

11
2ij



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

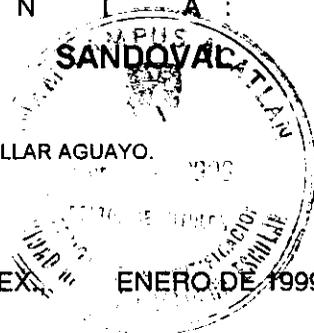
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLAN"

LA ADMINISTRACION DE USUARIOS OCUPANDO
LOS SERVICIOS DE NIS+ DENTRO DE LA RED
SUN DE LA ENEP ACATLAN

MEMORIA DEL DESEMPEÑO
P R O F E S I O N A L
QUE PARA OBTENER EL TITULO DE:
LICENCIADO EN MATEMATICAS
APLICADAS Y COMPUTACION
P R E S E N T A:
LORENZO MENDIOLA SANDOVAL

ASESOR: LIC. ADA RUTH CUELLAR AGUAYO.

SANTA CRUZ ACATLAN, EDO. DE MEX. ENERO DE 1999.



TESIS CON
FALLA DE ORIGEN

269658



Universidad Nacional
Autónoma de México



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.

AGRADECIMIENTOS

A Dios, que me dio la oportunidad de estar aquí y de convivir con mis semejantes, por darme lo que tengo hasta ahora y que para mí es más de lo que merezco.

A Reyna, mi mamá, por ser una persona que tiene ganada su entrada al cielo, por el cariño, comprensión, ternura y apoyo que siempre me ha brindado. Por todo lo imaginable e inimaginable de lo cual el papel que tengo a la mano no me alcanza para describirlo.

A Rodrigo, mi papá, por darme el buen ejemplo de un padre comprensivo y responsable, por corregirme en mis momentos de desenfreno, por fomentarme que la escuela y su cariño siempre serían la mejor herencia que me podría dar.

A mi hermana Patty, por el apoyo que me brinda y no solo por este trabajo, por hacerme ver mis errores, por la escuela de conocimientos que siempre me ha legado, por ser siempre mi amiga, " Por creer en mí ".

A Rubí, por ser mi compañera de aventuras, por estar conmigo en las buenas y en las malas, por apoyarme incondicionalmente, por estar cerca de mí, por ser parte de una riqueza de la cual me siento poseedor.

A Alfredo, por todo lo bueno que ha sido conmigo, por compartirme sus alegrías y tristezas, por enseñarme a disfrutar de la vida, por ser un excelente cuñado y amigo.

A mi asesora Ada Ruth, por su apoyo incondicional en la elaboración del presente trabajo, por ser una persona con una calidad humana indescriptible, por tenerle verdadero amor a la carrera.

A mis amigos, a mis grandes amigos de la Universidad con los que sé que contaré por siempre, con quienes tuve vivencias buenas y malas que para mi serán inolvidables, por soportarme en una etapa de mi vida que recuerdo como de persona insoportable, por darme ánimos en los momentos de decaimiento en los que quería desertar de la Carrera, por mostrarme lo que es el verdadero significado del compañerismo desinteresado.

A todos aquellos profesores de Acatlán que ponen interés al dar clases y que se pueden contar con los dedos de las manos, por impulsarnos como personas, por hacernos ver a los alumnos que no estamos completos hasta que no nos titulamos, por mostrarnos que el conocimiento es inmenso y que nosotros podemos ser parte de ese conocimiento en la medida que lo apliquemos.

A la Universidad y en particular al Departamento de Servicios de Cómputo, por permitirme ser parte de su equipo de trabajo, por todos los conocimientos adquiridos, por darme la oportunidad de realizar el presente trabajo en sus instalaciones, por las relaciones de trabajo con mis compañeros a los cuales al paso del tiempo he llegado a considerar como amigos.

Índice

INTRODUCCIÓN

I FUNDAMENTOS DE REDES

1.1 INTRODUCCIÓN	2
1.2 DEFINICIÓN DE LAN	3
1.3. TOPOLOGÍAS MÁS COMUNES	3
1.3.1. Topología Jerárquica	4
1.3.2. Topología Horizontal	4
1.3.3. Topología en Estrella	5
1.3.4. Topología en Anillo	5
1.3.5. Topología en Malla	6
1.4. ARQUITECTURA CLIENTE SERVIDOR	7
1.5. IDENTIFICADORES DE LAS MÁQUINAS	7
1.6. PROTOCOLOS DE COMUNICACIÓN	8
1.7. USUARIOS	11

II DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

2.1. DEPARTAMENTO DE SERVICIOS DE CÓMPUTO.	14
2.1.1. Descripción	14
2.1.2. Áreas que forman parte del D.S.C.	15
2.2. LA RED SUN	17
2.2.1. Antecedentes	17
2.2.2. Topología y Arquitectura ocupada por la red	18
2.3. SOLARIS, EL SISTEMA OPERATIVO NECESARIO PARA EL FUNCIONAMIENTO DE LA RED	20
2.3.1. Características Principales	21
2.3.2. Plataformas soportadas por Solaris	22
2.3.3. Definición de los servicios de Nombres	22

III ADMINISTRACIÓN DE LOS USUARIOS DE NIS+

3.1. BREVE INTRODUCCIÓN AL NIS+ NAMESPACE.	32
3.1.1. Estructura del Namespace	32
3.1.2. Estructura de las Tablas e Información	35

3.1.3. Seguridad del Sistema	37
3.1.4. Credenciales	42
3.1.5. Solstice el Manipulador de los Servicios de NIS+	45
3.2. MANEJO DE SEGURIDAD RESPECTO A LOS USUARIOS	46
3.2.1. Permisos de Acceso a las Tablas	46
3.2.2. Información Concerniente a los Passwords de los Usuarios	51
3.2.3. Tablas relacionadas con el manejo de los Usuarios e importancia de las mismas	53
Capitulo IV Desarrollo de herramientas auxiliares creadas con TCL/TK	56
4.1. INTRODUCCIÓN	57
4.1.1. Creación de nuevas herramientas administrativas	57
4.1.2. TCL/TK un lenguaje propicio para la creación de las nuevas herramientas	57
4.1.3. Los sistemas X-windows	59
4.2. ESTRUCTURA DE DESARROLLO CON TCL/TK	61
4.2.1 Entendimiento del TCL	61
4.2.2 El TK	75
4.3. ANÁLISIS Y DISEÑO DE LAS HERRAMIENTAS	82
4.3.1. Objetivos	82
4.3.2. Calendarización de actividades	83
4.3.3. Requerimientos	84
4.3.4. Análisis de factibilidad	85
4.3.5. Consideraciones importantes para el desarrollo de las herramientas	85
4.3.6. Diseño de menús, pantallas y procedimientos para las herramientas	87
4.4. IMPLEMENTACIÓN DE LAS HERRAMIENTAS	93
4.4.1. Pruebas realizadas	93
4.4.2 Ajustes necesarios	93
4.5 REPORTE DE RESULTADOS	94
Conclusiones	95
Anexo	97
Glosario	99
Bibliografía	104

Introducción

Para el autor del presente trabajo, no fue una tarea fácil escoger un tema de investigación para obtener el título de Lic. en Matemáticas Aplicadas y Computación; sin embargo, la oportunidad de trabajar en el Departamento de Servicios de Cómputo (D.S.C.) de la ENEP Acatlán abrió las puertas hacia una infinidad de conocimientos y de propuestas de proyectos de investigación que podían servir como tema de titulación.

Existían metas personales respecto al trabajo, las cuales se pueden resumir en los siguientes puntos:

- a) El tema del trabajo debía ser una aportación real, con utilidad y que no quedara almacenado para la posteridad sin uso alguno.
- b) De ser posible, la aportación que se hiciera debía beneficiar a la Universidad; esto como agradecimiento a todos los conocimientos aportados por la máxima casa de Estudios.
- c) El trabajo debía contener conocimientos innovadores que sirvieran como base para generaciones posteriores.

Como respuesta a dichas inquietudes se obtuvo el presente trabajo, el cual aporta conocimientos básicamente computacionales y cuya problemática surgió y fue resuelta en el ambiente laboral; de aquí que la propuesta de titulación sea una "Memoria del Desempeño Profesional", elaborada en el D.S.C. de la ENEP Acatlán.

El D.S.C. es un organismo que se encarga de prestar todo tipo de servicios computacionales que se encuentren a su alcance para la comunidad de Acatlán; dentro de estos servicios podemos mencionar el préstamo de equipo así como de aulas para exposiciones, impartición de cursos, impresión de documentos, utilización de equipo, etc. El D.S.C. al ser prestador de servicio tiene dentro de sus prioridades el brindar el mejor servicio posible; sin embargo, debido al crecimiento tan acelerado que tiene de forma conjunta el ámbito computacional y el Departamento, ocasionan que la población de usuarios crezca y por tanto la carga de trabajo respecto a la administración se dificulte y en algunas ocasiones no se brinde un servicio eficaz y rápido; tal es el caso que se describe a continuación y que es motivo del presente trabajo.

En la red SUN de la ENEP Acatlán se observó que la administración de los usuarios utilizando el servicio de NIS+ era un tema totalmente desconocido para los administradores de la red, principalmente por que el concepto de NIS+ es relativamente nuevo y existe poca documentación de ayuda. La falta de conocimientos sobre NIS+ ocasionaba un retardo en el tiempo de respuesta para la solución de problemas y la realización de algunas tareas cotidianas

INTRODUCCIÓN

de administración, por lo que se decidió incursionar en este tema y así mejorar los tiempos de respuesta que permitiría mejorar la calidad de servicio

El trabajo por sí mismo tiene el siguiente objetivo:

Objetivo General: Diseñar e implementar herramientas auxiliares para facilitar la administración de usuarios de la red SUN de la ENEP Acatlán, a través de la investigación de métodos de manipulación de información basados en los principios del servicio conocido como NIS+ (Network Information Service Plus), utilizando lenguajes de programación que permiten el desarrollo de aplicaciones que, en un futuro, puedan ser portables hacia un navegador de internet.

Para el cumplimiento de este objetivo el trabajo se estructuró en cuatro capítulos, cuyo contenido va de lo general a lo particular.

En el primer capítulo se describen los conceptos más generales de redes como lo son las topologías de red, los protocolos de comunicación, conceptos comunes relacionados con los usuarios del sistema, etc. La descripción de estos conceptos resulta indispensable para el mejor entendimiento de capítulos posteriores.

El capítulo número dos tiene como finalidad describir el funcionamiento de la red SUN Microsystems de la ENEP Acatlán, ubicada en el D.S.C.; lugar donde nace, se desarrolla e implementa el contenido de este trabajo. En este capítulo también se introduce el concepto de servicio de nombres, profundizando en el caso específico de NIS+ dado que es el servicio que maneja la red, además de ser parte vital en el tema de investigación.

El tercer capítulo es la descripción del funcionamiento de "NIS+" aplicado a la administración de los usuarios del mismo. Este capítulo fue uno de los más difíciles en cuanto a investigación se refiere ya que las fuentes de información son limitadas y no se conocían personas que pudieran orientar para el desarrollo del estudio. Se puede considerar que el tercer capítulo de forma conjunta con el cuarto forman la espina dorsal del trabajo.

En el cuarto capítulo se plasma el resultado de toda la investigación previa; asentando la misma en la creación de herramientas administrativas que facilitan el trabajo del administrador de la red. Dentro de este capítulo también se describen los elementos necesarios e indispensables para la creación y utilización de las nuevas herramientas.

*“ Los mejores momentos de mi vida
han sido aquellos que he disfrutado
en mi hogar, en el seno de mi familia ”*

Jefferson

I. FUNDAMENTOS DE REDES

*" Nunca consideres el estudio como
un deber, sino como una oportunidad
para penetrar en el maravilloso mundo
del saber "*

Albert Einstein



1.1. INTRODUCCIÓN

La introducción de tecnología computacional en las grandes empresas, donde la dinámica del flujo de información es extraordinaria, resolvió en su momento, grandes conflictos de manipulación de datos. Sin embargo, la creciente necesidad de compartir información entre las entidades que componen dichas empresas, originó nuevos problemas de comunicación. Por ejemplo, trasladar grandes volúmenes de información de una máquina a otra, por medio de cintas de respaldo, discos flexibles u otros medios de almacenamiento de información, significa una enorme pérdida de recursos.

Las redes surgieron como solución a esta problemática. Se diseñó una estructura diferente en la que las computadoras pudieran estar conectadas entre sí, disminuyendo considerablemente los gastos por la compra de materiales necesarios para la comunicación. Además se almacenó la información de una manera más eficiente y se compartieron recursos entre las máquinas con mayor seguridad e integridad.

El auge que tienen las redes en la actualidad tiene un comportamiento exponencial, de tal suerte que hace algunos años se hablaba de redes pequeñas compuestas por un número reducido de computadoras, y ahora las redes que se manejan cuentan con millones de terminales; tal es el caso de Internet, que es la red más grande del mundo.

Internet es una red en la que participan el sector académico, organizaciones nacionales e internacionales, organismos gubernamentales, proveedores de servicios de telecomunicaciones y empresas de todo tipo y tamaño que hacen uso de las diversas aplicaciones que ésta ofrece a sus usuarios. Tiende a enfocar sus servicios a la transmisión de información, por lo que en un futuro se convertirá en el mayor medio de comunicación. A la fecha, permite realizar desde compras por catálogo, hasta usarla como medio informativo de eventos importantes a nivel mundial, como las misiones espaciales. Ejemplo de estas misiones es la PATHFINDER, en la cual la mayor cantidad de información presentada al público fue por medio de Internet, donde se pueden observar datos e imágenes de dicha misión.

Internet es considerada como la “super carretera de la información” o la “red de redes mundial”. Internet está compuesta por redes LAN (Local Area Network) y otras de mayor extensión como las WAN (Wide Area Network). A continuación se describen algunos de los fundamentos de estas redes.



1.2. DEFINICIÓN DE LAN

Una LAN es considerada como una red de computadoras de área local o red de comunicación de datos, usada por una organización específica y que está delimitada por un área geográfica que no es mayor a la de unos cuantos edificios. La finalidad de ésta, es permitir la comunicación entre diversos usuarios, además de compartir recursos como discos duros, CD-ROMs, impresoras, paquetes, etc.

Cuando se conectan varias LAN entre sí, forman un nuevo tipo de red conocido como WAN o red de trabajo extensa, la cual está delimitada por una zona geográfica mayor a la de una LAN. Al seguir creciendo la conexión de la red, en cuanto a diversidad de equipos y zona geográfica, se da origen a la Internet.

Algunas de las características principales de una LAN son:

- Las conexiones entre las estaciones de trabajo suelen tener longitudes de cientos de metros e, incluso, kilómetros.
- Una LAN transmite datos entre estaciones de usuario y ordenadores (aunque algunas redes pueden transportar también imágenes y sonido).
- La capacidad de transmisión de una LAN suele ser mayor que la de una WAN, las velocidades de transmisión suelen estar comprendidas entre 10 y 100 Mb. dependiendo del hardware y la topología que se ocupe.
- La tasa de errores de una LAN suele ser considerablemente menor que la del canal telefónico orientado a una WAN.

Componentes que forman parte de una red LAN

- Máquinas, para ser designadas como servidores o como estaciones de trabajo.
- Tarjetas de red.
- Sistema operativo con soporte para red.
- Sistema de cableado, el cual es necesario para la comunicación entre las máquinas (topología).
- Usuarios.

1.3. TOPOLOGÍAS MÁS COMUNES

La topología de una red es la ruta por la cual pasan los datos de la misma. Es importante mencionar que la topología es la forma en la que está conformada la conectividad física de la red, por lo que el diseñador de ésta deberá hacer hincapié en aspectos como confiabilidad de los datos, costos de instalación, etc. A continuación se describen algunas de las topologías más comunes:

1.3.1. Topología Jerárquica

La topología jerárquica lleva ese nombre debido a la forma de árbol invertido que tiene. Está formada por niveles, dentro de los cuales, los nodos¹ de un mismo nivel dependen directamente del nodo superior al que pertenecen. El principal defecto de esta topología es que pueden generarse “cuellos de botella”, ocasionando que las comunicaciones se alenten. Sin embargo, los fabricantes de software y hardware la prefieren sobre las demás, debido a que, escalar de una red menor como una LAN hacia una red mayor como una WAN resulta más sencillo.

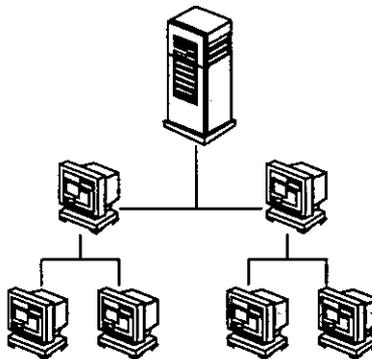


Fig. 1.1 Representación de la topología jerárquica.

1.3.2. Topología Horizontal (Bus)

Esta topología es frecuente en las redes de tipo LAN debido a que su costo es relativamente menor que el de las demás, gracias a que los datos corren a través de un solo bus², de tal forma que todas las máquinas estén recibiendo información constantemente. Sin embargo, dicha característica se convierte en un conflicto ya que, al ser solamente un canal de comunicación, las máquinas de la red están compitiendo constantemente por el medio (Bus) para poder transmitir la información deseada, causando colisiones de datos o fallas en la transmisión. Al dañarse o crearse errores en el bus, sin importar en que parte se hayan suscitado, se origina la caída total de la red.

¹ Un nodo es el punto de conexión de una máquina hacia la línea de comunicación por donde son transmitidos los datos. Una LAN puede tener tantas máquinas conectadas como nodos existan en la red.

² Considérese como un BUS al medio de comunicación por el cual pasan los datos. Los más conocidos son los cables UTP o cable de par trenzado, el cable coaxial y la fibra óptica.

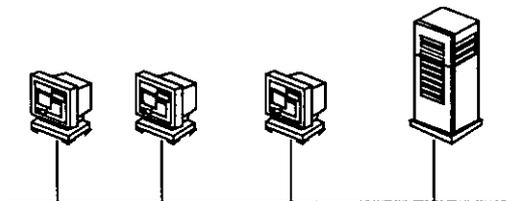


Fig. 1.2 Representación de la topología en bus.

1.3.3. Topología en Estrella

El funcionamiento de este tipo de topología es el siguiente: existe un dispositivo central que se encarga de organizar el tráfico de datos, de tal forma que cada máquina recibirá información que sea utilizable para sí misma. Cabe mencionar que existe un bus de cada máquina hacia un dispositivo que recibe el nombre de concentrador o hub el cual es responsable de establecer una ruta de comunicación entre las máquinas que así lo requieran. En la actualidad este tipo de topología es el más ocupado, ya que aunque su costo de instalación es más elevado, brinda mejores resultados en cuanto a rendimiento, confiabilidad y crecimiento.

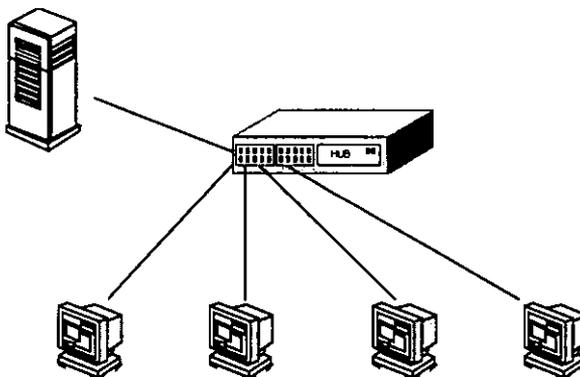


Fig. 1.3 Representación de la topología en estrella.

1.3.4. Topología en Anillo

En esta topología el cableado va de máquina en máquina, sin que exista un principio o un final definidos. Cada nodo recibe datos y en caso de no ser utilizados, los retransmite al siguiente nodo y así sucesivamente hasta que lleguen a su destino, por lo que el flujo de información sigue una misma dirección. Esta trayectoria aparenta la formación de un anillo, es de ahí de donde surge su nombre. Esta topología tiene la desventaja de que, al igual que en la



horizontal, en el momento en que falla el canal entre dos nodos, se ocasiona la caída completa de la red. Para solucionar el problema anterior algunos diseñadores de topologías agregan un canal de seguridad.

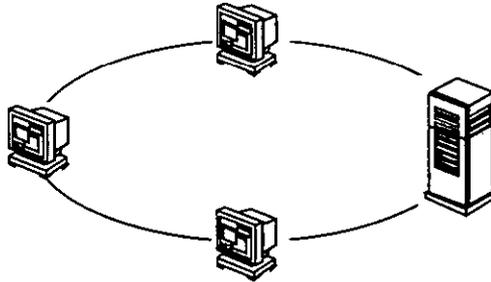


Fig. 1.4 Representación de la topología en anillo.

1.3.5. Topología en Malla

Cuenta con un nivel de confiabilidad mucho más alto en comparación a las topologías mencionadas hasta este momento, pero su costo de instalación es el más elevado también. Para solucionar el problema de “embotellamiento” de la transmisión de datos, esta topología genera caminos alternos de transmisión que pueden ser procesos muy complejos. Su uso se restringe a grandes empresas que requieren de una alta confiabilidad en la transmisión de sus datos y que tienen los recursos suficientes; tal es el caso de algunas instituciones bancarias.

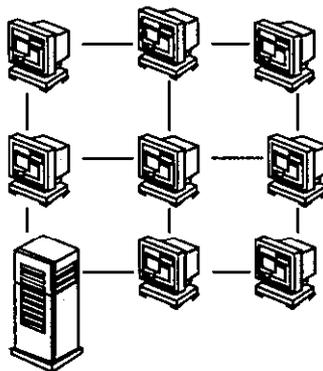


Fig. 1.5 Representación de la topología en malla.



1.4. ARQUITECTURA CLIENTE SERVIDOR

La arquitectura cliente servidor, se basa en la existencia de una máquina que recibe el nombre de servidor que se encarga de realizar los procesos más complejos (pesados) de la red, como lo son: el servicio de archivos, de impresión y la ejecución de programas. El servidor se encarga de procesar las peticiones hechas por otras máquinas que requieren ser atendidas, las cuales reciben el nombre de clientes o de estaciones de trabajo.

Este tipo de arquitectura es el que tiene la mayor demanda por la forma de organización que establece. En la mayoría de los casos el servidor es el que almacena toda la información de los usuarios y permite que, al conectarse a un cliente, el usuario tenga disponible su información sin importar que no exista de forma local.

1.5. IDENTIFICADORES DE LAS MÁQUINAS

Para que una máquina en particular pueda acceder información almacenada en la red, es necesaria la identificación de la misma por medio de una clave, y para evitar conflictos de localización, se requiere que la máquina tenga un identificador único. El número de identificadores variará dependiendo del sistema operativo (S.O.) de la red que se maneje y de los servicios que ésta ofrezca. Para fines de este trabajo se manejarán los siguientes 3 identificadores básicos, los cuales son necesarios en el tipo de servicio que se brinda:

Host Name. Este es un identificador alfanumérico que consta de 1 hasta n caracteres donde n depende del S.O. Con este identificador el manejo de las máquinas resulta sencillo, por que es más fácil recordar nombres que números. En una red que maneje estaciones de trabajo y en particular las redes UNIX, siempre será indispensable asignar este identificador. Es importante destacar que el manejo del host name es exclusivo del administrador de la red, el cual puede manipularlo por medio de software. Por otro lado, existen protocolos³ como NFS (Network File System) que requieren de este identificador para enviar y contestar requisiciones de información de una máquina a otra.

