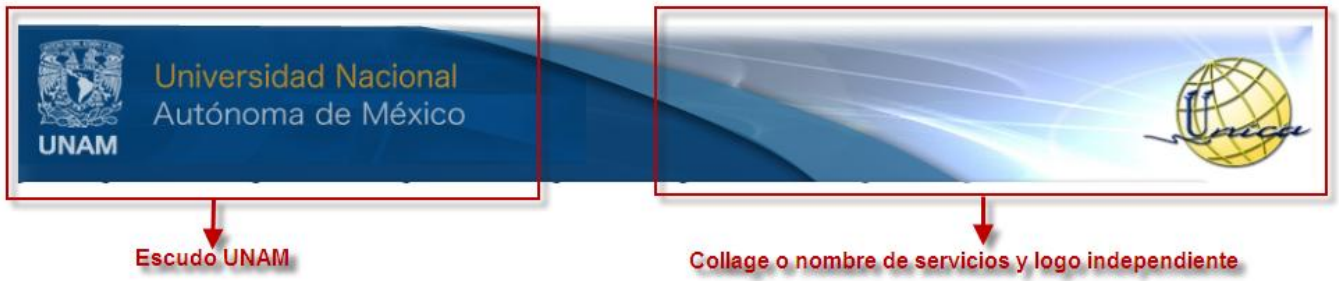


Figura 5.5.3. Encabezado Institucional UNAM.

- Se puede contar con un **segundo encabezado opcional,** que puede ser utilizado para colocar el nombre del servicio y el logotipo de la dependencia ala derecha, pero si el logo y el nombre del servicio fueron incluidos en la imagen de la derecha del encabezado institucional, se puede omitir éste (Figura 5.5.4).

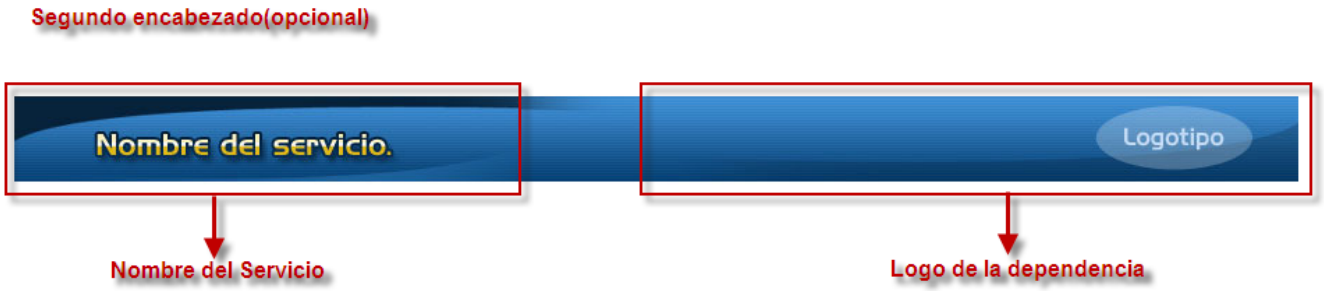


Figura 5.5.4. Segundo Encabezado opcional.

- El diseño gráfico después del encabezado principal es libre.
- La página principal en donde se presenta la oferta de la dependencia debe ajustarse para evitar el uso de scroll horizontal y vertical; es decir, debe visualizarse en una vista de pantalla. El resto de las páginas que conforman el sitio Web pueden tener la extensión necesaria para cada caso.

Otro elemento muy importantes es la leyenda legal, el cual será colocado dentro del pie de página y deberá estar presente para los sitios Web estáticos, para los sitios Web dinámicos es opcional la colocación (Figura 5.5.5).

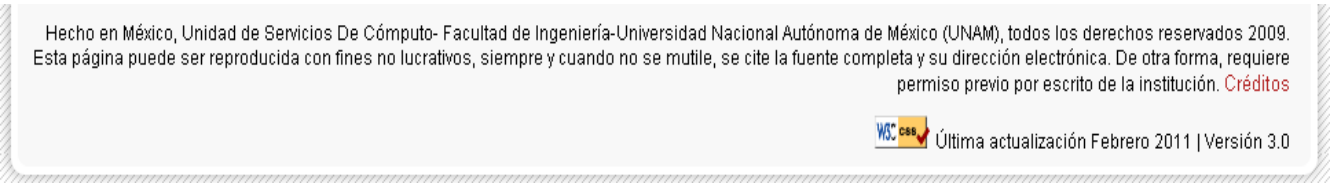


Figura 5.5.5. Leyenda legal de pie de página.

5.6. Disponibilidad

La disponibilidad es la seguridad que tenemos para tener acceso a la información en el momento en que se necesite y a la hora que se necesite para evitar pérdidas o bloqueos en la productividad de ciertos procesos de alguna entidad. Está nos garantiza que los usuarios autorizados tengan acceso a la información y a los recursos relacionados con la misma, todas la veces que se requiera.

Por tal motivo, sabemos que las aplicaciones que se encuentran ejecutándose en el servidor Web de la Facultad de Ingeniería son críticos, por tal motivo, dichas aplicaciones y el sistema requiere de un mayor nivel de disponibilidad.

La alta disponibilidad es un diseño del sistema y su implementación asociada a asegurar un cierto grado de continuidad operacional durante un período de medición dado, como el tiempo.

5.6.1. Riesgos

El hecho de no contar con un sistema disponible puede acarrear pérdidas de productividad y de dinero en muchos de los casos y ya en los más críticos, hasta pérdidas humanas; es por ello q hay que tener un minucioso cuidado de los riesgos de fallos de un comportamiento incorrecto

en nuestros sistemas y ser proactivos para evitar incidentes o para restablecer el servicio en un tiempo aceptable.

Debemos diferenciar dos tipos de interrupciones en nuestros sistemas.

- **Las interrupciones previstas.**
  Las que se realizan cuando paralizamos el sistema para realizar cambios o mejoras en nuestro hardware o software.
- **Las interrupciones imprevistas**
  Las que suceden por acontecimientos imprevistos (como un apagón, un error del hardware o del software, problemas de seguridad, un desastre natural, virus, accidentes, caídas involuntarias del sistema)

Algunas de las causas que pueden ser factores que afectan la disponibilidad de un sistema:

- Causas físicas (de origen natural o delictivo) :
  - Desastres naturales (inundaciones, terremotos, incendios)
  - Ambiente (condiciones climáticas adversas, humedad, temperatura)
  - Fallas materiales
  - Fallas de la red
  - Cortes de energía
- Causas humanas (intencionales o accidentales):
  - Error de diseño (errores de software, aprovisionamiento de red insuficiente)
- Causas operativas (vinculadas al estado del sistema en un momento dado):
  - Errores de software
  - Falla del software

5.6.2. Técnicas para mejorar la disponibilidad

Para favorecer la disponibilidad, debemos prever, detectar y resolver automáticamente los errores de hardware y software antes de que se produzcan las fallas de servicio de tal forma que minimicemos el tiempo de inactividad de nuestros servicios. Dicho de otra manera es hacer todo para crear una infraestructura y unos componentes de aplicaciones fiables y en el caso de producirse errores crear alternativas de recuperación rápida para minimizar e incluso eliminar el tiempo de inactividad.

Sistemas redundantes.

Los sistemas redundantes, son aquellos sistemas de software o hardware de carácter crítico que se quiere que asegurar ante los posibles fallos que puedan surgir pos su uso continuo.

Los componentes redundantes en nuestros sistemas suelen ser:

- **Discos RAID :** Éstos son un conjunto de unidades de discos, que se muestran lógicamente como si fueran uno mismo, distribuyéndose los datos en bandas en dos o más unidades.
- **Tarjeta de red**: Para tratar de garantizar la comunicación con los clientes y poder usar una o más tarjetas como un sólo dispositivo.
- **Fuentes de alimentación:** Las encargadas de brindar la electricidad a nuestros servidores, enrutadores y conmutadores.
- **Sistema de alimentación ininterrumpida (UPS):** Son baterías que se conectan entre el servidor y la fuente de suministro eléctrico, garantizando éste por un tiempo determinado.
- **Generadores eléctricos:** Funcionan generalmente con diesel y se conectan entre los UPS y la red de suministro eléctrico. Estos motores se ponen en marcha cuando el suministro se corta por más de un tiempo determinado.
- **Líneas independientes de suministros:** En los CPD grandes, se suelen tener al menos 2 conexiones diferentes e independientes a la red de suministro eléctrico.
- **Componentes de red:** Aunque se tenga un servidor redundante, si uno de estos componentes fallara, no se llegaría nunca a ese servidor. Para evitar éste fallo, se suelen crear dos caminos diferentes entre los dos componentes de la red.

Sistemas de clusters.

Un clúster es un grupo de múltiples computadoras unidas mediante una red de alta velocidad, de tal forma que el conjunto es visto como una única, más potente que los comunes de

escritorio .Para que un clúster funcione es necesario proveer un sistema de manejo de clúster, el cual se encargué de la interacción del usuario y los procesos que corren para optimizar el funcionamiento del mismo y se dice que es un clúster de alta disponibilidad si a su vez se encuentran monitorizando entre sí. Éstos se dividen en dos clases:

- **Alta disponibilidad de infraestructura:** Si se produce un fallo de hardware en alguna de las máquinas del clúster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios en cualquiera de las otras máquinas del clúster (failover). Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original (failback). Ésta capacidad de recuperación automática de servicios nos garantiza la alta disponibilidad de los servicios ofrecidos por el clúster, minimizando así la percepción del fallo por parte de los usuarios.

- **Alta disponibilidad de aplicación:** Si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas del clúster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del cluster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original. Ésta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdida de datos, y además evita molestias a los usuarios, que no tienen por qué notar que se ha producido un problema.

SAN, NAS, FiberChannel.

SAN (Storage Area Network- red de área de almacenamiento): es una red concebida para conectar servidores, matrices de discos y librerías de soporte. Principalmente, está basada en tecnología fibre channel y más recientemente en iSCSI. Su función es la de conectar de manera rápida, segura y fiable los distintos elementos que la conforman.

Una red SAN es utilizada para transportar datos entre servidores y recursos de almacenamiento. La tecnología SAN permite conectividad de alta velocidad, de servidor a almacenamiento, almacenamiento a almacenamiento, o servidor a servidor. Se puede hacer una copia exacta de los datos a una distancia de 10Km lo que las hace más seguras.

NAS (Network Attached Storage-Almacenamiento conectado a red): es un dispositivo de almacenamiento de red. Un NAS es un servidor de almacenamiento que se puede conectar

fácilmente a la red de una compañía para asistir al servidor de archivos y proporcionar espacio de almacenamiento tolerante a fallas.

Un NAS es un servidor separado que tiene su propio sistema operativo y un software de configuración parametrizado con valores predeterminados que se adaptan a la mayoría de los casos.
Por lo general, posee su propio sistema de archivos que aloja al sistema operativo, así como también una serie de discos independientes que se utilizan para alojar los datos que se van a guardar.

FiberChannel(Canal de fibra): es una tecnología de red Gigabit utilizada principalmente para redes de almacenamiento SAN y para la conexión de Cabinas de Discos DAS, capaz de funcionar sobre cables de fibra óptica y sobre cables de cobre), aunque en la práctica suele ser cableado de fibra óptica (multimodo o monomodo).

5.6.3. Cálculo de la disponibilidad

El cálculo de la disponibilidad en los sistemas, es una medida de la frecuencia con la cual se puede utilizar un sistema. La disponibilidad es el cálculo porcentual del tiempo en que un sistema, se encuentra verdaderamente disponible comparándolo con el tiempo de ejecución total disponible conocido, es decir, la probabilidad del no –fallo en un equipo o servicio.

La fórmula para el cálculo de la disponibilidad es:

$$\%\text{Disponibilidad} = \frac{MTBF}{(MTBF+MTTR)} \times 100$$

Dónde :

- MTBF(Tiempo medio entre errores): Duración media de funcionamiento del sistema antes de que se produzcan errores.

$$\text{MTBF} = \frac{Tiempo\ total\ de\ operacion(hrs)}{Número\ total\ de\ interrupciones}$$

- MTTR(Tiempo medio de recuperación): Tiempo medio necesario para reparar y restaurar el servicio después de que se produzca un error.

Suponiendo que tengamos un sistema critico de alguna manera, el cual tuviera que estar disponible las 24 hrs, los 7 días de la semana , los 365 días del año , calculando que tuviera 4 interrupciones no planeadas, en donde de acuerdo con práctica tardaríamos aproximadamente de 3 a 4 hrs para arreglar y restaurar el servicio, tendríamos que:

$$\text{MTBF}=\frac{24hrs\ x\ 365\ dias}{4\ interrupciones}=2190$$

$$\%\text{Disponibilidad}=\frac{2190}{(2190+3.5)}\times 100=99.84\%$$

Obtendríamos el 99.84% de disponibilidad.

