

18  
24



**UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO**

**FACULTAD DE CONTADURIA Y ADMINISTRACION**

**DISEÑO E IMPLEMENTACION DE UN SISTEMA DE  
ADMINISTRACIÓN Y CONTROL PARA EL SISTEMA DE  
INFORMACION PARA EL BACHILLERATO DE LA UNAM.**

**SEMINARIO DE INVESTIGACION INFORMATICA  
QUE PARA OBTENER EL TITULO DE:  
LICENCIADO EN INFORMATICA  
P R E S E N T A :**

**JULIAN JAVIER ROBLES RIVAS**

**ASESOR DEL SEMINARIO:  
ING. SANTIAGO SUAREZ CASTAÑON**



**MEXICO D.F.**

1997

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**A Dios, por darme incondicionalmente los medios para el logro de esta enhejada meta.**

**A mis padres, Sr. Nazario Robles y Sra. Guadalupe Rivas , quienes bajo su experta y cariñosa tutela tuvieron a bien darme en vida esta invaluable herencia.**

**A Nazario  
Rubén  
Juan Carlos y  
Gabriela**

**por convivir en armonía durante todos estos años, Gracias hermanos.**

**A la Universidad Nacional Autónoma de México por ofrecerme de forma desinteresada el sustento para mi vida futura.**

**A Don Javier ( † ) y Doña Francisca. A Mis abuelos.**

**Al Ing. Santiago Suárez y el Ing. Francisco Becerril, su apoyo fue fundamental en la realización de la presente investigación**

**A Tania Mejía, compañera inseparable y amiga leal siempre.**

**A mis amigos entrañables y seres queridos, por su comprensión, consejo y compañía.**

**Gracias.**

---

---

**INDICE**

<b>CAPITULO I. INTRODUCCIÓN</b>	<b>1</b>
<b>1.1 Antecedentes</b>	<b>1</b>
1.1.1 La motivación para trabajar entre redes.	1
1.1.2 Historia de Internet.	2
1.1.2.1 El IAB (Internet Architecture Board).	6
1.1.2.2 La Sociedad Internet.	6
1.1.3 Servicios de Internet.	6
1.1.4 Importancia de Internet.	7
1.1.5 Arquitectura de Internet.	8
1.1.6 Servicios de información implementados en Internet.	10
1.1.7 El modelo cliente-servidor.	12
<b>1.2 Redunam en Internet.</b>	<b>13</b>
1.2.1 Historia de RedUnam.	13
<b>1.3 DGSCA en RedUNAM.</b>	
1.3.1 Funciones de la DGSCA.	15
1.3.2 Tecnología de RedUNAM.	16
1.3.3 Servicios de información que ofrece RedUNAM.	20
1.3.4 La Coordinación de servicios de Red (CSR) .	22
1.3.5 Objetivo de la Investigación	23
<b>.CAPITULO II . CONTEXTO DEL PROYECTO BACHILLER</b>	<b>24</b>
<b>2.1 Introducción.</b>	<b>24</b>
<b>2.2 Problemática.</b>	<b>24</b>
2.2.1. Situación actual de los servicios.	24
2.2.2 Procedimientos administrativos actuales.	26
<b>2.3 Análisis.</b>	<b>28</b>
2.3.1. Causas y efectos.	29
<b>2.4 Propuesta de solución.</b>	<b>31</b>

---

---

**CAPÍTULO III. DESARROLLO DE LA PROPUESTA AL PROYECTO BACHILLER 34**

<b>3.1</b>	<b>Objetivos.</b>	<b>35</b>
<b>3.2</b>	<b>Metodología.</b>	<b>35</b>
<b>3.3</b>	<b>Especificación estructurada.</b>	<b>35</b>
	3.3.1 Diagrama Entidad - Relación ( DER ).	36
	3.3.2 Diagrama de Contexto del sistema.	37
	3.3.3 Diagramas de flujo de datos (DFD's).	38
<b>3.4</b>	<b>Lenguaje Estructurado.</b>	<b>46</b>
	3.4.1 Miniespecificaciones.	46
	3.4.2 Diccionario de Datos (DD).	50

**CAPÍTULO IV. IMPLEMENTACIÓN DEL ESQUEMA 55**

<b>4.1</b>	<b>Recursos disponibles para el Sistema Bachiller.</b>	<b>57</b>
<b>4.2</b>	<b>Correo electrónico.</b>	<b>57</b>
	4.2.1 Correo electrónico en Bachiller.	58
	4.2.2 Instalación y configuración del software servidor de correo electrónico.	60
	4.2.3 Administración del sistema de correo electrónico.	63
<b>4.3</b>	<b>Listas de correo.</b>	<b>65</b>
	4.3.1 Clasificación de las listas.	65
	4.3.2 Obtención e instalación del software de Majordomo.	66
	4.3.3 Administración de Majordomo.	69
	4.3.4 Empleo de las listas de correo.	72
<b>4.4</b>	<b>Servicio de Gopher.</b>	<b>73</b>
	4.4.1 Obtención e instalación del software servidor de Gopher.	74
	4.4.2 Incorporación de información al servicio de Gopher.	77
<b>4.5</b>	<b>Servicio de World Wide Web.</b>	<b>80</b>
	4.5.1 Obtención e instalación del software servidor HTTP.	82

**CAPÍTULO V. CONCLUSIONES 85**

<b>APENDICE 1</b>	<b>87</b>
<b>APENDICE 2</b>	<b>88</b>
<b>APENDICE 3</b>	<b>91</b>
<b>APENDICE 4</b>	<b>126</b>
<b>BIBLIOGRAFIA</b>	<b>133</b>

---

---

**CAPITULO I****INTRODUCCIÓN****1.1 Antecedentes****1.1.1 La motivación para trabajar entre redes.**

En la actualidad las comunicaciones se han convertido en parte fundamental de las actividades relacionadas de alguna forma con el cómputo. Las redes de computadoras alrededor del mundo manejan información de los más diversos temas, dentro de este ambiente es posible crear listas de correo electrónico, para grupos de distintas disciplinas, de tal manera que puedan compartir información de interés común; de la misma forma es posible intercambiar programas de diversión para más tarde ejecutarlos en casa.

En el mundo científico las redes de datos son básicas, ya que permiten a los científicos enviar programas y datos hacia computadoras remotas para su proceso, obtener su resultado y comentarlo con sus colegas.

Sin embargo muchas redes son independientes y están integradas para satisfacer las necesidades de un solo grupo. Los usuarios eligen entonces una tecnología apropiada a sus necesidades de comunicación.

Es importante señalar que es imposible construir una red universal (global) a partir de un solo tipo de tecnología de hardware, ya que esta no solucionaría los requerimientos específicos de cada una de las redes, ya que en algunas ocasiones se necesitarán altas velocidades de transmisión para las máquinas interconectadas y algunas de ellas no pueden ser conectadas a largas distancias, donde son requeridas.

En los últimos tres lustros, se involucra ya a una tecnología que permite interconectar muchas y variada redes, físicamente separadas, y las hace funcionar como una sola red, este concepto se llama " **Internetworking** " (trabajo entre redes), e incorpora diversas y variadas tecnologías de hardware, que proveen una forma de interconectar redes heterogéneas y definir convenciones de comunicación.

De este concepto surge la palabra "Internet", ocultándose los detalles físicos de la red, permitiendo a las computadoras comunicarse independientemente de sus características físicas, podemos decir que es un sistema abierto de interconexión, ya que incorpora sistemas de comunicación propietarios diversos y de distintos fabricantes .

Podemos puntualizar que Internet en su totalidad, se ha conformado de redes adicionadas paulatinamente, y junto con ellas se han diseñado equipos y dispositivos que agilizan la comunicación entre computadoras con diversas arquitecturas, que usen cualquier tipo de hardware de conmutación de paquetes en red, y distintos tipos de

sistemás operativos, es decir, que la han hecho más eficiente y la han ajustado a las nuevas necesidades.

### **1.1.2 Historia de Internet.**

Internet es el concepto de incorporación de redes más grande en el mundo, cuenta con 6 millones de computadoras conectadas, según los más recientes registros.

Permite además, la interconexión y comunicación entre prácticamente todo tipo de entidades y organizaciones, siendo su objetivo principal la implementación de los servicios de información, a partir de este sinnúmero de computadoras conectadas.

Internet surge con la interconexión de la Universidad de California en Los Ángeles, el Instituto de Investigaciones de Stanford, la Universidad de Utah y la Universidad de California en Santa Bárbara a finales de los años sesenta, con el fin de compartir los recursos de cómputo que tenían disponibles.

Siguiendo este ejemplo, el Departamento de Defensa de los Estados Unidos de Norteamérica a través de su Agencia de Proyectos de Investigación Avanzada (ARPA), introdujo el principio de compartir recursos de cómputo, no centraliza, construyendo la primera red de datos, conocida como ARPANET, que se caracterizó por la interconexión de sus nodos haciendo uso de la conmutación de paquetes.

La conmutación de paquetes toma su nombre debido a que la información a transmitir es descompuesta en trozos llamados "paquetes" (a diferencia de la conmutación de mensajes que envía mensajes completos), los cuales se insertan dentro de los campos de información de un protocolo y recorren la red como entidades independientes, cada uno lleva en su estructura el destino y el orden que tiene dentro del mensaje original, este tipo de conmutación implica el empleo de múltiples conmutadores a lo largo de la red, en los que se distribuye el proceso de conmutación, para crear rutas alternativas de transmisión y un proceso de ensamblaje al llegar los paquetes al destino. Contrariamente en la conmutación de mensajes, que apareció años antes, donde existe un solo conmutador que dedica un puerto a cada usuario, almacena los mensajes a transmitir, les asigna prioridades y los envía uno a uno, al decidir en cada ocasión la ruta que seguirá hasta su destino, por lo que la transmisión no es en tiempo real, y generalmente crea un cuello de botella, al pasar todos los mensajes por un solo conmutador.

La conmutación de paquetes fue desarrollada al buscar un método de transmisión vocal seguro, donde al transmitir conversaciones fraccionadas, a través de diversas rutas, su intercepción no representaría peligro.

Poco después se notó que el esquema de la conmutación de paquetes era aplicable a la transmisión de datos, ya que los dispositivos de cómputo se comunican en ráfagas, ocupando el canal de transmisión intermitentemente; el Departamento de Defensa tras un estudio acerca de la naturaleza del tráfico de datos, reveló que un 99% de las transmisiones de datos son más cortas que las conversaciones de voz, el 25% de las transmisiones de datos duran menos de 1 segundo, el 50% menos de 5 segundos, y el 90% menos de 50 segundos.

La tecnología de conmutación de paquetes está concebida para adaptarse a estas características, entrelazando las ráfagas de tráfico en paquetes sobre un mismo canal, equilibrando el tráfico de información, que generalmente es asimétrico, sin embargo implica igualmente un grado mayor de complejidad en cuanto a los enlaces.

Algunas ventajas que ofrece la conmutación de paquetes son:

- Incrementa la seguridad de la información, al estar ésta dividida en diversos paquetes autónomos.
- Reduce la vulnerabilidad a fallas en la red, al distribuir el proceso de conmutación de forma distribuida.
- Puede evitarse el paso por nodos averiados u ocupados, al existir rutas alternativas.
- Permite multiplexar sesiones enteras de un usuario (enviar varias señales por el mismo canal) por un mismo puerto de la computadora, en lugar de dedicar un puerto a cada usuario. Este sistema intercala en un mismo puerto las ráfagas de datos de los distintos usuarios, creando un canal virtual, que es cuando el usuario percibe que hay un puerto dedicado a él, pero en realidad lo comparte con otros usuarios.

En 1983 ARPANET es dividida en dos redes debido a su gran crecimiento, ARPANET y MILNET, esta última dedicada a interconectar centros militares.

A partir de este momento se aceleró el crecimiento de Internet, adicionando cada vez un mayor número de computadoras. Surge entonces la necesidad de contar con convenciones en los métodos de comunicación (protocolos) para evitar incompatibilidades subsecuentes.

Dicho crecimiento es ilustrado en la Figura 1.1.2, donde de forma cronológica se muestran las etapas de evolución de Internet, asociadas al número de máquinas interconectadas dentro de la misma.

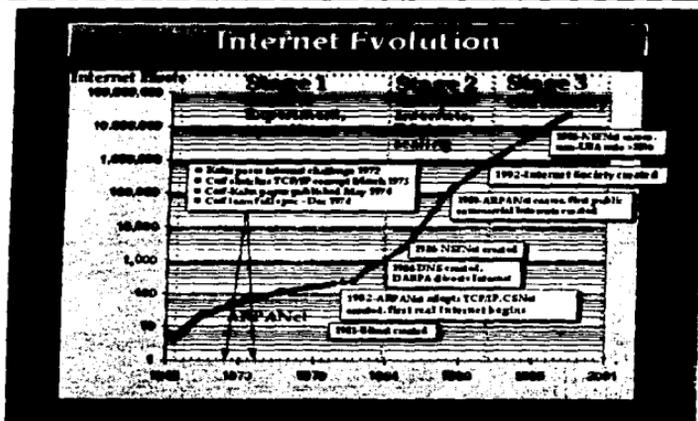


Figura 1.1.2

Etapas de evolución de Internet. Fuente <http://www.isoc.org>

La tecnología de ARPANET incluía un conjunto de estándares de red, que especificaban a detalle, la forma en que las computadoras debían conectarse, de la misma forma que definían un conjunto de convenciones de interconexión y ruteo de tráfico.

A este conjunto de estándares se le denomina oficialmente TCP/IP (Transmission Control Protocol/Internet Protocol) a raíz de los dos principales estándares que lo conforman, este tipo de implementación es especialmente interesante debido a que ha demostrado su viabilidad en redes de gran escala, y ha servido de base tecnológica en la implementación de una red global, que conjunte campus universitarios, centros de investigación, corporativos comerciales, gubernamentales, hogares y todo tipo de instituciones.

Bajo el esquema de los estándares TCP/IP cada equipo conectado a Internet debe contar con una identificación única, compuesta de dos elementos, una dirección numérica o IP (Internet Protocol) definida de forma convencional y una dirección Ethernet definida de forma física por el fabricante en el adaptador de red (tarjeta), ambas deben ser únicas.

Par facilitar su manejo, la dirección IP suele asociarse a un nombre, formado con el nombre dado a esa computadora, su dominio y subdominio respectivamente, dicho nombre hace referencia a la dirección de forma unívoca.

A raíz de la incorporación masiva de máquinas, la asignación de dichos identificadores (IP y nombres) se hizo más compleja, así como la actualización y manejo de las tablas que los contenían, y que debían ser enviadas a cada uno de las máquinas para que contaran con una lista de sus posibles destinos de conexión.

Es creado entonces el Servidor de Nombres de Dominio (DNS), que es un sistema de base de datos implementado de forma distribuida, que reside en un computador destinado a este fin, se encuentra basado en el modelo cliente-servidor.

Para facilitar el manejo de los identificadores se cuenta con programas denominados servidores de nombres, que contienen las tablas de identidad de las computadoras correspondientes a un segmento de red, así como de los clientes (en este caso "resolvers" que son programas cuya función es la de obtener las direcciones adecuadas).

Un DNS cuenta con una estructura de tipo jerárquica, como un árbol binario, su raíz es el centro de información de la red (NIC, Network Information Center), siendo sus hijos los dominios, de los cuales almacena las direcciones de sus respectivos DNS, estos a su vez cuentan con subdominios como hijos, sus DNS se encuentran identificados en el DNS de nivel superior, y así sucesivamente hasta llegar a las hojas, las computadoras de un segmento de red.

Un dominio puede estar constituido por una o varias redes, cada segmento de red cuenta con uno o más servidores de nombres donde residen las direcciones de todas las máquinas que componen dicho segmento, o una sola máquina en algún segmento puede concentrar los identificadores de las computadoras de varios de los mismos, si un DNS no cuenta con una dirección que le sea solicitada, la consultará con una máquina análoga de nivel superior hasta obtener dicha información por medio de la consulta por otra rama del árbol jerárquico.

En cuanto a las líneas de comunicación usadas entre las redes, podemos señalar que ARPANET comenzó con líneas de 56 Kbps hasta 1983, año en el que empleó la tecnología Ethernet a 10 Mbps.

En 1988 la Fundación Nacional de la Ciencia (NFS) de los Estados Unidos de Norteamérica implementó una red principal (backbone) de 1.5 Gbps para interconectar los principales centros de cómputo de los Estados Unidos.

El avance en las tecnologías de comunicación fue acompañado del desarrollo de protocolos adecuados a estas tecnologías, algunos de ellos son BSC, SDLC, TDMA, TDM, ALOHA en sus diversas modalidades, Token Ring, Token Bus y Ethernet, siendo el más difundido un conjunto de ellos que a través de una estructura de protocolos estratificados forman lo que conocemos como TCP/IP (Transmission Control Protocol / Internet Protocol); junto con los protocolos se implementaron diversos mecanismos de detección de errores, de los que podemos mencionar ACK/NACK (Acknowledge/Negative Acknowledge) y CRC (Comprobación de Redundancia Cíclica).

### **1.1.2.1 El IAB (Internet Architecture Board).**

Debido a que el conjunto de protocolos TCP/IP de Internet no surgieron de un proveedor específico o de alguna sociedad profesional reconocida, surgió el IAB (Consejo de Arquitectura de Internet), originalmente entendido como Consejo de Actividades de Internet. Su función es la de coordinar la investigación y desarrollo hechos bajo los protocolos TCP/IP, y dirige la evolución de Internet, decide cuales protocolos son parte requerida para el conjunto existente de protocolos TCP/IP y los establece como oficiales. Este organismo surgió en 1983, cada uno de sus miembros forma una fuerza de tarea, totalizando aproximadamente 10, que van desde la que investiga las distintas formas de tráfico entre varias aplicaciones, hasta una que maneja problemás de ingeniería en Internet a corto plazo.

Su última reorganización fue en 1989 debido a la diversificación entre el ámbito de investigación y el comercial, este último fue el que dió el impulso definitivo al crecimiento de Internet.

Los investigadores fueron reubicados del IAB hacia un organismo subsidiario, y el nuevo IAB fue constituido de tal forma que contara con representantes de toda la comunidad.

### **1.1.2.2 La Sociedad Internet.**

En 1992 debido a la separación de la Internet de sus raíces gubernamentales en los Estados Unidos de Norteamérica, se implementó una sociedad con el fin de promover la participación dentro de Internet (acción análoga a la de la Dirección de Servicios de Cómputo Académico de la UNAM), a esta organización se le dió el nombre de La Sociedad Internet<sup>1</sup>, este grupo está inspirado en la NGS (National Geographic Society).

### **1.1.3 Servicios de Internet.**

Para comprender los detalles técnicos bajo los cuales trabaja TCP/IP, es necesario entender los servicios que proporciona en una red.

Los protocolos de TCP/IP marcan las reglas para llevar a cabo las comunicaciones, contienen los detalles de los formatos de los mensajes que viajarán por la red, describen la forma de respuesta de la máquina que las recibe, y especifican cómo una máquina maneja los errores y condiciones anormales, podemos decir que los protocolos son a la comunicación, como los algoritmos a la computación, de manera similar un protocolo de comunicación permite especificar y entender la forma de comunicación

---

<sup>1</sup>Para mayor referencia consultar la suite de WWW <http://www.isoc.org>

independientemente de las particularidades físicas de los equipos entre los que se lleva a cabo dicho intercambio de información.

El no preocuparse por los detalles de comunicación es de gran ayuda a la productividad de las aplicaciones diseñadas para actuar en el ambiente de red con tres ventajas fundamentales :

- Los programadores interactúan con abstracciones de protocolos de los niveles más altos ( de aplicación), ya que su labor no implica ahondar en detalles de configuración de hardware y gracias a esto son capaces de crear nuevos programas de forma rápida.
- Debido a la misma razón los programas no son sustituidos cada vez que se modifica el equipo o la configuración del mismo.
- De igual forma es posible la comunicación directa para un par arbitrario de máquinas, es decir, no se requieren versiones especiales de aplicaciones para trasladar información entre computadoras distintas en su tipo.

Podemos concluir que los servicios que proporciona una red (tales como los mencionados anteriormente), son producto de los protocolos que son utilizados para implementarla.

#### **1.1.4 Importancia de Internet.**

La grandes ventajas que proporciona el usar Internet son:

- Comunicación directa sin importar la ubicación geográfica de los participantes.
- Acceso a bancos de información, ubicados en forma dispersa, generados y mantenidos por grupos de trabajo diversos, de forma transparente al usuario.
- Acceso a información generada en cualquier nodo a lo largo de la red.
- Los servicios que proporciona, siendo aquí solamente señalados los más importantes, como las sesiones remotas, transferencia de archivos, correo electrónico y los grupos de trabajo.
- Evita el traslado físico de información independientemente de su tipo.

Estas ventajas se traducen en facilidades de interacción entre grupos de trabajo, de investigación, académicos y muchos otros para realizar desarrollos de todo tipo entre personas que se encuentran geográficamente distantes, compartiendo distintos recursos de información (audio, video, voz, datos e imágenes) y tecnológicos (aplicaciones y herramientas de desarrollo) de forma casi instantánea; lo que a su vez ha incrementado la velocidad en el avance tecnológico principalmente. De la misma forma han favorecido la simplificación en la administración de todo tipo de organizaciones con filiales, dependencias o sucursales situadas en puntos localizados a diversas distancias, con lo

que la información que es generada desde alguno de ellos está disponible también para los demás.

También ha sido favorecida la comunicación alrededor del mundo de manera interdisciplinaria, incorporando a todo tipo de instituciones, aunque el esquema en su conjunto tiene un gran apoyo del ámbito comercial, que fomenta la investigación que se traduce en mejoras continuas al funcionamiento de Internet.

A través del conocimiento de como surgió el concepto "Internet" podemos concluir que el conjunto de redes que la conforman, están administradas de manera autónoma, y que cuentan con un(os) servidor(es) de nombres por cada uno de los dominios que las componen.

Existe una entidad llamada Internic (Internet Network Information Center) que es el centro de Información de la red, dedicado a la asignación y distribución de las direcciones IP de las máquinas que forman parte de Internet, para evitar duplicidades y facilitar su clasificación.

Lo comentado anteriormente nos lleva a definir el concepto de "Red universal", que hace posible unificar un mismo sistema de redes, manteniendo un servicio de comunicación estándar, lo que oculta los detalles de arquitectura implementados en cada una de las redes; también permite la comunicación entre cualquier par de nodos, independientemente de su ubicación dentro de la distribución de la red, además, las aplicaciones y programas empleados por el usuario son independientes del hardware que compone a la red.

De la misma manera podemos definir el concepto de "Red Global", que incluiría no solo las redes de datos (internet), sino todo tipo de redes (telefónicas, telex, satélites, microondas, etc) y sus dispositivos para interconectar con prácticamente todos los ámbitos de la vida, permitiendo la difusión del conocimiento, así como el monitoreo de todo tipo de actividad económica, comercial, educativa o de interés general.

### **1.1.5 Arquitectura de Internet.**

Las redes de computadoras se comunican a través de dispositivos que se encargan de intercambiar la información en forma de paquetes entre ellas, estas máquinas son denominadas encaminadores (gateways), que hacen que dos redes con la misma topología, parezcan una misma.

Un encaminador emplea conexiones directas a otras redes y transfieren la información necesaria entre las mismas, para intercambiar la información entre redes debe ser segmentada en paquetes que son enviados a un destino determinado, el encaminador los captura y los intercambia con el encaminador de la otra red, que los recaptura y transmite al destino final.

Un encaminador debe contar con funciones de ruteo, que son las que permiten elegir la trayectoria adecuada entre las redes, y que es necesaria para que los paquetes lleguen a

---

su destino, para lo anterior es necesario conocer las topologías implementadas en las redes más allá de las contiguas a la nuestra.

Cada encaminador cuenta con unas tablas que contienen la información relevante de las demás redes, esto para quien usa Internet no es evidente, ya que interactúa con un concepto de red única, y no se preocupa por la forma en que las redes se interconectan y comunican, independientemente del hardware y software que implementen.

Dentro de las formas en como se comunican las redes, se encuentran diversos estándares, el conjunto más importante es TCP/IP, generalmente aceptado en Internet y fundamentado en el modelo OSI (Open System Interconnection).

### **Descripción del protocolo TCP/IP.**

TCP/IP es un conjunto de protocolos, con independencia del equipo (hardware), del sistema operativo (software) y el tipo de red (topología) implementados.

La arquitectura de este conjunto de protocolos esta basada en cuatro capas, a diferencia del modelo OSI que cuenta con siete.

**Capa de acceso a la red:** son las rutinas en conjunto que permiten el acceso entre redes a nivel físico. Determina la forma de empleo de la red para transmitir un datagrama, que es un formato de paquete de información, con las características suficientes para poder ser enrutado de un origen a un destino.

**Capa Internet:** esta capa se encarga de definir los datagramas y el manejo del ruteo de los datos.

Contiene al protocolo internet (IP), IP es la piedra angular de internet, ya que sus actividades son básicas:

- Definir el esquema de direccionamiento.
- Trasladar los datos de la capa anterior ( de acceso a la red ) a la capa siguiente ( de transporte).
- Enrutar los datagramas hacia las máquinas correspondientes.
- Realizar la fragmentación de los datagramas.
- Realizar el reensamblaje de los datagramas.

**Capa de transporte:** también conocida como "host to host" que proporciona el servicio punto a punto de los datos.

Cabe señalar que como parte del IP se encuentra el protocolo IMCP (Protocolo de Control de Mensajes de Internet), cuyas funciones son el control de flujo, redireccionar rutas, ubicar máquinas y detectar máquinas inalcanzables o fuera de operación.

Esta capa contiene dos protocolos.

a) Protocolo de datagrama de usuario (UDP).

Proporciona un servicio no dirigido a la conexión ( en los que no existe una coordinación entre los nodos fuente y destino para la correcta transmisión de los datos), es decir, no cuenta con las técnicas para verificar la integridad del dato al llegar a su destino.

b) Protocolo de control de transmisión (TCP).

