

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

VNIVERSIDAD NACIONAL
AVTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

T É S I S

Para obtener el Título de Ingeniero en Computación

"INTEGRACIÓN DE UNA RED WINDOWS NT-LINUX PARA EL LABORATORIO DE CONTROL DIGITAL"



Presentado por:

Chanes Gutiérrez Vicente
Chapa López Javier
Saucedo Uribe Pedro Gabriel

Director: Ing. Rafael Iriarte V. Balderrama

JULIO 2005

Dedicatorias Javier…

A mi madre...
Por darme todo su amor, su comprensión, su cariño y en todo momento su apoyo incondicional, por enseñarme el camino de la responsabilidad y lograr ser quien soy ahora. Por siempre estar ahí.....

A mi padre...
Por su apoyo en todo momento, por su ejemplo profesional y por insistir tanto en que concluyera este proyecto.

A mi hermano...
Por saber agradecer el esfuerzo de nuestros padres. Siempre cuenta conmigo....

A mis tías... Carmen, Ángeles, Beatriz y Gloria...
Por todas sus buenas enseñanzas, por siempre estar conmigo en las buenas y en las malas...por quererme.

A mis amigos(as)...
Por creer en mí, por compartir momentos inolvidables y por motivarme a intentar siempre ser mejor.

A mis profesores(as)...
Por ayudarme a forjar mi vida profesional.

Dedicatorias Pedro…

A mi abuelo paterno
Por haberme enseñado el camino de la disciplina y a confiar en mi, por haber sido la imagen mas grande que guió mi vida y que probablemente lo hará hasta el fin de mis días

A mi abuelo materno
Por enseñarme a ser responsable y haberme enseñado el valor del trabajo, entre tú y yo hubo una promesa que me sacó adelante en los momentos más difíciles de mi vida

A mi madre
Por demostrarme el cariño real y efectivo de una madre que no se basa en bonitas palabras sino en acciones contundentes y haberme dado la paz que necesite en el peor momento de mi vida, tu sabes que eso yo nunca lo olvidare y siempre te lo agradeceré.

A mi hermana
Tú siempre confiaste en mí y al igual que mi madre, estuviste a mi lado siempre, sin preguntar y siempre diste lo mejor de ti para ayudarme, cuenta conmigo siempre.

A mi tía Estela
Eres la imagen que guió mi desarrollo hacia la ciencia y tu forma de ser y tu cariño tienen un eco muy grande en mi vida

A mi padre
Te agradezco por tus buenas intenciones.

A mi familia
Les agradezco a todos y cada uno por su apoyo y cariño y realmente tanto los malos momentos como los buenos siempre se que los puedo compartir con ustedes.

A Othon Tirado Rasso
Gracias por tu firmeza y carácter así como las palabras de aliento y tu comprensión. Has sido el faro que me ha guiado en los últimos años para llegar a buen puerto.

A Ana Luisa
El conocerte fue el premio que ha dado la vida o quizá Dios a mi esfuerzo y lucha constante

A mis amigos y amigas
Por haber compartido con ustedes mi infancia, mi juventud y por ser parte importante de mi vida. Tantas experiencias tan bonitas que vivimos juntos.

Dedicatorias Vicente…

A mi padre...
Que en paz descanse.

A mi madre...
Para que siga creyendo en mi.

TABLA DE CONTENIDO

INTRODUCCIÓN....11

A) → COMENTARIOS INICIALES....11

B) → OBJETIVOS....14

C) → ESTADO INICIAL DEL LCD....14

1. EVALUACIÓN DE ALTERNATIVAS DE SOLUCIÓN....17

1.1→ SISTEMAS OPERATIVOS....17

1.1.1 → Sistema LINUX....18
1.1.1.1 → LINUX - Breve Reseña Histórica....18
1.1.1.2 → LINUX - Principales Características....19
1.1.1.3 → LINUX - Ventajas....21
1.1.1.4 → LINUX - Desventajas....22

1.1.2. → Sistema UNIX....23
1.1.2.1 → UNIX – Breve reseña histórica....23
1.1.2.2 → UNIX – Características básicas....25
1.1.2.3 → UNIX – Componentes....26
1.1.2.4 → UNIX – Ventajas....27
1.1.2.5 → UNIX – Desventajas....28

1.1.3 → Sistema MS-DOS®....28
1.1.3.1 → MSDOS® – Breve reseña histórica....29
1.1.3.2 → MSDOS® – Sus componentes....32
1.1.3.3 → MSDOS® – Ventajas....32
1.1.3.4 → MSDOS® – Desventajas....33

1.1.4 → Sistema IBM® OS/2....34
1.1.4.1 → IBM OS/2 – Breve reseña histórica y características principales....34
1.1.4.2 → IBM OS/2 – Ventajas....37
1.1.4.3→ IBM OS/2 – Desventajas....38

1.1.5 → Sistema Macintosh® OS....38
1.1.5.1 → Macintosh® OS – Principales componentes....39
1.1.5.2 → Macintosh® OS – Ventajas....40
1.1.5.3 → Macintosh® OS – Desventajas....40
1.1.6 → Sistema Windows® NT....41
1.1.6.1 → Windows® NT – Breve reseña histórica....42
1.1.6.2 → Windows® NT Server – Descripción general....42
1.1.6.3 → Windows® NT Workstation – Descripción general....43
1.1.6.4 → Windows® NT Workstation – Ventajas....44
1.1.6.5 → Windows® NT Workstation – Desventajas....45

1.2 → SOLUCIÓN DEFINITIVA PROPUESTA....45

1.2.1 → Solución para los Servidores....45

1.2.2 → Solución para las Computadoras Clientes....47

1.2.3 → Solución para la Red Física....48

1.2.4 → Solución de Integración entre Sistemas Operativos....48
1.2.4.1 → SAMBA – Configuración general....49
1.2.4.2 → SAMBA – Configuración Linux Server con PC's Windows®....51
1.2.4.3 → SAMBA – Nuestra configuración....51

2. CARACTERÍSTICAS TÉCNICAS Y ADECUACIÓN....54

2.1→ DISEÑO Y DISTRIBUCIÓN DE LA RED DE DATOS....54

2.1.1→ Suministro e Instalación de Materiales....55
2.1.1.1 → Cableado....55
2.1.1.2 → Centro de Distribución de Cableado....56
2.1.1.3→ Organizador de Cables....58
2.1.1.4→ Panel de Parcheo (Patch Panel)....58
2.1.1.5 → Conectores....60
2.1.1.6 → Tapas de Pared (Wallplates)....61
2.1.1.7 → Remates....62
2.1.1.8 → Cables de Red (Patch Cord)....63
2.1.1.9 → Concentradores (HUB's)....65
2.1.1.10 → Hardware....66

2.2 → DOCUMENTACIÓN DE INSTALACIÓN (CABLEADO)....68

2.2.1 → Memoria Técnica....68
2.2.1.1 → Identificación y Etiquetado del Cableado....69
2.2.1.2 → Identificación de Tapas de Pared (Wallplates)....69
2.2.1.3 → Memoria Técnica de la Instalación....70

2.3 → ESTÁNDARES Y NORMAS UTILIZADAS....71

2.3.1 → Generalidades sobre Normas de Cableado Estructurado....72

2.3.2 → Norma ANSI EIA/TIA 568 y Su Aplicación al Proyecto 74

3. SISTEMA DE ACCESO POR INTERNET 81

3.1 → PLANTEAMIENTO 81

3.2 → DESARROLLO 82

3.3 → PROGRAMACION 84

3.3.1 → Programas SHELL SCRIPT 84
3.3.1.1 → Programas – Generalidades del administrador 84
3.3.1.2 → Programas – Grupos 88
3.3.1.3 → Programas – Profesores 90
3.3.1.4 → Programas – Alumnos 93

3.3.2 → Programas JAVA 94
3.3.2.1 → Programas – Desde Internet 95

4. MANUALES DE OPERACIÓN 121

4.1 → MANUAL DE OPERACIÓN - ALUMNOS 121

4.1.1 → Accediendo a Nuestros Archivos desde Internet 123

4.1.2 → Subiendo Archivos al Servidor del LCD desde Internet 123

4.1.3 → Cambiando la Contraseña de Acceso 125

4.1.4 → Revisando los Avisos del Profesor 127

4.1.5 → Saliendo del Sistema 129

4.2 → MANUAL DE OPERACIÓN - PROFESORES 130

4.2.1 → Accediendo al Sistema 131

4.2.2 → Publicando Avisos 131

4.2.3 → Cambiando la Contraseña 135

4.2.4 → Saliendo del Sistema 136

4.3 → MANUAL DE OPERACIÓN - ADMINISTRADOR 137

4.3.1 → Agregando Usuarios (Alumnos, Profesores Y Grupos) 138
4.3.1.1 → Crear Profesor 139
4.3.1.2 → Crear Alumno 140
4.3.1.3 → Crear un Grupo 141
4.3.1.4 → Vaciar Cuentas 142

4.3.2 → Administrando los Grupos del LCD 143
4.3.2.1 → Profesores 143
4.3.2.2 → Alumnos 145

4.3.3 → Cambiando la Contraseña del Administrador 146

5. CONCLUSIONES 148

APÉNDICE 153

A) EL FIREWALL 153

GLOSARIO DE TÉRMINOS 159

BIBLIOGRAFÍA 168

INTRODUCCIÓN

A) ® COMENTARIOS INICIALES

El Laboratorio de Control Digital (LCD) dependiente del Departamento de Ingeniería de Control dentro de la División de Ingeniería Eléctrica (DIE) de la Facultad de Ingeniería de la UNAM da servicio a los alumnos de la asignatura de Control Digital. Esta asignatura es cursada por los estudiantes de las carreras de Ingeniero Eléctrico Electrónico, Ingeniero en Computación e Ingeniero en Telecomunicaciones.

Con el objeto de brindar un mejor servicio a los estudiantes que en número son aproximadamente 250 alumnos por semestre, el LCD propuso desarrollar el presente tema de tesis cuya finalidad es ofrecer a los alumnos las facilidades que la tecnología de cómputo actual puede ofrecer con el objeto de hacer más eficiente el aprendizaje de los temas de control especificados en los contenidos de la asignatura respectiva.

Este trabajo de tesis propone una red mixta Linux-NT con cableado estructurado, protocolos **NetBeui*** y principalmente TCP/IP*. El servidor Linux forma el dominio con las máquinas cliente que corren en el ambiente Windows® NT mediante la interfase Samba. Uno de los objetivos de esta tesis es que los alumnos puedan acceder desde su casa a sus archivos locales en las máquinas cliente del LCD permitiendo a profesores y alumnos optimizar su tiempo efectivo de clase.

Se utilizó el protocolo TCP/IP con direcciones IP de clase C para redes virtuales. Se cuenta con una puerta de enlace (Gateway) con la cual la red tiene salida a Internet (controlada.) Dicha puerta de enlace tiene una dirección IP real suministrada por el administrador de la red del Departamento. La salida a Internet cuenta con un **Firewall*** instalado vía software para la protección de dicho sistema contra acciones negativas que pudieran afectar la seguridad de la red.

***** Ver glosario de términos, incluido en este documento

La máquina que sirve de **Firewall** bajo el sistema operativo Linux 6.2 tiene dos tarjetas de red; una para la conexión a Internet y otra para la red interna de área local del laboratorio.

Una de las razones por las cuales se eligió Linux como sistema operativo en ambos servidores, es porque no requiere hardware con tecnología de punta para ejecutarse de una forma eficiente, además de que este sistema operativo nos permite configurar el **Firewall** sin necesidad de realizar mayores inversiones en hardware.

Para la parte de la red de área local (LAN) se seleccionó la topología de red de tipo estrella debido a sus características de confiabilidad y eficiencia. Se usaron dos concentradores o **Hubs*** como núcleo de la misma y cableado de par trenzado CAT 5e el cual se remató en un panel de parcheo de 24 puertos CAT 5e.

Se instaló un centro de distribución para ubicar los siguientes equipos de comunicación de la red local: un panel de parcheo, dos concentradores, los CPUs y monitores de las máquinas que funcionan una como **Firewall** y la otra como servidor de dominio.

El estudio realizado en el capítulo uno nos llevó a utilizar la conexión de Windows® NT con Linux mediante la interface SAMBA para conjugar las cualidades de ambos sistemas operativos.

El dominio formado por esta conexión permite a los usuarios tener cuentas separadas ya que para entrar a este hay que introducir un nombre de usuario y su contraseña correspondiente. En este servidor se almacena la información de los alumnos en una partición dedicada a ello a la cual se hace referencia más adelante.

Las máquinas del LCD usan un software de ambiente gráfico conocido como Matlab; anteriormente los alumnos sólo podían conservar sus prácticas guardándolas en discos de 3 ½ pulgadas, a estas unidades se decidió eliminarlas para evitar la posibilidad de contaminarse a través de algún virus informático.

***** Ver glosario de términos, incluido en este documento

Con el desarrollo de este trabajo los programas realizados por los alumnos durante la clase práctica pueden ser abiertos desde una conexión externa a la red del LCD vía Internet, así como importar y modificar archivos desde un acceso remoto al servidor de dominio del LCD. Lo anterior redunda en un beneficio para los alumnos y profesores ya que el tiempo de clase es aprovechado de mejor forma.

Toda esta información será almacenada, como ya se mencionó, en una partición del servidor de Dominio, por esta razón la reparación o mantenimiento de alguna de las máquinas no afectan a los alumnos como anteriormente ocurría.

Se parte del principio de que los equipos de cómputo con los que se cuenta para este proyecto no son rápidos y eficientes, debido a lo anterior se tuvo cuidado en la selección del software a utilizar, escogiendo las versiones más adecuadas.

El acceso que tienen los usuarios a través de Internet a sus cuentas ubicadas en la partición **/home*** del servidor de dominio se implementó a través de programas realizados en **shell script*** y el lenguaje de programación JAVA, el cual es más poderoso que el lenguaje C++, sobretodo en cuanto a conectividad con Internet Se instaló **Tomcat*** que es una aplicación para servidor WEB.

Esta red se desarrolló utilizando la infraestructura con la que ya se contaba en el LCD, sin la necesidad de realizar una inversión grande en infraestructura de hardware, sin embargo se dan recomendaciones para su óptimo funcionamiento.

Se pensó en una solución integral sistemática y definitiva con el objeto de que cada semestre el administrador del LCD (o quien la coordinación determine) pueda realizar la carga de los nuevos grupos y profesores con el fin de continuar proporcionando a los profesores y alumnos las ventajas de acceder a sus prácticas y tareas desde Internet, así como de liberar el espacio utilizado en el servidor de dominio.

***** Ver glosario de términos, incluido en este documento

B) ® OBJETIVOS

Los objetivos que se persiguen en esta tesis se definen a continuación:

- Establecer una red LAN con 7 equipos inicialmente pudiendose ampliar hasta 12 equipos.
- Configurar una máquina auxiliar como **Firewall** y salida a Internet contando con una dirección IP real.
- Instalar un servidor de dominio con la interfase SAMBA cargada y configurarlo como servidor WEB para el acceso de los usuarios a través del Internet.
- Colocar un centro de distribución de cableado mediante el uso de un rack y dos concentradores.
- Cumplir adecuadamente los puntos anteriores a un costo muy bajo.

C) ® ESTADO INICIAL DEL LCD

Los problemas del LCD eran varios. En primer término, el sistema operativo de las computadoras instaladas para el uso de los alumnos (clientes) era Windows® 98, el cual, por su naturaleza no permite crear perfiles de acceso a los usuarios sobre los programas instalados en la máquina, por lo que el software instalado podía dañarse o borrarse. Esto generaba, dependiendo de los archivos dañados que el acceso al programa MATLAB se perdiera o inclusive que la carga completa del sistema operativo no se lograra, ambos casos eran extremadamente recurrentes. Para resolver estos problemas había que reinstalar el paquete afectado continuamente, pero si el daño era sufrido por algún archivo de configuración del sistema operativo la máquina se hacía inoperante, de ser así era necesario formatearla y los archivos con las prácticas de los alumnos que usaban esta máquina se perdían.

El programa MATLAB primeramente se instalaba en forma individual en cada una de las computadoras, el problema que se tenía es que la recarga del programa citado era difícil ya que ninguna computadora del LCD cuenta con unidad de CD-ROM .Posteriormente se intentó dar una solución temporal a esta situación y se montó un servidor Linux que daba acceso al

programa MATLAB a través de la interfase SAMBA, pero era demasiado inestable en su conexión y ambientación con Windows® 98 ocurriendo que a veces la interfase no daba acceso a MATLAB.

Como consecuencia no se podía impartir una clase práctica con todas las computadoras funcionando, lo cual generaba falta de interés tanto en los alumnos como en los profesores. En muchas ocasiones los profesores sólo podían impartir sus clases utilizando los programas CC o SIMNOM que por su diseño podían ejecutarse sólo en modo MS-DOS.

La falta de un sistema integral (computadoras cliente, servidor y red física) que controlara administrativa y operativamente el LCD le generaba muchos problemas a la coordinación ya que se distraía la atención en solucionar los recurrentes problemas generados en su interior.

Dadas las características del sistema operativo Windows® 98 instalado en las computadoras cliente, los alumnos no tenían un límite de utilización de los recursos de hardware, es decir, podían escribir en el disco duro de los equipos sin ninguna limitación ocasionando un desequilibrio en el uso de los recursos disponibles. Esto fue resuelto estableciendo cuotas de usuario en el servidor de dominio.

Todos los archivos guardados en las computadoras cliente eran de acceso libre a todo aquel que los solicitará, es decir, que cualquier alumno podría acceder al archivo que el quisiera aunque no fuera suyo y podía modificarlo o eliminarlo. Al darse cuenta de esto, los alumnos intentaban encontrar una alternativa para guardar en forma segura sus prácticas, un hábito muy común era que se utilizaran discos flexibles para respaldar sus trabajos y resguardarlos provocando en repetidas ocasiones que al introducir estos medios de almacenamiento externo en las computadoras se contaminaran con virus informáticos lo cual se reflejaba de inmediato en el rendimiento de los equipos.

Este trabajo está organizado de la siguiente manera: en el capítulo uno se presenta un análisis que permite evaluar las diferentes alternativas de solución, así como una descripción de las ventajas y desventajas de cada alternativa analizada; en el capítulo dos se describe la parte física del proyecto, el cableado estructurado y el centro de distribución, así como las normas de calidad utilizadas en el proyecto; en el capítulo tres se muestran los programas que forman la columna vertebral del acceso al sistema vía Internet y las diferentes tecnologías de software de

programación usados; en el capítulo cuatro aparecen las pantallas de captura para los usuarios agrupados por sus características de funcionamiento individuales ya sea un profesor , un usuario o el administrador del sistema y finalmente se presentan las conclusiones del trabajo.

1. EVALUACIÓN DE ALTERNATIVAS DE SOLUCIÓN

1.1® SISTEMAS OPERATIVOS

Sistema Operativo es el programa que maneja la interfase con los dispositivos de hardware, administra tareas, controla el uso de la memoria, además de que provee una interfase estandar cuando no se está ejecutando alguna aplicación.

Sus funciones son:

1. Coordinar y manipular el hardware de la computadora; como la memoria, las impresoras, las unidades de disco, el teclado o el mouse.
2. Organizar los archivos en diversos dispositivos de almacenamiento; como discos flexibles, discos duros, discos compactos o cintas magnéticas.
3. Gestionar los errores de hardware y la pérdida de datos

Los sistemas operativos más comunes son UNIX, Macintosh OS, MS-DOS, OS/2, la familia de productos de Microsoft (Windows 98, Windows 2000 y Windows NT, etc).

El UNIX y sus derivados permiten múltiples tareas y múltiples usuarios. Otros sistemas operativos multiusuario y multitarea son OS/2, desarrollado inicialmente por Microsoft e IBM, Windows NT y Win95 desarrollados por Microsoft. El sistema operativo multitarea de Apple se denomina Macintosh OS. El MS-DOS fue un sistema operativo popular entre los usuarios de PCs pero solo permite un usuario y una tarea.

A continuación se presentan las características de los sistemas operativos que se evaluaron para esta tesis:

1.1.1 ® SISTEMA LINUX

Linux es un sistema operativo. Es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Fue desarrollado para el i386 y ahora soporta los procesadores i486, Pentium®, Pentium Pro y Pentium® II, así como los AMD® y Cyrix®. También soporta máquinas basadas en SPARC®, DEC® Alpha, PowerPC/PowerMac, y MAC®/Amiga Motorola® 680x0.

Como sistema operativo, **Linux** es muy eficiente y tiene un excelente diseño. Es multitarea, multiusuario, multiplataforma y multiprocesador; en las plataformas Intel® corre en modo protegido; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; usa hasta 64 consolas virtuales; tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas; y soporta redes tanto en TCP/IP como en otros protocolos.

1.1.1.1 ® LINUX - BREVE RESEÑA HISTÓRICA

Linux fue creado originalmente por Linus Benedict Torvalds en la Universidad de Helsinki en Finlandia. Este ha sido desarrollado y mejorado con la ayuda de muchos programadores a través de Internet (**Open System*.**)

Linus originalmente inició el **hacking*** del núcleo como su proyecto favorito, inspirado por su interés en **MINIX***, un pequeño sistema basado en UNIX.

Linus se propuso a crear lo que en sus propias palabras seria un mejor Minix que el Minix. El 5 de octubre de 1991, Linus anunció su primera versión "oficial" de **Linux**, versión 0.02. Desde entonces, muchos programadoras han respondido a su llamado, y han ayudado a construir **Linux** como el sistema operativo completamente funcional que es hoy.

***** Ver glosario de términos, incluido en este documento

La última versión estable es la versión 2.2, que soporta muchos más periféricos, desde procesadores hasta joysticks, sintonizadores de televisión, CD ROM's y reconoce buena cantidad de tarjetas de sonido. Incluye también soporte para tipos de archivos para **Macintosh® HFS***, Unix UFS* y en modo de lectura, **HPFS*** de OS/2 y **NTFS*** de Windows® NT.

En la actualidad, este sistema operativo es utilizado por miles de usuarios para desarrollo de software, redes y para plataformas de usuarios finales. **Linux**, entre los muy diversos sistemas operativos alternos que existen, se ha convertido en una opción interesante.

1.1.1.2 ® LINUX - PRINCIPALES CARACTERÍSTICAS

Linux sigue con el modelo de desarrollo abierto. Se desarrolla por un grupo no estructurado de programadores de todo el mundo unidos a través de la Internet. Todas las nuevas versiones se liberan al público, se consideren o no con calidad de producción. El método que se sigue para determinar si se tiene una versión estable o no es el siguiente: Las versiones se numeran como x.y.z; aquellas con y par son versiones estables; aquellas con y impar son versiones de desarrollo. Por ejemplo, la versión 1.2.13 es estable; la 2.1.44 es de desarrollo. La última versión estable es la 2.2.10. Sin embargo, este modelo de desarrollo abierto mantiene siempre la última versión disponible, y las nuevas versiones se liberan cada pocas semanas. En ocasiones las funciones nuevas contienen errores, pero con cientos de personas en Internet probando y leyendo el código fuente suelen corregirse en cuestión de horas.

Todo esto resulta en un sistema de alta calidad tecnológica, con menos errores que los sistemas comerciales, a un costo cero o muy bajo y con la disponibilidad del código fuente que permite aprender, modificar o ayudar al desarrollo del sistema.

Para tener un sistema con todos los comandos importantes y una o dos aplicaciones pequeñas se requieren alrededor de 10 MB de disco duro.

***** Ver glosario de términos, incluido en este documento

Para un sistema más completo, se aconsejan 4 MB de memoria, u 8 si se piensa utilizar una interface gráfica. Si se van a tener muchos usuarios y/o muchos procesos a la vez, sería aconsejable hasta 16 MB. 32 MB es más que suficiente para cargas pesadas a un máximo rendimiento.

En lo que respecta a disco duro, depende de las aplicaciones que se instalen, se va desde los 10 MB básicos hasta los 350 MB de una distribución, instalado varias aplicaciones (incluye compiladores, paquetes de oficina, interface gráfica, etc.). Obviamente, un procesador más veloz siempre será ventajoso. El coprocesador matemático nunca es requisito, pero acelera aquellas aplicaciones de cálculo de punto flotante intensivo

Linux está disponible en Internet en cientos de servidores FTP* y en distribuidores en discos CD-ROM de revendedores que lo ofrecen empacado con manuales e información que es realmente la del costo, pues el programa es gratuito. Algunos de estos son: Caldera*, Suse*, Red Hat*, etc. Uno de los servidores más populares que ofrecen **Linux** está ubicado en ftp://sunsite.unc.edu/pub/Linux/distributions, con una gran cantidad de "mirrors"* alrededor del mundo. El núcleo del **Linux** está legalmente protegido por la licencia publica GNU* (GPL), esta licencia se encarga de que **Linux** permanezca siempre libre.

Linux incluye compiladores, ensambladores, debuggers, editores de texto, paquetes de email, lectores de noticias, navegadores, servidores y programas para la creación y edición gráfica.

Linux maneja los archivos de forma jerárquica, de la misma forma que MS-DOS, con la diferencia que el DOS está diseñado para procesadores x86 que no soportan verdaderas capacidades de múltiples tareas.

A últimas fechas la distribución y comercialización de este sistema operativo ha crecido mucho debido al desarrollo de una interfaz gráfica, muy similar a la que la mayor parte de la gente está acostumbrada a utilizar.

***** Ver glosario de términos, incluido en este documento

1.1.1.3 ® LINUX - VENTAJAS

Linux es una muy buena alternativa frente a los demás sistemas operativos. Más allá de las ventajas evidentes de costo, ofrece algunas características muy notables.

En comparación con las otras versiones de Unix para PC, la velocidad y confiabilidad de **Linux** son muy superiores. También está en ventaja sobre la disponibilidad de aplicaciones, ya que no hay mucha difusión de estos otros Unix (como Solaris®, XENIX® o SCO®) entre los usuarios de PC por sus altos costos.

Comparado con sistemas operativos como los diferentes Microsoft® Windows®, **Linux** también sale ganando. Los bajos requisitos de hardware permiten hacer un sistema potente y útil de aquel 486 que algunos guardan en un armario. Esta misma característica permite aprovechar al máximo las capacidades de las computadoras más modernas. Es poco práctico tener una PC con 16 Mb de RAM y ponerle un sistema operativo que ocupa 13 (que es lo que reporta sobre Windows 95 el System Information de Symantec®). No solo es superior respecto a el sistema de multitarea y de administración de memoria, sino también en la capacidades de networking (conectividad a redes) y de multiusuario (aún comparando con sistemas multiusuario como NT). La única desventaja de **Linux** frente a estos sistemas, es la menor disponibilidad de software, pero este problema disminuye con cada nuevo programa que se escribe para el proyecto GNU, y con algunas empresas que están desarrollando software comercial para **Linux.**

Utilizar un sistema operativo con arquitectura abierta nos permite tener muchísimas ventajas, sobre todo económicas, cuando menos a nivel amateur, ya que a últimas fechas empresas como Microsoft® han lanzado campañas publicitarias indicando que es más barato tener servidores de aplicaciones bajo su plataforma (Windows® 2003 Server) que sobre versiones Enterprise de **Linux**, pero esto solamente puede ser determinado en la práctica, dadas estas circunstancias las siguientes ventajas son mencionadas exclusivamente para nuestro proyecto.

1. Precio.
2. Estabilidad
3. Seguridad, es mucho más seguro que otros servidores, aunque no es un sistema perfecto, hasta el momento no se han desarrollado virus informáticos que lo afecten.
4. Compatibilidad,reconoce la mayoría de los otros sistemas operativos en una red.

5. Velocidad, es mucho mas veloz para realizar las tareas.
6. Posee el apoyo de miles de programadores a nivel mundial.
7. El paquete incluye el código fuente, lo que permite modificarlo de acuerdo a las necesidades del usuario.
8. Ideal para la programación, ya que se puede programar en **Linux** para distintas plataformas, como para Windows.
9. Un sistema de crecimiento rápido.
10. Se puede usar en forma eficiente en casi cualquier computadora, desde una 386, en nuestro caso uno de los servidores en una 486.
11. Multitareas REAL.
12. Puede manejar múltiples procesadores. Incluso hasta 16 procesadores.
13. Maneja discos duros de hasta 16 **TeraBytes***.
14. Se consiguen parches con facilidad, además de ser gratuitos.
15. Se posee el apoyo de millones de usuarios a nivel mundial.
16. Los fabricantes de Hardware le están dando su apoyo, como IBM® y COMPAQ®.
17. Vendedores y desarrolladores implementan un sistema de certificación para **Linux**.
18. La corporación DATA Internacional predice que el crecimiento de este programa será del orden de un 25 por ciento anual.

Para cada una de las ventajas descritas anteriormente podríamos hacer un capítulo entero si nos involucramos a detalle, pero debido a que el objetivo no es analizar cada una de ellas tan a fondo, preferimos dejarlas solo como una referencia superficial.

1.1.1.4 ® LINUX - DESVENTAJAS

1. **Linux** no cuenta con una empresa que lo respalde, por lo que no existe un verdadero soporte como el de otros sistemas operativos. Aunque hoy en día existen compilaciones profesionales que ofrecen soporte la opción de contar con soporte técnico, este no se puede llegar a comparar con sistemas de código cerrado.

2. Al ser un sistema operativo de código totalmente abierto, **Linux** corre el riesgo de llegar a fragmentarse como fue el caso de UNIX.

***** Ver glosario de términos, incluido en este documento

3. Algunas empresas pueden llegar a ayudar a **Linux** con la intención de mejorar sus relaciones públicas, aunque en el fondo no tengan ninguna intención de utilizarlo fielmente.

Estos son algunos sitios en los cuales se puede conocer aún más de este sistema operativo:

The Linux Home Page. El sitio oficial de **Linux** con todos los recursos sobre este sistema operativo disponibles.

Linux Preview. Un sitio Web para el usuario serio de **Linux** con muchas información actualizada regularmente. Proyecto Lucas. Biblioteca en español dedicada a **Linux** de todo el planeta. El rincón linuxero de Fidonet. Información y recursos sobre **Linux** en Fidonet. HispaLinux La asociación de usuarios españoles de **Linux**. Asociación Informática Abierta. Entidad sin fines de lucro que difunde el conocimiento de las nuevas tecnologías y promueve el uso de software libre.

1.1.2. ® SISTEMA UNIX

Bajo el término **Unix**, se hace referencia no a un solo sistema operativo sino a un grupo amplio de sistemas. Por este motivo haremos una breve reseña histórica para incluir el desarrollo de algunos de estos sistemas, así como de su evolución.

A pesar que hoy en día Windows® es el sistema operativo más popular, resulta importante conocer otros sistemas operativos como es el caso de UNIX que a pesar de ser un sistema “más difícil” que Windows® es mucho más eficiente que éste en el uso de los recursos.

1.1.2.1 ® UNIX – BREVE RESEÑA HISTÓRICA

Al final de la década de los 60 AT&T®, General Electric® y MIT comienzan un proyecto conjunto en respuesta a la necesidad de producir un sistema operativo que permitiera múltiples usuarios. El concepto requería un sistema donde se pudieran compartir archivos y recursos manteniendo la privacidad de cada usuario. Luego en AT&T® Bell Laboratorios®, varios programadores comenzaron a desarrollar un sistema operativo como una herramienta de ayuda en sus labores, lo llamaron "MULTICS", llamado luego UNIX. AT&T® comenzó a propagar **Unix** por las universidades y pronto se convirtió el favorito de la gente de ciencias de computación y

de investigadores. Hoy día hay muchísimas variantes del sistema operativo y es uno de los más utilizados en ambientes de trabajo de alto rendimiento.

Variantes de **Unix**

SYSTEM V (AT&T)
ULTRIX (DIGITAL, ahora Compaq)
HP UX (Hewlett-Packard)
AIX (IBM)
SCO (Santa Cruz Operation)
IRIX (Silicon Graphics)
BSD (Universidad de Berkley)
SOLARIS (SUN Microsystems)
LINUX

Actualmente BSD y SYSTEM V se han convertido en los estándares más utilizados en UNIX, todos los sistemas de alguna u otra forman utilizan uno de estos dos estándares.

A continuación enlistamos los acontecimientos históricos más importantes del desarrollo de este sistema operativo.