Por otra parte, otra forma conocida de describir la disponibilidad es mediante la regla de los "nueves" (Tabla 5.6.1). La regla de los nueves consiste en una escala utilizada para exponer el tiempo posible de no disponibilidad de un servicio. Hablamos de disponibilidad en término de "nueves", siendo "cinco nueves" el grado de alta disponibilidad; este principio aplica sobre el periodo de un año donde :
365 días * 24 horas/día * 60 minutos/hora * 0.00001 = 5.256 minutos

Disponibilidad	Tiempo de inactividad al año
99.0 %	3 días -15.6 hrs
99.9 %	8 hrs - 0 min – 46 seg
99.99 %	52 min – 34 seg
99.999 %	5 min -15 seg
99.9999 %	32 seg

Tabla 5.6.1 Disponibilidad, mediante regla de los nueves.

CAPÍTULO 6

IMPLEMENTACIÓN Y PRUEBAS

6.1. Transformación del servidor.

La transformación es la etapa final de la reingeniería y se llevará a cabo la solución que se propuso.

El sistema operativo para nuestro servidor CentOS 5.5, donde al elegir la paquetería correspondiente sólo se instalará esencialmente el sistema base y lo necesario como compiladores librerías y aplicaciones necesarias para el funcionamiento de nuestro servidor sin instalar nada en modo gráfico , únicamente modo texto, dejando para la post instalación la instalación de los servicios que se brindaran.

Una vez que se instala el sistema base en el servidor se realizó el hardening correspondiente para fortalecer lo mejor posible nuestro sistema utilizando las mejores prácticas para el hardening como se mencionó en el capítulo 5.

6.2. Instalación de los servicios

6.2.1. PostgreSQL

Como ya mencionamos anteriormente además de brindar el servicio de Web, también se brinda el servicio de bases de datos, por lo tanto comenzaremos por instalar el sistema manejador de bases de datos. Lo primero que haremos es descargar el paquete de código fuente del sitio oficial de postgresql http://www.postgresql.org.

Primero que nada desempaquetamos el paquete e ingresaremos al directorio correspondiente de la siguiente manera.

```
[********]# tar zxvf postgresql-x.x.x.tar.gz
[********]# cd postgresql-x.x.x.
```

Una vez realizado esto se procederá a configurar, compilar e instalar.

```
[********]#./configure --prefix=/path/pgsql
[********]#make
[********]#make install
```

Donde la opción de –pefix indica la dirección de donde queremos que se aloje el servicio en nuestro servidor.

6.2.2. OpenSSL

Obtenemos la última versión estable del paquete openssl-X.X.Xx.tar.gz del sitio oficial de OpenSSL. Una vez descargado, el paquete, ingresaremos en a la carpeta, configuraremos, compilaremos e instalaremos el software.

```
[*********]# cd openssl-X.X.Xx.tar.gz
[*********]#./configure
[*********]#make
[*********]#make intsall
```

6.2.3. Apache

La aplicación de Apache fungirá como el servidor Web y de igual manera se descargara el código fuente del sitio oficial de apache www.apache.org, obteniendo una versión estable del software.

De igual forma que el servicio anterior se desempaqueta, se acede a la carpeta correspondiente, se configura, compila e instala dicho servicio.

```
[********]# tar zxvf httpd-x.x.x.tar.gz
[********]# cd httpd-x.x.x
[********]#./configure --prefix=/path/apache –enable-so –enable-rewrite
[********]#make
[********]#make install
```

Como podemos observar se hacen uso de algunas opciones donde:

--prefix	Indica en que directorio se quiere guardar la aplicación.
--enable-so	Indica que posteriormente podemos instalar objetos dinámicos compartidos
--enable-rewrite	Permite la manipulación de reglas basadas en URL.

6.2.4. Configuración de los certificados

Para la creación y configuración de los certificados de seguridad de nuestro servidor Web, par a esto ingresamos al directorio donde se quedó instalado apache, y una vez dentro creamos los siguientes directorios ssl.csr, ssl.key y ssl.crt.

```
[*********]# cd /path/apache
[*********]# mkdir ssl.csr ssl.key  ssl.crt
```

Una vez realizado lo anterior, comenzaremos a crear los certificados de la siguiente manera:

```
[*********]# openssl req -new > ssl.csr/server.csr
```

Al teclear este comando nos pedirá información necesaria para crear los certificados.

Nos pedirá una clave (frase) la cual servirá para crear los certificados:

```
writing new private key to 'privkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Nos preguntará sobre información general la cual se incorporará a los certificados:

```
You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:CIUDAD_DE_MEXICO
Locality Name (eg, city) []:DISTRITO_FEDERAL
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNICA
Organizational Unit Name (eg, section) []:DROS
Common Name (eg, YOUR name) []:FACULTAD_DE_INGENIERIA,UNAM.
Email Address []:server@cancun.fi-a.unam.mx
```

Posteriormente nos pedirá unos datos extras para mayor seguridad:

```
Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:
An optional company name []:UNIDAD_DE_SERVICIOS_DE_CÓMPUTO_ACADÉMICO
```

Una vez llenado todos los datos necesarios, generaremos la llave privada con un algoritmo de cifrado RSA de 1024 bits:

```
[*********]# openssl genrsa -rand /dev/urandom 1024 > ssl.key/server.key
```

El siguiente paso consistirá en la creación del certificado, como vemos en el orden, se hace referencia a la llave privada que acabamos de generar.

```
[*********]# openssl x509 -days 365 -signkey ssl.key/server.key -in ssl.csr/server.csr -req -out
ssl.crt/server.crt
```

Por último modificamos los permisos de la llave privada que hemos generado.

```
[*********]# chmod 500 ssl.key
[*********]# chmod 400 ssl.key/server.key
```

Ya por último, se modificaran algunos parámetros del archivo de configuración de apache ***httpd.conf*** que se encuentra en /path_apache/conf y ***httpd-ssl.conf*** que se encuentra en /path_apache/conf/extras.

Comenzaremos con el archivo ***httpd.conf.***

Se debe comentar la línea en la cual se indica el puerto de escucha del servidor.

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
#Listen 80
```

Se verifica que las siguientes líneas para el modulo ssl estén descomentadas:

```
# Secure (SSL/TLS) connections
#Include conf/extra/httpd-ssl.conf
#
# Note: The following must must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.
#
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
```

Agregamos al final del archivo las siguientes líneas para que redireccione por el puerto 443 en modo https:

```
RewriteCond %{SERVER_PORT} !^443$
RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [L,R]
```

Ahora nos dirigiremos al archivo https-ssl.conf , donde verificaremos que existan las siguientes líneas y que se encuentren descomentadas:

```
SSLCertificateFile "/path_apache/ssl.crt/server.crt"
SSLCertificateKeyFile "/path_apache/ssl.crt/server.crt"
```

Una vez verificado todo esto, levantaremos el servidor apache y verificaremos que realmente se levante mediante el puerto seguro 443

```
Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2011-08-28 11:05 CDT
Interesting ports on 132.248.54.50:
Not shown: 3161 closed ports
PORT     STATE         SERVICE
22/tcp   open          ssh
443/tcp  open          https
```

6.2.5. Instalación de JDK

Descargar el archivo jdk-versión-SO-arquitectura-rpm.bin del sitio oficial de Java http://www.oracle.com/us/technologies/java/index.html

Cambiar los permisos de ejecución del archivo:

```
[*********]# chmod a+x jdk-versión-SO-arquitectura-rpm.bin
```

Se ejecutará el archivo de la siguiente forma para su instalación:

```
[*********]# ./jdk-6u6-linux-i586-rpm.bin
```

Una vez que quedó instalado, modificar el archivo .bash_profile y agregar la siguiente instrucción:

```
[*********]# export JAVA_HOME=/usr/java/jdk.versionXXX
```

6.2.6. Apache-Tomcat

Descargar el paquete binario de apache-tomcat-version. x.x.x.tar.gz del sitio oficial de apache tomcat http://tomcat.apache.org

Una vez que hemos descargado el paquete, hay que descomprimirlo en el directorio donde queremos alojar el servicio y le cambiamos de nombre al directorio del paquete por uno más corto, en éste caso le pondremos **"tomcat"**

```
[*********]# tar xvfz apache-tomcat-version. x.x.x.tar.gz
[*********]# mv apache-tomcat-version. x.x.x tomcat
```

Modificar el archivo .bash_profile y agregar la siguiente instrucción:

```
[*********]#vi .bash_profile

export PATH=$PATH:/path/tomcat/bin
export CATALINA_HOME=/aath/tomcat
export JAVA_HOME=/usr/java/jdk.version.x.x.x
```

Y ya por último actualizamos los cambios de la siguiente manera:

```
[*********]# . ~/.bash_profile
```

6.2.7. Instalación del conector mod_jk

Descargar el archivo jakarta-tomcat-connectors-x.x.xx-src.tar.gz del sitio oficial http://tomcat.apache.org/connectors-doc, extraer su contenido, configurar e instalar de la siguiente manera:

```
[*********] # tar xvzf jakarta-tomcat-connectors-x.x.xx-src.tar.gz
[*********]# cd jakarta-tomcat-connectors- x.x.xx-src /jk/native
[*********]# ./buildconf.sh
[*********]# ./configure --with-apxs=/path_apache/bin/apxs
[*********]# make
[*********]# make install
```

Donde –whit-apxs, crea una librería compartida para que pueda ser utilizada por apache y pueda conectarse con tomcat.

Verificar que el archivo mod_jk.so se encuentra en el directorio /path_apache/modules con permisos 755.

6.2.8. Conectar Apache con Tomcat

Crear el archivo workers.properties in /path_apache/conf y agregar lo siguiente:

```
#
# This file provides minimal jk configuration properties needed to
# connect to Tomcat.
#
# We define a worked named 'default'
#
workers.tomcat_home=/path/tomcat
workers.java_home=/usr/java/jdk.version.x.x.x
ps=/
worker.list=default

worker.default.port=8009
worker.default.host=localhost
worker.default.type=ajp13
worker.default.lbfactor=1
```

Editar el archivo httpd.conf y agrega la siguiente directiva del módulo mod_jk, colócalo después de la directiva de Hosts Virtuales.

```
#
# Mod_jk settings
#
# Load mod_jk module, se encaraga de cargar el modulo
LoadModule jk_module modules/mod_jk.so
# Where to find workers.properties, esta linea es para que lea el archive de #configuración workers.properties
JkWorkersFile conf/workers.properties
# Where to put jk logs
JkLogFile logs/mod_jk.log
```

```
# Set the jk log level [debug/error/info]
JkLogLevel debug
# 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"
#Esta parte monta los sitios que hacen uso de JSP´s y los sepa interpretar apache
JkMount /UNICA/* default
JkMount /COMUNICACION/* default
```

Agregar ahora todos los Alias de los directorios de Tomcat que queremos que apache reconozca, por default lo hacemos para los directorios de jsp-examples y servlets-examples, pero en éste caso sólo cargaremos los sitios Web que hagan uso de JSP´s. Colócalos en el archivo httpd.conf en el área de Alias:

```
# Static files in the jsp-examples webapp are served by apache
Alias /jsp-examples "/usr/local/tomcat/webapps/jsp-examples/"
<Directory "/usr/local/tomcat/webapps/jsp-examples/">
Options FollowSymLinks
AllowOverride None
Allow from all

# The following line prohibits users from directly access WEB-INF
AllowOverride None
deny from all
</Directory>
<IfModule alias_module>
        #Colocar todos los alias de sitios web en JSP's
        Alias /UNICA  /path/tomcat/webapps
        Alias /COMUNICACION  /path/tomcat/webapps

        ScriptAlias /cgi-bin/ "/sistema/apache/cgi-bin/"

</IfModule>
```

6.2.9. PHP.

Para la instalación de php ya deben estar instalados postresql y apache. Hay que descargar la última versión estable de php-x.x.x.tar.gz del sitio oficial http://www.php.net/

Una vez que lo hemos descargado hay que desempaquetarlo, entrar al directorio, configurar y compilar e instalar de la siguiente manera:

```
[*********] #tar zxvf php-x.x.x.tar.gz
[*********] #cd php-x.x.x
[*********] #./configure --prefix=/sistema/php --with-apxs2=/path_apache/bin/apxs --with-config-file-path=/sistema/php --with-pgsql=/path_postgres/ --enable-tracks-vars
[*********] # make
[*********] # make install
```

6.2.10. Perl.

Ahora procederemos a instalar el módulo de perl, para que podamos hacer uso de este lenguaje de programación en nuestros sitios web. Descargaremos la última versión estable del sitio oficial http://perl.apache.org dicho modulo mod_perl-x.x.x.tar.gz. Una vez que hemos descargado el paquete hay que parar el servicio de apache si está en función, desempaquetar el módulo e ingresar al directorio correspondiente y configurarlo e instalarlo de la siguiente manera:

```
[*********] # /etc/init.d/apache stop
[*********] # tar zxvf  mod_perl-x.x.x.tar.gz
[*********] # cd  mod_perl-x.x.x
[*********] # Perl Makefile.PL MP_APXS = /path_apache /bin/apxs
[*********] #make
[*********] #make test
[*********] #make install
```

Donde MP_APXS es la ruta completa de apxs ejecutable, que se encuentra en el directorio de apache. Si todo salió bien, nos tuvo que agregar el módulo mod_perl.so dentro de los módulos de apache, y deberemos verificar que tenga los permisos de 755, en caso contrario deberemos cambiarlos de la siguiente manera:

```
[*********] #chmod 755 /path_apache/modules/mod_perl.so
```