Proporciona el servicio de entrega de datos, que cuenta con un mecanismo de detección y corrección de errores a nivel red y de enlace.

**Capa de aplicación:** se encuentra formada por las aplicaciones y procesos que se emplean en una red, estos son específicamente los servicios que ofrece Internet, tales como sesiones remotas (Telnet), transferencia de archivos (FTP) y correo electrónico (SMTP Simple Mail Transfer Protocol).

### 1.1.6 Servicios de Información Implementados en Internet.

Con las facilidades de todo tipo que proporciona la arquitectura de internet, es posible implementar diversos y variados servicios de información, disponibles remotamente desde cualquier punto de la red, estos servicios son generados y mantenidos por diferentes e Innumerables grupos de trabajo a lo largo y ancho de la red. La información es accesible de forma sencilla en distintos formatos (texto, hipertexto, voz, imágenes, video) y puede localizarse con ayuda de herramientas de búsqueda. A continuación mencionamos los más importantes:

#### **Servicios de búsqueda.**

- **Archie.** Servicio de consulta de información a partir de una palabra clave, fue implementado por la Universidad de McGill en Montreal y circunscribe sus búsquedas a los servidores de FTP anónimo, siendo esta vía la indicada para obtener la información resultante de la búsqueda.
- **Verónica:** Servicio que mantiene un índice de títulos tomados de los menús de gopher y permite realizar búsquedas a partir de palabras clave en dicho índice, esta búsqueda se lleva a cabo mediante clientes de gopher y el resultado de la búsqueda es mostrado mediante un menú semejante a los presentados en gopher.
- **Judhead:** Es una herramienta que realiza búsquedas sobre menús de gopher a partir de una palabra clave, restringe su rango de búsqueda a los gophers que se encuentran en el mismo dominio de red, los resultados de la búsqueda son presentados como un nuevo menú de gopher.
- **WAIS.** (Wide Area Information Service). Es un servicio de información de área amplia, es principalmente un protocolo, y es un sistema diseñado para acceder información

indizada, baja un cierto formato especial que permite una recuperación de texto de forma completa. Esta compuesto de un indizador, un servidor y un cliente.

### **Servicios de información .**

- **Gopher.** Desarrollado por la Universidad de Minnessota, que ofrece información organizada de forma jerárquica de acuerdo a distintos criterios y presentada en forma de menús. Este servicio se basa en el modelo cliente-servidor y los servidores que lo proporcionan se encuentran en todo lo largo de internet.
- **FTP anónimo ( File Transfer Protocol),** son servidores de archivos correspondientes a información organizada por temás, que incluye herramientas de todo tipo, software de aplicación y otros, esta información es proporcionada con diversos fines, como probar software comercial o proporcionar recursos de forma gratuita, este servicio es accesados sin requerir una clave especial de usuario, es decir, son públicos, siendo la única restricción los derechos de autor del material que en ellos se proporciona.
- **WWW (World Wide Web).** Iniciado en 1989 en el Laboratorio Europeo de Física de Partículas (CERN), con fines de compartir investigaciones e ideas de proyectos científicos. inició con un sistema de hipertexto sencillo para redes, de forma útil para comunicarse y transmitir documentos. Actualmente incorpora audio, video e imágenes, lo que lo ha convertido en el servicio de Internet más popular, debido a su gran penetración y a las grandes ventajas ilustrativas que han sido apoyadas y aprovechadas por las empresas comerciales por su grado de alcance.

El software de WWW se basa en el modelo distribuido de cliente-servidor, que se explicará en un apartado sucesivo .

Los programás clientes en este caso son los llamados navegadores, el primero en popularizarse fue Mosaic, que es un cliente gráfico desarrollado por el Centro Nacional para Aplicaciones de Supercómputo (NCSA) de los Estados Unidos de Norteamérica.

Más tarde surge el más popular en la actualidad: Netscape.

El protocolo que emplea WWW es el HTTP (Hypertext Transfer Protocol), por dicha razón los servidores de WWW son denominados servidores de HTTP.

El "language" estándar para desarrollar documentos de hipertexto es el HTML (Hiptertext Markup Language), además del SGML (Standar Generalized Markup Language).

Estos lenguajes permiten la incrustación de audio, video e imágenes dentro de los textos transmitidos, proporcionándoles una presentación ilimitada.

### 1.1.7 El modelo cliente-servidor

El patrón principal de interacción y cooperación entre aplicaciones dentro del ámbito de red es el modelo cliente-servidor, este tipo de interacción conforma la base de la comunicación en red.

El término "servidor" es aplicable a cualquier programa que ofrece un servicio que puede ser obtenido a través de la red. Un servidor acepta una petición a través de la red, realiza un proceso y regresa el resultado a quien envió la petición .

Un programa ejecutable se convierte en un "cliente" cuando envía una petición a un servidor y espera una respuesta.

Debido a que este modelo es una extensión de la comunicación interproceso en una sola máquina, es fácil construir programas que usen este modelo para interactuar.

Algunas de las características de los servidores son :

- Capacidad de realizar tareas simples o complejas.
- Generalmente son implementados como programas de aplicación, es decir, se encuentran en el nivel de aplicación dentro del modelo de protocolos estratificados (por ejemplo el modelo OSI), la ventaja de implementarlos de esta forma es que pueden ser ejecutados en cualquier computadora que soporte el conjunto de protocolos TCP/IP, sin involucrarse con las capas más bajas del mismo modelo.
- Un servidor para un servicio en particular puede ser ejecutado en un sistema con tiempo compartido con otros programas, o puede ser ejecutado en una computadora personal.
- Pueden existir variados servidores para el mismo servicio, y pueden ser ejecutados en varias computadoras o en una misma.  
Es recomendable, de hecho, colocar copias de un determinado servidor de forma físicamente independiente (en varias máquinas), a fin de incrementar la disponibilidad y el desempeño de un determinado servicio.
- El término servidor, además de los programas descritos, es aplicable a las máquinas cuyo funcionamiento está basado en un programa servidor en particular.

El acceso a los servidores mediante programas clientes ( programas especializadas en realizar peticiones a los servidores y de recibir sus respuestas) consiste en que libera a la máquina donde reside el programa servidor , de ciertas rutinas no críticas, que en un número considerable de peticiones provocan mayor tiempo de proceso, liberándole de carga de procesos y agilizando sus respuestas, delegando dichas tareas a la máquina donde reside el cliente.

## 1.2 Redunam en Internet

### 1.2.1 Historia de RedUnam.

En un principio la Universidad Nacional Autónoma de México contaba con una infraestructura centralizada de equipo de cómputo, del tipo mainframe, conectados a estos se encontraban terminales cuya ubicación no era precisamente la más cercana a quienes las empleaban.

La operación de estos equipos se basaba en el intercambio de información a través de cintas y discos magnéticos.

A continuación se muestra a manera de tabla en orden cronológico, los acontecimientos y eventos relevantes en cuanto acómputo se refiere dentro de la UNAM, a fin de dar a conocer la evolución y crecimiento que en este campo ha experimentado a lo largo de los últimos años.

AÑO	ACTIVIDAD
1958	La red universitaria comienza a desarrollarse, al instalarse en la Facultad de Ciencias una IBM 650, conjuntamente se crea el Centro de Cálculo Electrónico (CCE).
1960	El CCE adquiere una Bendix G-15.
1963	El departamento de Biocibemética adquiere una Bull Gamma-30 y una AD-224.
1964	Se adquiere una PDP-8 de Digital Equipment Corporation (DEC).
1965	Se instala una Bendix G-20 y una G-15 creándose con las mismas el Centro Móvil de Cálculo Electrónico. Todas estas máquinas eran autosuficientes.
1969	Se inicia la etapa del Teleproceso al instalarse una Burroughs B-5500 y la Dirección General de Sistematización de Datos instala una IBM-360/40.
1970	Se crea el Centro de Investigación de Matemáticas Aplicadas, Sistemas y Servicios (CIMASS).
1973	El CIMASS instala una Burroughs B-6700 y es dividido en el CIMASS y el Centro de Servicios de Cómputo (CSC). Más tarde el CIMASS adquiere la calidad de Instituto.
1981	El CSC da lugar al Programa Universitario de Cómputo (PUC), que proporcionaría servicios a las áreas de investigación, docencia, administración y administración académica.
1985	Se crea el Consejo Asesor de Cómputo de la UNAM, el PUC desaparece dando lugar a la Dirección General de Servicios de

	<p><b>Cómputo Académico (DGSCA).</b></p> <p>A partir de la Dirección para la Administración Central surge la Dirección General de Servicios de Cómputo Académico para la Administración.</p> <p>Se forma la comisión de Teleinformática encargada del desarrollo de la Red Universitaria de Cómputo, formada finalmente en 1986 con uso de las líneas privadas, modems y enrutadores bajo un protocolo X.25 , que mediante la red de transmisión de datos Telepac, con enlaces nacionales e internacionales con velocidades de 19200 bps.</p>
1986	<p>La DGSCA instala una IBM-4381 conjuntamente a una red token ring, sustituyendo los equipos Burroughs B-6700 por dos equipos A-9 y A-3 de Unisys.</p>
1987	<p>Se presenta el plan para integrar la Red Universitaria a la Red Académica Mundial de Cómputo (BITNET), a la que se conectó mediante línea telefónica privada al ITESM Monterrey, después dicho enlace es sustituido por uno satelital UNAM-BOULDER-ITESM con BITNET II.</p> <p>La red Universitaria se incorpora a Internet, cuando se negocia con la NFS una conexión para el Instituto de Astronomía para acceder a bases de datos de imágenes, mediante un enlace satelital.</p>
1988	<p>Se sustituye la Burroughs B-6800 por una A-12 y una A'6 de Unisys, es en este año cuando son instaladas las primeras redes token ring en convenio con IBM, a fin de compartir recursos y realizar transferencias de archivos.</p>
1989	<p>Se planea la implementación de un anillo central como "backbone" que enlazaría los centros con token ring, que constaba de tres principales nodos: DGSCA, IIMÁS y el Instituto de Astronomía, incorporados como subanillos.</p> <p>en este año surge formalmente la Red Universitaria "RedUnam" con la inauguración de una red token ring , una red ethernet y la red satelital.</p> <p>se implementa también un enlace vía microondas entre el IIMÁS y la Dirección general de Servicios de Cómputo para la Administración.</p>
1990	<p>El A-6 de la Dirección de Cómputo para la Administración Académica (DCAA) se traslada ala Dirección General de Incorporación y Revalidación de Estudios (DGIRE), y a la DCAA llega un equipo Cyber 170/855 de Control Data.</p>
1991	<p>La UNAM inicia la era del supercómputo en América Latina con la adquisición de la Cray YMP 4/432, dedicada a labores de procesamiento de imágenes y supercómputo.</p>

Cabe señalar que gracias a los esfuerzos realizados por la UNAM, se convirtió en la base del desarrollo del cómputo en México, manteniendo una infraestructura que le ha permitido ser la red más importante de nuestro país, además de ser la casa de estudios que formó a los expertos más reconocidos a nivel nacional, y que actualmente dirigen los

organismos más importantes concernientes a las actividades relacionadas con el cómputo.

La UNAM a través de la DGSCA mantiene programas de formación de recursos humanos en el ámbito computacional de manera ininterrumpida, fomentando la investigación, difusión y desarrollo en éste campo a nivel nacional, promoviendo el intercambio de conocimientos con todo tipo de instituciones.

### **1.3 DGSCA en RedUNAM.**

La Dirección General de Servicios de Cómputo Académico surge en 1985 a raíz del Programa Universitario de Cómputo, conservando como recursos de cómputo una Burroughs 7800 y una IBM/4381, esta última empleada para correo electrónico únicamente.

La función de la DGSCA es la dar apoyo y soporte a la infraestructura de redes, telefonía y de servicios de cómputo a las dependencias de la UNAM.

Al mismo tiempo, se convirtió en lugar de concentración de los equipos de cómputo y dispositivos de comunicación más avanzados, identificándose como el símbolo del avance en materia tecnológica en cuanto a cómputo y comunicaciones dentro de UNAM.

Al mismo tiempo la Dirección se caracteriza por el empleo e introducción de equipos provistos por Sun Microsystems y Silicon Graphics, en sus distintos modelos, que son con los que actualmente cuenta .

Los equipos Sun van desde SparcClassic, Sparc 630MP, hasta SparcCenter 1000 , 2000 y Storage Arrays, los cuales soportan los diversos servicios de información y de cómputo ofrecidos por la UNAM, el detalle de las características y actividades a que se destinan los equipos, especialmente los administrados por la Coordinación de Servicios de Red se encuentra en el Apéndice 1.

Los equipos Silicon Graphics Indy, Indigo2 y Onix, son empleados por su gran capacidad de almacenamiento, de manejo de gráficos y velocidad de proceso son empleados para labores de visualización y simulación, así como diseño complejo.

Se cuenta con dos robots de discos magnéticos Hewlett Packard para fines de almacenamiento masivo, por su gran capacidad y versatilidad, así como por la facilidad en la disposición y asignación de espacio de almacenamiento.

#### **1.3.1 Funciones de la DGSCA.**

El objetivo de la DGSCA es el de proporcionar la infraestructura adecuada a las necesidades de comunicaciones, transmisión y procesamiento de información a las dependencias de la UNAM.

Dada la gran diversificación en estos campos, la DGSCA cuenta con distintas direcciones especializadas en la provisión de cada uno de los servicios.

Las Direcciones que conforman la DGSCA son:

- Dirección de Telecomunicaciones Digitales (DTD) .Encargada de proporcionar el servicio e infraestructura telefónica y todo tipo de redes de transmisión de información y acceso a RedUNAM, a las dependencias de la UNAM, dentro y fuera del campus universitario.

Recientemente tomó bajo su responsabilidad a la Coordinación de Servicios de Red, encargada de la generación, soporte y mantenimiento de los servicios que se brindan a los usuarios de RedUNAM, tales como, Correo electrónico, Gopher, FTP, W W W y Servicios de búsquedas en bases de datos nacional e internacionalmente.

- Dirección de Cómputo para la Investigación. Proporciona servicios al sector dedicada a la investigación en la UNAM, brinda apoyo a los proyectos que se derivan del ámbito científico que requieren de gran respaldo de actividades derivadas del cómputo, tales como el supercómputo.
- Dirección de Cómputo para la Docencia. Se encarga de la difusión, capacitación y promoción de las actividades educativas en lo que al cómputo se refiere. Siendo su principal función la planeación y organización de todo tipo de cursos relacionados con temas de computación en todos sus niveles, para lo que cuenta con dos instalaciones fuera de la Ciudad Universitaria, el Centro Máscarones y el centro Nuevo León.
- Dirección de Cómputo Académico para la Administración Académica (DCAA). Dedicada al apoyo a aspectos de Administración educativa dentro de la Universidad, principalmente a la Dirección General de Administración Escolar (DGAE) y a la Dirección General de Incorporación y Revalidación de Estudios (DGIRE).

### **1.3.2 Tecnología de RedUNAM**

La RedUNAM por sus características los servicios que ofrece y por el papel tan importante que desempeña dentro de la difusión de la cultura y el conocimiento en nuestro país se ha convertido sin lugar a dudas en una de las más importantes de Latinoamérica.

En atención a lo anterior, la Universidad Nacional Autónoma de México, consciente de las nuevas tendencias mundiales en tecnología de información, ha desarrollado uno de los proyectos más ambiciosos en lo que ha Telecomunicaciones se refiere: La red Integral de Telecomunicaciones de la UNAM.

## **La Red.**

RedUNAM incorpora diversos tipos de tecnología, entre las más importantes destacan:

- FDDI (Fiber Distributed Data Interfase)
- Ethernet

## **FDDI.**

La red FDDI es un anillo doble con un rango de transmisión de 100 Mbps implementado sobre fibra óptica, cumple con estándares de la ANSI y la ISO, reemplaza a Ethernet para conectar computadores e interconectar redes locales cumpliendo la función de red principal (backbone).

El estándar ANSI para FDDI se compone de cuatro elementos claves:

- **Capa de control de acceso a medios (MAC).**  
Define direccionamiento, calendarización y ruteo de datos, se comunica además con los protocolos de las capas más altas (aplicación) como TCP/IP, SNA e IPX por ejemplo.
- **Capa física (PHY).**  
Maneja la codificación y decodificación de paquetes de datos dentro de cadenas de símbolos a través del cableado, maneja también la sincronización del reloj en el anillo FDDI, el método de codificación que usa es el 4B/5B.
- **Capa de dependencia física de medios (PMD).**  
Maneja la transmisión análoga de banda base entre nodos en el medio físico.
- **Protocolo de administración de estación (SMT).**  
Maneja la administración del anillo FDDI, incluye identificación, detección de errores, reconfiguración, gerenciamiento de estadísticas de tráfico y monitoreo.

FDDI puede ser implementado en dos formas básicas, como un anillo doble y como un anillo implementado mediante concentradores.

Esta tecnología define de forma específica los componentes requeridos para formar una red de este tipo, el estándar FDDI define los dispositivos que deben usarse, el que debe ser seguido por los proveedores de concentradores, puentes, controladores y equipos de cómputo que deseen ser incorporados a una red FDDI.

## **Ethernet.**

Ethernet fue creado originalmente por Xerox, Intel y DEC y estandarizado como IEEE 802.3 e ISO 8803, fue la primera tecnología de redes locales y ha sido el más usado. Existen más de **30 millones** de computadoras conectadas en redes que usan Ethernet que permite operar con rango de transmisión de 10 Mbps.

Su diseño está basado en la filosofía de protocolos estratificados, se encuentra dividido en dos estratos el de enlace y el físico, cada uno es una entidad autónoma e intercomplementaria. Su eficiencia es buena cuando al utilización del canal es relativamente baja.

- **Estrato de enlace:** proporciona la lógica que gobierna realmente el enlace, es independiente del medio de transmisión, lleva acabo labores de direccionamiento de frames y detección de errores, así como el encapsulamiento mismo de la información a nivel físico y la gestión de los enlaces.
- **Estrato físico:** es dependiente del medio y se encarga de introducir las señales al medio físico de transmisión, de proporcionarle el sincronismo adecuado y de la codificación y decodificación de datos.

Se fundamenta en el protocolo ALOHA, y fundamentado en una topología de bus, donde todos los nodos cuentan con la misma prioridad de transmisión, razón por la cual compiten por el bus común, cabiendo la posibilidad de existencia de colisiones, este acceso al canal o bus emplea la técnica "Uno persistente".

Ethernet emplea una dirección física de 48 bits, los 3 primeros bytes son asignados por el administrador local, la IEEE y los 3 últimos son asignados al adaptador de red por el fabricante, esta dirección es fija y única en el universo Ethernet y es representada por seis pares de números hexadecimales separador por dos puntos.

Los primeros estándares de Ethernet se basaban en cable coaxial que se conectaba directamente de máquina a máquina, en vez de hacerlo de un punto central a cada máquina. Este arreglo causo muchos problemas, ya que el fallo en un nodo afectaba a toda la red. Estos problemas de cableado fueron solucionados con la estandarización de la versión 10Mbps, que especifica repetidores centralizados y cableado de par trenzado (UTP) de los concentradores a cada computadora, de esta forma es posible aislar nodos detectados con comportamiento anormal.

Una red local Ethernet es generalmente llamada segmento y una red grande puede ser compuesta por múltiples segmentos, cada uno tiene muchas computadoras incorporadas al mismo.

La red principal o backbone de RedUNAM fue diseñada con la topología de doble anillo, implementado con FDDI y fibra óptica, que ofrece una velocidad de transmisión de 100 Mbps, el anillo lo conforman seis ruteadores que dan salida a las LAN mediante segmentos Ethernet, en caso de falla de alguno de los ruteadores se encuentran instalados puentes pasivos llamados "bypass" que cierran el acceso a los segmentos derivados de dicho ruteador, cerrando el anillo en ese punto y permitiendo el flujo hacia el siguiente ruteador, creando un anillo simple, en caso de doble falla son creados dos anillos aislados uno del otro, ejecutando cada uno un Token Ring propio.

La Universidad incorpora a sus Facultades, Institutos y dependencias dentro y fuera del campus universitarios a través de enlaces punto a punto, implementados con fibra óptica, RDI (Red Digital Integrada), microondas y enlaces satelitales, esta incorporación se lleva a cabo mediante enlaces punto a punto y diversos protocolos como IP, Netware, Windows para trabajo en grupo, Windows NT y Apple Talk de forma localizada.

Además de las dependencias en el campus universitarios, se interconectan la Dirección General de Preparatorias, las Escuelas Nacionales Preparatorias (ENP), los Colegios de Ciencias y Humanidades (CCH), las Escuelas Nacionales de Estudios Profesionales (ENEP) y las Facultades de Estudios Superiores (FES), además de los centros de desarrollo dependientes de la DGSCA como los de Máscarones, Pitágoras y Nuevo León.

Los enlaces dentro del Distrito Federal son implementados con microondas y RDI (que incorpora fibra óptica, microondas y enlaces satelitales) contratado a Teléfonos de México y en el caso de los enlaces con instituciones en el interior del país se emplea RDI .

Las instituciones externas que se enlazan a través de la RedUNAM en el Distrito Federal y las externas conectadas a RedUNAM en el interior del país son ennumeradas dentro del Apéndice 2.

En un futuro se planea la incorporación de nuevas tecnologías, dado el crecimiento de la demanda de los servicios y la complejidad de su implementación, lo que implica nuevas necesidades de transmisión de información. Estas tecnologías serán ATM (Modo Asíncrono de Transmisión) para implementar la red principal (backbone), Ethernet conmutado dedicado, que ofrecerá velocidades de transmisión de 10 Mbps de forma local y Ethernet Rápido (Fast Ethernet) que proporcionará un ancho de banda de 100 Mbps.

Con todo lo anterior podemos concluir que la UNAM se mantiene a la vanguardia en los Sistemas de Cómputo y Telecomunicaciones, conceptos antes separados, y hoy cada vez más estrechamente vinculados e interdependientes, siendo además parámetros inequívocos del avance y desarrollo de las instituciones y los Estados alrededor del mundo.

### **Las tecnologías ATM (Modo Asíncrono de transmisión), Ethernet conmutado y Ethernet Rápidos (Fast Ethernet).**

Muchas tecnologías propuestas prometen mejor desempeño para las redes en el futuro, estas promesas crean dificultades a los propietarios o responsables del desarrollo y mantenimiento de las redes hoy, se trata de evitar comprar nuevo equipo que pueda ser no estándar o se torne obsoleto cuando este se estandarice.

Una tecnología de actualización es el Ethernet conmutado que explota los estándares actuales para renovar el rendimiento del equipo de red ya instalado. Con el Ethernet conmutado puede implementarse mayor rendimiento hoy, con cambios mínimos en el equipo, guardando opciones de apertura para el mañana.

Otras tecnologías de actualización compiten en este sentido: Ethernet rápido (Fast Ethernet) y ATM. el primero representa una opción a corto plazo ya que su implementación solo requiere de cambios de adaptadores de red en los equipos y nuevos concentradores, siendo los cambios en la configuración mínimos, con los que el ancho de banda aumentaría a 100MBPS .

ATM (Modo Asíncrono de Transmisión) es una tecnología de conmutación y multiplexión basada en celdas, y diseñada con propósitos generales, es aplicable a las LAN y

tecnologías de redes privadas, provee también de conexiones virtuales permanentes o conmutadas, en general hace que una red entera pueda ser construída con ATM y sus capas de aplicación, conmutando y multiplexando para soportar un rango alto de servicios de todo tipo. ATM proporciona un ancho de banda variable de acuerdo a la demanda, a través del uso de SVC's (Conexiones virtuales conmutadas) soportando también accesos tipo LAN para el ancho de banda disponible normalmente.

Tanto Ethernet rápido como ATM son sujetos de actividades de estandarización, aunque algunos productos han sido sacados al mercado por distintos proveedores y pueden proporcionar mejoras en circunstancias específicas, especialmente en la red principal, brindando un mejor desempeño, menor costo y simplificación en la administración.

Las ventajas derivadas de su implementación pueden resumirse en las siguientes:

- Mejora de los tiempos de transmisión al ampliar los anchos de banda.
- Facilitarían la difusión de los servicios.
- Mayor seguridad y protección de la integridad de la información que circule por la red.
- Disminución de las labores administrativas y de mantenimiento.
- Disminución de costos.
- Mejor rendimiento de equipos y servicios.

Debido al acelerado desarrollo de los servicios de información que la red universitaria proporciona y su cada vez mayor sofisticación, que consume cada vez mayores cantidades de recursos de emisión y transmisión, es de suma importancia el implementar en un futuro cercano estas tecnologías.

### **1.3.3 Servicios de información que ofrece RedUNAM.**

Con el fin de promover el intercambio de ideas y conocimiento que colaboren al mejoramiento de la Universidad y México, la UNAM brinda opciones para el libre tránsito de estos entre las diversas instituciones que generan y transforman en conocimiento a lo largo del país, estas opciones son los servicios que ofrece la RedUNAM.

Los servicios que proporciona la Red Universitarias son:

- Acceso a los acervos bibliográficos de la Dirección General de bibliotecas, donde es posible la consulta en línea de los catálogos de ejemplares con que cuentan las bibliotecas de la UNAM, proporcionando información acerca de localización física del material, fichas bibliográficas y temás que comprenden.