IP Address (Internet protocol address). La IP address es un identificador que ocupa el protocolo TCP/IP para ubicar alguna máquina que esté conectada a INTERNET. Este identificador esta conformado por 4 bytes, cada byte tiene una representación numérica que va de 0 a 255. Ahora bien, una vez que una máquina está conectada a Internet se podrá comunicar con otra tomando como nombre esta dirección. En la mayoría de los casos es necesario manejar un DNS (Domain Name Service), que es el encargado de asociar la IP address con el host name.

³ Más adelante se describe qué es un protocolo.



Ethernet Address. Esta dirección esta *conformada* por un número único de 12 caracteres (6 kb.) de nomenclatura hexadecimal (1 - F). Un ejemplo del ethernet address es 08:00:20:1c:ab:5e, aunque también puede tomar la forma de 8:0:20:1c:ab:5e, debido a que cuando el cero aparece en la primera posición de cada par, se puede omitir.

1.6. PROTOCOLOS DE COMUNICACIÓN

El software que se encarga de establecer los lineamientos o reglas para que los paquetes envíen información a través de la red, se conoce como protocolo de comunicación. Existen diferentes tipos de protocolos, que varían de acuerdo a los fabricantes de software y a las necesidades particulares de comunicación.

El protocolo más conocido a la fecha es TCP/IP. Sus lineamientos de comunicación de datos se basan en el modelo de referencia OSI (Open System Interconection), el cual describe el proceso de manipulación de los datos para su transmisión de una máquina a otra.

A continuación, se presenta la descripción del funcionamiento del modelo OSI como introducción del protocolo TCP/IP.

El modelo OSI. La necesidad de intercambiar información entre sistemas heterogéneos, i.e. , entre sistemas cuyas tecnologías son muy diferentes entre sí, llevó a la ISO (International Standard Organization) a buscar la manera de regular dicho intercambio. El modelo OSI surge en el año de 1983, y es el resultado del trabajo realizado por la ISO en la estandarización internacional de los protocolos de comunicación.

El modelo OSI consta de 7 capas individuales, sin las cuales sería imposible la comunicación. En la fig 1.6 se observa gráficamente cómo se realiza la transmisión de información en base al modelo OSI.

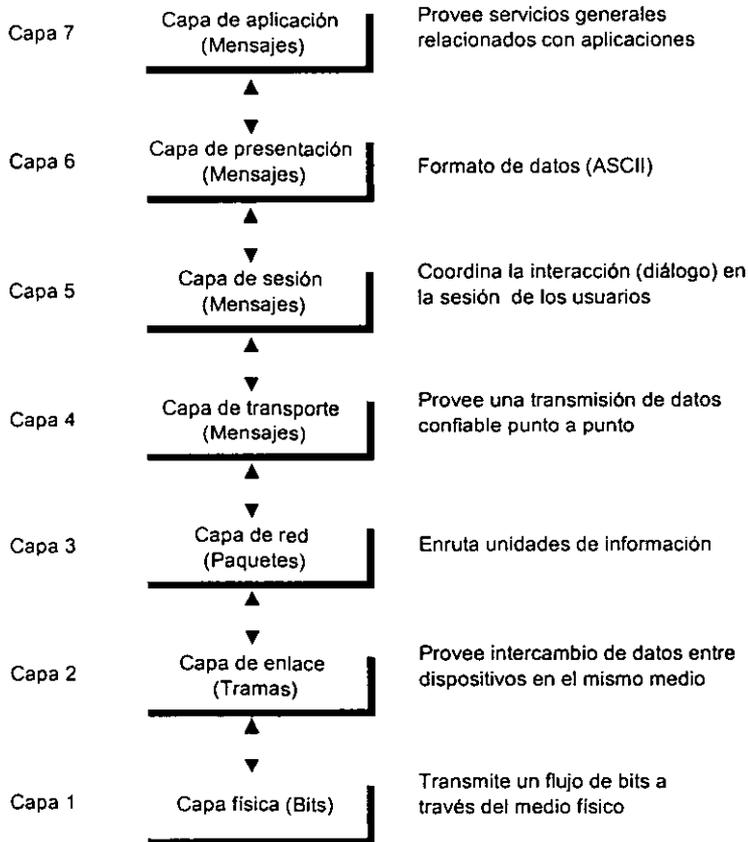


Fig. 1.6 Estructura de las capas del modelo OSI.

Las características generales de las capas son las siguientes:

- ◆ Cada una de las capas desempeña funciones bien definidas.
- ◆ Los servicios proporcionados por cada capa son utilizados por la capa superior.
- ◆ Existe una comunicación virtual entre 2 mismas capas de manera horizontal.
- ◆ Existe una comunicación vertical entre la capa n y la capa $n+1$
- ◆ La comunicación física se lleva a cabo en la capa 1.



TCP/IP. Surge como un proyecto de investigación de una organización conocida como DOD (Department Of Defense) cuya finalidad era conectar diversos tipos de red de distintos vendedores de hardware. El proyecto fue un éxito, se liberaron algunos servicios básicos como correo electrónico, sesión remota y transferencia de archivos, que la mayoría de los usuarios necesitaba en arquitecturas cliente/servidor.

TCP/IP está compuesto por un subconjunto de protocolos que le permiten realizar su tarea. A continuación se mencionán los más conocidos:

- ◆ **IP (Internet Protocol).** Es el responsable de mover los paquetes de datos de un nodo a otro. Esta comunicación se basa en la dirección IP o IP Address. IP trabaja en la capa 3 del modelo OSI y opera en gateways que son los responsables de direccionar la información a través de la jerarquía de redes.
- ◆ **TCP (Transport Control Protocol).** Es el responsable de verificar la correcta liberación de datos que se lleva a cabo de un nodo a otro, evitando que la información se pierda durante su transmisión. En caso de pérdida de datos o detección de errores, TCP adiciona el soporte adecuado para realizar las operaciones necesarias y retransmitir la información hasta que se transmita correctamente. Este protocolo se ubica en la capa 4 del modelo OSI.
- ◆ **UDP (User Datagram Protocol).** Este protocolo, al igual que el anterior, se sitúa en la capa de transporte y desempeña las mismas funciones. UDP resulta ser menos complejo pero poco confiable.
- ◆ **ICMP (Internet Control Message Protocol).** Como su nombre lo indica, es el responsable de llevar los mensajes de error de la red y notificar otras condiciones que requieren atención del software de la red.
- ◆ **ARP (Address Resolution Protocol).** Se encarga de modular mapas de direcciones en la capa de red (direcciones IP) a la dirección correspondiente en la capa de enlace (direcciones Ethernet). Es decir, es el que se encarga de convertir de direcciones IP a direcciones Ethernet dentro de una tecnología Ethernet.
- ◆ **RARP (Reverse Address Resolution Protocol).** Es la contraparte del protocolo ARP, es decir, convierte o interpreta de direcciones Ethernet a direcciones IP.

La fig. 1.7 muestra gráficamente cómo se realiza la comunicación entre los protocolos anteriormente mencionados.

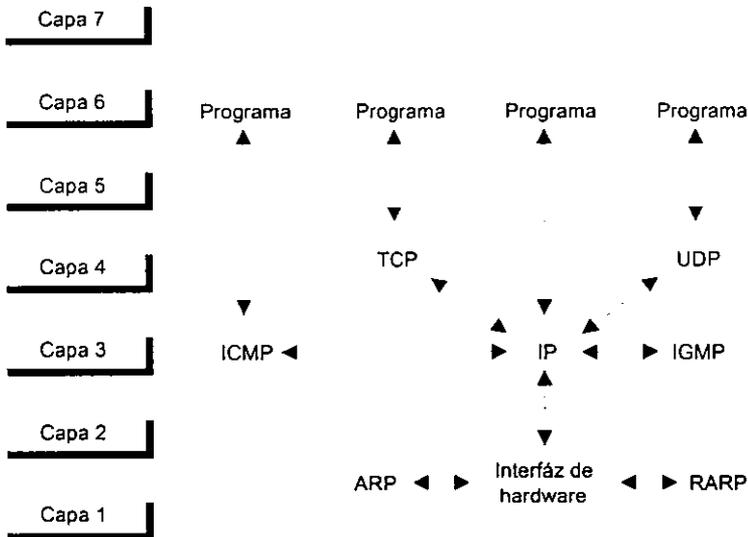


Fig. 1.7 La comunicación de los protocolos en base al modelo OSI.

1.7. USUARIOS

Los usuarios del sistema son aquellas personas que tienen permisos necesarios y capacidad suficiente para hacer uso de la red. Algunos conceptos relacionados con los usuarios son:

Login. Es un identificador único que le indica al sistema quién es el usuario que está haciendo una requisición de información. El login representa una de las mayores ventajas en cuestiones de seguridad que ofrecen los sistemas conectados en red, de tal forma que no cualquier persona podrá hacer uso de los recursos.

El login está compuesto por 2 caracteres como mínimo y 8 máximo (esta característica depende de la configuración del S.O.). Debe estar formado por caracteres alfanuméricos y nunca podrá contener caracteres especiales⁴.

Password. Aunado al login existe un identificador secreto denominado comúnmente password. Este identificador es una confirmación que pide el sistema para permitir el acceso a la red. Por seguridad, cuando el usuario está introduciendo el password, el eco de la máquina se desactiva, i.e., en la pantalla no aparece lo que se está tecleando, o aparecen caracteres que no corresponden a las teclas que se están oprimiendo.

⁴ Considérense como caracteres especiales: *, -, +, /, |, &, ', =, (, &, %, \$, #, ", |, {, [, entre otros.



Root Master. Es una cuenta especial que se asigna al administrador de la red, el cual tendrá algunos o todos los privilegios de acceso a la información. Cabe señalar que en la arquitectura cliente/servidor existirá una cuenta de root Master por cada estación de trabajo y que por conveniencia no deberá ser la misma.

Home Directory (directorio hogar). Cada usuario, al momento de entrar al sistema, accesará, por default, a un directorio designado por el administrador de la red. Dentro de este directorio el usuario tiene todos los permisos sobre los archivos. Este directorio recibe el nombre de Home directory, y su ubicación dentro del sistema de archivos, dependerá del administrador.

E-MAIL. El E-mail es el identificador con el cual los usuarios que tengan salida a Internet se comunican con usuarios de otras redes que también estén conectados a Internet. Su utilidad reside en el intercambio de mensajes a través de un buzón electrónico, servicio que ofrece Internet, y que es conocido como cuenta de correo electrónico. La dirección e-mail está compuesta por el login del usuario y el host name del servidor en el cual está declarado, unidos por una "@"; por ejemplo:

Servidor: apolo.acatlan.unam.mx

Usuario: rubi

Dirección e-mail: rubi@apolo.acatlan.unam.mx

Las condicionantes que rigen la asignación de un e-mail son precisamente la unión de las existentes para los Host name y los login. El hecho de que exista un usuario registrado en el sistema no implica necesariamente que tenga su cuenta de correo electrónico. En algunos sistemas operativos de red, el servicio de correo electrónico no está soportado por default, por lo que se tendría que comprar software adicional para poder utilizarlo.

II. DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

*“ El secreto de la felicidad no está
en hacer siempre lo que se quiere,
sino en querer siempre lo que se
hace”*

Leon Tolstoi



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

El Centro de Cómputo es el organismo que se encarga de administrar la mayor parte de recursos computacionales con que cuenta la ENEP Acatlán, y se divide en:

- a) Departamento de Servicios de Información (D.S.I.)
- b) Departamento de Servicios de Cómputo (D.S.C.)

2.1. DEPARTAMENTO DE SERVICIOS DE CÓMPUTO

2.1.1. Descripción

El objetivo del D.S.C. es proporcionar a los alumnos y profesores los servicios de procedimiento electrónico de datos de acuerdo a los requerimientos propios de sus actividades académicas; administrar, operar, controlar e inventariar los recursos de cómputo con que cuenta la Escuela, así como facilitar equipo y apoyo técnico requerido para las diferentes actividades que se realizan en el plantel.

Las Funciones de este Departamento¹ son:

- Establecer los programas y procedimientos de trabajo anuales, que permitan satisfacer las demandas de servicios de cómputo que solicitan los diferentes programas de la escuela.
- Elaborar un programa anual de los requerimientos de útiles de trabajo, materiales y equipos necesarios para el buen funcionamiento del departamento, así como controlar su correcta utilización.
- Elaborar el Anteproyecto de Presupuesto Anual del Departamento de Servicios de Cómputo, en base a los lineamientos establecidos por la Institución.
- Administrar, operar y controlar los recursos de cómputo con que cuenta la escuela.
- Mantener un inventario actualizado de los recursos de cómputo que están distribuidos en las diferentes dependencias de la escuela.
- Proporcionar mantenimiento preventivo y correctivo a todos los equipos de microcomputación con que cuenta la escuela.
- Definir y establecer los mecanismos y políticas específicas para la prestación de los servicios.
- Proporcionar equipo de cómputo y apoyo técnico a la comunidad de la escuela en las actividades que lo requieran.
- Asesorar a los usuarios en la operación adecuada de los sistemas de cómputo.
- Organizar e impartir cursos para los usuarios sobre los sistemas operativos de los diversos equipos.
- Organizar e impartir cursos de computación dirigidos tanto a la comunidad interna como externa de la escuela.

¹ ENEP Acatlán: Manual de Organización 1995 (1995; pp.349-352).



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

- Promover la capacitación del personal que labora en el Departamento, para elevar la calidad del servicio que éste proporciona.
- Redactar notas técnicas sobre temas específicos de computación, lenguajes de programación y paquetería como apoyo a los usuarios.
- Controlar los recursos materiales y humanos con que cuenta el Departamento, a fin de hacer más congruente la relación entre el equipo y la prestación del servicio.
- Generar las credenciales para el uso de los servicios que presta el Centro de Cómputo, tanto para profesores como alumnos.
- Elaborar un informe bimestral de las actividades que realiza el Departamento de Servicios de Cómputo.
- Evaluar los servicios prestados por el Departamento, a fin de realizar los ajustes y correcciones necesarias para mejorar los servicios que ofrece.

2.1.2. Áreas que forman parte del D.S.C.

Dentro del D.S.C. se han creado áreas cuya finalidad es brindar el servicio adecuado y cumplir con el objetivo y las funciones establecidas en el punto anterior. La fig 2.1 muestra el organigrama del departamento.

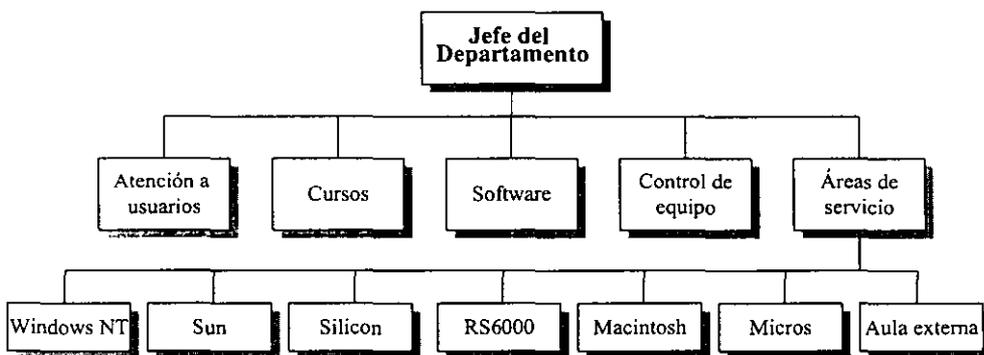


Fig. 2.1 Organigrama del D.S.C.

Dentro del organigrama se observan tres niveles que son:

- I) Jefe del Departamento. Encargado de la administración del D.S.C.
- II) Áreas Generales. Descritas en la tabla 2.1.
- III) Áreas de servicio. Son las clases de equipo con que cuenta el D.S.C.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

Área	Descripción
Atención a usuarios	Es el área responsable de administrar los recursos de los usuarios. Se encarga de la creación de credenciales, solución de problemas, agenda de préstamo de equipo, etc.
Cursos	Dentro del Departamento se imparten cursos de forma periódica tanto a la comunidad interna (alumnos, profesores y trabajadores de la UNAM) como a comunidad externa (toda persona ajena a la Institución). Esta área es la responsable de manejar todo lo relacionado con cursos, como la calendarización, las inscripciones, informes, asignación de espacios para la impartición de cursos, etc.
Software	Existe un conjunto de personas especializado en el manejo de paquetería, el cual conforma el "área de software". Estas personas son las encargadas de dar soporte a nivel software en las dependencias de la escuela que así lo requieran.
Control de Equipo	Debido a la cantidad de equipo de cómputo que maneja el D.S.C. es necesario llevar un inventario que sea independiente del que maneja el Departamento de Almacenes e Inventario ² . El área de Control de Equipo se responsabiliza de esta tarea y se encarga de generar los reportes necesarios que se soliciten.
Áreas de servicio	Las áreas de servicio fueron creadas para instalar, asesorar y dar mantenimiento a los distintos equipos con que cuenta el Departamento.

Tabla 2.1 Descripción de las áreas de servicio.

La administración de cada área de servicio está a cargo de 1 ó 2 personas, las cuales se responsabilizan de su buen funcionamiento. La red Sun constituye un área de servicio cuya administración está a cargo del autor de esta memoria.

La experiencia que se ha obtenido a lo largo de la labor como administrador de la red Sun Microsystems, desde su llegada al D.S.C. en 1995 hasta el momento, ha permitido analizar muchos de los aspectos técnicos que se requieren para crear un ambiente de trabajo propicio para los usuarios. Asimismo, con el uso frecuente de ciertas herramientas de administración de servicios, se pudo visualizar algunas carencias que tiene el S.O de este equipo, por ejemplo:

- Falta de herramientas gráficas para la manipulación de los procesos del sistema.
- Inexistencia de herramientas para el manejo de la jerarquía del servicio de información.
- Carencia de herramientas de administración masiva de usuarios del sistema (los usuarios tienen que ser manipulados de forma individual).

² El Departamento de Almacenes e Inventarios es una Dependencia de la Escuela cuyo objetivo es manejar el inventario general de toda la Escuela.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

Ejemplo de la carencia de herramientas de administración masiva, es que al inicio de cada semestre se tienen que dar de baja claves que ya no serán ocupadas, y esto se hace una por una. Asimismo, se tienen que crear nuevos usuarios que varían entre los 180 y 250, dependiendo de la demanda que tenga el equipo. La creación de un usuario, en el mejor de los casos, tarda aproximadamente 1 minuto, por lo que crear todos los usuarios puede tomar de 3 a 4 horas.

Es necesario tomar en consideración que la tarea de un administrador no se basa únicamente en la creación y borrado de cuentas de los usuarios, sino que existen muchas otras, tales como la depuración de cuentas, los cambios de permisos y de grupos, etc. Por ello surge la inquietud de complementar el S.O. diseñando e implementando herramientas gráficas que faciliten el trabajo del administrador.

Con la finalidad de proyectar esta inquietud, me es indispensable explicar el funcionamiento y desempeño de la red, particularizando gradualmente en los conceptos que me ayudaron a diseñar las herramientas. A continuación se describirán los puntos más relevantes.

2.2. LA RED SUN

2.2.1. Antecedentes

La red Sun llegó a la Institución el 25 de noviembre de 1994 pero, por problemas técnicos y de presupuesto comenzó a dar servicio hasta el período escolar 96-1.

Inicialmente el servicio se brindaba con un servidor SPARCserver 1000, cinco estaciones de trabajo SPARCclassic y 15 terminales gráficas NCD (las cuales requieren de un software especial para su funcionamiento).

Con el tiempo se adquirió nuevo equipo y actualmente la red está formada por el mismo SPARCserver 1000, 9 estaciones SPARCclassic, 13 ULTRA-1 y las 15 terminales NCD. Dado que el equipo NCD no es de interés para el presente trabajo, no será mencionado posteriormente.

La red fue adquirida para sustituir al equipo HP 3000, cuyo objetivo era que los alumnos de las carreras de Ingeniería, Actuaría y Matemáticas Aplicadas y Computación aprendieran a programar en lenguajes como Basic, Fortran y Pascal. Dando seguimiento al objetivo de la HP 3000, la red Sun fue adquirida con compiladores de Fortran, Pascal, C y C++. Sin embargo, debido a que la red permitía conectarse muy fácilmente a Internet, los alumnos dejaron de usarla sólo para compilar y empezaron a ocuparla para búsqueda de información, por lo que el enfoque de la red dió un giro considerable.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

En la actualidad, la red es ocupada por alumnos de todas las carreras que se imparten en la ENEP Acatlán, los cuales se dedican a buscar información referente a diversos temas que se imparten en clases. Además, se utiliza para aprender nuevos lenguajes de programación que pertenecen al grupo de mayor demanda en el mercado de trabajo, como lo son HTML³ y Java⁴. Otro servicio que proporciona es el acceso al correo electrónico, tanto para profesores como para diversas dependencias de la misma escuela.

Existen personas dedicadas a la investigación de proyectos que requieren de este equipo en particular, por lo que otro servicio que ofrece la red es dar cabida a proyectos de investigación con fines no lucrativos y meramente académicos.

A la fecha, el servicio se otorga a más de 100 profesores, 20 dependencias y 300 claves distribuidas entre alumnos y trabajadores del Centro de Cómputo.

2.2.2. Topología y Arquitectura ocupada por la Red

La red SUN Microsystems de la ENEP Acatlán es considerada como parte de una LAN, cuya extensión abarca el área misma de la escuela y su topología corresponde a la de tipo estrella.

Para esta red se ocupa la tecnología ethernet, la cual requiere de concentradores y switches⁵ que permitan la transmisión de datos a 10 megabits por segundo (Mbp). Ahora bien, como se puede observar en la fig 2.2, cada estación de trabajo está conectada de su puerto al concentrador, y a su vez, este concentrador está conectado al switch (conexión conocida como cascaré), en donde se encuentra conectado el servidor. A su vez, el switch está conectado a un anillo de fibra óptica o FDDI⁶. Existe una punta conectada al anillo por medio de un router⁷ que llega a la Dirección General de Servicios de Cómputo Académicos (DGSCA) que es la encargada de brindar la salida a Internet. Es importante mencionar que la mayoría de las conexiones están hechas por medio de cable de par trenzado (UTP), excepto el anillo FDDI.

³ HTML es un lenguaje de programación con el cual se diseñan las páginas que presentan los visualizadores del WEB o browsers.

⁴ Java es un lenguaje creado por la compañía Sun Microsystems cuya finalidad es poder correr aplicaciones portables a cualquier plataforma a través del WEB.

⁵ Un SWITCH es un dispositivo físico parecido a un Hub. Cumple con tareas similares a las de un concentrador, con la diferencia de que el Switch es considerado como inteligente, ya que almacena información en tablas para determinar qué dispositivos se tienen conectados a él.