El paso siguiente en entrar al archivo de configuración de apache httpd.conf y agregamos las siguientes líneas, después de donde cargamos anteriormente el mod_jk para tomcat.

```
#----------------------Indica que se cargue el modulo de perl para apache----------------------
LoadModule perl_module modules/mod_perl.so
#--------Si desea ejecutar código mod_perl 1.0 en mod_perl Server 2.0 permiten la capa de
#----------compatibilidad:
PerlModule Apache2::compat
```

Por último deberemos iniciar el servicio de apache.

```
[*********] # /etc/init.d/apache start
```

6.2.11. Instalación de ModSecurity

Si el servidor apache está levantado o en servicio, detenerlo y posteriormente descargar la última versión estable de modsecurity-apache_x.x.x.tar.gz del sitio oficial http://www.modsecurity.org una vez que descargamos el archivo, hay que descomprimirlo y entrar al directorio de la siguiente manera:

```
[*********]# tar zxvf modsecurity-apache_x.x.x.tar.gz
[*********]#cd modsecurity-apache_x.x.x
```

Una vez hecho lo anterior, se configurará para crear una librería compartida con apache, para que puedan trabajar en conjunto utilizando la opción –prefix –apxs, se compilará y se instalará de la siguiente manera.

```
[*********]# ./configure --with-apxs=/path_apache/bin/apxs
[*********]#make
[*********]#make install
```

Una vez instalado se editará el archivo de configuración de apache httpd.conf, colocar las siguientes líneas, después de las directivas de host virtuales.

```
#-------------Carga libreias necesarias para mod_security------------
 LoadFile /usr/lib/libxml2.so
 LoadFile /usr/lib/liblua5.1.so

#-------Craga el modulo ModSecurity a apache----------
 LoadModule security2_module modules/mod_security2.so
```

Una vez hecho esto, se realizarán algunas configuraciones a mod_security en httpd.conf como las siguientes:

```
# Activamos el Mod_Security
SecFilterEngine On

# Escanear el contenido de la petición POST
SecFilterScanPOST On

# Escanear la respuesta de la petición (si se quiere evitar mostrar ciertos mensajes de error)
SecFilterScanOutput On

# Chequear codificación URL
SecFilterCheckURLEncoding On

# Chequear Codificación Unicode
SecFilterUnicodeEncoding On

#Esta opción debe estar activada solo si la Aplicación utiliza codificación Unicode.
#En cualquier otro caso puede interferir con la operatoria normal del sitio web.
SecFilterCheckUnicodeEncoding Off

#Permitir solo cierto valores de los bytes.
#Hay que tener en cuenta cuales son los caracteres que se utilizan en nuestro sistema.
#En este caso estamos permitiendo todos
SecFilterForceByteRange 1 255

#Loguear peticiones, solo las invalidas, para posterior análisis.
SecAuditEngine RelevantOnly

#Ubicación de los ficheros de logs.
SecAuditLog logs/audit_log

#Por defecto, denegar las peticiones con mensaje de estado â€œ500â€.
SecFilterDefaultAction "deny,log,status:500"

# REGLAS
```

```
# De aquí en adelante irán las reglas que queremos aplicar, para proteger nuestra
# aplicación.

#Con esta directiva cambiaremos la identificación de la versión del Servidor Web.
#Nuestro servidor se identificará como si fuera un Server
SecServerSignature "Server"
SecServerResponseToken Off

# Aquí evitaremos ataques simples de XSS
SecFilter "<(.|\n)+>"
SecFilter "<[[:space:]]*script"

#SQL Injection
SecFilter "delete[[:space:]]+from"
SecFilter "insert[[:space:]]+into"
SecFilter "select.+from"
SecFilter ".*['%;\"]+.*"

#Controlo el Campo Nombre del Formulario para que no sea mayor a 6 caracteres,
#protección contra #Buffer Overflow, si nuestro parómetro "nombre" es vulnerable.
SecFilterSelective ARG_nombre ".{6,}" "redirect:http://www.google.es"

#Nuestra aplicación es muy compleja y depende del RegisterGlobals, pero hay un
#formulario de autenticación que se puede saltar enviando la variable valido=1 o ok=1
SecFilterSelective "ARG_valido|ARG_ok" "1"

#Controlamos que se envíen los dos cabezeras (HTTP_USER_AGENT y HTTP_HOST) en las
#peticiones generalmente los atacantes y algunas herramientas de escaneo
#no enviaran estas cabezeras.
SecFilterSelective "HTTP_USER_AGENT| HTTP_HOST" "^$"

# Prohibimos subir ficheros
SecFilterSelective "HTTP_CONTENT_TYPE" multipart/form-data)"
</IfModule>
```

Por último se iniciará apache y listo.

6.2.12. Ossec

Para instalar nuestro sistema detector de intrusos, de igual forma debemos descargarlo del sitio oficial http://www.ossec.net/ , desempaquetarlo, ingresar al directorio y una vez dentro correr el programa que se llama install.sh, el cual nos hará una serie de preguntas como se muestra a continuación.

[*********]#tar -zxvf ossec-hids-1.1.tar.gz
[*********]# cd ossec-hids-X.X
[*********]#./install.sh
[*********]## ./install.sh
Se pregunta por el tipo de idioma que erigiremos para la instalación del ids, únicamente escribimos "es".
(en/br/de/es/fr/it/jp/pl/ru/sr/tr) [en]: es
Posteriormente se muestra la información del servidor y del usuario quien procede la instalación, en este caso damos enter para continuar.

1- Qué tipo de instalación Usted desea (servidor, agente, local ayuda)?local

En este caso se elige "local" para que el servicio únicamente tenga alertas de nuestro host.

2- Configurando las variables de entorno de la instalación.

- Eliga donde instalar OSSEC HIDS [/var/ossec]:

Elegimos por default /var/ossec dando enter.

3- Configurando el sistema OSSEC HIDS.

3.1- Desea recibir notificación por correo electrónico? (s/n) [s]:

Elegimos "s" dando enter
- Cuál es vuestra dirección de correo electrónico? xxxxx@xxxx.unam.mx
- Cuál es la dirección nombre de vuestro servidor de correo SMTP? cancun.fi-a.unam.mx

3.2- Desea Usted agregar el servidor de integridad del sistema? (s/n) [s]:

Damos enter en esta opción.

3.3- Desea Usted agregar el sistema de detección de rootkit? (s/n) [s]:

Elegimos nuevamente opción por default.
- Desea Usted habilitar respuesta activa? (s/n) [s]:

Desea Usted habilitar la respuesta desechar en el Firewall? (s/n) [s]:
- Desea Usted agregar más IPs a la lista blanca? (s/n)? [n]: s

Agregamos las IPs para que no sean bloqueadas por el firewall

- IPs (lista separada por blancos): xxx.xxx.xxx.xx1 xxx.xxx.xxx.xx2 xxx.xxx.xxx.xxN

Por último aceptamos para comenzar la compilación e instalación de ossec. Ahora sólo iniciamos el servicio mediante el siguiente comando:

[*********]# /var/ossec/bin/ossec-control start

3.6- Estableciendo la configuración para analizar los siguientes registros:

-- /var/log/messages
-- /var/log/secure
-- /var/log/maillog

- Si Usted deseara monitorear algún otro registro, solo tendrá que editar el archivo ossec.conf y agregar una nueva entrada de tipo localfile.
Cualquier otra pregunta de configuración podrá ser respondida visitándonos en línea en http://www.ossec.net .

--- Presione ENTER para continuar ---
- El sistema es Redhat Linux.
- Init script modificado para empezar OSSEC HIDS durante el arranque.

- Configuración finalizada correctamente.

- Para comenzar OSSEC HIDS:
/var/ossec/bin/ossec-control start

- Para detener OSSEC HIDS:
/var/ossec/bin/ossec-control stop

- La configuración puede ser leída modificada en /var/ossec/etc/ossec.conf

Gracias por usar OSSEC HIDS.
Si tuviera Usted alguna duda, sugerencia Ã³ haya encontrado algún desperfecto, contáctese con nosotros a contact@ossec.net o usando nuestra lista pública de correo en ossec-list@ossec.net

Más información puede ser encontrada en http://www.ossec.net

--- Presione ENTER para finalizar. ---
(Tal vez encuentre más información a continuación).

6.3. Pruebas

6.3.1. Pruebas de Benchmarking

Una vez que ya hemos instalado todo nuestro software de aplicación, será el momento de realizar las pruebas de benchmarking

6.3.1.1. Unixbench

Para poder instalar ésta herramienta, es necesario que contemos con los compiladores de C, C++ o bien gcc o gcc++. Primero que nada descargaremos el software del sitio http://www.hermit.org/Linux/Benchmarking, una vez descargado el software, hay que desempaquetarlo, compilarlo y ejecutarlo de la siguiente manera:

```
[*********]# tar zxvf  unixbench-x.x.x.tar.gz
[*********]# cd unixbench-x.x.x
[*********]#make
[*********]#./Run
```

Una vez que haya finalizado la ejecución del benchmark dentro del mismo directorio, encontraremos una carpeta llamada result, la cual contendrá tres archivos que contendrán información del resultado del benchmark, es tos tres archivos son (Figura 6.3.1):

- ***report***: el archivo que contendrá los resultados finales.
- ***log***: un archivo que contendrá información detallada sobre los resultados de cada una de las pruebas realizadas, facilitando por ejemplo información relativa a la ejecución de cada una de las partes de las pruebas o devolviendo los resultados empleando diferentes unidades de medición.
- ***times***: el archivo donde se almacenan todos los tiempos empleados para ejecutar cada una de las repeticiones de cada prueba del *benchmark*.

```
[r[redacted] UnixBench]# ./Run
make all
make[1]: Entering directory `/root/sw/benchmarking/UnixBench'
Checking distribution of files
./pgms  exists
./src  exists
./testdir  exists
./tmp  exists
./results  exists
make[1]: Leaving directory `/root/sw/benchmarking/UnixBench'
sh: 3dinfo: command not found

   #    #  #    #  #  #    #          #####   ######  #    #   ####   #    #
   #    #  ##   #  #   #  #           #    #  #       ##   #  #    #  #    #
   #    #  # #  #  #    ##            #####   #####   # #  #  #       ######
   #    #  #  # #  #    ##            #    #  #       #  # #  #       #    #
   #    #  #   ##  #   #  #           #    #  #       #   ##  #    #  #    #
    ####   #    #  #  #    #          #####   ######  #    #   ####   #    #

   Version [redacted]                  Based on the Byte Magazine Unix Benchmark

   Multi-CPU version                   Version[redacted]revisions by Ian Smith,
                                       Sunnyvale, CA, USA
   January 13, 2011                    johantheghost at yahoo period com


1 x Dhrystone 2 using register variables  1 2 3 4 5 6 7 8 9 10

1 x Double-Precision Whetstone  1 2 3 4 5 6 7 8 9 10

1 x Execl Throughput  1 2 3

1 x File Copy 1024 bufsize 2000 maxblocks  1 2 3

1 x File Copy 256 bufsize 500 maxblocks  1 2 3

1 x File Copy 4096 bufsize 8000 maxblocks  1 2 3

1 x Pipe Throughput  1 2 3 4 5 6 7 8 9 10

1 x Pipe-based Context Switching  1 2 3 4 5 6 7 8 9 10

1 x Process Creation  1 2 3

1 x System Call Overhead  1 2 3 4 5 6 7 8 9 10

1 x Shell Scripts (1 concurrent)  1 2 3

1 x Shell Scripts (8 concurrent)  1 2 3

========================================================================
   BYTE UNIX Benchmarks (Version [redacted])

   System: [redacted]fi-a.unam.mx: GNU/Linux
   OS: GNU/Linux -- [redacted]-- #1 SMP Fri Apr 2 14:58:35 EDT [redacted]
   Machine: [redacted]
   Language: en_US.utf8 (charmap="UTF-8", collate="UTF-8")
```

Figura 6.3.1.Ejecución de Unixbench.

6.3.1.2. Lmbench

Descargamos la última versión estable de lmvench-.x.x.x del siguiente sitio http://switch.dl.sourceforge.net y una vez descargado, lo desempaquetamos y entramos a la carpeta generada y una vez dentro lo ejecutamos de la siguiente manera:

```
[*********]#tar zxvf lmbench-x.x.x
[*********]#cd lmbench-x.x.x
[*********]#make result
```

Para obtener los resultados de la ejecución con un formato amigable, lo haremos con el siguiente comando:

```
[*********]#make LIST=[carpeta que contiene los archivos con los resultados]/*
```

Si nos interesa guardar dichos resultados en un archivo de texto, simplemente tendremos que sustituir la utilización del comando anterior por el siguiente comando:

```
[*********]# make LIST=[carpeta que contiene los archivos con los resultados]/* >> [nombre de fichero]
```

6.3.1.3. Iperf

Para poder realizar el benchmarking con Iperf debe ser ejecutado en dos servidores; necesitamos tener en ambos equipos instalado alguno de los sistemas operativos que están siendo empleados en las pruebas, aunque en el servidor se puede emplear prácticamente cualquier otro sistema operativo derivado de Unix que cumpla los requisitos para poder ejecutar Iperf.