- 1969 El **Unix** nace en los laboratorios Bell, creado por Ken Thompson. Se desarrollo sobre un equipo DEC PDP-4.
- 1970 Thompson reescribe todo el sistema en "Lenguaje B". Dennis Ritchie lo desarrolla y denomina "Lenguaje C", resscribiendo todo el **Unix** en este lenguaje.
- 1973 El **Unix** es transportado a distintos ordenadores, naciendo el **Unix** versión 5.
- 1974 AT&T cede la licencia de explotación del **Unix** a Universidades y centros de investigación. Nace entonces la versión 6.
- 1977 se crea la versión 7.
- 1981 Se incluyen numerosas utilidades. Lanzamiento comercial del **Unix** System III.
- 1982 AT&T* comercializa el **Unix** y presenta el System V.
- 1988 Se creó la OSF (Open Software Fundation) para elaborar una norma universal **Unix** .

1.1.2.2 ® UNIX – CARACTERÍSTICAS BÁSICAS

El **Unix** es un sistema operativo de propósito general con las siguientes características :

1. Sistema Multiusuario.
2. Sistema Multitarea.
3. Trabaja en tiempo compartido.
4. Sistema interactivo.
5. Estandarizado.
6. Potente.
7. Versátil.
8. Software y sistema operativo portable.
9. Permite ejecutar procesos en Background y Foreground.
10. Presenta niveles altos de seguridad.
11. Presenta una estructura jerárquica de ficheros.
12. Posee un gran número de utilidades : Editores de línea , de pantalla, formateadores, depuradores de programas...
13. Posibilidad de comunicación Unix - Unix (UUCP)
14. Fácil integración en sistemas de red.
15. Posibilidad de usar distintos lenguajes de programación: C, Fortran, Cobol, Pascal,... Ø Gran cantidad de software disponible en el mercado.

Unix es un sistema operativo multiusuario y multitarea, que corre en diferentes computadoras, desde supercomputadoras, **Mainframes***, Minicomputadoras, computadoras personales y estaciones de trabajo.

Es un sistema operativo que fue creado a principios de los setentas por los científicos en los laboratorios Bell. Fue específicamente diseñado para proveer una manera de manejar científica y especializadamente las aplicaciones computacionales. Este sistema operativo se adapto a los sistemas de cómputo personales así que esta aceptación reciente lo convierte en un sistema popular. **Unix** es más antiguo que todos los demás sistema operativo de PC y de muchas maneras sirvió como modelo para éstos. Aun cuando es un sistema operativo extremadamente sólido y capaz, la línea de comandos Unix, no es apta para cardiacos, debido a que ofrece demasiados comandos.

1.1.2.3 ® UNIX – COMPONENTES

El sistema operativo **Unix** tiene dos componentes fundamentales:

- El Kernel.
- El Shell.

El KERNEL · Constituye el núcleo del sistema operativo. Actúa como interfase con el hardware del sistema. Se encarga de realizar las funciones de más bajo nivel, como son :

1. Manejo y control de los dispositivos físicos.
2. Control de interrupciones.
3. Control de procesos y tiempos.
4. Control de ficheros.
5. Control de memoria.

Está escrito fundamentalmente en lenguaje "C" (90%) y el resto en lenguaje ensamblador.

El SHELL

- Interfase del sistema operativo con los usuarios del mismo.
- Actúa como interprete de comandos.
- Características semejantes a un lenguaje de programación de alto nivel. Shellscripts. · Permite configurar diferentes entornos de trabajo.
- Es un comando : /sbin/sh.

1.1.2.4 ® UNIX – VENTAJAS

1. UNIX es un sistema operativo muy conveniente desde el punto de vista de la eficiencia y aprovechamiento de los recursos.
2. Aunque el sistema operativo UNIX tiene algunas deficiencias, la mayor parte de estas pueden rectificarse utilizando los instrumentos que el mismo UNIX proporciona.
3. El enfoque singular que el sistema adopta hacia los problemas de estandarización y portabilidad, su poder como herramienta de desarrollo y su interfaz de usuario tan variada, hacen que sobresalga como la opción estándar para usuarios
4. *Portabilidad* Unix o alguna variante de UNIX están disponibles en muchas máquinas. Los programas escritos con UNIX y C estándares deben correr en cualquier máquina prácticamente sin ningún problema.
5. *Multiusuario/Multitarea* Muchos programas pueden compartir la capacidad de procesamiento de las máquinas.
6. *Manejo de archivos* El sistema jerárquico de archivos emplea muchas rutinas de manejo de archivos.
7. P*rogramación del Shell* UNIX suministra un intérprete de comandos poderoso que entiende mas de 200 comandos y que puede también correr UNIX o programas del usuario.
8. *Entubamiento o Pipe* Permite la conexión entre programas, en donde la salida de un programa puede ser la entrada de otro. Lo anterior puede hacerse desde la línea de comandos o dentro de un programa de C.
9. *Utilerías de UNIX* Hay cerca de 200 utilerías que permiten ejecutar muchas rutinas sin escribir nuevos programas. Por ejemplo: make, grep diff, awk, etc.
10. *Llamadas al sistema* UNIX tiene aproximadamente 60 llamadas al sistema, que son el corazón del sistema operativo o del *kernel* de UNIX. Las llamadas están actualmente escritas en C. Todas ellas pueden ser accesadas desde programas de C. Ejemplos de estas son el sistema básico de E/S, acceso al reloj del sistema. La función open() es un ejemplo de una llamada al sistema.
11. *Biblioteca de funciones* Que son adiciones al sistema operativo

1.1.2.5 ® UNIX – DESVENTAJAS

1. Hemos podido apreciar que lo que se vende hoy en realidad tiene que ver con la facilidad del trabajo, que las personas prefieren la comodidad antes que la eficiencia.
2. El sistema es extremadamente común en ambientes educacionales, Universidades por ejemplo, pero debido a diversos factores no ha tenido una expansión comercial.
3. Hay pocos especialistas en él.
4. Las versiones comerciales mencionadas anteriormente son caras.

1.1.3 ® SISTEMA MS-DOS®

Fue por mucho tiempo el más común y popular de todos los sistemas operativos para PC. La razón de su continua popularidad se debe al aplastante volumen de software disponible y a la base instalada de computadoras con procesador Intel®.

Cuando Intel® liberó el 80286, D OS se hizo tan popular y firme en el mercado que DOS y las aplicaciones DOS representaron la mayoría del mercado de software para PC. En aquel tiempo, la compatibilidad IBM®, fue una necesidad para que los productos tuvieran éxito, y la "compatibilidad IBM®" significaba computadoras que corrieran MS-DOS® tan bien como las computadoras IBM® lo hacían.

El significado de estas letras es el de MicroSoft Disk Operating System. Microsoft es el nombre de la compañía que diseño este sistema operativo, e IBM la compañía que lo hizo estándar al adoptarlo en sus microordenadores.

Este sistema operativo emplea discos flexibles con una organización determinada. Los discos se pueden grabar por una o por dos caras y la información se organiza en 40 pistas de 8 ó 9 sectores de un tamaño de 512 caracteres, reservándose el sistema para la propia información del disco, que puede ser disco removible o disco duro, teniendo en el segundo más capacidad pero similar estructura.

Este sistema operativo emplea discos flexibles con una organización determinada. Los discos se pueden grabar por una o por dos caras y la información se organiza en 40 pistas de 8 ó 9 sectores de un tamaño de 512 caracteres, reservándose el sistema para la propia información

del disco, que puede ser disco removible o disco duro, teniendo en el segundo más capacidad pero similar estructura.

Los de extensión .EXE precisan que el cargador del DOS los coloque en memoria, lo que significa que el sistema operativo debe estar en memoria. Los del tipo .BAT son los compuestos de comandos que se ejecutan secuencialmente.

1.1.3.1 ® MSDOS® – BREVE RESEÑA HISTÓRICA

El progenitor más directo del MS-DOS fue un sistema operativo llamado 86-DOS, escrito por Tim Paterson de la Seattle Computer Products hacia la mitad de 1980.

En aquel tiempo el CP/M-80 de Digital Research era el sistema operativo aplicado más comúnmente a microordenadores, y existía un cierto número de programas de aplicación (procesadores de texto, gestores de bases de datos, etc) que funcionaban bajo este sistema operativo. Con el fin de facilitar el proceso de transporte de aplicaciones CP/M-80 de ocho bits a los nuevos entornos de 16 bits, se creó el 86-DOS empleando instrucciones que imitaban las prestaciones del CP/M-80, tanto en las funciones disponibles como en el estilo de funcionamiento.

En Octubre de 1980, IBM® tomó contacto con las principales casas de software para microordenadores, buscando un sistema operativo para la nueva línea de ordenadores personales que estaba diseñando.

Microsoft® no disponía en ese momento de ningún sistema operativo propio, pero pagaba a Seattle Computer Products una autorización para vender el 86-DOS de Paterson. Por aquel tiempo, Seattle Computer Products disponía a su vez de una licencia para utilizar y vender los lenguajes Microsoft®.

En Julio de 1981, Microsoft® compró todos los derechos a 86-DOS, realizó alteraciones sustanciales en el sistema original y lo bautizó MS-DOS®. Cuando apareció el primer IBM® PC en otoño de 1981, IBM® ofreció el MS-DOS® (llamándole PC-DOS 1.0) como sistema operativo primario.

Esta versión inicial del DOS® (1.0) suministró varias órdenes básicas para el usuario. Lo más importante fue que introdujo el sistema de gestión de archivos que aún hoy día usan el DOS y el OS/2. Este sistema de archivos está fundamentado en la tabla de ubicación de archivos (FAT), todas las versiones DOS desde la 1.0 han mantenido la compatibilidad de archivos. Esto significa que el DOS 3.x y el OS/2 pueden leer incluso archivos creados con el DOS 1.0. Aunque el conjunto de órdenes incluido en el DOS 1.0 parece muy limitado a la luz de los sistemas operativos actuales, las previsiones de los diseñadores del DOS 1.0 han sido incomparables.

En 1982 Microsoft® mejoró el DOS de forma que admitiese disquetes de doble superficie. Se trataba dela versión 1.1del DOS.

En 1983 IBM® y Microsoft®, conjuntamente, lanzaron el DOS 2.0. En esta ocasión se toma la decisión de que el DOS siga al UNIX en sus aspectos funcionales. El DOS 2.0 gestiona completamente subdirectorios DOS, encauzamientos DOS, y un conjunto de órdenes mejorado. Además el DOS 2.0 dispone de la facultad de trasladar bloques de control de archivos (FCBs) a archivos manipulables, que permite la implantación de redireccionamiento de E/S.

A lo largo de 1983 tanto IBM® como Microsoft® lanzaron mejoras del DOS menores, principalmente para eliminar imperfecciones del sistema. Una explicación rápida para familiarizarnos con los números de versiones del DOS sería, que cada número de versión del DOS contiene dos partes, un número principal de versión y otro subsidiario. Por ejemplo, en el DOS 3.2, 3 es el número principal de versión, y 2 es el secundario.

La mayoría de los desarrolladores de software siguen esta convención cuando van mejorando su software; si la actualización es una mejora importante del paquete software, incrementan el número principal de versión (2.0 cambiaría a 3.0). Si los cambios son secundarios (tal como depurar un error), incrementa el número secundario de versión(la 2.0 pasaría a ser la 2.1). En 1.983 se comercializaron las versiones 2.0, 2.01, 2.10, 2.11 y 2.25 del DOS.

En 1.984, con el advenimiento del IBM PT AT, conjuntamente Microsoft e IBM lanzaron el DOS 3.0. El DOS 3.0 admite la gestión de unidades de disquetes de 1.2 MB y discos fijos de mayor capacidad. El mismo año ambas compañías lanzaron el DOS 3.1, que contiene funciones para conexión a una red de área local. Aunque los cambios funcionales y globales fueron menores, el DOS 3.1 cambió gratamente el aspecto del DOS e incluyó la posibilidad de ejecuciones en segundo plano para servidores multiusuarios y puestos de trabajo. Habiéndose lanzado sólo tres

años antes como sistema básico monousuario, el DOS ya comienza a competir con el UNIX y el XENIX en entornos multiusuario.

En 1985 salió el 3.1, que añadía soporte para redes de área local.

Un avance importante con computadoras ocurrió en 1.986. Para hacer máximo el espacio de almacenamiento a la vez que se reducía el espacio de la unidad de disco, los sistemas superiores usaban disquetes de 3.5 pulgadas. Conjuntamente IBM y Microsoft introdujeron el 3.2 para admitir disquetes de 3.5 pulgadas.

En abril de 1.987 IBM anunció la serie de computadoras PS/2 (“Personal System/2”). Esta línea de computadoras contiene las nuevas mejoras tecnológicas de la línea 8088, 80286 y 80386 de microprocesadores. A la vez IBM lanzó el DOS 3.3, que es requerido por la serie PS/2, y Microsoft e IBM anunciaron el OS/2, un sistema operativo para entornos de multiprogramación dentro de la línea de microprocesadores 80286 y 80386.

La versión 4.0 apareció en noviembre de 1988 y gestiona discos duros de particiones de más de 32 MB (hasta 512 MB). Además, dispone de una nueva interfase gráfica y soporte de memoria expandida, esta versión permite además el empleo de la memoria expandida del ordenador (anteriormente sólo se podían emplear 640 Kb de memoria RAM). Actualizar a la versión 4.0 desde una versión anterior puede traer dificultades ya que habrá que reformatear el disco duro bajo la nueva versión y es posible que algunos programas necesiten funciones de DOS que ya no estén disponibles.

La versión 5.0 salió en junio de 1991, y proporciona drivers para gestionar ampliaciones de memoria y se incorpora un editor de pantalla y un shell bastante potente, además de poder instalarse independientemente de la versión anterior de sistema operativo.

La versión 6.0 se lanzó en abril de 1993. Como contenía abundantes errores fue sustituida el mismo año por la versión 6.2. Las mejoras de la versión 6.0 incluyen: herramientas de compresión de discos, antivirus, programas de copias de seguridad por menú, desfragmentador de disco y otras utilidades, como por ejemplo un administrador de memoria ampliada, denominado MemMaker.

A finales de 1993 se sacó al mercado la versión 6.2. Esta nueva versión incluía mejoras en el duplicador de espacio en disco y la posibilidad de borrar un directorio independientemente de su contenido, entre otras ventajas.

1.1.3.2 ® MSDOS® – SUS COMPONENTES

El sistema operativo tiene varios componentes que son:

- - Rutinas de control, que funcionan con el programa IBM.DOS, y se encargan de las operaciones de entrada / salida.
- - Procesador de comandos, también llamado COMMAND.COM, que procesa los dos tipos de comandos de que dispone el DOS; es decir, los residentes en memoria o internos, y los no residentes o externos, que residen en el disco del sistema operativo.
- - Rutinas de servicios accesibles desde el programa control.

También existe la posibilidad de subdividir el disco en subdirectorios que permiten un empleo más ágil de toda la información.

MS-DOS® esta lejos de ser el sistema operativo ideal, ya que, de momento, se trata de un sistema monotarea, pero aunque esto se resolviera, seguiría presentando problemas de diseño que provocan que el comportamiento de la máquina sea poco fiable. A pesar de estas desventajas y de que existen otros sistemas operativos en el mundo de la microinformática, hay que tener siempre presente la enorme cantidad de software que se ha desarrollado para DOS y que conviene aprovechar en lo posible.

1.1.3.3 ® MSDOS® – VENTAJAS

Una gran ventaja del MS-DOS sobre otros sistemas operativos es que ofrece una amplia gama de comandos, que cubren todas las arreas de trabajo con el ordenador. Así, operaciones que se debían realizar con programas externos o complicadas secuencias de ordenes, se realizan en el MS-DOS con comandos estándar del mismo, pudiendo disponer de:

1. Comandos de tratamiento de ficheros, que permiten realizar desde la creación de un fichero, hasta la comparación de los contenidos de dos ficheros.
2. Comandos de salida de ficheros, con opciones para hacer la salida por diferentes dispositivos.
3. Comandos de tratamiento y mantenimiento de discos, incluyendo ordenes muy útiles para la revisión y recuperación de discos dañados
4. Comandos para manejo de parámetros del sistema, con posibilidades de actualizar diferentes valores de trabajo del MS-DOS
5. Comandos para manejo de directorios, capaces de trabajar en la estructura jerárquica de directorios que ofrece el MS-DOS
6. Comandos internos para los ficheros “batch”. Estos comandos, a los que podríamos llamar “pseudo-instrucciones”, sirven para estructurar el contenido de un fichero batch de modo que se puedan ejecutar comandos alternativamente, de forma selectiva, etc.
7. Comandos de utilidad, con misiones tan diversas como ordenar ficheros o borrar pantallas.
8. Comandos para programación, herramientas útiles para el trabajo con lenguaje ensamblador

1.1.3.4 ® MSDOS® – DESVENTAJAS

1. Sistema muy poco intuitivo.
2. Es un producto que muy poca gente utiliza actualmente.
3. Técnicamente es limita, cuando menos para el objetivo que buscamos.
4. No se puede ejecutar en modo protegido, modo en el cual no se pueda acceder directamente a la memoria.
5. No maneja seguridad, este sistema operativo permite acceder hasta el corazón del ordenador.

1.1.4 ® SISTEMA IBM® OS/2

OS/2 son las siglas en inglés de sistema operativo de segunda generación. La idea de OS/2 surgió entre IBM y Microsoft a mediados de los 80, en un intento de hacer un sucesor de MS-DOS, el cual ya empezaba a acusar el paso del tiempo y desaprovechaba los recursos de las máquinas de la época (basadas en el Intel 286). Sin embargo, la sociedad no duró mucho. Las diferencias en opiniones técnicas y la percepción de IBM al ver a Windows como una amenaza para el OS/2 causó una desavenencia entre las compañías que al final las llevó a la disolución de la sociedad. Entonces IBM continuó el desarrollo y promoción del OS/2.

El OS/2 al principio fue muy parecido al MS-DOS, tiene una línea de comando, pero la diferencia que existe con el DOS es el intérprete de comandos, el cual es un programa separado del kernel del sistema operativo y aparece únicamente cuando se hace clic en uno de los iconos “OS/2 prompt” dentro del Workplace Shell. Otra diferencia es que este sí en un sistema operativo multitarea.

1.1.4.1 ® IBM OS/2 – BREVE RESEÑA HISTÓRICA Y CARACTERÍSTICAS PRINCIPALES

OS/2 1.0 salió en abril de 1987 y era un sistema operativo de 16 bits, pues estaba pensado para trabajar sobre el microprocesador 286. Sin embargo, aprovechaba plenamente el modo protegido de este ordenador, haciendo uso de sus capacidades para protección de memoria, gestión de multitarea, etc. El resultado fue un sistema operativo estable, rápido y muy potente.

OS/2 ya tenía incorporada desde esa primera versión la multitarea real. Se podían ejecutar varias sesiones simultáneamente, en cada una de ellas se podían tener múltiples programas, y cada uno de ellos podía tener múltiples threads en ejecución. Se trataba de una multitarea jerárquica, con cuatro niveles de prioridad: Crítico (útil para programas que requieran atención casi constante por parte de la CPU, como un módem), Primer plano (correspondiente al programa que tiene acceso a la pantalla, teclado y ratón), Medio (programas lanzados por el usuario que se ejecutan en BackGround) y Desocupado (tareas de poca importancia o lentas, como el Spooler de impresión). Dentro de cada nivel (a excepción del de Primer plano), existen

32 niveles de prioridad, los cuales son asignados dinámicamente a cada programa por el sistema operativo en función del porcentaje de uso de la CPU, de los puertos de E/S, etc.

OS/2, además, permitía memoria virtual, con lo que se podían ejecutar programas más largos que lo que la memoria física instalada permitiría en principio (los requerimientos de aquella versión eran un 286 con 2 megas de memoria). Por otro lado, incluía la característica de compartición de código: al cargar dos veces un mismo programa, el código de este no se duplicaba en memoria, sino que el mismo código era ejecutado por dos Threads diferentes. Esto permitía ahorrar mucha memoria.

Esta versión de OS/2 era íntegramente en modo texto. Si bien el Sistema Operativo daba la posibilidad de usar los modos gráficos de la tarjeta del ordenador, no incluía ningún API que ayudase en ello, recayendo todo el trabajo de diseño de rutinas de puntos, líneas, etc en el programador de la aplicación. Esto no era realmente tan problemático, pues era lo que se hacía (y se hace) en el mundo del MS-DOS. Sin embargo, se echaba en falta un entorno gráfico como Windows.

En la versión 1.1, aparecida en octubre de 1988, llegó por fin el Presentation Manager, un gestor de modo gráfico, junto con la primera versión de Work Place Shell.

El gran problema de OS/2 es que seguía siendo un sistema operativo de 16 bits, con lo que no aprovechaba plenamente las capacidades de los 386 de la época, que empezaron a extenderse con más velocidad de la esperada. Según una revista del sector, Microsoft sugirió hacer una versión de 32 bits (que obligaría a ejecutarla en ordenadores 386 o superiores), pero IBM insistió en perfeccionar la de 16 bits. Sobre quien dijo cada cosa realmente solo se puede especular. Lo único que se sabe a ciencia cierta es que la versión de OS/2 de 32 bits presentada por MicroSoft en 1990 era casi igual que la versión 1.3, con la única diferencia de que el kernel era de 32 bits. IBM, por su parte, quería un escritorio orientado a objetos, y no el clásico shell de OS/2 1.x (el cual MicroSoft copiaría para su Windows 3.0). Puestas así las cosas, finalmente se rompió el acuerdo entre ambos.

OS/2 2.0, la primera versión de OS/2 de 32 bits, iba a salir inicialmente a finales de 1990; pero al no contar con la ayuda de Microsoft, IBM no fue capaz de sacarlo hasta 1992, dándole a Windows 3.0 el tiempo suficiente para asentarse en el mercado

A principios de 1994 aparece OS/2 Warp, nombre comercial de la versión 3.0 de OS/2. En ella surgen nuevos elementos: un kit completo de multimedia (mejora del que traía la versión 2.1) y el Bonus Pak, un kit de aplicaciones que permite ponerse a trabajar con el ordenador nada más instalar el Sistema Operativo, pues contiene elementos como un kit de conexión a Internet completo, el paquete integrado IBM Works (formado por un procesador de textos, hoja de cálculo, base de datos y gráficos de empresa, junto con el PIM, que añade más funcionalidades aprovechando las capacidades drag&drop del WPShell), soft de terminal, soft de captura y tratamiento de video, etc. Así mismo, la cantidad de hardware soportado fue ampliado de manera considerable, soportando casi cualquier dispositivo existente en el mercado: CD-Roms, impresoras, tarjetas de sonido, soporte PCMCIA, tarjetas de video, tarjetas de captura de video, tarjetas SCSI, etc. Los requisitos mínimos de esta versión seguían siendo un 386sx a 16MHz con 4 megas de RAM, los mismos que Windows 3.11, y podía ejecutar programas DOS, OS/2 16bits, OS/2 32 bits, Windows 2.x y Windows 3.x (incluía además el API Win32s, con lo que se podían ejecutar incluso programas Windows de 32bits).

IBM se metió en una campaña publicitaria a nivel mundial para promocionar esta nueva versión, la cual, sin embargo, no dio los resultados esperados por ser demasiado ligero. A pesar de eso, OS/2 es ampliamente utilizado en múltiples empresas, bancos sobre todo, en donde su estabilidad es la mayor garantía (los propios cajeros automáticos funcionaban inicialmente con OS/2 1.0, si bien actualmente usan OS/2 Warp).

En Noviembre de 1996 se hizo la presentación de **Merlín**, nombre clave de OS/2 4.0, y que, en contra de lo que mucha gente piensa, no tiene nada que ver con el mítico mago de la corte del rey Arturo, sino con un pájaro parecido a un águila (siguiendo la nueva filosofía de IBM de nombrar sus creaciones con nombres de aves). Merlín trae todo lo que ofrecía OS/2 3.0.

Es un sistema operativo de multitarea para un solo usuario que requiere un microprocesador Intel 286 o mejor. Además de la multitarea, la gran ventaja de la plataforma OS/2 es que permite manejar directamente hasta 16 MB de la RAM (en comparación con 1 MB en el caso del MS-DOS). Por otra parte, el OS/2 es un entorno muy complejo que requiere hasta 4 MB de la RAM. Los usuarios del OS/2 interactúan con el sistema mediante una interfaz gráfica para usuario llamada Administrador de presentaciones. A pesar de que el OS/2 rompe la barrera de 1 MB del MS-DOS, le llevo tiempo volverse popular. Los vendedores de software se muestran renuentes a destinar recursos a la creación de un software con base en el OS/2 para un mercado dominado por el MS-DOS. Los usuarios se niegan a cambiar al OS/2 debido a la falta de

software que funcione en la plata forma del OS/2 y a que muchos tendrían que mejorar la configuración de su PC para que opere con el OS/2.

En el OS/2 muchos de los comandos son idénticos a los de su contra parte pero tiene más comandos debido a que es más grande, completo y moderno

El ambiente gráfico es el Workplace Shell (WS), es el equivalente a un administrador del área de trabajo para el WS.

1.1.4.2 ® IBM OS/2 – VENTAJAS

No tienes que preocuparte acerca de la seguridad de tus conexiones a Internet. No hay agujeros de seguridad

1. Mejor compatibilidad con las aplicaciones de DOS y Windows 16-bit.
2. Un interfaz de usuario orientado a objetos (conocido como Work Place Shell, o WPS), fácil de utilizar y extremadamente potente.
3. Mejor soporte de Internet y redes en el sistema operativo base.
4. Soporte integrado de Java, por lo que puedes ejecutar directamente las aplicaciones Java sin necesidad de utilizar un navegador web. Y el Java de OS/2 es más rápido que el de Windows NT.
5. Mejor soporte de los estándares multiplataforma en vez de utilizar tecnologías propietarias de Microsoft de menor prestaciones
6. REXX, un potente (y a la vez extremadamente fácil de utilizar) lenguaje de programación y sistema de creación de macros totalmente integrado. No encontrarás nada parecido en Windows.

1.1.4.3® IBM OS/2 – DESVENTAJAS

1. Al ser un SO desarrollando para una plataforma IBM, una solución completa con esta arquitectura es extremadamente cara.
2. El soporte técnico es muy exclusivo y caro.
3. Se requiere capacitación compleja para conocerlo a fondo.
4. Es un SO relativamente obsoleto, ya que por su precio solo empresas muy grandes pueden acceder al él.

1.1.5 ® SISTEMA MACINTOSH® OS

Macintosh® es una máquina netamente gráfica. De hecho, no existe una interfaz de línea de comando equivalente para ésta. Su estrecha integración de SO, GUI y área de trabajo la hacen la favorita de la gente que no quiere saber nada de interfaces de línea de comando.

Las capacidades gráficas de la Macintosh® hicieron de esa máquina la primera precursora en los campos gráficos computarizados como la autoedición por computadora.

La familia de microcomputadoras de Apple® Macintosh® y su sistema operativo define otra plataforma importante. Las PC de Macintosh®, que se basan en la familia de microprocesadores de Motorota®, usan la arquitectura de Bus de 32 bits. La plataforma para Macintosh® incluye muchas capacidades sofisticadas que comprende la multitarea, una GUI, la memoria virtual y la capacidad para emular la plataforma MS-DOS. Las PC de Macintosh también tiene la capacidad integrada de compartir archivos y comunicarse con o tras PC de Macintosh en una red.

El sistema operativo constituye la interfaz entre las aplicaciones y el hardware del Macintosh. El administrador de memoria obtiene y libera memoria en forma automática para las aplicaciones y el sistema operativo. Esta memoria se encuentra normalmente en un área llamada cúmulo. El código de procedimientos de una aplicación también ocupa espacio en el cúmulo.

1.1.5.1 ® MACINTOSH® OS – PRINCIPALES COMPONENTES

Ahora se presenta una lista de los principales componentes del sistema operativo.

- El cargador de segmentos carga los programas por ejecutar. Una aplicación se puede cargar completa o bien puede dividirse en segundos individuales que se pueden cargar de manera dinámica conforme se necesiten.
- El administrador de eventos del sistema operativo informa de la ocurrencia de diversos eventos de bajo nivel, como la presión de un botón del mouse o el tecleo. En condiciones normales, el administrador de eventos de la caja de herramientas transfiere estos eventos a las aplicaciones.
- El administrador de archivos se encarga de la entrada / salida de archivos; el administrador de dispositivos se encarga de la entrada / salida de dispositivos.
- Con el manejador de impresoras las aplicaciones pueden imprimir datos en diversas impresoras.
- Los manejadores de dispositivos son programas con los cuales los diversos tipos de dispositivos pueden presentar interfaces uniformes de entrada / salida a las aplicaciones. Tres manejadores de dispositivo están integrados al sistema operativo en ROM: el manejador de disco se encarga del acceso a la información en discos, el manejador de sonido controla los generadores de sonido, y el manejador en serie envía y recibe datos a través de los puertos seriales (estableciendo así la comunicación con dispositivos periféricos en serie como impresoras y módems).
- Con el administrador de AppleTalk las aplicaciones pueden transmitir y recibir información en una red de comunicaciones AppleTalk.
- El Administrador de retrazado vertical programa las actividades por realizar durante las interrupciones de retrazado vertical que ocurren 60 veces cada segundo cuando se refresca la pantalla de vídeo.
- El manejador de errores del sistema toma el control cuando ocurre un error fatal del sistema y exhibe un cuadro de error apropiado.
- Los programas de utilidad general del sistema operativo ofrecen diversas funciones útiles como la obtención de la fecha y la hora, la comparación de cadenas de caracteres y muchas más.

- El paquete de iniciación es llamado por el paquete de archivos estándar para iniciar y nombrar discos; se aplica con más frecuencia cuando el usuario inserta un disco al que no se le han asignado valores iniciales.
- El paquete de aritmética de punto flotante ofrece aritmética de doble precisión. El paquete de funciones trascendentales ofrece un generador de números aleatorios, así como funciones trigonométricas, logarítmicas, exponenciales y financieras. Los compiladores de Macintosh generan en forma automática llamadas a estos paquetes para realizar manipulaciones numéricas.

1.1.5.2 ® MACINTOSH® OS – VENTAJAS

Utiliza el microprocesador Power PC, que por su velocidad y eficiencia tiene un futuro promisorio.

1. Ambiente de usuario amigable. Es el sistema operativo más sencillo de usar.
2. Capacidades poderosas en el manejo y diseño de gráficos.
3. Compatibilidad consigo mismo. Al comprar nuevo hardware o software, es mucho menos propenso a fallas de instalación.
4. Seguridad para servidor de Internet.
5. Características de Plug and Play.
6. Longevidad. En Macintosh los cambios drásticos de arquitecturas que descontinúan aplicaciones, no se dan.
7. Fácil manejo de red.

1.1.5.3 ® MACINTOSH® OS – DESVENTAJAS

1. Participación de mercado muy reducida.
2. No es compatible con PC's y otras arquitecturas.
3. No existen sistemas operativos similares o derivados..
4. Algunas veces hay problemas con la multitarea. Una aplicación puede impedir que otras funcionen.

1.1.6 ® SISTEMA WINDOWS® NT

Con Windows NT, Microsoft ha expresado su dedicación a escribir software no sólo para PC de escritorio sino también para poderosas estaciones de trabajo y servidores de red y bases de datos. Microsoft Windows NT no es necesariamente un sustituto de MS-DOS ni una nueva versión de éste; es, en conjunto, un nuevo SO diseñado desde sus bases para las máquinas más modernas y capaces disponibles.

Windows NT de Microsoft ofrece características interconstruidas que ningún otro SO para PC ofrece, con excepción de Unix.

Además de las características tradicionales de estricta seguridad de sistema, red interconstruida, servicios de comunicación y correo electrónico interconstruidos, herramientas de administración y desarrollo de sistema y una GUI, Windows NT puede correr directamente aplicaciones de Windows de Microsoft y de Unix.

Windows NT, al igual que el OS/2 ver 2.0 y algunas versiones de Unix, es un SO de 32 bits, que puede hacer completo uso de los procesadores de estas características.

Además de ser multitarea, está diseñado para tomar ventaja del multiproceso simétrico.

Desde el principio, Windows NT ha estado disponible de dos formas: como un sistema operativo de estación de trabajo (Windows NT Workstation) y como un sistema operativo de red (Windows NT Server). Microsoft vende las dos plataformas con licencias diferentes y diferente software empaquetado y los pinta como si fueran apropiados para diferentes cargas de trabajo. La cuestión es ¿qué diferencia hay cuando no se tienen en cuenta ni el software ni las licencias? Microsoft ha declarado públicamente que Workstation y Server comparten la misma arquitectura de kernel, y un artículo reciente en el Web de Microsoft va más allá, al describir cómo se ajustan Workstation y Server de una variedad de maneras para que cada uno de ellos cumplan con sus distintos papeles: Workstation es para un uso de sobremesa interactivo y Server es para su uso como servidor de archivos, de impresión y Web. Dada esta información, la suposición razonable es que Microsoft construyó Workstation y Server a partir de diferentes, aunque posiblemente en muchos aspectos compartidas, bases de código. Sin embargo, una comparación binaria de los CDs de instalación revela que el código base es el mismo.