- **Acceso a consultas hemerográficas y diversas agencias de noticias, así como a medios de información de ámbito internacional como periódicos y agencias informativas, permite consultar los periódicos nacionales de mayor circulación, así como la Gaceta UNAM, que son mostrados digitalmente, asemejando su apariencia habitual, permitiendo consultas por fecha y por sección, la consulta a las agencias noticiosas permite consultar los artículos y notas más relevantes ordenadas por artículo y fechas.**
- **Consulta de información organizada por temás, este servicio proporciona referencias a la información más relevante, disponible a lo largo del país en variados formatos, organizada y clasificada temáticamente.**
- **Correo electrónico, servicio de mensajes con gran volumen de usuarios, permite a los usuarios contar con una cuenta donde pueden recibir y enviar mensajes compuestos de todo tipo de datos hacia cualquier punto en la Internet.**
- **Acceso a un servidor de FTP de acceso anónimo, donde es posible obtener herramientas y aplicaciones de manera gratuita con fines de uso personal, dicho material es seleccionado y probado por personal capacitado, ántes de ser puesto a disposición general y su fin es el de enriquecer los sistemas mediante los que los usuarios accesan a los servicios de información.**
- **Impresión remota: este servicio cuenta con dos impresoras accesibles remotamente, y de características de alta calidad y volumen: una Xerox II 4700 en color y una HP IISI de 300 dpi.**
- **PhoneBook: libro telefónico que alberga la información más relevante del personal que labora en la UNAM y que cuenta con mecanismos de búsqueda que lo hacen de referencia rápida.**
- **Un servidor de WWW: permite accesar a los servicios de información a través de clientes de WWW, centraliza la información disponible de las distintos Facultades, Escuelas, Institutos y demás dependencias que conforman la UNAM.**
- **Acceso a bases de datos alrededor del mundo, a través del Gopher de la UNAM.**
- **Consultoría a usuarios de la red, vía personal, telefónica y correo electrónico, que es llevada a cabo por personal debidamente capacitado para este fin. Su objetivo es solucionar problemás acceso a los servicios y del uso de las aplicaciones y programás para consultarlos.**

En la *Figura 1.3.3* se ilustra el crecimiento del empleo de los servicios de WWW, Gopher y Correo electrónico, describiendo el acelerado crecimiento que se disparó en el año de 1995 en estos servicios ofrecidos por RedUNAM.

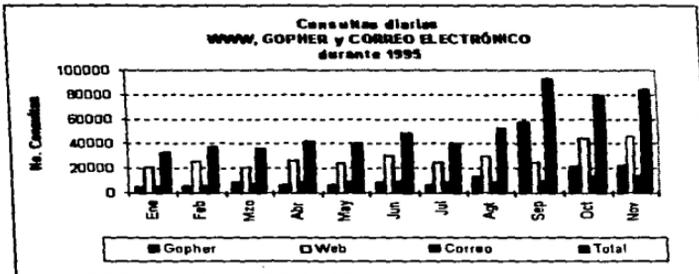


Figura 1.3.3.

Volumen de accesos diarios a los servicios de RedUNAM durante 1995.

### 1.3.4 La Coordinación de servicios de Red (CSR) .

#### Descripción, infraestructura y funciones.

La Coordinación de Servicios de Red depende de la dirección de Telecomunicaciones Digitales de la DGSCA.

Es en este lugar donde se lleva a cabo la administración de los servicios de información que ofrece RedUNAM, así como la integración de la mayoría de ellos .

Para su operación cuenta con estaciones de trabajo Unix, terminales X y PC's con ambiente X y total conectividad . La descripción y finalidad de cada uno de sus equipos se detalla en el Apéndice 1.

La CSR se encuentra dentro del edificio de la DGSCA y está conectada a la sala de operación, recinto físico de los equipos descritos en el apéndice 1, a través de par trenzado desde sus máquinas locales (estaciones de trabajo Unix, terminales X y PC's) a un concentrador interconectado por la misma vía a los ruteadores correspondientes a los segmentos que albergan a los equipos bajo su responsabilidad, y que soportan los servicios de información (Véase apéndice 1).

### **1.3.5 Objetivo de la Investigación.**

Para concluir este capítulo se definirá el objetivo del desarrollo de la presente investigación.

**Objetivo:**

- Proponer un nuevo esquema de Administración y Control de los servicios de RedUNAM que son dirigidos al sistema de bachillerato, actualmente en función y que debido a su acelerado crecimiento en cuanto al número de usuarios requiere de modificaciones que incrementen en rendimiento, tanto del equipo disponible como de los servicios que se ofrecen.

El esquema propuesto incluirá las siguientes aspectos :

- Diseño
- Implementación
- Mantenimiento y
- Consideraciones a futuro.

Lo anterior a fin de dar alternativa de solución a la problemática actual de este servicio en particular, que será definida y analizada capítulos siguientes.

---

## CAPITULO I I

### CONTEXTO DEL PROYECTO BACHILLER

#### 2.1 Introducción

El proyecto Bachiller fue implantado en 1995 con el fin de proporcionar los servicios de información en red básicos a la comunidad universitaria en su nivel bachillerato.

Se implementó una infraestructura para soportar los servicios de correo electrónico, gopher, listas de correo e IRC, en el caso de los tres últimos se implementó la estructura únicamente, y en el caso del correo se implementó todo el sistema.

Este proyecto se puso en marcha, y actualmente funciona accedando a la computadora identificada como `bachiller.dgsca.unam.mx`, es ahí donde residen físicamente las cuentas de correo, el gopher, las listas y el IRC.

Estos servicios son administrados en la Coordinación de Servicios de Red, el correo en su esquema completo, y en el caso de los restantes se mantiene su estructura básica, dejándole a los usuarios indicados mediante los canales específicos definidos la alimentación de información a los restantes servicios.

El servicio de mayor importancia actualmente es el de correo electrónico con 550 usuarios, los restantes servicios no son aprovechados en la proporción que deberían, de acuerdo al volumen de la comunidad del bachillerato. Actualmente se encuentran subutilizados, en relación al equipo con el que se cuenta, que es una Sparc Classic de Sun Microsystems con sistema operativo tipo BSD SunOS 4.1.3.

#### 2.2 Problemática.

##### 2.2.1. Situación actual de los servicios.

Los servicios residen en la computadora identificada como `bachiller.unam.mx`, y han sido implementados a partir de GNU's, que son programas servidores de cada uno de los servicios que en este caso tratamos, disponible en internet como software de carácter público, las versiones de cada servidor son :

Gopher	GNU Gopher Versión 2.016
Listas de correo	Lists Versión 1.93
IRC(Internet Relay Chat)	IRC Versión 1.02
Correo electrónico	Send Mail

- **Correo electrónico.**

Cuenta actualmente con 550 usuarios, las claves asignadas tienen las siguientes características:

- Acceso a consulta de correo únicamente mediante programás cliente.
- Acceso a transferencia de archivos únicamente mediante programás cliente.
- No permiten acceso vía sesión remota (telnet).
- Tiempo de vida de la clave ilimitado.
- No permite al tenedor el cambio de password.
- Cuota de espacio asignada 700 KB.

### **2.2.2 Procedimientos administrativos actuales.**

Dentro de los procedimientos administrativos actualmente empleados, mediante los que se llevan a cabo las actividades involucradas con los servicios se distinguen los siguientes:

- Obtención de claves de acceso vía telefónica a RedUNAM.
- Obtención de claves de correo electrónico.
- Alimentación de información al servicio de Gopher.
- Generación de listas de correo.
- Generación de canales para los foros de discusión.

#### **Obtención de claves de acceso vía telefónica a RedUNAM.**

1. Un usuario (alumno, profesor o administrativo del sistema del bachillerato de la UNAM) solicita su clave en el plantel que le corresponda.
  2. Lleva a cabo el pago correspondiente mediante los canales definidos a ese fin.
  3. Obtiene la solicitud del servicio en el Departamento de atención a usuarios de la DGSCA.
  4. Recaba autorización de su jefe inmediato en el caso de profesores o administrativos, avalando su solicitud mediante una carta.
  5. La solicitud llenada debidamente es entregada en el departamento en el que se solicitó.
  6. El Departamento de atención a usuarios canaliza las solicitudes aprobadas para generar las claves.
  7. La clave es creada en la CSR, generándose login y password, el cual es proporcionado al usuario vía telefónica o fax.
8. El usuario emplea el servicio.

**Generación de canales en IRC.**

1. Se genera la estructura necesaria para la implementación del servicio de canales IRC.
2. El usuario presenta la solicitud explícita de creación de un canal de IRC al administrador del servicio.
3. Se determina la viabilidad de la propuesta de acuerdo a las políticas establecidas.
4. La solicitud es aprobada.
5. El canal puede ser creado.
6. El servicio es empleado por los usuarios.
7. El solicitante del canal se encarga de su administración y actúa en él como moderador.

**2.3 Análisis.**

Como resultado del análisis de la situación actual del servicio que proporciona bachiller, los procedimientos administrativos que implica su funcionamiento y su contexto en general es posible reducir la problemática a una expresión.

Inadecuado aprovechamiento de la infraestructura de servicio implementada en Bachiller, destinada a satisfacer las necesidades de información de la comunidad universitaria en su sistema de bachillerato.

El análisis se comenzará identificando los síntomas y efectos en los aspectos más importantes como son el costo, el tiempo, la seguridad, el empleo de recursos humanos y el aprovechamiento del equipo, con el fin de localizar las causas y proponer nuevos mecanismos que permitan el establecimiento de un esquema más eficiente.

Efectos	Costo	Tiempo	Seguridad	Recursos humanos	Equipo
Tiempo excesivo para completar trámite de solicitud de clave.	X	X		X	
Retraso en la actualización y modificación de la información de los servicios.	X	X		X	
Servicios con información no actualizada, incompleta e inoportuna.	X		X		X
Inconsistente método de transmisión de datos confidenciales al usuario.		X	X	X	
Mínimo aprovechamiento de la infraestructura de los servicios.	X			X	X

Como podemos apreciar en la tabla anterior, existen efectos que inciden directamente en los aspectos más importantes de un servicio, como la rapidéz y la eficiencia, y de los cuales podemos identificar los recursos y costos que se ven afectados, de esta manera podemos iniciar un análisis de la problemática mas detallado.

### **2.3.1. Causas y efectos.**

Tomando en consideración la problemática definida, es posible determinar la causas más importantes en el actual funcionamiento de los servicios de Bachiller.

- Inexactitud en los procedimientos involucrados con el sistema.

Este punto es susceptible de ser mejorado con la implementación de nuevos y reestructurados procedimientos, que reduzcan al mínimo necesario todo tipo de trámite administrativo. Asimismo permitan establecer un esquema sencillo y eficiente de las labores propias de la administración del servicio propiamente.

- Demasiadas funciones administrativas.

Lo que provoca duplicidad de funciones, información inconsistente y mayor demora al ser necesarias más conciliaciones en la información.

Este punto debe mejorarse simplificando y automatizando el mayor número de actividades administrativas asociadas al mantenimiento del servicio de información, lo que reduciría el número de personal asignado, mejorando tiempos de respuesta y disposición inmediata del servicio, dándole una mayor consistencia y eficiencia.

- Comunicación inadecuada.

En cuanto a lo institucional se refiere a la desinformación de la existencia de este sistema, en el aspecto administrativo del servicio es derivada del involucramiento de un mayor número de personas del necesario, en todo tipo de labores asociadas al funcionamiento del sistema de información, lo que aumenta el número de conciliaciones de información que a su vez provoca duplicidad de funciones.

Esto es posible eliminando y definiendo claramente los canales de comunicación necesarios entre los administradores del servicio, flexibilizando además los procedimientos que involucran trámites para la obtención de los servicios, estos canales deben ser los más sencillos posible y ser reducidos en número, para agilizar la toma de decisiones y aplicación de criterios durante el funcionamiento del sistema de información.

- **Deficiente retroalimentación de los servicios de información.**

**Provocada por un innecesario procedimiento que resulta poco práctico, que dificulta la incorporación y renovación de información a los servicios de información.**

**Es posible mejorar notablemente este punto, descentralizando este tipo de actividad, automatizándola al máximo para hacer lo más sencillo las modificaciones en la información dándole un carácter de oportuna.**

- **Desinterés por el uso de los servicios que ofrece este sistema de información.**

**Provocada conjuntamente por las causas antes descritas, lo que ha ido en aumento, al dejar de cumplir este sistema de información con su función básica de difundir información de interés de forma oportuna, consistente y completa.**

**Asimismo podemos identificar las ventajas y desventajas del sistema actual, para puntualizar las principales características positivas y negativas involucradas en el funcionamiento del sistema de información Bachiller que lo han llevado al estado en el que actualmente se encuentra.**

#### **Desventajas:**

- **Centralización de la incorporación de información a los servicios de información.**
- **Mínima actualización de la información proporcionada por este servicio.**
- **Desinterés en el empleo del sistema de información.**
- **Poca participación del Bachillerato en general en el funcionamiento de este servicio.**
- **Información disponible inconsistente, inoportuna e incompleta.**
- **Desaprovechamiento de la infraestructura dispuesta para el funcionamiento de este sistema.**
- **Centralización de trámites y actividades de tipo administrativo.**
- **Empleo inadecuado de los sistemas de archivos dispuestos para el sistema.**
- **Subutilización de las estructuras existentes para albergar los distintos servicios de información, especialmente gopher, listas y los foros.**
- **Carencia de servicios de información con interfase gráfica.**

#### **Ventajas:**

- **Centralización de la administración de los servicios de información lo que ofrece total control sobre la información en el depositada.**
- **Facilidad de su administración, ya que es realizada por una sola persona que domina el esquema de información en su totalidad.**

- Mantenimiento mínimo, al no existir modificaciones constantes a la información de los servicios.
- Acceso únicamente mediante usuarios, lo que reduce la carga de proceso al equipo que alberga este servicio, ofreciendo así un tiempo de respuesta menor al usuario.

## **2.4 Propuesta de solución.**

Con el conjunto de características que actualmente presenta el sistema de información Bachiller, que han sido descritas a lo largo de este capítulo, es posible concluir que resulta inaplazable una reestructuración del esquema en su totalidad, con el fin explícito de mejorar su eficacia y rendimiento, así como aumentar el aprovechamiento de los recursos existentes, lo que debe redundar positivamente en mejoras tales como:

- Cumplimiento adecuado de la función de proporcionar información de interés, relevante de una forma oportuna y completa.
- Simplificación de procedimientos administrativos.
- Proporcionará una mayor seguridad en el manejo de información confidencial de manejo de claves de usuario.
- Aumento en el desempeño del servicio en general, proporcionando mejores tiempos de respuesta, mayor eficiencia y una mayor consistencia.
- Ampliación y actualización de los servicios actualmente prestados, incorporándose al sistema de información los siguientes servicios:
  - ◊ Correo electrónico.
  - ◊ Gopher.
  - ◊ Listas de correo.
  - ◊ WWW
- Administración descentralizada de los servicios mediante una interfase gráfica de fácil manejo y efectividad.

La estrategia que se propone como fin de esta investigación tiene un carácter participativo por parte del personal de las instituciones del bachillerato, pretendiendo involucrar a cada plantel en la obtención de información generada en su plantel que se adecúe a los lineamientos que para ser incorporada al sistema Bachiller serán definidas, destacando la función de difundir entre la comunidad del bachillerato de la UNAM la información competente a cada plantel, así como al bachillerato en general, el primer caso se pretende dar a conocer mediante medios electrónicos de información de gran relevancia e interés como calendarios de exámenes, eventos, notificaciones diversas, planta docente, temarios y en general información de relevancia. En el caso de bachillerato en su conjunto será información análoga salvo que será de interés a todos

los planteles que lo integran y generada hacia el bachillerato como parte del funcionamiento de la Universidad.

La implementación de este proyecto propuesta consta de una serie de etapas caracterizadas a continuación.

**1a. Etapa.**

Conformación de la estructura logística del sistema de información que sustituirá al sistema que actualmente opera, incluye proporcionar los elementos de software necesarios (programas servidores, interfases y mecanismos diversos), es decir, la preparación de la estructura base que sostendrá el funcionamiento y mantenimiento de información con que contará, así como la definición explícita y formal de los procedimientos de carácter administrativo que van asociados al funcionamiento del sistema.

**2a. Etapa.**

Selección e incorporación del personal que se encargará del mantenimiento y supervisión del sistema, a fin de familiarizarlos con el ambiente mediante el cual llevarán a cabo las labores de mantenimiento, que son proporcionadas por el mismo sistema.

**3a. Etapa.**

Liberación del servicio, fase en la cual los servicios que ofrece el sistema pueden ser empleados, y tras la cual serán llevadas de forma ininterrumpida las labores de mantenimiento y actualización de los servicios, así como la incorporación de usuarios, produciendo además mediante el aprovechamiento de las características de los mismos servicios, una difusión de las ventajas que ofrece la utilización del sistema para la comunidad universitaria.

Caracterizando el esquema propuesto, estará basado en una administración de sistema y servicios de forma descentralizada, contará con un administrador principal y administradores locales como planteles sean incorporados al esquema en su conjunto, estos últimos deberán contar con elementos básicos de administración de servicios, sin mayor especialización, y se encargarán de la obtención y generación de la información relevante en cada uno de sus planteles, con los formatos específicos que para cada servicio sean requeridos y especificados, lo cual mediante interfases sencillas e ilustrativas permitirán su incorporación desde sus propios planteles vía remota. Podrán además realizar la inscripción de usuarios de la institución mediante la misma vía, reduciendo al mínimo los trámites administrativos y facilitando la obtención de los requisitos necesarios para este fin.

Las actividades derivadas de las labores de los administradores locales serán minitoreadas por el administrador principal, que además se encargará del mantenimiento de la estructura básica que soptra al al servicio en su totalidad.

De esta manera se propone un aparato de manejo sencillo , flexible y de fácil implementación, a fin de dar a este sistema y a los servicios que lo componen un alto grado de eficiencia, rendimiento y actualidad, para cumplir de la mejor manera su principal función, la de difundir información de interés y relevancia a las instituciones del bachillerato con una adecuada presentación, rapidez, consistencia y oportunidad.

## CAPÍTULO I I I

### DESARROLLO DE LA PROPUESTA AL PROYECTO BACHILLER

Durante los capítulos anteriores se ha definido el objetivo del desarrollo de la presente Tesis, de la misma forma se ha analizado la situación actual del Sistema Bachiller, destinado a proporcionar servicios de información a la comunidad universitaria en el nivel bachillerato.

Es en este capítulo donde se fundamenta formalmente la propuesta de solución a la problemática, presente en el sistema descrito y en actual operación, esta propuesta está basada principalmente en la técnica de modelado conocida como Análisis Estructurado.

Caracterizando el esquema propuesto, estará basado en una administración de sistema y servicios de forma descentralizada, contará con un administrador principal y tantos administradores locales como planteles sean incorporados al esquema en su conjunto, estos últimos deberán contar con elementos básicos de administración de servicios, sin mayor especialización, y se encargarán de la obtención y generación de la información relevante en cada uno de sus planteles, con los formatos específicos que para cada servicio sean requeridos y especificados, lo cual mediante interfases sencillas e ilustrativas permitirán su incorporación desde sus propios planteles vía remota. Podrán además realizar la inscripción de usuarios de la institución mediante la misma vía, reduciendo al mínimo los trámites administrativos y facilitando la obtención de los requisitos necesarios para este fin.

Las actividades derivadas de las labores de los administradores locales serán monitoreadas por el administrador principal, que además se encargará del mantenimiento de la estructura básica que sostendrá todo el servicio.

#### 3.1 Objetivos.

De esta manera los procedimientos y mecanismos propuestos tendrán como objetivos principales los siguientes:

- Integrar un sistema de fácil manejo, flexible y de fácil implementación.
- Dar a este sistema y a los servicios que los componen un alto grado de eficiencia, rendimiento y vigencia.
- Cumplir de la mejor manera su principal función, la de difundir información de interés y relevancia a las instituciones del bachillerato.
- Ofrecer una adecuada presentación, corto tiempo de respuesta, consistencia y oportunidad.

### **3.2 Metodología.**

Para la elaboración de una documentación formal, en la cual se plasme el análisis del sistema propuesto, que sea útil para la adecuada y completa documentación del mismo, se empleó la metodología Yourdon: el análisis estructurado, útil en la ilustración clara y detallada de los procedimientos y esquemas que formarán parte de la propuesta, esta metodología fue elegida de entre otras debido a que el sistema que formará parte de la propuesta global, no implica mayor complejidad, por el contrario de su sencillez se deriva su principal característica, y es está técnica la que resulta eficiente para una correcta y completa formalización y documentación del sistema objeto de esta investigación.

### **3.3 Especificación estructurada.**

La metodología de Análisis Estructurado implica la elaboración de una serie de elementos, en el entendido que se conoce a detalle la forma de operación del sistema anterior, los elementos y variables que se involucran, en este caso del actual sistema de operación de los servicios del Bachillerato, y se tiene idea clara de lo que con la nueva estructura se pretende, así como de los nuevos elementos y procedimientos que se incorporaran, es el caso del objeto de la presente investigación.

Los elementos de esta metodología que se han tomado para modelar los nuevos procedimientos, operaciones y comportamiento del sistema propuesto son los siguientes:

### 3.3.1 Diagrama Entidad - Relación ( DER ).

Es un esquema donde se identifican las entidades que participan en el funcionamiento del sistema, el tipo de relación que se entabla y la cardinalidad de las mismas, esto con el fin de identificar las interdependencias funcionales de los elementos que conforman el esquema en su totalidad.

La simbología especial usada para la elaboración de estos diagramas es la mostrada en el siguiente cuadro :

1, N, M	cardinalidad de la relación
	Entidad
	relacion
	origen de la relación
	unión de relación y entidades
	subconjunto de la relación

A continuación se muestra el diagrama Entidad-Relación con que se inicia el modelado de la propuesta del Sistema de Control para Sistemás de Información del Servidor para el Bachillerato de la UNAM.



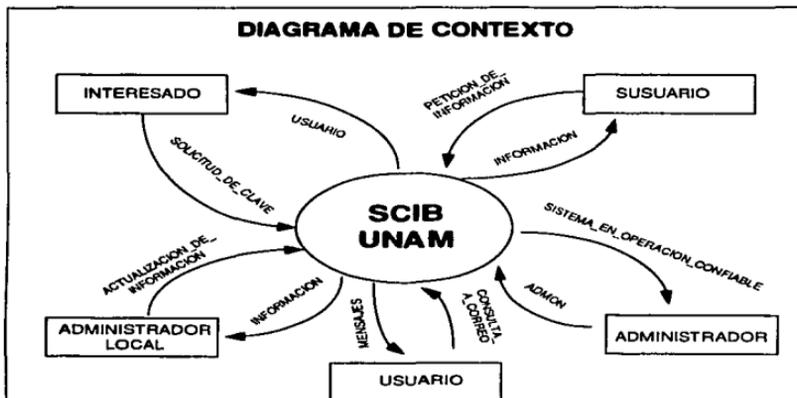
**Figura 3.1 Diagrama Entidad - Relación  
Propuesto para el proyecto SCSIB-UNAM.**

### 3.3.2 Diagrama de Contexto del sistema.

En el diagrama de contexto del sistema se presentan todas las entidades externas al sistema, que envían o reciben datos del mismo, y para los cuales este tiene una respuesta, estos son identificados claramente con nombres nemónicos específicamente definidos a este fin.

A partir de este esquema se inicia un proceso de descomposición a fin de obtener niveles de mayor detalle. Este proceso continúa hasta llegar a un nivel primitivo o terminal, donde de forma clara es posible describir las operaciones.

A continuación se presenta el diagrama de contexto correspondiente al SCSIB-UNAM:



**Figura 3.2 Diagrama de Contexto  
Propuesta para el proyecto SCSIB-UNAM.**

### 3.3.3 Diagramas de flujo de datos (DFD's).

Para poder describir gráficamente de forma clara el conjunto de operaciones que conformarán el sistema propuesto, nos auxiliaremos de herramientas gráficas que nos proporciona la técnica de Análisis Estructurado, antes comentada, estas son los diagramas de flujo de datos, conocidos también como DFD's, que como sabemos, son una herramienta fácil de elaborar y comprender, por medio de la cual es posible representar los componentes que intervendrán en el sistema desde el punto de vista de los datos.

Las ventajas del uso de esta herramienta, radican principalmente en permitir al analista plasmar el sistema global hasta el nivel de detalle que desee manejar.

Para la elaboración de un diagrama de flujo de datos (DFD) es necesario hacer uso de una simbología especial, compuesta de los siguientes elementos:

<b>Símbolo</b>	<b>Función</b>
	Entidad fuente o destino. Origen o recepción de datos.
	Proceso, o transformación de datos.
	Archivo o Base de datos.
	Flujo de datos.

La elaboración de los diagramas nos aseguran la comprensión de las operaciones que en conjunto conformarán el sistema, de acuerdo a diversos niveles de detalle, en el caso del sistema propuesto emplearemos los siguientes niveles de abstracción.

- Diagrama de flujo de datos. Nivel 0.

Es en este diagrama donde se muestran de forma global todos los procesos principales, los flujos de insumos, salidas y archivos de los procesos hacia el sistema y los mismos con que este responde a los primeros .

- Diagrama de flujo de datos. Nivel 1.

En los diagramas de este nivel se detallan de forma particular cada uno de los procesos principales identificados como parte del sistema global. El nivel de detalle es general para cada uno de los procesos en particular. En ellos se ilustran los insumos y salidas principalmente identificadas de los principales procesos que componen el sistema, de la misma forma, se detallan las operaciones que cada uno de estos procesos involucran.

- Diagramas de flujo de datos. Nivel 2.

Ilustran el detalle de las operaciones que conforman los diagramas de Nivel 1, hasta el grado de detalle que permita su división en suboperaciones.

En el caso particular del esquema propuesto, el Nivel 2 fue el último en emplearse, pero generalmente estos niveles aumentan en grado de detalle hasta que su grado

de complejidad lo permita, llegando hasta identificar operaciones no divisibles o primitivas.

Cabe señalar que hasta este punto es imprescindible que las entradas y salidas, así como el flujo de datos identificados a partir del diagrama general (Nivel 0), deben corresponder exactamente en los siguientes niveles de detalle, es decir deben los DFD's estar balanceados, lo que dará consistencia a la representación gráfica de todos los procesos que integran el sistema en su totalidad.