⁶ FDDI es una forma de interconexión de cableado semejante a la topología en anillo en la cual los datos corren en una sola dirección. La velocidad de transmisión de datos de FDDI es de 100 Mbps.

⁷ El router es otro dispositivo físico que se encarga de dirigir paquetes desde una red a otra. En la mayoría de los casos, estos dispositivos también manipulan las líneas telefónicas.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

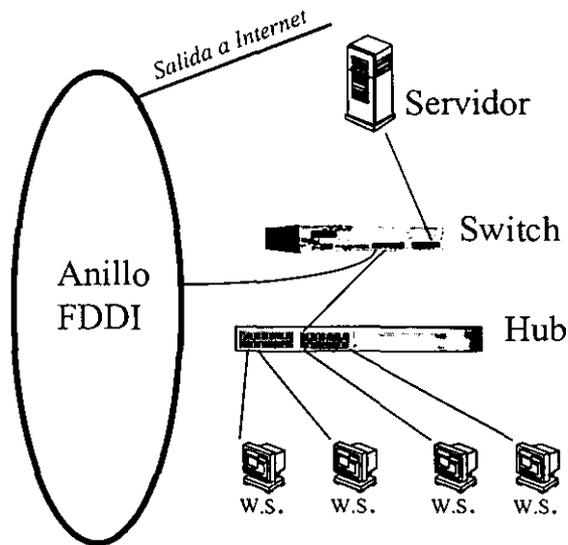


Fig 2.2 Conexión de la red SUN.

La arquitectura que se maneja es de cliente/servidor. Se tiene un servidor central que es el encargado de almacenar la información de los usuarios, así como las aplicaciones que consumen más recursos, además de responsabilizarse de uno de los servicios básicos de Internet que es el correo electrónico. Por otra parte se tienen clientes, los cuales se pueden acceder para manejar la información que contiene la máquina central.

Por mencionar algunas características del servidor haremos referencia a las siguientes:

Característica	SPARC Server 1000
Disco duro	2 Gb. de disco duro interno Gabinete externo de 4 Gb.
Procesadores	2, SPARC
Memoria	198 Mb.
Host Name	apolo.acatlan.unam.mx
IP Address	132.248.80.198
Ethernet Address	8:0:20:1c:ab:5e
Sistema Operativo	Solaris 2.5.1

Tabla 2.2 Características principales del servidor SUN.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

Los clientes (estaciones ULTRA-1 y SPARCclassic), son equipo con características inferiores:

Característica	Ultra-1	SPARCclassic
Disco duro	1 y 2 Gb., dependiendo del modelo.	210 y 500 Mb., dependiendo del modelo
Procesadores	1 SPARC a 140 Mhz.	1 SPARC a 80Mhz.
Memoria	32 Mb.	16 Mb.
Sistema Operativo	Solaris 2.5.1	Solaris 2.5.1

Tabla 2.3 Características principales de las estaciones de trabajo.

La arquitectura cliente/servidor aplicada a este equipo presenta algunas deficiencias, ya que los discos duros son propensos a dañarse debido a la falta de corriente eléctrica repentina; en el caso específico del servidor, los daños a su disco duro significarían la pérdida de información vital para el funcionamiento de la red. Resultó interesante observar qué soluciones propone el S.O. para resolver problemas de esta índole. Para evitar estas anomalías se pueden crear respaldos de la información (bases de datos del sistema) en una máquina que se llama espejo o mirror, la cual, independientemente de las fallas que pueda sufrir el servidor, podrá sacar a flote la información más importante de la red. La capacidad de prevenir problemas de esta índole es una de las características del S.O. Solaris y en particular de NIS+, del cual se hablará más adelante.

2.3. SOLARIS, EL SISTEMA OPERATIVO NECESARIO PARA EL FUNCIONAMIENTO DE LA RED

El Sistema Operativo de Sun (SunOS) fue desarrollado por Bill Joy. SunOS se fundamenta en la versión 4.2 de BSD y añade características que aumentan en gran medida la capacidad del S.O., especialmente en el área de red. Entre esas capacidades se incluye el Sistema de Archivos de Red (NFS), el Sistema de Información de Red (NIS y NIS+), las llamadas a Procedimientos Remotos (RPC) y la Presentación de Datos Externos (XDS).

La siguiente generación de sistemas operativos de Sun se ha denominado Solaris. SunOS ahora forma parte del S.O. Solaris, que está basado en *System V Versión 4 (SVR4)*. Sun Microsystems ha transportado el sistema operativo Solaris a la familia de procesadores de Intel (Intel 80486). Solaris utiliza la Interfaz Gráfica de Usuario (GUI) OpenWindows y el entorno de Computación en Redes Abiertas (ONC). Como parte del acuerdo para el Entorno Software Abierto Común (COSE), una alianza formada por vendedores para estandarizar elementos relativos a UNIX), Sun soporta hoy en día la interfaz gráfica de usuario MOTIF de OSF. Aunque la mayoría del equipo donde se instala este sistema operativo tiene arquitectura para trabajar a 64 bits, Solaris es un sistema operativo que corre a 32 bits.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

2.3.1. Características Principales

Solaris se deriva del sistema operativo UNIX y conserva la mayor parte de las características esenciales de este sistema, entre las cuales se pueden mencionar:

Fue desarrollado en un lenguaje de alto nivel. Está escrito en lenguaje "C", lo cual lo hace portable, ya que se puede correr en máquinas con distinta arquitectura.

Interfaz simple (Shell). Cuando se entra al sistema, el usuario se dirige a un intermediario, llamado intérprete de comandos o shell. El shell es el programa que se encuentra entre el usuario y las facilidades del núcleo.

Interfaz simple a periférico. Permite conectar cualquier periférico⁸ de cualquier fabricante.

Sistemas de archivos jerárquicos. Esto se refiere a la estructura de árbol invertido que tienen los directorios y subdirectorios, permitiendo clasificar y ordenar de una manera sencilla la información.

Multiusuario y Multitarea. Esta característica, en su momento, fue la que mayor auge le dio a este sistema operativo, ya que una máquina podía estar atendiendo simultáneamente a varios usuarios; además, cada usuario podía estar ejecutando más de una tarea a la vez.

Seguridad: archivos privados y compartidos. Al igual que con la mayoría de los sistemas operativos de red, en Solaris un usuario puede restringir o denegar el acceso a su información. Este tipo de seguridad se denomina "permisos de los archivos", y puede ser modificada de acuerdo a la conveniencia del dueño de la información.

Múltiples programas simples resuelven situaciones complejas. Como resultado de la ejecución de comandos y de ocupar utilerías de UNIX, se pueden crear nuevos programas que resuelven situaciones complejas.

Capacidad de trato directo con Internet. Gracias a que el sistema UNIX tiene el soporte necesario para el protocolo TCP/IP, es fácil conectar los equipos en red para que puedan utilizar los servicios de Internet, siguiendo el flujo de datos explicado en la sección del protocolo TCP/IP.

Otras características específicas de Solaris y que no son totalmente herencia de UNIX son:

⁸ Entiéndase por un periférico a todo dispositivo físico que permite la entrada o salida de información en la computadora.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

Escalabilidad. Una red puede crecer en el sentido de pasar de una red de no más de 15 máquinas a una red con estructura jerárquica con unos cuantos cientos de equipos, dentro de los cuales algunos deberán ser designados como servidores.

Manejo de X-windows. En caso de desearlo, se puede trabajar con un ambiente gráfico llamado OpenWindows que maneja Openlook. Ahora bien, salió un nuevo producto al mercado, complemento de Solaris a nivel gráfico, conocido como CDE (Common Desktop Environment) que permite trabajar con MOTIF.

2.3.2. Plataformas soportadas por Solaris

Inicialmente Solaris corría exclusivamente sobre arquitectura SPARC (creada por la compañía SUN Microsystems), pero en la actualidad se ha extendido su mercado y puede correr sobre máquinas con procesadores INTEL y POWERPC.

Para poder inicializar una máquina, el S.O. requiere como recursos mínimos un procesador SPARC a 80 Mhz, 16 Mb. de memoria RAM y 80 Mb. de disco duro (End Core system support), hablando de equipo SUN Microsystems. Para una PC se requiere de un procesador INTEL 386 a 33 Mhz, 16 Mb de memoria RAM (incluyendo 640 kb. de memoria base), y 70 Mb. de disco duro. Sin embargo, por experiencia se puede sugerir que, para que el equipo funcione adecuadamente sobre procesadores INTEL, se requiere al menos de un procesador 486 a 33 Mhz con 16 Mb. de memoria y 80 Mb. de disco duro.

2.3.3. Definición de los Servicios de Nombres

I) Introducción.

El servicio de nombres es el encargado de almacenar información de usuarios, estaciones de trabajo y aplicaciones que tienen que comunicarse a través de la red. De no existir el servicio de nombres, cada estación necesitaría una copia de la información mencionada. Esta información incluye los identificadores de las máquinas (IP address, Ethernet address y Host names), nombres de usuarios (login) passwords y permisos de acceso. La información puede ser almacenada en archivos o tablas de bases de datos.

El centralizar la información facilita la administración de la red. Por ejemplo, imagínese una red sencilla que consiste de tres máquinas, como se muestra en la figura 2.3



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

Como se puede observar en el esquema anterior, a mayor cantidad de estaciones de trabajo, corresponde mayor número de configuraciones a realizar, lo cual resulta bastante tedioso. Para facilitar esta labor se utilizan los servicios de la red, almacenando la información necesaria en servidores, que son los encargados de proveerla a cada estación de trabajo que así lo requiera.

En la fig. 2.5 se muestra cómo las estaciones de trabajo son clientes reconocidos por el servidor. Al momento en que se cambia o actualiza la información en el servidor, los clientes hacen referencia a los datos ya modificados, sin tener que realizar cambios en sus discos duros locales, lo cual reduce el número de errores e inconsistencias de los clientes en cuestión.

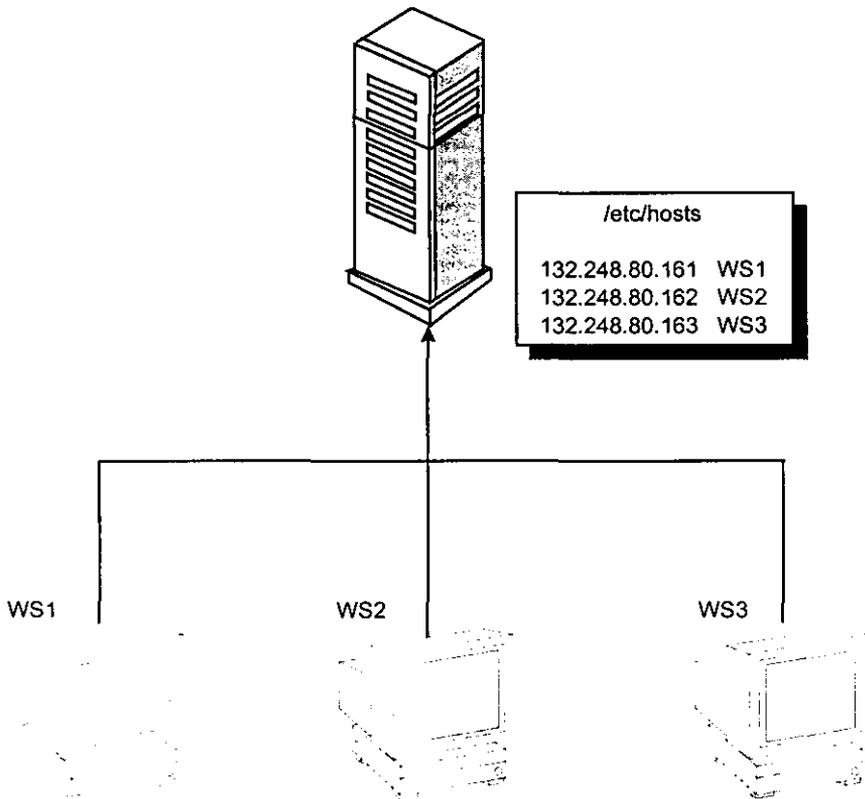


Fig 2.5 Centralización de datos en el servidor.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

II) Servicios de Nombres más comunes.

Los servicios de nombres más comúnmente usados en el sistema Solaris son DNS, FNS, NIS y NIS+. A continuación se presenta una descripción de cada servicio, profundizando en los 2 últimos debido al objeto de estudio.

- a) **DNS (Domain Name Service).** Es el servicio de nombres que provee la red Internet por medio del protocolo TCP/IP con la finalidad de que cada estación de trabajo pueda ser identificada no sólo por su dirección IP, sino también por el Host Name que resulta de más fácil manejo. El conjunto de estaciones de trabajo y de redes es referenciado como el DNS Namespace. Este DNS Namespace esta dividido en dominios jerárquicos soportados por uno o dos servidores de nombres (un servidor principal y uno o varios servidores secundarios).
- b) **FNS (Federal Name Service).** Soporta el uso de diferentes servicios autónomos en un ambiente de Solaris simple; además de poder trabajar con una interfaz del sistema que resulta de fácil uso.
- c) **NIS (Network Information Service).** Fue creado independientemente de DNS y tiene un enfoque significativamente diferente. Mientras que DNS tiene como fin el proporcionar nombres asociados a las direcciones IP, NIS busca centralizar la información vital, tanto de identificadores de estaciones de trabajo como de usuarios y de otros servicios, para el funcionamiento de la red.
 - ◆ **Arquitectura de NIS.** NIS ocupa una estructura parecida a la de DNS. Se usan dominios para agrupar estaciones de trabajo y así conformar el namespace. Es importante mencionar que NIS no tiene una estructura jerárquica, sino plana, ya que no se pueden crear subdominios dentro de un dominio establecido. Un dominio de NIS no puede estar conectado de forma directa a la Internet, sin embargo, existen algunas organizaciones que requieren de esta conexión, por lo que combinan DNS con NIS; manejando toda la información local con NIS y la resolución de los host names con DNS. NIS provee rutinas especiales para que los clientes, al momento de hacer una requisición de información, se dirijan directamente al servidor de NIS, siempre y cuando la petición no esté relacionada con las direcciones IP.
 - ◆ **Los mapeos de NIS.** La información que maneja NIS es almacenada en archivos que reciben el nombre de mapeos (NIS maps) y que substituyen algunos archivos ubicados en el directorio /etc. Estos archivos se organizan básicamente en tablas de dos columnas. Una de las columnas hace referencia a la llave y la otra a información acerca de la llave (la búsqueda de datos de un cliente se hace a través de la llave). Mucha de la información es almacenada en distintos mapeos, ocupando una llave por cada uno. Por ejemplo, los Host name y las Ethernet address son almacenados en dos mapeos: hosts.byname y hosts.byaddr respectivamente. En la tabla 2.4 se presentan los mapeos de NIS, así como una breve descripción de cada uno de ellos.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

NIS MAP	Descripción
bootparams	Lista de los nombres de clientes que requieren del servidor para bootear, así como ubicación de los archivos requeridos durante el booteo.
Ethers.byaddr	Lista de las Ethernet address de estaciones de trabajo y su correspondiente Host Name.
Ethers.byname	Lista de los Host name de las estaciones de trabajo y su correspondiente Ethernet address.
Group.bygid	Provee información de miembros de grupos usando el identificador del grupo como llave.
Group.byname	Provee información de miembros de grupos usando el nombre como llave.
Hosts.byaddr	Lista de los Host name y las Ethernet address utilizando la primera como llave.
Hosts.byname	Lista de los Host name y las Ethernet address utilizando la segunda como llave.
Mail.aliases	Lista de los alias (sinónimos) del namespace y todas las estaciones de trabajo a las que se hace referencia.
Mail.byaddr	Lista de todos los alias del namespace utilizando la Ethernet address como llave.
Netgroup	Contiene los grupos de la red ocupando los nombres de los grupos como la llave.
Netgroup.byhost	Contiene la información de los grupos ocupando los host name como llave.
Netgroup.byuser	Contiene la información de los grupos utilizando a los usuarios como llave.
Netid.byname	Contiene los números identificadores de los usuarios y de las estaciones de trabajo.
Netmasks.byaddr	Contiene la máscara utilizada con su respectiva IP utilizando la Ethernet address como llave.
Networks.byaddr	Contiene los nombres y sus Ethernet address de las máquinas que están dentro del namespace y sus direcciones IP.
Networks.byname	Contiene los nombres y sus Ethernet address de las máquinas que están dentro del namespace utilizando el nombre como la llave.
Passwd.byname	Contiene la información de los password utilizando el nombre como la llave.
Passwd.byuid	Contiene la información de los password utilizando el userid como la llave.
Protocols.byname	Lista de los protocolos usados.
Protocols.bynumber	Lista de los protocolos de red usados, utilizando su número como llave.
Publickey.byname	Contiene la llave pública y privada para la seguridad de RPC.
rpc.bynumber	Lista de los nombres de programas conocidos y su número de RPCs.
Services.byname	Lista de los servicios disponibles de Internet.
Ypservers	Lista de los servidores de NIS, con sus direcciones IP.

Tabla 2.4 Mapeos de NIS



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

d) NIS+. La red Sun del D.S.C. ocupa el servicio de nombres NIS+ como base para su funcionamiento, por lo que la administración de los usuarios se fundamenta en este servicio.

Descripción de NIS +. NIS+ surge a partir de la versión 2.x de Solaris y pretende substituir a NIS. Cabe aclarar que NIS+ no es la nueva versión de NIS, ya que los conceptos que maneja para la manipulación de los servicios en general son muy diferentes.

La principal finalidad de NIS+ es simplificar el manejo del sistema y de la administración de la red, centralizando la información de las direcciones de las estaciones de trabajo, la información de seguridad y los servicios de red. Una vez que la información está centralizada, puede ser accesada por cualquier máquina de la misma red que tenga los permisos necesarios. A este servicio se le denomina el NIS+ namespace.

El NIS+ namespace es jerárquico y resulta ser similar al sistema de archivos que maneja UNIX, permitiendo de esta forma que el namespace pueda ser dividido en múltiples dominios que pueden ser administrados de forma independiente. Ofrece la ventaja de que estaciones de distintos dominios puedan acceder la información.

En teoría NIS+ incluye una sofisticada seguridad del sistema para proteger la estructura del namespace y su información. Para lo anterior, se utiliza un sistema de autenticación y de autorización que verifica que el cliente tenga la autorización necesaria para acceder y/o modificar información.

Es importante destacar que los únicos sistemas operativos que pueden trabajar con los servicios de NIS+ son Solaris 2.* y las últimas versiones de LINUX (la versión de UNIX más comercial para PC). En un principio se pretendía que todas las versiones de UNIX portaran sus servicios a NIS+, pero a la fecha se ha observado que compañías como Silicon Graphics o IBM aun desconfían de los beneficios que ofrece NIS+ por lo que sus sistemas operativos aún trabajan con el servicio de NIS.

III) El surgimiento de NIS+ como sustituto de NIS.

La arquitectura cliente servidor de red ha crecido enormemente desde mediados de la década de los 80's. En la actualidad se puede decir que en cualquier institución gubernamental, escolar o de otra índole y en cualquier parte del mundo, existe al menos una LAN trabajando con la arquitectura cliente servidor.

El tamaño y complejidad de las redes requiere de la práctica de la administración autónoma, característica que distingue a NIS+.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

Algunas diferencias entre NIS y NIS+

NIS	NIS+
Manejo del dominio de forma plana, no jerárquico.	Manejo de dominio jerárquico, la información es asignada en diferentes niveles dentro del NIS+ namespace.
Los datos de la información de la red son asignados en un mapeo simple de dos columnas.	La información es asignada en una tabla multicolumnas.
El uso de la información no requiere autenticación.	Uso de la autenticación (DES authentication).
Preferencia particular por la información de la red original.	Uso del servicio switch, que permite al cliente escoger entre ocupar el código de información de NIS, NIS+, DNS o archivos locales ubicados en el directorio /etc del cliente.
Atraso de las actualizaciones debido a la propagación en conjunto.	Actualizaciones incrementales propagadas de forma inmediata (crecimiento de la red por medio de creación de subdominios de la red).

Tabla 2.5 Comparación entre NIS y NIS+

La tabla 2.5 hace referencia a las diferencias más importantes y trascendentales en el uso de cualquiera de los dos servicios. Asimismo, en dicha tabla se puede observar que NIS+ ha sido diseñado para que reemplace a NIS, ya que este último fue creado pensando en las necesidades que se tenían en la década de los 80's, en la que se utilizaban sólo unos cuantos cientos de máquinas y en donde la necesidad de seguridad de la información no requería de un sistema sofisticado que hiciera uso de la autenticación.

El manejo de un dominio plano por parte de NIS es una de sus principales desventajas, ya que no está diseñado para el crecimiento de la red. Por su parte, NIS+ sigue los mismos pasos que realizaría una compañía para su crecimiento, por ejemplo, si se tiene una empresa que cuenta con departamentos de compras y ventas, al momento en que crezca la compañía estos departamentos también crecerán y lo más conveniente es que sean manejados de forma independiente; para esto con NIS+ se pueden hacer las adecuaciones pertinentes para que el dominio de la red sea dividido en dos y dé pie a la creación de nuevos dominios. Lo anterior no significa que al dividirse el dominio, las partes serán totalmente independientes ya que NIS+ ofrece la posibilidad de hacerlas tan independientes como se quiera.

Aunque toda la información se encuentra centralizada en algo que es llamado las tablas de NIS+, no significa que no esté segura, ya que además de estar almacenada en el servidor principal (master server), puede estar respaldada en otras máquinas que son conocidas como réplicas (mirrors); estas réplicas son de gran utilidad y tienen muchas ventajas, ya que al momento en que las tablas sufren cambios, la actualización de las mismas en la red es



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

automática, cosa que con los mapeos de NIS no sucede, ya que la actualización es manual, y en algunos casos estas actualizaciones pueden tomar bastante tiempo, por ejemplo en empresas grandes.

NIS+ maneja 16 tablas predefinidas; sin embargo, dependiendo del administrador, este número puede aumentar.

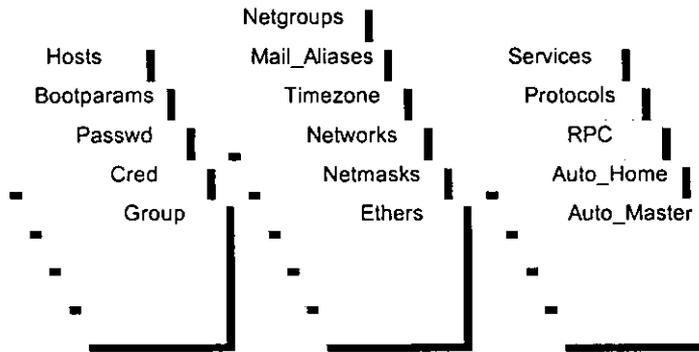


Fig 2.6 Tablas predefinidas de NIS+.