1.1.6.1 ® WINDOWS® NT – BREVE RESEÑA HISTÓRICA

En 1985 IBM y Microsoft realizaron un acuerdo para desarrollar un nuevo sistema operativo para el chip de CPU 286. La versión 1.0 de OS/2 no fue lanzada sino hasta 1988, y por entonces el 386 se estaba haciendo popular. Se hizo claro que el hardware estaba cambiando muy rápido con relación al desarrollo de software. De manera que en 1988 decidieron hacerlo de nuevo, IBM y Microsoft decidieron comenzar a trabajar simultáneamente en dos productos.

OS/2 versión 2 sería un refinamiento evolutivo de sistemas previos, actualizado para las nuevas características de hardware del 386. Continuaría el soporte a las aplicaciones y a los drivers de dispositivos desarrollados para el sistema previo. Esto se convirtió en el IBM OS/2.

OS/2 versión 3 se basaría sobre **N**ueva **T**ecnología. Este sería escrito desde cero y se desarrollaría un sistema basado sobre los mejores principios de ingeniería de software. En un principio sería para CPUs Intel, pero sería portable a otros chips de CPU. Esto se convirtió en Windows NT.

1.1.6.2 ® WINDOWS® NT SERVER – DESCRIPCIÓN GENERAL

Windows NT Server es un sistema operativo para servidores, ampliable e independiente de la plataforma. Puede ejecutarse en sistemas basados en procesadores Intel x86, RISC y DEC Alpha, ofreciendo al usuario mayor libertad a la hora de elegir sus sistemas informáticos. Es ampliable a sistemas de multiproceso simétrico, lo que permite incorporar procesadores adicionales cuando se desee aumentar el rendimiento.

Internamente posee una arquitectura de 32 bits. Su modelo de memoria lineal de 32 bits elimina los segmentos de memoria de 64 KB y la barrera de 640 KB de MS-DOS. Posee múltiples threads (subprocesos) de ejecución, lo que permite utilizar aplicaciones más potentes. La protección de la memoria garantiza la estabilidad mediante la asignación de áreas de memoria

independientes para el sistema operativo y para las aplicaciones, con el fin de impedir la alteración de los datos. La capacidad de multitarea de asignación prioritaria permite al sistema operativo asignar tiempo de proceso a cada aplicación de forma eficaz. Windows NT Server incluye, asimismo, diversas funciones de red.

1.1.6.3 ® WINDOWS® NT WORKSTATION – DESCRIPCIÓN GENERAL

Facilidad de uso: Su aspecto y funcionamiento son similares a Windows 95 (menú inicio, explorador, asistentes, ayuda interactiva y mucho más). Su interfase mejorada aumenta su eficiencia y posibilidades.

Acceso a Internet: Incluye Microsoft Internet Explorer y todos los componentes TCP/IP estándar (Windows Sockets, FTP, Telnet, Ping e IPConfig)

Administración remota: Permite a los administradores desarrollar y aplicar normas y estándares para configuraciones de equipos. Incluye el Supervisor de rendimiento y el Supervisor de eventos para solucionar problemas locales y remotos.

Alto rendimiento y productividad: Se ejecuta en todas las arquitecturas Intel y RISC, lo que ofrece máxima flexibilidad en cuanto a hardware. Aplica automáticamente criterios multitarea a todas las aplicaciones de 16 y 32 bits.

Alta compatibilidad: Es compatible con Client Service for NetWare y otros muchos protocolos de red: TCP/IP (DHCP, WINS, PPP, PPTP y SLIP), RAS, Windows Sockets estándar, NetBEUI y IPX-SPX

Sistema integrado de seguridad: Ofrece un sistema integrado de seguridad que garantiza que las aplicaciones fundamentales de la empresa, los datos críticos y el sistema operativo básico estén protegidos frente a errores o manipulaciones malintencionadas de los usuarios. Asigna zonas de memorias independientes para aplicaciones de 16 bits, protege el código crítico del sistema operativo, los controladores del dispositivo y los datos frente a las aplicaciones.

Mayor protección: Incluye sistemas de reinicio automático como copia de seguridad en cinta y alimentación eléctrica ininterrumpida (UPS) para una mayor protección

1.1.6.4 ® WINDOWS® NT WORKSTATION – VENTAJAS

1. La instalación es muy sencilla y no requiere de mucha experiencia.
2. Es multitarea y multiusuario.
3. Apoya el uso de múltiples procesadores.
4. Soporta diferentes arquitecturas.
5. Permite el uso de servidores no dedicados.
6. Soporta acceso remoto, ofreciendo la detección de intrusos, y mucha seguridad en estas sesiones remotas.
7. Apoyo para archivos de DOS y MAC en el servidor.
8. El sistema está protegido del acceso ilegal a las aplicaciones en las diferentes configuraciones.
9. Permite cambiar periódicamente las contraseñas.
10. Soporta múltiples protocolos.
11. Carga automáticamente manejadores en las estaciones de trabajo.
12. Trabaja con impresoras de estaciones remotas.
13. Soporta múltiples impresoras y asigna prioridades a las colas de impresión.
14. Muestra estadísticas de Errores del sistema, Caché, Información Del disco duro, Información de Manejadores, Nº de archivos abiertos, Porcentaje de uso del CPU, Información general del servidor y de las estaciones de trabajo, etc.
15. Brinda la posibilidad de asignar diferentes permisos a los diferentes tipos de usuarios.
16. Permite realizar diferentes tipos de auditorías, tales como del acceso a archivos, conexión y desconexión, encendido y apagado del sistema, errores del sistema, información de archivos y directorios, etc.
17. No permite criptografía de llave pública ni privada.
18. No permite realizar algunas tareas en sesiones remotas, como instalación y actualización.

1.1.6.5 ® WINDOWS® NT WORKSTATION – DESVENTAJAS

1. Tiene ciertas limitaciones por RAM, como: Nº Máximo de archivos abiertos y almacenamiento de disco total.
2. Requiere como mínimo 16 Mb en RAM y un procesador Pentium de 133 MHz o uno superior.
3. El usuario no puede limitar la cantidad de espacio en el disco duro.
4. No soporta archivos de NFS.
5. No ofrece el bloqueo de intrusos.
6. No soporta la ejecución de algunas aplicaciones para DOS.

1.2 ® SOLUCIÓN DEFINITIVA PROPUESTA

Dada las características de todos y cada uno de los sistemas operativos que fueron evaluados, algunos de ellos física y otros teóricamente, llegamos a la conclusión de utilizar un sistema híbrido, mezclando software de fuente abierta, para los servidores (de dominio y "firewall" de acceso) y propietario para las computadoras cliente que serán utilizadas por los alumnos dentro del Laboratorio.

La evaluación teórica de algunos de los sistemas tuvo que realizarse de esa forma debido a que no todos ellos pueden llegar a funcionar con hardware común, muchos de ellos requieren una arquitectura específica que regularmente se vende en conjunto con el software, además de que conseguir dicho software requiere licencias de utilización.

1.2.1 ® SOLUCIÓN PARA LOS SERVIDORES

Para el caso de los servidores, elegimos utilizar **Linux** en dos versiones diferentes de la recopilación de Red Hat, para el caso del servidor de dominio instalamos la versión 7.3 y para el "firewall" utilizamos la versión 6.2, la razón de elegir versiones diferentes es debido a las características del hardware con el que se cuenta.

En este caso una de las razones más importantes por las cuales se opto por versiones de **Linux**, es que este sistema operativo es una excelente elección para trabajar con **UNIX** a nivel personal. Pero sabemos que **Linux** no es solo para los usuarios personales de **UNIX**. Es robusto y suficientemente completo para manejar grandes tareas, así como necesidades de cómputo distribuidas y servidores de red.

Solo como una referencia, muchas empresas, especialmente las pequeñas se están cambiando a **Linux** en lugar de a otros entornos de estación de trabajo basados en **UNIX**. Grandes vendedores de software comercial ya se dieron cuenta de las oportunidades que puede brindar un sistema operativo gratuito cómo este.

Linux corre completamente en el ***modo protegido del procesador*** y explota todas las características de éste.

Puede acceder directamente a toda su memoria disponible (e incluso más allá de la disponible, usando RAM virtual). **Linux** proporciona una interfase **UNIX** completa, no disponible bajo sistema operativo como **MS-DOS**, el desarrollo y adaptación de aplicaciones **UNIX** bajo **Linux** es cosa fácil, mientras que, bajo **MS-DOS**, estás limitado a un pequeño subgrupo de la funcionalidad de programación **UNIX**.

Como pudimos concluir del análisis realizado en este capítulo, IBM **OS/2** y **Windows NT** son sistemas operativos completamente multitarea, muy parecidos a **Linux**. Técnicamente, **OS/2**, **Windows NT/2000** y **Linux** son bastante similares:

Soportan aproximadamente las mismas características en términos de interfase de usuario, redes, seguridad, y demás. Sin embargo, la diferencia real entre **Linux** y los otros es el hecho de que **Linux** es una versión de **UNIX**, gratis y de código abierto y por ello se beneficia de las contribuciones de la comunidad **UNIX** en pleno.

La importancia de los sistemas operativos basados **UNIX** (**Linux**, Solaris, HP-UX, etc) radica en que no solo es el sistema operativo más popular para máquinas multiusuario, también es la base de la mayoría del mundo del software de libre distribución. Como lo podemos comprobar en Internet casi todo el software de libre distribución disponible está específicamente escrito para sistemas **UNIX**. (Internet en si está profundamente basada en **UNIX**).

Hay muchas implementaciones de **UNIX**, de muchos vendedores, y ni una sola organización es responsable de su distribución. Hay un gran pulso en la comunidad **UNIX** por la estandarización en forma de sistemas abiertos, pero ninguna corporación controla este diseño. Por otro lado, **OS/2** y **Windows NT/2000** son sistemas propietarios, y el pensar en implementarlos implicaría adquirir licenciamiento o negociar un patrocinio.

El factor más importante a considerar por muchos usuarios es el precio. El software de **Linux es gratis**, teniendo acceso a Internet (o a otra red de computadoras) podemos acceder a él. Y en el caso de no contar con acceso a otras redes que lo contengan, tenemos la opción de comprarlo en CD-ROM.

1.2.2 ® SOLUCIÓN PARA LAS COMPUTADORAS CLIENTES

Es importante mencionar que para el caso de las computadoras que utilizarán los alumnos, prácticamente nos vimos obligados a utilizar una solución de MIcrosoft, esto debido a que los programas matemáticos como Matlab, SIMNOM y CC que se utilizan en la Facultad en ejecutarse en alguna versión de Windows o MS-DOS, es bueno aclarar que prácticamente todas las versiones de Windows cuanta con una emulación de MS-DOS (interfaz de comandos).

Decidimos utilizar Windows NT Workstation debido a que es el único SO de Microsoft que maneja seguridad y funciona aceptablemente en computadoras con procesador Pentium I y 16 Mb en RAM, el laboratorio de Control Digital cuanta con varios equipos con estas características. Evidentemente esta versión de Windows es mucho más estable y segura que sus antecesoras 95 y 98.

.Entrando un poco más al detalle con Windows NT Workstation podremos compartir las impresoras, con esto podremos conectar cada impresora a una impresora determinada y utilizarla desde cualquier otra. Esto se realiza mediante el Administrador de impresión o utilizando el comando net share.

También podremos utilizar localmente el administrador de usuarios, la cual es una herramienta de Windows NT Workstation que permite administrar la seguridad de cada computadora, con

esto evitaremos que los alumnos puedan borrar atributos o programas que dañen la integridad de los equipos.

Con esta opción estamos aprovechando la característica que brinda Windows NT Workstation a través del uso del sistema de archivos NTFS para dar permisos de acceso, escritura, eliminación y demás acciones a los usuarios contenidos en diversos grupos sobre programas contenidos en el disco duro local de cada máquina cliente. Así se evita el borrado de archivos clave ya sea por cualquier razón. Solo el administrador del sistema NT puede dar los permisos a los usuarios sobre los programas contenidos en la máquina cliente.Y los usuarios se autentifican en a través del servidor **Linux** (Red Hat 7.3) para poder entrar al dominio de red.

Pero el uso de servicios o protocolos esta regulado cien por cien por el "firewall" establecido en nuestro caso en la puerta de enlace con un **Linux** Red Hat versión 6.2.

1.2.3 ® SOLUCIÓN PARA LA RED FÍSICA

En este caso, se tomó la unánime decisión de realizar una instalación física de red local (LAN) totalmente nueva y rediseñada.

En el siguiente capítulo se detalla minuciosamente la solución técnica para la nueva red local que unirá las computadoras del Laboratorio de Control Digital.

El único material que se reutilizó fueron los concentradores (HUB), ya que estos equipos son los asignados por la Facultad de Ingeniería y reemplazarlos implicaría un desembolso considerable de recursos económicos.

1.2.4 ® SOLUCIÓN DE INTEGRACIÓN ENTRE SISTEMAS OPERATIVOS

Para poder integrar la eficiente convivencia e integración de dos SO de diferente concepción y filosofía (**Linux** y Windows NT Workstation) en una sola red física fue necesario utilizar un API

(Application Program Interface) que es parte de **Linux**. Dicho API* o paquete de herramientas es conocido como SAMBA (Protocolo SMB).

SMB (Session Message Block) es también llamado NetBIOS o LAN Manager con Linux, este protocolo es utilizado por Window 3.11, NT y 95 para compartir discos e impresoras. Usando el paquete de herramientas *SAMBA* creado por Andrew Tridgell, las máquinas UNIX (incluyendo **Linux**) pueden compartir discos e impresoras con servidores o estaciones de trabajo que usan alguna de las versiones anteriores de Windows.

Hay cuatro cosas que uno puede hacer con *SAMBA*:

1. Compartir una unidad de Linux con máquinas Windows.
2. Compartir una unidad de Windows con máquinas Linux.
3. Compartir una impresora de Linux con máquinas Windows.
4. Compartir una impresora de Windows con máquinas Linux.

En nuestro caso utilizaremos SAMBA para integrar funcionalmente la opción 1.

1.2.4.1 ® SAMBA – CONFIGURACIÓN GENERAL

La configuración de *Samba* en un **Linux** (u otra máquina UNIX) es controlada por un solo archivo, /etc/smb.conf. Este archivo determina qué recursos del sistema quieres compartir con el mundo exterior y que restricciones deseas poner en ellos.

Es importante señalar como el ejemplo de configuración de este ejemplo de archivo smb.conf mostrado en esta sección es lo más simple posible, solo para propósitos introductorios, más adelante veremos a detalle el archivo que utilizamos nosotros con la configuración de definitiva.

Cada sección del archivo empieza con una cabecera como [global], [impresoras], etc.

La sección [global] define unas pocas variables que *SAMBA* usará para definir la compartición de todos los recursos.

La sección [homes] permite a los usuarios remotos acceder a sus respectivos directorios principales en la máquina **Linux** local . Esto es, si un usuario de Windows intenta conectar a este recurso desde su máquina Windows, será conectado a su directorio personal. Para hacer esto, tiene que tener una cuenta en la máquina **Linux**. Por esta razón tomamos nombres de usuario genéricos que no tengan que ser cambiados semestre a semestre.

El archivo smb.conf que viene debajo como ejemplo permite a los usuarios remotos acceder a su directorio principal en la máquina local y escribir en un directorio temporal. Para que un usuario de Windows vea estos recursos, la máquina **Linux** debe estar en la red local, lo cual en nuestro caso se cuimple sin problemas. Entonces el usuario simplemente conecta una unidad de red desde el Explorador de Windows o el Windows File Manager.

A continuación se menciona un ejemplo genérico de configuración del archivo /etc/smb.conf.

```
; /etc/smb.conf
;
; Reinicia el servidor cada vez que hagas cambios a este fichero, ej:
; /etc/rc.d/init.d/smb parar
; /etc/rc.d/init.d/smb empezar

[global]
; Hay que quitar el comentario de la siguiente línea si se usan cuentas de invitado
; guest account = nobody
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba
   share modes = yes

[homes]
   comment = Directorios principales
   browseable = no
   read only = no
   create mode = 0750

[tmp]
```

comment = Espacio de archivos temporales
path = /tmp
read only = no
public = yes

1.2.4.2 ® SAMBA – CONFIGURACIÓN LINUX SERVER CON PC'S WINDOWS®

Como se muestra en el ejemplo del archivo smb.conf anterior, compartir una unidad **Linux** con usuarios Windows es relativamente fácil. De todas maneras, como todo lo demás con *SAMBA*, podemos tener las cosas controladas.

Para compartir un directorio con todo el mundo, crea una copia de la sección [tmp] añadiendo algo como esto al smb.conf:

```
[public]
  comment = Todos los alumnos
  path = /home/public
  public = yes
  writable = yes
  printable = yes
```

Para que este directorio lo pueda leer todo el mundo, pero que sólo lo puedan modificar gente del grupo GPO01, modifica la entrada de esta manera:

```
[public]
  comment = Todos los alumnos
  path = /home/public
  public = yes
  writable = yes
  printable = no
  write list = @GPO01
```

1.2.4.3 ® SAMBA – NUESTRA CONFIGURACIÓN

```
# Samba config file created using SWAT
```

```
# Global parameters
[global]
        workgroup = CONTROLDIGITAL
        server string = Servidor bellman
        encrypt passwords = Yes
        passwd program = /usr/bin/passwd %u
        passwd chat = *password* %n\n *password* %n\n *successfull*
        passwd chat debug = Yes
        username map = /etc/samba/smbusers
        unix password sync = Yes
        log file = /var/log/samba/%m.log
        max log size = 0
        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
        logon script = %msamba.bat
        logon path = \\%L\%U\profile
        logon drive = g:
        logon home =\\%L\%U
        domain logons = Yes
        preferred master = Yes
        domain master = Yes
        printing = lprng
[netlogon]
        comment = Login en red
        path = /home/samba/scripts
        guest ok = Yes
        browseable = No
[homes]
        comment = Directorio de usuarios
        writeable = Yes
        preexec = echo "%u se conecta a %L desde %m (%I) a las %T" >> /var/log/bitacoras/.log
        browseable = No
#[matlab]
#       comment = matlab
#       valid users= gpo0101 gpo0102 Administrador
#       path = /homes/matlab
#       writeable = No
#       guest ok = Yes
```

2. CARACTERÍSTICAS TÉCNICAS Y ADECUACIÓN

2.1® DISEÑO Y DISTRIBUCIÓN DE LA RED DE DATOS

La red de comunicación de datos del LCD, cubrirá las siguientes áreas:

Laboratorio de Control digital
Coordinación de Laboratorios
Área de Mantenimiento

Todos los puntos de red estarán interconectados a través de un centro de cableado (centro de distribución) ubicado en el salón de apoyo, bajo la topología (Capa 1 OSI) física de estrella *.

Se utilizó en la parte física (Capa 2 OSI)* una red tipo Ethernet * (10BaseT), esto debido a que los dispositivos encargados de distribuir la señal por todos los nodos trabajan a 10 Mbps, aunque el tipo de cable y conectores utilizados permitirán que en un futuro, si cambian estos dispositivos por otros que soporten Fast Ethernet (100 Base T)* la red estará lista sin necesidad de realizar ninguna modificación.

Un punto importante es que con base a la capacidad instalada, la red podría extenderse en forma organizada, ya que nuestro centro de distribución quedará lo suficientemente holgado para ampliarse, siempre y cuando se sigan los estándares descritos más adelante en este capitulo.
Los nodos instalados en las mesas del LCD son doce. La red podría crecer tomando en cuenta las limitaciones de espacio, de suministro eléctrico y de equipo informático.
.

***** Ver glosario de términos, incluido en este documento

2.1.1 ® SUMINISTRO E INSTALACIÓN DE MATERIALES

En esta sección se presenta una descripción técnica de los materiales y accesorios utilizados en la red de datos para el LCD. El material utilizado en este proyecto fue producto de una donación que realizamos al LCD.

2.1.1.1 ® CABLEADO

Para poder alcanzar el objetivo principal de este proyecto, fue necesario acondicionar una red de área local (LAN), misma que es la columna vertebral, desde el punto de vista de la transmisión de datos entre nodos, para la instalación del sistema de cableado en el LCD se utilizó el cable AVAYA 1071 Series cable Cat 5e*, el cual esta diseñado para transmitir voz y datos con altos estándares de calidad de servicio (Quality of service, QoS) cumpliendo con los requerimientos de la norma ISO/IEC 11801* y los estándares TIA/EIA 568-B los cuales aceptan la aplicación de 10Base-T y 100BaseT.

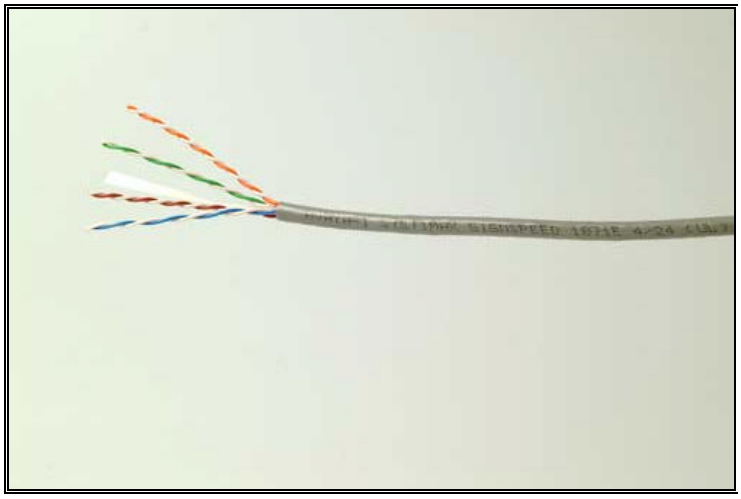


Fig. 2.1 - Detalle de cable

***** Ver glosario de términos, incluido en este documento.

Este cable se conoce como par trenzado por tratarse de pares de cable entrelazados de colores similares, se puede apreciar en la figura anterior que se cuenta con cuatro pares identificados por los colores naranja, naranja-blanco,verde,verde-blanco,azul,azul-blanco , café y café-balnco.

En todos los nodos se consideraron trayectorias rectas (horizontales y verticales) en el tendido del cableado, mismo que antes de ser cortado y rematado fue peinado previamente para así prevenir cualquier tipo de daño e interferencia adicional.

No se utilizó ni piso ni techo falso, esto por el tipo de construcción en el área de la Facultad, donde esta ubicado el LCD sino que en su lugar se utilizaron canaletas plásticas fijas en las paredes.

El cableado se efectuó con conexiones punto a punto, esto es se utilizó un segmento de cable por cada punto a conectar, teniendo como lugar de origen el panel de parcheo, mismo que está instalado en el centro de distribución y como lugar de llegada, las tapas de remate en cada uno de los puntos. Cada segmento de cable se terminó y se remató en sus extremos por un conector RJ45* CAT 5e hembra (Jack*) con recubrimiento de 50 micrones de oro que garantizan transmisiones de 100 Mbps.

Todos los elementos del cable fueron debidamente identificados con etiquetas en ambos extremos, una a nivel del panel de parcheo y otra en el extremo del cable interior de cada nodo (tapa). El frente de cada tapa se identificó con la numeración del segmento correspondiente usando una etiqueta de fácil visualización.

2.1.1.2 ® CENTRO DE DISTRIBUCIÓN DE CABLEADO

Es el espacio físico en donde se encuentran instalados los equipos de comunicación.

Este centro esta constituido por un chasis metálico de 19 x 84 pulgadas, de color negro, en donde se instalaron:

- 2 Concentradores (HUBS) de 19“ de (12 y 16 Puertos)

***** Ver glosario de términos, incluido en este documento.

- 1 Panel de parcheo de 24 Puertos
- 1 organizador de cables frontal.
- 2 charolas
- 4 tapas ciegas

Este centro de distribución se aterrizó según la norma ISO/IEC 11801* además de la **EIA/TIA 607.**



Fig. 2.2 - Centro de distribución de cableado con equipó y conexiones instaladas

---

***** Ver glosario de términos, incluido en este documento.

2.1.1.3® ORGANIZADOR DE CABLES

Los organizadores de cables son accesorios que se utilizan para dar al centro de distribución de cableado una estructura ordenada, así como una excelente presentación.Existen varios fabricantes, tipos y medidas de organizadores de cables, los hay horizontales (frontales y posteriores) y verticales, se fabrican en plástico o metal, su estructura depende del fabricante



Fig. 2.3 - Organizador de cables horizontal

En nuestro caso utilizamos organizadores horizontales frontales metálicos de una y dos unidades de rack, ambos marca Hubell, iguales al mostrado en la Fig.2 , que es un organizador de 2 unidades de rack

2.1.1.4® PANEL DE PARCHEO (PATCH PANEL)

El panel de parcheo es un conjunto de pines y puertos que se pueden montar en un bastidor en el centro de distribución. Los paneles de parcheo actúan como tableros de conmutación que conectan los cables de las estaciones de trabajo unos con otros dispositivos OSI capa 2, como un concentrador y con el exterior con dispositivos OSI de Capa 3 cómo un switch o un ruteador.

El modelo del panel de parcheo que se utilizó fue utilizado es el AVAYA Patch Panel Type 1100. Este es un panel de parcheo modular de 19 pulgadas con 24 Puertos de 8 pines Categoría 5e, acomodados uno tras otro en línea horizontal.



Fig. 2.4 - Vista frontal del panel de parcheo



Fig. 2.5 - Vista trasera del panel de parcheo utilizado



Fig. 2.6 - Ampliación de los remates del panel de parcheo

Se incluyeron etiquetas de identificación frontal en cada puerto del panel, las cuales coinciden con cada nodo terminal.

2.1.1.5 ® CONECTORES

Para hacer posible la comunicación entre el centro de distribución y cada uno de los nodos, se instaló en cada nodo un conector que se utiliza como terminador del cable en la punta, para que desde ahí se conecte cada computadora a la red de datos. Este conector es de tipo hembra de 8 pines RJ-45.

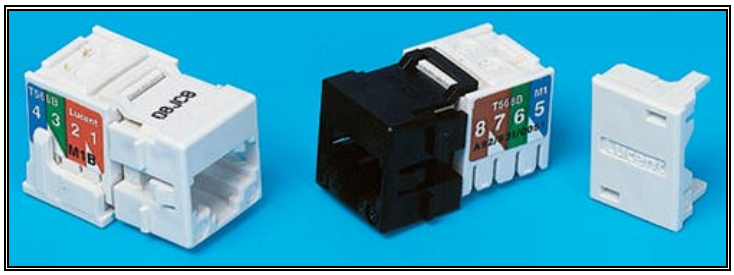


Fig. 2.7 Conector seleccionado para realizar la conexión a las computadoras.

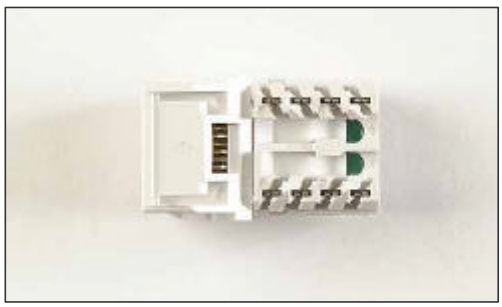


Fig. 2.8 Vista superior del conector RJ-45 utilizado.

2.1.1.6 ® TAPAS DE PARED (WALLPLATES)

Las tapas de pared son coberturas regularmente plásticas que se utilizan para montar los módulos de conexión hembra en un nodo remoto, los que se instalaron en este proyecto son modulares, que aceptan conectores telefónicos y/o de datos. Igualmente tienen la capacidad de aceptar tapas ciegas para este conector con el fin proteger las tomas no activas.

Las tapas de pared utilizadas fueron las siguientes:

En el LCD:	Se instalaron tapas AVAYA modelo M13L las cuales cuentan con 3 módulos de conexión.
En el Área de Mantenimiento	Se instalaron tapas AVAYA modelo M14L que cuenta con 4 módulos de conexión.
En la Coordinación de Laboratorios:	Se instalaron tapas AVAYA modelo M12L que cuenta con 2 módulos de conexión.

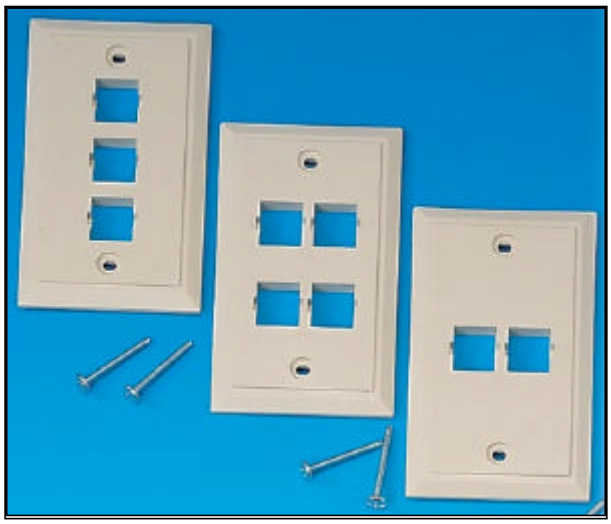


Fig. 2.9 - Tapas de pared

2.1.1.7 ® REMATES

Para unir dos segmentos de cableado se utilizaron los diferentes tipos de conectores descritos anteriormente, ahora, a la acción de enlazar los cables UTP * con algún conector para poder extender o adaptar la red se le denomina remate. Dichos remates se realizaron con una herramienta especial diseñada para este fin dependiendo del tipo de remate que se requiera, lo cual garantiza una conexión adecuada.

Los lugares físicos en donde se realizaron remates fueron:

En nodos.- En todos los nodos , para poder unir los módulos conectores hembra con uno de los extremos de cada cable UTP , y de ahí se conecta a la tarjeta de red de la computadora.

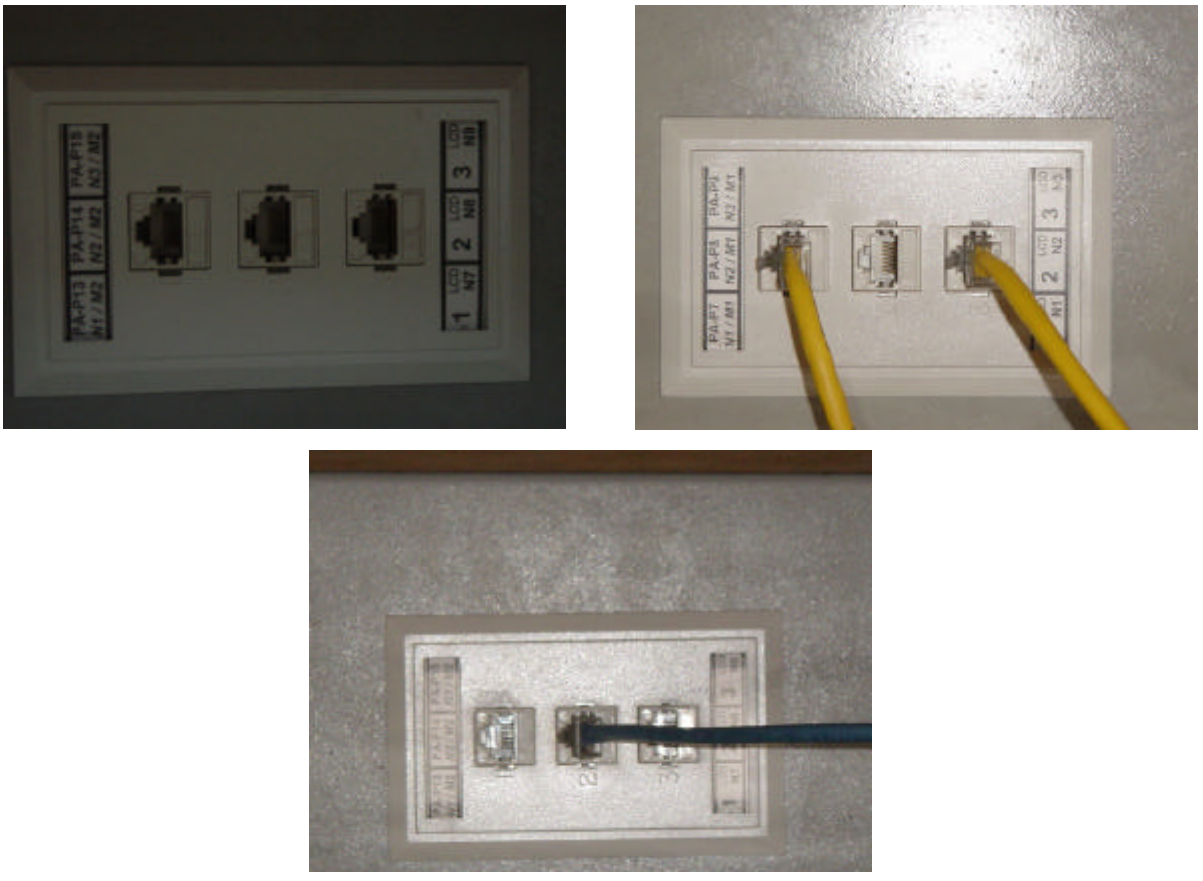


Fig. 2.10 - Ejemplos de remates realizados en los nodos del LCD

***** Ver glosario de términos, incluido en este documento.

Se tomó el mismo criterio de etiquetado en todos los nodos, de tal manera que cualquiera de ellos puede ser identificado sin ningún problema con ayuda de la memoria técnica.