A continuación se presenta la serie de diagramas de flujo de datos que ilustran la propuesta del sistema SCSIB-UNAM:

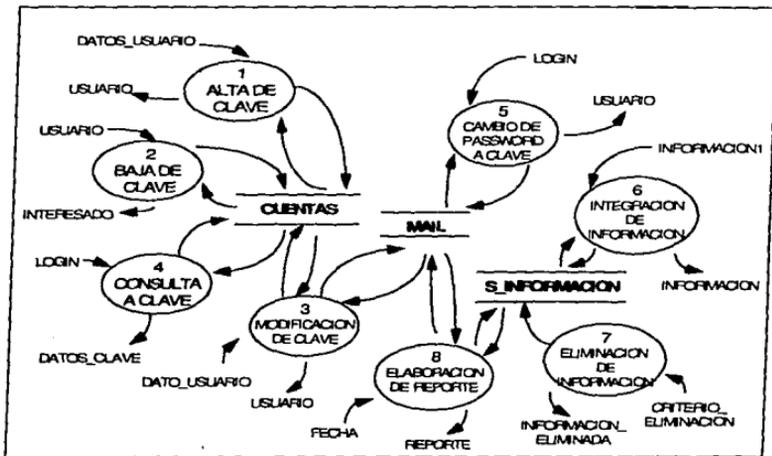


Figura 3.3 DFD Nivel 0 SCSIB-UNAM.

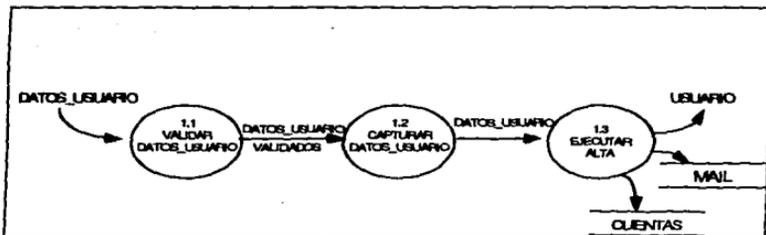


Figura 3.4 DFD1 Nivel 1 Alta de clave.

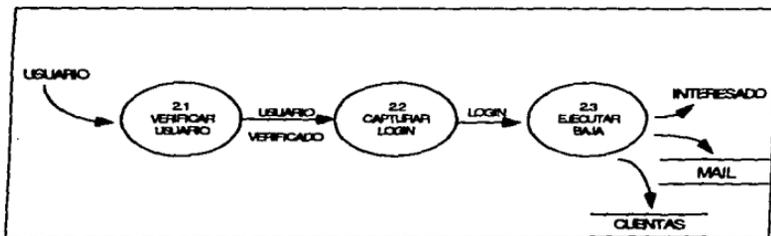


Figura 3.5 DFD2 Nivel 1 Baja de clave.

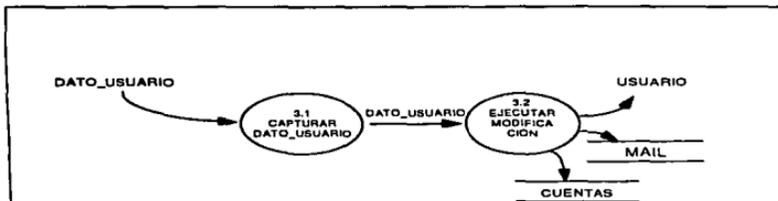


Figura 3.6 DFD3 Nivel 1 Modificación a clave.

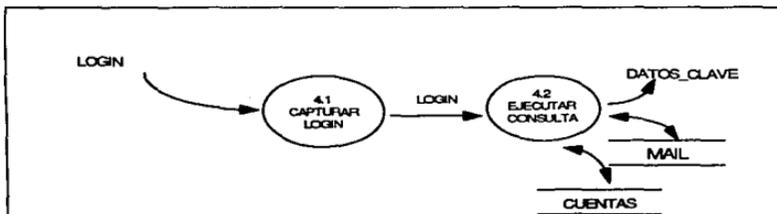


Figura 3.7 DFD4 Nivel 1 Consulta a cuenta.



Figura 3.8 DFD5 Nivel 1 Cambio de password.

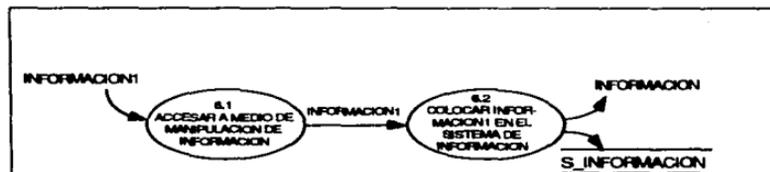


Figura 3.9 DFD6 Nivel 1 Integración de Información.

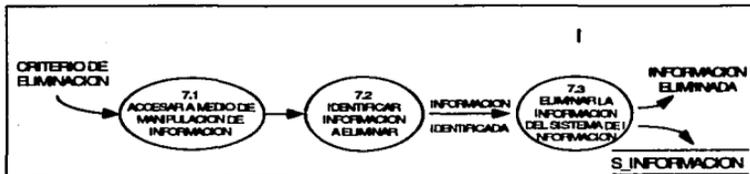


Figura 3.10 DFD7 Nivel 1 Eliminación de la información.

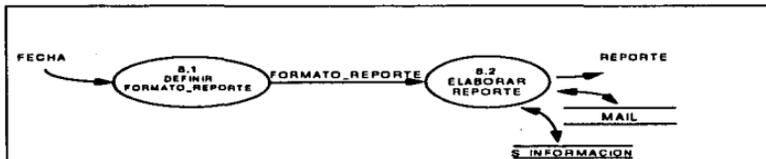


Figura 3.11 DFD8 Nivel 1 Elaboración de reportes.

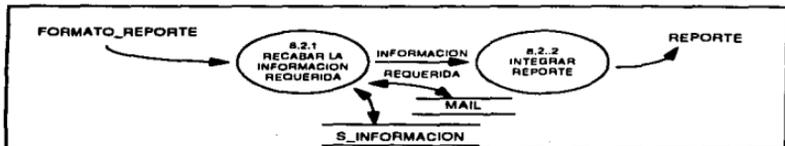


Figura 3.12 DFD9 Nivel 2 Elaborar reporte.

### **3.4 Lenguaje Estructurado.**

Además de las herramientas gráficas aplicables para la descripción del sistema que nos ocupa, podemos hacer uso de una herramienta de tipo textual, por medio de la cual se describen en forma precisa las transformaciones que sufrirán los datos al pasar por un proceso. Para cada proceso primitivo del sistema, es decir, aquel que ya no requiere ser desagregado, para cada uno de estos debe existir una descripción de lo que el realizará con los flujos de datos que lo alimentan, a esto le llamamos miniespecificación.

Desagregar los procesos principales del sistema hasta obtener un nivel de detalle que permita conocer lo que cada proceso realizará, es una gran ayuda a los analistas, sin embargo, la desagregación por sí sola, no permite especificar que hará cada proceso con los flujos de datos que lo alimentan. Por esta razón, después de haber finalizado el proceso de desagregación y observar que existe un conjunto de procesos primitivos o terminales en el sistema, fue necesario describir en forma precisa lo que el proceso realizará.

#### **3.4.1 Miniespecificaciones.**

Son las descripciones precisas del procedimiento para realizar las operaciones primitivas del sistema; se realizan en forma de algoritmo, empleando los nombres mnemónicos específicamente definidos a partir del diagrama de contexto y nivel 0, y que son empleados en la serie de diagramación de procesos. Estos se encuentran explícitamente definidos en el Diccionario de Datos.

La terminología que forma las especificaciones se compone de:

- **Verbos:** realizan funciones específicas.
- **Elementos de datos, archivos y flujos de datos compuestos:** contenidos en el diccionario de datos, son representados con letras mayúsculas. Cuando exceden una palabra son ligados mediante un guión.
- **Numeración:** Se escribe del lado izquierdo del título del proceso primitivo, e indica subordinación y secuencia en la ocurrencia de procesos.

A continuación se muestran las miniespecificaciones de los procesos terminales que componen el SCSIB-UNAM. **Miniespecificaciones.**

<b>1.1</b>	<b>Validar DATOS_USUARIO</b>
	Obtener DATOS_USUARIO Recabar COMPROBANTE_DE_INSCRIPCION o AUTORIZACION_DE_SUPERIOR Verificar DATOS_USUARIO Verificar COMPROBANTE_DE_INSCRIPCION o AUTORIZACION_DE_SUPERIOR

<b>1.2</b>	<b>Capturar DATOS_USUARIO</b>
	Obtener DATOS_USUARIO

<b>1.3</b>	<b>Ejecutar ALTA</b>
	Incorporar DATOS_USUARIO a MAIL y a CUENTAS

<b>2.1</b>	<b>Verificar USUARIO</b>
	Obtener DATOS_USUARIO Comprobar que USUARIO es quien debe ser eliminado de MAIL y CUENTAS

<b>2.2</b>	<b>Capturar LOGIN</b>
	Obtener LOGIN de USUARIO

<b>2.3</b>	<b>Ejecutar ALTA</b>
	Eliminar información correspondiente a LOGIN en CUENTAS

<b>3.1</b>	<b>Capturar DATO_USUARIO</b>
	Obtener DATO_USUARIO
<b>3.2</b>	<b>Ejecutar modificación.</b>
	Sustituir con DATO_USUARIO su correspondiente en CUENTAS Sustituir con DATO_USUARIO su correspondiente en MAIL
<b>4.1</b>	<b>Capturar LOGIN</b>
	Obtener LOGIN
<b>4.2</b>	<b>Ejecutar consulta.</b>
	Obtener DATOS_CLAVE de CUENTAS y MAIL
<b>5.1</b>	<b>Generar PASSWORD</b>
	Realizar procedimiento para generar un PASSWORD
<b>5.2</b>	<b>Sustituir PASSWORD</b>
	Sustituir con NUEVO_PASSWORD el PASSWORD correspondiente en CUENTAS Sustituir con NUEVO_PASSWORD el PASSWORD correspondiente en AIL
<b>6.1</b>	<b>Accesar a MEDIO_DE_MANIPULACION de información.</b>
	Lograr acceso al mecanismo con que cuenta el.SCSIB-UNAM para manipular la información del S_INFORMACION
<b>6.2</b>	<b>Colocar INFORMACION1 en el S_INFORMACION</b>
	Incorporar INFORMACION1 en el S_INFORMACION.

<b>7.1</b>	<b>Accesar a MEDIO DE MANIPULACION de información</b>
	Lograr acceso al mecanismo con que cuenta el SCSIB-UNAM para manipular la información del S_INFORMACION
<b>7.2</b>	<b>Identificar la INFORMACION a eliminar.</b>
	Ubicar en S_INFORMACION la INFORMACION sujeta a ser eliminada
<b>8.1</b>	<b>Definir FORMATO REPORTE.</b>
	Identificar el FORMATO_DE_REPORTE necesario segun se requiera
<b>8.2.1</b>	<b>Recabar la INFORMACION requerida.</b>
	Obtener la INFORMACION correspondiente al FORMATO_REPORTE y a la FECHA
<b>8.2.2</b>	<b>Integrar REPORTE</b>
	Integrar el FORMATO_REPORTE correspondiente con INFORMACION_REQUERIDA obtenida de MAIL y S_INFORMACION Imprimir REPORTE

### 3.4.2 Diccionario de Datos (DD).

La totalidad de los flujos de datos y nombres de archivos ilustrados en la serie de diagramas ya descrita, deben contar con un identificador de tipo nemónico único, estos identificadores deben ser empleados de forma consistente a lo largo de la diagramación para evitar cualquier tipo de ambigüedad. La serie de términos identificadores componen un diccionario, donde se especifican de forma precisa los elementos que componen a cada término.

La simbología de los operadores lógicos utilizados en las definiciones del Diccionario de Datos es la que se muestra a continuación:

<u>Operador</u>	<u>Significado</u>
=	Equivalencia de los elementos encontrados del lado derecho e izquierdo del operador.
+	Conjunción
[ ]	Disyunción .Selección de una sola de las operaciones contenidas en los corchetes. Cada opción puede estar formada por uno o más datos.
{ }	Iteración de los elementos encerrados entre llaves. Los números antes y después de las llaves definen el número de iteraciones deseadas.
	Separan los elementos objetos de una disyunción en el conjunto expresado entre corchetes.
• •	Delimitadores de comentario de tipo textual.

A continuación se muestra el Diccionario de Datos correspondiente al SCSIB-UNAM.

### DICCIONARIO DE DATOS

CONSULTA_A_CORREO=	*Acción del USUARIO con el fin de obtener sus MENSAJES *
ACTUALIZACION_DE_INFORMACION=	* Acción de un ADMINISTRADOR_LOCAL con el fin de sustituir INFORMACION con INFORMACION1 en el SCSIB-UNAM o modificar información relativa a los USUARIO que se encuentra en MAIL y CUENTAS *
ADMINISTRACION =	* Conjunto de actividades de tipo administrativo que lleva a cabo el ADMINISTRADOR a fin de supervisar las acciones de los ADMINISTRADORES_LOCALES para mantener y asegurar el adecuado funcionamiento del SCSIB-UNAM *
ADMINISTRADOR =	* Persona responsable del mantenimiento del SCSIB-UNAM en estado de SISTEMA_EN_OPERACION_CONFIABLE *
ADMINISTRADOR_LOCAL =	* Persona única por PLANTEL que interactúa directamente con el SCSIB-UNAM con el fin de manipular la información relativa a los USUARIO del PLANTEL al que pertenece *
ALUMNO =	* Alumno inscrito en algún plantel del Bachillerato de la UNAM *
AÑO =	1{0..9}2
CAUSA =	* Consideración del ADMINISTRADOR_LOCAL para que determinada INFORMACION deba ser sustituida *

**CLAVE =** \* Cuenta de correo electrónico correspondiente a un USUARIO EN EL SCSIB-UNAM \*

**CRITERIO\_ELIMINACION =** [ FECHA | CAUSA ]

**CUENTAS =** \* Alias de la base de datos que contiene la información de los USUARIO \*

**CUOTA\_EMPLEADA =** \* Espacio ocupado por una CLAVE en MAIL \*

**DATO\_USUARIO =** NOMBRE+ [ NO\_CTA | SUPERIOR ]  
PLANTEL + TELEFONO + TIPO

**DATOS\_CLAVE =** DATOS\_USUARIO +  
CUOTA\_EMPLEADA +  
ULTIMO\_ACCESO

**DATOS\_USUARIO =** NOMBRE + [NO\_CTA [SUPERIOR] +  
PLANTEL + TELEFONO + TIPO

**DATO\_USUARIO =** NOMBRE | [NO\_CTA [SUPERIOR] |  
PLANTEL | TELEFONO | TIPO |  
PASSWORD

**DB\_GOPHER =** \*Alias de la base de datos de servicio Gopher en el SCSIB-UNAM \*

**DB\_LISTAS =** \*Alias de la base de datos de servicio Listas en el SCSIB-UNAM \*

**DB\_WWW =** \*Alias de la base de datos de servicio WWW en el SCSIB-UNAM \*

**DIA =** 1{0..9} 31

**FECHA =** DIA + MES+ AÑO

**FORMATO\_REPORTE =** \* tipo de reporte a elegir, en este caso se dispone de consulta a una CLAVE \*

**INFORMACION =** \* Información disponible en el S\_INFORMACION mediante el acceso a SCSIB-UNAM \*

INFORMACION_ELIMINADA =	* INFORMACION retirada del S_INFORMACION debido a una CAUSA
INFORMACION1 =	* Información debidamente autorizada y formateada para ser incorporada al S_INFORMACION Y convertirse en INFORMACION *
INTERESADO =	[ PROFESOR   ALUMNO ]
LOGIN =	*Identificador único correspondiente a un USUARIO*
MAIL =	*Alias del sistema de archivos del servicio de correo electrónico del SCSIB-UNAM *
MATERNO =	*Apellido MATERNO del USUARIO *
MENSAJES =	* Conjunto de mensajes electrónicos destinados a un USUARIO *
MES =	1{0..9}12
NO_CTA =	1{0..9}7 + (" ") +1{0..9}1
NOMBRE =	PATERNO + MATERNO + NOMBRES
NOMBRES =	* Nombre(s) de pila del USUARIO *
PASSWORD =	*Contraseña confidencial con que cuenta un USUARIO para acceder a su CUENTA *
PATERNO =	*Apellido paterno del USUARIO *
PETICION_DE_INFORMACION =	*Acción de un SUSUARIO tendiente a obtener INFORMACION de su interés en el SCIB-UNAM *
PLANTEL =	* Denominación oficial del Plantel al que pertenece el USUARIO *
PROFESOR =	* Personal docente del Bachillerato de la UNAM*

<b>PUBLICO =</b>	* Persona que mediante algún canal puede acceder al S_INFORMACION *
<b>REPORTE =</b>	<b>FORMATO_REPORTE</b>
<b>S_INFORMACION =</b>	[ DB_GOPHER   DB_WWW  DB_LISTAS]
<b>SCSIB-UNAM =</b>	* Alias del Sistema de Control para el Sistema de Información del Bachillerato de la UNAM *
<b>SISTEMA_EN_OPERACION_CO NFIABLE =</b>	* SCSIB-UNAM en funcionamiento adecuado, confiable y consistente *
<b>SOLICITUD_DE_CLAVE =</b>	* Formato de petición de CLAVE compuesto ppor DATOS_USUARIO *
<b>SUPERIOR =</b>	* Nombre del jefe inmediato del PROFESOR que autoriza la SOLICITUD_DE_CLAVE *
<b>SUSUARIO =</b>	[ USUARIO   PUBLICO ]
<b>TELEFONO =</b>	1{0..9} 7
<b>TIPO =</b>	[ PROFESOR   ALUMNO ]
<b>ULTIMO_ACCESO =</b>	* FECHA de la última ocasión en que un USUARIO accedió a su CLAVE *
<b>USUARIO =</b>	DATOS_USUARIO+LOGIN+PASSWORD

## CAPÍTULO IV.

### **IMPLEMENTACIÓN DEL ESQUEMA.**

En capítulos anteriores se ha definido el esquema alternativo para dar solución a la problemática identificada en el Sistema de información destinado al Bachillerato de la UNAM, el punto de vista dado ha sido de tipo administrativo y de procedimientos, identificando de forma metódica y formal los elementos, variables y condiciones sobre los que actuará el sistema, es hasta este punto posible conocer la operación del sistema propuesto independientemente de su implementación.

Este capítulo se llevará a cabo la descripción de la implementación, de los elementos, servicios y herramientas que forman parte del esquema propuesto, a partir de los recursos con que se cuenta en este momento. Es importante señalar que la propuesta contempla ser adaptado en el futuro a extensiones tanto en su plataforma como en sus servicios.

La implementación se llevará a cabo dentro de un ambiente de red, Unix específicamente, y es para este tipo de plataforma que se dirige el conjunto de labores destinadas a implementar cada uno de los componentes del esquema en su totalidad.

Los servicios de red que serán incorporados a este sistema serán detallados uno a uno en el transcurso de este capítulo y son los siguientes.

- Correo electrónico.
  - Configuración
  - Control
  - Diseño de las formas de Web
  - Integración
  
- Listas de correo.
- Servicio de Gopher.
  - Configuración
  - Integración
  
- WWW ( World Wide Web ) .
  - Configuración
  - Integración

En cuanto al sistema de administración descentralizada de las claves de correo electrónico, generadas en cada uno de los planteles del bachillerato, parte fundamental de esta investigación, será implementado a través del servicio de WWW, para lo cual el sistema contará con un servidor de este tipo destinado a este fin.

Como parte del mismo sistema de administración y mantenimiento de los servicios, la actualización de la información disponible a través del servicio de Gopher se llevará a cabo mediante el uso del correo electrónico.

#### 4.1 Recursos disponibles para el Sistema Bachiller.

Hardware:

1 CPU Sparc Classic de Sun Microsystems.  
16 Megas de memoria RAM.  
1.5 Gygabytes en Disco duro.  
Sin monitor.  
Sin consola.  
Sin teclado.

- Sistema operativo: Unix BSD, SunOS 2.1.4
- Nombre del equipo: bachiller.dgsca.unam.mx
- Dirección electrónica: 132.248.10.20
  
- Actualmente cuenta con los siguientes servicios :
  - Correo electrónico .
  - Gopher.
  - Listas de discusión.
  - Foros de discusión.

Los procedimientos mediante los que operan estos servicios se describe en el Capítulo II.

#### 4.2 Correo electrónico.

El correo electrónico es sin duda el servicio de mayor utilidad dentro de Internet, ya que permite el intercambio de mensajes y documentos entre usuarios de este servicio sin importar las distancias.

La mayoría de los usuarios de Internet comienzan manejando el correo electrónico que de forma sencilla puede aprenderse. El correo electrónico es conocido como un servicio de "almacena y redirecciona", ya que el correo pasa de una máquina a otra hasta llegar a su destinatario final, partiendo de la máquina en la que se generó el mensaje donde sea que esta se encuentre.

El correo electrónico significa más que envíos de mensajes de forma casi instantánea, sino que permite al enviar un mensaje incorporarle cualquier tipo de documento (en forma de archivo) y una firma, el destinatario al recibirlo podrá depositarlo en una computadora o en su una cuenta de correo, para mas tarde poder hacer uso del mismo.

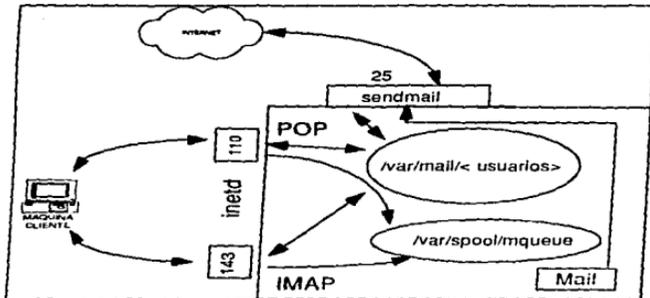


Figura 4.1

Diagrama de funcionamiento del correo electrónico.

De la misma forma cuando son generados mensajes por los usuarios para ser enviados a través de un programa cliente, estos son depositados en un recipiente que actúa como una cola de mensajes por enviar, este recipiente es el archivo de sistema `/var/spool/mqueue`, y de este son enviados a sus destinatarios como el sistema es capaz de despacharlos.

Para que los programas clientes puedan realizar las labores de enviar y recibir los mensajes de los usuarios, deben cumplir con alguno de los siguientes protocolos de acceso de mensajes, para interactuar con el sistema Unix, POP (Post Office Protocol) y/o IMAP (Internet Message Access Protocol).

#### Diferencias entre POP e IMAP.

Con POP, el correo es depositado en un recipiente compartido, y los programas clientes ejecutados en computadoras personales, periódicamente se conectan al servidor y extraen todo el correo pendiente. Después todo el procesamiento de los mensajes es delegado a la máquina cliente. POP es un proveedor del servicio de "almacena y redirecciona", implementado para mover correo desde un servidor intermediario (recipiente compartido) hacia la máquina destinatario, usualmente una computadora personal, al ejecutar lo anterior, generalmente mueve el mensaje

### Instalación y configuración.

- Los directorios necesarios para el funcionamiento del servicio de correo, tanto el repositorio común como el de la cola de mensajes por enviar. En el caso de los directorios `/var/mail` y `/var/spool/queue` ya se encuentran definidos en el sistema, por lo que no es necesario crearlos.
- El daemon servidor de correo también ya se encuentra como parte del sistema operativo, se localiza en `/var/lib/sendmail`, para su adecuado funcionamiento es necesario realizar una serie de modificaciones al archivo de configuración siguiente:

`/etc/sendmail.cf` En este archivo es necesario modificar información, tal como el nombre de la máquina donde reside nuestro servidor, el dominio donde se encuentra y alias.

- Una vez realizada la edición, para que el sistema reconozca los cambios hechos, el daemon `sendmail` debe ser terminado y vuelto a activar, desde la línea de comandos del administrador de bachiller:

```
bachiller% kill -9 < pid de sendmail >
bachiller% /usr/lib/sendmail -bd -q1h
```

Ahora es necesario implementar los protocolos de manejo de mensajes POP (en este caso la versión POP3d) e IMAP (en este caso la versión IMAPd), para ello es necesario seguir los siguientes pasos.

- Obtener los servidores de POP e IMAP para sistema Unix, esto es posible a través del servicio de FTP anónimo.

En cuanto a POP, uno de los servidores de FTP anónimo que lo contiene es mantenido por Qualcomm creador del cliente de correo implementado a través de este protocolo, se localiza en :

FTP anónimo **FTP.qualcomm.com** en el directorio `/quest/Unix/servers/popper` archivo **qpop**

En relación a IMAP su distribución es acompañada de `ipop2d` y `ipop3d`, que son POP2 y POP3 respectivamente, y se encuentra en:

FTP anónimo **FTP.cac.washington.edu** directorio `/imap` archivo **imap.tar.z**

- Una vez obtenidos ambos servidores deben ser declarados como disponibles desde el siguiente directorio del sistema:

**/etc/services**

ambos podrán residir en diversas localidades del sistema:

- Debe editarse el archivo **/etc/inet.conf** para incluirlos en sus tablas.
- El archivo **imap.c** , que contiene ambos servidores debe ser compilado:

```
bachiller% cc imap.c -o <nombre y ruta deseada que ocupe el programa objeto>
```

- Reinicializar las tablas del daemon **inetd**, esto se consigue deteniendo el daemon y volviéndolo a activar :

```
bachiller% kill -HUP <pid del daemon inetd>
bachiller% /usu/etc/inetd
```

A partir de este momento han sido creados los daemons de POP e IMAP en sus versiones POP3d e IMAPd, para hacer posible el alta usuarios en nuestro sistema de forma mas completa.