Con el uso de las tablas de NIS+ se tienen 2 características que facilitan el uso de los datos. La primera es que la información puede ser buscada en cualquier columna y no sólo en la primera, con lo que se elimina la necesidad de duplicar mapeos, por ejemplo para buscar información en NIS se requería de utilizar los archivos `/etc/hosts.byname` y `/etc/hosts.byaddr`. El segundo es que con las tablas se pueden manipular los datos en tres niveles de acceso; el de tabla, el de entrada (entry) y el de columna.

Como una medida de protección para la estructura del NIS+ namespace y la información manipulada, se ofrecen procesos complementarios como lo son la autenticación y la autorización, los cuales se describen a continuación:

- ♦ **Autenticación:** NIS+ prueba la autenticidad de cada requisición hacia el namespace. La requisición puede provenir de un "principal" (un proceso, una máquina, la cuenta de root o de cualquier usuario). Para poder llevar a cabo la validación del principal, es necesario checar su credencial principal que se encuentra en una de las tablas de NIS+.
- ♦ **Autorización:** Cada componente del namespace especifica el tipo de operación que aceptará y de donde proviene.



DESCRIPCIÓN DE LA RED SUN DEL D.S.C. DE LA ENEP ACATLÁN

Otra de las ventajas de NIS+ sobre NIS es que el primero puede trabajar de forma conjunta con un programa independiente que se llama servicio de nombres del switch (name service switch), el cual permite que las estaciones de trabajo que utilizan Solaris 2.x puedan obtener su información de más de un servicio de nombres, especificando para máquinas locales los archivos que se encuentran bajo /etc, mapeos de NIS, archivos de zonas del DNS (Domain Name System) o las mismas tablas de NIS+.

III. ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

*" Es detestable esa avaricia espiritual
que tienen los que sabiendo algo, no
procuran la transmisión de esos
conocimientos "*

Miguel de Unamuno



3.1. BREVE INTRODUCCIÓN AL NIS+ NAMESPACE

3.1.1. Estructura del Namespace

Como se mencionó en el capítulo anterior, el NIS+ namespace es el servicio que centraliza la información concerniente a los identificadores de las máquinas, protocolos de comunicación, etc. El NIS+ namespace ha sido designado para conformar la línea de organización de mayor flexibilidad, en cuanto a configuración de red se refiere. El namespace puede estar estructurado en una gran variedad de formas, dependiendo del tipo de organización y necesidades que se tengan. Por ejemplo, si una organización tiene tres divisiones, el namespace puede ser dividido en tres partes (una por cada división). Cada parte almacena información de usuarios, estaciones de trabajo y servicios de red de su división, pero a su vez, puede comunicarse de manera fácil con las demás. El manejo de información del ejemplo, facilitaría las tareas tanto de los usuarios como del administrador.

La forma de organizar el NIS+ namespace puede variar dependiendo del lugar de aplicación; sin embargo, cada sitio ocupa la misma estructura de componentes: directorios, tablas y grupos; los cuales son llamados "objetos de NIS+". Los objetos pueden ser agrupados de forma jerárquica similar al sistema de archivos de UNIX. En el lado izquierdo de la fig 3.1 se muestra un namespace integrado por diferentes objetos: tres directorios, tres grupos y tres tablas. En el lado derecho de la misma figura se muestra parte de un sistema de archivos de UNIX que consiste de tres directorios y tres archivos.

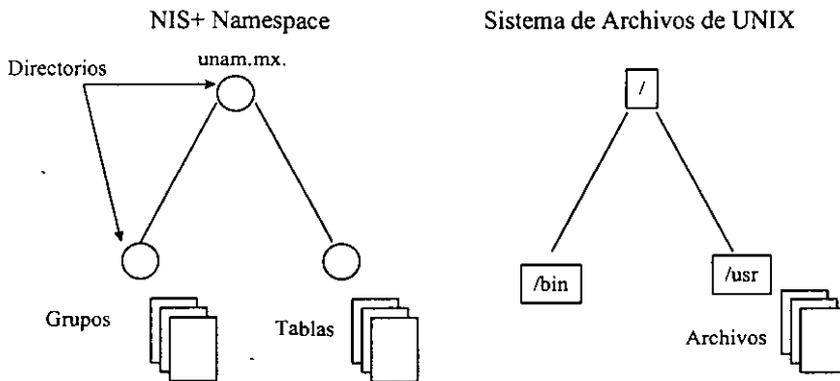


Fig 3.1 Similitud de un NIS+ Namespace con el sistema de archivos de UNIX.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

No obstante a que el namespace tiene mucho parecido al sistema de archivos de UNIX, existen cinco diferencias importantes:

- Aunque ambos usan directorios, en el namespace existen objetos como tablas y grupos, los cuales no son conocidos en el sistema de archivos, en donde sólo se almacenan archivos.
- El namespace únicamente puede ser administrado mediante comandos de NIS+ o por una herramienta gráfica como SOLSTICE, creada para ese propósito; por lo que no puede ser administrado con comandos estándar del sistema de archivos del sistema operativo.
- Los nombres de los componentes del sistema de archivos están separados por “/” (/usr/bin), y en el namespace por “.” (unam.mx.).
- El directorio base o raíz en el sistema de archivos se encuentra a la izquierda de los directorios, y los subniveles se leen de izquierda a derecha (/usr/subdir1/archivo); por el contrario, en el namespace la raíz se encuentra a la derecha y los subniveles se leen de derecha a izquierda (acatlan.unam.mx.).
- Debido a que los objetos del namespace están estructurados de izquierda a derecha los nombres siempre terminan en “.”. Los nombres de objetos en NIS+ que no terminen en punto, serán asumidos como nombres relativos, es decir, no son objetos.

A continuación se describirán algunos de los componentes del NIS+ namespace:

Directorios. Estos objetos son el esqueleto del namespace. Cuando se toma una estructura de tipo árbol invertido, el namespace se divide en partes separadas. Cuando se visualiza la estructura del namespace, la raíz se verá en la parte superior. Si el namespace es plano (no tiene bifurcaciones), solo existirá un directorio, el cual será la raíz. Cualquier directorio que almacene grupos de NIS+ se llama “groups_dir”; el que almacene tablas se llama “org_dir”. Técnicamente se pueden arreglar los directorios, las tablas y los grupos en cualquier estructura que se requiera. Ahora bien, al conjunto de directorios, tablas y grupos dentro de un namespace se le llama “dominio”, el cual es designado para soportar porciones del namespace. Por ejemplo, dentro de la UNAM, un dominio podría abarcar la información de la ENEP Acatlán, otro el de la FES Zaragoza, etc.

Dominios. Un dominio de NIS+ consiste de un directorio org_dir, un conjunto de tablas, un directorio de grupos groups_dir y un conjunto de grupos, tal cual se muestra en la fig 3.2.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

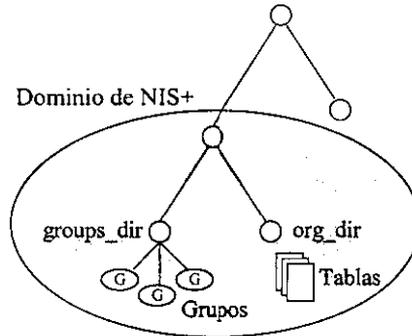


Fig 3.2 Representación de un dominio de NIS +

Los dominios de NIS+ no son elementos tangibles del namespace, tan solo son una forma conveniente de referenciar las secciones usadas para soportar organizaciones del mundo real. Retomando el ejemplo del inciso “a”, en el cual se maneja información de dos escuelas de la UNAM, el namespace se representaría gráficamente como en la fig 3.3, formado por tres directorios principales, seis subdirectorios y varios conjuntos de tablas y grupos, cuya cantidad variará dependiendo de cada dominio.

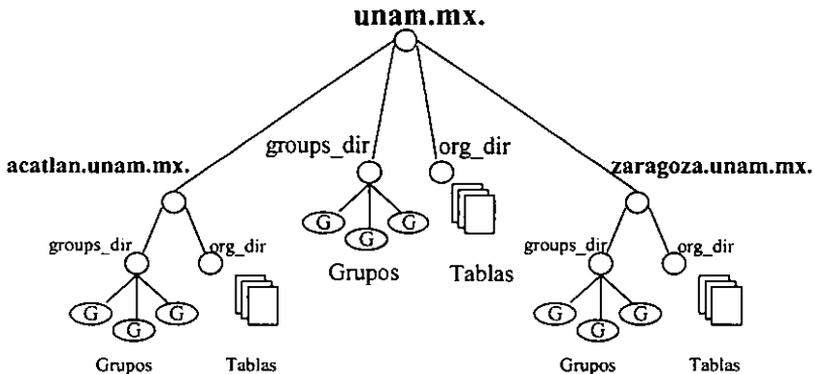


Fig 3.3. Representación de un dominio de NIS+ dividido en dos partes.

Servidores. Cualquier dominio de NIS+ es soportado por uno o un conjunto de servidores de NIS+. Los servidores son los encargados de almacenar los directorios, grupos y tablas de los dominios, además de responder a las requisiciones de información por parte de los usuarios, administradores (root master) y aplicaciones. En términos generales, el dominio puede ser soportado por un conjunto de servidores y dicho conjunto puede soportar varios dominios a la vez.



Toda máquina que se encarga de proveer los servicios del namespace recibe el nombre de servidor de NIS+ y aquellas que se dedican a hacer requisiciones de información son llamadas clientes de NIS+. Dentro de las máquinas que actúan como servidores, existe el "root master server" que es el servidor de la raíz del dominio (solo existirá uno por dominio). También existen servidores de los demás dominios (en caso de que los haya), los cuales son llamados "master servers". En caso de que existan servidores de respaldo de NIS+ estos se identifican como réplicas.

Tanto los servidores master o principales y las réplicas almacenan la información de las tablas y responden a las peticiones de información de los clientes. Por su parte, el servidor principal guarda la copia más importante de la información de las tablas. Cuando el administrador modifica la información, lo hace sobre el servidor principal, el cual se encarga de propagar la información necesaria a través de los clientes.

Cientes y Principals. Los NIS+ principals son entidades (clientes) que hacen requisiciones de servicios.

- ♦ **Principals.** Un principal puede ser alguien que se conecta en una máquina cliente como un usuario regular o como superusuario (root). En el primer caso, el requerimiento proviene de un usuario que es cliente; en el segundo caso la requisición viene de la estación de trabajo que también es un cliente. Por lo tanto, un principal puede ser tanto un usuario como una estación de trabajo.
- ♦ **Cliente.** Una estación cliente es aquella que ha sido preparada para recibir los servicios de NIS+ y que puede acceder cualquier parte del namespace sujetándose a las restricciones de seguridad pertinentes. Como parte de la seguridad, la máquina deberá tener credenciales en las tablas del servidor principal, además verificará su dominio hogar y revisará su archivo de switch. Cabe aclarar que los servidores de NIS+ también son clientes, exceptuando el "root master server".

3.1.2. Estructura de las Tablas e Información

Las tablas de NIS+ proveen mucha información que no está contenida en archivos simples de texto o en mapeos como en NIS. Este tiene una estructura de columnas de entrada, las cuales aceptan rutas de búsqueda que pueden ser ligadas entre sí y que pueden ser configuradas de diferente forma. NIS+ provee 16 tablas preconfiguradas (Véase la Tabla 3.1) y el administrador puede crear sus propias tablas.



Información contenida en la tabla	
hosts	IP address y Host name de cada estación de trabajo del dominio.
Bootparams	Ubicación de la raíz, swap y las particiones de cada cliente de tipo diskless ¹ en el dominio.
Passwd	Información referente a los password de todos los usuarios del dominio.
Cred	Credenciales para principal que se conectan al dominio.
Group	El nombre del grupo, el grupo del password, el ID del grupo, y los miembros de cada grupo UNIX en el dominio.
Netgroup	Los grupos de red a los cuales se pueden conectar los usuarios y las estaciones de trabajo.
mail_aliases	Información acerca de los sinónimos de correo de los usuarios del dominio.
Timezone	La zona horaria de cada estación de trabajo en el dominio.
Networks	Las redes en el dominio y su nombre canónico.
Netmasks	Las redes en el dominio y su correspondiente máscara.
Ethers	La dirección ethernet de cada estación de trabajo en el dominio.
Services	Los nombres de los servicios del protocolo IP y el número de puerto correspondiente.
Protocols	Los protocolos de red manejados por el S.O.
RPC	Los números de los programas para los servicios de RPC, en el dominio.
Auto_home	La ubicación de los directorios hogar de todos los usuarios en el dominio.
auto_master	La información de los mapeos automontables.

Tabla 3.1. Tablas predefinidas de NIS +.

I) Columnas y Entradas.

En general, todas las tablas tienen la misma estructura, están compuestas por columnas y renglones conocidos como entradas. Un cliente puede acceder información de las tablas a través de una llave o por medio de una columna revisable. Por ejemplo: para encontrar la dirección IP de una estación de trabajo llamada "orion", el cliente puede buscar en la columna referente al hostname hasta encontrar el nombre deseado, y así ya tendría ubicado el renglón correspondiente a orion. Finalmente el cliente se moverá a través del renglón hasta llegar a la columna referente a la dirección IP (Fig. 3.4.), obteniéndose así la información deseada.

¹ El diskless es la forma de identificar un tipo de instalación que se hace del S.O. en máquinas que tienen poca capacidad de disco duro.

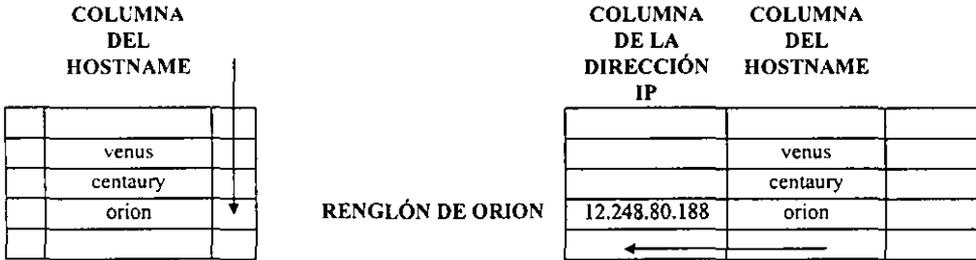


Fig 3.4. Ejemplo de búsqueda de una dirección IP.

Debido a que los clientes pueden acceder la información de las tablas en cualquier nivel, NIS+ provee mecanismos de seguridad para todos los niveles de la estructura de árbol que maneja. El administrador de la red puede asignar los siguientes permisos²:

- De lectura para cualquiera en la tabla a nivel objeto.
- De modificación para el propietario a nivel columna.
- De modificación para el grupo a nivel entrada.

II) Actualización de tablas.

Cuando un dominio es dado de alta, los servidores reciben la primera versión de las tablas. Esta versión se encuentra almacenada en el disco duro, pero cuando el servidor empieza a operar, llama a las tablas y las sube a memoria. Cuando el servidor recibe una modificación de las tablas, actualiza inmediatamente la copia que está almacenada en memoria. En el caso de requisiciones, el servidor ocupa la información almacenada en memoria.

Existe un momento en el cual el servidor almacena la información en las tablas del disco duro; este proceso lleva tiempo, por lo que los servidores de NIS+ manejan archivos "log", los cuales son designados para almacenar los cambios de forma temporal hasta que las tablas del disco duro sean actualizadas. Los nombres de estos archivos se componen con el nombre de la tabla que sufrirá cambios, anexándole el prefijo ".log"; por ejemplo:

```
hosts.org_dir.log  
bootparams.org_dir.log  
passwd.org_dir.log
```

3.1.3. Seguridad del Sistema

En esencia la seguridad de solaris esta provista por puertas que los usuarios necesitan pasar en orden para entrar al ambiente del S.O. y por matrices de permisos que determinan qué le es permitido al usuario una vez que está dentro de éste; la fig. 3.5 muestra el proceso.

² En puntos subsiguientes se tratarán de forma más profunda los permisos.

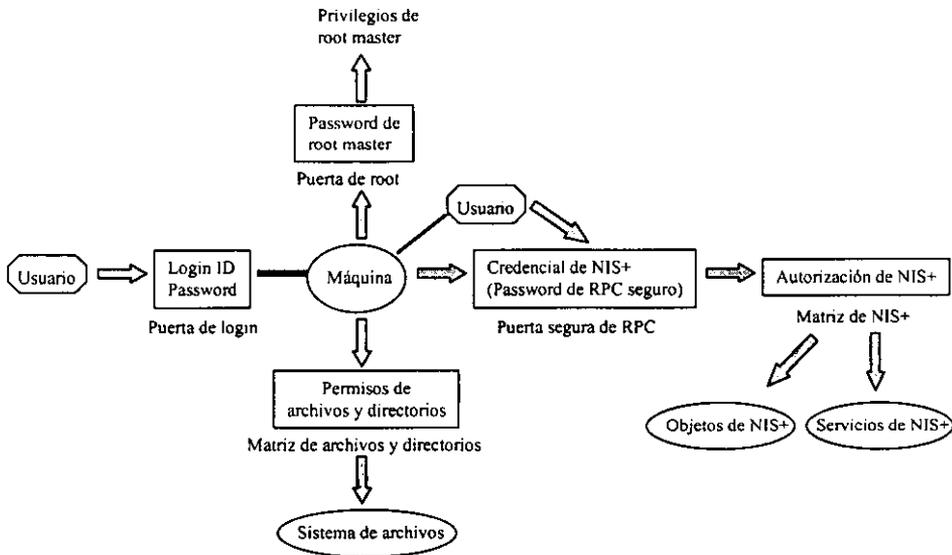


Fig. 3.5 Ejemplo de las puertas y filtros de seguridad en Solaris.

- Puerta de login. Para entrar al ambiente del sistema operativo se necesita proveer un login y un password.
- Matriz de archivos y directorios. Una vez que se ha accedido al sistema la habilidad para leer, ejecutar, modificar, crear y destruir archivos y directorios es controlada por los permisos aplicables de la matriz.
- Puerta de root. Para tener acceso a la cuenta de root se requiere de un password de superusuario.
- Puerta segura de RPC. En ambientes de NIS+ corriendo a niveles de seguridad 2 (es el que se maneja por default), cuando se intenta usar los servicios de NIS+ se realiza un chequeo de acceso a los objetos (servidores, directorios, tablas, entradas de tablas, etc.); la identidad del usuario es confirmada por NIS+ usando el proceso seguro de RPC.

Para pasar la puerta segura de RPC se requiere una presentación del password de RPC. Normalmente el password de RPC y el password del usuario son los mismos y cuando éste es el caso, se pasa la puerta de forma automática sin tener que volver a teclear el password.

Un grupo de credenciales son ocupadas para pasar la puerta de RPC. El proceso de generar, presentar, y validar las credenciales es llamado "*autenticación*" porque confirma quién es el usuario y que tiene un password válido de RPC. Esta autenticación es realizada de forma obligatoria y automática cada vez que se hace una requisición de servicio.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

- Matriz de objetos de NIS+. Una vez que se tiene la apropiada autenticación de NIS+, la habilidad del usuario para leer, modificar, crear y destruir objetos de NIS+ es controlada por la matriz de permisos aplicables. Este permiso es llamado "autorización".

1) Seguridad de NIS+.

Debido a que la seguridad de NIS+ es un poco compleja se divide en los siguientes incisos.

a) **Autenticación y autorización.** La seguridad de NIS+ es una parte integral del namespace, por lo que no se puede dar de alta la seguridad y el namespace de forma independiente. Son dos los puntos primordiales que abarcan la seguridad dentro del NIS+ namespace.

- ◆ **Autenticación.** NIS+ usa credenciales para confirmar quién es el usuario que pretende acceder al sistema (en el caso específico en el cual los password de RPC y login sean distintos, se tendrá que realizar un paso más antes de acceder al sistema, llamado keylogin).
- ◆ **Autorización.** Una vez que la identificación se encuentra establecida por el proceso de autenticación, NIS+ determina la clase del usuario identificado. Todo lo que pueda hacer el usuario con respecto a los objetos de NIS+ y el sistema mismo, dependerá directamente de la clase a la que pertenezca.

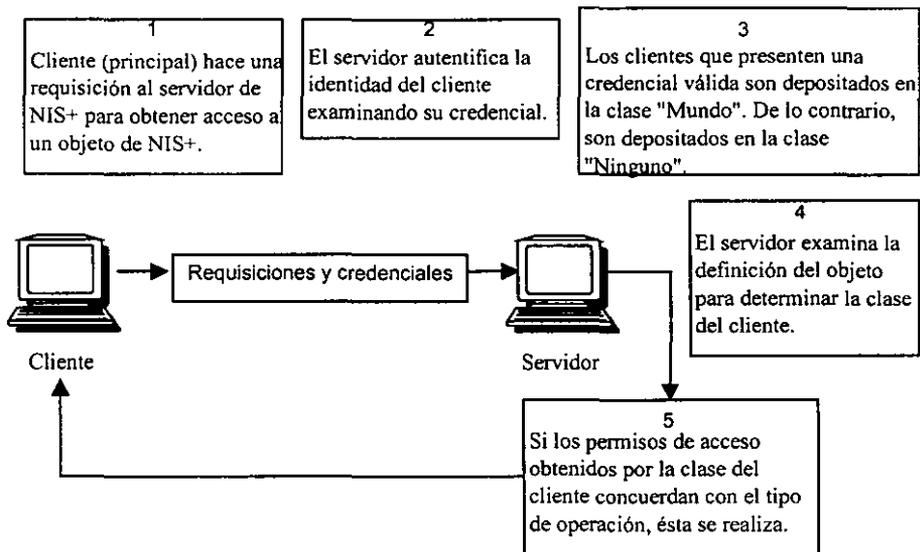


Fig. 3.6 Esquema del proceso de seguridad de NIS +.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

- b) **NIS+ principals.** Son entidades (clientes) que envían requisiciones de información para los servicios de NIS+. Un NIS+ principal puede ser cualquier usuario regular conectado en una máquina cliente, alguien que se encuentre conectado como superusuario, o cualquier proceso que corre con permisos de superusuario en una máquina cliente.
- c) **Niveles de seguridad de NIS+.** Los servidores de NIS+ operan a uno o dos niveles de seguridad³. NIS+ esta designado para correr al nivel más seguro que el nivel 2. El nivel 0 es provisto con propósitos de pruebas, configuración y depuración.

La siguiente tabla presenta una breve descripción de los niveles de seguridad.

Nivel de seguridad	Descripción
0	Un servidor corriendo en el nivel de seguridad 0 garantiza permisos de acceso total a los objetos de NIS+ en el dominio. El nivel 0 esta diseñado con propósitos de configuración y sólo deberá ser utilizado por el administrador del sistema. Este nivel no deberá ser utilizado en condiciones normales de operación de la red por usuarios regulares.
1	Este nivel ocupa la seguridad AUTH_SYS. No se profundizará debido a que los servidores de NIS+ no soportan este nivel y por ende nunca es ocupado.
2	Este nivel es el considerado como el más seguro provisto por NIS+. Autentifica solo las requisiciones que utilizan las credenciales de tipo DES. Requisiciones que no tengan credenciales son asignadas a la clase de "nobody" y tendrán los permisos de acceso correspondientes a esta clase.

Tabla 3.2. Niveles de seguridad dentro de NIS+.