CENTRO DE DISTRIBUCIÓN: En los puertos del panel de parcheo, con el fin de unir la otra punta del cableado por nodo con los dispositivos Capa 2 de red, en nuestro caso HUBS, con esto se completa la topología.

Cabe mencionar que para que pueda haber comunicación entre las computadoras de la red, todos los remates deben realizarse siguiendo el mismo estándar, en nuestro caso, se optó por utilizar el estándar ANSI EIAITIA 568 B

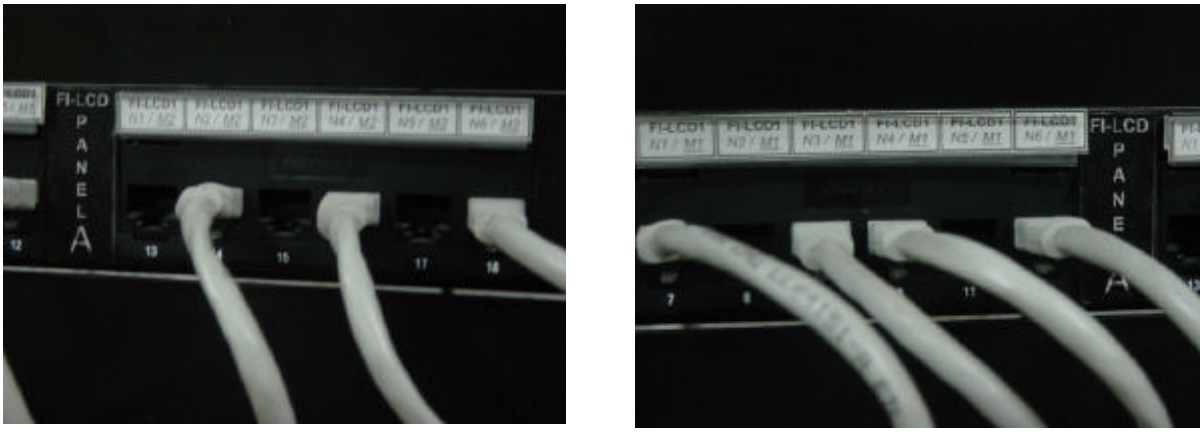


Fig. 2.11 - Conexiones frontales en el centro de distribución

2.1.1.8 ® CABLES DE RED (PATCH CORD)

Un cable de red o patch cord es un tramo corto de cable que se utiliza para unir un dispositivo de red con otro, dicho cable debe ser de alta resistencia ya que está considerado que el usuario pueda conectarlo y desconectarlo las veces que sea necesario.

En nuestro caso, se utilizaron para unir la tarjeta de red de cada computadora con el respectivo conector de pared hembra y para unir cada puerto rematado del panel de parcheo con los dispositivos de capa 2, los concentradores.

***** Ver glosario de términos, incluido en este documento.

En el caso de los nodos remotos, los cables de red o patch cords utilizados fueron construidos manualmente teniendo las siguientes características:

Se utilizó cable para patch cord BELDEN CAT 5e, el cual es más flexible que el utilizado en el cableado horizontal y vertical, se les dió una longitud de 1.5 m., terminado en ambos extremos por conectores RJ45 (plugs) CAT 5e .



Fig. 2.12 - Cables de red.

En el caso de los cables utilizados en el centro de distribución se utilizaron patch cords marca NEWLINK Cabling Systems con las siguientes especificaciones:

Modelo:	15841GY
Descripción:	Patch Cable Cat-5e 24 AWG de 4 pares
Color:	Gris
Longitud:	1 m.
Estándares:	ISO/IEC 11801, TIA/EIA 568
Impedancia:	100 Ohm +/- 15%
Cubierta:	Policarbonato
Pines:	Baño de oro de 50 um sobre filamentos de niquel-plata

Los cuales conectarán el puerto correspondiente del panel de parcheo con el puerto del concentrador respectivo.

2.1.1.9 ® CONCENTRADORES (HUB'S)

El concentrador o hub es el término que se usa para describir un dispositivo que sirve como centro de una red de topología en estrella.

Es un dispositivo de hardware o software que contiene múltiples módulos independientes pero conectados a equipos en red. Los concentradores pueden ser activos cuando repiten señales enviadas a través de ellos o pasivos cuando no repiten, sino que meramente dividen las señales enviadas a través de ellos.

Los concentradores utilizados en esta implementación tienen las siguientes especificaciones:

Marca: SynOptics

Modelo LattisHub 2803
16 Puertos
10 Base T
1 - 0.5 A
60 - 50 Hz
Hecho en Estados Unidos

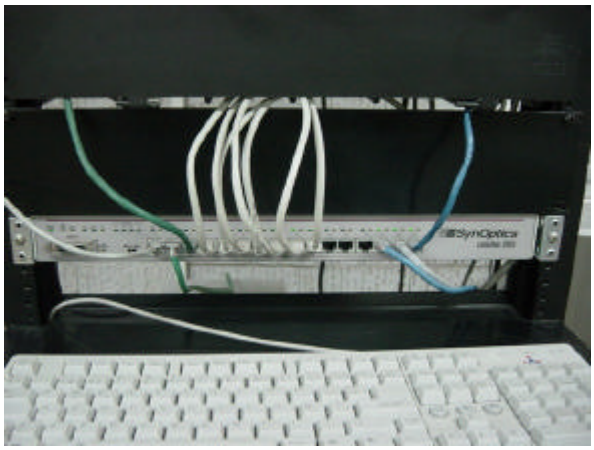


Fig. 2.13 - Concentrador SynOptics

Marca: Hewlett Packard

Modelo Advanced Snack HPJ2600A
12 Puertos
10 Base T
100 – 240 V
0.6 – 0.3 A
60 – 50 Hz
Hecho en Singapur



Fig. 2.14 - Concentrador HP

2.1.1.10 ® HARDWARE

Especificaciones Técnicas de Hardware



(Firewall)

Fig. 2.15 - Firewall IP real

Marca: ACER

RAM: 16 Mb

Disco duro: 1 GB

SO: Linux Red Hat 6.2

Fig. 2.16 - Servidor de dominio



Marca: GENÉRICA

RAM: 64 Mb

Disco duro: 7 GB

SO: Linux Red Hat 7.3

Estaciones de Trabajo

COMPAQ Y ACER

Velocidad de procesador a 120MHz
Memoria RAM 32 MB
PENTIUM I



Fig. 2.17 - Estaciones de trabajo

Las estaciones de trabajo se encuentran en el LCD mientras que el servidor de dominio y el Firewall están en el área de mantenimiento.

2.2 ® DOCUMENTACIÓN DE INSTALACIÓN (CABLEADO)

Con el fin de contar con un expediente organizado que integre la documentación técnica completa y actualizada sobre el trabajo de cableado realizado y las pruebas de funcionamiento de este se realizó la memoria técnica.

Mas adelante se presenta el detalle de cada elemento, trayectoria de cableado, ubicación dentro del edificio, pruebas de transmisión y pruebas de rendimiento realizadas a los servicios instalados.

La intención de elaborar esta memoria técnica es que el LCD cuente con la documentación necesaria para facilitar futuras modificaciones, cambios o adhesiones y para garantizar la correcta transmisión de datos en cada uno de los servicios instalados aún sin tener un equipo en uso en cada salida.Lo anterior permitirá tanto a los usuarios como al administrador de la red conocer en detalle cada elemento, trayectoria y ubicación dentro de la red y así facilitar futuras modificaciones para garantizar la correcta conexión aún sin tener un equipo en uso en cada salida, esta documentación se integra en la memoria técnica que se describe mas adelante.

Para poder describir más claramente lo anterior, dividiremos la documentación del sistema de cableado en dos grandes rubros:

1. Pruebas del cableado y equipos activos
2. Memoria Técnica

2.2.1 ® MEMORIA TÉCNICA

La memoria técnica del cableado tiene como objetivo proveer una identificación consistente de los cables, del centro de distribución, del panel de parcheo y las tapas de manera que se pueda obtener una identificación del sistema de cableado haciendo un uso mínimo de la documentación.

El sistema de identificación involucra cada tapa individual, el panel de parcheo , y el recorrido del cableado.

El Sistema de Identificación del Cableado.

El centro de cableado está constituido por un centro de distribución de piso, donde comienza la distribución del cableado hacia los diferentes puntos de la red, con su respectivo Panel de parcheo, organizador de cables y concentrador.

2.2.1.1 ® IDENTIFICACIÓN Y ETIQUETADO DEL CABLEADO

La nomenclatura a seguir para la identificación de los cables, las tapas y el panel de parcheo es como sigue:

Leyenda de cuatro dígitos: XX-YY donde:

XX: Indica el numero del panel de parcheo donde se conectará el segmento de cable.

YY: Dos dígitos para indicar el numero del puerto que ocupa en el panel de parcheo .

2.2.1.2 ® IDENTIFICACIÓN DE TAPAS DE PARED (WALLPLATES)

Se conoce como tapa (wallplate) a la toma de red en pared, el cual permite la conexión del computador vía Patch cord con los elementos activos del cableado (concentradores, switches,etc) la cual debe identificar una zona única del edificio y punto de red.

Las tapas deben ser etiquetadas haciendo referencia al centro de cableado de origen como sigue:

Identificador de tapa.

Identificación de Patch Panel.

Los puertos del panel de parcheo deben ser identificados haciendo referencia hacia las tapas.

2.2.1.3 ® MEMORIA TÉCNICA DE LA INSTALACIÓN

Panel	Puerto	IP	Usuario
A	1	Libre	Libre
A	2	Libre	Libre
A	3		Millán-Internet DGSCA
A	4		Edgar-Coordinación de LCD
A	5		Rafael-Coordinación de LCD
A	6	Libre	Libre
A	7	192.168.5.101	Máquina 1 (LCD)
A	8	Libre	Libre
A	9	192.168.5.102	Maquina 2 (LCD)
A	10	192.168.5.104	Máquina 4 (LCD)
A	11	Libre	Libre
A	12	192.168.5.103	Maquina 3 (LCD)
A	13	Libre	Libre
A	14	192.168.5.105	Máquina 5 (LCD)
A	15	Libre	Libre
A	16	192.168.5.108	Máquina 6 (LCD)
A	17	Libre	Libre
A	18	192.168.5.110	Máquina 7 (LCD)
A	19	Libre	Libre
A	20	Libre	Libre
A	21	Libre	Libre
A	22		Apoyo LCD
A	23	132.248.59.122 192.168.5.1	Bellman
A	24	192.168.5.2	Bellman

Tabla 2.1 – Memoria Técnica

2.3 ® ESTÁNDARES Y NORMAS UTILIZADAS

En la elaboración del proyecto se procuró cumplir con normas internacionales y así poder garantizar la correcta comunicación física, con la suficiente holgura con el fin de permitir crecimientos ordenados posteriores.

La norma propone un sistema de cableado estructurado (SCS),el cual es un sistema de elementos pasivos, flexibles, independientes de marcas que sirve para interconectar equipos activos. Orientado a solucionar las necesidades tecnològicas mediante la integraciòn de las aplicaciones de voz, datos, video y control.

Los sistemas de cableado estructurado permiten asegurar un ancho de banda capaz de asumir las necesidades de las modernas redes de transmisión de datos. Las soluciones propuestas por los fabricantes se caracterizan por manejar una creciente velocidad.

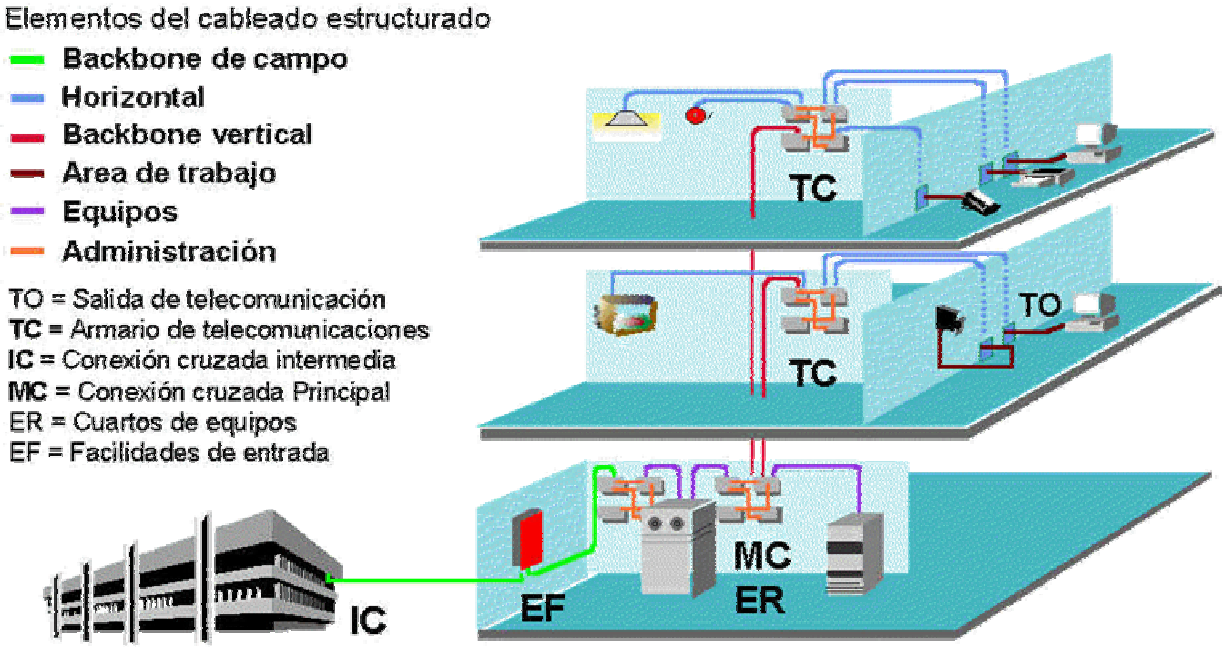


Fig. 2.18 - Elementos de un sistema de cableado estructurado

2.3.1 ® GENERALIDADES SOBRE NORMAS DE CABLEADO ESTRUCTURADO

Se tomó en cuenta para el cableado de cobre de la red la proximidad del cableado y la posibilidad de que pudieran verse afectado por ciertos niveles de interferencias electromagnéticas. Por ejemplo, para este caso las normas 569 ANSI* EIA/TIA* especifican la separación de las trayectorias desde las fuentes típicas de interferencia electromagnéticas.

El estándar 568 de ANSI* EIA/TIA* es uno de varios estándares que divide en "categorías" (comúnmente conocido solo como "CAT") a los sistemas de cableado de par trenzado sin blindaje (UTP) así como sus aditamentos (alambres, ensambladuras y conectores) en términos de la capacidad de transmisión de datos que pueden sostener.

Las especificaciones describen el material del cable así como los tipos de conectores y de bloques de conexión que se utilizarán para conformarse con una categoría. Estas categorías son:

Tabla 2.2 – Categorías de cableado de par trenzado

CATEGORÍA	TRANSMISIÓN MÁXIMA DE DATOS	USADO GENERALMENTE EN:
CAT 1	Menos de 1 Mbps	Voz análoga. Similar al cableado para timbre de casa.
CAT 2	4 Mbps	Utilizado principalmente en el sistema que cablegrafía de la IBM para las redes token ring.
CAT 3	16 Mbps	Voz y datos sobre Ethernet 10BaseT
CAT 4	20 Mbps	Utilizado en el token ring de 16 Mbps. NO es comúnmente utilizado.
CAT 5	100 Mbps (4 pares)	100 Mbps TPDDI (100baseT o Fast - Ethernet)
CAT 5e	100 Mbps	100 Mbps TPDDI (100baseT o Fast Ethernet, Gigabita Ethernet
CAT 6	200-250 Mbps	Usos De banda ancha estupendamente rápidos.

Para CAT 5e, la distancia horizontal máxima deberá ser de 100 Mts (325 Pies) independientemente del tipo de medio (cobre) es decir lo largo del cable desde la terminación mecánica del medio (tapa de remate en nodo) y el centro de distribución horizontal. En nuestro caso no se presentará ningún problema ya que en todos los nodos de acceso se cumple con lo especificado por la norma anterior.

Se permite usar 3 Mts (10 pies) adicionales (Patch cords) desde la roseta del lugar de trabajo (nodo) a la estación de trabajo.

Para conexiones de tipo Ethernet a mayores distancias se utiliza fibra óptica, la meta es que con cableados de tipo par trenzado (UTP*) CAT 5, el cual se encuentra instalado en la mayoría de las organizaciones para unir sus PC's de escritorio pueda ser utilizado para resolver la limitante de la distancia entre nodos y dispositivos generadores de señal.

Las dos especificaciones más populares en cableado comercial son CAT 3 y CAT 5. Estos dos cables pueden parecer idénticos, el CAT 3 se prueba a un sistema más bajo de especificaciones y puede causar errores de la transmisión si está es llevada a velocidades de transmisión más rápidas (por arriba de los 16 Mbps). El CAT 3 que uno dos nodos es certificado de extremo a extremo solamente para una señal máxima de 16 Mhz, mientras que el cable del CAT 5 debe pasar una prueba de 100 Mhz.

El conector modular usado en CAT 5 es de tipo RJ-45. Abajo se muestran los diagramas con un apropiado arreglo en el remate de los pines. NOTA: El conectador RJ-11 es el conector estándar norteamericano que se usa en las conexiones telefónicas comunes, es similar al RJ-45 pero un poco más pequeño.

Se consideraron los siguientes estándares internacionales:
?

- ANSI* EIA*/TIA* 568
- ANSI EIA/TIA 569
- ANSI TIA/EIA 606
- ANSI EIA/TIA 607
- ISO*/IEC* 11801

Cada uno de estas normas funciona en conjunto. Cuando se diseña e instala cualquier sistema de telecomunicaciones, se deben revisar las normas adicionales como el código eléctrico nacional (NEC) de los E.U.A., o las leyes y previsiones locales como las especificaciones NOM (Norma Oficial Mexicana)

***** Ver glosario de términos, incluido en este documento.

A continuación se presenta una breve descripción del detalle de las normas

2.3.2 ® NORMA ANSI EIA/TIA 568 Y SU APLICACIÓN AL PROYECTO

La norma central que especifica un género de sistema de cableado para telecomunicaciones que soporte un ambiente multi producto y multi proveedor, es la norma ANSI TIA/EIA 568, "Norma para construcción comercial de cableado de telecomunicaciones". Esta norma fue desarrollada y aprobada por comités del Instituto Nacional Americano de Normas (ANSI), la Asociación de la Industria de Telecomunicaciones (TIA), y la Asociación de la Industria Electrónica, (EIA), todos de los E.U.A. Estos comités están compuestos por representantes de varios fabricantes, distribuidores, y consumidores de la industria de redes. La norma establece criterios técnicos y de rendimiento para diversos componentes y configuraciones de sistemas.

Esta norma establece entre otras cosas, dos estándares (A y B) para el cableado Ethernet 10Base-T determinando que color corresponde a cada pin del conector RJ-45. El estándar B también se le llama especificación AT&T y es usado comercialmente mas frecuentemente que el A. Pero muchas instalaciones están diseñadas para el del tipo A, también denominado ISDN.

Los cables conductores de datos (CAT 3 o CAT 5) están constituidos de 4 pares de alambres, cada par tiene un alambre de color sólido y un blanco rayado con ese mismo color. Cada uno de los pares se tuercen juntos. Para mantener confiabilidad en Ethernet, dichos cables no deben ser desenroscados más de lo necesario para el remate en cualquier tipo de conector (cerca de 1 cm.).

Los pares de alambre seleccionados para transmitir datos Ethernet 10BaseT son anaranjados y verdes, los otros dos pares, café y azul, no se usan. Dichas selección se aplica en conectores RJ45 (macho).

Esta norma es muy importante ya que es utilizada para los siguientes fines:

1. Conectar una estación de trabajo a la roseta de una instalación de cableado estructurado.

2. Conectar los puertos de un panel de parcheo con un equipo de comunicación LAN (concentrador o switch).
3. Conectar directamente una estación de trabajo a equipo de comunicación LAN (concentrador o un switch).
4. Interconectar (cascadear) dos o más equipos de comunicación LAN (concentradores o switches).

Así mismo, incluye los siguientes 6 subsistemas:

- **Subsistema de área de trabajo:** Se extiende desde la placa de pared (wallplates) hacia el equipo de trabajo del usuario (nodo).
- **Subsistema de cableado horizontal:** Se define desde el àrea de trabajo hasta el closet de telecomunicaciones. Incluye: cable, accesorios, cross connects. Debe ser topologìa estrella.
- **Subsistema de cableado vertical:** Es la interconexiòn entre los closets de telecomunicaciones, cuarto de equipos y entrada de servicios. Tambièn incluye el cableado entre edificios, en nuestro caso no fue necesario ya que el alcance la este proyecto no incluye la interconexción de cableado entre edificios.
- **Subsistema Closet de Telecomunicaciones:** Area exclusiva dentro de un edificio para el equipo de telecomunicaciones, su funciòn principal es la terminaciòn de los cableados horizontales y verticales.
- **Subsistema de Cuarto de Equipos (MDF*):** Area exclusiva dentro de un edificio en donde se centralizan todas las consolas de control de voz, datos, video y control.
- **Subsistema de entrada o acometida de servicios**: es el punto donde la instalación exterior y dispositivos asociados entran al edificio. Este punto puede estar utilizado por servicios de redes públicas, redes privadas del cliente, o ambas. Este es el punto de demarcación entre el portador y el cliente, y en donde están ubicados los dispositivos de protección para sobrecargas de voltaje.

El tipo de estandar para esta norma que seleccionamos para realizar los remates de cableado en nuestro proyecto fue la configuración B.

***** Ver glosario de términos, incluido en este documento.

Configuración de la norma 568:

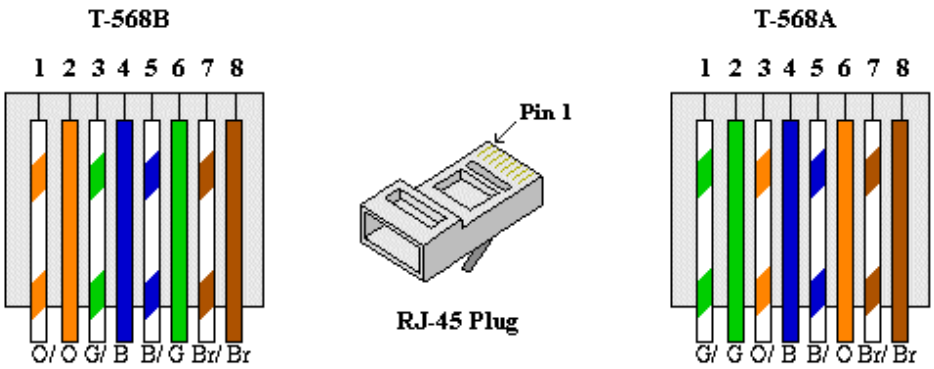


Fig. 2.19 – Código de colores de la Norma 568B

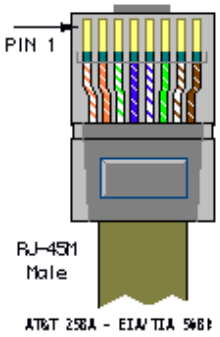
Definición importante:

Cable cruzado (crossover) : Así se le conoce a un cable que utiliza el estándar 586-A en un extremo y uno 586-B en el otro y son utilizados para conectar entre sí concentradores y switche's o para conectar dos estaciones de trabajo directamente.

En general, los pines que se utilizan en las conexiones Ethernet son de tipo Straight-Through, lo cual significa que el pin 1 del conector en un extremo está conectado con el pin 1 del conector en el otro extremo (para cualesquiera de los dos estándares). La única ocasión que los pines deben ser cruzados en conexiones 10/100 Base-T es cuando se enlazan dos dispositivos de Ethernet en forma directa (Concentradores, ruteadores, switches,etc). Entonces necesitamos un cable "cruzado", el cual intercambia de un lado los pines de transmición y recepción. Una manera fácil de recordar el cómo hacer un cable cruzado es rematar los alambres un uno de los extremo con el estándar T-568A y el otro con el estándar de T-568B.

A continuación se muestra en forma gráfica lo comentado anteriormente sobre la configuración de un cable cruzado.

Primera punta de Cable Cruzado	
PIN RJ-45	*PIN RJ-45*
1 Rx+	3 Tx+
2 Rc-	6 Tx-
3 Tx+	1 Rc+
6 Tx-	2 Rc-



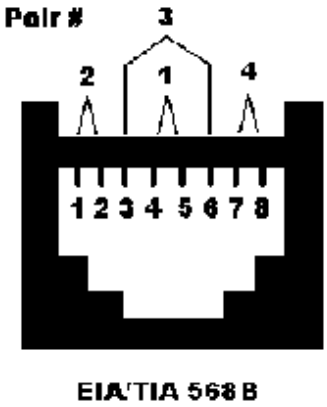
Segunda punta de *Cable Cruzado*	
PIN RJ-45	**PIN Rj-45**
1 Tx+	1 Rc+
2 Tx-	2 Rc-
3 Rc+	3 Tx+
6 Rc-	6 Tx-

Fig. 2.20 – Configuración de un cable cruzado

El conector de pared debe venir con un diagrama o señalar por lo menos los números de pin debajo de los cuales se puede seguir la misma secuencia de color en los remates, la cual debe ser igual a la que tiene el conector RJ-45.

A CONTINUACIÓN SE MUESTRA EL DETALLE DE LA CONFIGURACIÓN ANSI EIA/TIA 568B, ASÍ COMO LA DESCRIPCIÓN DE USO DE CADA PIN.

Fig. 2.21 – Código de colores en un conector de pared (jack) en base a Norma 568B



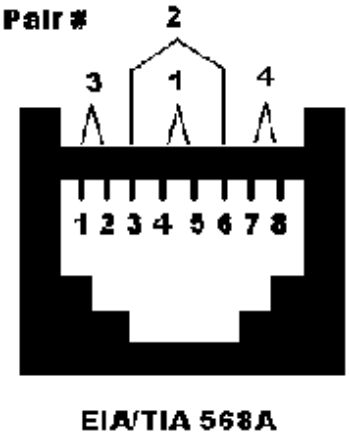


Pin	COLOR	PAR	DESCRIPCIÓN
1	Blanco/Naranja	2	TX - Data +
2	Anaranjado	2	TX - Data -
3	Blanco/Green	3	RX - Data +
4	Azul	1	No se usa
5	Blanco/Azul	1	No se usa
6	Verde	3	RX - Data -
7	Blanco / Café	4	No se usa
8	Café	4	No se usa

Tabla 2.3 – Código de colores en un conector de pared (jack) en base a Norma 568B

A CONTINUACIÓN SE MUESTRA (SÓLO COMO REFERENCIA) EL DETALLE DE LA CONFIGURACIÓN ANSI EIA/TIA 568A, ASÍ COMO LA DESCRIPCIÓN DE USO DE CADA PIN.

Fig. 2.22 – Código de colores en un conector de pared (jack) en base a Norma 568B





Pin	COLOR	PAR	DESCRIPCIÓN
1	Blanco / Verde	3	RX - Data +
2	Verde	3	RX - Data -
3	White / Naranja	2	TX - Data +
4	Azul	1	No se usa
5	Blanco / Azul	1	No se usa
6	Naranja	2	TX - Data -
7	Blanco / Café	4	No se usa
8	Café	4	No se usa

Tabla 2.4 – Código de colores en un conector de pared (jack) en base a Norma 568B

Notas importantes:

- Los números de PIN impares deberán tener rematado un alambre rayado (blanco/color sólido).
- Los números de PIN pares deberán tener rematado un alambre de color sólido.
- En el cable CAT5 se deben utilizar conectores de tipo RJ-45.

Pin#	Par#	Función	Color del Cable	10/100 Base-T Ethernet	100 Base-T4 y 1000 Base-T Ethernet
1	2	Transmite	Blanco/Naranja	Si	Si
2	2	Recibe	Naranja/Blanco	Si	Si
3	3	Transmite	Blanco/Verde	Si	Si
4	1	Telefonía	Azul/Blanco	No	Si
5	1	Telefonía	Blanco/Azul	No	Si
6	3	Recibe	Verde/Blanco	Si	Si
7	4	Respaldo	Blanco/Marrón	No	Si
8	4	Respaldo	Marrón/Blanco	No	Si

Tabla 2.5 - Principales características pin a pin del estándar 568B

Después de lo anteriormente expuesto es muy importante concluir que no importando de la configuración de remate elegida (A ó B), se deben tener todas las conetorizaciones bajo el mismo estándar, no es posible mezclarlas si es que no se desea tener problemas, es decir, si elegimos la configuración A para las rosetas de los nodos remotos (jacks) es imprescindible que las los remates en el panel de parcheo (centro de distribución del cableado) también estén bajo este mismo estándar.

El motivo por el cual seleccionamos la configuración B de esta norma, es porque es la más comúnmente utilizada por la industria para instalaciones comerciales, la configuración A es utilizada generalmente por grandes compañía de telecomunicaciones (carriers) para conectorizar circuitos telefónicos.

***** Ver glosario de términos, incluido en este documento.

3. SISTEMA DE ACCESO POR INTERNET

3.1 ® PLANTEAMIENTO

Necesitamos para el sistema de acceso a los archivos un manejador de base de datos o aprovechar la interfase de **SAMBA** para redes mixtas NT-LINUX. Finalmente decimos utilizar **SAMBA** para direccionar los archivos de las máquinas cliente a la partición */home* del servidor de dominio. Al tener esta interfase establecida, a través de los programas SHELL SCRIPT se pueden realizar operaciones con los archivos de los usuarios, así como también modificar los archivos */etc/passwd, etc/group y etc/samba/smbpasswd* que permiten agregar, eliminar o cambiar usuarios. Pero para desplegar el sistema en Internet se necesita un servidor Web.

Se instaló Tomcat en el servidor de dominio. Este implementa las tecnologías de *SERVLETS* JAVA y JAVA Server Page. Está disponible bajo la licencia ASF desde el sitio http://www.apache.org . La finalidad fue usar las ventajas de la tecnología JSP que permite el rápido desarrollo de aplicaciones basadas en Web que son independientes de la plataforma. La tecnología JSP separa la interfase de usuario del contenido de la programación posibilitando cambiar la plantilla del diseño sin cambiar la dinámica subyacente.

Ahora los JSP usan etiquetas del tipo HTML que encapsulan la lógica que genera el contenido para la página. La lógica de la aplicación se apoya en recursos del servidor que la página accesa con estas etiquetas. Puede usar tanto etiquetas del tipo HTML o XML las cuales son pasadas de regreso a la página de respuesta. Al separar la lógica de la página de su diseño y a la vez permitiendo la reutilización de componentes hace rápido y fácil construir aplicaciones que se ejecutan en Internet. La tecnología JSP es una extensión de la tecnología Java Servlet.

Los SERVLETS son módulos que corren del lado del servidor, independientes de la plataforma. También son componentes de aplicación que se descargan según sean demandados por la parte del sistema que los necesita.

Los objetos implícitos request y response del API de los servlets funcionan el primero para obtener los parámetros del cliente y el segundo representa la respuesta del servidor a alguna petición.

La conexión entre los SHELL SCRIPT y los JSP se logró al programar clases dentro de Tomcat en la dirección absoluta:

/var/www/jakarta-tomcat-3.2/webapps/cdigital/WEB-INF/classes/samba/

Que usan la salida de los SHELL SCRIPT para indicar por medio de una bandera si la acción realizada fue exitosa. La clase SAMBA se muestra mas abajo así como los SHELL SCRIPT y una breve explicación de su contenido. Esta clase fue realizada en código java por las ventajas en la conectividad a Internet como lo manifiestan sus paquetes java.io, java.net, y java.applet principalmente. El manejo de interrupciones en java try y catch permiten al sistema no bloquearse ni detenerse por causa de algún error tanto del usuario como el de algún caso no previsto por el desarrollador.