El procedimiento de creación de una cuenta de correo electrónico implica la generación de cuatro datos importantes al sistema:

<b>login</b>	Identificador único de usuario en el sistema.
<b>userid</b>	Identificador numérico único de usuario, con el que el sistema interactúa verdaderamente.
<b>password</b>	contraseña de acceso a la clave, se genera por medio de alguna función, no es posible consultarlo, incluso por el administrador, lo que se almacena junto con los datos de usuario es una encripción del mismo.
<b>cuota</b>	este último dato, en caso de no ser definido, dará libertad de espacio ilimitado al usuario, lo cual no es conveniente.

Para que una cuenta sea creada, debe crearse un directorio para cada usuario que se da de alta, esto es crear un directorio en **/home/user** como usuarios albergue nuestro servicio, los datos de los usuarios se encuentran en el directorio **/etc/passwwd**, los datos que contiene son:

login  
password encriptado  
identificador de usuario (uid)  
identificador de grupo (gid)  
comentarios (generalmente el nombre real)

Para el funcionamiento del sistema de correo es necesario crear la clave de correo electrónico, esto se lleva a cabo creando un archivo por usuario bajo el directorio **/var/spool/mail**, este archivo es creado automáticamente al recibir el usuario el primer mensaje en su cuenta enviado generalmente es por el administrador, donde le da la bienvenida y le dan a conocer las normas del servicio, a este archivo van anexándose los mensajes destinados al usuario.

Por el contrario la operación de baja de clave implica la eliminación del directorio de usuario en **/home/user** tomando como referencia el login, además de la eliminación del archivo del usuario de correo en **/var/spool/mail**.

Las operaciones que impliquen algún tipo de modificación a los datos de usuario pueden incluir cambio de password, cambio de cuota o alguno otro de los datos de usuario excepto **login** y **uid**, si estos últimos son los motivos de modificación el procedimiento a seguir debe ser la eliminación de la clave y la generación de una nueva.

#### 4.2.3 Administración del sistema de correo electrónico.

El mecanismo que actuará como el medio a través del cual se lleven a cabo labores administrativas del servicio de correo electrónico, tales como altas, bajas, cambios y consultas de las claves generadas para este servicio. Estará implementado a través de "formas" accesibles desde el servicio de WWW (descrito en el apartado 4.5 de este capítulo), con el fin de proporcionar una interfase lo suficientemente ilustrativa de las actividades que se estén llevando a cabo, el acceso a estas formas se realiza a través de contraseñas que únicamente los Administradores Locales tendrán para proporcionar seguridad en cuanto al acceso a las labores de manipulación de información de los usuarios del servicio de correo electrónico.

Con el uso de este tipo de interfaces se busca reducir el grado de especialización de los Administradores locales que conformen el esquema de administración, en cuanto a labores de administración en ambiente Unix.

Una forma es un formato que permite recabar información específica a través de su llenado, los datos obtenidos por este medio son enviados hacia alguna localidad para servir como insumos a un proceso que será ejecutado en un servidor remoto. Este vínculo lo establece la forma mediante una referencia. Estos procesos tendrán la tarea de realizar las acciones relacionadas a la administración de la información como editar archivos,

asignar contraseñas o hacer consultas, en este caso concernientes al sistema de archivos del servicio de correo electrónico de Bachiller.

El documento de WWW que contendrá las formas mencionadas proporcionará básicamente cinco opciones Altas, Bajas, Cambios, Cambios de Password y Consultas de claves de correo. Cada opción desplegará una "forma" distinta de acuerdo los datos necesarios para llevar a cabo cada una de las cuatro acciones mencionadas, los datos que cada acción necesitará para llevarlas a cabo son los siguientes:

**Altas de claves.**

Nombre del usuario  
Plantel al que pertenece.  
Teléfono.  
Tipo de usuario (Profesor ó alumno).  
Número de cuenta en el caso de los alumnos.  
Persona que autorizó la clave en caso de profesores.

**Baja de clave.**

login (identificador de usuario en el sistema) del usuario.

**Cambios.**

login  
Dato de usuario a modificar (excepto login).  
Nuevo dato de usuario.

**Consultas.**

login del usuario.

**Cambio de Password.**

login del usuario.

Los datos proporcionados en cada una de las formas, serán tratados de distintas formas a través de un conjunto de programas y CGI's que tendrán la tarea de editar archivos, como el caso de `/etc/passwd`, manipular (consultar, crear o eliminar) los directorios de `/var/mail/home/user/<usuarios>` y en el caso de Altas y Cambios la generación de passwords.

El detalle de los CGI y su implementación se describe en el **Apéndice 3.**

### 4.3 Listas de correo.

Las listas de correo surgen como necesidad ante la generación de grupos de trabajo, que tratan habitualmente temas comunes, una forma de tratar estos temas y enterarse de forma simultánea de las intervenciones de los demás es pertenecer a una lista de correo.

Una lista de correo un foro donde son tratados temas de interés colectivo, y quienes pertenecen a dicha lista reciben los mismos mensajes que los demás. Cualquiera que tenga un domicilio electrónico, es decir, una clave de correo, puede pertenecer a ellas, de la misma forma alguien con el solo hecho de dirigirse a alguna lista está enviando su mensaje a todos los integrantes de esta.

Toda lista de discusión debe ser monitoreada y administrada, el software que realiza esta tarea es Majordomo, y es posible anejarlo como un accesorio al conjunto de software instalado en Bachiller.

Majordomo es un conjunto de programas que permiten manejar rutinas de administración de listas de correo en Internet. Sirve además como un domicilio electrónico al que los usuarios de las listas residentes en un servidor, consultan toda la información referente a este servicio. Las peticiones de información se envían a través de correo electrónico en forma de comando asociado a una petición.

Existen tres tipos de usuario para Majordomo:

- Usuarios.
- Propietarios de las listas
- Administrador del servidor Majordomo (majordomo-owner) .

Un usuario es aquella persona que se encuentra inscrita a alguna de las listas ya creadas.

Los propietarios de las listas son las personas las cuales diariamente se encargan de dar respuesta a la serie de mensajes que llegan a la lista de correo generados por los usuarios. Cada lista de correo en el sistema debe contar con un propietario.

#### 4.3.1 Clasificación de las listas.

Dentro del sistema de listas de Majordomo existen varios tipos:

De acuerdo a la forma en que se lleva a cabo la inscripción:

**Abiertas:** la suscripción a este tipo de listas no requiere de ningún requisito especial o aprobación, se lleva a cabo con solo enviar un mensaje a la dirección de la lista .

**Cerradas:** La inscripción a una de estas listas debe pasar por un proceso de aprobación del propietario de la misma.

De acuerdo a la difusión de los mensajes:

**Moderadas:** todo mensaje que llega a la lista es verificado por su propietario antes de ser enviado a todos los usuarios de dicha lista.

**No moderadas :** no es necesaria la verificación de los mensajes dirigidos a la lista, al llegar automáticamente se envían a los usuarios de la misma.

De acuerdo a la consulta de la información de usuarios:

**Lista públicas:**En estas listas los suscriptores tienen acceso a información de otros a través de algunos comandos.

**Listas privadas:**Es en una lista de este tipo donde no puede realizar lo hecho en las listas públicas.

De los tipos anteriores pueden generarse combinaciones, que proporcionarán variedad en los tipos de listas que se deseen implementar.

#### **4.3.2 Obtención e instalación del software de Majordomo.**

El software servidor de listas de correo Majordomo se encuentra disponible en la siguiente localidad de FTP anónimo **FTP.greatcircle.com** dentro del directorio **/pub/majordomo** .

Una vez obtenido el software necesario debe considerarse que debe constarse con una versión reciente de Perl en el sistema, ya que Majordomo es un conjunto de programas en Perl, es necesario también, contar con un compilador de C.

La serie de acciones encaminadas a instalar el servidor de listas de correo es la siguiente:

- Elegir un nombre que identifique al servidor, es recomendable elegir "Majordomo".
- Agregar al sistema un grupo y un usuario que pertenezca a este, el usuario debe tener como nombre **majordom** y grupo **majordom**, esto para llevar a cabo con estos

identificadores la ejecución de los programas que suministrarán el servicio de listas de correo:

Se agregará una línea al archivo `/etc/passwd` del sistema de archivos, en carácter de superusuario (root). La línea deberá tener la siguiente información :

**majordomo:<contraseña encriptada>:<userid>:<groupid>Servidor de Listas  
Majordomo>: /usr/local/majordom : /bin/csh**

- Depositar los archivos conseguidos vía FTP en pasos anteriores en el directorio del usuario creado: `/usr/local/majordom`
- Como superusuario agregar la clave majordom como un usuario tipo **trusted** ( aquel que puede enviar mensaje como si fueran enviados por el mismo sistema , como el superusuario ).
- Crear los directorios necesarios para el manejo de listas:

```
/usr / local / mail
```

```
/usr / local / mail / list
```

```
/usr / local / mail / archive
```

Estos directorios deben contar con permisos 775 ( rwxrwx-x ) .

- Editar el archivo `/usr/local/majordomo/Makefile` para especificar los paths (trayectorias de archivos) adecuados a nuestro sistema, donde se localizan herramientas que se emplearán durante la compilación de los programas:

```
W_BIN= / usr / local / majordomo
```

```
W_PATH= / bin / usr / ucb
```

```
W_HOME= $(W_BIN)
```

```
W_SHELL= / bin / csh
```

- Par generar los programas objetos que serán compilados es necesario ejecutar desde la línea de comando, dentro del directorio `/usr/local/majordomo` :

```
bachiller% make
```

- La compilación de los archivos objetos generados se lleva a cabo ejecutando desde la línea de comando, dentro del directorio `/usr/local/majordomo` :

```
bachiller% make install
```

- Ejecutar como superusuario desde la línea de comando, dentro del directorio **/usr/local/majordomo**:

**bachiller% make install wrapper**

Lo anterior generará el archivo **wrapper** necesario para ejecutar otros programas necesarias para el funcionamiento del sistema de listas.

Hasta este punto se ha compilado la serie de archivos necesario para poner en funcionamiento el servidor de listas.

Para configurar el programa servidor de acuerdo a las características de nuestro sistema es necesario:

- Editar el archivo **/usr/local/majordomo/majordomo.cf**. Este archivo es un conjunto de variables que definen el ambiente bajo el que operará Majordomo, esto incluye la configuración de las listas, la especificación de los directorios donde se almacenará el correo generado por las listas, datos del propietario del servidor de Majordomo y la máquina donde este reside:
- Copiar el archivo **/usr/local/majordomo/majordomo.cf** sobre el **/etc/majordomo** para actualizar la configuración generada en el paso anterior.
- Asignar el alias que identificará al servidor, agregando al archivo **/etc/aliases** las siguientes líneas:

**# ALIASES FOR MAJORDOMO**

**#**

**majordomo: "/usr/local/majordomo/wrapper majordomo"  
majordomo-owner:< Identificador del propietario e majordomo>  
owner-majordomo:< Identificador del propietario e majordomo>**

- Si es necesario editar las primeras líneas delos scripts de Perl del directorio de Majordomo, a fin de que localizen el compilador de Perl que deberá estar ubicado en **/usr/local/bin/perl** ya que ambas cosas son necesarias para la operación del servidor de listas :

```
/usr/local/majordomo/approve  
/usr/local/majordomo/bounce  
/usr/local/majordomo/bounce-remind  
/usr/local/majordomo/majordomo  
/usr/local/majordomo/medit  
/usr/local/majordomo/new-list  
/usr/local/majordomo/request-answer  
/usr/local/majordomo/resend
```

A partir de este momento la dirección de Majordomo donde se recibirán las peticiones de inscripción a las distintas listas, así como solicitud de información de las mismas es:

***majordomo@bachiller.dgsc.unam.mx***

#### **4.3.3 Administración de Majordomo.**

El administrador es la persona encargada de instalar el software de Majordomo, y puede crear listas y manipular sus tipos.

A diferencia de los propietarios de las listas el Administrador tiene acceso como usuario al servidor donde reside el sistema de listas (en este caso a bacjhiller.dgsc.unam.mx).

Majordomo es un conjunto de programas hechos en lenguaje Perl, los mas importantes son:

- |                  |  |
|------------------|--|
| <b>majordomo</b> | este programa se ejecuta cada vez que un mensaje llega a la dirección <b><i>majordomo@bachiller.dgsc.unam.mx</i></b> . Resuelve los requerimientos enviados y devuelve respuesta a los mensajes de los usuarios, advierte de requerimientos que necesitan aprobación y notifica a los propietarios de las listas sobre peticiones de inscripción. No es necesario que se encuentre ejecutándose de forma constante (como un daemon), lo que proporciona una notable ventaja. |
| <b>resend</b>    | este programa se encarga de verificar el mensaje a una lista con el fin de incorporarse a ella, realiza la verificación en base a los parámetros establecidos para este fin. Al presentarse algún inconveniente respecto a los mensajes que aparecen en la lista, son enviados al propietario de la lista, este se encargará de editar los encabezados de los mensajes para asegurarse lleguen a los usuarios de la lista.   |
| <b>wrapper</b>   | es el único programa en C, y es el que hace posible la ejecución de los demás programas necesarios para el funcionamiento de las listas.   |
| <b>bounce</b>    | es el programa encargado de manejar los mensajes de tipo "bounce" (son aquellos ) que al ser enviados a la lista presentan algún problema. El programa es de gran ayuda para el propietario de la lista.   |
| <b>approve</b>   | es un programa para propietarios de las listas y sirve para aprobar las inscripciones a las listas.  |

`majordomo.cf` es el archivo de configuración para el servidor de Majordomo.

### **Creación de listas de correo.**

El procedimiento para la solicitud de creación de una lista es la siguiente:

Los interesados en crear una lista y ser el moderador de la misma si así lo requiere el carácter de esta, es decir, según la temática a que se destinará, deberán enviar un mensaje al administrador de Majordomo, para exponerle los motivos de la creación de la lista, el empleo que se le dará y las características con que contará, es decir, de que tipo será. Además deberá incluir la dirección electrónica de quien actuará como moderador .

Cabe señalar que la solicitud de la lista debe haber sido ya estudiada por el Administrador Local correspondiente, quien deberá notificar al Administrador de Majordomo su opinión al respecto a través del correo electrónico a la dirección `majordomo-owner@bachiller.dgsc.unam.mx` .

Una vez aceptada la petición se procederá a la creación de la lista por el administrador de Majordomo.

El procedimiento para crear una lista de correo descrito a continuación, requiere los siguientes datos :

- Nombre de la lista.
- Dirección electrónica del moderador o propietario.
- Tipo de la lista.
- Contraseña

La creación de listas requiere la creación de algunos archivos, los cuales sean depositados en el directorio `/usr/local/mail/lists` .

- Crear un archivo denominado como la lista, este nombre debe ser lo suficientemente representativo de la temática tratada en la lista, puede incluir guiones bajos, guiones, letras y dígitos:

`/usr/local/mail/lists/ <nombre de la lista>`

- Crear un archivo que contendrá la contraseña que requerirán los propietarios de las listas para realizar las aprobaciones de inscripciones y cancelaciones de suscripción, cuando sea necesario. El archivo deberá tener el mismo nombre de la lista con la extensión `.passwd` .

`/usr/local/mail/lists/ <nombre de la lista >.passwd`

- Editar el archivo que contenga la información referente a la lista, como nombre, temática, dirección donde suscribirse, dirección para obtener información, por ejemplo. El archivo deberá tener el mismo nombre de la lista con extensión **.info**, este archivo será desplegado cuando algún usuario solicite información acerca de esa lista.

**/usr/local/mail/lists/ <nombre de la lista >.info**

- Adicionar los aliasos (otros nombres con los es posible referenciar a la lista) necesarios para las listas, dentro del archivo **/etc/aliasos**. es en este archivo también donde se define al que será el propietario y moderador de la lista.

```
#
< nombre de la lista >: "/home/majordomo/majordomo-1.93/wrapper resend -p
bulk -M 10000 -l < nombre de la lista > -f < nombre de la lista >-owner -h
bachiller.dgsca.unam.mx -s < nombre de la lista >-outgoing"
owner-< nombre de la lista >: < nombre de la lista >-owner
< nombre de la lista >-outgoing: :include:/home/majordomo/majordomo-
1.93/mail/lists/< nombre de la lista >, < nombre de la lista >-archive
owner-< nombre de la lista >-outgoing: < nombre de la lista >-owner
< nombre de la lista >-archive: /home/majordomo/majordomo-1.93/mail/archive/<
nombre de la lista >
owner-< nombre de la lista >-archive: < nombre de la lista >-owner
< nombre de la lista >-request:
"/home/majordomo/majordomo-1.93/wrapper request-answer < nombre de
la lista >"
owner-< nombre de la lista >-request: < nombre de la lista >-owner
< nombre de la lista >-approval: < e-mail del dueño de la lista >
< nombre de la lista >-owner: < e-mail del dueño de la lista >
owner-< nombre de la lista >-owner: < e-mail del dueño de la lista >
```

- Opcionalmente crear el archivo **/usr/local/mail/archive/ <nombre de la lista>** donde serán archivados todos los mensajes que lleguen a la lista.
- Asegurarse que los archivos y directorios creados hasta este momento cuente con los permisos 664 (rw-rw-r--) para archivos y 775 (rwxrwx-w-) para directorios, ya que son necesarios para la operación del sistema.
- Hasta este punto la lista creada es de tipo abierta y no moderada, es decir no existe restricción para inscribirse a ella y no requiere que el propietario de la lista apruebe las inscripciones o cancelaciones a inscripciones en la lista.

En el caso de listas cerradas es necesario realizar además las siguientes operaciones:

- Crear un archivo vacío denominado igual que la lista y con extensión **.closed**.

**/usr/local/mail/lists/ <nombre de la lista >.closed**

- Enviar un mensaje a **majordomo@bachiller.dgscs.unam.mx**, lo que creará automáticamente el archivo **/usr/local/mail/lists/ <nombre de la lista >.config**, en este archivo el sistema indica si la lista es abierta o cerrada ( al advertir la presencia del archivo **/usr/local/mail/lists/ <nombre de la lista >.closed** ), agregará una línea indicando **<nombre de la lista > = closed**

En el caso de que la lista sea moderada deberá :

- Editar el archivo **/usr/local/mail/lists/ <nombre de la lista >.config** indicando en el apartado correspondiente **moderate = yes** y **approve\_passwd = <nombre de la lista >.passwd**

#### 4.3.4 Empleo de las listas de correo.

A continuación se enumeran la serie de acciones que son posibles de realizar mediante el empleo de Majordomo, a petición de los suscriptores de las listas o a los aspirantes a serlo.

El procedimiento es simple, se envía un mensaje dirigido a **majordomo@bachiller.dgscs.unam.mx**, en el cuerpo del mismo colocará alguno de los comandos que le permitirán ejecutar alguna de las siguientes peticiones .

Petición de suscripción:

**subscribe** <nombre\_de\_la\_lista> [<dirección\_electrónica\_del-interesado>]

Petición de desistimiento de suscripción :

**unsubscribe** <nombre\_de\_la\_lista> [<dirección\_electrónica\_del-interesado>]

Petición del índice de archivos disponibles en alguna lista :

**index** <nombre\_de\_la\_lista>

Petición de obtener un archivo disponible a través de alguna lista :

**get** <nombre\_de\_la\_lista> <nombre\_del\_archivo>

Petición de saber a que listas se encuentra suscrito (si se indica la dirección\_electrónica\_del\_interesado, lo hará para ese suscriptor) :

**which** [<dirección\_electrónica\_del\_interesado>]

---

Petición de conocer los inscritos a una lista en particular :

**who**            <nombre\_de\_la\_lista>

Petición de información general sobre alguna lista en particular :

**Info**            <nombre\_de\_la\_lista>

Petición del índice de las listas que alberga Majordomo :

**lists**

Petición de la lista de comandos y su sintaxis , disponibles para los suscriptores y aspirantes a serlo.

**help**

En cuanto a los propietarios de las listas llevarán a cabo las labores tales como aceptación de suscripciones, desistimiento de suscripciones en el caso de ser listas cerradas, y la aceptación de mensajes que llegan a la lista para su distribución hacia sus suscriptores en el caso de listas moderadas, para ello necesitan conocer la contraseña proporcionada al crear la lista para este fin.

#### **4.4 Servicio de Gopher.**

Gopher es un servicio de información que proporciona navegación a través de un menú de opciones etiquetadas con texto, y el usuario puede seleccionar alguna, que nos muestra otro menú o información en algún formato.

Los menús funcionan de manera jerárquica, generalmente por tema o en base a algún ámbito de la cultura, incluida en esta jerarquía se encuentran documentos que pueden ser visualizados o grabados en las máquinas clientes , pueden llevarse a cabo búsquedas de información. Algunos de los elementos de los menús pueden ser ligas a otros Gophers , o algún otro tipo de servidor de información como los de FTP.

Este tipo de organización de la información resulta muy poderosa al estar adecuadamente ordenada y mantenida.

Gopher es un servicio con una administración sencilla, ya que la estructuración de la información consiste en crear un directorio organizado y sobre el que va a trabajar el servidor de Gopher, los subdirectorios hijos serán los elementos del menú principal y los archivos correspondientes a estos directorios serán parte de la información presentada a los clientes de este servicio.

Las características que hacen de este servicio ser efectivo son: mantener los datos disponibles actualizados, que los nombres de las opciones sean verdaderamente representativos y mantener las ligas a otros servidores constantemente actualizadas.

Gopher solo puede ligar los elementos que contiene desde los menús lo que representa una desventaja, pero puede soportar cualquier tipo de información, que puede ser interpretada en las computadoras cliente a través de algún programa dedicado a este fin.

Los clientes de Gopher pueden ser orientados a texto o gráficos, pero el servicio fue pensado en ambiente de texto ASCII, por lo que las búsquedas y despliegues de la información suelen ser bastante rápidos y favorecen a comunicaciones lentas como la hecha a través de módem.

El potencial de Gopher reside en que un elemento de menú puede hacer referencia a archivos o directorios en otros servidores de Gopher, a archivos en algún servidor de FTP, a algún servicio de telnet o a otros Gophers.

#### **4.4.1 Obtención e instalación del software servidor de Gopher.**

En el caso de Bachiller que es un sistema Unix, con sistema operativo tipo BSD SunOS 4.1.3 se optó por una versión sencilla y de fácil manejo, esta versión es la 2.016 de Gopher.

Obtención de los archivos fuentes para instalación del servidor de Gopher:

Se lleva a cabo a través de FTP anónimo en la localidad **boombox.micro.umn.edu** obteniendo del directorio **/Unix/old-version** el archivo **Gopher2.016.tar.z**, este archivo incluye tanto el cliente como el servidor de Gopher, y es durante la instalación cuando se decide que instalar.

Una vez obtenido el archivo mencionado deberá colocarse en un directorio vacío para descomprimirlo:

```
bachiller% uncompress Gopher2.016.tar.z
```

Ya descomprimido, ahora es necesario quitarle el formato tar de compresión:

```
bachiller% tar -xvf Gopher2.016.tar
```

Para iniciar el proceso de compilación requerimos modificar el archivo **makefile.config** generado durante los dos pasos anteriores:

Indicar el compilador de C disponible en el sistema:

**CC= /usr/bin/cc**

Especificar la orientación del sistema operativo, en este caso:

**GSYSTYPE=DBSD**

Especificar la trayectoria del generador de librerías dinámicas:

**RANLIB= /usr/bin/ranlib**

Esopecificar el dominio de la máquina que albergará este servidor:

**DOMAIN= .dgsca.unam.mx**

Definir el directorio de datos sobre el que actuará el servidor:

**SERVERDATA= /home/Gopher/datos**

Definir el puerto asignado por el sistema para este servicio:

**SERVERPORT=70**

Ahora es necesario modificar el archivo **config.h** para definir los servidores de Gopher que atenderán a los clientes de este servicio y en que puertos:

```
# define CLIENT1_HOST "bachiller.dgsca.unam.mx"
# define CLIENT2_HOST "bachiller.dgsca.unam.mx"

# define CLIENT1_PORT 70
# define CLIENT2_PORT 70
```

Para que los cambios en los dos archivos anteriores tengan efecto debe darse de baja el daemon **inetd** encargado de monitorear las requisiciones en los puertos del sistema y volverlo a activar:

```
bachiller% kill -9 < pid de inetd >
bachiller% /etc/inetd
```

Es ahora cuando debemos compilar los archivos que con los que se llevará a cabo la instalación, en nuestro caso los referentes al servidor de Gopher, que se encuentran depositados en el directorio **object**, bajo el directorio donde descomprimimos todos los archivos hasta el momento:

```
bachiller% make server
```

Una vez generados los archivos con el paso anterior , ahora debemos ejecutar la compilación de los mismos, esta acción debe hacerla el superusuario, y únicamente hará un movimiento de los archivos necesarios para el funcionamiento del servidor hacia los directorios definidos por nosotros en los pasos anteriores:

```
bachiller# make.install
```

Hecho lo anterior podemos poner a funcionar el servidor de Gopher, activando el daemon **Gopherd** e indicando el archivo bitácora :

```
bachiller# ./Gopherd -cu Gopher -l /home/Gopher/log70 -o /etc/Gopherd.config  
/home/Gopher
```

Una vez instalado nuestro servidor debemos configurarlo, para hacerlo es necesario editar el archivo **Gopherd.config** en las siguientes líneas:

```
cachetime=180  
Admin= <Nombre del administrador >  
AdminEmail= <dirección electrónica del administrador>  
  
Site=< Departamento del administrador>  
Org=< Organización que pertenece el administrador>  
Loc= <Lugar donde se encuentra la organización>
```

Las siguientes líneas del mismo archivo ayudan a que los clientes de forma automática puedan interpretar la información que reciben de acuerdo al formato, lo que se conoce como tabla MIME:

```
Decoder: <extensión de archivo comprimido> <ruta del archivo decompresor  
correspondiente>  
viewer: <extensión del archivo a visualizar> <ruta del programa visualizador>
```

También podemos indicar que directorios bajo nuestro directorio de datos no queremos sean tomados en cuenta por el servidor:

```
Ignore: <ruta del directorio/archivo que será ignorado>
```

Otro aspecto es definir el mensaje de error, útil para el caso en que se genere un error durante el uso del servicio:

```
BummerMsg: <mensaje de error que aparecerá en pantalla>
```

A partir de este momento es posible comenzar a introducir información para ser accesada por los usuarios de este servicio:

#### 4.4.2 Incorporación de información al servicio de Gopher.

El mecanismo propuesto para incorporar información al servicio de Gopher, funciona de la siguiente forma, cada uno de los Administradores Locales en cada uno de los planteles, podrá introducir la información generada en su plantel de dos formas distintas: por medio de correo electrónico y mediante la interacción con un directorio de información que pertenece al servicio y que será administrado por el Administrador Local y del cual, éste será responsable.