II) Introducción a la autenticación y credenciales.

El propósito de las credenciales de NIS+ es autenticar (confirmar) la identidad de las requisiciones de un servicio o acceso a los objetos de NIS+. En esencia, el proceso de credencial/autenticación es una implementación del sistema de seguridad de RPC. Una de las tareas principales del proceso de credencial/autenticación, es el prevenir que un usuario trate de asumir la identidad de otro usuario sin los permisos debidos; esto es, previene que alguien que tenga privilegios de super-usuario en una máquina por medio del comando "su", asuma la identidad de un segundo usuario y de esta forma accesar a los objetos de NIS+ con los permisos del segundo usuario.

Con respecto a los usuarios y a las máquinas existen dos tipos de "principal" los cuales utilizan distintos tipos de credenciales:

³ Los niveles de seguridad determinan los tipos de credenciales principales que serán enviadas para hacer la autenticación de requisiciones de información.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

- ◆ Credenciales de usuario. Son aquellas que se utilizan cuando un usuario regular está conectado a un cliente y se están haciendo requisiciones hacia el servidor.
- ◆ Credenciales de máquina. Cuando un usuario se conecta a un cliente de NIS+ con la cuenta de root master del cliente y hace requisiciones, está haciendo uso de las credenciales de la máquina.

Por su parte, los “NIS+ principals” manejan dos tipos de credenciales: DES y LOCAL, los cuales se describen a continuación.

- ◆ **DES (Data Encryption Standard).** Es el tipo de credencial que provee la autenticación segura. Siempre se requerirá de una credencial de tipo DES para poder pasar en orden la autenticación del sistema y poder tener la autorización de las clases de propietario, grupo y mundo (world class).
- ◆ **LOCAL.** Son simplemente mapas entre los identificadores numéricos de los usuarios y los nombres de los “NIS+ principals” que incluyen el nombre del dominio hogar. Cuando los usuarios se conectan, el sistema busca su credencial LOCAL para saber cual es su dominio hogar en el cual se encuentra almacenada su credencial DES.

III) Introducción a la autorización y acceso.

El propósito básico de la autorización es especificar los permisos del NIS+ principal respecto a los objetos y servicios. Una vez que una requisición ha pasado la autenticación, NIS+ la asigna a una “clase de autorización”, la cual se encarga de asignar los permisos de acceso que tiene respecto a los objetos y servicios de NIS+.

Los objetos de NIS+ no permiten el acceso de forma directa a los NIS+ principals; sino que permiten el acceso a los permisos de cuatro de las clases del principal, descritas en la tabla 3.3.

Clase	Descripción
Propietario	Es la clase en la que el propietario es un simple NIS+ principal. Por default el propietario de un objeto es el principal que creó el objeto.
Grupo	Es en la que el grupo de objeto es un simple grupo de NIS+. Un grupo de NIS+ es una colección de NIS+ principals, agrupados por conveniencia para dar acceso al namespace. Es importante mencionar que la información concerniente a los grupos de NIS+ se encuentra almacenada debajo del directorio <i>groups_dir</i> de cada dominio de NIS+.
Mundo	Contiene todos los NIS+ principals que han sido autenticados por NIS+. En otras palabras cualquiera que pertenezca a las clases de propietario y de grupo.
Ninguno	Esta compuesto por cualquiera que no esté propiamente autenticado. Es decir, cualquiera que no presente una credencial DES válida.

Tabla 3.3. Clases existentes dentro de NIS +.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

Como parte de la autorización es necesario asignar los permisos de los objetos de NIS+ a las clases arriba mencionadas. Los permisos de acceso son asignados en la misma forma que lo maneja UNIX para los archivos. Además, especifican el tipo de operación que los NIS+ principales están permitidos a ejecutar en los objetos de NIS+. Las operaciones que se realizan con NIS+ varían mucho, dependiendo del objeto que se esté manejando, pero siempre residirán en uno de los siguientes 4 permisos de acceso:

- ◆ Lectura. Permite acceder a la información sin poder realizar cambios de ningún tipo.
- ◆ Modificar. Permite modificar el contenido del objeto.
- ◆ Destruir. Permite destruir o borrar el objeto.
- ◆ Crear. Permite crear objetos dentro del namespace, dependiendo del nivel en el que se haya dado el permiso. Es decir, si se tiene el permiso de crear sobre un directorio de NIS+, se podrán generar nuevas tablas dentro de ese directorio. Si se tiene este permiso sobre una tabla, entonces se podrán crear nuevas columnas y entradas.

3.1.4. Credenciales

Para entender cómo son creadas y como trabajan las credenciales, es necesario distinguir entre las credenciales y la información utilizada para su creación y verificación.

I) Información de la credencial.

Son los datos usados tanto para generar la credencial DES como por el servidor para verificar la misma.

II) La credencial DES.

Es el conjunto de caracteres que es enviado por el principal al servidor para realizar el proceso de autenticación. La credencial del principal es generada y verificada cada vez que hace una requisición de NIS+. La credencial DES esta compuesta por:

- a) El nombre de red seguro de RPC para el principal. Esta porción de la credencial es usada para identificar al principal de NIS+. Cada nombre esta compuesto por 3 elementos:
 - ◆ Prefijo. El prefijo siempre es la palabra "unix".
 - ◆ Identificador. Si el principal es un usuario, el campo ID es el UID⁴ del usuario.
 - ◆ Nombre del dominio. Es el nombre del dominio al cual pertenece el principal.

⁴ El UID (User Identifier) es un identificador numérico único que asigna la máquina a cada usuario del sistema.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

	Prefijo	Identificador	Dominio	Ejemplo
Usuario	unix	UID	El Dominio que contiene el password y la credencial del usuario	unix.24601@acatlan.unam.mx
Máquina	unix	Host name	El dominio que arroja la ejecución del comando domainname en la estación de trabajo	unix.zeus@acatlan.unam.mx

Tabla 3.4. Composición del nombre de red seguro de RPC.

b) Un campo de verificación. El campo de verificación sirve para asegurarse de la creación de la credencial. Esta es generada por medio de la información de la credencial almacenada en la tabla cred.org_dir. Los elementos que constituyen este campo son:

- ◆ La llave encriptada DES del principal, generada desde la llave privada del principal y la llave pública del servidor.
- ◆ El tiempo señalado encriptado.
- ◆ El tiempo de la ventana.

La información concerniente a las credenciales es almacenada en diferentes partes del namespace. NIS+ actualiza las tablas de forma periódica y en algunas ocasiones lo hace a petición del administrador del sistema. Véase la tabla 3.5. para ubicar los lugares donde se almacena la información concerniente a las credenciales.

Lugar	Información que almacena.
Tabla cred.org_dir	La llave privada y pública del principal de NIS+. Estas son las copias maestras de las llaves.
Objeto de tipo directorio	La copia de la llave pública del servidor
Keyserver	La llave secreta del principal que se encuentra conectado al momento.
Demonio ⁵ de NIS+	Copia de los objetos de tipo directorio que voltean el contenido de las copias de las llaves publicas de los servidores.
Directorio cache	Una copia de los objetos de tipo directorio que voltean el contenido de las copias de las llaves publicas de los servidores.
Archivo cold-start	Una copia de los objetos de tipo directorio que voltean el contenido de las copias de su propia llave pública.
Tabla passwd.org_dir	Los password encriptados de los usuarios.
Archivo passwd	El password del root master de la máquina en cuestión.

Tabla 3.5. Ubicación de información correspondiente a las credenciales.

⁵ Un demonio es un programa que se encuentra siempre activo y que reside en la memoria de la máquina esperando un suceso específico para que sea ejecutado.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

La tabla `cred.org_dir`, es la que contiene la información de credenciales de todos los principals que se pueden conectar al dominio, almacena credenciales locales para los usuarios que tienen permiso de conectarse al sistema a través de las estaciones de trabajo. En el caso de los usuarios que tienen el dominio como hogar, se almacena la credencial DES.

III) Métodos para crear y administrar credenciales.

A continuación se describen algunos métodos para crear y administrar las credenciales:

- Usar el script⁶ llamado "nisclint", el cual es el más conveniente en la administración de credenciales individuales.
- Usar el comando "nisaddcred" del S.O., el cual crea, actualiza y elimina credenciales Locales y DES. Ejemplos:

Para creación de credenciales de tipo Local.

```
nisaddcred -p UID -P nombre-principal local
```

Para creación de credenciales de tipo DES

```
nisaddcred -p UID -P nombre-de-red-rpc local
```

Para actualizar la credencial propia Local

```
nisaddcred local
```

Para actualizar la credencial propia DES

```
nisaddcred -DES
```

Para eliminar credenciales

```
nisaddcred -r nombre-de-red-rpc
```

Obsérvese que en los ejemplos anteriores se maneja un identificador no mencionado hasta el momento, que es *nombre-principal*. De ahora en adelante se deberá entender por este identificador a aquel que está formado por el nombre del principal, aunado al dominio de NIS+ al que pertenece el cliente en cuestión. Véase la tabla 3.6. para un mejor entendimiento.

⁶ Un Script es un archivo que contiene un programa codificado en base a comandos de línea del S.O.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

principal	Nombre	Dominio	nombre-principal
Usuario	patty	acatlan.unam.mx	patty.acatlan.unam.mx.
Máquina	vulcano	acatlan.unam.mx	vulcano.acatlan.unam.mx.

Tabla 3.6. Componentes del nombre-principal.

c) En caso de tenerse disponibles, ocupar las herramientas gráficas de Solstice ya que proveen métodos sencillos para administrar las credenciales.

3.1.5. Solstice el Manipulador de los Servicios de NIS+

Solstice es una herramienta gráfica que nace a partir de Solaris 2.5 con la finalidad de administrar el servicio de nombres de una forma fácil, sin importar cual es el servicio utilizado (NIS, NIS+ o los archivos de /etc). Solstice está compuesto por un conjunto de 7 programas básicos que facilitan la administración y manipulación de la información contenida en las bases de datos (en el caso de NIS+, las tablas ya conocidas).

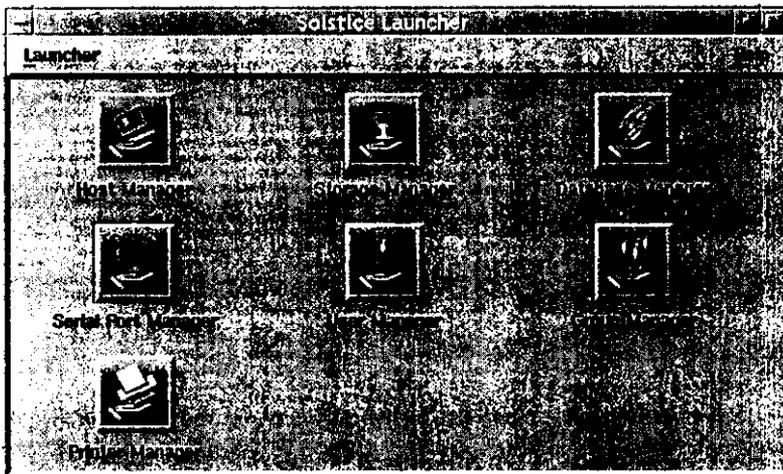


Fig 3.7. Pantalla principal del programa Solstice.

Cada uno de los programas tiene tareas bien definidas y realiza cambios sobre tablas específicas:



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

- Host Manager. Es el encargado de manipular la información concerniente a los hosts de la red (tabla `hosts.org_dir`)
- Storage Manager. Maneja la forma de organización de información en el sistema de archivos. (archivo `/etc/nsswitch.conf`)
- Database Manager. Manejo de todos los servicios. (todas las tablas de NIS+).
- Serial Port Manager. Visualizador de los puertos seriales.
- User Manager. Manejador individual de usuarios y sus credenciales: creación, borrado y modificación (tablas `passwd.org_dir` y `cred.org_dir`).
- Group Manager. Manejo de los grupos a los que pueden pertenecer los usuarios: creación, borrado y modificación (tabla `group.org_dir`)
- Printer Manager. Manejador de las impresoras locales y de red (archivos relacionados a las impresoras bajo el directorio `/etc`).

Solstice es una herramienta que facilita bastante la administración de los servicios de NIS+, ya que elimina casi por completo la necesidad de ocupar comandos de línea de NIS+ como lo son `nisaddcred` o `nistbladm`. Sin embargo, desde el punto de vista del que escribe, esta herramienta carece de componentes que ayudarían a la administración de los usuarios de una forma más eficaz y rápida, lo cual es el motivo del presente trabajo.

3.2. MANEJO DE SEGURIDAD RESPECTO A LOS USUARIOS

3.2.1. *Permisos de Acceso a las Tablas*

Cuando se crea un objeto de NIS+, éste último asigna un conjunto de permisos por default a las clases de grupo y propietario. El dueño del objeto es aquel que lo crea a menos que se indique lo contrario. El grupo al que pertenece el nuevo objeto será el que se encuentre asignado en la variable de ambiente⁷ “`NIS_GROUP`”.

Para especificar diferentes permisos a los asignados por default, es necesario seguir alguno de los dos procedimientos siguientes:

- Modificar la variable de ambiente “`NIS_DEFAULTS`”, la cual almacena el conjunto de información de seguridad asignada por default, en la parte que hace referencia a los permisos de acceso.
- Ocupar la opción “`-D`”, que está disponible en gran cantidad de comandos de NIS+. Cuando se ocupa esta opción, se le indica al sistema que se salte los defaults que se tienen asignados.

⁷ Las variables de ambiente son contenedores de información que se encuentran residentes en memoria y que sirven para indicarle al sistema configuraciones específicas.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

En el caso de requerirse la modificación de los permisos de acceso se podrá hacer por medio de los comandos "nischmod" y "nistbladm" (para modificar las columnas de las tablas).

I) Seguridad de las tablas, columnas y entradas.

Las tablas de NIS+ permiten especificar permisos sobre las mismas en tres formas: se pueden dar permisos sobre toda la tabla, sobre la columna de forma individual y sobre la entrada también de forma individual.

Los permisos de acceso a nivel columna y entrada (renglón) dan la oportunidad de manejar permisos más precisos, los cuales sobreescriben los permisos de restricción a nivel tabla. Cabe aclarar que los permisos de columna y renglón sólo sirven para agregar permisos de acceso; de ninguna forma servirán para quitarlos.

Por ejemplo, supóngase que existe una tabla que no da permisos de acceso a las clases Mundo y Ninguno; es decir no podrán crear, borrar, leer o modificar información en la tabla; se podrán ocupar los permisos de acceso de renglón y columna para que estas clases puedan realizar alguna o todas las operaciones mencionadas con la información en dicha tabla. En el caso inverso, si la tabla garantiza que las clases Mundo y Ninguno tienen permisos de acceso, los permisos de renglón y columna, no pueden hacer nada para quitar los permisos asignados por la tabla.

En base a las consideraciones del párrafo anterior, se puede formar casi cualquier combinación necesaria para el manejo de permisos, tomando en cuenta que las acciones permitidas en base a las clases y a los permisos de acceso serán los que presentan en la tabla 3.7.

	Directorio	Tabla	Columna	Renglón
Leer	Listar el contenido del directorio	Visualizar el contenido de la tabla	Visualizar el contenido de la columna	Visualizar el contenido del renglón
Creación	Creación de nuevos directorios o tablas	Agregar nuevos renglones	Introducción de nuevos datos en la columna	Nada
Modificación	Cambia los nombres de los objetos	Cambia los valores de los datos en cualquier parte de la tabla	Cambia el valor de los datos de una columna	Cambia el valor de los datos de un renglón
Eliminar	Borra objetos del directorio como tablas	Borra renglones	Borra valores de datos en una columna	Borra valores de datos en un renglón

Tabla 3.7. Permisos que maneja NIS +.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

Los permisos de acceso son asignados y almacenados como parte de la definición del objeto y no son almacenados en tablas de NIS+. Para ver los permisos de acceso que tiene un objeto es necesario ocupar el comando "niscat" como lo muestra el siguiente ejemplo:

```
niscat -o objeto
```

Esta forma de visualizar los permisos provee la siguiente información del objeto: propietario, grupo, permisos de acceso para las clases a) Ninguno, b) Propietario, c) Grupo d) Mundo. Los permisos de acceso para las cuatro clases son presentados en una secuencia de 16 caracteres como la siguiente:

r---rmdr---r---

donde cada caracter representa un tipo de permiso de acceso:

- r** representa permiso de lectura.
- m** representa permiso de modificación.
- c** representa permiso de borrado.
- d** representa permiso de creación.
- representa un no permiso.

Para identificar los permisos de acceso, cada clase tiene asociados 4 caracteres en posiciones bien especificas como lo muestra la fig. 3.8.

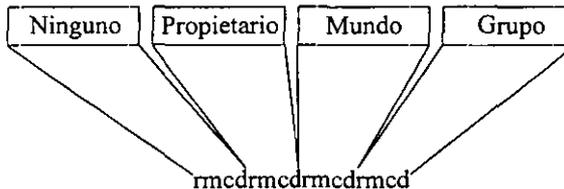


Fig. 3.8. Despliegue de permisos de acceso.

Para entender como se modifican estos permisos es necesario entender que se manejan tres tipos de argumentos que son:

a) Clase.

Clase	Referencia
Ninguno	n
Propietario	o
Grupo	g
Mundo	w
Todas las clases	a



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

b) Operador.

Operador	Descripción
+	Adiciona el permiso de acceso especificado.
-	Revoca el permiso de acceso especificado
=	Reemplaza los permisos de acceso existentes por los nuevos permisos especificados

c) Permiso.

Permiso	Descripción
r	Lee la definición del objeto o la entrada de la tabla
m	Modifica la definición del objeto o la entrada de la tabla
c	Crea una entrada o una columna de la tabla
d	Destruye una entrada o columna de la tabla

Para una mejor comprensión de la manipulación de los permisos se presentan los siguientes ejemplos:

Operaciones	Sintaxis
Adiciona el permiso de acceso a la clase Propietario.	o+r
Cambia los permisos de todas las clases para sólo tener el acceso de modificación.	a=m
Adiciona los permisos de lectura y modificación a las clases Mundo y Ninguno.	wn+rm
Elimina los cuatro permisos de acceso a las clases Grupo, Mundo y Ninguno.	gwn-rmcd

Para referenciar las entradas de las tablas que se modificarán, es necesario referenciarlas en base a un indexado⁸ generado de la siguiente forma:

[Nombre_de_columna=valor],Nombre_de_la_tabla

Los nombres indexados pueden especificar más de un valor de columna, de tal manera que la operación aplica sólo sobre aquellos renglones que cumplen con todos los valores especificados. Ejemplos:

⁸ Un indexado es una selección de datos realizada en base a una característica específica. Los indexados normalmente se realizan en bases de datos, lo cual ayuda a la manipulación de la información contenida; en el caso de NIS+, funciona de forma semejante, con la diferencia de que aquí la selección de datos se hace sobre la tabla de NIS+ indicada.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

Tipo	Ejemplo
Renglón de la tabla.	'[uid=33555],passwd.org_dir.acatlan.unam.mx.'
Dos valores del renglón de la tabla.	'[name=reyna,gid=2],group.org_dir.unam.mx.'

II) Cambios de permisos de objetos y renglones.

El servicio de NIS+ provee el comando "nischmod" que manipula los permisos de los objetos o renglones de una tabla. En el caso de necesitarse modificaciones de permisos referentes a columnas, es necesario ocupar el comando "nistbladm" con la opción "-D". Para realizar modificaciones con el comando nischmod primero es necesario tener permisos sobre el objeto o el renglón. Ejemplos:

a) Agregando permisos.

- ◆ Para un objeto.

```
nischmod clase+permiso Nombre_del_objeto
```

Supóngase que se quieren agregar los permisos de lectura y modificación al grupo de acatlan.unam.mx. la instrucción necesaria es:

```
nischmod g+rm acatlan.unam.mx.
```

- ◆ Para el renglón de una tabla.

```
nischmod clase+permiso [Nombre_de_columna=valor],Nombre_de_tabla
```

Para agregar los permisos de lectura y modificación para el grupo con la entrada "name=centaury" en la tabla hosts.org_dir.acatlan.unam.mx. la instrucción correspondiente sería:

```
nischmod g+rm '[name=centaury],hosts.org_dir.acatlan.unam.mx.'
```

- ◆ Para una columna.

```
nistbladm -u [columna=Permisos_de_acceso,...],Nombre_de_tabla
```

El comando siguiente agrega permisos de lectura y modificación para las columnas "name" y "addr" en la tabla hosts.org_dir.acatlan.unam.mx.

```
nistbladm -u '[name=g+rm,addr=g+rm], hosts.org_dir.acatlan.unam.mx.'
```



b) Quitando permisos.

- ◆ Para un objeto.

Para quitar los permisos de creación y destrucción del grupo de `acatlan.unam.mx` :

```
nischmod g-cd acatlan.unam.mx.
```

- ◆ Para un renglón.

Para quitar el permiso de destrucción del grupo con la entrada “name=centaury”:

```
nischmod g-d '[name=centaury],hosts.org_dir.acatlan.unam.mx.'
```

- ◆ Para una columna.

Para quitar los permisos de lectura y modificación de la columna “name” en la tabla `hosts.org_dir.acatlan.unam.mx`.

```
nistbladm -u '[name=g-rm], hosts.org_dir.acatlan.unam.mx.'
```

3.2.2 Información Concerniente a los Passwords de los Usuarios

El password de cada usuario como ya se ha comentado, se encuentra almacenado en la tabla “`passwd.org_dir`”. Esta tabla y en general el concepto de password no solamente se refiere a la contraseña encriptada del usuario, sino al conjunto de información relacionada con el mismo. En general, la existencia de un usuario dentro del dominio, depende directamente del renglón correspondiente en la tabla `passwd.org_dir`, es decir que un usuario es reconocido por el sistema gracias a que existe un renglón correspondiente para dicho usuario. La información de la tabla `passwd.org_dir` contiene las 7 columnas descritas en la tabla 3.8.

Columna	Descripción
Name	El login asignado al usuario, cuando éste es creado.
Passwd	El password encriptado del usuario.
UID	El identificador numérico del usuario.
GID	El identificador numérico del grupo del usuario.
GCOS	El nombre completo del usuario, o el identificador con el que el usuario se da a conocer en el sistema.
Home	El directorio hogar del usuario.
Shell	El intérprete de comandos usado por el usuario.
Shadow	Véase la tabla 3.9. para ver el contenido de esta columna.

Tabla 3.8. Información contenida en la tabla `passwd.org_dir`



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

Descripción	
Lastchg	El número de días entre el 1 de enero de 1970 y la fecha de la última modificación del password.
Min	El número mínimo de días recomendado entre los cambios del password.
Max	El número máximo de días que el password es válido.
Warn	El número de días de advertencia de expiración de password antes de que expire.
Inactive	El número de días de inactividad permitidos para el usuario.
Expire	La fecha de expiración del password del usuario.
Flag	Reservada para uso futuro: actualmente teniendo el valor de 0.

Tabla 3.9 Información de la columna shadow de la tabla passwd.org_dir

La administración y manipulación de información de la tabla passwd.org_dir, se resume en el manejo de dos comandos de NIS+:

I) nispasswd.