La instalación de este programa Tomcat en la partición /var del ambiente Linux del servidor de dominio proporciona las ventajas de una conectividad a Internet pudiéndose utilizar la URL* de http://bellman.fi-b.unam.mx con dirección IP real numérica 132.248.59.122 y tener libre acceso a dicha interfase desde cualquier parte de la Web. Solo en el caso de máquinas dentro de la red interna protegida por el **Firewall** se usa la dirección IP virtual 192.168.5.2.

3.2 ® DESARROLLO

Los programas SHELL SCRIPT se hallan en la dirección absoluta, del servidor de dominio Linux 7.3.

/var/www/Yakarta-tomcat-3.2/smbscripts

Los programas usan los archivos etc/passwd, etc/groups y etc/samba/smbpasswd principalmente.

El uso de los comandos cat, grep y awk para SHELL SCRIPT aplicados a dichos archivos permiten conjunto con las estructuras de control como el for, if y do-while poder obtener las salidas esperadas a dichos programas como un exit 0,exit 1 o un exit $?, para usar estas salidas como banderas para programar una clase en código java la cual valida dichas banderas determinando si dichas funciones realizadas por los SCRIPTS fueron exitosas y pudiéndose registrar en un entorno de Internet afectándose los archivos mencionados anteriormente. La interfase gráfica realizada con formularios, enlaces, tablas e imágenes ligadas a acciones realizadas por los programas JAVA Server Page forman el sistema de acceso al LCD.

Se programaron tres roles principales de usuario como son la de administrador del sistema con todas las herramientas disponibles para modificar grupos, usuarios, profesores y asignar contraseñas a dichos elementos, la de alumno donde este puede importar y modificar archivos y la de profesor donde este puede dejar avisos a sus alumnos y mandarles archivos a las cuentas de sus alumnos.

Cada rol de usuario es distinto por sus alcances y estos son determinados por los programas ubicados en el caso del administrador en:

/var/www/jakarta-tomcat-3-2/webapps/cdigital/administrador
o
/var/www/jakarta-tomcat-3.2/webapps/cdigital/alumno

Para el alumno, y para el profesor en:

/var/www/Yakarta-tomcat-3.2/webapps/cdigital/profesor

Cabe mencionar que dichos archivos y otros mas como las imágenes tipo JPG, un archivo especial de flash , la carpeta de WEB-INF donde se programó la clase especial de conexión con los SHELL SCRIPTS y otras utilidades se concentraron en un WAR, el cual hace más portátil el sistema permitiendo trasladarlo fácilmente. La clase a que se hace referencia con SAMBA se encuentra en la dirección absoluta:

/var/www/Yakarta-tomcat-3.2/webapps/cdigital/WEB-INF/classes/samba/Samba.java.

3.3 ® PROGRAMACION

A continuación están los programas SHELL SCRIPT y los programas en JAVA así como una descripción breve de la función de cada uno dentro del proyecto, por razones de ordenamiento hemos divido los programas SHELL SCRIPT en cuatros tipos, los que involucran a los alumnos, los que tienen que ver con profesores, los correspondientes a los grupos de alumnos y la cuarta clasificación es la correspondiente a la administrador del sistema.

3.3.1 ® PROGRAMAS SHELL SCRIPT

En pocas palabras un SHELL SCRIPT es un programa que se puede ejecutar desde la línea de comandos de un sistema Linux. La SHELL de Linux tiene su propio lenguaje, en el cual son escritos los SHELL SCRIPTS. El lenguaje SHELL SCRIPT es muy versátil aunque hay que advertir que es bastante ineficiente. Son mucho más eficientes los programas escritos en lenguaje C. Este es el lenguaje en el que se escribió el KERNEL de Linux y otros Sistemas Operativos. El valor de este lenguaje es que permite hacer cosas complicadas con muy poco esfuerzo en perfecta combinación con una serie de comandos también muy potentes disponibles en Linux. Este es el motivo principal por el cual nos decidimos a utilizar este tipo de programas como la columna vertebral del sistema de administración del LCD.

3.3.1.1 ® PROGRAMAS – GENERALIDADES DEL ADMINISTRADOR

Programa 3.1 (Agregando un usuario)

```
#adduser.sh
#!/bin/bash
#
#Script para agregar un usuario
USER=$1
PASSWD=$2
GROUP=$3
/usr/sbin/adduser -g $GROUP -s /dev/null $USER
/usr/sbin/edquota -p smbpass $USER
/usr/bin/smbpasswd -a -s $USER $PASSWD
exit 0
```

El programa 3.1 recoge tres parámetros desde la línea de comandos de ejecución los cuales son por orden usuario, contraseña y grupo. Se aplica el comando adduser agregando al grupo establecido por el parámetro grupo con shell /dev/null y con el nombre del usuario.

La segunda instrucción establece las cuotas de almacenamiento para el nuevo usuario

La tercera instrucción agrega al usuario y su contraseña en modo silencioso al archivo /etc/samba/smbpasswd y regresa un cero al concluir exitosamente las operaciones.

Programa 3.2 (Borrando un usuario)

```
#deluser.sh
#!/bin/bash
#
#Script para borrar un usuario
USER=$1
/usr/bin/smbpasswd -x $USER
/usr/sbin/userdel -r $USER
exit $?
```

El programa 3.2 borra al usuario del archivo etc/passwd mediante el parámetro –r se borra todo el directorio del usuario y su mail spool*.

Programa 3.3 (Comprobando que el usuario es de tipo Administrador)

```
#checkadmin.sh
#!/bin/bash
#
#Script para comprobar si un usuario es de tipo administrador
USER=$1
GROUP=$(grep ^$USER: /etc/passwd | awk -F":" '{print $4}')
if [ $GROUP = "503" ]
then
  exit 0
fi
  exit 1
```

El programa 3.3 revisa en el archivo etc/passwd, si en la columna cuatro hay un valor de 503 el cual se refiere al GID (identificador de grupo) de administrador, si es así manda un cero como salida el cual será usado como fue mencionado anteriormente por la clase Samba.java

Programa 3.4 (Comprobando que el usuario existe en la interfaz SAMBA)

```
#checkpasswd.sh
#!/bin/bash
#
#Script para comprobar si un usuario existe en samba
USER=$1
PASS=$2
/usr/bin/smbpasswd -s smbpass $PASS 1>/dev/null
PASS1=$(cat /etc/samba/smbpasswd | grep smbpass | awk -F":" '{print $3$4}')
PASS2=$(cat /etc/samba/smbpasswd | grep $USER | awk -F":" '{print $3$4}')
if [ $PASS1 = $PASS2 ];
then
exit 0
fi
exit 1
```

El programa 3.4 utiliza dos argumentos, el nombre de usuario y su contraseña. La primera instrucción esta en modo silencioso obligando a ingresar tanto el usuario como la contraseña desde el teclado.

La variable PASS1 se obtiene a partir del archivo smbpasswd haciéndose una búsqueda determinada por smbpass y de allí se imprimen las columnas tres y cuatro. La variable PASS2 se obtiene del mismo archivo y se hace una búsqueda usándose el usuario introducido a través de la interfaz de comandos y de allí se imprimen las columnas tres y cuatro. Se usa una sentencia de control if para comparar las variables y de ser la condición evaluada verdadera se regresa un cero.

Programa 3.5 (Comprobando si existe un usuario)

```
#checkuser.sh
#!/bin/bash
#
#Script para comprobar si existe un usuario
USER=$1
grep ^$USER: /etc/passwd 1>/dev/null
exit $?
```

El programa 3.5 emplea un solo argumento el cual sirve para hacer una búsqueda a través del comando grep con dicho argumento sobre el archivo etc/passwd.

Programa 3.6 (Agregando una PC al dominio)

```
#addmachine.sh
#!/bin/bash
#Script para agregar una máquina
USER=$1
GROUP=$2
/usr/sbin/adduser  -s /bin/bash $USER$
/usr/bin/smbpasswd -a -m $USER
exit 0
```

El programa 3.6 pasa un parámetro el cual es el nombre de la máquina. El parámetro de grupo nos es indiferente hasta ahora.

Mediante el comando adduser agregamos la maquina al archivo etc/passwd con el formato de NOMBREMAQUINA$ y el segundo comando agrega este al /etc/samba/smbpasswd con el parámetro –a de agregar y –m de maquina.

Programa 3.7 (Cambiando la contraseña de acceso de un usuario en SAMBA)

```
#!/bin/bash
#
#Script para cambiar el password de samba de un usuario
LOGIN=$1
PASSWD=$2
/usr/bin/smbpasswd -s $LOGIN $PASSWD 1>/dev/null
exit $?
```

El programa 3.7 toma dos argumentos pasados a través de la línea de comandos los cuales son el nombre del usuario y su contraseña. La instrucción smbpasswd con la opcion –s escucha en forma silenciosa dichos argumentos.

3.3.1.2 ® PROGRAMAS – GRUPOS

Programa 3.8 (Agregando un grupo de alumnos)

```
#addgroup.sh
#!/bin/bash
#
#Script para agregar un grupo
GROUP=$1
/usr/sbin/groupadd CTRL$GROUP 1>/dev/null
exit $?
```

El programa 3.8 lo usamos para agregar un grupo con el prefijo CTRL. Agregándolo al archivo etc/passwd.

Programa 3.9 (Borrando un grupo de alumnos)

```
#delgroup.sh
#!/bin/bash
#
#Script para borrar un grupo
GROUP=$1
/usr/sbin/groupdel $GROUP
exit $?
```

El programa 3.9 aplica la instrucción groupdel con el argumento GROUP y manda la salida mediante el comando exit$?.

Programa 3.10 (Obteniendo los alumnos pertenecientes a cierto grupo)

```
#getalumns.sh
#!/bin/bash
#
#Script para obtener los usuarios pertenecientes a un grupo
GROUP=$1
rm /tmp/alumns$GROUP.txt
cat /etc/passwd | awk -F":" '{print $1":"$4}' | grep `cat /etc/group | grep $GROUP | awk -F":" '{print $3}'`$ | awk -F":" '{print $1}' >>
/tmp/alumns$GROUP.txt
```

El programa 3.10 recibe un solo parámetro. Lo primero que hace es borrar el archivo /tmp/alumns$GROUP.txt, inmediatamente después del archivo /etc/passwd con separador de campos: imprime las columnas uno y cuatro , siendo el nombre de usuario y el identificador de grupo en forma numérica. Se busca en el archivo /etc/group con la variable pasada a través del argumento con la cual se imprime la columna tres que es el gid y después se imprime la columna uno mandándose todo al archivo limpio de /tmp/alumns$GROUP.txt.

Programa 3.11 (Obteniendo el grupo al que pertenece un alumno o usuario)

```
#getalumngrouo.sh
#!/bin/bash
#
#Script para obtener el grupo al que pertenece un usuario
USER=$1
rm /tmp/grp$USER.txt 2>/dev/null
cat /etc/group | awk -F":" '{print $1":"$3}' | grep `grep ^$USER: /etc/passwd | awk -F":" '{print $4}'`$ | awk -F":" '{print $1}' >> /tmp/grp$USER.txt
exit 0
```

El programa 3.11 toma el nombre de un usuario en este caso un alumno. Lo primero que hace este programa es borrar el archivo /tmp/grp$USER.txt donde USER es la variable pasada a través de la línea de comandos. Después del archivo etc/group se imprimen las columnas uno y tres.

Después de esto se hace una búsqueda en el archivo etc/passwd de aquellas líneas o línea que empiece con el nombre de dicho usuario. De este se imprime la columna cuatro y después la columna uno mandándose esta salida al final del archivo que se borro anteriormente en la primera instrucción de este programa.

Programa 3.12 (Obteniendo los grupos asignados a un profesor)

```
#groupsprof.sh
#!/bin/bash
#
#Script para obtener los grupos asignados a un profesor
USER=$1
rm /tmp/group$USER.txt
for GROUP in $(grep ^$USER: /etc/passwd | awk -F":" '{print $5}')
do
  echo $GROUP >> /tmp/group$USER.txt
done
exit 0
```

El programa 3.12 usa un argumento pasado a través de la línea de comandos. Escribe la variable group al final del archivo /tmp/group concatenado con el nombre del usuario que en este caso debe ser un profesor y lo establece como un archivo de texto.

Programa 3.13 (Obteniendo los grupos existentes)

```
#getgroups.sh
#!/bin/bash
#
#Script para obtener los grupos existentes
cat /etc/group | awk -F":" '{print $1":"$3}' | grep ^CTRL > /tmp/sambagroups.txt
```

El programa 3.13 toma las columnas uno y tres del archivo etc/group y luego hace una búsqueda de aquellas líneas que empiecen con CTRL y las direcciona al archivo /tmp/sambagroups.txt

3.3.1.3 ® PROGRAMAS – PROFESORES

Programa 3.14 (Asignando un grupo a un profesor)

```
#asignprof.sh
#!/bin/bash
#
#Script para asignar un grupo a un profesor
PROF=$1
GROUP=$2
GRUPOS=`cat /etc/passwd | awk -F":" '{print $1":"$5}' | grep ^$PROF: | awk -F":" '{print $2}'`
GRUPOS=$GRUPOS" "$GROUP
/usr/bin/chfn -f "$GRUPOS" $PROF
exit $?
```

El programa 3.14 recoge dos parámetros. La variable GRUPOS toma su valor a partir de una serie de operaciones sobre el archivo etc/passwd donde se imprimen las columnas uno y cinco que son respectivamente el nombre del usuario y los grupos de los que es profesor el usuario. Solamente los profesores tienen la columna cinco con valores, de esta salida se busca solamente los nombres de usuario con comienzo PROF y de este arreglo se imprime la columna dos que es el numero de grupos asignados a ese profesor.

Después se concatena con un espacio vacío y se repite dicha variable. La siguiente instrucción cambia la información finger contenida en el archivo /etc/passwd del nombre real del usuario por otro.

Programa 3.15 (Comprobando que el usuario es un profesor)

```
#checkprf.sh
#!/bin/bash
#
#Script para comprobar si un usuario es de tipo profesor
USER=$1
GROUP=$(grep ^$USER: /etc/passwd | awk -F":" '{print $4}')
if [ $GROUP = "502" ]
then
exit 0
fi
exit 1
```

El programa 3.15 utiliza un solo parámetro que es el de usuario ingresado a través de la línea de comandos de Linux. La variable GROUP se asigna a partir de una instrucción tipo grep la cual busca aquella línea que empiece con el argumento introducido en la línea de comandos en el archivo etc/passwd de la cual se imprime la columna cuatro y se compara con 502; si es cierta esta comparación regresara este shell script un cero y sino un uno.

Programa 3.16 (Quitando la asignación de un grupo a un profesor)

```
#freeprof.sh
#!/bin/bash
#
#Script para liberar a un profesor de un grupo
PROF=$1
GROUP=$2
for GRUPO in $(grep ^$PROF: /etc/passwd | awk -F":" '{print $5}')
do
  if [ $GROUP != $GRUPO ]
  then
    NGRUPO=$NGRUPO" "$GRUPO
  fi
done
/usr/bin/chfn -f "$NGRUPO" $PROF
exit $?
```

El programa 3.16 toma dos de los parámetros anteriormente obtenidos a través de la línea de comandos de la cual se sigue una estructura de control for donde la variable se establece a partir del archivo /etc/passwd con prefijo establecido como el nombre del profesor desde la cual se imprimirá la columna cinco que es la ubicación del /home del usuario. Aquí se aplica una sentencia de control if que compara la variable ingresada a través de la interfaz de comandos con la variable leída dentro del ciclo for y de ser verdadera la condición se asigna a NGRUPO un valor especificado por las variables NGRUPO y GRUPO. Después acabado el ciclo for se usa el comando chfn para cambiar la información conocida en el ambiente Linux como finger aplicándose al nombre real del usuario.

Programa 3.17 (Obteniendo el profesor asignado a un grupo)

```
#getprof.sh
#!/bin/bash
#
#Script para obtener el profesor asignado a un grupo
GROUP=$1
rm /tmp/prf$GROUP.txt
for PROF in $(cat /etc/passwd | awk -F":" '{print $1":"$5}' | grep $GROUP | awk -F":" '{print $1}')
do
  echo $PROF >> /tmp/prf$GROUP.txt
done
exit 0
```

El programa 3.17 borra el archivo temporal /tmp/prf$GROUP.txt y como ya se revisó, se utilizó la variable GROUP introducida a través de la línea de comandos para el nombre de este archivo.

Después se entra a un for donde la variable a ser pasada se obtiene del archivo etc/passwd del cual se imprimen las columnas uno y cinco , se busca aquellas líneas que tengan la variable GROUP y se imprime solo la columna uno.

La acción a realizarse es la de ingresar la variable PROF al final del archivo borrado anteriormente. Dicho archivo será utilizado posteriormente

Programa 3.18 (Obteniendo la lista de profesores existentes)

```
#getallprof.sh
#!/bin/bash
#
#Script para obtener a todos los profesores
cat /etc/passwd | awk -F":" '{print $1":"$4}' | grep :502$ | awk -F":" '{print $1}' > /tmp/profesores.txt
exit 0
```

El programa 3.18 imprime primero las columnas uno y cuatro del archivo etc/passwd , luego busca en esta salida la cadena :502 y de aquí solo imprime la columna uno que es el nombre del usuario y lo direcciona al archivo tmp/profesores.txt y da de salida un 0.

3.3.1.4 ® PROGRAMAS – ALUMNOS

Programa 3.19 (Vaciando las cuentas de los alumnos)

```
#emptyhome.sh
#!/bin/bash
#
#Script para vaciar las cuentas de alumnos
cat /etc/group | grep ^CTRL | awk -F":" '{print $3}' > /tmp/emptygroups.txt
CUENTAS=`cat /etc/passwd | awk -F":" '{print $1":"$4}' | grep -f /tmp/emptygroups.txt | awk -F":" '{print $1}'`
for N in $CUENTAS ; do
rm -rf /home/$N/*
done
exit $?
```

El programa 3.19 saca del archivo etc/group los que empiezan con CTRL de ahí imprime la columna tres donde esta el gid y lo manda al archivo /tmp/emptygroups.txt. Después la variable de CUENTAS toma su valor a partir del archivo /etc/passwd donde mediante la instrucción awk con separador de campo ":" se imprimen la columna uno y cuatro donde esta el nombre de usuario y su identificador de grupo, después mediante el grep –f se obtienen patrones del archivo /tmp/emptygroups.txt línea por línea comparándolos De allí se imprime la columna 1. Se establece un ciclo for usándose la variable CUENTAS y borra todo en forma forzada y recursiva desde la partición /home/$N y se pasa la salida mediante el comando exit $?.

3.3.2 ® PROGRAMAS JAVA

No es la idea de este proyecto detallar explicaciones sobre JAVA, por lo tanto, sólo daremos una mirada muy superficial a esta herramienta. El lenguaje de programación JAVA, es un lenguaje orientado a objetos y fue diseñado por la compañía Sun Microsystems Inc, con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadoras formadas por más de un tipo de computadora, ya sean PC, MAC's, estaciones de trabajo, etc.), y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma.

Algunas definiciones importantes en JAVA:

Objeto: Es una persona, lugar o cosa. Ejemplo: Una manzana.
Todos los objetos tienen un estado y un comportamiento. Esto es muy importante. Para el ejemplo de la manzana, la manzana (hablando de una manzana normal de color rojo) recien bajada de un arbol, está completa, es dura y de color rojo. Este es su estado. Uno sabe que si se muerde la manzana va a sonar un "crunch", y esto se puede interpretar como un comportamiento. Después de haberla mordido la manzana cambió de estado. Los objetos pueden tener más de un comportamiento y más de un estado.

Clases: Un grupo de objetos con las mismas características. Una clase contiene:
Una descripción de las variables internas de los objetos de la clase.
Las operaciones que se pueden aplicar al objeto de la clase.

Encapsulación: Principio en el cual un objeto esconde la información de su implementación a otro objeto y a los usuarios. En otras palabras los objetos no te dicen cómo estan hechos por dentro. La encapsulación tiene dos beneficios primordiales:
Modularidad. El código de un objeto puede ser escrito y se puede mantener independiente del código de otros objetos.Un objeto se puede mover de sistema en sistema, se puede quitar, modificar y volver a colocar sin alterar el sistema general.
Esconder la información (*information hiding*). Un objeto tiene una interfase con la que otros objetos se pueden comunicar, pero puede mantener información privada para sí misma que puede cambiar en cualqueir momento sin afectar a los objetos que dependen de esta.

Herencia: Principio por el cual una clase se puede derivar de otra clase ya existente, y hereda las características del padre. Un ejemplo:

Tengo el objeto fruta, con algunas propiedades inherentes a la fruta. Luego tengo el objeto manzana, que es hijo de fruta y por lo tanto hereda las propiedades de la fruta pero puede tener otras propiedades propias de la manzana. Este es un ejemplo sencillo pero más o menos ilustra lo que estoy tratando de explicar. El hijo hereda las características del papá, pero tiene otras propias y mejora (o emperora) algunas características del papá.

3.3.2.1 ® PROGRAMAS – DESDE INTERNET

A continuación se muestran las clases contenidas en algunos programas desarrollados dentro del contenedor de SERVLETS de JAVA llamado Tomcat. Estos se ubican dentro de la dirección absoluta, del servidor de dominio:

/var/www/jakarta-tomcat-3.2/webapps/cdigital/WEB-INF/classes/samba/

Programa 3.20 (Programa comodín, interfaz con SCRIPTS SHELL)

```
//SAMBA.JAVA
package samba;
import java.io.*;
import java.util.Vector;
public class Samba {
  String path = "/var/www/jakarta-tomcat-3.2/smbscripts/";
  Runtime rt;
  Process ps;
  public Samba() {
    this.rt = Runtime.getRuntime();
  }
  public boolean isAdmin(String user) {
     Process ps;
     int ret;
     try {
       ps = this.rt.exec(this.path + "checkadmin.sh " + user);
       ps.waitFor();
       ret = ps.exitValue();
       if(ret == 0) {
```

```
            return true;
          }
        }catch(Exception e) {
          System.out.println(e);
        }
        return false;
    }
    public boolean isProfesor(String user) {
        Process ps;
        int ret;
        try {
          ps = this.rt.exec(this.path + "checkprf.sh " + user);
          ps.waitFor();
          ret = ps.exitValue();
          if(ret == 0) {
            return true;
          }
        }catch(Exception e) {
          System.out.println(e);
        }
        return false;
    }
    public boolean adduser(String user, String passwd, String group) {
        Process ps;
        int ret;
        try {
          ps = this.rt.exec(this.path + "adduser.sh " + user + " " + passwd + " " + group);
          ps.waitFor();
          ret = ps.exitValue();
          if(ret == 0) {
            return true;
          }
        }catch(Exception e) {
          System.out.println(e);
        }
        return false;
    }
    public boolean addgroup(String group) {
        Process ps;
        int ret;
        try {
          ps = this.rt.exec(this.path + "addgroup.sh " + group);
          ps.waitFor();
          ret = ps.exitValue();
          if(ret == 0) {
            return true;
          }
```

```
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }
  public boolean deluser(String user) {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "deluser.sh " + user);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }
  public boolean delgroup(String group) {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "delgroup.sh " + group);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }
  public boolean clearuser() {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "emptyhome.sh ");
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
```

```
    }
    return false;
  }
  public boolean checkuser(String user) {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "checkuser.sh " + user);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }

  public boolean checkPass(String user, String passwd) {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "checkpasswd.sh " + user + " " + passwd);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }
  public boolean changePass(String user, String passwd) {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "changepasswd.sh " + user + " " + passwd);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
```

```
    return false;
  }
  public boolean checkGroup(String group) {
    Process ps;
    int ret;
      try {
      ps = this.rt.exec(this.path + "checkgroup.sh " + group);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }
  public boolean asignProfesor(String profesor, String grupo) {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "asignprof.sh " + profesor + " " + grupo);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }
  public boolean freeProfesor(String profesor, String grupo) {
    Process ps;
    int ret;
    try {
      ps = this.rt.exec(this.path + "freeprof.sh " + profesor + " " + grupo);
      ps.waitFor();
      ret = ps.exitValue();
      if(ret == 0) {
        return true;
      }
    }catch(Exception e) {
      System.out.println(e);
    }
    return false;
  }
```

```
public Vector getAllProfs() {
   Process ps;
   int ret;
   String line;
   Vector profs = new Vector();
   try {
      ps = this.rt.exec(this.path + "getallprof.sh ");
      ps.waitFor();
      File file = new File("/tmp/profesores.txt");
      FileInputStream fis = new FileInputStream(file);
      BufferedInputStream bis = new BufferedInputStream(fis);
      DataInputStream dis = new DataInputStream(bis);
      while((line = dis.readLine()) != null) {
        profs.add(line);
      }
   }catch(Exception e) {
      System.out.println(e);
   }
   return profs;
}
public Vector getAlumnsGroup(String grupo) {
   Process ps;
   int ret;
   String line;
  Vector alumns = new Vector();
   try {
      ps = this.rt.exec(this.path + "getalumns.sh " + grupo);
      ps.waitFor();
      File file = new File("/tmp/alumns" + grupo + ".txt");
      FileInputStream fis = new FileInputStream(file);
      BufferedInputStream bis = new BufferedInputStream(fis);
      DataInputStream dis = new DataInputStream(bis);
      while((line = dis.readLine()) != null) {
        alumns.add(line);
      }
   }catch(Exception e) {
      System.out.println(e);
   }
   return alumns;
}
public Vector getGroups() {
   Process ps;
   int ret;
   String line;
   Vector groups = new Vector();
   try {
      ps = this.rt.exec(this.path + "getgroups.sh ");
```

```
        ps.waitFor();
        File file = new File("/tmp/sambagroups.txt");
        FileInputStream fis = new FileInputStream(file);
        BufferedInputStream bis = new BufferedInputStream(fis);
        DataInputStream dis = new DataInputStream(bis);
        while((line = dis.readLine()) != null) {
          groups.add(line);
        }
      }catch(Exception e) {
        System.out.println(e);
      }
      return groups;
    }
    public Vector getAllGroups() {
      Process ps;
      int ret;
      String line;
      Vector groups = new Vector();
      try {
        ps = this.rt.exec(this.path + "getgroups.sh ");
        ps.waitFor();
        File file = new File("/tmp/sambagroups.txt");
        FileInputStream fis = new FileInputStream(file);
            BufferedInputStream bis = new BufferedInputStream(fis);
        DataInputStream dis = new DataInputStream(bis);
        while((line = dis.readLine()) != null) {
          groups.add(line);
        }
      }catch(Exception e) {
        System.out.println(e);
      }
      return groups;
    }
    public Vector getGroups(String profesor) {
      Process ps;
      int ret;
      String line;
      Vector groups = new Vector();
      try {
        ps = this.rt.exec(this.path + "groupsprof.sh " + profesor);
        ps.waitFor();
        File file = new File("/tmp/group"+profesor+".txt");
        FileInputStream fis = new FileInputStream(file);
            BufferedInputStream bis = new BufferedInputStream(fis);
        DataInputStream dis = new DataInputStream(bis);
        while((line = dis.readLine()) != null) {
          groups.add(line);
```

```
        }
      }catch(Exception e) {
        System.out.println(e);
      }
      return groups;
    }
    public Vector getProfesor(String grupo) {
      Process ps;
      int ret;
      String line;
      Vector profs = new Vector();
      try {
        ps = this.rt.exec(this.path + "getprof.sh " + grupo);
        ps.waitFor();
        File file = new File("/tmp/prf"+grupo+".txt");
        FileInputStream fis = new FileInputStream(file);
            BufferedInputStream bis = new BufferedInputStream(fis);
        DataInputStream dis = new DataInputStream(bis);
        while((line = dis.readLine()) != null) {
          profs.add(line);
        }
      }catch(Exception e) {
        System.out.println(e);
      }
      return profs;
    }
    public String getGroupAlumn(String alumno) {
      Process ps;
      int ret;
      String line = "";
      String line2 = "";
      Vector groups = new Vector();
      try {
        ps = this.rt.exec(this.path + "getalumngroup.sh " + alumno);
        ps.waitFor();
        File file = new File("/tmp/grp"+alumno+".txt");
        FileInputStream fis = new FileInputStream(file);
            BufferedInputStream bis = new BufferedInputStream(fis);
        DataInputStream dis = new DataInputStream(bis);
        while((line = dis.readLine()) != null) {
          line2 = line;
        }
      }catch(Exception e) {
        System.out.println(e);
      }
      return line2;
    }
```

}

En el programa 3.20, la palabra reservada ***package*** funciona para generar un paquete el cual puede ser usado por otras aplicaciones. A primera vista se ve como se incluyeron los SHELL SCRIPT en esta clase. Los paquetes importados **java.io** y ***java.util.Vector*** permiten respectivamente manejar flujos de entrada/salida y el uso de vectores que son arreglos que pueden crecer y no tienen dimensiones fijas. Al establecerse como una clase pública puede ser vista esta clase desde otras clases.

La herencia es una de las mejores armas de JAVA pero este solo permite herencia sencilla, es decir, que cada clase sólo puede tener una superclase. Pero el uso de paquetes o clases definidos por nosotros actúa como las bibliotecas en C las cuales pueden ser llamadas a través de un simple ***include***, en este caso los programas JSP usaran esta clase llamándola desde el encabezado. En la clase publica SAMBA se define una cadena con nombre ***path*** la cual define la ruta absoluta en la cual se encuentran todos los programas de SHELL SCRIPT. Se crean dos objetos a continuación, uno de tipo ***runtime*** y otro de tipo ***process***. Al generarse una instancia de esta clase permite a la aplicación hacer una interfaz con el sistema operativo o medio ambiente donde la aplicación corre. El metodo ***exec*** se usa para realizar el comando especificado por la cadena en un proceso por separado. Este método regresa un objeto del tipo ***process*** por lo cual no fue necesario hacer ningún ***cast*** o conversión de objetos.

El segundo objeto del tipo ***process*** permite como lo dice su nombre manejar procesos desde sus entradas, sus salidas, sus tiempos de espera, comparar el status de salida de un proceso y destruir a dicho proceso.

El método ***getRunTime()*** regresa un objeto ***runtime*** asociado con la aplicación JAVA en ejecución. La mayoría de los métodos ***runtime*** necesitan para ser invocados de un objeto ***runtime*** actual ya que son métodos de instancia y no de clase.

Haciéndose referencia a que los métodos de instancia no tienen la palabra clave ***static*** en su declaración.

Se usa un método constructor para el objeto rt por lo mencionado anteriormente acerca de los métodos de instancia ***Runtime.getRunTime()***

Los métodos declarados boleanos regresan un valor de acuerdo con esto y para ejemplo el método publico ***isAdmin*** con un argumento especificado como de cadena y reconocido como a ***user*** el cual al ver el programa SHELL SCRIPT asociado recibía de parámetro también a ***user***.

En este método se declaran dos variable que son locales; una de tipo ***process ps*** y otra entera llamada ***ret*** para hacer las comparaciones mediante una sentencia de control ***if***. El uso de las excepciones en JAVA impiden que un programa se detenga en su ejecución por algún error, esto se logra mediante las sentencias ***try*** y ***match***.

Se iguala ***ps*** a un método aplicado sobre un objeto ***runtime*** y dicho método es ***exec*** el cual regresa un objeto del tipo ***process*** por lo cual hay compatibilidad.

Los argumentos son la ruta establecida por el objeto actual y expresado por la palabra clave ***this*** y la variable ***path*** concatenada con el SHELL SCRIPT específico. También en esta parte se pone el segundo argumento que en este caso es el usuario determinado por la variable ***user***.

El método ***waitFor()*** hace que el flujo actual espere hasta que el objeto **process** ejecutado **termine**.

El método ***exitValue()*** recoge el valor de salida del proceso y este valor será usado para validar las banderas establecidas a través de los programas SHELL SCRIPT que dicen si su objetivo fue logrado y su valor de salida se establecía mediante la instrucción ***exit (valor.)*** Aquí se ve que la salida sirve para una sentencia de control conocida como ***if***. Las salidas de este ***if*** son un falso o un verdadero ya que la función se estableció como una de tipo boleano.

Todos los métodos boleanos usados actúan de la misma forma como se indico aquí, y su conexión es en realidad sencilla.
Los métodos realizados con vectores usan los flujos de entrada y salida como se ve en la clase SAMBA, usan métodos conocidos como ***readLine()*** y ***add()*** los cuales se usan con un objeto del tipo ***dataInputStream.*** Primero se genera un objeto de tipo File para hacer referencia al archivo a ser usado. Después se usa un constructor del tipo ***FileInputStream(File)*** el cual permite leer bytes desde un fichero. Después se liga esta con un objeto *del tipo* ***BufferedInputStream*** la cual aporta como lo dice su nombre un buffer que sirve como memoria

intermedia para lecturas futuras lo que aumenta la velocidad de ejecución del programa ya que los datos obtenidos del origen pueden ser depositados en este buffer en bloques más grandes que los que realmente esta leyendo el programa, así que si el programa necesita leer mas datos no tendrá que esperar por ellos porque ya los tendrá en el buffer.