Antes de comenzar la introducción de los datos debe tenerse una idea plenamente clara de la forma en que se organizará la información, a fin de que en ningún momento la navegación entre menús resulte confusa o inconsistente.

##### a) Incorporación de información a través de correo electrónico.

Esta es una alternativa y consiste en que los administradores envían la información a integrar el sistema mediante un mensaje electrónico, al cual se anexa un archivo que contiene la información debidamente formateada para este fin, el accuso a la dirección destino a que llegará el mensaje es restringido únicamente a las claves destinadas a los Administradores locales, una vez enviado el mensaje es recibido en una cuenta especial de correo, la cual es monitoreada por un programa encargado de colocar los archivos anexos a los mensajes en los lugares del sistema de archivos del sistema de información, igualmente definidos en el mensaje.

##### b) Incorporación de información a través del manejo de directorios.

Esta opción es la más viable y será la que formará parte de la propuesta, consiste en asignar privilegios a los Administradores Locales sobre un determinado directorio, este directorio situado en el directorio de datos del servicio de Gopher, corresponderá a la información generada en el plantel a que corresponda, la forma en que debe incorporarse la información se explica más adelante.

De esta manera cada Administrador será responsable de la información generada en su plantel de igual forma su constante e ininterrumpida actualización y depuración. Es con este mecanismo que el Administrador Central únicamente se encargará de monitorear y asegurarse de que la estructura de la información, es decir el directorio de datos, no presente anomalías y de incorporar información de interés general a través de avisos.

Se propone que la información se encuentre estructurada de la siguiente forma :

**Menú principal.**

- *Acerca de estos servicios de información.*

Contendrá la descripción de los servicios disponibles para el Bachillerato de la UNAM, es decir, información genérica sobre Gopher, Correo electrónico y Listas de discusión.

- *Información sobre la UNAM.*

Será una liga al Gopher principal de la UNAM, es decir, hacia **gopher.unam.mx**

- *Avisos y eventos.*

En este apartado se encontrará información de interés para el bachillerato de la UNAM, tal como , cursos, exposiciones, conferencias, convocatorias, concursos y eventos en general dirigidos al Bachillerato y a la comunidad universitaria en general.

- *Información de los planteles del Bachillerato de la UNAM.*

Esta sub-sección es la mas importante del servicio de Gopher, existirá un apartado para cada plantel incorporado a este servicio, se recomienda que la estructura, sea idéntica en cada uno de los planteles , y será de la siguiente forma :

* <i>Avisos.</i>	Se conformará de avisos de relevante importancia para la comunidad de ese plantel.
* <i>Eventos.</i>	Estará compuesta por anuncios de eventos importantes formales e informales dentro del plantel, tales como, exposiciones, conferencias, concursos y festividades.
* <i>Calendarios.</i>	Contará con información completa de calendarios de Exámenes ordinarios, extraordinarios, intersemestrales.
* <i>Cursos.</i>	Se informará en este apartado de cursos dirigidos al Bachillerato que serán impartidos en alguno de los planteles del sistema.
* <i>Espacio cultural.</i>	Se incorporará información sobre temas de interés a la comunidad del abchillerato, siendo los temas libres, siempre y cuando cumplan con las políticas del buen uso del servicio.

- **Calendarios.**

Contendrá los calendarios oficiales de cursos emitidos por la UNAM para las actividades dentro de los planteles que integran el Bachillerato.

Como se mencionó los Administradores Locales tendrán la tarea y responsabilidad de incorporar la información generada en cada uno de sus planteles, para lo cual deben considerar la siguiente información.

**c) Características de la información.**

La información disponible a través de este servicio debe estar contenida en el directorio de datos, en él solo deben existir dos tipos de elementos, archivos y directorios.

Los directorios contenidos en bajo el directorio de datos serán las etiquetas que formarán mi menú principal, por cada nivel de archivos deberá existir un directorio **.cap** este contendrá archivos con los mismos nombre del directorio de nivel inmediato superior al que haran referencia, en estos archivos del directorio **.cap** se indican los siguientes parámetros:

**Type=** <tipo de elemento>  
**Name=** <título que hará referencia a la información contenida en el directorio o archivo a que se referencia >  
**Numb=** < lugar ascendente que ocupara esta etiqueta en el menú>  
**Path=** <Type/ trayectoria del archivo o directorio en el directorio de datos>

En el caso de que la etiqueta de menú apunte hacia otro servidor de Gopher, este elemento no será tangible, para este tipo de elemento debe crearse el el nivel correspondiente de archivos del directorio de datos donde se desee colocar dicha etiqueta, un archivo que inicie con punto y el nombre que se le asigne, este deberá tener su respectivo archivo en el directorio **.cap** de ese nivel, con los siguientes datos

**Type=** <tipo de elemento>  
**Name=** <título que hra referencia a la información contenida en el directorio o archivo a que se referencia>  
**Numb=** < lugar ascendente que ocupara esta etiqueta en el menú>  
**Path=** <Type/ trayectoria del archivo o directorio en el directorio de datos>  
**Host=** < servidor donde reside la información referenciada>  
**Port=** <número del puerto asignado en el host para este servicio>

Es a partir de este punto que es posible acceder a la información colocada en el servicio con el mecanismo anterior, a través de un cliente de Gopher, a la dirección :

**bachiller.dgsca.unam.mx**

#### **d) Eliminación de información.**

Si por el contrario si el objetivo es eliminar información, el procedimiento es el siguiente:

- Eliminar el archivo referencia contenido en el directorio **.cap** correspondiente.
- Eliminar el directorio/archivo a que hacia referencia el archivo antes eliminado, dentro del directorio de datos.

Con lo anterior se elimina la referencia y la información, y la etiqueta ya no estará mas disponible en la posición del menú donde residía.

#### **4.5 Servicio de World Wide Web .**

World Wide Web ( WWW o Web ) es el servicio de información en red mas popular, actualmente cuenta con una serie de características que le permiten incorporar innovadoras técnicas para mostrar la información. Este servicio trabaja en base a documentos de hipertexto e hipermedia.

A diferencia de Gopher, la estructura de la información a que tiene acceso no está estructurada de forma absoluta (jerárquica fija), sino que puede ser referenciada por y desde múltiples documentos situados a lo largo y ancho de Internet.

Un documento hipermedia es aquel que además de mostrar información de tipo texto, puede incorporar imágenes y animaciones, además mediante el empleo de hipertexto (palabras resaltadas) establecer vinculos hacia objetos ( imágenes, sonido, videos, animaciones) o a otros documentos de este tipo.

Otra característica muy importante es que através de él son incorporados otros servicios de información como Correo electrónico, FTP y Gopher . disponibles desde la misma interfase de WWW.

Esta serie de cualidades hacen de este servicio dar opciones ilimitadas para mostrar información, aunadas a la cobertura que proporciona Internet, es un medio muy solicitado para dar a conocer todo tipo de información de forma totalmente ilustrativa.

Los documentos que forman parte de este servicio pueden estar ubicados en cualquier punto de Internet siendo generados y administrados en los lugares que cuenten con un servidor para este tipo de servicio.

Estos documentos son desarrollados mediante el empleo de HTML (Hiper Text Markup Language).

---

## **HTML.**

Es un conjunto de palabras reservadas que mediante una estructuración definida para este tipo de documentos permiten incorporar hipertexto e hipermedia a los documentos. Cabe aclarar que no es un lenguaje de programación, sino que es una serie de especificaciones de formato que al ser interpretadas por un servidor HTTP permiten definir características de presentación al texto, hipertexto e imágenes que crearán vínculos hacia otra información.

## **HTTP (Hiper Text Transfer Protocol)**

Es el protocolo mediante el cual se comunican los clientes y servidores de este tipo de servicio, un servidor HTTP funciona aceptando peticiones de solicitud de documentos, hecha a través de una referencia en algún documento, (hipertexto e hipermedia), establece una conexión hacia la localidad que la contiene, la pasa al cliente y finalmente cierra la conexión.

## **Navegadores de WWW.**

Un navegador de WWW es un programa cliente que permite interactuar con documentos de hipertexto e hipermedia, proporcionando una interfase para la consulta de este tipo de documentos e información. Pueden añadirse una serie de programas accesorios para poder visualizar la información disponible a través de los documentos.

Este tipo de programas tanto clientes como servidores se encuentran disponibles para todo tipo de plataformas, sin embargo demandan una gran cantidad de recursos a los equipos que los soportan como almacenamiento, capacidad de procesamiento y memoria.

Los proveedores más conocidos son NCSA y Netscape Inc. por sus clientes NCSA Mosaic y Netscape, respectivamente, aunque ambos proporcionan tanto clientes como servidores para casi todo tipo de plataformas.

En el caso particular de este proyecto solo se contempla la incorporación del servidor HTTP al conjunto de software del sistema, ya que permitirá la implementación de la parte referente a la administración de las cuentas de correo electrónico, mediante el empleo de formas. El acceso a estas formas por parte de los Administradores de bachiller se realizará de forma restringida, mediante un cliente de WWW.

## **Las Formas dentro de la administración del sistema de correo electrónico.**

Una forma es un método de interactuar con un usuario del servicio de WWW, para obtener información en un espacio donde esta le es solicitada explícitamente.

Las formas empleadas en ambiente WWW, al igual que todo documento disponible a través de este servicio, se encuentran construidas con HTML.

Una forma implica dos elementos, requiere código tanto del servidor (proveedor del servicio) como del cliente (documento de WWW que el usuario accesa), a fin de recabar los datos y enviarlos para su procesamiento.

Las formas hechas en HTML permiten al crear el documento, definir campos que serán llenados con datos proporcionados por quien las consulta, cuando un usuario llena la forma y oprime un botón indicando que los datos pueden ser enviados al ser completados, la información contenida en la forma (los datos que el usuario proporcionó) es enviada a un servidor para su proceso, esta acción puede regresar una respuesta al usuario por el mismo medio resultado de tal proceso.

En el servidor un CGI (Common Gateway Interfase) procesa esta forma usando como argumentos la información proporcionada por el usuario a través de la forma.

La función del CGI es tomar la información capturada a través de la forma y darle algún tipo de proceso, generalmente es la actualización de algún sistema de archivos de un servicio o algún otro tipo de base de datos, para lo cual incorpora una serie de operaciones definidas durante su construcción. El CGI debe ser escrito y declarado para poder ser empleado, es implementado a través de un shell, el lenguaje Perl u otro disponible en el servidor que alberga el servicio de WWW o en algún otro script en Unix.

Los datos proporcionados en cada una de los formas, serán tratados de distintas formas a través de un conjunto de programas y CGI's que tendrán la tarea de editar archivos, como el caso de `/etc/passwd`, manipular (consultar, crear o eliminar) los directorios de `/var/mail /home/user/<usuarios>` y en el caso de Altas y Cambios la generación de passwords.

El detalle de los CGI y su implementación se describe en el **Apéndice 3**.

#### **4.5.1 Obtención e instalación del software servidor HTTP.**

En el caso de Bachiller que es un sistema Unix, con sistema operativo tipo BSD SunOS 4.1.3 se optó por una versión actualizada del servidor HTTP, esta versión es la 1.5.1 de NCSA.

- Obtención de los archivos fuentes para instalación del servidor de Gopher:

Se obtuvo el archivo para insytalación a través de FTP anónimo en la localidad **FTP.ncsa.uiuc.edu** del directorio **/Web/httpd/Unix/ncsa\_httpd/httpd\_1.5** .

El nombre de archivo es **httpd\_1.5.1-export\_sunos4.1.3.tar.z** , este archivo incluye el servidor HTTP y el conjunto de archivos necesarios para su instalación.

Antes de empezar el proceso tendiente a instalar el servidor HTTP debemos contar con los siguientes elementos:

- Contar con una página de Web, que será la bienvenida a este servidor, denominada `index.html` generalmente.
- Crear un usuario dentro del grupo # 30 , que usualmente alberga a los servicios como Gopher y WWW, en este caso se denominara `oasys` .
- El archivo **httpd\_1.5.1-export\_sunos4.1.3.tar.z**

- Una vez obtenido el archivo indicado debe colocarse en un directorio vacío para ser descompactado:

```
bachiller% uncompress httpd_1.5.1-export_sunos4.1.3.tar.z
```

- Ya descomprimido, ahora es necesario quitarle el formato `tar` de compresión, que generará la serie de archivos necesarios para la compilación e instalación del servidor :

```
bachiller% tar -xvf httpd_1.5.1-export_sunos4.1.3.tar
```

- Para iniciar el proceso de compilación requerimos modificar el archivo `makefile` generado durante los dos pasos anteriores, solo debemos indicar el compilador de C disponible en nuestro sistema mediante la ubicación del mismo:

```
cc= /opt/gnu/bin/gcc
```

- Es ahora cuando debemos compilar los archivos que con los que se llevará a cabo la instalación, en nuestro caso los referentes al servidor HTTP, depositados bajo el directorio donde descomprimimos todos los archivos hasta el momento:

```
bachiller# make
```

- Una vez compilados los archivos con el paso anterior, debe llevarse a cabo un procedimiento similar al de una instalación, esta acción debe hacerla el superusuario, y únicamente hará un movimiento de los archivos necesarios para el funcionamiento del servidor hacia los directorios definidos el mismo proceso de instalación:

```
bachiller# make.file
```

- Contamos ya con la estructura de archivos necesaria para poner en funcionamiento el servidor HTTP, ahora es debemos llevar a cabo la Configuración de los mismos, para definir los datos específicos referentes la máquina donde se instaló el servidor .

Editar el archivo **http.config** e indicar lo siguiente:

Port <b>80</b>	Puerto de recepción de peticiones de http.
Max Servers <b>7</b>	Numero máximo de sesiones en el sistema.
User <b>oasys</b>	Usuario definido para el servidor HTTP
Group <b>#30</b>	Grupo al que pertenece el usuario definido.
Server Admin <b>oasys@bachiller</b>	Usuario administrador del servicio.
Server Root <b>/home/oasys/http</b>	Directorio raíz del servidor de http.

ServeName **bachiller.dgsca.unam.mx** servidor HTTP

Editar el archivo **http.config** e indicar lo siguiente:

<Directory <b>/home/oasys/httpd/cgi-bin</b>	Directorio alternativo a Server Root definido en el archivo <b>http.config</b>
>	
<Directory <b>/home/oasys/httpd/datos</b>	Directorio de datos para el servidor HTTP
>	

Editar el archivo **srm.config** e indicar lo siguiente:

<b>Document</b> <b>/home/oasys/httpd/datos</b>	Directorio de datos para el servidor HTTP
<b>Directory</b> <b>Index.html</b>	Archivo que contiene la página de bienvenida al servidor que se encuentra en el directorio de datos.

Se ha concluido la configuración necesaria para el funcionamiento adecuado del servidor HTTP en **bachiller.dgsca.unam.mx**.

- La activación del daemon **httpd** , es el último paso para completar la implementación del servidor HTTP, se realiza con la siguiente acción:

**/home/oasys/httpd/bin/httpd -f /home/oasys/httpd/conf/http.config**

A partir de este momento un servidor HTTP se encuentra funcionando en nuestro sistema, aunque ha sido implementado con el fin de albergar las "formas" que formarán la interfase de usuario, y mediante la cual se llevará a cabo la administración de las cuentas de correo por parte de los Administradores Locales (el detalle de su funcionamiento y manejo se explica en el **Apéndice 4**) ; este no será su único fin, ya que podrá ampliarse para albergar mas páginas de diversos temas si las futuras consideraciones de la administración del sistema lo definen como necesario, tomando en cuenta las características físicas con que cuenta el equipo en relación a la demanda de recursos que implica este tipo de servicio.

---

**CAPÍTULO V****CONCLUSIONES**

Podemos concluir sin duda que la implementación del sistema propuesto redundará en una mayor eficiencia, al descentralizar responsabilidades y acciones. Ofreciéndoles a los administradores locales mayor facilidad en su manejo, al evitarles labores de administración de servicios directamente con los sistemas de archivos, reduciendo el grado de especialización que requieran estos últimos.

Por supuesto todo redundará en una mayor sencillez del esquema y una mas ágil incorporación de la información y atención a los usuarios, la primera podrá ser accesada de diversas formas por la comunidad estudiantil, proporcionando el invaluable servicio de la oportunidad de la información.

Por otra parte del sistema propuesto se desprende la necesidad de acrecentar los recursos con que actualmente se cuenta, a fin de optimizar el servicio, dándole una mayor capacidad, para que de esta manera sea posible expandir y diversificar la gama de servicios que son ofrecidos, este sistema podrá ser aplicado con más avanzadas tecnologías, siendo las modificaciones necesarias de fácil aplicación.

De la misma manera durante la realización de la presente tesis, sujeta a criterio del lector, fue posible identificar elementos tan importantes como para referirlos dentro de las conclusiones.

Estos puntos son los siguientes:

**En relación a los sistemas de información.**

I. Que para la implantación de cualquier sistema, que tenga por objeto el adecuado manejo de información y operaciones, es indiscutible, además de contar con la habilidad de abstracción y observación suficientes, el conocer la variedad de técnicas y metodologías existentes, así como las herramientas, elementos y aplicaciones que se involucren en su implementación, al grado de detalle necesario que permita la elección de la mejor alternativa, con la finalidad de asegurar en el mayor grado posible la eficiencia y ahorro de recursos, que nos lleva a la maximización de los mismos, aunque estos dos elementos no sean necesariamente proporcionales de forma inversa.

II. Que es necesario tener perfectamente claros los conceptos básicos relativos al análisis, diseño, desarrollo, implementación y mantenimiento de los sistemas, ya que de esta concepción se sustenta la aplicación que el Licenciado en Informática de al ciclo de vida de los sistemas, y que es siempre tomando en cuenta todo este ciclo y su interacción con otros sistemas que lleve a cabo sus decisiones a lo largo de cada etapa. Y que los principios fundamentales que nos ocupan, están basados en ideas sencillas, y es la

sencillez la mejor característica con que puede contar un sistema, ya que facilitará su manejo e interacción desde que es concebido ante una necesidad hasta que una nueva provoca su modificación o sustitución.

III. Que un sistema debe ser conducido desde su origen con método, orden y consistencia, para asegurar su adecuado seguimiento en cualquiera de sus etapas, por cualquier miembro del equipo encargado de su desarrollo, en cada aspecto que del sistema le corresponda desde su particular perspectiva.

IV. Que durante la implementación del sistema, debe tenderse hacia la descentralización y en su caso a la distribución, de las tareas operativas y procesos respectivamente, a fin de obtener un tiempo de respuesta menor desde cualquier punto que el sistema sea operado, sea de forma remota o local.

V. Que no hay sistemas de información aislados, que de su interacción con otros sistemas que formen parte del mismo entorno, es posible la debida retroalimentación que redundará en mayor consistencia y compatibilidad.

#### **En relación a los servicios de información en red.**

I. Que la importancia de estos servicios en de total vigencia, por lo cual su importancia dentro de las organizaciones se toma vital, para asegurar la oportunidad de operación, eficiencia y adecuada toma de decisiones.

II. Que cada vez más, de la importancia que se les otorgue dentro de las organizaciones y del aprovechamiento que de los mismos se de, depende el grado de éxito de estas últimas, independientemente de su campo de acción, principalmente en las de educación en todos los niveles que redundará directamente en el sector productivo del país.

III. Que de la misma manera, dada la gran variedad de tipos de servicios y de información que en ellos es manejada, desde, hacia y entre las organizaciones, es necesario implementar medios efectivos, suficientes y debidamente reglamentados, que aseguren el adecuado y seguro empleo de los mismos.

IV. Que al desarrollar la presente tesis dentro de la Universidad Nacional Autónoma de México, fue grato constatar que es sin duda, vanguardia en lo que a sistemas de cómputo se refiere, y es tarea de quienes nos involucramos de alguna manera en este rubro, el mantener y acrecentar esta primacía.

## APÉNDICE 1

**Descripción de las características del equipo de cómputo administrado por la Coordinación de Servicios de Red (CSR) de la DGSCA.**

Q	Modelo	Fabricante	Dirección Internet	Servicios
1	Sparc 1000	Sun Microsystems	servidor.unam.mx	Destinado al manejo de correo electrónico, cuenta actualmente con más de 7200 claves.
2	Sparc Center 2000	Sun Microsystems	fenix.dgsc.a.unam.mx serpiente.unam.mx	Ambos soportan por sus características de alto volumen de procesamiento y almacenamiento, los servicios de WWW, así como los servicios de búsqueda como Verónica, Judhead, Archie, Wais. También contiene los servicios de consultas hemerográficas y el de Phonebook.
2	Sparc Center 630 MP	Sun Microsystems	condor.dgsc.a.unam.mx  genesis.dgsc.a.unam.mx	Dedicado el primero al servicio de FTP anónimo, sesiones remotas y sustenta el servicio de Gopher de la UNAM, así como el servicio de impresión remota. El segundo es empleado para actividades de desarrollo y pruebas.
1	Sparc Classic	Sun Microsystems	bachiller.dgsc.a.unam.mx	Destinada a proporcionar servicios de correo electrónico y de información al sistema de bachillerato de la UNAM, cuenta con 550 usuarios de correo que accesan mediante el uso de clientes.

Debido a la importancia de los servicios que proporciona este último equipo, y a la difusión en estos niveles académicos, estos serán sujetos de modificaciones, a fin de incorporar los servicios de información básicos e incrementar su rendimiento. La propuesta de mejora se aborda en capítulos anteriores.

La coordinación cuenta habitualmente para su operación con Estaciones de trabajo Unix, terminales X y computadoras personales con ambiente X y total conectividad.

---

**APÉNDICE 2****A. Instituciones conectadas a RedUNAM en el Distrito Federal.**

1. American Computers S:A de C.V.
  2. Centro Internacional de Mejoramiento de maíz y trigo.
  3. Comisión Federal de Electricidad.
  4. Instituto Tecnológico Autónomo de México.
  5. Instituto Mexicano de Investigaciones Tecnológicas.
  6. Centro Nacional para la Prevención de Desastres.
  7. Tecnología Uno-Cero.
  8. Universidad de Norteamérica. Centro Lomas.
  9. Universidad Simón Bolívar.
  10. Universidad Iberoamericana.
  11. Universidad La Salle.
  12. Universidad Panamericana.
  13. DIF.
  14. Multix S.A de C.V.
  15. Tornad S.C.
  16. Bruno Unna.
  17. Banco de México.
  18. Colegio Miraflores.
  19. Centro Nacional de las Artes.
  20. Colegio de México.
  21. Instituto Mexicano del Petróleo.
  22. Televisa S.A. de C.V.
  23. Infolatina S.A de C.V.
  24. Centro de Investigaciones y Estudios Avanzados I.P.N.
  25. Centro de Tecnología Electrónica e Informática.
  26. Centro Nacional de Información y Documentación en la salud.
  27. Comisión Nacional para la Biodiversidad.
  28. Consorcio Red-Uno.
  29. Instituto Nacional de Pediatría.
  30. Secretaría de Comunicaciones y Transportes.
  31. Universidad Anahuac del Sur.
  32. Universidad Autónoma Metropolitana. Unidad Iztapalapa.
  33. Universidad Tecnológica de Netzahualcoyotl.
  34. Escuela Militar de Ingenieros.
  35. Comisión Nacional del Agua.
  36. ELAI S.C.
  37. TIC Comunicaciones.
  38. Programa de las Naciones Unidas para el Medio Ambiente.
  39. Supernet.
  40. Ingeniería de Software de Alta Calidad.
  41. Software de Alta Calidad.
- Academia Mexicana de Informática.

## APÉNDICE I

**Descripción de las características del equipo de cómputo administrado por la Coordinación de Servicios de Red (CSR) de la DGSCA.**

Q	Modelo	Fabricante	Dirección Internet	Servicios
1	Sparc 1000	Sun Microsystems	servidor.unam.mx	Destinado al manejo de correo electrónico, cuenta actualmente con más de 7200 claves.
2	Sparc Center 2000	Sun Microsystems	fenix.dgsc.unam.mx serpiente.unam.mx	Ambos soportan por sus características de alto volumen de procesamiento y almacenamiento, los servicios de WWW, así como los servicios de búsqueda como Verónica, Judhead, Archie, Wais. También contiene los servicios de consultas hemerográficas y el de Phonebook.
2	Sparc Center 630 MP	Sun Microsystems	condor.dgsc.unam.mx  genesis.dgsc.unam.mx	Dedicado el primero al servicio de FTP anónimo, sesiones remotas y sustenta el servicio de Gopher de la UNAM, así como el servicio de impresión remota.  El segundo es empleado para actividades de desarrollo y pruebas.
1	Sparc Classic	Sun Microsystems	bachiller.dgsc.unam.mx	Destinada a proporcionar servicios de correo electrónico y de información al sistema de bachillerato de la UNAM, cuenta con 550 usuarios de correo que accesan mediante el uso de clientes.

Debido a la importancia de los servicios que proporciona este último equipo, y a la difusión en estos niveles académicos, estos serán sujetos de modificaciones, a fin de incorporar los servicios de información básicos e incrementar su rendimiento. La propuesta de mejora se aborda en capítulos anteriores.