La finalidad de este comando es poder modificar información concerniente al renglón de un determinado usuario en la tabla passwd.org_dir. El uso de este comando variará considerablemente dependiendo de los permisos de acceso que tenga asignados el renglón del usuario. En el caso del administrador del sistema, éste tiene los permisos necesarios para realizar cambios de cualquier índole en el renglón de cualquier usuario; por otra parte, un usuario simple solamente podrá utilizar este comando para cambiar su password encriptado, sustituyéndolo por otro.

Para realizar los cambios necesarios de información, el comando proporciona una serie de opciones específicas para realizar esta tarea. Ejemplos:

Para cambiar el directorio hogar del usuario rodrigo (solamente lo puede hacer el administrador):

```
nispasswd -h rodri
```

El comando para asignar el número máximo de días válidos del password de rodrigo a 20 es (solamente lo puede hacer el administrador):

```
nispasswd -x 20 rodrigo
```

Para que un usuario pueda cambiar su password (solamente lo puede hacer el usuario):

```
nispasswd
```



II) nistbladm.

Como ya se comentó, Solstice es la herramienta más recomendada para manejar la información contenida en las tablas, pero en caso de no existir Solstice, el manejo de la información tendría que ser en base al comando "nistbladm". Lo mismo sucede con la tabla passwd.org_dir y por medio de este comando se pueden realizar las siguientes operaciones:

- Crear renglones para nuevos usuarios (uno de los pasos en la creación de un usuario del sistema).

Ejemplo:

```
nistbladm -a name=reyna uid=301 gid=1 gcos="Reyna Sandoval" home=/home/reyna  
shell=/bin/csh passwd.org_dir
```

- Borrar renglones de usuarios (uno de los pasos en la eliminación de un usuario del sistema). Ejemplo:

```
nistbladm -r name=reyna passwd.org_dir
```

- Cambiar información de los renglones (modificar información relacionada con el password de los usuarios).

```
nistbladm -m shadow=n1:n2:n3:n4:n5:n6:n7 [name=reyna],passwd.org_dir
```

donde n1=Lastchg, n2=Min, n3=Max, n4=Warn, n5=Inactive, n6=Expire, n7=Flag.

Con el comando nistbladm se puede realizar casi cualquier operación necesaria, pero por seguridad y por lo delicado de NIS+, deberá ser ocupado solamente por administradores con experiencia.

3.2.3 Tablas relacionadas con el manejo de los Usuarios e importancia de las mismas

Como se ha venido comentando a lo largo de este capítulo, las tablas sirven para almacenar información, pero no todas están ligadas de forma directa a los usuarios, por lo que es conveniente mencionar cuales son las más importantes para el manejo de los usuarios.

auto_home.org_dir. Cuando un usuario entra al sistema por medio de un cliente, tiene acceso a toda la información que contiene su directorio hogar, esto es logrado gracias al servicio de NFS y a la tabla auto_home.org_dir, que es la que indica al sistema operativo cual es el directorio que compartirá con dicho usuario. Lo que hace el S.O. es buscar la información correspondiente al usuario dentro de la tabla, la cual contiene la información de una forma similar a:



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

lorenzo apolo:/export/home/lorenzo

En este caso, el login del usuario es "lorenzo", el servidor donde está alojado su directorio hogar es "apolo", y su directorio es "/export/home/lorenzo"⁹. Una vez identificada esta información lo que hace el S.O. es compartir el directorio por medio de NFS y automontarlo¹⁰ en el cliente debajo del directorio "/home". Dándole seguimiento al ejemplo anterior, el usuario lorenzo tendría su información disponible en el directorio "/home/lorenzo" del sistema de archivos del cliente donde accedió al sistema.

En caso de darse la no existencia de un renglón correspondiente al usuario dentro de la tabla `auto_home.org_dir`, lo que hace el sistema es enviarle de inmediato un mensaje de error al usuario, avisándole que no tiene directorio hogar; acto seguido, el sistema deposita al usuario dentro de la raíz del sistema de archivos (/).

cred.org_dir. Dado que en la sección 3.1.4. se habló de la tabla `cred.org_dir`, lo más importante por recordar, es que para que un usuario pueda acceder al sistema y realizar operaciones, necesita de credenciales de tipo DES y Locales, debido a los procesos de autenticación y autorización.

group.org_dir. Esta tabla es la encargada de almacenar la información concerniente a los grupos de trabajo que reconoce el servicio de NIS+. Cada usuario que es creado para acceder al sistema necesita de un grupo de trabajo, de lo contrario, tendría conflictos en la realización de cualquier tipo de actividad.

La tabla está compuesta por las siguientes columnas:

Columna	Descripción
Name	Es el nombre del grupo.
Passwd	Password del grupo (Casi nunca existe).
GID	Identificador numérico del grupo.
Members	Los identificadores (logins) de los miembros del grupo, separados por una coma.

mail_aliases.org_dir. Es la tabla que contiene los sinónimos de los identificadores de los usuarios para el correo electrónico. La información que almacena tiene una forma similar a:

⁹ Por convencionalismos la mayoría de los directorios hogar de los usuarios se encuentran bajo el directorio "/export/home".

¹⁰ Entiéndase por automontar a la acción de cargar información alojada en el disco duro dentro del sistema de archivos, para que ésta pueda ser reconocida de forma automática.



ADMINISTRACIÓN DE LOS USUARIOS DE NIS +

lorenzo lorenzo@apolo.acatlan.unam.mx

Lo que hace el sistema es que cuando se envía un correo al usuario lorenzo, el servidor ya sabe que la dirección de correo electrónico completa del usuario lorenzo es lorenzo@apolo.acatlan.unam.mx.

passwd.org_dir. El contenido y la importancia de esta tabla ya se ha mencionado con anterioridad (sección 3.2.2), sólo queda resaltar que la existencia de un usuario depende directamente de su respectivo renglón en la tabla *passwd.org_dir*; de no existir dicho renglón significaría el no reconocimiento del usuario dentro del sistema.

4. Desarrollo de herramientas auxiliares creadas con TCL/TK

“ Una colección de pensamientos debe ser una farmacia donde se encuentra remedio a todos los males “

Voltaire



4.1. INTRODUCCIÓN

4.1.1. Creación de nuevas herramientas administrativas

El poder utilizar Solstice como herramienta administrativa del sistema Solaris y en particular de los servicios de NIS+ ha sido de gran utilidad, ya que se puede comprobar que realmente facilita la administración del sistema; sin embargo, en base a la experiencia obtenida a través de la administración de la red SUN se ha podido comprobar que Solstice es una herramienta muy interesante y completa en algunos aspectos, pero en otros tiene carencias. Como se mencionó en la sección 3.1.5, Solstice carece de herramientas de manipulación masiva de usuarios, las cuales son las de mayor importancia dentro del uso de la red SUN de la ENEP Acatlán.

Un punto esencial en la decisión de desarrollar herramientas complementarias, fue la necesidad de “prestar el mejor servicio a la brevedad posible”, ya que realizar las operaciones con los recursos que se tenían, tomaba demasiado tiempo. Además conocer varios sistemas operativos ayudó a idear las herramientas que facilitarían el trabajo de administrador de red. El nuevo conjunto de herramientas está compuesto por: *userglb*, *userprt*, *userprc* y *userarch* las cuales serán descritas en secciones subsecuentes.

El hecho de que la manipulación de NIS+, y en particular de los usuarios, era totalmente desconocida por los administradores de la red y de que no se conocía persona alguna que manejara medianamente los conceptos de NIS+, originó que el desarrollo de las herramientas fuera una tarea complicada y a su vez innovadora. Otro factor que llamó la atención era el poder desarrollar herramientas gráficas que corrieran sobre la interfaz gráfica de los sistemas UNIX (X-windows), de la cual se hablará en la sección 4.1.3., además de tener la capacidad de ser portable hacia un navegador de internet, debido a las tendencias que tiene el ámbito computacional y a los objetivos planteados.

4.1.2. TCL/TK un lenguaje propicio para la creación de las nuevas herramientas

La elección de un lenguaje propicio para el desarrollo de las herramientas fue difícil, ya que existían varios que podían ser utilizados. Los lenguajes a los que más atención se les prestó fueron a Perl, C, Java y TCL/TK. Posteriormente en base a pruebas e investigaciones realizadas se concluyó que el lenguaje más propicio era TCL/TK, por las ventajas que se presentarán más adelante.

TCL/TK es un lenguaje de programación desarrollado en “C” por John Ousterhout en los laboratorios de la compañía SUN Microsystems. Este lenguaje está formado por dos elementos fundamentales “TCL” y “TK” los cuales se describen a continuación:



TCL (*Tool Command Language*). Como su nombre lo dice, es un lenguaje basado en línea de comandos que se asemeja mucho a la programación shell¹. El manejo y creación de programas creados con TCL es muy fácil y permite integrar programas desarrollados en otros lenguajes como C, C++, Java, shell script, etc. Básicamente está compuesto por un lenguaje de tipo script y su intérprete² (Véase la sección 4.2.1. para entender la estructura de desarrollo).

TK (*Tool Kit*). Es un sistema de manejo de ventanas para la creación de una interfaz gráfica amigable para el usuario. TK en conjunto con TCL permiten la creación de ventanas con objetivos y características específicas conocidas como widgets. El diseño de un programa, así como la creación de la interfaz para el usuario, resulta ser mucho más sencillo utilizando los widgets.

Existen muchas ventajas en la utilización de TCL/TK para el desarrollo de programas, dentro de los cuales se pueden mencionar:

- El desarrollo y la depuración de programas es más sencilla.
- Permite incluir programas de otros lenguajes.
- La interacción con el sistema operativo es fácil.
- Permite el desarrollo de programas que corran sobre sistemas X-windows.
- Los programas que se crean son portables hacia el browser, es decir, casi todas las aplicaciones desarrolladas en este lenguaje pueden ser ejecutadas sobre un navegador de internet.
- Existen versiones del lenguaje para diversas plataformas tales como Windows, Solaris, Irix, System VII, AIX, etc.
- El software es freeware, es decir, no es necesario pagar por su uso.
- Se pueden crear nuevos comandos o widgets por medio de lenguaje C y después llamarlos desde un programa en TCL/TK.

¹ La programación shell es aquella que está compuesta por comandos de línea del S.O. UNIX y cuya estructura es muy parecida a la del lenguaje "c".

² Un intérprete es un programa encargado de traducir código sin generar un archivo ejecutable.



Desarrollo de herramientas auxiliares creadas con TCL/TK

Desafortunadamente y al igual que todos los lenguajes, TCL/TK también tiene desventajas, las cuales deben ser consideradas antes de ocuparlo para el desarrollo de programas. A continuación se presentan las más relevantes:

- Depende en todo momento del intérprete, i.e. no se crea un programa ejecutable.
- La ejecución del programa resulta ser un poco más lenta que un programa de C, debido nuevamente al intérprete.
- La cantidad de widgets existente es limitada.
- Dado que es un lenguaje que todavía se está desarrollando, su código contiene errores (bugs).
- Aunque TCL/TK es un lenguaje de tipo visual, no se cuenta con herramientas gráficas confiables para la creación de los programas, como lo hace "Visual Basic"³.

En general, se considera que TCL/TK es un lenguaje no estructurado, ya que no hay una estructura propia para la codificación de los programas como en "Pascal". En TCL/TK no existe una cabecera de declaración de variables o procedimientos ya que pueden ser definidos en cualquier parte de la aplicación, ni tampoco marcas (instrucciones o caracteres especiales) que indiquen el cuerpo principal de la misma.

4.1.3. Los sistemas X-windows

Los sistemas X-windows, conocidos también como X, son una colección de programas que forman una interfaz para el usuario, orientada al uso del teclado, del ratón (mouse) y de una pantalla gráfica.

Un sistema de ventanas, como X, es una colección de programas que permiten utilizar y mostrar muchas aplicaciones en una sola pantalla. Para lograr ésto, divide la pantalla en varias más pequeñas llamadas ventanas, con lo que se pueden tener simultáneamente varias sesiones en diferentes hosts, conteniendo programas que muestren gráficos, lectura de correspondencia, despliegue de texto, etc., todo corriendo al mismo tiempo.

La mayoría de los sistemas de ventanas utilizan una abstracción llamada "metáfora de escritorio" (desktop). Esto es, la pantalla es tratada como si fuera la parte superior de un escritorio, y cada ventana de programa como si fuera un pedazo de papel. En un escritorio real, se pueden poner piezas de papel una al lado de otra, apilarlas unas sobre otras, moverlas

³ Visual Basic es un lenguaje creado por la compañía Microsoft cuya finalidad es crear programas que puedan ser ejecutados bajo el ambiente windows.



Desarrollo de herramientas auxiliares creadas con TCL/TK

alrededor, doblarlas, etc. De forma similar, un sistema de ventanas como X permite colocar ventanas en la pantalla, apilarlas, colocarlas contiguas, moverlas, etc.

El sistema de ventanas es llamado X-windows System, versión 11, abreviado como X11. Este software ha sido desarrollado por el MIT (Massachusetts Institute of Technology), y debido a su diseño independiente del hardware corre en una gran variedad de estaciones de trabajo (workstations) de diferentes proveedores.

El sistema de X11 es similar en filosofía a las interfaces para usuarios de Apple Macintosh, Microsoft Windows, OS/2 de IBM, etc. Si se está familiarizado con cualquiera de estos entornos de ventanas, entonces X no es completamente extraño. El sistema X-windows tiene los conceptos de ventanas, iconos y menús colgantes, justamente como el Microsoft Windows. Por lo cual se pueden mover ventanas, modificar su tamaño (resize), transformarlas en iconos, etc.

A continuación se definen algunos de los conceptos más generales de los sistemas X-windows.

Aplicación. Una aplicación es cada programa que se usa en el sistema X. Algunos ejemplos de aplicación (o programas de aplicación) pueden ser: los programas para leer correo, un programa para dibujar una figura, un editor de texto, etc.

Cursor. Es un objeto pequeño que aparece en la pantalla y usualmente toma la forma de una caja o de una línea de subrayado. El cursor indica donde aparecerá el primer carácter cuando se comience a escribir. En la pantalla de una terminal normal, generalmente hay un solo cursor; sin embargo, cuando se utiliza un sistema de ventanas, pueden existir tantos cursores como ventallas haya.

Display. En la terminología de los sistemas de ventanas se refiere a la pantalla de la estación. Cabe señalar que para algunas estaciones de trabajo es posible que exista más de una pantalla (por ejemplo una blanco y negro y otra a color).

Ícono. Un ícono es usualmente un pequeño dibujo simbólico, el cual representa una aplicación cuando ella no está en uso. Por ejemplo, un dibujo de una casilla de correos puede representar un programa de lectura de correo. Generalmente, el usuario puede asignar el ícono de su preferencia a cada aplicación, de tal forma que sea representativo.

Ratón o mouse. Un ratón es un dispositivo físico con botones que sirve para apuntar en la pantalla de la terminal. Moviendo el ratón y presionando los botones se puede indicar a las aplicaciones que realicen acciones como desplegar una lista de opciones, dibujar una línea, copiar texto desde un área a otra, etc. Los ratones de las estaciones de trabajo normalmente tienen tres botones, usualmente conocidos como izquierdo, del medio y derecho, pero algunas veces se numeran como uno, dos y tres de izquierda a derecha.



El puntero visto en la pantalla de la terminal es llamado "*Cursor del ratón*" el cual es muy similar al cursor regular como se definió previamente. Cuando el ratón es movido, el cursor del mismo se mueve dentro de la pantalla para indicar donde apunta. Sin embargo, a diferencia de los cursores regulares, el del ratón frecuentemente cambia de forma para proveer de información al usuario. Por ejemplo, puede parecer como una flecha cuando se señala algo o como un pincel cuando se está pintando una línea.

Ventanas. Una ventana es un área limitada en que una aplicación realiza la mayoría de sus funciones de entrada y salida. Una aplicación puede usar cualquier número de ventanas, aunque la mayoría sólo usa una o dos. Aunque algunos sistemas de ventanas permiten que las ventanas sean de cualquier forma, el sistema X11 requiere que todas las ventanas sean rectangulares.

Ventana root. Es la ventana utilizada por el administrador de ventanas para poner todas las otras ventanas de las aplicaciones. Usualmente es usada para representar el escritorio, y por lo tanto, ocupa la pantalla completa. Generalmente la ventana root es de fondo color gris o negro, de textura lisa, y las ventanas de las aplicaciones son de otro color, como blanco o gris claro; esto permite a la ventana root funcionar como fondo de las otras aplicaciones.

4.2. ESTRUCTURA DE DESARROLLO CON TCL/TK

4.2.1 Entendimiento del TCL

Como ya se comentó en la sección 4.1.2, TCL es un lenguaje de tipo script que depende de un intérprete y que se asemeja a la programación shell. En esta sección se describirán los puntos considerados como los más importantes para el desarrollo de programas en TCL, sin pretender darlo a conocer en su totalidad.

Por sencillez, a partir de este momento a un comando de TCL utilizado dentro de un programa se le llamará instrucción; y cuando se haga referencia a la palabra comando, se referirá a un comando del S.O.



1) Manipulación de variables⁴.

A diferencia de otros lenguajes de programación, dentro de TCL no es necesario reservar memoria para las variables, por lo que la creación se realiza de forma dinámica en tiempo de ejecución del programa.

Las variables dentro de TCL juegan un rol muy importante en el desarrollo de programas. Básicamente no existe la necesidad de declarar las variables como se acostumbra en "C o Pascal", todas las variables son de tipo carácter y no se tiene restricción en cuanto al uso de ellas.

Algunas características de las variables son:

- ◆ Los nombres de las variables pueden estar conformados hasta de 255 caracteres.
- ◆ No se deberán ocupar los siguientes caracteres en los nombres de las variables:
“.;!\$%&/()=?i[]{}*+”
- ◆ Se hace diferencia entre mayúsculas y minúsculas, de tal forma que las variables “UNO”, “Uno” y “uno” son diferentes.
- ◆ Para hacer referencia al contenido de la variable, se le debe anteponer el signo de “\$”.
- ◆ No se podrán ocupar palabras reservadas⁵ para la asignación de nombres.

La creación, destrucción y manejo de variables se hace por medio de instrucciones descritas en la tabla 4.1

Sintaxis	Descripción
<code>append Var1 Vall</code>	En caso de que la variable “Var1” no exista, la crea asignándole el valor descrito por “Vall”.
<code>incr Var1 Incremento</code>	Incrementa el valor de la variable “Var1” en la cantidad especificada por “Incremento”. Para la ejecución de esta instrucción, se requiere que la variable al igual que el incremento sean números enteros.
<code>set Var1 Vall</code>	Asigna el valor “Vall” en la variable “Var1”. Cuando en los argumentos solamente se pone el nombre de la variable (Var1), el intérprete despliega el contenido de la variable.
<code>unset Var1</code>	Destruye la variable especificada por “Var1”.

Tabla 4.1. Instrucciones utilizadas en la manipulación de variables.

⁴ Una variable es un contenedor de información que por lo regular es almacenado en memoria al momento de la ejecución de un programa.

⁵ Se considera como palabra reservada a toda aquella que el intérprete reconozca como instrucción.



Ejemplos:

<code>set Var 44</code>	<i>Asigna el valor 44 a la variable "Var"</i>
<code>set Var "Primer ejemplo"</code>	<i>Asigna la cadena de caracteres "Primer ejemplo" a la variable "Var"</i>
<code>set Var1 "Conca"</code>	<i>1) Asigna la cadena de caracteres "Conca" a la variable "Var1"</i>
<code>set Var2 "tenación"</code>	<i>2) Asigna la cadena de caracteres "tenación" a la variable "Var2"</i>
<code>set VarResultado \$Var1\$Var2</code>	<i>3) Asigna la concatenación de los valores de "Var1" y "Var2" en VarResultado.</i>
<code>set Var 10</code>	<i>1) Asigna el valor "10" a la variable "Var"</i>
<code>incr Var 2</code>	<i>2) Se incrementa el valor de "Var" en 2</i>
<code>unset VarTemporal</code>	<i>Destruye la variable "VarTemporal"</i>

II) Expresiones.

Las expresiones son aquellas que combinan valores (operandos) con operadores para crear nuevos valores. Por ejemplo, la expresión "4+2" contiene dos operandos "4" y "2" y un operador "+", que generan el valor 6 como resultado de la operación. Muchas de las instrucciones de TCL reciben argumentos que, en la mayoría de los casos son expresiones. La instrucción más simple y de más uso en la utilización de expresiones es "expr" la cual evalúa sus argumentos y arroja como resultado un valor de tipo string, ejemplo:

```
expr (18+14) * 16.2  
⇒ 518.4
```

En general los operadores ocupados para las expresiones en TCL son muy similares a los utilizados en "ANSI C". Por ejemplo, "14*12<167" es evaluado como 0, ya que el operado "*" tiene prioridad sobre "<". Los operadores aritméticos como +, -, *, / y % realizan las operaciones de suma, resta, multiplicación, división entera truncada y residuo entero respectivamente. En la tabla 4.2, se presenta un resumen de operadores más comunes así como el tipo de operandos y el resultado.

	Resultado	Tipos de Operandos	
-a	El negativo de "a".	Enteros, Reales	
!a	Negador Lógico: 1 si "a" es 0, o viceversa.	Enteros, Reales	
a*b	Multiplicación de "a" por "b".	Enteros, Reales	
a/b	Divide "a" entre "b".	Enteros, Reales	
a%b	Residuo de la división de "a" entre "b".	Enteros	
a+b	Suma de "a" más "b".	Enteros, Reales	
a-b	Resta de "a" el valor de "b".	Enteros, Reales	
a<b	1 si "a" es menor que "b", 0 en caso inverso.	Enteros	
a>b	1 si "a" es mayor que "b", 0 en caso inverso.	Enteros,	Reales,
		Caracter	
a>=b	1 si "a" es mayor o igual que "b", 0 en caso inverso.	Enteros,	Reales,
		Caracter	
a<=b	1 si "a" es menor o igual que "b", 0 en caso inverso.	Enteros,	Reales,
		Caracter	
a==b	1 si "a" es igual que "b", 0 en caso inverso.	Enteros,	Reales,
		Caracter	
a!=b	1 si "a" es diferente que "b", 0 en caso inverso.	Enteros,	Reales,
		Caracter	
a&& b	Operador lógico "AND".	Enteros, Reales	
a b	Operador lógico "OR".	Enteros,	Reales,
		Caracter	

Tabla 4.2 Descripción de expresiones más comunes en TCL.

Ejemplos:

```
set Var1 8
set Var2 5
expr $Var1-$Var2
⇒ 3
```

```
set Var1 "Variable de tipo texto"
set Var2 "Variable de tipo texto"
if { $Var1 == $Var2 } then {puts "Las variables son iguales"}
⇒ Las variables son iguales
```

En el ejemplo anterior se maneja la instrucción "puts", ésta sirve para mandar a pantalla la secuencia de caracteres indicada en el argumento de forma textual.



III) Listas.