Después se liga este buffer a un objeto de la clase ***DataInputStream*** derivada indirectamente de ***InputStream*** que permite leer un flujo de entrada subordinado datos de cualquier tipo primitivo lo cual se considera como un filtro. Entonces a dicho objeto se le aplica el método ***readLine()*** el cual lee a partir de un flujo y cuando se encuentra con el final del mismo retorna la constante ***null***. Después se aplica el método ***add*** con argumento ***line*** sobre un objeto ya creado en estos casos fue del tipo vector ya que es un arreglo incremental sin limites fijos y permite generar listas de usuarios y profesores en programas JSP.

El programa 3.21 se encuentra también en la misma ruta que la clase SAMBA. Esta clase sirve para validar contraseñas y cambiarlas si es posible.

Programa 3.21 (Validando y cambiado contraseñas)

```
package samba;
public class Password extends Samba {
  public String message;
  public boolean checkFields(String user,String passwd1, String passwd2, String passwd3) {
    if(!checkPass(user,passwd1)) {
      this.message = "Su contrase&ntilde;a actual es incorrecto";
      return false;
    }
    if(!passwd2.equals(passwd3)) {
      this.message = "El contrase&ntilde;a nuevo no coincide";
      return false;
    }
    if(!changePass(user,passwd2)) {
      this.message = "No fue posible cambiar la contrase&ntilde;a";
      return false;
    }
    this.message = "La contrase&ntilde;a ha sido cambiada";
    return true;
  }
  public String getMessage() {
```

```
    return this.message;
  }
}
```

El programa 3.22 se encuentra dentro del mismo paquete que la clase SAMBA. Todos los métodos públicos definidos dentro de clases incluidas en un mismo paquete pueden ser vistas desde otras clases dentro del mismo paquete.

Trabaja con vectores de archivos y directorios. Gracias al argumento pasado al método listDir se ubica la cuenta del usuario. De aquí la sentencia de control for permite avanzar a través de los documentos del usuario y la sentencia de control if determina si el documento es un archivo o un directorio. Los métodos isDirectory() e isHidden() se encuentran en la API* de java. La lista de archivos o directorios es importante ya que se visualiza en las pantallas del sistema permitiendo a los usuarios realizar acciones sobre estos documentos al seleccionarlos con el "mouse".

Programa 3.22 (Navegando entre archivos y directorios)

```
package samba;
import java.io.*;
import java.util.*;
public class Cuenta {
  Vector dirs = new Vector();
  Vector adirs = new Vector();
  Vector files = new Vector();
  Vector afiles = new Vector();
  File file;
  public void listDir(String path) {
    String list[] = (new File(path)).list();
    for(int i=0; i<list.length; i++) {
      if((new File(path + "/" + list[i])).isDirectory()) {
        dirs.add(list[i]);
        adirs.add(path + "/" + list[i]);
      }else {
        if(!(new File(path + "/" + list[i])).isHidden()) {
          files.add(list[i]);
          afiles.add(path + "/" + list[i]);
        }
      }
    }
  }
```

```
  public Vector getDirs() {
    return this.dirs;
  }
  public Vector getFiles() {
    return this.files;
  }
  public Vector getDirsAbs() {
    return this.adirs;
  }
  public Vector getFilesAbs() {
    return this.afiles;
  }
}
```

El programa 3.23 sirve para mostrar las publicaciones realizadas por los profesores para que sus alumnos puedan verlas.

Programa 3.23 (Mostrando en Internet las publicaciones, tareas y prácticas dejadas por el profesor)

```
package samba;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class PublicacionServlet extends HttpServlet {
  public void doPost (HttpServletRequest request,
                      HttpServletResponse response)
  throws IOException,ServletException {
    String titulo,nota,grupo;
    PrintWriter out = response.getWriter ();
    FileUpload fu = new FileUpload (request);
    response.setContentType ("text/html");
    response.setHeader ("Cache-control", "no-cache");
    boolean ok = fu.upload ();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Laboratorio de Control Digital</title>");
    out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">");
    out.println("<style type=\"text/css\">");
    out.println("<!--");
    out.println(".justy {");
    out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");     out.println("       font-size: 11px;");
    out.println("       color: #FFFFFF;");
    out.println("       text-align: justify;");
    out.println("}");
```

```
out.println(".justback {");
out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("       font-size: 10px;");
out.println("       color: #333333;");
out.println("       text-align: justify;");
out.println("}");
out.println(".mininota {");
out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("       font-size: 10px;");
out.println("       color: #333333;");
out.println("       text-align: left;");
out.println("}");
out.println(".tituloscontent {");
out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("       font-size: 10px;");
out.println("       color: #FFFFFF;");
out.println("       background-color: #006699;");
out.println("}");
out.println(".links {");
out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("       font-size: 10px;");
out.println("       color: #333333;");
out.println("       text-align: center;");
out.println("}");
out.println(".centrado {");
out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("       font-size: 9px;");
out.println("       text-align: center;");
out.println("       color: #622229;");
out.println("}");
out.println("-->");
out.println("</style>");
out.println("<script language=\"JavaScript\" type=\"text/JavaScript\">");
out.println("");
out.println("<!--");
out.println("");
out.println("function doSubmit() {");
out.println("   document.forms[0].submit();");
out.println("}");
out.println("");
out.println("function MM_preloadImages() { //v3.0");
out.println("  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();");
out.println("    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)");
out.println("    if (a[i].indexOf(\"#\")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}");
out.println("}");
out.println("//-->");
out.println("");
out.println("</script>");
out.println("</head>");
out.println("");
out.println("<body leftmargin=\"0\" topmargin=\"0\" marginwidth=\"0\" marginheight=\"0\">");
out.println("<table width=\"100%\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\">");
```

```
    out.println("  <tr> ");
    out.println("    <td background=\"images/backtitles.jpg\" bgcolor=\"FFFFFF\"> ");
    out.println("                <td  align=\"right\"  valign=\"top\"  background=\"images/backtitles.jpg\"  bgcolor=\"FFFFFF\"><img
src=\"images/spacer.gif\" width=\"15\" height=\"8\"></td>");
    out.println("  </tr>");
    out.println("</table>");
    out.println("<table width=\"100%\" border=\"0\" cellpadding=\"5\" cellspacing=\"2\">");
    out.println("  <tr> ");
    out.println("    <td align=\"center\" valign=\"top\"> ");
    out.println("      <table width=\"100%\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\">");
    out.println("        <tr> ");
    out.println("          <td height=\"2\" class=\"justy\"> ");
    out.println("            <div align=\"right\"></div></td>");
    out.println("        </tr>");
    out.println("        <tr valign=\"top\"> ");
    out.println("                  <td  bgcolor=\"#FFFFFF\"  class=\"justback\">  <table  width=\"100%\"  border=\"0\"  cellspacing=\"0\"
cellpadding=\"0\">");
    out.println("              <tr> ");
    out.println("                <td align=\"center\" valign=\"middle\"> <p class=\"justback\"><span class=\"justback\"> ");
    out.println("                  </span></p>");
    out.println("                <table width=\"300\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\">");
    out.println("                  <tr> ");
     out.println("                          <td colspan=\"2\" align=\"center\" valign=\"middle\"><img src=\"images/spacer.gif\" width=\"10\"
height=\"40\"></td>");
    out.println("                  </tr>");
    out.println("                  <tr> ");
    out.println("                    <td height=\"16\" background=\"images/backtopdialog.jpg\"> ");
    out.println("                      <div align=\"left\"><img src=\"images/backcornerleft.jpg\" width=\"76\" height=\"20\"></div></td>");
     out.println("                                  <td  align=\"right\"  background=\"images/backtopdialog.jpg\"  bgcolor=\"FFFFFF\"><img
src=\"images/backcornerdialog.jpg\" width=\"76\" height=\"20\"></td>");
    out.println("                  </tr>");
    out.println("                  <tr bgcolor=\"eeeeee\"> ");
    out.println("                    <td colspan=\"2\"> <form name=\"form1\" method=\"post\" action=\"profesor/profesor.jsp\">");
    out.println("                      <table width=\"100%\" border=\"0\" cellspacing=\"0\" cellpadding=\"4\">");
    out.println("                        <tr> ");
    out.println("                          <td align=\"center\"><span class=\"centrado\"><font color=\"#666666\"><strong>");
    if (!ok) {
      out.println (fu.getErrString ());
    }
    else {
      HashMap m = fu.getMimeParts ();
      MimeBodyPart part = (MimeBodyPart) m.get ("titulo");
      titulo = part.getDataAsString ();
      part = (MimeBodyPart) m.get ("nota");
      nota = part.getDataAsString ();
      part = (MimeBodyPart) m.get ("grupo");
      grupo = part.getDataAsString ();
```

```
      part = (MimeBodyPart) m.get ("userfile");
      String file = part.getFileName ();
      if((new File("/home/samba/pub/"+grupo+"/"+titulo)).exists()) {
        out.println ("La publicaci&oacute;n ya existe");
      } else {
        (new File("/home/samba/pub/"+grupo+"/adjuntos")).mkdirs();
        boolean saved = part.saveFile("samba/pub/" + grupo + "/adjuntos");
        if (saved) {
          file = part.getFileName ();
        }
        try {
          PrintStream output = new PrintStream(new FileOutputStream("/home/samba/pub/"+grupo+"/"+titulo));
          output.println(nota);
          if(saved)
            output.println("Archivo adjunto: <a href=\"/home/samba/pub/"+grupo+"/adjuntos/"+file+"\">"+file+"</a>");
          output.close();
        }catch(Exception e) {
          System.out.println(e);
        }
        out.println ("Su informaci&oacute;n ha sido registrada");
      }
    }
    out.println("                    </strong></font></span><font color=\"#666666\">  ");
    out.println("                    </font></td>");
    out.println("                  </tr>");
    out.println("                  <tr> ");
    out.println("                    <td> <table width=\"100%\" border=\"0\" align=\"center\" cellpadding=\"0\" cellspacing=\"0\">");
    out.println("                      <tr> ");
     out.println("                        <td align=\"center\"> <a href=\"javascript:doSubmit();\"> <img src=\"images/aceptar.jpg\"
border=\"0\" width=\"88\" height=\"22\"></a></td>");
    out.println("                      </tr>");
    out.println("                    </table></td>");
    out.println("                  </tr>");
    out.println("                </table>");
    out.println("              </form></td>");
    out.println("          </tr>");
    out.println("          </table></td>");
    out.println("        </tr>");
    out.println("        <tr align=\"center\" valign=\"middle\"> ");
    out.println("          <td> </td>");
    out.println("        </tr>");
    out.println("      </table>");
    out.println("      </td>");
    out.println("    </tr>");
    out.println("  </table>");
    out.println("  ");
    out.println("  </td>");
```

```
    out.println("    <td width=\"2%\" align=\"left\" valign=\"top\">  </td>");
    out.println("  </tr>");
    out.println("  <tr> ");
    out.println("    <td colspan=\"2\" align=\"center\" valign=\"top\"> </td>");
    out.println("  </tr>");
    out.println("</table>");
    out.println("</body>");
    out.println("</html>");
  }

}
```

El programa 3.24 sirve para importar archivos a través de una conexión a Internet. Los alumnos pueden realizar esta función en sus cuentas respectivas.

Programa 3.24 (Importando archivos desde Internet)

```
package samba;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class FileUploadServlet extends HttpServlet {
public void doPost (HttpServletRequest request,
HttpServletResponse response)
throws IOException,ServletException {
    PrintWriter out = response.getWriter ();
    FileUpload fu = new FileUpload (request);
    response.setContentType ("text/html");
    response.setHeader ("Cache-control", "no-cache");
    boolean ok = fu.upload ();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Laboratorio de Control Digital</title>");
    out.println("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">");
    out.println("<style type=\"text/css\">");
    out.println("<!--");
    out.println(".justy {");
    out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");     out.println("       font-size: 11px;");
    out.println("       color: #FFFFFF;");
    out.println("       text-align: justify;");
    out.println("}");
    out.println(".justback {");
    out.println("       font-family: Verdana, Arial, Helvetica, sans-serif;");     out.println("       font-size: 10px;");
    out.println("       color: #333333;");
```

```
out.println("        text-align: justify;");
out.println("}");
out.println(".mininota {");
out.println("        font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("        font-size: 10px;");
out.println("        color: #333333;");
out.println("        text-align: left;");
out.println("}");
out.println(".tituloscontent {");
out.println("        font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("        font-size: 10px;");
out.println("        color: #FFFFFF;");
out.println("        background-color: #006699;");
out.println("}");
out.println(".links {");
out.println("        font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("        font-size: 10px;");
out.println("        color: #333333;");
out.println("        text-align: center;");
out.println("}");
out.println(".centrado {");
out.println("        font-family: Verdana, Arial, Helvetica, sans-serif;");    out.println("        font-size: 9px;");
out.println("        text-align: center;");
out.println("        color: #622229;");
out.println("}");
out.println("-->");
out.println("</style>");
out.println("<script language=\"JavaScript\" type=\"text/JavaScript\">");
out.println("");
out.println("<!--");
out.println("");
out.println("function doSubmit() {");
out.println("   document.forms[0].submit();");
out.println("}");
out.println("");
out.println("function MM_preloadImages() { //v3.0");
out.println("  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();");
out.println("    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)");
out.println("    if (a[i].indexOf(\"#\")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}");
out.println("}");
out.println("//-->");
out.println("");
out.println("</script>");
out.println("</head>");
out.println("");
out.println("<body leftmargin=\"0\" topmargin=\"0\" marginwidth=\"0\" marginheight=\"0\">");
out.println("<table width=\"100%\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\">");
out.println("  <tr> ");
out.println("    <td background=\"images/backtitles.jpg\" bgcolor=\"FFFFFF\"> ");
```

```
    out.println("                <td align=\"right\" valign=\"top\" background=\"images/backtitles.jpg\" bgcolor=\"FFFFFF\"><img
src=\"images/spacer.gif\" width=\"15\" height=\"8\"></td>");
    out.println("  </tr>");
    out.println("</table>");
    out.println("<table width=\"100%\" border=\"0\" cellpadding=\"5\" cellspacing=\"2\">");
    out.println("  <tr> ");
    out.println("    <td align=\"center\" valign=\"top\"> ");
    out.println("      <table width=\"100%\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\">");
    out.println("        <tr> ");
    out.println("          <td height=\"2\" class=\"justy\"> ");
    out.println("            <div align=\"right\"></div></td>");
    out.println("        </tr>");
    out.println("        <tr valign=\"top\"> ");
    out.println("                  <td bgcolor=\"#FFFFFF\" class=\"justback\"> <table width=\"100%\" border=\"0\" cellspacing=\"0\"
cellpadding=\"0\">");
    out.println("              <tr> ");
    out.println("                <td align=\"center\" valign=\"middle\"> <p class=\"justback\"><span class=\"justback\"> ");
    out.println("                  </span></p>");
    out.println("                <table width=\"300\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\">");
    out.println("                  <tr> ");
     out.println("                    <td colspan=\"2\" align=\"center\" valign=\"middle\"><img src=\"images/spacer.gif\" width=\"10\"
height=\"40\"></td>");
    out.println("                  </tr>");
    out.println("                  <tr> ");
    out.println("                    <td height=\"16\" background=\"images/backtopdialog.jpg\"> ");
    out.println("                      <div align=\"left\"><img src=\"images/backcornerleft.jpg\" width=\"76\" height=\"20\"></div></td>");
     out.println("                                    <td align=\"right\" background=\"images/backtopdialog.jpg\" bgcolor=\"FFFFFF\"><img
src=\"images/backcornerdialog.jpg\" width=\"76\" height=\"20\"></td>");
    out.println("                  </tr>");
    out.println("                  <tr bgcolor=\"eeeeee\"> ");
    out.println("                    <td colspan=\"2\"> <form name=\"form1\" method=\"post\" action=\"alumno/cuenta.jsp\">");
    out.println("                      <table width=\"100%\" border=\"0\" cellspacing=\"0\" cellpadding=\"4\">");
    out.println("                        <tr> ");
    out.println("                          <td align=\"center\"><span class=\"centrado\"><font color=\"#666666\"><strong>");
    if (!ok) {
      out.println (fu.getErrString ());
    }
    else {
      HashMap m = fu.getMimeParts ();
      MimeBodyPart part = (MimeBodyPart) m.get ("user");
      String user = part.getDataAsString ();
      part = (MimeBodyPart) m.get ("userfile");
      String file = part.getFileName ();
      boolean saved = part.saveFile (user);
      setPerm ("/home/" + user + "/" + file, user);
      if (!saved)
        out.println ("Ha ocurrido un error");
```

```
      else {
        out.println ("El archivo se ha guardado exitosamente");
        file = part.getFileName ();
        setPerm ("/home/" + user + "/" + file, user);
      }
    }
    out.println("                    </strong></font></span><font color=\"#666666\">  ");
    out.println("                    </font></td>");
    out.println("                  </tr>");
    out.println("                  <tr> ");
    out.println("                    <td> <table width=\"100%\" border=\"0\" align=\"center\" cellpadding=\"0\" cellspacing=\"0\">");
    out.println("                      <tr> ");
     out.println("                        <td align=\"center\"> <a href=\"javascript:doSubmit();\"> <img src=\"images/aceptar.jpg\"
border=\"0\" width=\"88\" height=\"22\"></a></td>");
    out.println("                      </tr>");
    out.println("                    </table></td>");
    out.println("                  </tr>");
    out.println("                </table>");
    out.println("              </form></td>");
    out.println("            </tr>");
    out.println("          </table></td>");
    out.println("        </tr>");
    out.println("        <tr align=\"center\" valign=\"middle\"> ");
    out.println("          <td> </td>");
    out.println("        </tr>");
    out.println("      </table>");
    out.println("    </td>");
    out.println("    <td width=\"2%\" align=\"left\" valign=\"top\">  </td>");
    out.println("  </tr>");
    out.println("  <tr> ");
    out.println("    <td colspan=\"2\" align=\"center\" valign=\"top\"> </td>");
    out.println("  </tr>");
    out.println("</table>");
    out.println("</body>");
    out.println("</html>");
  }
  public void setPerm (String file, String user) {
    try {
      Runtime rt = Runtime.getRuntime ();
      Process ps = rt.exec ("chown " + user + ":control " + file);
      ps.waitFor ();
      ps = rt.exec ("chmod 744 " + file);
      ps.waitFor ();
```

```
    } catch (Exception e) {
      System.out.println (e);
    }
  }
}
```

El programa 3.25 es usado en conjunto con la clase FileUpload para importar archivos y determinar si al leer datos se trata de un archivo y hace un manejo de directorios para guardar dicho archivo y puede sobrescribir o no un archivo. También permite compactar archivos.

Programa 3.25 (Manejando los archivos importados desde Internet)

```
package samba;
import java.io.*;
import java.util.zip.*;
import java.sql.*;
public class MimeBodyPart {
        private String name;
        private String contentType;
        private String fileName;
        private byte[] data;
        private boolean isFile;
        private String saveDir;
        private boolean overwrite = false;
        private final String tempDir = System.getProperty("java.io.tmpdir") + "/";
        private String err = "";
        public MimeBodyPart(String name, byte[] b, String fileName, String contentType) {
                this.name = name;
                this.data = b;
                this.contentType = contentType;
                this.fileName = fileName;
                this.isFile = (fileName != null);
                }
        /**Regresa la forma del atributo, el mismo como el nombre de entrada HTML. Regresa una cadena representando el
nombre del objeto */

        public String getName() {
        return name;
                }
        /**Regresa el objeto filename, si es un archivo. Regresa una cadena con su nombre de archivo de otra forma regresa una
cadena por default */
        public String getFileName() {
                if(isFile) {
                        return fileName;
```

```
                    }
/*Ya sea un campo de texto  o si esta vacio */
                }
        /**Regresa el tamaño del archivo si es que es un archivo. Regresa un entero con el tamaño del archivo, de otra forma regresa un cero  */
        public int getFileSize() {
                if(isFile && data!=null) {
                        return data.length;
                        }
                return 0;
                }
        /**Dice si el objeto es un archivo o no. Regresa un valor boleano de verdadero si es un archivo o falso de cualquier otra forma*/
        public boolean isFile() {
                return isFile;
                }
        /**Regresa un arreglo de byte. Sino es un archivo se hace una llamada a getDataAsString en su lugar. Un clon del arreglo de byte original es regresado de tal forma que no puede ser corrupto por manipulación directa del arreglo. Regresa un arreglo de bytes, clonado del original arreglo de bytes */
        public byte[] getBytes() {
                return (byte[])data.clone();
                }
        /**Regresa los datos de este objeto como una cadena.Se usa sino es un archivo.  Regresa una cadena representando el campo de datos */
        public String getDataAsString() {
                return new String(data,0,data.length);
                }
        /**Prepara el directorio para guardar el archivo. Crea un directorio especificado por saveDir si es que este no existe. saveDir : cadena representando la ruta del directorio*/
        public void setSaveDir(String saveDir) {
                this.saveDir = saveDir;
                File f = new File(saveDir);
                if(!f.exists()) {
                        f.mkdirs();  //
                        }
                }
        /**Regresa la ruta actual para guardar archivos. Regresa una cadena representando la ruta del directorio*/
        public String getSaveDir() {
        return saveDir;
                }
        /**Determina el comportamiento de archivos con el mismo nombre.      Se coloca en verdadero para sobrescribir un archivo ya nombrado */
        public void setOverwrite(boolean overwrite) {
                this.overwrite = overwrite;
                }
        public boolean getOverwrite() {
                return overwrite;
```

```
        }
/**Regresa un objeto de contenido*/
public String getContentType() {
        if(isFile) {
                return contentType;
                }
        else {
                return null;
                }
        }
/**Guarda los datos en un archivo temporal del sistema o en un directorio especificado por el método saveDir */
public boolean saveFile(String home) {
        if(isFile) {
                if(fileName != null && data != null) {
                        if(saveDir == null || saveDir.equals(""))
                                //saveDir = tempDir;
                                saveDir = "/home/" + home + "/";
                        try {
                                File f = new File(saveDir + fileName);
                                if( (f.exists() && overwrite) || !f.exists()) {
                                        OutputStream os = new FileOutputStream(f);
                                        os.write(data);
                                        os.close();
                                        }
                                else {
                        fileName = GeneralUtilities.generateFileName(saveDir,fileName);
                                        f = new File(saveDir + fileName);
                                        OutputStream os = new FileOutputStream(f);
                                        os.write(data);
                                        os.close();
                                        }
                                }
                        catch(IOException e) {
                                err = e.toString();
                                return false;
                                }
                        }
                } // isFile
        else {
                err = "Este objeto no es un archivo y no será guardado.";
                return false;
                }
        return true;
        }
/**Guarda el archivo en un directorio temporal o en un directorio especificado por el
                        método setSaveDir compactándolo en un archivo zip */
                                public boolean saveAsZipFile() {
```

```
                        if(isFile) {
                                if(fileName != null && data != null) {
                                if(saveDir == null || saveDir.equals(""))
                                saveDir = tempDir;
                                try {
                                File f = new File(saveDir + fileName + ".zip");
                                if( (f.exists() && overwrite) || !f.exists()) {
                                ZipOutputStream zos = new ZipOutputStream(new
FileOutputStream(f));
                                ZipEntry ze = new ZipEntry(saveDir + fileName);
                                zos.putNextEntry(ze);

                                zos.write(data);
                                zos.closeEntry();

                                zos.close();
                                fileName += ".zip";
                                }
                        else {
                fileName = GeneralUtilities.generateFileName(saveDir,fileName + ".zip");
                f = new File(saveDir + fileName);
                ZipOutputStream zos = new ZipOutputStream(new FileOutputStream(f));
                ZipEntry ze = new ZipEntry(saveDir + fileName);
                zos.putNextEntry(ze);
        zos.write(data);
        zos.closeEntry();
        zos.close();
                        }
                }
        catch(IOException e) {
        err = e.toString();
        return false;
                        }
                }
        } // isFile
        else {
        err = "Este objeto no es un archivo y no será guardado";
        return false;
                }
        return true;
                }
/**Guarda el archivo en un directorio temporal o en un directorio especificado por el método setSaveDir compactándolo en un archivo Gzip */
public boolean saveAsGZipFile() {
if(isFile) {
        if(fileName != null && data != null) {
                if(saveDir == null || saveDir.equals(""))
```

```
                    saveDir = tempDir;
                    try {
                    File f = new File(saveDir + fileName + ".gz");
                    if( (f.exists() && overwrite) || !f.exists()) {
            GZIPOutputStream zos = new GZIPOutputStream(new FileOutputStream(f));
                    zos.write(data);
                    zos.close();
                    fileName += ".gz";
                                                }
                                                else {
            fileName = GeneralUtilities.generateFileName(saveDir,fileName + ".gz");
            f = new File(saveDir + fileName);
            GZIPOutputStream zos = new GZIPOutputStream(new FileOutputStream(f));
            zos.write(data);
            zos.close();
                                }
                                        }
            catch(IOException e) {
            err = e.toString();
            return false;
                                }
                                        }
} // isFile
                    else {
                                err = "Este objeto no es un archivo y no será guardado";
                                return false;
                                }
                    return true;
                    }
public boolean saveToDatabaseAsBLOB(String driver, String url,
String uid, String pwd, String command){
boolean ok = true;
try {
Class.forName(driver);
Connection c = DriverManager.getConnection(url,uid,pwd);
if(c!=null) {
PreparedStatement ps = c.prepareStatement(command);
ps.setBytes(1,data);
ps.execute();
ps.close();
c.close();
                                        }
                                }
                    catch(SQLException e) {
                                err = e.toString();
                                ok = false;
                                }
```

```
                    catch(ClassNotFoundException e1) {
                        err = e1.toString();
                        ok = false;
                        }
                    return ok;
                    }
                public String getErrString() {
                    return err;
                    }
                } // mimebodypart
```

La siguiente clase genera un archivo con un nombre único en un directorio específico

Programa 3.26 (Generando los archivos importados en un directorio)

```
package samba;
import java.io.*;
public class GeneralUtilities {
public static String generateFileName(String dir,String fileName) {
                fileName = fileName.replace(' ','_');
                File f = new File(dir + fileName);
                while(f.exists()) {
                    fileName = "c_" + fileName;
                    f = new File(dir + fileName);
                    }
                return fileName;
                }
        } // GeneralUtilities
```

Se generaron tres carpetas principales, para distinguir entre los diferentes comportamientos que tienen cada uno de los roles de nuestro sistema. A saber estos son Administrador, Profesor y Alumno. En cada carpeta hay programas JSP que definen lo que cada rol puede hacer.

La carpeta con las imágenes esta en: /var/www/jakarta-tomcat-3.2/webapps/cdigital/images a las cuales se hace referencia en estos programas.

4. MANUALES DE OPERACIÓN

4.1 ® MANUAL DE OPERACIÓN - ALUMNOS

La finalidad de este manual es ayudarte a sacar provecho de las opciones y servicios que te ofrece el LCD tanto en el salón de clase como con su conexión vía Internet.

El LCD consta de siete máquinas conectadas en red las cuales cuentan con software especializado como son MATLAB, SIMNON y Cc principalmente.

Es necesario que cada vez que utilices estos programas direcciones su escritura y ejecución a la unidad G: que es la dirección del servidor Linux. Esto implica la seguridad de que si falla la máquina donde hiciste tu programa éste no se perderá. También puedes acceder a tu cuenta desde cualquiera de las siete máquinas del laboratorio.

El uso del sistema vía Internet permite que aproveches al máximo la clase con el profesor para explicaciones y evitar perder tiempo al escribir los programas a realizar en el pizarrón, tu profesor podrá subir a una sección que llamaremos **"Avisos"**, las tareas y/o archivos pertinentes así como simples mensajes que serán visibles para los alumnos del grupo únicamente.

La dirección de acceso vía Internet será http://bellman.fi-b.unam.mx donde la primera pantalla mostrará el logotipo del laboratorio con dos cajas de texto y dos botones. La primera de las cajas de texto pide tu nombre de usuario que te será asignado por tu profesor; la segunda caja de texto pide tu contraseña. Más adelante se explica como cambiar y personalizar esta contraseña.

Para cualquier problema relacionado con los temas vistos en este manual pueden recurrir a el cubículo con nombre Servicios Mantenimiento e Investigación de Laboratorios de Control con Pedro Gabriel Saucedo Uribe o escribir al correo electrónico pietro@dctrl.fi-b.unam.mx.

Al acceder desde Internet, la siguiente es la primera pantalla que aparece al entrar al sistema.



Fig. 4.1 – Pantalla inicial (Internet).

Al autentificarse la cuenta aparecerá la siguiente pantalla.



Fig. 4.2 – Pantalla principal con opciones de acceso (Internet).

En la Fig. 4.2 podemos observar cinco botones de comando los cuales ejecutan diversas acciones.

4.1.1 ® ACCEDIENDO A NUESTROS ARCHIVOS DESDE INTERNET

Al pulsar el botón de **"Archivos"** aparecerá un campo de texto en el cual podremos ver nuestros archivos y directorios residentes en la unidad G:. Para poder abrir uno de estos bastará con colocar el puntero del ratón sobre la flecha → y podremos ver ese archivo o el contenido de un directorio dependiendo de la naturaleza de esa aplicación.

El detalle importante de esta parte es que podrán ver los archivos realizados usualmente en el LCD desde Internet. Por cuestiones de seguridad, cuando los alumnos accedan a esta sección sólo tendrán permisos de lectura, es decir, podrán ver sus archivos, pero no modificarlos, pero podrán usar la técnica de copiar y pegar para pasar su archivo a un editor de texto cualquiera y modificarlo allí mismo, conservando el archivo fuente. Se recomienda renombrar este nuevo archivo modificado con otro nombre diferente del documento origen.

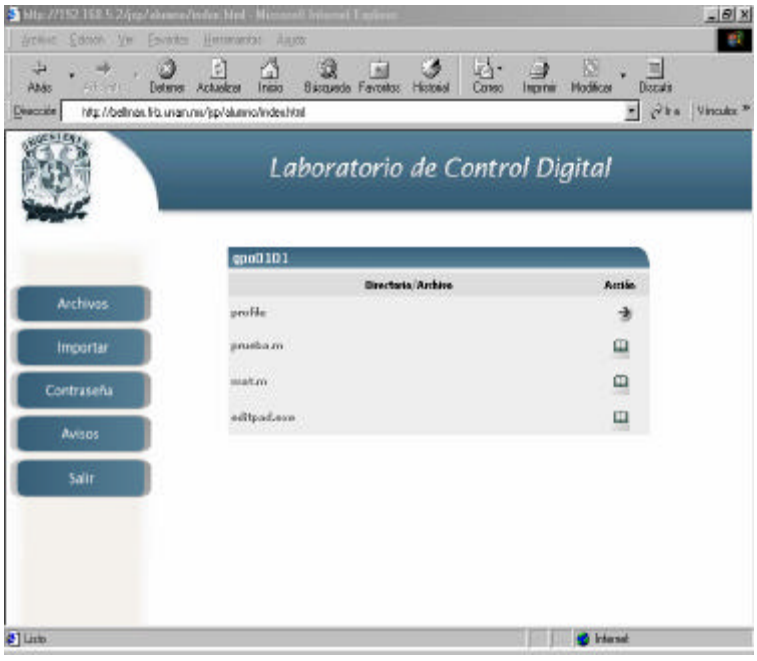


Fig. 4.3 – Pantalla que muestra los archivos creados por un equipo en particular en el LCD (desde Internet).

4.1.2 ® SUBIENDO ARCHIVOS AL SERVIDOR DEL LCD DESDE INTERNET

A continuación podemos subir nuestro archivo modificado a nuestra cuenta a través del botón con etiqueta **"Importar"** el cual funciona como un buscador, donde se podrá abrir la unidad donde se encuentre el archivo a subir a nuestra cuenta y colocarlo en el campo de texto y a continuación pulsar el botón con etiqueta **"Aceptar"**.

Si por error se subiera un archivo modificado con el mismo nombre del archivo origen a la hora de visualizar nuestra cuenta el nombre del archivo modificado aparecerá precedido de los caracteres "c-".