Primero que nada, hay que descargar Iperf, tanto en el servidor como en el cliente del sitio http://switch.dl.sourceforge.net/sourceforge, donde el cliente es el equipo que se quiere poner a prueba y una vez descargada, se desempaqueta e ingresamos al directorio para su instalación.

```
[*********]#tar zxvf  iperf-x.x.x.tar.gz
[*********]#cd iperf-x.x.x
[*********]#./configure
```

```
[*********]#make
[*********]#make install
```

Una vez instalado en ambos equipos, se ejecutará, tanto en el servidor como en el cliente. La ejecución en modo servidor, se ejecuta de la siguiente manera:

```
[*********]# iperf -s
```

Y para ejecutar en modo cliente lo haremos con el siguiente comando:

```
[*********]# iperf –c [IP del servidor]
```

6.3.2. Resultados de las pruebas de Benchmarking

La siguiente tabla muestra los parámetros tomados en cuenta para la realización del benchmarking y la comparación entre el equipo virtualizado con CentOS y el equipo físico con fedora (Tabla 5.6.1).

Tabla 5.6.1. Resultados de Benchmarking.

		Equipo Virtual CentOS	Equipo Físico Fedora
Rendimiento del procesador			
Operaciones aritméticas	Short (lps)	2511236.5	176451.7 l
	Int (lps)	2534245.7	168494.3
	Long (lps)	2535029.5	168549.9
	Float (lps)	1113375.4	169681.6
	Doublé (lps)	1113958.3	169454.3 l
Rendimiento de memoria			
Latencia de lecturas	Cache L1 (ns)	1.1460	4.6210
	Cache L2 (ns)	5.75	20.1
	Memoria principal (ns)	204.2	161.7
Ancho de banda	Lectura (Mb/s)	1897	355
	Escritura (Mb/s)	758.4	128.2
Rendimiento de Red			
Ancho de banda	TCP (Mb/s)	94.7	94.7
	UDP (Mb/s)	6135	6135
Rendimiento del almacenamiento físico			
Ancho de banda	Lectura (Kb/s)	931726,0	163298,0
	Escritura (Kb/s)	541774,0	57400,0

Latencia del sistema de archivos	Crear (µs)	30.5	254.1
	Borrar (µs)	14	103.7
Rendimiento de Funciones típicas del Sistema Operativo			
	Llamadas al sistema (µs)	0.50	0.61
	Creación de procesos (µs)	2658	9823
Cambios de contexto	2 procesos de 0 Kb (µs)	6.54	8.24
	8 procesos de 16 Kb (µs)	9.03	34.8
	8 procesos de 64 Kb (µs)	8.78	180.2
	2 procesos de 16 Kb (µs)	7.76	10.8
	2 procesos de 64 Kb (µs)	7.07	43.1
	16 procesos de 16 Kb (µs)	9.15	57.8
	16 procesos de 64 Kb (µs)	8.73	181.6

Como podemos observar en la tabla anterior, tenemos que en cuanto al rendimiento del procesador y el número de operaciones aritméticas que puede realizar en lps (loops por segundo), es mucho mayor en el equipo con CentOS que con Fedora y se nota la gran diferencia entre uno y otro.

El segundo rubro es en cuanto al rendimiento de memoria en cache de primer y segundo nivel, donde nos deja ver que la latencia es mucho menor en el equipo virtualizado que en el físico y que la cache de primer nivel es mucho más rápida que la de segundo nivel y esto es correcto ya que ésta suele estar integrada en el procesador y por lo tanto tiene mayor acceso mucho más rápido por parte del CPU. En memoria principal es mucho más lenta con una diferencia de 42.5 nano segundos. En cuanto al ancho de banda se observa que es mejor en CentOS que en Fedora.

En cuanto al rendimiento de red ambos equipos están muy parejos.

Rendimiento de almacenamiento físico tenemos mayor ancho de banda en CentOS tanto para lectura como escritura aunque en escritura no varía por mucho. En cuanto a latencia en el sistema de archivos es mucho más rápido CentOS tanto para crear y borrar archivos.

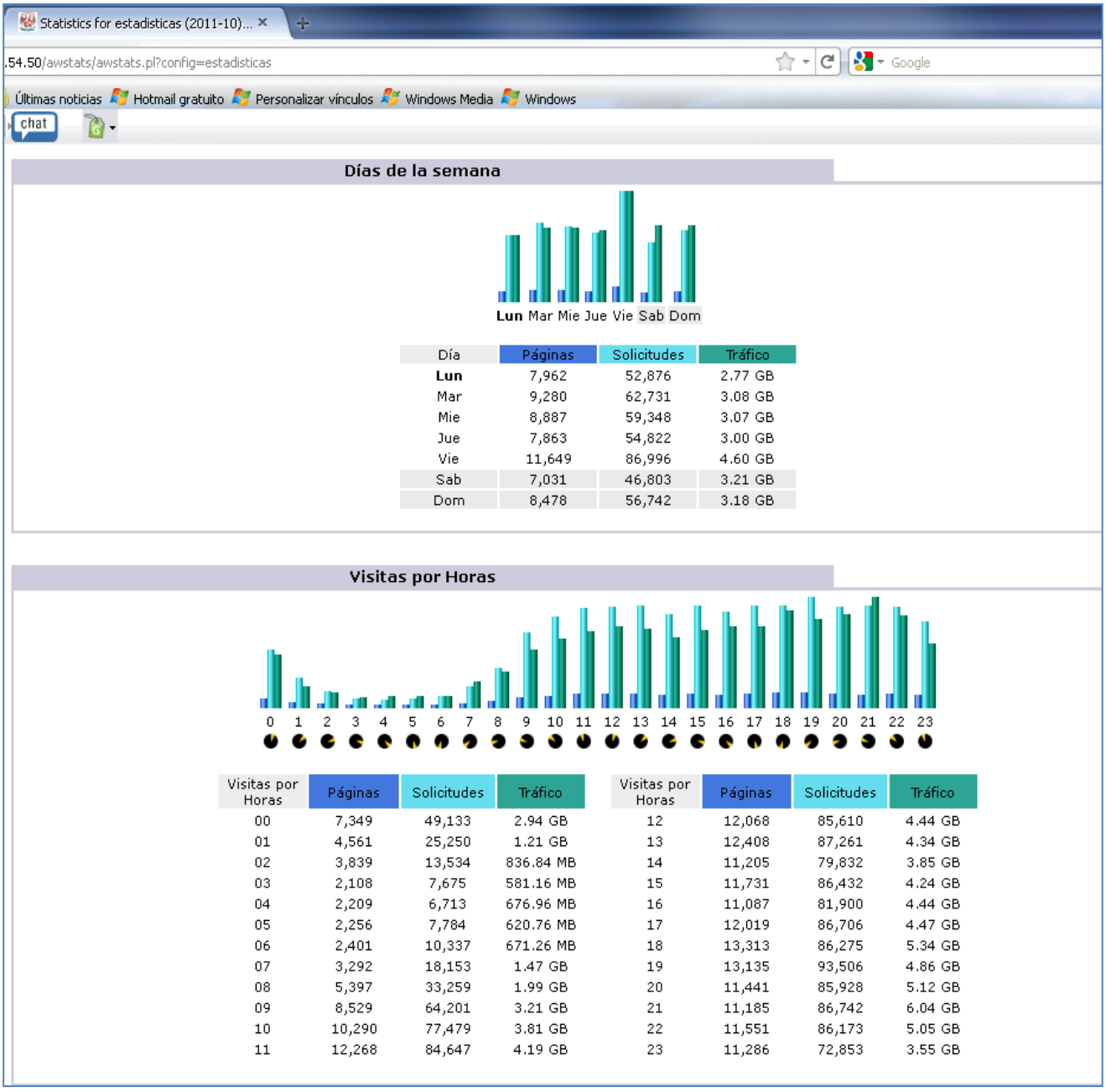
Ya por ultimo tenemos el análisis del rendimiento de tareas comunes que se pueden realizar en sistemas operativos tipo Unix, tenemos como primer parámetro de medición las llamadas al sistema creación de procesos, donde sigue siendo superado en menor tiempo el equipo virtual, al igual que en los cambios de contexto (tiempo que se necesita para guardar el estado de un proceso y restaurara el estado de otro).

En conclusión, podemos decir que el equipo virtualizado cuenta con mayor performance que el equipo físico trabajando y distribuyendo sus tareas de la manera más optimizada y en cuanto al sistema operativo el rendimiento fue mejor con CentOS realizando , más eficientemente las tareas comunes del mismo.

6.4. Estadísticas

Es de gran importancia contar con estadísticas sobre la utilización de nuestro servidor Web así como de la utilización y manejo de los usuarios, pues dichos datos pueden ser de gran relevancia para la administración y utilización del servicio para los usuarios, así como para llevar acabo buenas prácticas para la mejor visibilidad de nuestros sitios Web, y que alcanza una mayor audiencia que otros me dios de comunicación científica, hablando de sitios Web institucionales.

Para la obtención de las estadísticas nos ayudamos de dos herramientas AWStats y de Google Analytics. El primero nos da un análisis detallado, mediante gráficos en barras que pueden ser visualizados desde cualquier navegador Web, donde podemos visualizar los días del mes o de la semana con mayor carga de trabajo, cuantas visitas tenemos por hora, los tipos de archivos que se manejan y son solicitados con mayor frecuencia, las páginas más solicitadas, entre otros (Figura 6.4.1, 6.4.2, 6.4.3).



Día	Páginas	Solicitudes	Tráfico
Lun	7,962	52,876	2.77 GB
Mar	9,280	62,731	3.08 GB
Mie	8,887	59,348	3.07 GB
Jue	7,863	54,822	3.00 GB
Vie	11,649	86,996	4.60 GB
Sab	7,031	46,803	3.21 GB
Dom	8,478	56,742	3.18 GB

Visitas por Horas	Páginas	Solicitudes	Tráfico	Visitas por Horas	Páginas	Solicitudes	Tráfico
00	7,349	49,133	2.94 GB	12	12,068	85,610	4.44 GB
01	4,561	25,250	1.21 GB	13	12,408	87,261	4.34 GB
02	3,839	13,534	836.84 MB	14	11,205	79,832	3.85 GB
03	2,108	7,675	581.16 MB	15	11,731	86,432	4.24 GB
04	2,209	6,713	676.96 MB	16	11,087	81,900	4.44 GB
05	2,256	7,784	620.76 MB	17	12,019	86,706	4.47 GB
06	2,401	10,337	671.26 MB	18	13,313	86,275	5.34 GB
07	3,292	18,153	1.47 GB	19	13,135	93,506	4.86 GB
08	5,397	33,259	1.99 GB	20	11,441	85,928	5.12 GB
09	8,529	64,201	3.21 GB	21	11,185	86,742	6.04 GB
10	10,290	77,479	3.81 GB	22	11,551	86,173	5.05 GB
11	12,268	84,647	4.19 GB	23	11,286	72,853	3.55 GB

Figura 6.4.1. Estadísticas Web por día de la semana y por hora.

Servidores (Top 10) - Lista completa - Última visita - Dirección IP no identificada

Servidores : 0 Conocidos, 58,853 Desconocidos (Dirección IP desconocida) 45,965 Visitantes distintos	Páginas	Solicitudes	Tráfico	Última visita
::1	7,213	7,213	0	24 Oct 2011 - 09:35
132.248.139.159	6,012	70,246	958.67 MB	24 Oct 2011 - 09:35
67.195.115.151	5,540	8,697	670.27 MB	24 Oct 2011 - 06:55
132.248.63.243	3,533	9,869	243.13 MB	24 Oct 2011 - 09:35
201.141.216.64	3,022	3,028	32.44 MB	18 Oct 2011 - 05:45
178.79.135.178	2,979	2,979	0	24 Oct 2011 - 09:35
178.79.135.13	2,975	2,975	0	24 Oct 2011 - 09:33
66.249.71.8	2,898	5,834	698.17 MB	22 Oct 2011 - 06:34
85.17.29.107	2,536	2,537	37.60 MB	19 Oct 2011 - 02:31
201.141.218.64	2,380	2,384	25.51 MB	19 Oct 2011 - 00:20
Otros	167,840	1,301,621	75.27 GB	

Visitas de Robots/Spiders (Top 10) - Lista completa - Última visita

0 robots distintos*	Solicitudes	Tráfico	Última visita

* Los Robots mostrados aquí son dados por solicitudes o tráfico "no visto" por los visitantes, y no se incluyen en otros apartados.

Duración de las visitas

Número de visitas: 60,198 - Media: 258 s	Número de visitas	Porcentaje
0s-30s	44,894	74.5 %
30s-2mn	3,590	5.9 %
2mn-5mn	2,813	4.6 %
5mn-15mn	3,412	5.6 %
15mn-30mn	2,104	3.4 %
30mn-1h	1,995	3.3 %
1h+	1,015	1.6 %
Desconocido	375	0.6 %

Figura 6.4.2. Estadísticas Web desde las distintas direcciones ips que nos visitan y de la duración de las mismas.