Las listas son usadas en TCL para crear colecciones de cosas, semejantes a todos los usuarios que pertenecen a un grupo de trabajo, o a todo el tipo de archivos que contiene un directorio. Una lista es una colección ordenada de elementos donde cada uno tiene un valor de tipo caracter, como lo puede ser la representación de un número, el nombre de una persona, etc. Las listas son representadas como una cadena de tipo caracter con una estructura particular; esto es, se pueden almacenar listas en variables normales.

La forma más sencilla de una lista es aquella en la que sus elementos están separados por un espacio. Por ejemplo: "Universidad Nacional Autónoma de México" es una lista formada por 5 elementos. En general, una lista puede tener cualquier cantidad de elementos y cada elemento puede ser arbitrario (no requiere de características específicas). Para identificar los elementos de cada lista se numeran de izquierda a derecha empezando en cero y recibiendo el nombre de índices; en el ejemplo anterior, el índice 0 sería "Universidad", el 1 sería "Nacional" y así sucesivamente.

En listas un poco más complejas, se puede crear recursividad en el sentido de que una lista puede estar compuesta por otras listas; para realizar la identificación de cada lista, se ocupan los caracteres { }. Por ejemplo: "Elemento0 Elemento1 {SubElemento1 SubElemento2} Elemento3" es una lista compuesta por 4 elementos en el cual, el tercero es una sublista compuesta por dos elementos, que son "SubElemento1" y "SubElemento2".

Para el manejo de listas TCL provee las instrucciones descritas en la tabla 4.3.

Sintaxis	Descripción
concat Lista1 Lista2 ... ListaN	Concatena (une) las listas "Lista1", "Lista2", ..., "ListaN", formando una nueva lista.
join Lista Caracter	Reemplaza los espacios entre los elementos de la lista indicada en "Lista" por el caracter indicado en "Caracter".
lappend VarLista Valor1 Valor2 ... ValorN	Crea una nueva variable llamada "VarLista" de tipo lista en la cual sus elementos serán "Valor1", "Valor2", ..., "ValorN".
lindex Lista índice	Arroja el elemento de Lista indicado por "índice".
linsert Lista índice Valor1 Valor2 ... ValorN	Genera una nueva lista en la cual se insertan los valores "Valor1", "Valor2", ..., "ValorN" en el índice indicado.
llength Lista	Arroja el número de elementos de una lista.



lrange Lista Primero Ultimo	Arroja los elementos de Lista comprendidos en el rango de los índices "Primero" y "Ultimo" .
lreplace Lista Primero Ultimo Valor1 ... ValorN	Elimina los elementos de Lista comprendidos en el rango indicado por los índices "Primero" y "Ultimo" substituyéndolos por los valores indicados por "Valor1", ..., "ValorN".
lsearch Lista Elemento	Realiza la búsqueda de un elemento dentro de la lista especificada.
Lsort Lista	Regresa una lista ordenada a partir de la lista indicada.
split Caracteres Bandera	Busca el caracter indicado por "Bandera" dentro de la secuencia de caracteres del primer argumento y los reemplaza por espacios, generando una lista.

Tabla 4.3. Instrucciones utilizadas para la manipulación de listas.

Ejemplos:

Creación de las listas "VarLista1" y "VarLista2".

```
lappend VarLista1 Lst1Indice0 Lst1Indice1 Lst1Indice2 {Lst1Sub1 Lst1Sub2 Lst1Sub3}  
lappend VarLista2 Lst2Indice0 Lst2Indice1 Lst2Indice2 {Lst2Sub1 Lst2Sub2 Lst2Sub3}
```

Para los siguientes ejemplos se toman como referencia las listas creadas en el ejemplo anterior.

```
concat $VarLista1 $VarLista2  
⇒ Lst1Indice0 Lst1Indice1 Lst1Indice2 {Lst1Sub1 Lst1Sub2 Lst1Sub3} Lst2Indice0  
Lst2Indice1 Lst2Indice2 {Lst2Sub1 Lst2Sub2 Lst2Sub3}
```

```
lindex $VarLista1 1  
⇒ Lst1Indice1
```

```
linsert $VarLista1 2 {SubLista Insertada}  
⇒ Lst1Indice0 Lst1Indice1 {SubLista Insertada} Lst1Indice2 {Lst1Sub1 Lst1Sub2 Lst1Sub3}
```

```
lrange $VarLista1 1 3  
⇒ Lst1Indice1 Lst1Indice2 {Lst1Sub1 Lst1Sub2 Lst1Sub3}
```



```
join $VarLista1 :  
⇒ Lst1Indice0:Lst1Indice1:Lst1Indice2:Lst1Sub1 Lst1Sub2 Lst1Sub3
```

```
set Variable "CREACION,DE,UNA,LISTA"  
split $Variable ,  
⇒ CREACION DE UNA LISTA
```

IV) Controles de flujo if, while,for, foreach.

Los controles de flujo son instrucciones que le indican al programa cuantas veces ejecutará un bloque de instrucciones. En general las instrucciones de control de flujo son las mismas que se manejan en lenguaje "C"; es decir "if", "while", "for", "foreach".

- a) **La instrucción if.** La instrucción if evalúa una expresión, y en base al resultado toma valores de 0 (falso) o 1 (verdadero) ejecutando la secuencia de instrucciones indicada para cada caso. Cuando la expresión es verdadera se ejecuta el bloque de instrucciones indicado en el segundo argumento y que está delimitado por los {}. En caso contrario, cuando la expresión es falsa, se ejecuta el bloque de instrucciones indicado en el tercer argumento.

Ejemplo:

```
Set x 1  
if { $x < 0 } {  
  puts "El valor de x es menor que cero"  
} { puts "La expresión fue falsa, y por tanto el valor de x es mayor o igual a cero"  
}  
⇒ La expresión fue falsa, y por tanto el valor de x es mayor o igual a cero
```

Con la estructura de control if, se pueden hacer recuestionamientos conocidos como "elseif" y se pueden poner tantos como el programador requiera.

Ejemplo:

```
set opción 0  
if { $x == 1 } {  
  puts "La opción elegida fue 1"  
} elseif { $x == 2 } {  
  puts "La opción elegida fue 2"  
} elseif { $x == 3 } {  
  puts "La opción elegida fue 3"
```



```
} else {  
  puts "No escogió ninguna opción"  
}
```

⇒ No escogió ninguna opción

b) Las instrucciones *while*, *for* y *foreach*. En general la finalidad de las tres instrucciones es ejecutar un número determinado de veces el bloque de instrucciones indicado; lo que varía es el criterio para determinar dicho número.

- ♦ *while*. En su primer argumento se pone una expresión a evaluar y mientras se cumpla dicha expresión, el bloque de instrucciones comprendido en el *while* seguirá ejecutándose.
- ♦ *for*. Es muy similar al *while*, con la diferencia de que se indica por medio de un número entero las veces que se repetirá la secuencia de instrucciones. El primer argumento es la declaración de la variable que lleva el conteo, el segundo argumento es el indicador de término del ciclo, el tercero es el incremento que se le hace a la variable cada vez que ha terminado de ejecutar el bloque de instrucciones y el último argumento es el bloque de instrucciones.
- ♦ *foreach*. Toma como referencia el número de elementos que contiene una lista para determinar las veces que se ejecutará el bloque de instrucciones. Su primer argumento es el nombre de la variable que contendrá el valor del elemento correspondiente, el segundo elemento es la lista que se tomará como referencia para la ejecución de la instrucción *foreach* y el último es el bloque de instrucciones.

Ejemplos:

En los 3 ejemplos subsecuentes se hace el copiado de los elementos de la lista "a" a la lista "b" ocupando las instrucciones *while*, *for* y *foreach* respectivamente.

```
set a "Indice0 Indice1 Indice2"  
set b ""  
set i [expr [llength $a ] -1 ]  
while { $i >= 0 } {  
  lappend b [ lindex $a $i ]  
}  
puts $b
```

⇒ Indice0 Indice1 Indice2



Copiado de los elementos de la lista "a" a la lista "b" ocupando la instrucción "for".

```
set a "Indice0 Indice1 Indice2"
set b ""
for { set i [ expr [ llength $a ] -1 ] } { $i >= 0 } { incr i -1 } {
  lappend b [ lindex $a $i ]
}
```

Copiado de los elementos de la lista "a" a la lista "b" ocupando la instrucción "foreach".

```
set a "Indice0 Indice1 Indice2"
set b ""
foreach i $a {
  set b [ linsert $b 0 $i ]
}
```

Como se puede observar en los ejemplos anteriores la instrucción que necesitó menos codificación fue "foreach", sin embargo esto no significa que sea la mejor o la más sencilla, todo depende de las necesidades y tareas a realizar.

V) Procedimientos.

Un procedimiento dentro de TCL es una nueva instrucción que es implementada por medio de un script dentro del mismo programa, de forma muy parecida a como lo maneja lenguaje C. La ventaja de ocupar procedimientos, al igual que en otros lenguajes, es que facilita la programación además de ahorrar código de programa, debido a la reutilización que pudiera tener el procedimiento.

El inicio de un procedimiento siempre es la instrucción *proc* como se muestra en el siguiente ejemplo:

```
proc suma { a b } { expr $a +$b }
```

El primer argumento de la instrucción es el nombre del procedimiento a ser creado, que en el ejemplo anterior es "suma". El segundo argumento es la lista de los parámetros a ser ocupados dentro del procedimiento ("a" y "b" en el ejemplo). El tercer argumento es el script que forma el cuerpo del procedimiento. Con la declaración de este procedimiento se dice que se ha creado una nueva instrucción la cual requiere de dos argumentos para su ejecución. Una vez que se ha declarado el procedimiento, se puede referenciar dentro del programa simplemente invocando su nombre y argumentos.



```
suma 3 4  
⇒ 7  
suma 3 -1  
⇒ 2
```

El resultado que regresa el procedimiento es la última línea que se encuentre en el mismo, a menos que se escriba la instrucción “return” que puede estar seguida de algún valor. Sin importar en que parte del programa se encuentre, lo que hace la instrucción return es indicar que se salga del procedimiento; en el caso de que el return esté seguido de algún valor, éste será tomado como el resultado de la ejecución del procedimiento. Otra de las ventajas del manejo de procedimientos es que permite el manejo de recursividad, es decir, desde el mismo procedimiento se permite hacer el llamado a sí mismo.

Ejemplo: Procedimiento que saca el factorial⁶ de un número entero mayor o igual a cero.

```
proc factorial x {  
    if { $x < 0 } {  
        return “No esta definido el factorial de un número negativo”  
    }  
    if { $x == 1 || $x == 0 } {  
        return 1  
    }  
    expr $x * [ factorial [ expr $x -1 ] ]  
}
```

```
factorial 4  
⇒ 24  
factorial 0  
⇒ 1
```

En TCL se manejan variables de tipo global y variables de tipo local; la diferencia básica entre ambas es que las segundas solamente existen mientras se ejecuta un procedimiento y las primeras existen en cualquier parte del programa que no sea un procedimiento. En ocasiones se requiere que una variable global exista dentro de un procedimiento, ésto se hace por medio de la instrucción “global” tal cual se muestra a continuación.

Ejemplo:

⁶ Se define el factorial como una función matemática de la siguiente forma: $n! = n * (n-1)!$, para todo n entero positivo; Cabe aclarar que el factorial de 0 es 1.



```
proc prueba {} {  
  global x y  
  puts "El valor de la variable X es $x , y el valor de la variable Y es $y" }  
}
```

```
set x 322  
set y 111  
prueba  
⇒ El valor de la variable X es 322 , y el valor de la variable Y es 111
```

VI) Ejecución de programas del S.O.

Una ventaja más de TCL es que ofrece la posibilidad de poder ejecutar programas alternos, pudiendo ser éstos comandos del S.O. u otros programas. Para realizar esta tarea se ocupa la instrucción "exec".

A continuación se presenta un ejemplo en el que se hace el llamado al comando "rm" del S.O. el cual realiza la tarea de borrar el archivo indicado.

```
exec rm main.o
```

El primer argumento de la instrucción exec es el programa que se desea ejecutar y los argumentos adicionales irán en virtud del programa indicado. En el ejemplo anterior el programa a ejecutar es un comando del S.O. (rm) y el segundo es el que necesita el comando rm.

Aunque aparentemente el manejo de la instrucción exec es fácil, puede complicarse cuando se necesita ejecutar programas externos que en la declaración de sus argumentos, requieren de caracteres especiales que causan conflictos dentro de TCL.

Supóngase que dentro de un programa en TCL se requiere borrar todos los archivos con extensión tmp; en base al ejemplo anterior y ocupando el comodín⁷ "*" la sintaxis debiera ser la siguiente:

```
exec rm *.tmp  
⇒ couldn't execute "*.tmp": no such file or directory
```

Se observa que el intérprete no reconoce el comodín * y éste es tomado como un carácter más; para solucionar este problema TCL proporciona la instrucción "eval", la cual le

⁷ Un comodín es un caracter especial que substituye una cadena de caracteres; dentro del S.O. UNIX los más ocupados son *, el cual substituye cualquier cantidad de carcateres y ?, que substituye sólo un caracter.



indica al intérprete que concatene todos sus argumentos con espacios para después evaluarlos. El problema del ejemplo se resolvería de la siguiente forma:

```
eval exec rm [ glob *.tmp]
```

En otras ocasiones, la ejecución de un programa requiere del agrupamiento de caracteres por medio de comillas dobles ("), sin embargo éstas tienen un significado especial dentro de TCL, por lo que se hace necesario indicarle al intérprete que cancele el significado especial del carácter mediante el uso de la \.

Ejemplo:

```
puts "\Las ganancias del día de hoy fueron \$13.00\""  
⇒ "Las ganancias del día de hoy fueron $13.00"
```

Además de cancelar el significado especial de las comillas, también fue necesario cancelar el de \$ ya que, como se recordará, éste sirve para referenciar el contenido de una variable, de no haber cancelado el significado de \$, el intérprete hubiera marcado un error que diría " can't read 13: no such variable".

VII) Manejo de archivos.

Para realizar operaciones con archivos tales como lectura y escritura de datos, TCL proporciona métodos específicos; sin embargo, dadas las facilidades que para tal fin proporciona el S.O., se optó por este último. La finalidad de este inciso no es dar a conocer la forma de manipulación de los archivos, sino que se pretenderá explicar cómo se manejaron los archivos para el desarrollo de las aplicaciones del presente trabajo.

a) **Lectura de datos.** La lectura de información de archivos se hace mediante el comando "cat", direccionando su salida hacia una variable que es la contenedora de la información. La idea es que la variable sea de tipo lista, en la que cada renglón del archivo sea expresado como un elemento en la variable.

Ejemplo:

Contenido del archivo prueba.

```
Renglon1  
Renglon2  
Renglon3
```

Vaciado del contenido del archivo "prueba" en la variable "Datos" y visualización del índice l

```
set Datos [ eval exec cat Nombre_de_Archivo ]
```



```
lindex $Datos 1  
⇒ Renglon2
```

Con la utilización del comando `split` y `join` se pueden manipular archivos con mayor complejidad (aunque siguen siendo de tipo ASCII) que simulan la utilización de campos y registros de una base de datos plana. Tomando como referencia el archivo `/etc/passwd` el cual contiene información de los usuarios, se puede decir que cada renglón representa un registro, y que los campos están delimitados por el carácter “:”.

Ejemplo del contenido de un archivo `/etc/passwd`.

```
lorenzo:x:1002:14:Cuenta de acceso a vulcano (Lorenzo):/export/home/lorenzo:/bin/csh  
irc:x:1500:15:Cuenta de IRC:/home/irc:/bin/csh  
alberto:x:1005:16:Luis Alberto Gonzales Escobar:/export/home/alberto:/bin/csh  
carlos:x:1003:16:Carlos Alberto Rangel Rojas:/export/home/carlos:/bin/csh  
ivan:x:1004:16:Ivan Rangel Rojas:/export/home/ivan:/bin/csh  
armando:x:1006:14:NT Master, DSI:/export/home/armando:/bin/ksh  
rubi:x:1007:14:Rubi Cruz Villegas Torres:/home/rubi:/bin/csh
```

En el ejemplo anterior, se observan 7 renglones (registros) y 7 campos. Para crear la lista equivalente a esta estructura, primero se vacía la información en una variable, después se rompe con los saltos de nueva línea remplazándolos por un carácter en blanco (espacio), y finalmente se almacenan en otra variable, obteniéndose el resultado deseado.

Ejemplo:

```
set VarTmp [ eval exec cat Nombre_de_Archivo ]  
set Datos [ split $VarTmp “\n” ]
```

- b) **Escritura de datos.** El proceso de escritura de datos, al igual que el de lectura, se basa en la utilización de comandos y direccionamientos de salida del S.O. Siguiendo con el ejemplo del archivo `/etc/passwd`, que ya tiene su equivalente en lista en una variable llamada `Datos`, ahora se vaciará en un archivo la información contenida en esta variable.

Lo primero por hacer es cambiar la forma de lista que tiene la información y pasarla a la forma de archivo que conocemos, para lograrlo se ocupa la instrucción “`join`”, que como ya se mencionó lo que hace es remplazar el caracter de separación de elementos de lista (espacio) por un carácter indicado; como siguiente paso se envía la información hacia el archivo, lo cual se hace por medio del comando “`echo`”, direccionando su salida hacia el archivo.



Ejemplo:

```
set LstTmp [ join $Datos "\n" ]
foreach i $LstTmp {
eval exec "echo $i >> Nombre_de_Archivo"
}
```

En este ejemplo se maneja el símbolo ">>" cuya función es indicarle al S.O. que el resultado de un comando, en lugar de desplegarlo en pantalla (salida estándar) lo envíe como contenido al final de un archivo indicado, esto es: la información arrojada de la ejecución de un comando es anexada al final de un archivo, sin afectar las líneas anteriores del mismo archivo.

VIII) Errores.

Cuando ocurre un error en la ejecución de un programa de TCL, lo que el intérprete hace es salirse del bloque de instrucciones que está ejecutando, es decir, si la instrucción en la que se ocasionó el error está fuera de un procedimiento se sale totalmente; si el error fue ocasionado mientras se ejecutaba un procedimiento se sale del bloque de instrucciones del procedimiento y continúa la ejecución del programa.

Dependiendo del diseño y la lógica que se maneje en el programa, no siempre es óptimo abortarlo cuando se genera un error. TCL proporciona métodos y variables de captura de errores que ayudan al programador a controlar la secuencia y salida del programa.

Las variables "errorInfo" y "errorCode" son variables de tipo global que toman ciertos valores cuando se genera un error; la primera contiene información referente a la causa que originó el problema, y la segunda es un identificador o número que se le asigna al error.

Para ignorar los errores generados en la ejecución de un programa de TCL se utiliza la instrucción "catch", como lo muestra el ejemplo siguiente:

```
catch {unset x} msg
⇒ 1
set msg
⇒ can't unset "x": no such variable
```

El primer argumento (unset x) es el comando que se quiere ejecutar y del cual se omitirá el error en caso de que éste se origine; el segundo (msg) es el nombre de una variable que contendrá la información del error generado. Una vez que el comando catch ha sido ejecutado, se obtiene como resultado un valor numérico que puede ser 0 en caso de que no hubiese error en la ejecución de la instrucción o 1 en caso de existir.



Existe también la posibilidad de codificar errores; véase el ejemplo siguiente:

```
set x -2
if { ( $x < 0 ) || ( $x > 100 ) } {
    error "x está fuera de rango" "Esto es un error" 16
}

⇒ x está fuera de rango

puts $errorInfo
⇒ Esto es un error
⇒ invoked from within
⇒ "if { ( $x < 0 ) || ( $x > 100 ) } {
⇒     error "x está fuera de rango" "Esto es un error" 16
⇒ }"

puts $errorCode
⇒ 16
```

4.2.2 El TK

TK resulta ser el complemento ideal para crear la interfaz gráfica de los programas creados en TCL. TK, al igual que la mayoría de los lenguajes de tipo visual como "Delphi", se basa en la programación eventual⁸ y en el manejo de objetos.

Al igual que en la sección 4.2.1, en ésta no se pretende explicar en su totalidad el funcionamiento de TK, sino dar un panorama general con la descripción de los puntos más relevantes para este trabajo.

I) Los TK widgets.

Los widgets son la estructura más elemental que existe dentro de TK y dependiendo de las necesidades del programa serán los widgets que se manejen. Existen básicamente 13 widgets que son:

⁸ La programación eventual es aquella en la que el flujo del programa es controlado por los eventos o sucesos que ocurran en la ejecución del mismo. Los eventos son acciones o situaciones reconocidas por algún objeto o widget. Ejemplo de ello es el click del botón del mouse sobre la superficie de un widget o el simple paso del puntero del mouse sobre éste.



- ◆ **Frame.** Dentro de TK, el frame es el widget más sencillo además de ser el que menos propiedades tiene. Su apariencia es solamente la de una ventana cuya finalidad es la de contener otros widgets.
- ◆ **Label.** Sirve para mostrar texto o un bitmap. Al igual que el frame no responde al mouse o al teclado, simplemente funciona como elemento decorativo en la ejecución del programa.
- ◆ **Button.** Resulta ser muy parecido al Label, con la diferencia de que el button responde al teclado o al mouse. En el caso del evento click lo que hace es ejecutar una secuencia de instrucciones indicada en el programa.
- ◆ **Checkbutton.** Permite hacer una elección sencilla en la que se tienen solamente los valores de cierto y falso. La elección es almacenada en una variable cuyo valor depende de si el checkbutton está o no seleccionado.
- ◆ **Radiobutton.** Permite seleccionar una opción de entre varias. Normalmente cuando es presentado, un radiobutton se encuentra acompañado por otros tantos de los cuales solamente se puede escoger uno. Al igual que los checkbutton se tiene designada una variable para cada radiobutton y dependiendo de la opción que se escoja será la variable que tenga asignado el valor “verdadero”.
- ◆ **Menubutton.** Sirve para manejar menús desplegables en los cuales se pueden incorporar submenús conformados por opciones de radiobutton, checkbutton, comandos o separadores.
- ◆ **Listbox.** El listbox es un widget que permite seleccionar una o más posibilidades de un rango de alternativas. La ventaja y principal diferencia que presenta con respecto a los checkbutton es que las opciones pueden ser creadas y destruidas de forma fácil, aún en tiempo de ejecución.
- ◆ **Entry.** Un entry es una caja que permite al usuario introducir y editar una línea de texto, la cual será almacenada en la variable que se le indique.
- ◆ **Scrollbar.** El scrollbar es un widget encargado de controlar lo que será desplegado en pantalla por otros widgets. Por si mismo el scrollbar no tiene mucha utilidad, pero una vez que ha sido vinculado con otro widget como el listbox su utilidad se incrementa considerablemente.
- ◆ **Text.** Este es muy similar a un entry, con la diferencia de que en el text se puede introducir más de una línea.