Si todo el procedimiento fue ejecutado correctamente, aparecerá un mensaje de confirmación de la acción realizada. Al darle **"Aceptar"** al botón de comando nos regresará a la pantalla donde aparece el contenido de las cuentas e inmediatamente veremos el archivo modificado que importamos dentro de la lista de archivos y directorios de nuestra cuenta. En las dos siguientes páginas se ven las pantallas que se despliegan al pulsar el botón de **"Importar".**



Fig. 4.4 – Pantalla para subir un archivo al servidor del LCD (desde Internet).

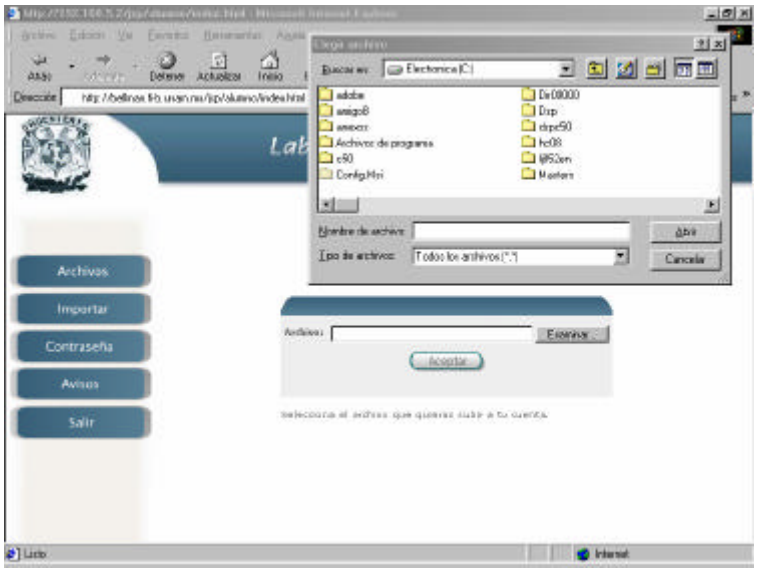


Fig. 4.5 – Subiendo un archivo al servidor del LCD (desde Internet).

Este buscador permite importar el archivo desde cualquier computadora con acceso a Internet a nuestra cuenta ubicada en el servidor del LCD.



Fig. 4.6 – Proceso de subir un archivo concluido correctamente.

4.1.3 ® CAMBIANDO LA CONTRASEÑA DE ACCESO

La siguiente opción a continuación es la de reasignar la contraseña de acceso. A cada uno de los grupos de alumnos formados al inicio de cada curso le es asignado un nombre de usuario y una contraseña para poder acceder a este sistema de administración para el LCD.

La contraseña inicial podrá ser cambiada por la acción del botón **"Contraseña"**.

Lo que veremos en la Fig. 4.7 es una pantalla donde se nos pide la contraseña actual a capturar en un campo de texto, el siguiente campo de texto nos pide la nueva contraseña y finalmente el último campo de texto nos pide confirmar la contraseña nueva. Al llenar correctamente todos los campos, se deberá pulsar el botón con la etiqueta **"Aceptar"** y la contraseña será cambiada. Consecuentemente aparecerá un mensaje de confirmación de la acción realizada. Fig. 4.8

Cualquier error al llenar los campos detendrá este proceso y la contraseña permanecerá sin cambios. De tener problemas en el futuro u olvidar su contraseña podrán pasar con Saucedo Uribe Pedro Gabriel en Servicios Mantenimiento e Investigación de Laboratorios de Control para resolver el problema.



Fig. 4.7 – Pantalla de cambio de contraseña (desde Internet).



Fig. 4.8 – Pantalla en la que se confirma un cambio de contraseña exitoso (desde Internet).

4.1.4 ® REVISANDO LOS AVISOS DEL PROFESOR

El botón que nos toca ahora tratar es el de **"Avisos"** el cual será utilizado por el alumno para tener acceso a los mensajes, avisos y/o archivos adjuntados por el profesor con la finalidad de optimizar el tiempo en laboratorio.

Por ejemplo, el profesor envía un archivo con el programa a realizar por su grupo. Los alumnos ya tendrán el programa escrito cuando lleguen al laboratorio y no perderán tiempo en escribir el programa y este será un tiempo ahorrado que podrá ser utilizado de diferentes formas más provechosas, como sería las explicaciones y demostraciones a cargo del profesor.

Una vez que pulsamos el botón **"Avisos"** aparecerá ante nosotros la Fig. 4.9 en la que se muestra una caja de texto con el título **"Publicaciones de gpo01"**, por ejemplo, y todos los alumnos de dicho grupo podrán ver el mensaje y/o programa.

En este campo de texto estará escrito el titulo del aviso en el lado izquierdo y del lado derecho habrá una columna con el titulo de visualizar donde al pulsar el icono respectivo se podrá ver el contenido del aviso.

Al estar dentro de esta sección tendremos dos formas para salir después de revisar su contenido. Una es pulsar el botón **"Salir"** o utilizando las propiedades de Internet como la de usar los botones de ATRAS y ADELANTE. Al pulsar el botón **"Salir"** estaremos fuera de nuestra cuenta y aparecerá la primera pantalla que como recordaremos nos pide el nombre de alumno y la contraseña. Pero si pulsamos el cuadro de ATRAS nos regresara a la pantalla anterior y así sucesivamente.

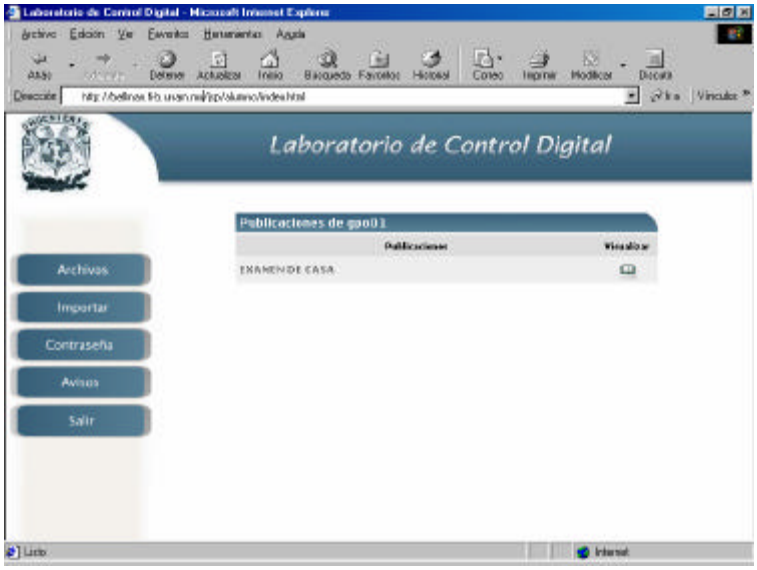


Fig. 4.9 – Pantalla de avisos del profesor (desde Internet).



Fig. 4.10 – Pantalla mostrando el contenido de un aviso dejado por el profesor

Al presionar el botón **"Visualizar"** podemos observar el contenido del mensaje o aviso dejado por el profesor, un ejemplo de esto se muestra en la Fig. 4.10.

4.1.5 ® SALIENDO DEL SISTEMA

Y finalmente el botón de salir cerrará nuestra sesión y aparecerá la pantalla mostrada en la Fig. 4.11, la cual mostrará una etiqueta con la inscripción de que la sesión en el sistema ha terminado y al pulsar aceptar regresaremos a la pantalla inicial donde se nos pide un nombre de usuario y contraseña.



Fig. 4.11 – Pantalla que muestra la salida del sistema para alumnos.

4.2 ® MANUAL DE OPERACIÓN - PROFESORES

Para el caso de los profesores, la forma de acceder desde Internet al sistema de administración del LCD es muy sencilla y muy similar a la de los alumnos. En la Fig. 4.12 se puede observar que la pantalla que aparecerá al entrar a la dirección http://bellman.fi-b.unam.mx es la misma que la de los alumnos.

La diferencia radica principalmente en el perfil del acceso, el cual le permitirá dejar avisos, tareas o archivos adjuntos a sus correspondientes grupos de laboratorio. Esta pantalla muestra dos campos los cuales deberán ser llenados con el nombre de usuario y su respectiva contraseña. Una vez hecho esto pulsamos el botón aceptar y si son correctos los campos se ingresará al sistema.

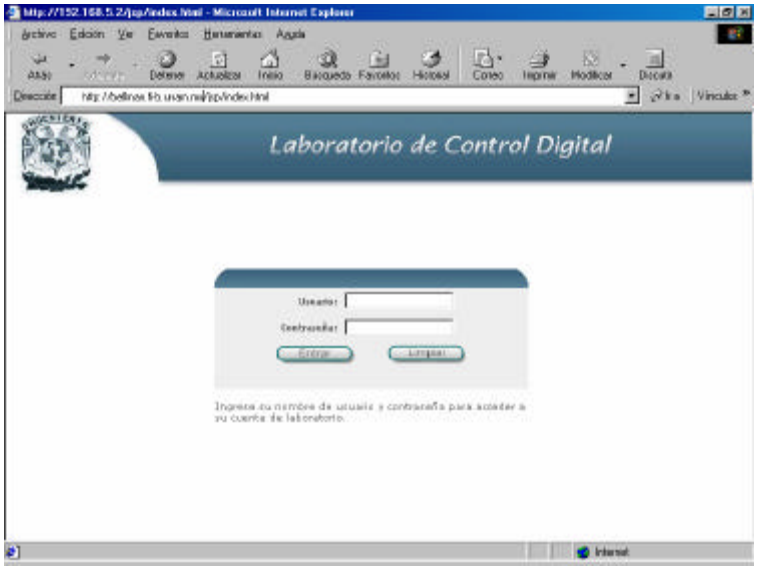


Fig. 4.12 – Pantalla inicial de acceso al sistema (a través de Internet)

4.2.1 ® ACCEDIENDO AL SISTEMA

Al profesor, al igual que a los grupos de alumnos, se les proporcionará un usuario y una contraseña al inicio de cada semestre. Una vez que ha ingresado correctamente dicho usuario y su respectiva contraseña, la Fig. 4.13 nos mostrará hasta arriba tres botones los cuales son **"Publicar"**, **"Contraseña"** y **"Salir"**. Dependiendo del perfil configurado de cada profesor, al momento de acceder al sistema le aparecerán cargados exclusivamente los grupos de alumnos del LCD que tiene asignados.



Fig. 4.13 – Pantalla inicial para los profesores (desde Internet)

4.2.2 ® PUBLICANDO AVISOS

Al pulsar sobre la opción **"Publicar"** accederemos a la pantalla mostrada en la Fig. 4.14 que se muestra a continuación.

En dicha pantalla aparecerán las opciones de **"Publicar"** y **"Ver",** estas opciones las podrá encontrar en la parte derecha de la pantalla; para identificarlas la opción **"Publicar"** tiene abajo íconos de carpetas y la opción **"Ver"** tiene flechas apuntando hacia la derecha.

En esta pantalla se presenta la posibilidad de dejar avisos y tareas exclusivamente a los grupos asignados al profesor que ingresó con su usuario y contraseña.



Fig. 4.14 – Pantalla para publicar tareas y avisos (desde Internet).

Por ejemplo, si el profesor desea dejar un examen de casa al grupo de alumnos identificado como **"gpo07",** hay que presionar el icono en forma de carpeta a la derecha del identificar **"gpo07"** y debajo de **"Publicar".**

Al presionar dicho ícono aparecerá la pantalla mostrada en la Fig. 4.15, la cual guiará al profesor para adjuntar archivos o información específica (en este caso un examen de casa) a su grupo donde cada uno de los miembros podrá consultar a través de sus cuentas de usuario dicha información.

En esta pantalla el profesor podrá, asignar un título al aviso, agregar un breve texto describiendo lo que desea que los alumnos hagan y adjuntar un archivo; para realizar lo último se presenta un buscador típico para guiar al profesor hasta encontrar la ubicación del archivo que desea adjuntar.

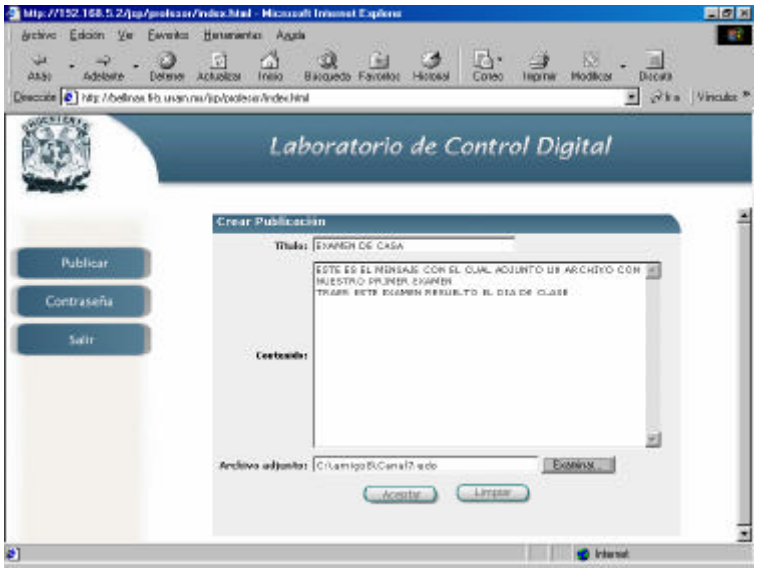


Fig. 4.15 – Pantalla que detalla la parte para publicar tareas y avisos (desde Internet).

Es importante mencionar que en esta pantalla es mandatario llenar los campos de **"Título"** y de **"Contenido"** de lo contrario, el sistema mostrará un mensaje avisando al profesor la falta de texto en alguno de los dos campos. La opción de "Archivo adjunto" no es mandataria, por lo tanto el profesor podrá presionar aceptar sin la necesidad forzosa de adjuntar un archivo. Después de agregar correctamente la información aparecerá la pantalla mostrada en la Fig. 4.16.



Fig. 4.16 – Pantalla que muestra la aceptación de un aviso publicado exitosamente

Al pulsar el botón **"Aceptar"** regresaremos a la pantalla mostrada en la Fig. 4.14.

Estando de nuevo en la pantalla de la Fig. 4.14, ahora seleccionaremos la opción **"Ver"** y obtendremos en respuesta la imagen de la Fig. 4.17.



Fig. 4.17 – Pantalla que muestra respuesta al pulsar la opción **"Ver".**

En esta nueva pantalla (Fig. 4.17) se presentan dos opciones para cada uno de los avisos o tareas previamente publicadas al **"gpo07"**, **"Visualizar"** y **"Eliminar".** En base al criterio utilizado anteriormente, aquí es claro si optamos por pulsar **"Visualizar"** podremos ver dicha publicación previa así como su archivo adjunto (en caso de que lo tenga) Y si elegimos la opción **"Eliminar",** lo que sucederá es que se eliminará la publicación, no sin antes preguntarle al profesor si está seguro de que desea borrar dicha publicación previa.

En la Fig.4.18 se muestra la pantalla que le pregunta al profesor, si está seguro de que desea eliminar la publicación. Es importante mencionar que una vez que se presiona el botón **"Aceptar"**, no hay marcha atrás y la publicación y su respectivo archivo no podrán ser recuperados.

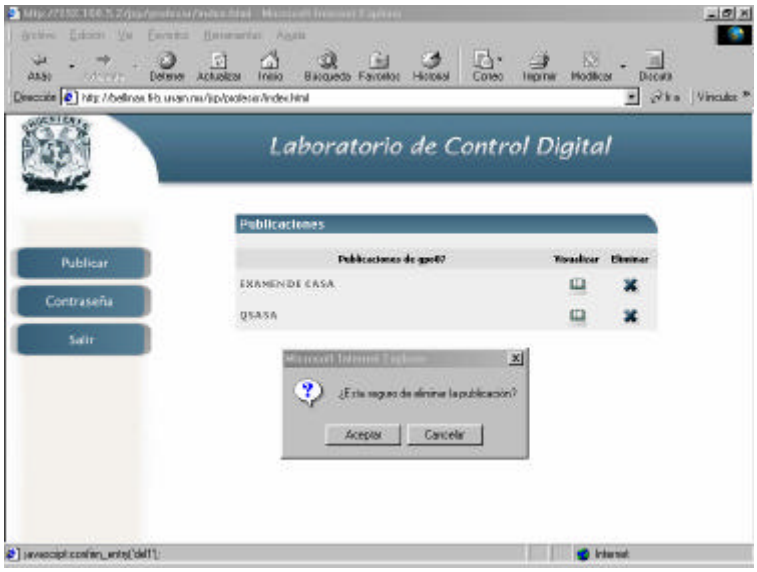


Fig. 4.18 – Eliminando una publicación.

Por ultimo al presionar **"Aceptar"** en la pantalla de la Fig 4.18 la respuesta del sistema es mostrar nuevamente la pantalla de la Fig. 4.14

4.2.3 ® CAMBIANDO LA CONTRASEÑA

De nuevo en la Fig. 4.14, pulsaremos la opción de **"Contraseña"**, al hacerlo el sistema mostrará la pantalla mostrada abajo en la Fig. 4.19.



Fig. 4.19 – Cambiando la contraseña de acceso de un profesor.

Los tres campos de texto son mandatarios, por lo cual, deberán ser llenados como se indica; el sistema al revisar los campos y validarlos correctamente ejecutará la acción de cambio de contraseña. Se mostrará un aviso informando el cambio de contraseña, al presionar **"Aceptar"** el sistema nos volverá a mostrar la pantalla de la Fig. 4.14.

4.2.4 ® SALIENDO DEL SISTEMA

Estando nuevamente en la pantalla de la Fig. 4.14 utilizaremos el último botón **"Salir"** para terminar nuestra sesión. Al presionar esta opción el sistema mostrará la pantalla mostrada en la Fig. 4.20.



Fig. 4.20 – Saliendo del sistema.

Y el sistema regresará a la pantalla inicial mostrada en la Fig. 41, la cual nos solicita el nombre de usuario y la contraseña.

Para cualquier duda o aclaración sobre este subtema favor de presentarse en el cubículo de servicios de mantenimiento de los laboratorios de control con Pedro Gabriel Saucedo Uribe o comuníquese directamente o a los teléfonos (55) 5622 3132 y (55) 5622 3137.

4.3 ® MANUAL DE OPERACIÓN - ADMINISTRADOR

Este manual tiene por objeto que la(s) persona(s) asignada(s) a configurar el sistema en cada inicio de semestre, conozca(n) las funciones y capacidades de administración de sistema automatizado que controla el acceso de los alumnos y los profesores al dominio lógico del LCD, descrito en esta tesis.

La administración de este sistema usa un nombre de usuario con un perfil bien definido, que solamente tendrá el administrador.

La página de acceso desde Internet para el administrador es la misma que para los alumnos y profesores: http://bellman.fi-b.unam.mx . Dicha pantalla se muestra en la Fig 4.21.



Fig. 4.21 – Página general de acceso al sistema del LCD.

Después de ingresar el nombre de usuario y su contraseña que en este caso son específicas para el administrador del sistema se mostrará la pantalla mostrada en la Fig. 4.22

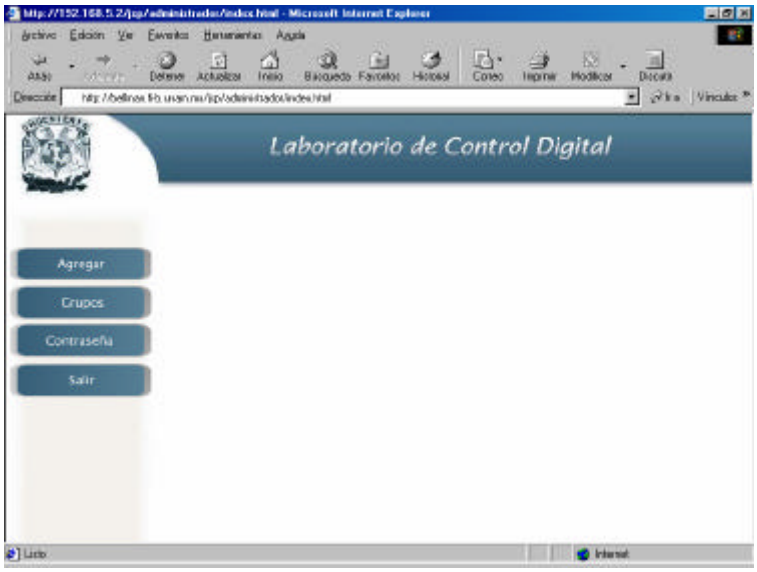


Fig. 4.22 – Página inicial para el administrador del sistema (desde Internet).

4.3.1 ® AGREGANDO USUARIOS (ALUMNOS, PROFESORES Y GRUPOS)

En la pantalla que se muestra en Fig. 4.22 se observan cuatro botones en la parte izquierda, los cuales tienen funciones específicas. Revisaremos y explicaremos las funciones de cada uno de ellos

El primero de estos es el de **"Agregar"**, por medio de esta opción el administrador podrá dar de alta profesores, alumnos, crear un grupo de alumnos o vaciar las cuentas que hayan sido creadas con anterioridad.

Al presionar el botón **"Agregar"** se mostrará la pantalla de la Fig. 4.23. Dicha pantalla nuestra 4 opciones adicionales dentro de la administración de usuarios, las opciones son: **"Crear Profesor"**, **"Crear Alumnos"**, **"Crear Grupos"** y **"Vaciar Cuentas"**, más adelante detallaremos cada una de ellas.



Fig. 4.23 – Pantalla para dar de alta nuevos profesores, alumnos o grupos.

4.3.1.1 ® *CREAR PROFESOR*

SI deseamos dar de alta un profesor al inicio de un semestre, se deberá pulsar la opción de **"Crear Profesor"**, el sistema mostrará la pantalla de la Fig. 4.24. En la cual aparece simplemente aparece un cuadro de texto en el cual se deberá ingresar el nombre de usuario que se asignará al profesor que se está dando de alta, posteriormente se presionará **"Aceptar"**.

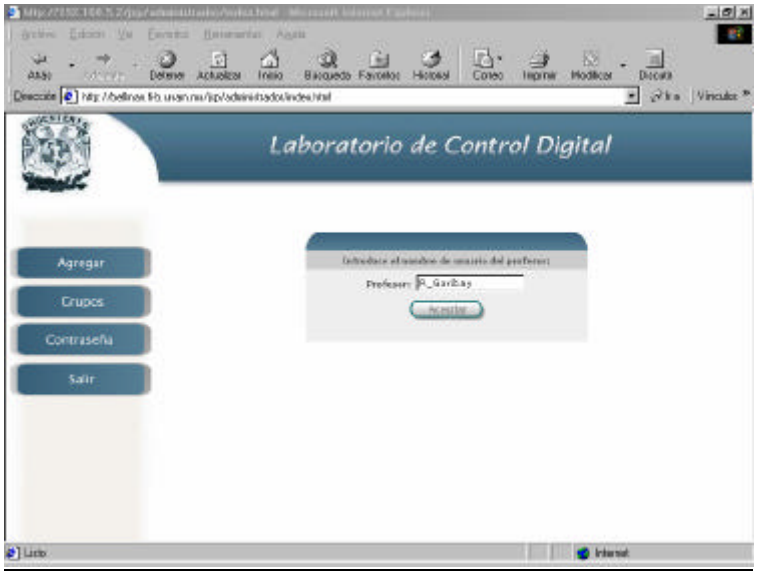


Fig. 4.24 – Pantalla para dar de alta un nuevo profesor en el sistema al inicio de un semestre.

Al presionar la opción **"Aceptar"** dentro de la pantalla de la Fig. 4.24, el sistema mostrará un mensaje avisando que el alta fue realizada exitosamente (Fig. 4.25).



Fig. 4.24 – Pantalla confirmando el alta exitosa de un nuevo profesor.

4.3.1.2 ® *CREAR ALUMNO*

Si regresamos a la pantalla mostrada en la Fig 4.23 podremos ver que la segunda opción disponible en la parte central de la pantalla es la de **"Crear Alumnos"**, con esta opción podremos dar de alta los alumnos que tomarán clases dentro del nuevo curso del LCD, además de que podremos asignar dichos alumnos nuevos a un grupo determinado.

Por cuestiones de logística, es recomendable que no se personalicen los accesos a un alumno en particular ya que regularmente en las clases del LCD se forman equipos de alumnos dentro de cada grupo, recordemos que hay sólo 7 computadoras y en ocasiones hay más de 10 alumnos por clase.

Si presionamos el botón **"Crear Alumnos"** se desplegará la pantalla de la Fig. 4.25.



Fig. 4.25 – Dando de alta un nuevo alumno y asignándole un grupo.

Los grupos deberán estar ya creados, es decir, el administrador debió haberlos creado anticipadamente por medio de la opción **"Crear Grupo"**; en la Fig. 4.25 se observa un cuadro de texto libre y un cuadro tipo combo que despliega los grupos disponibles. Cuando se halla especificado el nombre del alumno y el grupo al que pertenecerá, se mostrará una pantalla confirmando el alta del nuevo alumno, en forma similar a cuando creamos un profesor.

4.3.1.3 ® *CREAR UN GRUPO*

Ahora revisaremos la tercera opción de la Fig. 4.23, **"Crear Grupo"**, la siguiente pantalla en la Fig. 4.26 será la que utilizaremos para agregar un grupo.

En la Fig. 4.26, aparece un cuadro de testo libre, ahí capturaremos el nombre del grupo nuevo que deseamos dar de alta. Una vez colocado el nombre en el cuadro de texto, presionamos la opción **"Aceptar"**, el sistema nos presentará una pantalla indicando el alta exitosa del nuevo grupo.

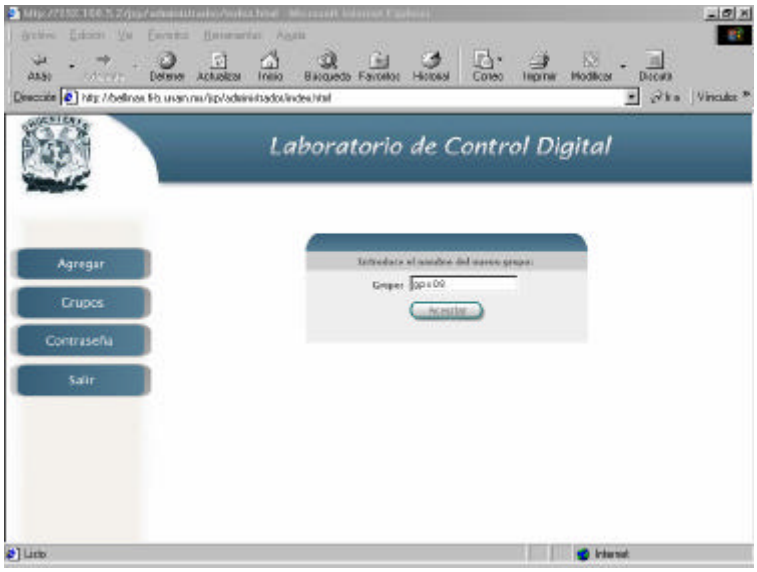


Fig. 4.26 – Dando de alta un nuevo grupo.

4.3.1.4 ® VACIAR CUENTAS

Y por ultimo el botón de vaciar cuentas el cual eliminara todos los archivos de las cuentas de usuarios del sistema de administración del Laboratorio de Control Digital, está función generalmente será utilizada al final de cada semestre para liberar la memoria del servidor, preparándolo para un nuevo semestre.

Es muy importante que el administrador del sistema, tenga cuidado al final de cada semestre con el fin de que toda la información referente al periodo anterior sea eliminada, ya que podrían presentarse confusiones.

4.3.2 ® ADMINISTRANDO LOS GRUPOS DEL LCD

En la misma pantalla que se muestra en Fig. 4.22, se observan cuatro botones en la parte izquierda, continuaremos revisando las funcionalidades de cada uno de ellos, el segundo es el que aparece como **"Grupos"**, por medio de esta opción el administrador podrá controlar los profesores y alumnos que son asignados a cada grupo, también podrá eliminar grupos, visualizar la alumnos asignados a cada grupo y además, cambiar contraseñas de usuarios (profesores y alumnos). Al presionar la opción "Grupos" veremos la pantalla que se muestra en la Fig. 4.28.



Fig. 4.28 – Administrando los grupos del LCD.

4.3.2.1 ® PROFESORES

En esta sección podremos cambiar las contraseñas a profesores. Primeramente veremos que pasa al seleccionar la opción de **"Mostrar Profesor"** en la Fig. 4.28, desde la cual se generará otra pantalla mostrada en la Fig. 4.29 con tres opciones.



Fig. 4.29 – Administrando la cuenta de un profesor.

En la Fig. 4.29, el primer campo **"Cambiar Contraseña"** se utilizará, precisamente para eso, cambiar la contraseña de acceso del profesor, la siguiente, **"Liberar"**, retira la asociación del profesor con el grupo seleccionado y la tercera **"Eliminar"**, elimina de la lista al profesor desplegado.

El siguiente campo de la Fig. 4.28 a revisar, será el de **"Asignar Profesor"**, con el cual podremos asignar un profesor a un grupo, para este ejemplo, usaremos el gpo01 y la pantalla mostrada será la de la Fig.4.30



Fig. 4.30– Asignando un profesor a un grupo.

Aquí debe quedar claro que el profesor a asignar ya debe estar en la lista de profesores, que se muestra en el cuadro de texto tipo combo como se ilustra.

4.3.2.2 ® ALUMNOS

Una de las partes más importantes del sistema de administración del LCD se encuentra en la opción **"Mostrar Alumnos"** de la Fig. 4.28, desde la cual aparte de mostrarse los alumnos pertenecientes al grupo se pueden cambiar las contraseñas de los mismos, ya sea por actualización o por olvido del usuario de su contraseña y la pantalla quedará como en la Fig. 4.31.



Fig. 4.31– Administrando las cuentas de los alumnos.

4.3.3 ® CAMBIANDO LA CONTRASEÑA DEL ADMINISTRADOR

De las opciones iniciales de los cuatro botones mostradas en la Fig. 4.22 ahora si pulsamos el botón de contraseña podremos cambiar la contraseña del administrador cuyo nombre de usuario es **admin**, el cual no podrá ser cambiado. Para poder cumplir la función de cambiar la contraseña del administrador se tendrán que llenar los campos de texto mostrados en la pantalla de la Fig. 4.32. Una vez que los campos sean llenados de forma correcta, la acción de cambio de contraseña será llevada a cabo satisfactoriamente.



Fig. 4.32– Cambiando la contraseña de administrador.

La opción final a ejecutar será la del botón de salir con lo cual se cerrará la sesión de administrador y aparecerá un mensaje de acuerdo con esta acción como se ve a continuación:



Y al pulsar el botón de aceptar se regresará a la pantalla inicial pidiendo el nombre de usuario y la contraseña, para así iniciar una nueva sesión.

5. CONCLUSIONES

Los objetivos de la tesis fueron cumplidos, al ser instalada una red con cableado estructurado y servicios de Internet en el laboratorio de control digital. Interconectando dos sistemas operativos con plataforma totalmente diferente en un solo sistema unificado.

Pero independientemente de estos valores provistos, estamos convencidos que el principal logro de la implementación de este sistema fue el conseguir que las clases prácticas de la asignatutra de Control Digital se pudiesen impartir en forma constante y con la menor cantidad de problemas técnicos.

Además de conseguir la impartición constante de clases prácticas a lo largo ya de varios semestres, se benefició a los alumnos y profesores con un sistema de administración que permite reutilizar la infraestructura de una forma sencilla y práctica, con una interfaz 100 % gráfica, con esto se evitó tener que utilizar la interfaz de comandos del sistema abierto. Estos procedimientos se explican a detalle en el capítulo 4 de este documento.

La privacidad y confiabilidad con la que cuentan ahora los alumnos al desarrollar sus prácticas, no tiene precedente en el Laboratorio de Control Digital, ya que actualmente cada alumno puede acceder al sistema durante la clase ó desde Internet para elaborar ó descargar un programa y guardarlo, sin la menor preocupación de alguien no autorizado pueda borrarlo o modificarlo.

Otro de los puntos que nosotros consideramos que tenemos a favor es, que gran parte del proyecto se llevó a la práctica con el mismo equipo de cómputo y comunicaciones con el que el Laboratorio de Control digital contaba originalmente. Dada la eficiencia de la solución presentada se pudieron utilizar computadoras Pentium I (obsoletas para los programas actuales), lo cual nos permitió dotar al Laboratorio de una solución práctica a muy bajo costo.

En lo único que nosotros invertimos dinero fue en la parte de la red (cableado), pero estamos convencidos que el dejar una instalación física (de capa 1) sólida y profesional permitirá que, sí en un futuro desean reemplazar los equipos por otros más modernos y robustos se realice sobre la misma plataforma de red. Además, el centro de distribución de cableado está lo

suficientemente holgado con el fin de permitir crecimientos organizados y controlados, basados en las normas que fueron utilizadas en este proyecto.