Downloads (Top 10) - Lista completa

Downloads: 4902	Solicitudes	206 Solicitudes	Tráfico	Tamaño medio
/~jkuri/Apunt_Planeacion_internet/TEMAIV.1.pdf	921	175	43.48 MB	40.62 KB
/~jkuri/Apunt_Planeacion_internet/TEMAII.1.pdf	884	174	47.21 MB	45.70 KB
/~jkuri/Apunt_Planeacion_internet/TEMAII.5.pdf	851	187	38.40 MB	37.88 KB
/Trabajo%20en%20Equipo,%20M%F3nica%20Mu%F1oz%20C,%20Jonathan%20H...	607	128	696.41 MB	970.24 KB
/~jkuri/Apunt_Planeacion_internet/TEMAVI.5.pdf	469	132	28.97 MB	49.36 KB
/~jkuri/Apunt_Planeacion_internet/TEMAII.4.pdf	388	100	19.21 MB	40.31 KB
/~luiscr/pce_1427/ProcconstrACERO.ppt	332	95	2.79 GB	6.69 MB
/Motivaci%F3n,%20Martha%20Ocampo%20S.,%20Sergio%20Zepeda%20A,%20...	328	55	54.21 MB	144.95 KB
/Sexto%20Habito,%20La%20sinergia,%20Jose%20Luis%20Rodriguez,%202...	318	43	329.15 MB	933.64 KB
/~luiscr/pce_1427/curado.ppt	297	104	119.28 MB	304.60 KB

Páginas-URLs (Top 10) - Lista completa - Página de entrada - Salida

5,092 páginas diferentes	Accesos	Tamaño medio	Página de entrada	Salida
/	28,637	15.05 KB	14,973	9,116
/~materiacfc/CCostos.html	13,704	84.65 KB	11,170	11,108
/paginas/informacion/directorio/acads_busqueda.php	9,185	14.59 KB	371	518
*	7,214		40	39
/~fjgv/ingenieria_de_transito.html	5,933	10.89 KB	10	10
/~materiacfc/clasif_cuentas.html	5,412	79.93 KB	4,334	4,310
/UNICA/	2,982	2 Bytes	16	12
/COMUNICACION/./noticias/	2,865	61 Bytes	11	16
/paginas/alumnos.htm	2,826	11.89 KB	229	697
/~unica/UNICAPHP/	2,476	14.34 KB	31	32
Otros	125,694	28.32 KB	29,013	33,965

Figura 6.4.3. Estadísticas Web de los archivos más descargados del servidor y de los sitios más vistos dentro del servidor.

Por otra parte Google Analytics que nos da un informe más detallado en cuanto al uso de los sitios Web, desde las resoluciones de pantalla, en que navegadores visualizan nuestro sitio, en que motores de búsqueda nos encuentran, así como las palabras clave con las que nos identifican, todo éste tipo de parámetros nos ayudara a tomar medidas al desarrollar nuestro portal para aumentar la visibilidad como sitio Web institucional (Figura 6.4.4, 6.4.5, 6.4.6 y 6.4.7).

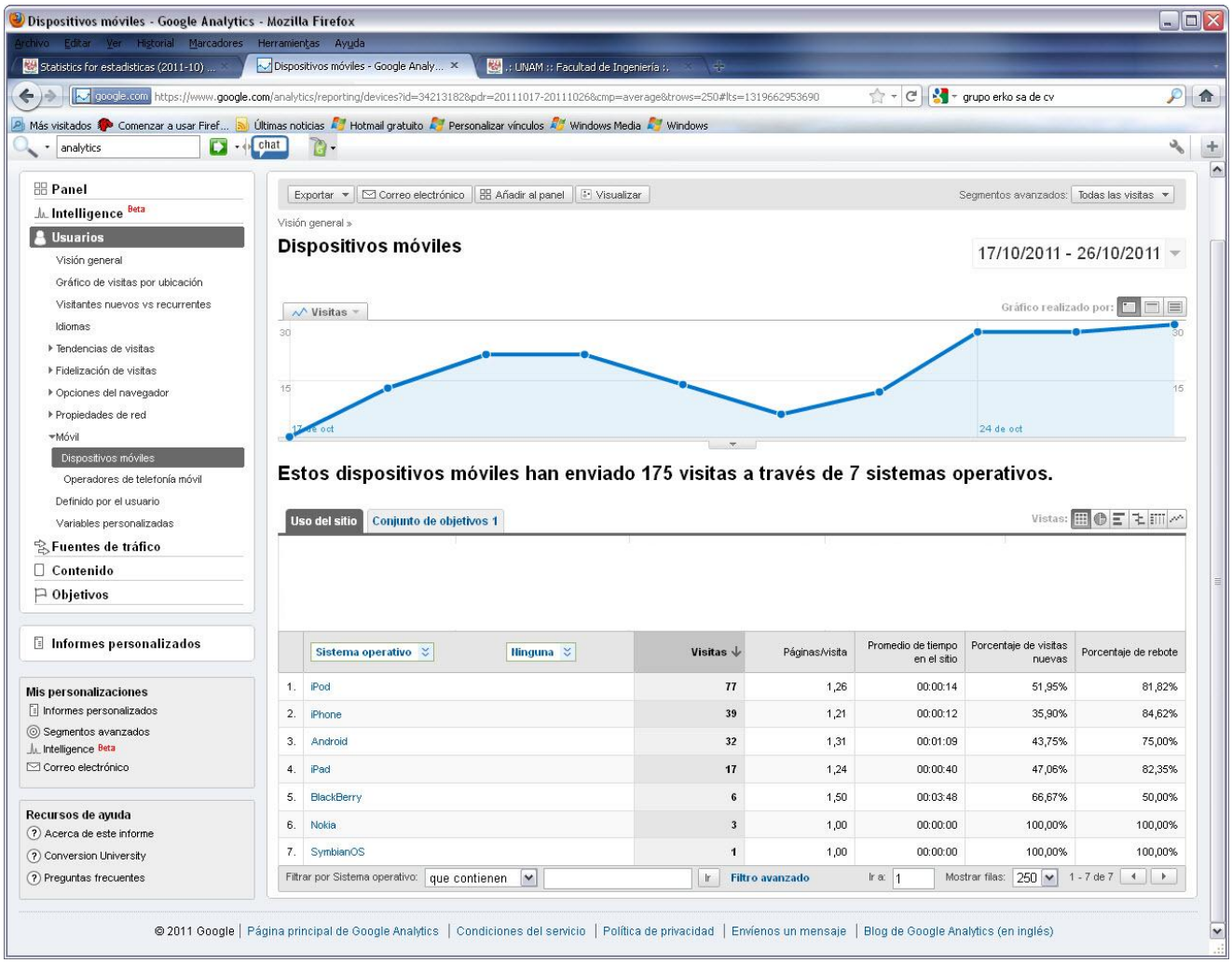


Figura 6.4.4. Estadísticas de visitas a través de dispositivos móviles.

	Sistema operativo	Visitas	Visitas
1.	Windows	2.340	86,54%
2.	Macintosh	132	4,88%
3.	iPod	76	2,81%
4.	Linux	42	1,55%
5.	iPhone	39	1,44%
6.	Android	32	1,18%
7.	iPad	17	0,63%
8.	(not set)	16	0,59%
9.	BlackBerry	6	0,22%
10.	Nokia	3	0,11%
11.	SymbianOS	1	0,04%

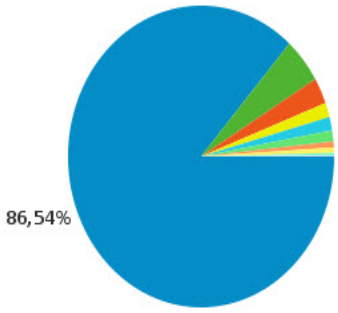


Figura 6.4.5. Estadísticas de sistemas operativos más utilizados.

	Palabra clave	Visitas	Páginas/visita	Promedio de tiempo en el sitio	Porcentaje de visitas nuevas	Porcentaje de rebote
1.	facultad de ingenieria	325	1,44	00:01:17	39,69%	73,23%
2.	facultad de ingenieria unam	218	1,52	00:01:06	40,37%	71,56%
3.	fi unam	189	1,24	00:01:02	30,16%	82,54%
4.	fi	168	1,21	00:00:25	38,10%	86,90%
5.	ingenieria unam	164	1,34	00:00:33	42,68%	81,10%
6.	ingenieria	86	1,28	00:00:25	36,05%	80,23%
7.	facultad de ingeniería unam	30	1,63	00:00:55	50,00%	60,00%
8.	unam ingenieria	24	1,79	00:02:32	50,00%	70,83%
9.	facultad de ingeniería	19	1,42	00:01:27	36,84%	68,42%
10.	ingenieria.unam	16	1,19	00:00:28	43,75%	93,75%
11.	ingenieria.unam.mx	16	1,06	00:00:03	31,25%	93,75%

Figura 6.4.6. Estadísticas Palabras claves por las que nuestro sitio es buscado.

	Navegador	Visitas	Visitas
1.	Internet Explorer	1.064	39,35%
2.	Chrome	650	24,04%
3.	Firefox	632	23,37%
4.	Safari	263	9,73%
5.	Opera	45	1,66%
6.	Android Browser	26	0,96%
7.	Opera Mini	11	0,41%
8.	Mozilla Compatible Agent	4	0,15%
9.	BlackBerry8520	3	0,11%
10.	RockMelt	2	0,07%
11.	BlackBerry8350i	1	0,04%
12.	BlackBerry9300	1	0,04%
13.	BlackBerry9550	1	0,04%
14.	IE with Chrome Frame	1	0,04%

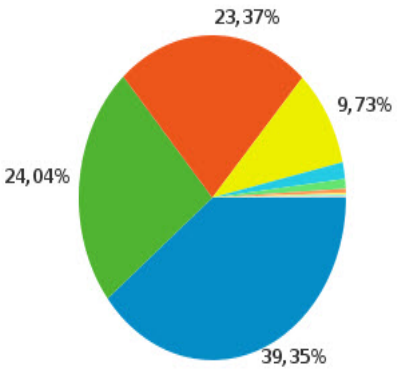


Figura 6.4.7. Estadísticas de navegadores usados.

6.5. Administración.

La administración de nuestro servidor es una tarea de gran importancia y que se debe de llevar a cabo con mucho cuidado, pues debemos estar siempre actualizados y contar con personal que se encuentre debidamente capacitado en cuanto a los servicios con los que se cuentan, los cuales realizaran las tareas pertinentes de manera periódica para que nuestros servicios se encuentren siempre disponibles y en las mejores condiciones, en nuestro caso para hacer más amigable la administración se contara con la herramienta de webmin.

6.5.1. Webmin

Para instalar nuestro software de administración, debemos entrar al sitio oficial de Webmin http://www.webmin.com/download.html y descarga la versión estable del mismo en su versión .tar.gz; una vez que hemos descargado el paquete, hay que desempaquetarlo e ingresar al directorio generado donde correremos el script que se llama setup.sh de la siguiente manera:

```
[*********]# tar zxvf  webmin-x.xxx.tar.gz
[*********]#cd webmin-x.xxx.tar.gz
[*********]#./setup.sh
```

Una vez que hemos arrancado el script nos hará una serie de preguntas en torno a la configuración del mismo como se muestra a continuación.

```
**********************************************************************
*           Welcome to the Webmin setup script, version 1.570        *
**********************************************************************
Webmin is a web-based interface that allows Unix-like operating
systems and common Unix services to be easily administered.

Installing Webmin in /root/sw/webmin-1.570 ...
**********************************************************************
Webmin uses separate directories for configuration files and log files.
Unless you want to run multiple versions of Webmin at the same time
you can just accept the defaults.

Config file directory [/etc/webmin]:
Log file directory [/var/webmin]:
**********************************************************************
Webmin is written entirely in Perl. Please enter the full path to the
Perl 5 interpreter on your system.

Full path to perl (default /usr/bin/perl):

Testing Perl ...
Perl seems to be installed ok
**********************************************************************
Operating system name:    CentOS Linux
Operating system version: X.X
```

Una vez que identifica sobre qué plataforma estamos trabajando nos indica el puerto de default donde trabajara el servidor web de administración y nos pedirá un usuario y contraseña para acceder.

```
**********************************************************************
Webmin uses its own password protected web server to provide access
to the administration programs. The setup script needs to know :
 - What port to run the web server on. There must not be another
   web server already using this port.
 - The login name required to access the web server.
 - The password required to access the web server.
 - If the webserver should use SSL (if your system supports it).
 - Whether to start webmin at boot time.
```

```
Web server port (default 10000): xxxx
Login name (default admin): root
Login password:
Password again:
Use SSL (y/n): y
Start Webmin at boot time (y/n): n
*************************************************************
Creating web server config files..
……
……..

Running postinstall scripts ..
PID file /var/webmin/miniserv.pid does not exist
..done

Enabling background status collection ..
PID file /var/webmin/miniserv.pid does not exist
..done

Attempting to start Webmin mini web server..
Starting Webmin server in /root/sw/webmin-1.570
Pre-loaded WebminCore
..done
*************************************************************
Webmin has been installed and started successfully. Use your web
browser to go to

  http://localhost.localdomain:xxxx/
and login with the name and password you entered previously.
```

Ya por último sólo nos pregunta si deseamos usar SSL y esto va a ser siempre y cuando y tengamos instalada la librería Open SSL, le decimos que sí para que todos los datos que enviemos a través de webmin cuando estemos administrando nuestro servidor viajen en la red cifrados y no puedan ser leídos de forma directa (Figura 6.5.1).