La coordinación cuenta habitualmente para su operación con Estaciones de trabajo Unix, terminales X y computadoras personales con ambiente X y total conectividad.

---

**APÉNDICE 2****A. Instituciones conectadas a RedUNAM en el Distrito Federal.**

1. American Computers S:A de C.V.
  2. Centro Internacional de Mejoramiento de maíz y trigo.
  3. Comisión Federal de Electricidad.
  4. Instituto Tecnológico Autónomo de México.
  5. Instituto Mexicano de Investigaciones Tecnológicas.
  6. Centro Nacional para la Prevención de Desastres.
  7. Tecnología Uno-Cero.
  8. Universidad de Norteamérica. Centro Lomas.
  9. Universidad Simón Bolívar.
  10. Universidad Iberoamericana.
  11. Universidad La Salle.
  12. Universidad Panamericana.
  13. DIF.
  14. Multix S.A de C.V.
  15. Tornad S.C.
  16. Bruno Unna.
  17. Banco de México.
  18. Colegio Miraflores.
  19. Centro Nacional de las Artes.
  20. Colegio de México.
  21. Instituto Mexicano del Petróleo.
  22. Televisa S.A. de C.V.
  23. Infolatina S.A de C.V.
  24. Centro de Investigaciones y Estudios Avanzados I.P.N.
  25. Centro de Tecnología Electrónica e Informática.
  26. Centro Nacional de Información y Documentación en la salud.
  27. Comisión Nacional para la Biodiversidad.
  28. Consorcio Red-Uno.
  29. Instituto Nacional de Pediatría.
  30. Secretaría de Comunicaciones y Transportes.
  31. Universidad Anahuac del Sur.
  32. Universidad Autónoma Metropolitana. Unidad Iztapalapa.
  33. Universidad Tecnológica de Netzahualcoyotl.
  34. Escuela Militar de Ingenieros.
  35. Comisión Nacional del Agua.
  36. ELAI S.C.
  37. TIC Comunicaciones.
  38. Programa de las Naciones Unidas para el Medio Ambiente.
  39. Supernet.
  40. Ingeniería de Software de Alta Calidad.
  41. Software de Alta Calidad.
- Academia Mexicana de Informática.



**Figura A.**

Diagrama de Instituciones conectadas a RedUNAM en el Distrito Federal.

**B. Instituciones conectadas a RedUNAM en el interior de la República.**

1. Universidad Autónoma de Querétaro.
2. Universidad de Guanajuato.
3. Universidad de Mexicali.
4. Universidad Autónoma de Ciudad Juárez.
5. Universidad Autónoma de Chiapas.
6. Universidad Autónoma de Guerrero.
7. Universidad de Quintana Roo.
8. Universidad de Sonora.
9. Benemérita Universidad de Puebla.
10. Centro de Investigación Científica y Educación Superior de Ensenada.
11. Instituto Nacional de Salud pública.
12. Instituto Nacional de Investigaciones Nucleares.
13. Universidad Autónoma de Veracruz.
14. Instituto de Investigaciones eléctricas.
15. Instituto Mexicano de Tecnología del agua.
16. Instituto Mexicano del Transporte.
17. Internet de Cancún.

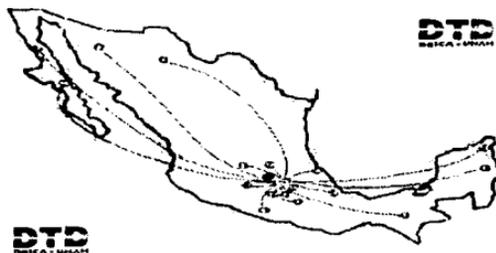
**Figura B**

Diagrama de instituciones conectadas a RedUNAM en el interior de la república.

---

**APÉNDICE 3**

**Detalle CGI's y su implementación.**

A continuación se incorpora el detalle de los CGI desarrollados para integrar el sistema de administración de cuentas de correo, su funcionamiento se detalla en el texto que los compone, es decir, en su documentación.

Estos CGI han sido desarrollados en lenguaje Perl, lenguaje nativo para funcionar como soporte funcional del lenguaje Html y del ambiente de WWW.

Los CGI son los siguientes:

**ReadParse.**

Subrutina que permite la entrada de argumentos a través de formás en ambiente html y WWW.

**alta.pl**

Genera nuevas cuentas en el servicio de correo electrónico.

**baja.pl**

Elimina cuentas existentes en el servicio de correo electrónico.

**consulta.pl**

Proporciona los datos de cuentas existentes en el servicio de correo electrónico .

**cambios.pl**

Realiza modificaciones en los datos de cuentas existentes en el servicio de correo electrónico.

**camblop.pl**

Genera una nueva contraseña de acceso para cuentas existentes en el servicio de correo electrónico.

```
##
## ReadParse
##
## Subrutina pública que permite la entrada de argumentos para un CGI
## a través del empleo de formás en ambiente html.
```

```
sub ReadParse (
```

```
    local (*in) = shift if @_; # CGI input
    local (*incfn, # Client's filename (may not be provided)
        *inct, # Client's content-type (may not be provided)
        *insfn) = @_; # Server's filename (for spooled files)
    local ($len, $type, $meth, $errflag, $cmdflag, $perlwarn, $got);
```

```
    # Disable warnings as this code deliberately uses local and environment
    # variables which are preset to undef (i.e., not explicitly initialized)
    $perlwarn = $^W;
    $^W = 0;
```

```
    binmode(STDIN); # we need these for DOS-based systems
    binmode(STDOUT); # and they shouldn't hurt anything else
    binmode(STDERR);
```

```
    # Get several useful env variables
    $type = $ENV{'CONTENT_TYPE'};
    $len = $ENV{'CONTENT_LENGTH'};
    $meth = $ENV{'REQUEST_METHOD'};
```

```
    if ($len > $cgi_lib'maxdata) { #
        &CgiDie("cgi-lib.pl: Request to receive too much data: $len bytes'n");
    }
}
```

```
if (!defined $meth || $meth eq "" || $meth eq 'GET' ||
    $type eq 'application/x-www-form-urlencoded') {
    local ($key, $val, $i);
```

```
    # Read in text
    if (!defined $meth || $meth eq "") {
        $in = $ENV{'QUERY_STRING'};
        $cmdflag = 1; # also use command-line options
    } elsif ($meth eq 'GET' || $meth eq 'HEAD') {
        $in = $ENV{'QUERY_STRING'};
    } elsif ($meth eq 'POST') {
        if (($got = read(STDIN, $in, $len)) != $len)
            {$errflag="Short Read: wanted $len, got $got'n"};
    } else {
```

```

    &CgiDie("cgi-lib.pl: Unknown request method: $meth\n");
}

@in = split(/&./,$in);
push(@in, @ARGV) if $cmdflag; # add command-line parameters

foreach $i (0 .. $#in) {
    # Convert plus to space
    $in[$i] =~ s/\+ / /g;

    # Split into key and value.
    ($key, $val) = split(/=/,$in[$i],2); # splits on the first =.

    # Convert %XX from hex numbers to alphanumeric
    $key =~ s/%{[A-Fa-f0-9]{2}}/pack("c",hex($1))/ge;
    $val =~ s/%{[A-Fa-f0-9]{2}}/pack("c",hex($1))/ge;

    # Associate key and value
    $in{$key} .= "0" if (defined($in{$key})); # \0 is the multiple separator
    $in{$key} .= $val;
}

} elsif ($ENV{'CONTENT_TYPE'} =~ m/^multipart/form-data#/) {
    # for efficiency, compile multipart code only if needed
    $errflag = !(eval <<"END_MULTIPART");

    local ($buf, $boundary, $head, @heads, $cd, $ct, $fname, $ctype, $blen);
    local ($bpos, $lpos, $left, $amt, $fn, $ser);
    local ($bufsize, $maxbound, $writefiles) = ($cgi_lib'bufsize, $cgi_lib'maxbound,
    $cgi_lib'writefiles);

    # The following lines exist solely to eliminate spurious warning messages
    $buf = "";

    ($boundary) = $type =~ /boundary="([^\"]+)"/; # find boundary
    ($boundary) = $type =~ /boundary=(S+)/ unless $boundary;
    &CgiDie ("Boundary not provided: probably a bug in your server")
    unless $boundary;
    $boundary = "--" . $boundary;
    $blen = length ($boundary);

    if ($ENV{'REQUEST_METHOD'} ne 'POST') {
        &CgiDie("Invalid request method for multipart/form-data: $meth\n");
    }

    if ($writefiles) {

```

```

local($me);
stat ($writefiles);
$writefiles = "/tmp" unless -d _ && -r _ && -w _;
# ($me) = $0 == m#([^\/*]*)$#;
$writefiles .= "/$cgi_lib$filepre";
}

$left = $len;
PART: # find each part of the multi-part while reading data
while (1) {
  die @$ if $errflag;

  $amt = ($left > $bufsize+$maxbound-length($buf)
    ? $bufsize+$maxbound-length($buf) : $left);
  $errflag = (($got = read(STDIN, $buf, $amt, length($buf))) != $amt);
  die "Short Read: wanted $amt, got $got\n" if $errflag;
  $left -= $amt;

  $in{$name} .= "\0" if defined $in{$name};
  $in{$name} .= $fn if $fn;

  $name=~/[^\-\/]+/; # This allows $insfn($name) to be untainted
  if (defined $1) {
    $insfn{$1} .= "\0" if defined $insfn{$1};
    $insfn{$1} .= $fn if $fn;
  }

  BODY:
  while (($bpos = index($buf, $boundary)) == -1) {
    die @$ if $errflag;
    if ($name) { # if no $name, then it's the prologue -- discard
      if ($fn) { print FILE substr($buf, 0, $bufsize); }
      else { $in{$name} .= substr($buf, 0, $bufsize); }
    }
    $buf = substr($buf, $bufsize);
    $amt = ($left > $bufsize ? $bufsize : $left); # $maxbound==length($buf);
    $errflag = (($got = read(STDIN, $buf, $amt, $maxbound)) != $amt);
    die "Short Read: wanted $amt, got $got\n" if $errflag;
    $left -= $amt;
  }
  if (defined $name) { # if no $name, then it's the prologue -- discard
    if ($fn) { print FILE substr($buf, 0, $bpos-2); }
    else { $in{$name} .= substr($buf, 0, $bpos-2); } # kill last \r\n
  }
  close (FILE);
  last PART if substr($buf, $bpos + $blen, 4) eq "--\r\n";
}

```

```

substr($buf, 0, $bpos+$blen+2) = "";
$amt = ($left > $bufsize+$maxbound-length($buf)
        ? $bufsize+$maxbound-length($buf) : $left);
$errflag = (($got = read(STDIN, $buf, $amt, length($buf))) != $amt);
die "Short Read: wanted $amt, got $got\n" if $errflag;
$left -= $amt;

undef $head; undef $fn;
HEAD:
while (($ipos = index($buf, "\n\n\n")) == -1) {
    die $@ if $errflag;
    $head .= substr($buf, 0, $bufsize);
    $buf = substr($buf, $bufsize);
    $amt = ($left > $bufsize ? $bufsize : $left); # $maxbound == length($buf);
    $errflag = (($got = read(STDIN, $buf, $amt, $maxbound)) != $amt);
    die "Short Read: wanted $amt, got $got\n" if $errflag;
    $left -= $amt;
}
$head .= substr($buf, 0, $ipos+2);
push (@in, $head);
@heads = split("\n\n", $head);
($cd) = grep (/^\s*Content-Disposition/i, @heads);
($ct) = grep (/^\s*Content-Type/i, @heads);

($name) = $cd =~ /bname="([^\"]+)"/i; #;
($name) = $cd =~ /bname=([\s:;]+)/i unless defined $name;

($fname) = $cd =~ /filename="([^\"]+)"/i; #; # filename can be null-str
($fname) = $cd =~ /filename=([\s:;]+)/i unless defined $fname;
$inctfn($name) .= (defined $in{$name} ? "\0" : "") . $fname;

($ctype) = $ct =~ /\s*Content-type:\s*"([^\"]+)"/i; #;
($ctype) = $ct =~ /\s*Content-Type:\s*"([^\s:;]+)"/i unless defined $ctype;
$inct($name) .= (defined $in{$name} ? "\0" : "") . $ctype;

if ($writefiles && defined $fname) {
    $serr++;
    $fn = $writefiles . ".$serr";
    open (FILE, ">$fn") || &CgiDie("Couldn't open $fn\n");
    binmode (FILE); # write files accurately
}
substr($buf, 0, $ipos+4) = "";
undef $name;
undef $ctype;
}

```

```
1:
END_MULTIPART
if ($errflag) {
    local ($errmsg, $value);
    $errmsg = $@ || $errflag;
    foreach $value (values %insfn) {
        unlink(split("\0", $value));
    }
    &CgiDie($errmsg);
} else {
    # everything's ok.
}
} else {
    &CgiDie("cgi-lib.pl: Unknown Content-type: $ENV{'CONTENT_TYPE'}\n");
}

# no-ops to avoid warnings
$insfn = $insfn;
$incfn = $incfn;
$inct = $inct;

$^W = $perlwarn;

return ($errflag ? undef : scalar(@in));
}
```

```

#!/usr/local/perl/bin/perl
require "/infoss/blitz/httpd/cgi-bin/cgi-lib2.pl";
&ReadParse;

## Programa: alta.pl
##
## Acción: Entrada de nuevos usuarios del servicio de correo electrónico
##
## Argumentos:
##
## $Login
## $Nombre
## $Plantel
## $Tipo          (alumno/profesor)
## $cta_resp      (numero de cuenta/responsable)
## $Telefono
## $Shell         (csh, sh, ninguno)

## Inicializacion de variables

$Dir      = "/etc";
$Mail     = "/usr/var/mail";
$Encrypta = "$Dir/encrypta"; ## binario que genera un password y su
                          ## correspondiente encriptacion
$Passwd   = "$Dir/passwd";
$Temporal = "$Dir/temp";
$shell    = "/bin/false";
$semaforo = "$Dir/flag";

$Login    = ${Login};
$Nombre   = ${Nombre};
$Plantel  = ${Plantel};
$Tipo     = ${Tipo};
$cta_resp = ${cta_resp};
$Telefono = ${Telefono};
$Shell    = ${Shell};
$Pagina   = ${Pagina}; # No se utiliza este valor

sub alta_entrada {
    print "Content-type: text/html\n\n";
    print <<EOH;

    <html>
    <head><title>Nueva Entrada</title></head>

```

```
<body><h1>Sistema de Correo Electrónico <br>
Nivel Bachillerato <br>
$Mensaje</h1>
```

```
<form method="POST" action="/cgi-bin/alta.pl">
```

```
Nombre: <input type="text" name="Nombre" size="40" value=$Nombre> <br>
```

```
No. de cuenta o Responsable: <input type="text" name=cta_resp
```

```
value=$cta_resp><br>
```

```
Plantel: <input type="text" name="Plantel" size="10" value=$Plantel>
```

```
Telefono: <input type="text" name="Telefono" size="10"
```

```
value=$Telefono><br>
```

```
Tipo de usuario: <select name="Tipo">
```

```
<option selected value="alumno">Alumno
```

```
<option value="profesor">Profesor
```

```
</select><br><br><br>
```

```
Login: <input type="text" name="Login" size="8">
```

```
Ambiente: <select name="Shell">
```

```
<option value="/bin/sh">Bourn Shell
```

```
<option selected value="/bin/csh">C-Shell
```

```
<option value="/bin/false">Ninguno
```

```
</select> <br>
```

```
<input name="Pagina" type="hidden" value=1>
```

```
<br> <br>
```

```
<input type="submit" value="Continuar">
```

```
<input type="reset" value="Limpia forma">
```

```
</form>
```

```
</body>
```

```
</html>
```

```
EOH
```

```
exit(0);
```

```
}
```

```
sub alta_login {
```

```
local($flag_esta);
```

```
## Verificar que el login no exista en el archivo /etc/passwd
```

```
## Si ya existe, solicitar nuevo login
```

```
open (PASSWD,$Passwd) || warn "no puedo abrir el archivo";
while (<PASSWD){

    if ($_ =~ /^$Login/) {

$Mensaje="<br>El login propuesto ya existe, favor de cambiarlo<br><br><br>";
&alta_entrada;

##    el login ya existe
    }

}
close (PASSWD);

## $flag_esta;
}

sub alta_contraseña {

    ## Generar el password mediante el programa "encripta"
    ## y enviarlo a un archivo temporal.
    ## Despues separar el password de la encripcion

    system ("`$Encripta > $Temporal` ");

    open (TEMP,$Temporal) || warn "no puedo abrir el archivo";
    while (<TEMP>) {
        $Hola=$_;
    }
    close(TEMP);

    system ("rm -f $Temporal");

    $Contraseña=substr($Hola, 0, 8);
    $Encripcion=substr($Hola, 14, 12);

}

sub alta_ocupado {

print "Content-type: text/html\n\n";
print <<ADIOS;

<html>
<head><title>SISTEMA EN USO</title></head>
```

---

<h2> SISTEMA EN USO INTENTE SEGUNDO MÁS TARDE </h2>

```
<form method="POST" action="/cgi-bin/alta.pl">
<input type="submit" value="continuar">
</form>
```

```
</html>
ADIOS
```

```
exit(0);
}
```

```
sub alta_alta {
```

```
    &alta_login;
    &alta_contrasena;
```

```
##Ver si el /etc/passwd esta abierto, si lo esta no sigue el shell
## Para implementar control de concurrencia
```

```
if ( -e $semaforo ) {
    #print "Archivo en uso, continua segundos más tarde \n";
```

```
    &alta_ocupado;
```

```
} else {
```

```
    system ("touch $semaforo");
```

```
## obtener ultima linea del /etc/passwd y obtener campos
```

```
open(PASSWD, "$Passwd");
```

```
while ($p = <PASSWD>) {
    chop $p;
    $ultima=$p;
```

```
}
#print "ultima linea --> $ultima \n";
@linea = split(/:/,$ultima);
```

```
@comments=split(/./,$linea[4]);
```

```
$nombre=$comments[0];
$plantele=$comments[1];
```

```
$home=$linea[5];
```

```
##Calcular nuevo UID
```

```
$nuevo_uid=1+$linea[2];
close(PASSWD);

## Crear el home directory
mkdir ("${Dir}/${Login}.0700")|| die "no puedo crear directorio";

## Agregar el registro al archivo /etc/passwd
open(PASSWD,">>${Passwd}");

print PASSWD "${Login}: ${Encriptcion}: $nuevo_uid: 20: $Nombre, $Tipo, $cta_resp,
$Telefono, $Plantele: ${Dir}/${Login}/: $Shell\n";

close(PASSWD);

system ("rm -f $semaforo");
&alta_ver;
}

sub alta_ver {
print "Content-type: text/html\n\n";
print <<WEB_PAGE;

<html>
<head><title>Nueva clave</title></head>

<h1>CLAVE DADA DE ALTA</h1>
<h2>

Login:           $Login <br>
Password:        $Contrasenia <br>
Nombre:          $Nombre <br>
Home directory:  $Dir/${Login} <br>
Shell:           $Shell <br>

</h2>
<form method="POST" action="/cgi-bin/alta.pl">
<input type="submit" value="Continuar">
</form>

</html>
WEB_PAGE
exit(0);
}
```

```
unless ($Nombre && $Pantel && $Telefono && $Login && $Shell)
{
    $Pagina=1;
    &ata_entrada;
}
&ata_ata;
```

```
#!/usr/local/perl/bin/perl
&ReadParse;

## Programa: baja.pl
##
## Acción: Elimina cuentas de usuarios de correo electrónico
##
## Argumentos:
## login

## Archivos y Directorios
$Dir      = "/etc";
$Passwd   = "$Dir/passwd";
$Var      = "/usr/var/Mail";
$baja_log = "$Dir/baja.log";
$temp     = "$Dir/tempo";
$semaforo = "$Dir/flag";

# variables
$Login    = $in('Login');
$Pagina   = $in('Pagina');

sub baja_login {
    print <<EOH;
    Content-type: text/html

    <html>
    <head><title>BAJA DE CLAVE</title></head>

    <body><h1>Sistema de Correo Electrónico de cuentas de correo<br>
    Eliminación de cuentas de correo</h1>

    <form method="POST" action="/cgi-bin/baja.pl">

        Login: <input type="text" name="Login" size="8" value=$Login> <br><br>
        Nombre: $Nombre <br><br>
        Plantel: $Plantel <br><br>
        Responsable/No_cta: $no_cta <br><br>
        Home directory: $Home <br><br>
        Shell: $Shell <br><br>

        <input name="Pagina" type="hidden" value=$Pagina>

        <input type="submit" value="Continuar">
        <input type="reset" value="Limpiar forma">
    </form>
```

```
</body>
</html>
EOH
exit(0);
}
```

```
sub baja_localiza {
```

```
# Hacer el grep para localizar línea que inicie con ese login
```

```
$si=0;
open (PASSWD, $Passwd) || warn "no puedo abrir el archivo";
while (<PASSWD>) {
  if ($_ =~ /^$Login/) {
    #print $_;
    @locate = split(/./,$_);
    $si = 1;
```

```
# consulta datos para verificar baja
@comments = split(/./,$locate[4]);
```

```
$Nombre = $comments[0];
$Login = $locate[0];
$Plantel = $comments[4];
$no_cta = $comments[2];
$Home = $locate[5];
$Shell = $locate[6];
```

```
$name = $comments[0];
```

```
close(<PASSWD>);
$Pagina = 3;
##&consulta_error;
&baja_login;
```

```
} #fin del if
```

```
} #fin del while
```

```
close(<PASSWD>);
```

```
if ($si == 0) {
  # Reporte: El login no existe.
  $Pagina = 1;
  $Login="";
  #&baja_login;
```

```
&baja_error;
##&consulta_error;
}

} # fin baja_localiza

sub confirma_baja {
    if ($$i_esta != 1){
# print "Imposible eliminar, el login $Login no existe \n";
    } else {
# DESEA RELIZAR LA BAJA
    }
} #fin confirma_baja

sub consulta_error {
    print "Content-type: text/html\n\n";
    print <<EOH;

    <html>
    <head><title>CONFIRMAR DATOS</title></head>

<body><h1>Sistema de Correo Electrónico <br>
Consulta </h1>

<h2> CLAVE $Login INEXISTENTE REALIZE NUEVA BUSQUEDA $Nombre </h2>

<form method="POST" action="/cgi-bin/baja.pl">

    <input type="submit" value="Continuar">
</form>

</body>
</html>
EOH
exit(0);
}

sub baja_error {
    print "Content-type: text/html\n\n";
    print <<EOH;
```

```
<html>
<head><title>BAJA</title></head>
<body><h1>Sistema de Correo Electrónico <br>
Consulta </h1>
<h2> LA CLAVE $Login NO EXISTE</h2>
<form method="POST" action="/cgi-bin/baja.pl">
  <input type="submit" value="Continuar">
</form>
</body>
</html>
EOH
exit(0);
}
```

```
sub actualiza {
```

```
  ## Funcion que actualiza el archivo /etc/passwd
```

```
  open (PASSWD,">$Passwd") || warn "no puedo abrir el archivo";
  open (TEMP,$temp) || warn "no puedo abrir el archivo";
```

```
  while (<TEMP>) {
    print PASSWD $_;
  }
  close (<TEMP>);
  close (<PASSWD>);
}
```

```
sub baja_baja {
```

```
  ## Bloquea el archivo /etc/passwd con un semaforo
```

```
  while ( -e $semaforo ) {}
  system ("touch $semaforo");
```

```
  if ( -e $temp ) {
    system ("rm -f $temp");
  }
```

```
  system ("touch $temp");
  system ("rm -fr /$Dir/$Login");
```