Evidentemente, todo lo invertido durante esta implementación lo hicimos pensando en el beneficio de nuestra Facultad, por lo tanto, a partir de que el sistema de cableado y cómputo (desarrollo de software) fue instalado y configurado pasa a ser propiedad del Departamento de Control.

Una de las cosas que aprendimos durante nuestra época de estudiantes en la Facultad de Ingeniería, fue siempre tratar de maximizar los recursos con el fin de obtener más con menos, y creemos que es este proyecto lo hemos demostrado, al preferir utilizar como base un sistema operativo de fuente abierta, el cual se requiere licenciamiento, nos permitió dejar los costos por concepto de software en cero. Evidentemente, el tomar esta decisión implicó hacia nosotros, trabajo adicional, por lo cual tuvimos que ocupar nuestro propio ingenio y recursos intelectuales para poder solventar los problemas prácticos que se nos presentaron.

Por otro lado el utilizar este tipo de software (fuente abierta) libera a la Facultad de cualquier responsabilidad con empresas de desarrollo de software por la utilización de programas y paquetes sin autorización.

Consideramos que la elección del sistema operativo de las computadoras (con licenciamiento) cliente así como el del servidor (fuente abierta) y el Firewall (fuente abierta) fue adecuado, esto debido a la situación actual del Laboratorio de Control Digital, la cual no le permite invertir fuertes sumas de dinero en la compra de software de última generación.

En el caso del sistema operativo de las máquinas cliente que fue el de Windows NT 4.0 Workstation se considera esto debido a que la protección de los programas de aplicación como Matlab, Simnon y CC (software que la Facultad ha ocupado desde hace muchos años) principalmente instalados en estas máquinas no fueron modificados por los usuarios en cuanto a su contenido estando dichas aplicaciones disponibles y en correcto funcionamiento para su uso todo el tiempo. La elección de este sistema operativo permitió generar un dominio con un servidor central con sistema operativo Linux el cual fue muy estable permitiendo las sesiones en red constantemente.

El establecimiento de cuotas en el servidor de dominio evitó la saturación del disco duro por parte de los usuarios. La estabilidad de Linux fue constante todo el tiempo y no propicio retrasos en la impartición de las clases de dicho laboratorio. La elección de Linux fue adecuada ya que los respaldos realizados a dicho servidor donde se encuentran las cuentas de usuarios en la partición homes no fueron afectados aun y cuando fue necesario cambiar de servidor por una falla técnica del procesador de dicha máquina. Esta máquina a su vez permite el redireccionamiento del sistema de administración vía Internet alojado en el servidor central a través del paquete Tomcat usando la IP real de esta maquina. El Firewall fue establecido con la utilización de ipchains lográndose la cadena "forward" necesaria para realizar dicho direccionamiento entre la tarjeta interna de la maquina con el Firewall y la máquina con la interfase samba y el programa Tomcat.

Se eligió una red de clase tipo C por considerarse el número de máquinas conectadas reducido siendo en un principio de siete maquinas cliente a parte de la maquina servidor y la maquina con el Firewall.

Los remates del cableado estructurado soportan bien el uso y no ha habido problemas en ninguno de los nodos establecidos y no han tenido que ser reinstalados. El rack y su panel de parcheo probaron ser confiables desde todos los puntos de vista soportando el uso constante así como la instalación de dos concentradores siendo uno dedicado exclusivamente para el segmento de direcciones reales del departamento de control y otro dedicado a las direcciones virtuales del laboratorio de control digital y del área de mantenimiento.

En los últimos años el Internet ha transformado la forma de comunicarse de las personas. Actualmente es más sencillo realizar actividades que antes requerían de tiempo ya que era necesario trasladarse de un lugar a otro para realizar trámites, compras, transacciones bancarias, etc. Las empresas e instituciones incorporan nuevas tecnologías de información en los servicios que ofrecen para realizar transacciones en línea a un menor tiempo y costo para sus usuarios, optimizando mejor sus recursos.

La educación no ha escapado a la influencia de Internet. No sólo se han creado nuevos métodos de enseñanza (como la educación a distancia en tiempo real) sino que han cambiado los métodos de investigación y de recolección de datos.

Actualmente la mayoría de los estudiantes y los profesores buscan información en Internet antes que ir a una biblioteca o a una librería. Entre muchas de las ventajas de utilizar Internet para estos fines están: la rapidez con que se puede encontrar la información, la gran cantidad de datos que se pueden conseguir acerca de un mismo tema de interés, el bajo costo que significa el no tener que comprar determinado libro, etc.

La tendencia de uso del Internet nos permitió colocar al Laboratorio de Control Digital en un marco de tecnología de vanguardia, en el que las empresas e instituciones que no brindan este tipo de servicios quedan en desventaja. Los profesores y alumnos cuentan con una herramienta que facilita la comunicación y permite el acceso a información que anteriormente tenían que manejar por otros medios.

El sistema de administración con acceso desde Internet, conjugó las características de la programación tipo shell script del ambiente Linux por tratarse de la partición homes del servidor central donde están las cuentas de usuarios establecidos en el dominio la cual se ejecuta en Linux. Se hizo referencia a estos programas a través de la inserción de una clase especial localizada en él paquete SAMBA ubicado dentro de la aplicación de Tomcat creado con el fin de enlazar el ambiente de los programas realizados en shell script de Linux y la programación en jsp del lenguaje de programación Java. Se prefirió usar JSP por la potencia del lenguaje JAVA a diferencia de los programas ASP que usan código vbscript o javascript y son dependientes hasta hace poco de plataformas microsoft. Otra ventaja de los JSP es que son independientes de la plataforma y en este caso instalamos servidores Linux y máquinas cliente NT por lo que decidimos usar JSP.
Los CGI usan lenguaje otros lenguajes (PERL o C) que son más complicados y es difícil rastrear los errores cuando se habla de programas grandes lo cual no pasa en el lenguaje JAVA por el manejo de excepciones.

El uso de tecnologías del lado del servidor como los ASP, JSP o PHP permiten emplear las características del servidor tanto de hardware como software instalado en el servidor.

Tomcat demostró ser una buena opción estando siempre disponible la página de acceso al sistema de administración vía Internet.

A través de una serie de consideraciones y pasos metódicos llegamos al final de la presente tesis, en la que el contenido puede servir para diversos propósitos relacionados con la arquitectura y configuración de una red y el desarrollo de un sistema WEB

Nos gustaría hacer un reconocimiento público por este medio a nuestro asesor de Tésis, el Ing. Rafael Iriarte V. Balderrama, ya que gracias a su paciencia y comprensión, nos permitió por fin concluir este proyecto después de varios años de haber trabajado en él.

APÉNDICE

A) EL FIREWALL

A continuación se muestra el "firewall" instalado en bellman y que controla el tráfico en la red virtual a través de la dirección ip real 132.248.59.122. Cabe mencionar que enteramente, el Firewall fue programado en Linux, con esto se disminuye considerablemente el costo ya que como sabemos un equipo (hardware) que haga la misma función implicaría invertir varios miles de pesos.

```
#!/bin/sh
echo "Starting firewalling... "
# algunas definiciones para un fácil mantenimiento:
# ------------------------------------------------------------------
#.
EXTERNAL_INTERFACE="eth1"          # Interfase conectada al internet
LOOPBACK_INTERFACE="lo"            # Convención local de nombre
LAN_INTERFACE_1="eth2"             # Interfase de la red interna LAN
IPADDR="132.248.59.122"            # Dirección IP real
LAN_1="192.168.5.0/24"             # Rango a usar de la red interna
LAN_IPADDR_1="192.168.5.1"         # Dirección IP de la interfase interna
ANYWHERE="any/0"                   # Compatibilidad con cualquier dirección IP
LOOPBACK="127.0.0.0/8"             # Rango reservado  de dirección loopback
CLASS_A="10.0.0.0/8"               # Redes privadas clase A
CLASS_B="172.16.0.0/12"            # Redes privadas clase B
CLASS_C="192.168.0.0/16"           # Redes privadas clase C
CLASS_D_MULTICAST="224.0.0.0/4"    # Direcciones multicast de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5" # Direcciones reservadas de clase E
BROADCAST_SRC="0.0.0.0"            # Direcciones fuente de broadcast
BROADCAST_DEST="255.255.255.255"   # Direcciones destino de broadcast
PRIVPORTS="0:1023"                 # Rango de puertos privilegiados bien conocidos
UNPRIVPORTS="1024:65535"           # Rango de puertos sin privilegios
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"
# SSH empieza en 1023 y disminuye hasta 513 para
# cada conexión entrante simultanea.
SSH_PORTS="1020:1023"              # Conexiones simultaneas
```

```
# ------------------------------------------------------------------
   # Habilitar TCP SYN protección contra Cookies
   echo 1 > /proc/sys/net/ipv4/tcp_syncookies
   # Habilitar siempre protección contra defragging
   echo 1 > /proc/sys/net/ipv4/ip_always_defrag
   # Habilitar protección broadcast echo
   echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
   # Habilitar protección por mensaje de mal error
   echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
   # Habilitar protección IP contra spoofing
   # Hhabilitar verificación de dirección de origen
   for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
   done
   # Deshabilitar aceptación directa de ICMP
   for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
      echo 0 > $f
   done
   for f in /proc/sys/net/ipv4/conf/*/send_redirects; do
      echo 0 > $f
   done
   # Deshabilitar paquetes ruteados de origen
   for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
      echo 0 > $f
   done
   # Registrar paquetes Spoofed, ruteados de origen y redireccionados
   for f in /proc/sys/net/ipv4/conf/*/log_martians; do
      echo 1 > $f
   done
# Enmascaradamiento de servicios
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
/sbin/modprobe ip_masq_vdolive
/sbin/modprobe ip_masq_cuseeme
/sbin/modprobe ip_masq_quake
# ------------------------------------------------------------------
# Borrar cualquier regla existente de todas las cadenas
ipchains -F
# Colocar la política de default a negar
ipchains -P input   DENY
```

```
ipchains -P output  ACCEPT
# Establecer el tiempo máximo a 10 horas para conexiones TCP
ipchains -M -S 36000 0 0
# Rechazar paquetes fragmentados
ipchains -A input -f -i $EXTERNAL_INTERFACE -j DENY
# ------------------------------------------------------------------
# LOOPBACK
# Tráfico ilimitado en la interfase de loopback
ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT
# ------------------------------------------------------------------
# Tráfico ilimitado dentro de la red local
# ------------------------------------------------------------------
# Enmascaradamiento del trafico interno
# Todo el trafico interno es enmascarado externamente
ipchains -A forward -i $EXTERNAL_INTERFACE -s $LAN_1 -j MASQ
ipchains -A forward -i $EXTERNAL_INTERFACE -s 192.168.5.2 -j MASQ
ipchains -A forward -i $EXTERNAL_INTERFACE -s 192.168.5.3 -j MASQ
ipchains -A forward -i $EXTERNAL_INTERFACE -s 192.168.5.4 -j MASQ
ipchains -A forward -i $EXTERNAL_INTERFACE -s 192.168.5.5 -j MASQ
# ------------------------------------------------------------------
# SPOOFING Y MALAS DIRECCIONES
# Rechazar paquetes de tipo spoofed
# Ignorar direcciones de origen ilegales
# Autoprotección para no mandar a malas direcciones
# Rechazar paquetes de tipo spoofed a partir de la interfase externa
ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -j DENY -l
# Rechazar paquetes provenientes hacia o desde una red privada Clase A
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY -l
# Rechazar paquetes provenientes hacia o desde una red privada Clase B
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY -l
# Rechazar paquetes provenientes hacia o desde una red privada Clase C
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY -l
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY -l
ipchains -A input  -i $LAN_INTERFACE_1 -s $CLASS_C -j ACCEPT

# Rechazar paquetes hacia la interfase de loopback
ipchains -A input  -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY -l
# Bloquear broadcast directos:
# Direcciones base de redes
# Direcciones de broadcast de redes
SUBNET_BROADCAST="you.you.you.255"
SUBNET_BASE="you.you.you.0"
ipchains -A input -i $EXTERNAL_INTERFACE -d $SUBNET_BASE   -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -d $SUBNET_BROADCAST -j DENY -l

# Rechazar paquetes malformados desde el broadcast
ipchains -A input  -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC  -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC  -j DENY -l

# Rechazar direcciones multicast de clase D
# Multicast solo es ilegal como una dirección origen.
# Multicast usa UDP

# Rechazar direcciones IP reservadas de clase E
# Bloqueadas las entrantes abajo
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_E_RESERVED_NET \
      -j REJECT
# ------------------------------------------------------------------
# Aceptamos cualquier paquete que no sea de entrada
# ------------------------------------------------------------------
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
-s $ANYWHERE -d $IPADDR -j ACCEPT
# ------------------------------------------------------------------
# NOTA:
#     Los nombres simbólicos usados en /etc/services para los números de puerto
#     varían por proveedor.
#
# ------------------------------------------------------------------
# Servicios requeridos
# Modos  de cliente DNS  (53)
```

```
# ---------------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $ANYWHERE 53 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT
# ------------------------------------------------------------------------
# Servicios TCP en puertos seleccionados
# Recibir Correo como un servidor local SMTP (25)
# ----------------------------------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 25 -j ACCEPT

# TELNET (23) – Permite conexiones entrantes a su servidor local
# ---------------------------------------------------------------
# SSH (22) – Permite acceso remoto a su servidor local
# ---------------------------------------------------------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 22 -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $SSH_PORTS \
        -d $IPADDR 22 -j ACCEPT
# HTTP (80) – permite acceso remoto a su servidor Web
# ---------------------------------------------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 80 -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 8080 -j ACCEPT
# ------------------------------------------------------------------------
# UDP aceptados solo en  puertos específicos
# TRACEROUTE
# traceroute usualmente usa -S 32769:65535 -D 33434:33523
# ------------------------------------------------------
# ------------------------------------------------------------------------
# ICMP
#
#   Peticiones entrantes y salientes  como flujo de control
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 4 -d $IPADDR -j ACCEPT
```

```
# Problemas de parámetros
# Mensajes de error salientes y entrantes
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp \
      -s $ANYWHERE 12 -d $IPADDR -j ACCEPT
# Destino inalcanzable, servicio no disponible
#  Negociación de entradas y salidas, servicio o
#  Destino no disponible, traceroute final
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp \
      -s $ANYWHERE 3 -d $IPADDR -j ACCEPT
# Tiempo excedido
#  Condiciones de entrada y salida de tiempo máximo
#   También respuesta TTL a traceroutes
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp \
      -s $ANYWHERE 11 -d $IPADDR -j ACCEPT
# permitir pings de entrada
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp \
          -s $ANYWHERE -d $IPADDR -j ACCEPT
# Permitir ping de salida a cualquier destino
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp \
      -s $ANYWHERE 0 -d $IPADDR -j ACCEPT
echo "done"
exit 0
```

GLOSARIO DE TÉRMINOS

ANSI (AMERICAN NATIONAL STANDARDS INSTITUTE).

El ANSI es una organización no lucrativa, privada que administra y coordina el sistema voluntario del gravamen de la estandarización y de la conformidad de Estados Unidos.

API(APLICATION PROGRAM INTERFACE)

Realizan determinadas funciones estas interfaces de programación de aplicaciones como el uso de dispositivos, de multimedia, de pantallas y gráficos, etc. Generalmente se almacenan en ficheros dll o exe

AT&T

Compañía de soluciones en conectividad de redes. Provee comunicaciones en voz y datos a través del mundo. Atiende telefonía, larga distancia, Internet y hosting principalmente.

Caldera

Es una de las distribuciones de Linux que hace compatible el sistema de openlinux con openunix. Las distribuciones varian en cuanto a herramientas , utilerias y aplicaciones.

CAT 5e (UTP)

La opción más asentada en el mercado es la de Categoría 5. No en vano, es la más veterana ya que fue completamente definida a mediados de la presente década. Sus especificaciones hacen referencia a un cable de cuatro pares trenzados recubiertos por un material aislante y con propiedades ignífugas.

Se trata de un cable no apantallado (UTP, *Unshielded Twisted Pair*), una característica que juega en contra de una potencial evolución de la red ya que presenta más susceptibilidad frente a fenómenos electromagnéticos (interferencias electromagnéticas, diafonías, atenuación de lseñal). Fenómenos que se presentan con mayor facilidad cuanto mayor es la frecuencia, es decir, cuanto mayor es el ancho de banda.

El ancho de banda previsto para la Categoría 5 es de 100 MHz, lo que permite acomodar redes del tipo Fast Ethernet, a una velocidad de transmisión de 100 Mbps, así como posteriormente las ATM-LAN (redes locales basadas en el modo de transferencia asíncrono) a una velocidad de 155 Mbps. La correspondencia normativa de la Categoría 5 (que, de hecho, hace referencia al

hardware de conexión y no al cable, que es de Clase D), es la ISO 11801 y, en Europa, EN 50173. Con el fin de introducir mejoras a partir de estas premisas surgió la denominada Categoría 5 Mejorada (*Enhanced*), muy similar a la anterior si bien en este caso el cable UTP es de mayor calidad. Ello permite mejorar la respuesta frente a fenómenos de diafonía y ruido, así como incrementar la velocidad de transmisión. De este modo es posible implementar redes del tipo Gigabit Ethernet, que alcanzan ya el orden de los gigabits por segundo.

Insistiendo en el aspecto de la velocidad, la Categoría 6 (que utiliza cable de Clase E) se inscribe dentro de la necesidad de disponer del orden de los gigabits por segundo, ya que no existía la seguridad de que la Categoría 5 Mejorada pudiera asumir este nivel de prestaciones.

El ancho de banda contemplado llega ya hasta los 250 MHz, si bien por el momento tan sólo se ha definido completamente hasta los 200 MHz. El cable es un UTP, y su instalación resulta sensiblemente más cara que para la Categoría 5. Actualmente se encuentra en fase de desarrollo en lo que a la descripción del estándar se refiere.

COAXIAL

Medio destinado a la conectividad de redes principalmente.Esta compuesto de un núcleo de cobre sólido, o de acero con capa de cobre, o bien de una serie de fibras de alambre de cobre entrelazadas dependiendo del fabricante.

Una capa de aislante que recubre el núcleo o conductor, generalmente de material de polivinilo, este aislante tiene la función de guardar una distancia uniforme del conductor con el exterior. Una capa de blindaje metálico, generalmente cobre o aleación de aluminio entretejido (a veces solo consta de un papel metálico) cuya función es la de mantenerse lo mas apretado posible para eliminar las interferencias, además de que evita de que el eje común se rompa o se tuerza demasiado, ya que si el eje común no se mantiene en buenas condiciones, trae como consecuencia que la señal se va perdiendo, y esto afectaría la calidad de la señal. Por último, tiene una capa final de recubrimiento, de color negro en el caso del cable coaxial delgado o amarillo en el caso del cable coaxial grueso, este recubrimiento normalmente suele ser de vinilo, xelón ó polietileno uniforme para mantener la calidad de las señales.

CONECTOR RJ-45

Los conectores del NIC RJ45 de un sistema están diseñados para conectar un cable UTP (Unshielded Twisted Pair [par Trenzado sin Blindaje]) para red Ethernet equipado con enchufes convencionales compatibles con el estándar RJ45. Se coloca, presionando un extremo del cable UTP dentro del conector NIC hasta que el enchufe se asiente en su lugar.

DOMINIO

Es un entorno de red protegido mediante el uso de contraseñas para usuarios los cuales comparten recursos tales como impresoras o archivos. Generalmente se logra la inserción de las maquinas en un dominio especifico usando los sistemas operativos de windows nt, windows 2000 y linux.

ETHERNET (10 BASE-T)

Tipo de red de área local (LAN) desarrollada por Xerox, Digital e Intel. Es el método de acceso LAN que más se utiliza (seguido por Token Ring). Ethernet es una LAN de medios compartidos. Todos los mensajes se diseminan a todos los nodos en el segmento de red. Ethernet conecta hasta 1,024 nodos a 10 Mbits por segundo sobre un par trenzado, un cable coaxial y una fibra óptica. Los tres tipos principales son: (1) 10Base5 Standard Ethernet, que utiliza un cable coaxial grueso en una topología de bus entre nodos con una longitud de segmento máxima de hasta 1,640 pies, (2) 10Base2 Thin Ethernet, también llamado ThinNet y CheaperNet, que utiliza un cable coaxial más delgado de hasta 607 pies por segmento y (3) 10BaseT, que utiliza pares trenzados conectados a una configuración de estrella a través de un centro con una longitud de segmento máxima de 328 pies. Ethernets más rápidas están surgiendo: una Ethernet conmutada da a cada usuario un canal dedicado de 10 Mbps. Una Ethernet rápida corre a 100 Mbps compartidos

FAST ETHERNET (100 BASE-T)

La conectividad es mucho mas rápida aplicándose a 100Mbps. Fast Ethernet o 100Base-T tiene el mismo fundamento que Ethernet o 10Base-T ofreciendo una velocidad 10 veces superior de 100 Mbps en redes con ancho de banda compartido empleando el método de acceso al medio CSMA/CD. Siendo Respaldada por un grupo de empresas que formaron una alianza para desarrollar un estandar respaldado por la IEEE llamado 802.3u para Fast Ethernet.

FIBRA ÓPTICA

Los circuitos de fibra óptica son filamentos de vidrio (compuestos de cristales naturales) o plástico (cristales artificiales), del espesor de un pelo (entre 10 y 300 micrones). Llevan mensajes en forma de haces de luz que realmente pasan a través de ellos de un extremo a otro, donde quiera que el filamento vaya (incluyendo curvas y esquinas) sin interrupción. Las fibras ópticas pueden ahora usarse como los alambres de cobre convencionales, tanto en pequeños

ambientes autónomos (tales como sistemas de procesamiento de datos de aviones), como en grandes redes geográficas (como los sistemas de largas líneas urbanas mantenidos por compañías telefónicas).

FIDONET

Red de 10,000 usuarios que usan el protocolo UUCP para compartir emails y archivos y solo usan modems

FIREWALL

Serie de reglas las cuales controlan el tráfico interno de una red con conexión a Internet mediante el control de puertos y protocolos así como a las clases de redes por medio del uso restringido de ips. Puede ser por software o hardware**.**

FTP

Protocolo de transferencia de archivos, opera en redes de cómputo mediante software. Permite compartir archivos entre usuarios.

GNU (GPL)

Desde 1983 es conocido como General Public License aplicado al sistema operativo Linux donde los usuarios pueden hacer mejoras al software y se considera ademas gratuito basado en la FSF (Free software Foundation)

HACKING

Hacking es lo practicado por los hackers y consiste en investigar y buscar fallas en sistemas públicos o privados para poder notificarlas y posteriormente buscar una solución para ella. Los hackers no dañan sistemas solo espían en busca de fallas de seguridad

HFS

Sistema de archivos para los ambientes usados por las computadoras Macintosh

HOME

Partición generada en el disco duro por el instalador del sistema operativo linux

HPFS

Sistema de archivos de alto rendimiento

ISO (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION)

Organización internacional de estándares. Fundada en 1946, con sede principal en Ginebra, ISO establece o fija estándares internacionales. Se ocupa de todos los campos, excepto la electricidad y la electrónica, que se rigen por la International Electrotechnical Commission (IEC), también en Ginebra. Con respecto al procesamiento de información, ISO e IEC crearon recientemente JTC1, Joint Technical Committee para la tecnología de información.

La ISO desarrolla su trabajo a través de más de 160 comités técnicos y 2,300 subcomités y grupos de trabajo, y está constituida por organizaciones de estándares de más de 75 países, algunas de las cuales sirven como secretariados para estos cuerpos técnicos. En los Estados Unidos, la ANSI es miembro de la ISO. Para mayor información, dirigirse a: ANSI, 1430 Broadway, New York, NY 10018.

Se trata del organismo norteamericano encargado de regular las reglas de estandarización en muchos ámbitos, sus decisiones y normas tienen un importante peso específico sobre la industria informática mundial. Incluye el **IEEE** (Institute of Electrical and Electronics Engineers) y la **EIA** (Electronic Industries Association).

LAN (LOCAL AREA NETWORK)

Consiste de varias computadoras y periféricos unidos por un cable o algun otro medio en una área limitada, como un departamento de una compañía o en un mismo edificio. La red permite a la gente compartir recursos como archivos e impresoras, y usar aplicaciones interactivas como una agenda y correo electrónico

MAINFRAME

Es una plataforma de propósito general para el mejor desempeño de las aplicaciones se entiende sobre todo como una computadora de gran tamaño en cuanto a sus características principales.

MINIX

Sistema operativo clon de unix de código abierto basado en un micro-kernel

MODELO OSI

Modelo de referencia de interconexión de sistemas abiertos. Modelo de Arquitectura de redes desarrollado por ISO e ITU-T. El modelo consiste en siete capas, cada una de las cuales especifica funciones particulares de la red, como por ejemplo direccionamiento, control de flujo, control de errores, encapsulación y transferencia confiable de mensajes. La capa superior (capa de aplicación) es la más próxima al usuario; la capa inferior (capa física) es la más próxima a la tecnología de medios. La capa siguiente a la capa inferior está implementada en hardware y en software mientras que las cinco capas superiores están implementadas únicamente en software. El modelo de referencia OSI se utiliza mundialmente para enseñar y comprender la funcionalidad de una red. Es similar en ciertos aspectos a SNA. Véase capa de aplicación, capa de enlace de datos, capa de red, capa física, capa de presentación, capa de sesión, y capa de transporte

NETBEUI

Su significado es NetBIOS Extended User Interface y es un protocolo desarrollado por IBM. Soporta pequeñas y medianas redes en cuanto a transporte y capa de red. No es un protocolo ruteable.

NIC

Placa de circuitos integrados que proporciona la posibilidad de comunicarse entre ordenadores mediante una LAN, normalmente se usan las de tipo ethernet aunque también se basan en el token ring. Su nombre en ingles es network interface card.

NODO

Punto final de una conexión de red, o unión común a dos o más líneas en una red. Los nodos pueden ser procesadores, controladores, o estaciones de trabajo. Los nodos, que pueden variar según su capacidad de enrutamiento y otras capacidades funcionales, pueden estar interconectados por enlaces, y servir como puntos de control en la red. El término nodo se emplea a veces de modo genérico para indicar cualquier entidad que puede tener acceso a una red, y es utilizado a menudo en forma intercambiable con dispositivo.

LA NORMA ISO 11801

La ISO/IEC ha generado en julio de 1994 la norma ISO 11801 que define una instalación completa (componentes y conexiones) y valida la utilización de 100 o 120 W, así como los de 150 W.La ISO 11801 reitera las categorías de la EIA/TIA pero con unos valores de impedancia, de paradiafonía y de atenuación que son diferentes según el tipo de cable. La ISO 11801 también define las clases de aplicación.

La norma ISO 11801 define la distancia máxima (sin introducción de repetidores, amplificadores de señas, etc.)

Clase A	Aplicaciones de voz o bajo fercuencias hasta 100kHz
Clase B	Aplicaciones de datos de baja velocidad hasta 1MHz
Clase C	Aplicaciones de datos alta velocidad hasta 16MHz
Clase D	Aplicaciones de datos de gran velocidad hasta 100MHz
Clase Óptica	Todas las aplicaciones para las cuales el ancho de banda de la fibra óptica es suficiente.

MEDIO	CLASE A	CLASE B	CLASE C	CLASE D	ÓPTICA
Categoría 3	2 Km.	500 m	100 m	-	-
Categoría 4	3 Km.	600 m	150 m	-	-
Categoría 5	3 Km.	700 m	160 m	100 m*	-
Par trenzado 150 W	3 Km.	1 Km.	250 m	150 m	-
FO Multimodo	ND	ND	ND	ND	2 Km.
FO Monomodo	ND	ND	ND	ND	3 Km.

NTFS

Sistema de archivos usados por el sistema operativo Windows NT

OPEN SYSTEM (SISTEMA ABIERTO)

Es un software el cual puede ser modificado por el usuario adaptándose a los requerimientos específicos de el mismo. Un ejemplo claro de este tipo de software es Linux el cual ha ido mejorándose a través de las contribuciones de gente de todo el mundo aparte de los desarrolladores iniciales del sistema.

OSF

Sistema de archivos desarrollado por Digital Equipment Corporation, archivos de sistema operativo.

OSI (OPEN SYSTEM INTERCONNECTION)

Interconexión de sistemas abiertos. Programa de estandarización internacional creado por ISO e ITU-T para desarrollar normas para networking de datos que faciliten la interoperabilidad entre equipos de diversos fabricantes

RED HAT

Es una distribución de linux la cual ha tenido una gran evolución en cuanto a sus versiones . La distribución de más difusión en el mercado.

RED INALÁMBRICA

Una red inalámbrica es un sistema flexible de transmisión de datos sin cables conocido como gíreles. Utiliza tecnología de radiofrecuencia, trasmite y recibe datos utilizando como medio el aire, permitiendo la combinación de conectividad y movilidad.

SHELLSCRIPT

Un shell es un interprete de la línea de comandos. Toma comandos y los ejecuta. Los scripts entonces son programas interpretados y ejecutados por el shell.

SUSE

Distribución Linux desarrollada por Novell de código abierto

TCP/IP

Protocolos encargados de la transferencia de datos entre computadoras. Desarrollados primeramente con fines militares y después encaminados a Internet. Manejan direcciones numéricas conocidas como direcciones IP. La transferencia de datos se realiza mediante sockets.

TERABYTE

Es una medida de la capacidad de almacenamiento de una pc y es equivalente a 2 elevado a la cuarenta potencia.

TOMCAT

Es el contenedor de servlets usado para la implementacion de servlets y java server pages. Es multiplataforma y de distribución gratuita.

UFS

Sistema de archivos manejados por unix.

URL

Se conoce en ingles como uniform resource locator y es un puntero a un recurso en el world wide web.

TOPOLOGÍA ESTRELLA

Esta topología consiste en un dispositivo central al cual van conectados todas las terminales. dicho dispositivo se denomina HUB. En ésta topología toda la información pasa por el centro. La única manera de que deje de trabajar la red es al fallar el HUB, en cualquier otro caso la red sigue funcionando (como lo es la desconexión o falla de algún equipo o dispositivo conectado al HUB).

BIBLIOGRAFÍA

Título	Redes de computadoras : Protocolos, normas e interfaces
Autor	Black, Uyless D.
Editorial	Macrobit : Ra-Ma, c1990
ISBN	970-604-016-1

Título	Redes de computadoras, internet e interredes
Autor	Comer, Douglas E
Editorial	Prentice Hall, c1997
ISBN	970-17-0021-X

Titulo	Linux : installation, configuration and use
Autor	Kofler, Michael
Editorial	Addison-Wesley, c2000
ISBN	0-201-59628-8

Titulo	Linux networking clearly explained
Autor	Pfaffenberger, Bryan,
Editorial	M. Kaufmann, c2001
ISBN	0125331711

Titulo	The Linux Kernel book
Autor	Card, Remy
Editorial	J. Wiley, 1998
ISBN	0471981419 (rustica : papel alcalino)

Titulo	JSP : manual de referencia
Autor	Hanna, Phil
Editorial	McGraw-Hill Interamericana : Osborne Media, c2002
ISBN	84-481-3264-5

Titulo	JSP : ejemplos practicos

Autor	Patzer, Andrew
Editorial	Anaya Multimedia, c2003
ISBN	84-415-1466-6

Titulo	Firewalls y la seguridad en Internet
Autor	Siyan, Karanjit
Editorial	Prentice-Hall Hispanoamericana, c1997
ISBN	968-880-806-7

Titulo	Java : manual de referencia
Autor	Naughton, Patrick
Editorial	Osborne/McGraw-Hill, 1997
ISBN	84-481-1137-0

Titulo	HTML 4 : manual de referencia del programador
Autor	Mullen, Robert
Editorial	Paraninfo, 1998
ISBN	84-283-2490-5

Titulo	Servlets y JavaServer pages : guia practica
Autor	Hall, Marty
Editorial	Pearson Educacion, c2001
ISBN	970-26-0118-5

Titulo	El os/ 2 a fondo
Autor	Letwin, Gordon
Editorial	Anaya multimedia ; redmond, washington : Microsoft, 1989
ISBN	84-7614-214-5

Titulo	Mac OS 8
Autor	Danuloff, Craig
Editorial	Paraninfo, 1998
ISBN	84-283-2442-5
Titulo	Todo sobre Windows NT 4.0
Autor	Kretschmer, B.
Editorial	Marcombo, 1996
ISBN	84-267-1075-1

Titulo	Unix shell programming
Autor	Arthur, Lowell Jay
Editorial	J. Wiley, c1994
ISBN	0-471-59941-7