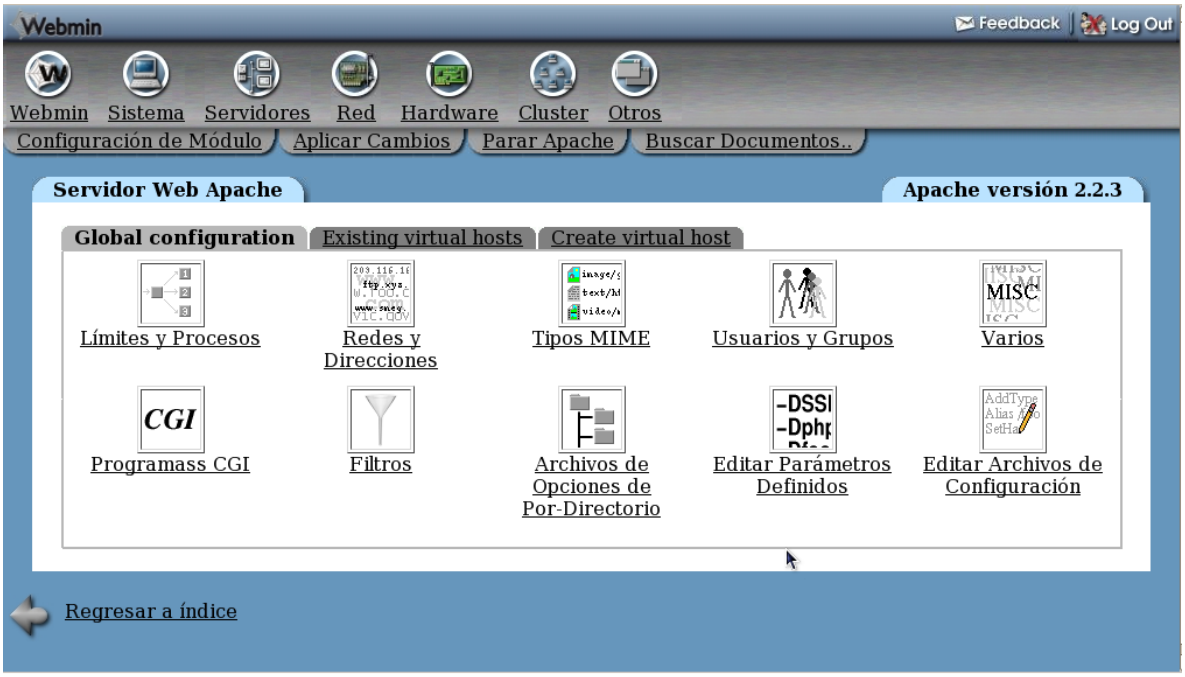


Figura 6.5.1. Administración gráfica con Webmin, módulo de apache.

6.6. Costos.

Para la determinación de los costos sólo se tomarán en cuenta los costos de la instalación de los servicios requeridos para la implementación se nuestro servidor Web pues se cuenta con la infraestructura adecuada para brindar dicho servicio, como lo son el servidor, cableado, elementos de red entre; se cuenta con toda la infraestructura y en cuanto al software que se requirió todo fue software libre por lo cual no suma ningún costo al proyecto en cuanto al pago de licencias.

Para la obtención de costo del Proyecto, nos basamos en los sueldos que puede ganar un Ingeniero en Sistemas o en Computación que se dedica a la instalación y administración de servidores Linux.

Punto 1. El Factor Humano.

Dependiendo de los conocimientos, experiencia, habilidades y capacidades con las que cuente el administrador de servidores Linux, los sueldos pueden variar, pero dada mi experiencia, en nuestro caso se cuentan con los conocimientos necesarios para el desarrollo del proyecto. Tomando un sueldo base que otorgan empresas de dicadas a la administración, mantenimiento e instalación de servidores dedicados con software libre, obtenemos que el sueldo mensual es de $16,000 pesos. Dicha cantidad la podemos traducir a una cantidad por hora, por una jornada

de trabajo de 8 horas diarias, cinco días a la semana por 4 semanas que conforman un mes; lo cual nos da un total de 160 horas en un mes teniendo un costo por hora de $100 pesos por hora.

Punto 2. Horas de trabajo.

En la realización del proyecto de Reingeniería del servidor, se invirtieron alrededor de 10 meses, tomado en cuenta todo el desarrollo del proyecto, desde la investigación, análisis, pruebas, implementación, documentación y más. Además si tomamos en consideración el punto número 1, y que se trabajó 8 horas a la semana cada mes, tenemos que las horas totales que se trabajó en el proyecto son 1600 horas.

Punto 3. Subtotal del costo.

Para obtener el total sólo hay que multiplicar, las horas de trabajo por el precio por hora, siendo esto 1600 hrs x $100 peso nos da un total de ***$160 ,000.***

Punto 4. Total y Extras.

Debemos tomar en cuenta algunos otros factores como lo son la migración del servidor anterior, la capacitación del personal encargado para obtener el costo total, lo cual nos da la siguiente cantidad.

Servidor web	$ 160,000
Migración	$ 6,000
Capacitación a 4 personas ($7500 por persona)	$ 30,000
Subtotal	$ 196,000
IVA	$ 31,360
Total	***$ 227,360***

Conclusiones y Comentarios Finales

Se logró el objetivo de contar con un servidor Web que ofreciera un servicio de alta calidad que contara con las herramientas necesarias para su óptimo funcionamiento de manera confiable y eficiente a todo aquel que cuenta con el acceso a este servicio, tomado en cuenta muchos aspectos, relacionados con el mismo. Por otra parte también se recreó nuestro servidor replanteando todos nuestros procesos haciendo su administración más óptima y fácil; es decir ,se dio una reingeniería al servicio, cambiando los paradigmas anteriores abandonando los viejos pasos y creando nuevos, rompiendo la estructura y cultura de trabajo anterior. Se cambió en forma radical en un enfoqué de reingeniería y no de mejoras continua.

Apoyandonos en nuestros objetivos iniciales podemos decir que se lograron cumplir de manera satisfactoria. Primero que nada, se dejaron atrás los viejos procesos, donde en primera instancia implantamos nuestro servidor Web en una arquitectura de virtualización del tipo hipervisor, con tecnología VMWare ESX, dándonos mayor portabilidad, ya que se cuenta con una plataforma de hardware compatible; aumenta la eficiencia aprovechando al máximo los recursos del servidor físico sin sobrecargarlo.

Un factor muy importante en todo servidor es la alta disponibilidad, y gracias las herramientas de virtualización, se cuentan con la funcionalidad de la clonación de los servidores, donde en caso de de algún cambio en nuestro sistemas se puede regresar a un instante anterior del sistema, sin impactar tanto en tiempo y costo para su recuperación.

En la instalación de nuestro sistema operativo se llevaron a cabo mecanismos de seguridad para fortalecer al mismo, realizando todo un "Hardening" para reforzar al máximo la seguridad del servidor minimizando ataques removiendo servicios innecesarios y vulnerabilidades, protegiendo los controles de acceso añadiendo una capa de acceso seguro para sitios Web, encriptando los datos que se envían de una máquina a otra en una red, así como también se instalaron herramientas, como OSSEC como sistema de detección de intrusos, que nos permite detectar comportamientos inadecuados poniéndonos en alerta de algún posible ataque. Cabe destacar por otra parte que la Instalación de la mayoría de los paquetes del sistema se realizó a través de código fuente y se compilaron para el sistema evitando problemas de arquitectura del procesador. Por otra parte se agregó un nuevo servicio de aplicación, para el manejo de sitios Web dinámicos con soporte java (Java Server Pages o JSP's) agregando el servidor Tomcat de apache.

Otro factor importante que tomamos en cuenta fue la administración de nuestros servicios y de los usuarios, haciendo uso de herramientas de administración como lo son Webmin, que nos permiten automatizar tareas y procesos facilitando el trabajo de los administradores.

Se hizo uso de herramientas como AWSTATS y Google Analitics para obtener estadísticas de la utilización de nuestro servidor Web, donde gracias a la recopilación de estos datos se realizaron recomendaciones para la creación de sitios Web Institucionales en la Facultad de Ingeniería y mejorar la interacción de los usuarios mediante buenas prácticas, políticas y lineamientos en la creación de nuestros sitios Web Institucionales y así tener una mayor visibilidad en los motores de búsqueda como sitios institucionales de la UNAM, aumentando nuestro Ranking a nivel de universidades del mundo.

Se realizaron pruebas de benchmarking para evaluar nuestros servicios en un entorno virtual, donde podemos decir que el equipo virtualizado, cuenta con mayor performance que el equipo físico con el que se contaba anteriormente, trabajando y distribuyendo sus tareas de la manera más óptima donde el sistema operativo CentOS obtuvo mayor rendimiento que Fedora.

Es de gran importancia la capacitación constante de las personas quienes se encargarán del sistema, realizar planes de contingencia, manuales de operación, para después poder evaluar mejoras, ya que todo sistema es susceptible a éstas y hacer una revisión periódica del mismo aun después de haber implantado éstos nuevos servicios pues pueden mejorar.

BIBLIOGRAFÍA

- Casad J. (2004). *Sams teach yourself TCP/IP in 24 hours.* Indianapolis*:* Sams Publishing.

- Calle Guglien J.A.(1996). *Reingeniería y Seguridad en el Ciberespacio*. España: Díaz de Santos, 1996.

- Hammer M., & Stanton S. A. (1997). *Revolución de la Reingeniería*. Madrid, España: Díaz de Santos.

- Hammersley E. (2007). *Professional VMware Server*. Indianapolis, Indiana: Wiley Publishing, Inc.

- Herrera Pérez E. (2003). *Tecnologías y redes de transmisión de datos.* D.F, México: Limusa.

- Hsletky E. (2007) *VMWare ESX Server in the Enterprice*. Estados Unidos: Prentice Hall.

- Manganelli R. L ., & Klein M. M.(1995). *Como hacer reingeniería*. Baecelona, España: Norma, 1995.

- Mañas J. A. (2004). *Mundo IP Introducción a los secretos de Internet y las redes de datos*. Madrid, España : Nowtilus S.L.

- Ruest Nelson D. (2009). *Virtualization a Beginner's Guide*. Estados Unidos : Mc Graw Hill.

- Rule D., & Dittner R. (2007). *Server virtualitation*. Estados Unidos: Syngress Publishing .

- Siebert E. (2009). *VMware VI3 Implementation and administration.* Estados Unidos: Prentice Hall.

- Spendollini M. J. (2005). *Benchmarking*. Bogota: Norma.

- Tanenbaum A. S. (2003). *Redes de computadoras.* México: Pearson Education.

MESOGRAFÍA

Benchmarking

- Balsas A.D. (1997). Linux Benchmarking COMO. Disponible en : http://es.tldp.org/COMO-INSFLUG/es/pdf/Benchmarking-COMO.pdf
- Revista Benchmark (s.f.). Herramientas de Benchmark . Disponible en: http://www.revistabenchmark.com/
- Benchmarck en Linux (s.f.). Disponible en: http://www.psicofxp.com/forums/gnu-linux.50/605129-benchmark-en-linux.html
- Linux benchmark Suite Home Page(s.f.). Disponible en: http://lbs.sourceforge.net/
- Mindcraft (2002). Webstone. Disponible en: http://www.mindcraft.com/webstone/
- Netlib(s.f.). Benchmark. Disponible en: http://www.netlib.org/benchweb/
- Camacho J. (s.f.) La nueva era de mini ordenadores. Disponible en: http://www.adminso.es/wiki/images/8/8a/PFC_Jesus_Camacho_Rodriguez_Capitulo_3.pdf

Facultad de Ingenieria.