---

**#eliminar la linea del /etc/passwd**

```
open (PASSWD,$Passwd) || warn "no puedo abrir el archivo";
open (TEMP,">$temp") || warn "no puedo abrir el archivo";
```

```
while (<PASSWD>) {
  if ($_ =~ /^$Login/) {
  }
  else {
    print TEMP "$_";
  }
}
```

```
close (<TEMP>);
close (<PASSWD>);
```

```
&actualiza;
system ("touch $baja_log");
system ("echo $Login,'date', $name >> $baja_log");
```

```
## Elimina el semaforo
system ("rm -f $semaforo");
system ("rm -f $temp");
```

```
## Elimina directorio de usuario /home/users/$Login;
system ("rmdir -f /home/users/$Login");
```

```
## Elimina directorio de usuario /home/users/$Login;
system ("rmdir -f /var/mail/$Login");
```

```
&baja_efectuada;
```

```
) #fin de baja
```

```
sub baja_efectuada {
```

```
  print "Content-type: text/html\n\n";
  print <<EOH;
```

```
  <html>
  <head><title>BAJA</title></head>
```

```
<body><h1>Sistema de Correo Electrónico<br>
```

```
Consulta </h1>
<h2> CLAVE DADA DE BAJA</h2>

<form method="POST" action="/cgi-bin/baja.pl">

  <input type="submit" value="Continuar">
</form>

</body>
</html>
EOH
exit(0);
}

sub baja_resultado {
  print <<EOH;
  Content-type: text/html

  <html>
  <head><title>Login a dar de Baja</title></head>

  <body><h1>Clave $Login dada de Baja</h1>
  </html>
  EOH
  exit(0);
}

unless ($Login)
{
  $Pagina=1;
  &baja_login;
}

If ( $Pagina == 3 ) {

  &baja_baja;
  &aconsulta_error;
}

$Pagina=2;
&baja_localiza;
```

```
#!/usr/local/perl/bin/perl
require "/inboss/blitz/httpd/cgi-bin/cgi-lib2.pl";
&ReadParse;

## Programa consulta.pl
##
## Acción: realiza una consulta de la información correspondiente a un usuario en base a
su login.
##
## Argumentos:
## login_s
##

$pwd      = "/etc";
$Passwd   = "$Dir/passwd;

#verificar si hay otro shell corriendo
$existe = "$pwd/flag";

# Variables.

$login    = ${'login'};
$Nombre   = ${'Nombre'};
$Plantel  = ${'Plantel'};
$no_cta   = ${'no_cta'};
$Home     = ${'Home'};
$Shell    = ${'Shell'};

sub consulta_login {
    print "Content-type: text/html\n\n";
    print <<EOH;

    <html>
    <head><title>Consulta</title></head>

<body><h1>Sistema de Correo Electrónico<br>
    Consulta </h1>

    <form method="POST" action="/cgi-bin/consulta.pl">
    Login: <input type="text" name="login" size="8" value=$login>
<br><br>
    Nombre: $Nombre <br><br>
    Plantel: $Plantel <br><br>
    Responsable/No_cta: $no_cta <br><br>
    Home directory: $Home <br><br>
    Shell: $Shell <br><br>
```

```
<input type="submit" value="Continuar">
<input type="reset" value="Limpiar forma">
</form>

</body>
</html>
EOH
exit(0);
}

sub consulta_ver {
print "Content-type: text/html\n\n";
print <<WEB_PAGE;

<html>
<head><title>Visualizacion de datos</title></head>

<h1>RESULTADO DE CONSULTA</h1>
<h2>

Nombre:                $comments[0] <br>
Login:                 $locate[0] <br>
Plantol:              $comments[4] <br>
Responsable/No_cta:   $comments[2] <br>
Home directory:      $locate[5] <br>
Shell:                $locate[6] <br>

</h2>

<form method="POST" action="/cgi-bin/consulta.pl">
<input type="submit" value="Continuar">
</form>

</html>
WEB_PAGE
exit(0);
}

sub consulta_consulta {
#Hacer el grep para localizar linea que inicie con ese login

open (PASSWD,$Passwd) || warn "no puedo abrir el archivo";

## $patron = $login;
$si = 0;
```

```
while (<PASSWD>){
  if ( $_ =~ /^$login/ ) {
    #print $_ ;
    @locate = split(/:/,$_);
    $si = 1;

    #CONSULTA DE DATOS

    @comments = split(/,/, $locate[4]);
    $Nombre = $comments[0];
    $Login = $locate[0];
    $Plante1 = $comments[4];
    $no_cta = $comments[2];
    $Home = $locate[5];
    $Shell = $locate[6];

    $name = $comments[0];

    close(<PASSWD>);

    system ("rm -f $existe");

    &consulta_ver;
  }
}

if ( $si == 0 ) {
  &consulta_error;
}

} #fin de la funcion

sub consulta_error {

  ## Mensaje de error si la clave no existe

  print "Content-type: text/html\n\n";
  print <<EOH;

  <html>
  <head><title>Consulta</title></head>

  <body><h1>Sistema de Correo Electrónico <br>
  Consulta </h1>
```

---

```
<h2> CLAVE INEXISTENTE REALIZE NUEVA BUSQUEDA</h2>
<form method="POST" action="/cgi-bin/consulta.pl">
  <input type="submit" value="Continuar">
</form>

</body>
</html>
EOH
exit(0);
}

unless ($login)
{
  &consulta_login;
}

&consulta_consulta;
```

```

#!/usr/local/perl/bin/perl

&ReadParse

## Programa: cambios.pl
##
## Acción: Realiza cambios en la información de usuarios de correo electrónico.
##
## Argumentos:
## login

## Inicialización de variables

$Dir      = "/etc";
$Encripta = "$Dir/encrypta";
$Passwd   = "$Dir/passwd";
$temp     = "$Dir/tempo";
$shell    = "/bin/false";
$semaforo = "$Dir/flag";
$Mensaje  = "";

$Login    = $in{'Login'};
$Nombre   = $in{'Nombre'};
$Plantel  = $in{'Plantel'};
$Tipo     = $in{'Tipo'};
$cta_resp = $in{'cta_resp'};
$Telefono = $in{'Telefono'};
$Shell    = $in{'Shell'};
$Pagina   = $in{'Pagina'};    # No se utiliza este valor

$Encripcion = $in{'Encripcion'};
$uid        = $in{'uid'};
$gid        = $in{'gid'};

sub cambios_entrada {
  print "Content-type: text/html\n\n";
  print <<EOH;

  <html>
  <head><title>CAMBIOS EN CLAVE</title></head>

  <body><h1>Sistema de Correo Electrónico <br>
  CAMBIO DE INFORMACION EN CUENTAS DE CORREO<br>
  $Mensaje</h1>

```

```

<form method="POST" action="/cgi-bin/cambios.pl">
  Login: $Login <input type=$Tipo_login name="Login" size="8"
value="$Login"><br><br>
  Nombre: <input type=$Tipo_campo name="Nombre" size="40" value="$Nombre">
<br>
  No. de cuenta o Responsable: <input type=$Tipo_campo name=cta_resp
value="$cta_resp"><br>
  Plantel: <input type=$Tipo_campo name="Plantel" size="10"
value="$Plantel"><br>
  Telefono: <input type=$Tipo_campo name="Telefono" size="10"
value="$Telefono"><br>
  Tipo de usuario: <select name="Tipo">
  <option $tipo_alumno value="alumno">Alumno
  <option $tipo_profesor value="profesor">Profesor
</select><br><br>
  Ambiente: <select name="Shell">
  <option $tipo_sh value="/bin/sh">Bourn Shell
  <option $tipo_csh value="/bin/csh">C-Shell
  <option $tipo_ninguno value="/bin/false">Ninguno
</select> <br>
  <input name="Pagina" type="hidden" value="$Pagina">
  <input name="Encripcion" type="hidden" value="$Encripcion">
  <input name="uid" type="hidden" value="$uid">
  <input name="gid" type="hidden" value="$gid">
  <br> <br>
  <input type="submit" value="Continuar">
  <input type="reset" value="Limpiar forma">
</form>
</body>
</html>
EOH
exit(0);
}

sub cambios_localiza {
  # Hacer el grep para localizar linea que inicie con ese login
  $si=0;

```

```
open (PASSWD, $Passwd) || warn "no puedo abrir el archivo";
while (<PASSWD>) {
  if ( $_ =~ /^$Login/ ) {
    #print $_;
    @locate = split(/./,$_);
    $si = 1;

    # consulta datos para verificar cambios
    @comments = split(/./,$locate[4]);

    $Nombre = $comments[0];
    #$Login = $locate[0];
    $Plante1 = $comments[4];
    $cta_resp = $comments[2];
    $Tipo= $comments[1];
    if ($Tipo=~ "alumno"){
      $tipo_alumno="selected";
      $tipo_profesor="";
    }
    else{
      $tipo_alumno="";
      $tipo_profesor="selected";
    }
    $Telefono=$comments[3];
    $Home = $locate[5];
    $Shell = $locate[6];
    if ($Shell=~ "/bin/csh"){
      $tipo_csh="selected";
      $tipo_sh="";
      $tipo_ninguno="";
    }
    else{
      if ($Shell=~ "/bin/sh"){
        $tipo_csh="";
        $tipo_sh="selected";
        $tipo_ninguno="";
      }
      else {
        $tipo_csh="";
        $tipo_sh="";
        $tipo_ninguno="selected";
      }
    }
  }
}

$name = $comments[0];
$Encripcion=$locate[1];
```

---

```
$uid=$locate[2];
$gid=$locate[3];

close(<PASSWD>);
$Pagina = 3;
&cambios_entrada;

} #fin del if
} #fin del while
close(<PASSWD>);

if ($si == 0) {
  # Reporte: El login no existe.
  $Pagina = 1;
  $Login="";
  #&cambios_login;
  &consulta_error;

}

} # fin baja_localiza

sub confirma_baja {

  if ($si_esta != 1){
    # Login no existe
  } else {

    # desea realizar la baja
  }

} #fin confirma_baja

sub consulta_error {

  print "Content-type: text/html\n\n";
  print <<EOH;

  <html>
  <head><title>BAJA</title></head>
```

```

<body><h1>Sistema de Correo Electrónico <br>
Consulta </h1>

<h2> LA CLAVE $Login NO EXISTE</h2>

<form method="POST" action="/cgi-bin/cambios.pl">

  <input type="submit" value="Continuar">
</form>

</body>
</html>
EOH
exit(0);
}

```

**sub actualiza {**

```

open (PASSWORD,">$Passwd") || warn "no puedo abrir el archivo";
open (TEMP,$temp) || warn "no puedo abrir el archivo";

while (<TEMP>) {
  print PASSWORD $_;
}
close (<TEMP>);
close (<PASSWORD>);
}

```

**sub cambio\_cambio {**

```

## Bloquea el archivo /etc/passwd con un semaforo
while ( -e $semaforo ) {}
system ("touch $semaforo");

if ( -e $temp ) {
  system ("rm -f $temp");
}

system ("touch $temp");

##eliminar la linea del /etc/passwd

open (PASSWORD,$Passwd) || warn "no puedo abrir el archivo";
open (TEMP,">$temp") || warn "no puedo abrir el archivo";

```

```
while (<PASSWD>) {
  if ($_ == /^$Login/) {
    print TEMP
"$Login:$Encriptcion:$uid:$gid:$Nombre,$Tipo,$cta_resp,$Telefono,$Plantel:$Dir/$Login/:
$Shell\n";
  }
  else {
    print TEMP "$_";
  }
}

close (<TEMP>);
close (<PASSWD>);

&actualiza;
system ("touch $cambios_log");
system ("echo $l.ogin,"date",$name >> $cambios_log");

## Elimina el semaforo
system ("rm -f $semaforo");
system ("rm -f $temp");

&cambio_efectuado;
} #fin de cambios

sub cambio_efectuado {
  print "Content-type: text/html\n\n";
  print <<EOH;

  <html>
  <head><title>BAJA</title></head>

<body><h1>Sistema de Correo Electrónico<br>
Consulta </h1>
<h2>CAMBIOS EFECTUADOS</h2>

<form method="POST" action="/cgi-bin/cambios.pl">

  <input type="submit" value="Continuar">
</form>
```

```
</body>
</html>
EOH
exit(0);
}

sub baja_resultado {
  print "Content-type: text/html\n\n";
  print <<EOF;

  <html>
  <head><title>Login a dar de Baja</title></head>

  <body><h1>Clave $Login dada de Baja</h1>
  </html>
  EOF
  exit(0);
}

unless ($Login)
{
  $Pagina=1;
  $Tipo_login = "text";
  $Tipo_campo = "hidden";
  $Tipo_alumno = "selected";
  $Tipo_profesor = "";
  &cambios_entrada;
}

$Tipo_login = "hidden";
$Tipo_campo = "text";

if ( $Pagina == 3) {
  &cambio_cambio;
}

$Pagina=2;
&cambios_localiza;
```

---

```
#!/usr/local/perl/bin/perl
```

```
## Programa: cambiop.pl
```

```
##
```

```
## Acción: Genera una nueva contraseña para claves de correo.
```

```
##
```

```
## Argumentos:
```

```
## login
```

```
&ReadParse;
```

```
# Archivos y Directorios
```

```
$Dir      = "/etc";  
$Passwd  = "$Dir/passwd";  
$Var      = "/usr/var/Mail";  
$baja_log = "$Dir/baja.log";  
$Temp     = "$Dir/tempo";  
$Temporal = "$Dir/temporal";  
$semaforo = "$Dir/flag";  
$Encripta = "$Dir/encripta";
```

```
# variables
```

```
$Login    = ${'Login'};  
$Pagina   = ${'Pagina'};
```

```
sub cambiop_login {
```

```
    print <<EOH;  
    Content-type: text/html
```

```
    <html>  
    <head><title>CAMBIO DE PASSWORD</title></head>
```

```
    <body><h1>Sistema de Correo Electrónico <br>  
    Cambio de Password</h1>
```

```
    <form method="POST" action="/cgi-bin/cambiop.pl">
```

```
        Login: <input type="text" name="Login" size="8" value=$Login> <br><br>  
        Nombre: $Nombre <br><br>  
        Plantel: $Plantel <br><br>  
        Responsable/No_cta: $no_cta <br><br>  
        Home directory: $Home <br><br>
```

```

Shell: $Shell <br><br>

<input name="Pagina" type="hidden" value=$Pagina>

<input type="submit" value="Continuar">
<input type="reset" value="Limpiar forma">
</form>

</body>
</html>
EOH
exit(0);
}

sub cambiop_localiza {

# Hacer el grep para localizar línea que inicie con ese login

$si=0;
open (PASSWD, $Passwd) || warn "no puedo abrir el archivo";
while (<PASSWD>) {
  if ( $_ =~ /^$Login/ ) {
    #print $_;
    @locate = split(/./,$_);
    $si = 1;

    # consulta datos para verificar baja
    @comments = split(/./,$locate[4]);

    $Nombre = $comments[0];
    $Login = $locate[0];
    $Plante1 = $comments[4];
    $no_cta = $comments[2];
    $Home = $locate[5];
    $Shell = $locate[6];

    $name = $comments[0];

    close(<PASSWD>);
    $Pagina = 3;
    ##&consulta_error;
    &cambiop_login;

  } #fin del if
} #fin del while

```

---

```
close(<PASSWD>);

if ($$i =~ 0) {
    # Reporte: El login no existe.
    $Pagina = 1;
    $Login="";
    &baja_error;
}
} # fin cambiop_localiza

sub confirma_cambiop {
    if ($$i_esta != 1){
        # print "Imposible eliminar, el login $Login no existe \n";
    } else {
        # desea rellzar la baja
    }

    ) #fin confirma_camblop

sub consulta_error {
    print "Content-type: text/html\n\n";
    print <<EOH;

    <html>
    <head><title>CONSULTA DE DATOS PARA CONFIRMAR </title></head>
<body><h1>Sistema de Correo Electrónico</br>
    Consulta </h1>

    <h2> CLAVE $Login INEXISTENTE REALIZE NUEVA BUSQUEDA $Nombre </h2>

    <form method="POST" action="/cgi-bin/cambiop.pl">
        <input type="submit" value="Continuar">
    </form>

</body>
</html>
```

---

```
EOH
exit(0);
```

```
}
```

```
sub genera_password {
```

```
system ("`$Encripta > $Temporal` ");
```

```
open(TEMP,$Temporal) || warn "no puedo abrir el archivo";
```

```
while(<TEMP>) {
```

```
    $Hola=$_;
```

```
}
```

```
close(TEMP);
```

```
$Password=substr($Hola, 0, 8);
```

```
$Encripcion=substr($Hola, 14, 12);
```

```
}
```

```
sub actualiza {
```

```
open (PASSWD,">$Passwd") || warn "no puedo abrir el archivo";
```

```
open (TEMP,"<$temp") || warn "no puedo abrir el archivo";
```

```
while (<TEMP>) {
```

```
    print PASSWD $_;
```

```
}
```

```
close (<TEMP>);
```

```
close (<PASSWD>);
```

```
}
```

```
sub cambiop_cambiop {
```

```
## Bloquea el archivo /etc/passwd con un semaforo
```

```
while ( -e $semaforo ) {}
```

```
system ("touch $semaforo");
```

```
&genera_password;
```

```
if ( -e $temp ) {
```

```
    system ("rm -f $temp");
```

```
}
```

```
system ("touch $temp");
```

---

```

#Cambiar el campo encriptacion
open (PASSWD,"<Passwd") || warn "no puedo abrir el archivo";
open (TEMP,">$temp") || warn "no puedo abrir el archivo";

while (<PASSWD>) {
  if ($_ =~ /^$Login/) {
    @linea = split(/:/,$_);
    $log=$linea[0];
    $comments=$linea[4];
    $uid=$linea[2];
    $home=$linea[5];
    $shell=$linea[6];

#escribe una coma al final del archivo

    printTEMP"$log:$Encriptacion:$uid:20:$comments:$home:$shell";
  }
  else {
    print TEMP "$_";
  }
}

close (<TEMP>);
close (<PASSWD>);

### cambiar archivos move
&actualiza;

system ("touch $cambio.password.log");
system ("echo $Login,'date', $name >> $cambio.password.log");

## Elimina el semaforo
system ("rm -f $semaforo");
system ("rm -f $temp");

&cambiop_efectuado;

} #fin de baja

sub cambiop_efectuado {
  print "Content-type: text/html\n\n";
  print <<EOH;

```

---

```

<html>
<head><title>CAMBIO DE PASSWORD</title></head>

<body><h1>Sistema de Correo Electrónico <br>
Consulta </h1>
<h2> NUEVO PASSWORD PARA $Login ES : $Encipcion</h2>

<form method="POST" action="/cgi-bin/cambiop.pl">

  <input type="submit" value="Continuar">
</form>

</body>
</html>
EOH
exit(0);
}

sub cambiop_resultado {
print <<EOH;
Content-type: text/html

<html>
<head><title>Login a dar de Baja</title></head>

<body><h1>Nuevo password para $Login es:$Password</h1>
</html>
EOH
exit(0);
}

unless ($Login)
{
  $Pagina=1;
  &cambiop_login;
}

if ( $Pagina == 3) {
&cambiop_cambiop;
}

$Pagina=2;
&cambiop_localiza;

```

---

**APÉNDICE 4****Manual de Operación de las formás a través de WWW.**

Como se mencionó en apartados anteriores, la operación del sistema propuesto se hará a través de formás en ambiente **WWW**, estas formás interactúan con los cgi, de tal forma que permiten al usuario una interfaz ilustrativa de las acciones que se realizan con las cuentas de correo electrónico, tienen la característica de ser de fácil manejo y de ser ilustrativas de la acción que realizan.

De esta manera solo los administradores locales en cada plantel podrán realizar consistentemente las siguientes cinco acciones : **altas, bajas, cambios, consultas y cambios de contraseña**, ya que como se explicó en capítulos anteriores, serán los únicos que tendrán la facultad de accederlas.

**Altas.**

En esta forma es necesario proporcionar los datos siguientes:

- Nombre.
- Número de cuenta o responsable.
- Plantel.
- Teléfono.
- Tipo de usuario. Cuenta con un botón de despliegue para elegir opciones predefinidas.
- Login sugerido.
- Tipo de shell deseado. Cuenta con un botón de despliegue para elegir opciones predefinidas.

**Formulario de Correo Electrónico Nivel Bachillerato**

Nombre: [Redacted]

Apellido: [Redacted]

Teléfono: [Redacted]

Ambiente: [Redacted]

Continuar Limpia forma

En caso de que el login propuesto tenga ya una ocurrencia en el sistema, se presentará nuevamente la forma con los datos anteriores, donde es posible elegir otra opción de login; de lo contrario son mostrados los datos correspondientes al nuevo usuario, como son: **login, nombre, directorio home, tipo de shell y password**, este último es la única vez que el sistema lo muestra por razones de seguridad.

**PANTALLA DE ALTA**

Nombre: [Redacted]

Apellido: [Redacted]

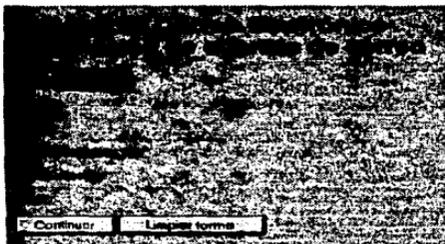
Teléfono: [Redacted]

Ambiente: [Redacted]

Continuar

### Bajas.

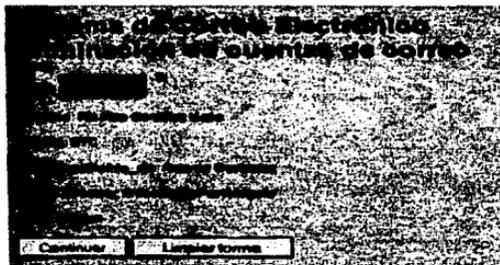
Esta forma permite eliminar cuentas existentes de correo electrónico con solo proporcionar el login correspondiente a la misma. El administrador proporcionar el login correspondiente al usuario que desea eliminar, hecho lo anterior podrá consultar los datos correspondientes al mismo, a fin de ratificar la eliminación de forma definitiva.



En caso contrario, si la clave que se pretende eliminar no existe en el sistema, aparecerá un aviso, tras el cual es posible repetir la operación inicial.



Si la clave a eliminar efectivamente existe, es desplegada su información, a fin de ratificar la baja, una vez hecho esto, la clave es eliminada del sistema.

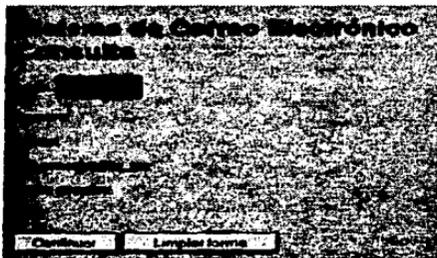


Esta baja es notificará al administrador mediante un mensaje alusivo.



### Consultas.

Esta opción permite consultar la información representativa de una cuenta de correo electrónico, a fin de conocer los datos del usuario, es también empleada por las demás formás, con el fin de mostrar la información a modo de confirmación. Antes de realizar alguna otra acción, como baja o cambios, es importante señalar que **no incluye consulta a la contraseña de la cuenta**, por razones evidentes de seguridad.



Para llevarla a cabo es necesario proporcionar el login de la clave que pretendemos consultar. Si no existe, el sistema enviará un mensaje alusivo.

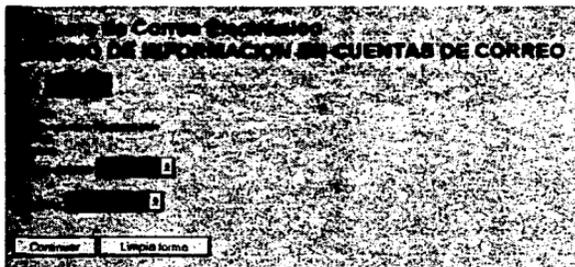


De los contrario, mostrará la consulta correspondiente.

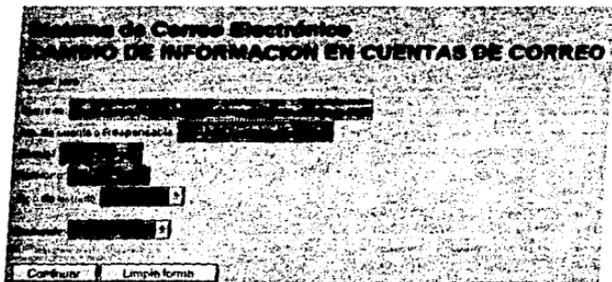


### Cambios.

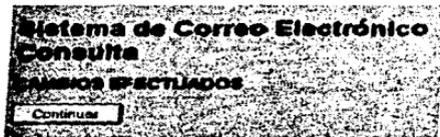
A través de esta forma es como podemos modificar la información correspondiente a un usuario del sistema de correo electrónico. En la forma correspondiente a la opción de cambios debemos proporcionar el login del usuario sujeto a modificación de información.



De igual forma que altas y bajas, en cambios, si la clave ya existe o no forma parte del sistema, aparecerá un mensaje que lo notifique y permita realizar la operación nuevamente. Si la clave existe se desplegará su correspondiente información, y es en la misma forma desplegada donde es posible editar los campos que son mostrados.



Una vez realizados los cambios, deben ser confirmados; es entonces cuando el sistema despliega un aviso de realización de los cambios.



#### **Cambio de contraseña.**

Dado que la contraseña no se incluye en la opción de consulta, este módulo ha sido implementado por separado, con el fin de que en caso de extravío de contraseña se genere una nueva, sin tener que conocer la anterior, es decir, se asigna una nueva contraseña de forma incondicional, cabe señalar una vez más que las acciones anteriores, y en especial esta, solo las podrán llevar a cabo los administradores locales, ya que serán los únicos capaces de acceder a las formás.

Para realizar el cambio de contraseña, solo es necesario proporcionar el login de la clave deseada, en caso de no existir, se desplegará un mensaje emergente alusivo.



En caso contrario, se desplegará la correspondiente consulta, a modo de confirmación, tras la cual se lleva a cabo el cambio de contraseña, es en este momento cuando se despliega un aviso de la nueva contraseña, el cual será desplegado de forma única.



---

**BIBLIOGRAFIA**

**ANALISIS ESTRUCTURADO MODERNO**

Yourdon, Edward  
CA, E.U.A. 1992

**INTERNETWORKING WITH TCP/IP**

Comer, Douglas E.  
Prentice Hall, 2a.Edición.  
Englewood Hills, New Jersey, 1991.

**LEARNING PERL**

Wall, Larry & Schwartz, Randall  
O'Really Associates. Inc.  
CA, E.U.A. , 1992

**REDES DE COMPUTADORES**

Tanenbaum, Andrew S.  
Prentice Hall  
México, 1991

**SUN OS 4.1.3. AND OPEN WINDOWS VER.3**

Release Manual  
Sun Microsystems  
CA, E.U.A., 1992

**UNIX SIN FRONTERAS**

Hahn, Harley  
McGraw Hill  
México, 1995

---

**REFERENCIAS DE SITIOS DE WWW.**

**CARLOS FORMS TUTORIAL**

[http:// robot0.ge.uiuc.edu/~carlosp/cs317/ctf.html](http://robot0.ge.uiuc.edu/~carlosp/cs317/ctf.html)

**COMPUTING DICTIONARY**

[http:// womcombat.doc.ic.ac.uk](http://womcombat.doc.ic.ac.uk)

**EDUARD YOURDON**

<http://www.yourdon.com/index.html>

**FDDI**

[http:// www.ccf.auth.gr/authnet/fddi](http://www.ccf.auth.gr/authnet/fddi)

**GOPHER**

[ftp:// boombox.micro.umn.edu/pub/gopher](ftp://boombox.micro.umn.edu/pub/gopher)

**HTML DOCS-FORMS**

<http://www.cowan.edu.au/ecuwis/docs/ecudocs/forms.html>

**NETSCAPE SERVER SUPPORT**

[http:// home.netscape.com/assit/support/server](http://home.netscape.com/assit/support/server)

**SOCIEDAD INTERNET**

[http:// www.isoc.org](http://www.isoc.org)

**UNIX MANUAL PAGES**

<http://pelican.cl.cam.ac.uk/htbin/man>