- (2013)www.ingenieria.unam.mx

Hardening

- Ximark (s.f.)Hardening de sistemas. Disponible en: http://www.ximark.com/EthicalHacking/HardeningdeSistemas/tabid/87/Default.aspx
- OpenSSH(2013). Disponible en : http://www.openssh.com/

Herramientas

- Perl (2013).The perl programming language Disponible en :www.perl.org
- OpenSSL(2013). About OpenSSL http://www.openssl.org/
- Google Analytics.(2013). Herramientas de análisis. Disponible http://www.google.com/analytics/
- ModSecurity(2012).Disponible en: http://www.modsecurity.org/documentation/modsecurity-apache/2.0.2/modsecurity2-apache-reference.html#01-introduction
- Ossec(2012)How it works .Disponible en :http://www.ossec.net/

Instalación del mod_perl

- Mod_perl (2011). Getting your feet wet with mod_perl. Disponible en : http://perl.apache.org/docs/2.0/user/intro/start_fast.html
- Mod_perl (2011)Installing mod_perl . Disponible en : http://perl.apache.org/docs/2.0/user/install/install.html#Prerequisites

Instalación de modsecurity

- ModSecurity(2012). Installation. Disponible en: http://www.modsecurity.org/documentation/modsecurity-apache/2.5.12/html-multipage/installation.html
- Modsecurity install (2009). Disponible en: http://comments.gmane.org/gmane.comp.apache.mod-security.user/7052

Jpef e iperf

- Bycoders (2009). Medir la red con Iperf .Disponible en: http://bytecoders.net/content/medir-la-red-con-iperf.html
- Iperf(s.f). Disponible en:http://iperf.sourceforge.net/
- Iperf Rendimiento de la red (2009).Disponible en : http://seguridadyredes.nireblog.com/post/2009/10/23/jperf-el-frontend-grafico-de-iperf-rendimiento-de-la-red

Lenguajes de programación:

- Alegsa(2013). Lenguaje de programación. Disponible en: http://www.alegsa.com.ar/Dic/lenguaje%20de%20programacion.php
- PHP(2013).¿Que es PHP?. Disponible en: http://php.net/manual/es/intro-whatis.php

Libreia lua

- Lua.(2013).Disponible en: http://www.lua.org/download.html
- Lua(2012).Pre-compiled Lua libraries and executables. Disponible en: http://luabinaries.sourceforge.net/

Lmbench

- LMbench (s.f.). Why Lmbench. Disponible en:.http://www.bitsmover.com/lmbench/

Protocolos de Comunicación

- El Rincón del vago.(s.f). Protocolos de comunicación. Disponible en : http://html.rincondelvago.com/protocolos-de-comunicacion_1.html
- Kioskea (s.f). ¿Qué es un protocolo? Disponible en: http://es.kioskea.net/contents/internet/protocol.php3

Ranking web

- Ranking de los mejores servidores (s.f). Disponible en: http://www.grupoinformatica.com/noticias/1141-ranking-de-los-mejores-servidores-mayo-2010.html
- Netcraft (2011).Web Server Survey . Disponible en : http://news.netcraft.com/archives/2010/05/14/may_2011_web_server_survey.html
- Netcraft (2010). Ranking de servidores.Disponible en : http://www.grupoinformatica.com/noticias/130-ranking-de-servidores-julio-2010.html
- Netcraft(20110).Web Server Survey .Disponible en: http://news.netcraft.com/archives/2010/07/16/july-2010-web-server-survey-16.html
- Webometrics.(2013).Ranking Web of universities.Disponible en: http://www.webometrics.info/about_rank_es.html
- Goroztiza I.(2007). Como mejorar el ranking de una web en google .Disponible en: http://www.hellogoogle.com/ranking-de-una-web-en-google/

Seguridad

- Nixtcraf (2013).CentOS/Red-Hat: Turn on SELinux protection. Disponible en: http://www.cyberciti.biz/faq/rhel-fedora-redhat-selinux-protection/
- Seguridad en unix (s.f).Disponible en: http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/doc-unixsec/unixsec-html/node1.html
- Gomez Aguilar D. & García Navaro J.F (2010). Mecanismos de Seguridad de la información en aplicaciones web. Disponible en: http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07Trabajos/SeguridadAppWeb.pdf
- QeiMED (s.f.). Mecanismos de seguridad. Disponible en: http://www.qeimed.com/index.php?option=com_content&view=article&id=23:mecanismos-de-seguridad&catid=36:consultoria
- Apuntes de Seguridad(s.f.).Disponible en: http://www.dasumo.com/libros/seguridad-informatica-pdf-2.html

Servidor Web.

- Apache (2013).Disponible en: www.apache.org/
- Apache-Tomcat (2013). Disponible en :www.tomcat.apache.org
- Cibernetia(s.f). Conceptos básicos. Disponible en: http://www.cibernetia.com/manual/instalacion_servidor_web/1_conceptos_basicos.php

SSL y TLS

- IETF(1996). The SSL Protocol. Disponible en: http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00
- IETF (2000). HTTP over TLS. Disponible en: http://www.ietf.org/rfc/rfc2818.txt
- IETF (2006). The Transport Layer Security (TLS) Protocol. Disponible en: http://tools.ietf.org/html/rfc4346

TCP/IP

- Microsoft (2005). Modelo TCP/IP. Disponible en: http://technet.microsoft.com/es-es/library/cc786900(WS.10).aspx
- Kioskea (2013).TCP/IP. Disponible en : http://es.kioskea.net/contents/internet/tcpip.php3

RFC 2616

- W3C(2004). Disponible en: http://www.w3.org/Protocols/rfc2616/rfc2616.html
- IETF(2004) Disponible en: http://www.ietf.org/rfc/rfc2616.txt

Unidad de Servicios de Cómputo Académico

- UNICA(2013). Organización. Disponible en : http://www.ingenieria.unam.mx/~unica/

Unixbench

- Byte-UnixBench (2011). UnixBench. Disponible en: http://code.google.com/p/byte-unixbench/

vmware

- VMware(2013). Virtualization Technology & Industry. Disponible en : http://www.vmware.com/

WayBack Machine, Histórico de Páginas.

- WayBack Machine, Histórico de Páginas (2013). Disponible en : http://wayback.archive.org/

Webstone

- Mindcraft(2003).Webstone. Disponible en: http://www.mindcraft.com/webstone/

TABLAS

FIGURAS

APÉNDICE

A

API

Interfaz de programación de aplicaciones del inglés ***API*** *-Application Programming* Interface. Es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

ARP

Protocolo de resolución de Direcciones, por sus siglas en inglés ***ARP****-Address Resolution* Protocol, es un protocolo de la capa de enlace de datos responsable de encontrar la dirección hardware que corresponde a una determinada dirección IP. Para ello se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast (MAC = FF FF FF FF FF FF)) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (ARP reply) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga. ARP permite a la dirección de Internet ser independiente de la dirección Ethernet, pero esto sólo funciona si todas las máquinas lo soportan.

C

CERN

La Organización Europea para la Investigación Nuclear (nombre oficial), comúnmente conocida por la sigla ***CERN*** (sigla provisional utilizada en 1952, que respondía al nombre en francés *Conseil Européen pour la Recherche Nucléaire*, es decir, Consejo Europeo para la Investigación Nuclear) es el mayor laboratorio de investigación en física de partículas a nivel mundial.

CPU

La Unidad Central de Procesamiento del inglés ***CPU***-*Central Processing Unit* o procesador, es el componente principal de una computadora y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos.

CSIC

Consejo Superior de Investigaciones Científicas (CSIC) es la mayor institución pública dedicada a la investigación en España y la tercera de Europa. Adscrita al Ministerio de Economía y Competitividad de España, a través de la Secretaría de Estado de Investigación, su objetivo fundamental es desarrollar y promover investigaciones en beneficio del progreso científico y tecnológico, para lo cual está abierta a la colaboración con entidades españolas y extranjeras.

CRLF

CRLF se refiere a la combinación de dos códigos de control: CR (retorno de carro) y LF (salto de línea), uno detrás del otro; con el objetivo de crear una nueva línea. Algunos de los primeros protocolos de red, que transmitían principalmente texto, establecieron que el terminador de línea debía ser CRLF y no otro. Son ejemplos HTTP, FTP, IRC, o SMTP, que marca el final del mensaje mediante CRLF. CRLF

D

DD

DD, se refiere a disco duro, es un dispositivo de almacenamiento de datos no volátil que emplea un sistema de grabación magnética para almacenar datos digitales.

DES

Estándar de encriptación de datos o por sus siglas en ingles ***DES****-Data Encryption Standard* **.**Es un algoritmo de cifrado, es decir, un método para cifrar información, escogido como un estándar FIPS en los Estados Unidos en 1976. El algoritmo fue controvertido al principio, con algunos elementos de diseño clasificados, una longitud de clave relativamente corta, y las continuas sospechas sobre la existencia de alguna puerta trasera para la *National Security Agency (NSA).* Posteriormente DES fue sometido a un intenso análisis académico y motivó el concepto moderno del cifrado por bloques y su criptoanálisis.

La forma de trabajar del DES se basa en convertir el mensaje en una larga serie de dígitos binarios (bits). Tras llevar a cabo esta acción, se divide en bloques de 64 bits cifrando, cada uno de ellos, de forma separada. Después se toma, de forma independiente, cada uno de los bloques, ya que primero se realiza una permutación inicial, tras la cual se dividen los 64 bits en dos partes de 32 bits cada una. Posteriormente se realizan 16 vueltas repitiendo las mismas operaciones, en las cuales los datos se combinan con la clave de 56 bits. Tras las 16 vueltas, las dos mitades vuelven a unirse y se les aplica una permutación final inversa a la realizada al comienzo del proceso.

Hoy en día, DES se considera inseguro para muchas aplicaciones; las claves de DES se han roto en menos de 24 horas.

DNS

Sistema de Nombres de Dominio o por sus siglas en inglés ***DNS****-Domain Name System* .Es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para las personas en identificadores binarios asociados con los

equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

DoD

El Departamento de Defensa de Estados Unidos o por sus siglas en inglés ***DoD**-Department of Defense* es el departamento ejecutivo del gobierno de los Estados Unidos encargado de coordinar y supervisar todas las agencias y funciones del gobierno relacionadas directamente con la seguridad nacional y las Fuerzas Armadas de los Estados Unidos.

Drivers

Elemento software utilizado en diversos sistemas operativos, también llamado manejador de dispositivo, controlador de dispositivo o driver.

DROS

Departamento de Redes y Operación de Servidores (DROS). Responsable de la administración, operación, mantenimiento y seguridad de la red de comunicación de la Facultad de Ingeniería y de la intercomunicación con la red central de la UNAM; así como del desarrollo e implementación de proyectos para la expansión del servicio.

VMWare DRS

VMware DRS (Distributed Resource Scheduler), es una utilidad que equilibra la carga de trabajo computacional, de los recursos de un entorno virtualizado. Ésta utilidad es parte de una suite de visrtualizaciónde VMware Infrastructure 3. VMware DRS, los usuarios administradores de un entorno vistrualizado definen las reglas de los recursos físicos entre las máquinas virtuales.

F

FTP

***FTP**-File Transfer Protocol,* 'Protocolo de Transferencia de Archivos') en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un

equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

G

GNU

GNU es un sistema operativo similar a Unix que es software libre y respeta su libertad. Puede instalar versiones de GNU (más precisamente, sistemas GNU/Linux) que son completamente software libre.

El Proyecto GNU se inició en 1984 para desarrollar el sistema GNU. El nombre GNU(que significa "ñu" en inglés) es un acrónimo recursivo de ¡GNU No es Unix! y en español se pronuncia fonéticamente como una sílaba sin vocal entre la *g* y la *n*.

Los sistemas operativos similares a Unix se construyen a partir de un conjunto de aplicaciones, bibliotecas y herramientas de programación, además de un programa para alojar recursos e interactuar con el hardware, denominado núcleo.

GPL

La **Licencia Pública General de GNU** o más conocida por su nombre en inglés **GNU General Public License** (o simplemente sus siglas del inglés **GNU GPL**) es la licencia más ampliamente usada[1] en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios. Esta licencia fue creada originalmente por Richard Stallman fundador de la Free Software Foundation (FSF) para el proyecto GNU (GNU project).

H

HA(higth avility)

Alta disponibilidad (High availability) es un protocolo de diseño del sistema y su implementación asociada que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado. Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos. Si un usuario no puede acceder al sistema se dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible.

Hardening

Hardening (palabra en ingles que significa endurecimiento) en seguridad informática es el proceso de asegurar un sistema mediante la reducción de vulnerabilidades en el mismo

Hosting

El alojamiento web o en inglés web hosting es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

HTML

Lenguaje de marcas de hipertexto o por sus siglas en ingles ***HTML****-HyperText Markup Language*, hace referencia al lenguaje de marcado para la elaboración depáginas web. Es un estándar que, en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

HTTP

Protocolo de Transferencia de Hipertexto HTTP son las siglas en inglés de HiperText Transfer Protocol. Es un protocolo de red (un protocolo se puede definir como un conjunto de reglas a seguir) para publicar páginas de web o HTML. HTTP es la base sobre la cual está fundamentado Internet, o la WWW.

HTTPS

Protocolo Seguro de Transferencia de Hipertexto o por sus siglas en ingles ***HTTPS****-Hypertext Transfer Protocol Secure* más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De éste modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

El puerto estándar para este protocolo es el 443.

I

IBM

International Business Machines (IBM) (NYSE: IBM) es una empresa multinacional estadounidense de tecnología y consultoría con sede en Armonk,Nueva York. IBM fabrica y comercializa hardware y software para computadoras, y ofrece servicios de infraestructura, alojamiento de Internet, y consultoría en una amplia gama de áreas relacionadas con la informática, desde computadoras centrales hasta nanotecnología.

IDS

El término IDS (Sistema de detección de intrusiones) hace referencia a un mecanismo que, sigilosamente, escucha el tráfico en la red para detectar actividades anormales o sospechosas, y de éste modo, reducir el riesgo de intrusión.

Existen dos claras familias importantes de IDS:

- El grupo N-IDS (Sistema de detección de intrusiones de red), que garantiza la seguridad dentro de la red.
- El grupo H-IDS (Sistema de detección de intrusiones en el host), que garantiza la seguridad en el host.

IETF

Grupo de Trabajo de Ingeniería de Internet o por sus siglas en inglés ***IETF**-Internet Engineering Task Force* .Es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, como transporte, encaminamiento, seguridad. Fue creada en EE. UU. en 1986. El IETF es mundialmente conocido por ser la entidad que regula las propuestas y los estándares de Internet, conocidos como RFC.

ISO

La Organización Internacional de Normalización o ISO, nacida tras la Segunda Guerra Mundial (23 de febrero de 1947), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación (tanto de productos como de servicios), comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones (públicas o privadas) a nivel internacional.

L

LAN

LAN son las siglas de Local Área Network, Red de área local. Una LAN es una red que conecta las computadoras en un área relativamente pequeña y predeterminada (como una habitación, un edificio, o un conjunto de edificios).

Una red de área local es una red en su versión más simple. La velocidad de transferencia de datos en una red de área local puede alcanzar hasta 10 Mbps (por ejemplo, en una red Ethernet) y 1 Gbps (por ejemplo, en FDDI o Gigabit Ethernet). Una red de área local puede contener 100, o incluso 1000, usuarios.

M

MAC

Control de Acceso al Medio o ***MAC**-Media Access Control*; es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red. Se

conoce también como dirección física, y es única para cada dispositivo. Está determinada y configurada por el IEEE (los últimos 24 bits) y el fabricante (los primeros 24 bits) utilizando el organizationally unique identifier. La mayoría de los protocolos que trabajan en la capa 2 del modelo OSI usan una de las tres numeraciones manejadas por el IEEE: MAC-48, EUI-48, y EUI-64, las cuales han sido diseñadas para ser identificadores globalmente únicos. No todos los protocolos de comunicación usan direcciones MAC, y no todos los protocolos requieren identificadores globalmente únicos.

MD5

En criptografía, MD5 es la abreviatura de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5 es un algoritmo de reducción criptográfico de 128 bits. MD5 es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (Massachusetts Institute of Technology, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese su debilidad.

N

NAS

Almacenamiento conectado a Red o por sus siglas en inglés ***NAS*** *-Network Attached Storage*, que es un sistema de almacenamiento digital pensado para conectarse a la red y dar servicio a computadoras y servidores. Un sistema NAS es un lugar de almacenamiento de archivos centralizado. Generalmente, los sistemas NAS son dispositivos de almacenamiento específicos a los que se accede desde los equipos a través de protocolos de red (normalmente TCP/IP). También se podría considerar un sistema NAS a un servidor (Microsoft Windows, Linux) que comparte sus unidades por red, pero la definición suele aplicarse a sistemas específicos.

NCP

Protocolo de Control de Red o del inglés ***NCP****-Network Control Protocol* es un protocolo de control del nivel de red que se ejecuta por encima de PPP (Point-to-point Protocol en español Protocolo punto a punto,). Se usa para negociar y configurar la red que va sobre PPP. Es específico para cada tipo de red.

O

OSSEC

El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (en inglés, Open System Interconnection 'sistemas de interconexión abiertos') es el modelo de red descriptivo, que fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1980.1 Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones. Está basado en un modelo de cliente-servidor, con lo cual tendremos un servidor centralizado que se encarga de recibir y actuar en base a la información que reciba de los agentes que, en definitiva, son las máquinas que están siendo monitorizadas y atacadas. La forma de actuar del servidor es, por una parte, enviando notificaciones, y por otro lado, si así se configura, generando reglas firewall que se ejecutarán en los propios agentes.

OSI

El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (en inglés, Open System Interconnection 'sistemas de interconexión abiertos') es el modelo de red descriptivo, que fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1980.1 Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.

P

PC

Computadora personal, también conocida como PC o por sus siglas en inglés de *PC-Personal Computer*, es una microcomputadora diseñada en principio para ser usada por una sola persona a la vez.

POP

Post Office Protocol (Protocolo de Oficina de Correos **o** Protocolo de Oficina Postal). Es un protocolo de nivel de aplicación en el Modelo OSI. Las versiones del protocolo POP, informalmente conocido como POP1 y POP2, se han hecho obsoletas debido a las últimas

versiones de POP3. En general cuando se hace referencia al término POP, se refiere a POP3 dentro del contexto de protocolos de correo electrónico.

R

Ranking

Un ranking es una lista que establecerá una relación entre el conjunto de elementos que se reúnen en la misma, es decir, hay una característica en común que comparten y que los hace pertenecer a esa lista, en tanto, cada elemento poseerá una característica propia y especial que lo hará estar por arriba o por debajo de los otros elementos.

RAM

Memoria de Acceso Aleatorio o por sus siglas en ingles Random Access Memoryse utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo. Se denominan «de acceso aleatorio» porque se puede leer o escribir en una posición de memoria con un tiempo de espera igual para cualquier posición, no siendo necesario seguir un orden para acceder a la información de la manera más rápida posible.

RARP

Protocolo de Resolución de Dirección Inversa o por sus siglas en ingles ***RARP****-Reverse Address Resolution Protocol* es mucho menos utilizado. Es un tipo de directorio inverso de direcciones lógicas y físicas.

En realidad, el protocolo RARP se usa esencialmente para las estaciones de trabajo sin discos duros que desean conocer su dirección física. El protocolo RARP le permite a la estación de trabajo averiguar su dirección IP desde una tabla de búsqueda entre las direcciones MAC (direcciones físicas) y las direcciones IP alojadas por una pasarela ubicada en la misma red de área local (LAN).

RC4

RC4 es el cifrado de flujo software más utilizado y se utiliza en algunos de los protocolos más populares como Transport Layer Security (TLS/SSL) (para proteger el tráfico de Internet) y Wired Equivalent Privacy (WEP) (para añadir seguridad en las redes inalámbricas). RC4 fue excluido enseguida de los estándares de alta seguridad por los criptógrafos y algunos modos de usar el algoritmo de criptografía RC4 lo han llevado a ser un sistema de criptografía muy inseguro, incluyendo su uso WEP. No está recomendado su uso en los nuevos sistemas, sin embargo, algunos sistemas basados en RC4 son lo suficientemente seguros para un uso común.

RC4 fue diseñado por Ron Rivest de RSA Security en 1987 - Mientras se denomina oficial "Rivest Cipher 4" las siglas RC se entiende como alternativa a colocarse para el "Código de Ron".

RHEL

Red Hat Enterprise Linux también conocido por sus siglas RHEL es una distribución comercial de Linux desarrollada por Red Hat. Es la versión comercial basada en Fedora que a su vez está basada en el anterior Red Hat Linux, de forma similar a como Novell SUSE Enterprise (SUSE Linux Enterprise Desktop y SLE Server) lo es respecto de OpenSUSE o Mandriva Corporate respecto de Mandriva Linux One.

Mientras que las nuevas versiones de Fedora salen cada aproximadamente 6 meses, las de RHEL suelen hacerlo cada 18 o 24 meses.

RPM

Del ingles ***RPM****-Red Hat Package Manager*, pero se convirtió en acrónimo. Es una herramienta de administración de paquetes pensada básicamente para GNU/Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas. RPM es el formato de paquete de partida del Linux Standard Base.

Originalmente desarrollado por Red Hat para Red Hat Linux, en la actualidad muchas distribuciones GNU/Linux lo usan, dentro de las cuales las más destacadas son Fedora Linux, Mandriva Linux y SuSE Linux CentOS.

S

SAN

Red de Área de almacenamiento o por sus siglas en inglés ***SAN****-Storage Area Network*. Una SAN es una red dedicada al almacenamiento que está conectada a las redes de comunicación de una compañía. Además de contar con interfaces de red tradicionales, los equipos con acceso a la SAN tienen una interfaz de red específica que se conecta a la SAN.

SHA

Algoritmo de Hash Seguro o del inglés ***SHA****-Secure Hash Algorithm*. Es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el National Institute of Standards and Technology (NIST). El primer miembro de la familia fue publicado en 1993 es oficialmente llamado SHA. Sin embargo, hoy día, no oficialmente se le llama SHA-0 para evitar confusiones con sus sucesores. Dos años más tarde el primer sucesor de SHA fue publicado con el nombre de SHA-1. Existen cuatro variantes más que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512 (llamándose SHA-2 a todos ellos).

SMTP

Protocolo simple de transferencia de correo o por sus siglas en ingles Simple Mail Transfer Protocol, es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

Éste es un protocolo que funciona en línea, encapsulado en una trama TCP/IP. El correo se envía directamente al servidor de correo del destinatario. El protocolo SMTP funciona con comandos de textos enviados al servidor SMTP (al puerto 25 de manera predeterminada). A cada comando enviado por el cliente (validado por la cadena de caracteres ASCII CR/LF, que equivale a presionar la tecla Enter) le sigue una respuesta del servidor SMTP compuesta por un número y un mensaje descriptivo.

SNMP

Protocolo simple de administración de red o por sus siglas en inglés ***SNMP****-imple Network Management Protocol.* Es un protocolo de la capa de aplicación que les permite a los administradores de red administrar dispositivos de red y diagnosticar problemas en la red.

SO

Sistema Operativo (SO) es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes y anteriores próximos y viceversa.

SSH

SSH Secure SHell, intérprete de órdenes segura es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix y Windows) corriendo.

Además de la conexión a otros dispositivos, SSH nos permite copiar datos de forma segura (tanto archivos sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

SSL

Capa de conexión segura o del inglés ***SSL****-Secure Sockets Layer* y su sucesor Transport Layer Security (TLS; en español «seguridad de la capa de transporte») son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet. Las aplicaciones que utilizan el protocolo Secure Sockets Layer sí saben cómo dar y recibir claves de cifrado con otras aplicaciones, así como la manera de cifrar y descifrar los datos enviados entre los dos.

T

TCP

Protocolo de Control de Transmisión o por sus siglas en inglés TCP- Transport control Protocol es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. En el nivel de aplicación, posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él, (es decir, el protocolo IP). Cuando se proporcionan los datos al protocolo IP, los agrupa en datagramas IP, fijando el campo del protocolo en 6 (para que sepa con anticipación que el protocolo es TCP). TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión.

Telnet

El protocolo Telnet es un protocolo de Internet estándar que permite conectar terminales y aplicaciones en Internet. El protocolo proporciona reglas básicas que permiten vincular a un cliente (sistema compuesto de una pantalla y un teclado) con un intérprete de comandos (del lado del servidor).

El protocolo Telnet se aplica en una conexión TCP para enviar datos en formato ASCII codificados en 8 bits, entre los cuales se encuentran secuencias de verificación Telnet. Por lo tanto, brinda un sistema de comunicación orientado bidireccional (semidúplex) codificado en 8 bits y fácil de implementar.

TLS

El protocolo TLS Transport Layer Security o Capa de transporte segura es una evolución del protocolo SSL (Secure Sockets Layer), es un protocolo mediante el cual se establece una conexión segura por medio de un canal cifrado entre el cliente y servidor. Así el intercambio de información se realiza en un entorno seguro y libre de ataques.

U

UDP

El protocolo UDP (Protocolo de datagrama de usuario) es un protocolo no orientado a conexión de la capa de transporte del modelo TCP/IP. Este protocolo es muy simple ya que no

proporciona detección de errores (no es un protocolo orientado a conexión). Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco se sabe si ha llegado correctamente, ya que no hay confirmación de entrega o recepción. Su uso principal es para protocolos como DHCP, BOOTP, DNS y demás protocolos en los que el intercambio de paquetes de la conexión/desconexión son mayores, o no son rentables con respecto a la información transmitida, así como para la transmisión de audio y vídeo en tiempo real, donde no es posible realizar retransmisiones por los estrictos requisitos de retardo que se tiene en éstos casos.

UNAM

Universidad Nacional Autónoma de México es la institución de educación superior más importante del país. Desde su fundación ha sido el arquetipo de la educación universitaria en México. La docencia, la investigación y la extensión de la cultura son sus tareas primordiales, mismas que se manifiestan en todos los ámbitos de la vida nacional por su labor formativa, propositiva y de servicio a la sociedad.

UNICA

La Unidad de Servicios de Cómputo Académico (UNICA) lleva a cabo las tareas académicas y administrativas de la Facultad de Ingeniería. Las funciones que desempeña la Unidad de Servicios de Cómputo Académico son:

- Mantener el liderazgo en cuanto a tópicos en cómputo.
- Continuar proporcionando recursos de cómputo de calidad a la comunidad de la Facultad.
- Impulsar a nivel de la Facultad la creación de una política de cómputo definida.
- Lograr la capacitación cada vez más completa y actualizada para la formación de recursos humanos.
- Aplicar todos los conocimientos y las herramientas de cómputo con los que cuenta la Unidad para realizar las actividades de forma más eficiente y segura.

V

VM

Maquína Virtual o Virtual Machine es un software que simula a una computadora y puede ejecutar programas como si fuese una computadora real. Este software en un principio fue definido como "un duplicado eficiente y aislado de una máquina física". La acepción del término actualmente incluye a máquinas virtuales que no tienen ninguna equivalencia directa con ningún hardware real.

W

WWW

World Wide Web (WWW) o Red informática mundial1 comúnmente conocida como la web es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.

X

XML

XML, siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. XML no ha nacido sólo para su aplicación para Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.