- ♦ **Canvas.** Es un contenedor de superficie de dibujo en el cual se pueden presentar cualquier cantidad de rectángulos, elipses, arcos, líneas, polígonos y texto.
- ♦ **Scale.** Es un widget que despliega un valor numérico y que permite al usuario editarlo. Su apariencia es como la de una escala lineal con una pequeña barra deslizable, además de tener opción de etiquetas.
- ♦ **Message.** El widget message muestra varias líneas de texto, de forma parecida a como lo hace el text, con las desventajas de que no permite editar el texto y no se le puede vincular con un scrollbar, entre otras.

Para dar una idea de la apariencia de los widgets a continuación se presenta la figura 4.1.

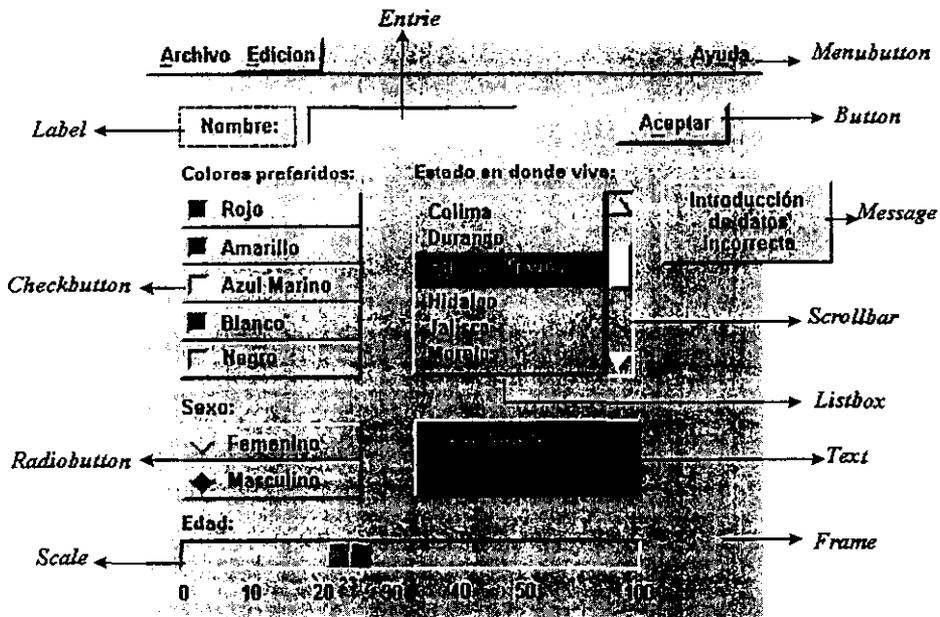


Fig. 4.1. Apariencia de los widgets de TK.

La forma de crear los widgets es prácticamente la misma para todos y se hace por medio de instrucciones que proporciona TK. Dichas instrucciones son precisamente el nombre del widget que se quiera crear; por ejemplo:

En el caso de un button sería.



```
button Nombre_del_boton Opciones
```

En el caso de un frame sería

```
frame Nombre_del_frame Opciones
```

Y así sucesivamente. Por otra parte, para destruir los widget basta con utilizar la instrucción "destroy", indicando como argumento el nombre del widget que se desea eliminar. Ejemplo:

```
destroy .label1
```

La organización de los widgets es jerárquica, de tal forma que un widget puede contener otros widgets. Esta forma de organización es conocida como el "árbol de los widgets", en donde la base es la ventana principal (root) identificada por ".". La figura 4.2. muestra un ejemplo de la estructura de árbol de una ventana de TK.

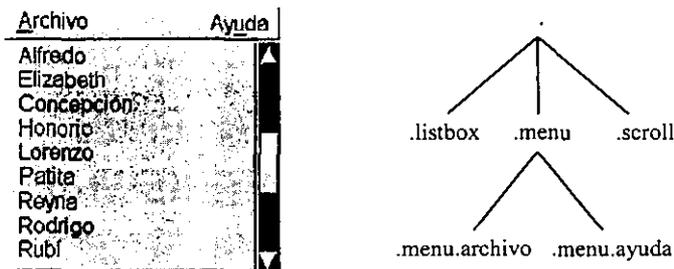


Fig. 4.2. Ejemplo gráfico de la estructura de árbol de una ventana de TK.

Como se puede observar en la figura 4.2., los niveles del árbol se van conformando dependiendo de la procedencia que tengan, es decir, los que provienen de la ventana principal, se encuentran en el primer nivel y su descendencia conforma el segundo nivel. Cuando se crea un widget y se quiere que su procedencia sea otro widget, lo que se hace es, al momento de crearlo, ponerle como nombre el del widget "padre" anexándole un punto y el nombre del nuevo widget.

Ejemplo: Supóngase que se crea un frame que contendrá un boton, la forma de hacerlo es:

```
frame .frame1
button .frame1.boton1
```



```
pack .frame1 .frame1.boton1
```

Cabe mencionar que no basta con crear los widgets para que estos sean vistos en pantalla; sino que es necesario indicarle al intérprete que permita su visualización, y esto se hace por medio de la instrucción "pack" tal cual se manejó en el ejemplo anterior.

II) Opciones de configuración.

Las opciones de configuración son aquellas que indican las características de los widgets, como lo son color, tamaño, tipo de letra, etc. y que son identificadas por que les antecede el carácter "-". Existen varias formas de configurar los widget, pero las principales son las 3 que a continuación se mencionan.

a) Configuración directa al momento de crear el widget; por ejemplo:

```
button .help -text "Ayuda" -foreground red
```

La instrucción anterior lo que hace es crear un botón con las opciones "-text" y "-foreground" que indican el texto y su color respectivamente. En general, como ya se vio en el apartado a) de esta sección, la creación de todos los widgets tiene la misma forma, lo único que varía son las opciones de configuración que se le pueden asignar a cada widget.

b) Por medio del archivo .Xdefaults. Cuando se ocupa un sistema de tipo X-windows se crea un archivo llamado .Xdefaults dentro del Home directory de cada usuario; dentro de éste se pueden manejar las opciones de configuración de los widgets.

c) Por medio de "configure". La instrucción configure permite que una vez creado cualquier widget se puedan modificar sus opciones de configuración. Ejemplo:

```
button .salir  
.help configure -text "Salir del Programa" -command exit
```

Todos los widgets soportan la instrucción configure, la única consideración importante para su uso es limitarse a las opciones de configuración del propio widget.

Los tipos de letra que pueden ser utilizados dependen en su totalidad del sistema X-windows y las librerías que se esté manejando. Los colores pueden manejarse en una gama de combinaciones de rojo, verde y azul expresados en numeración hexadecimal; en el caso de ser colores primarios, éstos también pueden ser identificados por su nombre común escrito en inglés. Ejemplos:

```
label .prueba1 -text "Color del texto azul" -foreground blue
```



```
label .prueba2 -text "Color de fondo, lila" -background "#A000F0"
```

La forma de configurar los tamaños de los widgets brinda una gran ventaja, ya que puede ser manejada en centímetros (c), pulgadas (i) o milímetros (m). Por ejemplo la instrucción para crear un button con un espacio adicional de 2 centímetros en el eje "x" y 2 pulgadas en el eje "y" sería:

```
button .boton -text nada -padx 2c -pady 2i
```

III) Manejadores de geometría.

Los manejadores de geometría son entidades que determinan la ubicación y dimensiones de los widgets. A diferencia de otros lenguajes de programación, la estructura de presentación de los widgets en una ventana de TK depende casi por completo de los manejadores de geometría y no de alguna propiedad de los widgets como lo hace Visual Basic.

Algunos manejadores de geometría asignan el espacio requerido para la ventana principal que contendrá los widgets hijos conocidos también como "esclavos"; sin embargo existen aplicaciones que en tiempo de ejecución pueden ser redimensionadas y que no pueden ser controladas del todo por el manejador de geometría.

A continuación se presentan los dos manejadores de geometría más comunes dentro de TK.

a) **Place.** El place parte de la existencia de un esclavo para después posicionarlo en la parte de la ventana principal indicada por medio de la instrucción "place". Ejemplo:

```
place .etiqueta -x 0 -y 0
```

Con la utilización de la instrucción place se puede indicar la posición en "x" y en "y" de un widget indicado, tal cual lo muestra la figura 4.3.

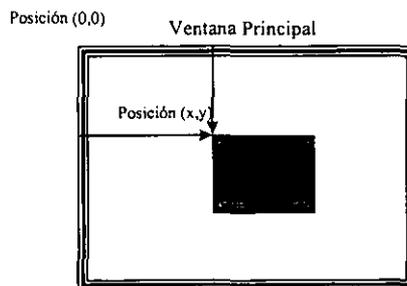


Fig. 4.3. Manejo del posicionamiento de los widgets por medio del *place*.

De igual forma se pueden manejar el ancho y alto del widget. Ejemplo:

Visualización en pantalla de un widget con 2 centímetros de largo y una pulgada de ancho.

```
place .label -x 10 -y 20 -width 2c -height 1i
```

- b) **Packer**. El packer es un manejador de geometría un poco más complejo, pero a cambio de ello es más poderoso. El packer, a diferencia del placer, tiene la capacidad de agrupar conjuntos de widgets y posicionarlos en un renglón; además de poder alinear widgets de forma central o a los lados de la ventana (esta tarea es muy similar a la que realiza Word cuando de alinea un párrafo) sin tener que estar calculando las posiciones en “x” y “y”.

Ejemplo: Despliegue de dos botones sobre un mismo renglón alineándolos a la izquierda.

```
button .ok -text "Continuar"
button .cancel -text "Cancelar"
pack .ok .cancel -side left
```

El proceso más conveniente para ocupar el packer es el siguiente: Se agrupan los widgets en un contenedor que puede ser un frame y se despliega dentro de la ventana principal; pudiéndose así ocupar el espacio sobrante por otro frame. Véase la fig. 4.4.

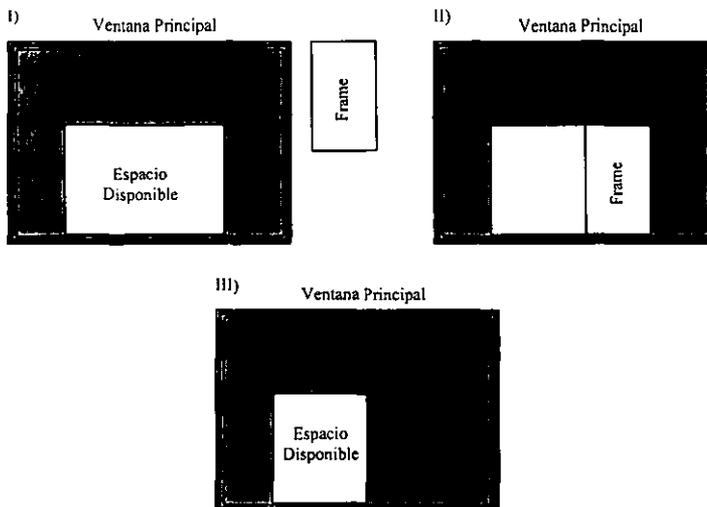


Fig. 4.4. Proceso de asignación de espacios a los frames por medio del packer.



IV) Eventos.

Como ya se mencionó, los eventos son sucesos que ocurren en tiempo de ejecución del programa. En general los eventos que maneja TK son precisamente los que soporta el sistema X-windows; ejemplo de ellos son los que se muestran en la tabla 4.4.

Evento	Descripción
<Button-Numero>	Espera el click de un botón del mouse, tomando como referencia que el botón de la izquierda es el Número 1.
<Control-caracter>	Espera la combinación de la tecla CONTROL y el caracter indicado.
<Double>	Espera el doble click de un botón del mouse.
<Destroy>	Espera que el widget sea destruido.
<Enter>	Espera la tecla ENTER.
<FocusIn>	Espera que el Focus ⁹ sea recibido por el widget.
<FocusOut>	Espera la salida del Focus.
<Key>	Espera la tecla indicada.

Tabla 4.4. Eventos de TCL/TK.

Los eventos de cada widget no son creados de forma automática, para hacerlo se tienen que programar. Para crear un nuevo evento es necesario utilizar la instrucción "bind" tal cual lo muestran los ejemplos siguientes:

```
bind .etiqueta <Button-1> {puts "Se esta oprimiendo el boton izquierdo del mouse "}  
bind .texto <FocusIn> {puts "Introduce tu nombre en la caja de texto"}  
bind .entry1 <Key-ñ> {puts "Este es un carácter no válido"}
```

4.3. ANÁLISIS Y DISEÑO DE LAS HERRAMIENTAS

4.3.1. Objetivos

General. Desarrollar herramientas de fácil uso con una interfaz gráfica soportable en el browser que facilen parte de las tareas de un administrador, optimizando los tiempos de respuesta.

Las herramientas que se desarrollaron son:

⁹ El Focus es el indicador del widget que se encuentra activo. Normalmente el focus es recibido cuando se le da un click al widget o cuando por medio del tabulador se llega a él. El focus también puede ser asignado a un widget por medio de instrucciones.



- a) userglb
- b) userrpt
- c) userprc
- d) userarch

Los *objetivos particulares* se enuncian por cada herramienta.

userglb

- Crear usuarios de forma masiva en base a una lista generada por la misma herramienta.
- Eliminar usuarios del sistema de forma masiva tomando como base una lista propuesta por la herramienta.

userrpt

Generar reportes de los usuarios existentes en el sistema en base a los siguientes criterios:

- Grupo al que pertenezcan.
- Número identificador del usuario.
- Orden alfabético.

userprc

Monitorear el S.O. por medio de la visualización de los procesos de los usuarios, permitiendo eliminar aquellos que provoquen inestabilidad en el sistema.

userarch

- Presentar los archivos de los usuarios que sean considerados como basura o información innecesaria dentro del sistema y que permita la eliminación de los mismos.
- Permitir el copiado de archivos indispensables para los usuarios a sus cuentas, realizando los cambios de seguridad pertinentes.

4.3.2. *Calendarización de actividades*

Una de las etapas más difíciles de un trabajo de investigación es la calendarización de actividades, ya que de no tener la debida precaución se puede caer en errores que repercutirían en el desarrollo del trabajo.



Desarrollo de herramientas auxiliares creadas con TCL/TK

Es importante mencionar que cuando se inició el trabajo de desarrollo de las herramientas, se estipuló un calendario diferente al presentado en el diagrama de Gantt de la fig. 4.5. Esto es debido a que la etapa de investigación, en la parte de NIS+ se llevó más tiempo del previsto, ocasionado por las escasas fuentes de información referentes al tema.

Otro factor que influyó en el cambio de fechas fue que además del desarrollo del proyecto se tenían que cumplir con otras actividades laborales que no pueden ser previstas y que tienen que ser atendidas de forma inmediata, como el mantenimiento del servidor Sun así como de los clientes del mismo.

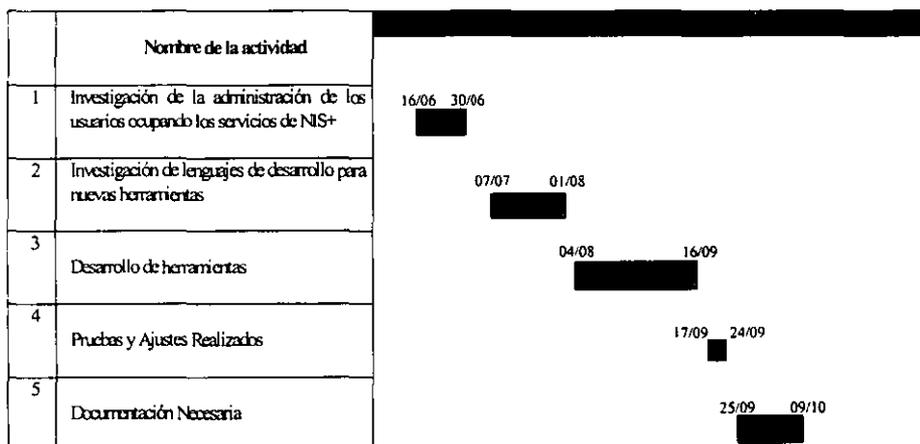


Fig. 4.5. Calendarización de actividades para el desarrollo de las herramientas por medio de un diagrama de Gantt.

4.3.3. Requerimientos

Como ya se ha mencionado, el desarrollo de las herramientas se basó en la utilización de TCL/TK en versiones 7.6 y 4.2 respectivamente. Existen versiones más actuales de estos paquetes, pero al realizar pruebas de confiabilidad se observó que tenían más errores en su código que las versiones utilizadas. Debido a las diferencias existentes entre las versiones es estrictamente necesario ocupar la 7.6 y 4.2 para la ejecución de las aplicaciones.

Así mismo es necesario el S.O. Solaris 2.5 o superior con soporte para sistemas X-windows. También se requiere de Solstice AdminSuite 2.2 o superior aunque no es indispensable.



En cuanto a hardware se requiere al menos de las características necesarias para el S.O. (mencionadas en el capítulo 2) agregando la necesidad de 10 Mb. libres de disco duro.

4.3.4. Análisis de factibilidad

Desde un principio las herramientas fueron planeadas de tal forma en que se ocuparan los recursos con que cuenta el D.S.C. y en particular la Red Sun, por lo que no se requirió de un estudio de factibilidad muy complejo; básicamente se partió de las necesidades existentes y de ahí crear la forma de solventar dichas necesidades por medio de los recursos humanos y materiales con que se contaba. La aceptación que tuvieran las herramientas iría en función a los tiempos de respuesta que se obtuvieran de los procesos que realizaran las herramientas y la portabilidad que pudieran tener hacia un navegador de internet.

4.3.5. Consideraciones importantes para el desarrollo de las herramientas

Para el desarrollo de las herramientas fue indispensable la utilización de comandos tanto del S.O., como de algunos otros proporcionados por Solstice AdminSuite. La importancia de ambos es descrita a continuación.

1) Los comandos de UNIX.

El sistema operativo UNIX es uno de los más completos en cuanto a comandos se refiere, ya que proporciona una lista muy variada para realizar tareas indispensables dentro del sistema. En la sección 2.3.1 se mencionaba como una característica de UNIX que varios programas sencillos resolvían situaciones complejas. Ahora bien, para darle utilidad a esta característica, dentro de las herramientas creadas se ocupan conjuntos de comandos para resolver tareas que el programarlas en TCL hubiera resultado más complicado. Un claro ejemplo se presenta en la herramienta "userprc" en la que es necesario visualizar los procesos de los usuarios de forma individual; para lograrlo se hizo uso de los comandos presentados en la tabla 4.5.

Comando	Descripción
who	Despliega los usuarios que se encuentran conectados al sistema.
sort	Ordena de forma ascendente o descendente una lista de nombres.
uniq	Toma una lista ordenada de nombres y elimina los que se encuentren repetidos.
ps	Visualiza los procesos del sistema.

Tabla 4.5. Comandos de UNIX utilizados en la herramienta userprc.

La combinación de comandos de UNIX con instrucciones de TCL permitieron realizar tareas de forma más fácil. Aunque por medio de TCL es muy sencillo manejar listas, no existe instrucción alguna que permita eliminar elementos repetidos, y aunque se puede hacer un



procedimiento que realice dicha tarea, esto no es óptimo ya que aumentarían el código del programa de forma considerable.

Para mayores referencias sobre comandos de UNIX, en el apéndice “A” se presenta una lista de los comandos más ocupados dentro del sistema, así como de los que se necesitan para la ejecución de las herramientas.

II) Solstice AdminSuite.

Refiriéndonos a la herramienta `userglb`, Solstice proporciona un ahorro de trabajo tanto para la creación como para el borrado de usuarios. En el capítulo 2 se describió la forma en que se almacena la información de los usuarios a través de tablas, lo cual resulta ser una tarea delicada. El crear un usuario implica tener que interactuar con la información de las tablas, y aunque hacerlo de forma manual no es difícil, si requiere de mucho cuidado. Solstice proporciona el comando “`admuseradd`”, el cual permite crear usuarios desde la línea de comando y no requiere que el administrador se vea involucrado en el manejo directo de las tablas; además por sí mismo realiza la validación en el manejo de éstas.

Ejemplo: Forma general de la creación de un usuario.

```
admuseradd -u UID -c Nombre_del_user -d Home_directory -e Caducidad_de_cuenta
-g Grupo -m -k Directorio_Skel -s Shell -x ns=NIS+ -x serv=Nombre_Host -x
domain=Dominio -x pw=NORM -x passwd=Password -x cred=Y
-x mail=Host.Dominio -x perm=0700 -x autohome=Y Login
```

Cuando se habló de los requerimientos de software, se mencionó qué Solstice no era indispensable lo cual se debe a que se puede hacer una función dentro del programa `userglb` que realice las tareas del comando `admuseradd`, llevando a cabo el manejo de las tablas como se presenta en el ejemplo siguiente:

Ejemplo: Creación del usuario “otto”

```
nistbladm -a name=otto uid=30001 gid=15 gcos="Prueba otto" home=/home/otto
shell=/bin/csh passwd.org_dir
nisaddcred -l sun -p 30001 -P otto.dominio1.acatlan.unam.mx.local
dominio1.acatlan.unam.mx.
nisaddcred -l sun -p unix.30001@dominio1.acatlan.unam.mx
-P otto.dominio1.acatlan.unam.mx. des dominio1.acatlan.unam.mx.
nisclient -o -c -l sun otto
```

En apariencia el volumen de información manejado en ambos casos es el mismo, pero se debe tomar en cuenta que para el segundo caso el trabajo para la creación del usuario no



termina en el manejo de las tablas, sino que es necesario detallar algunos aspectos de la creación de la cuenta tales como la creación del home directory, el copiado de los archivos de inicialización y la asignación correcta de los permisos de acceso a las tablas¹⁰, entre otros.

Para el caso del borrado de una cuenta sucede algo similar, Solstice proporciona el comando "admuserdel".

Ejemplo: Eliminación del usuario otto.

```
admuserdel -r -x rmmail=Y -x serv=apolo.acatlan.unam.mx -x ns=NIS+  
-x domain=acatlan.unam.mx otto
```

De forma manual con el manejo directo de las tablas sería:

```
nistbladm -r otto  
nisadddred -r otto
```

Para terminar la eliminación de una cuenta sólo restaría eliminar el home directory del usuario, así como su correo electrónico que es almacenado en el directorio /var/mail.

De lo expuesto en esta sección se desprende que lo más recomendable para la creación y destrucción de usuarios, es utilizar los comandos que Solstice proporciona para tal efecto, debido a la flexibilidad que brindan; sin embargo es necesario recalcar que existen otros medios para realizar estas tareas. Por lo anterior, en el presente trabajo, a Solstice se le asigna la característica de necesario más no indispensable.

4.3.6. Diseño de menús, pantallas y procedimientos para las herramientas

I) Menús.

La estructura de los menús fue basada en las necesidades del administrador con respecto a cada herramienta; cuando se inició el proyecto existió la definición de un menú básico el cual ha sufrido modificaciones conforme han crecido y variado las necesidades del administrador. A la fecha la estructura de menús de cada herramienta está conformada como lo muestran las siguientes figuras.

¹⁰ Cuando se crea un usuario de forma manual los permisos de acceso a la tabla passwd.org_dir que se asignan por default son "----rmdr---r---", siendo que por seguridad del sistema debieran ser "-----r-----r---"

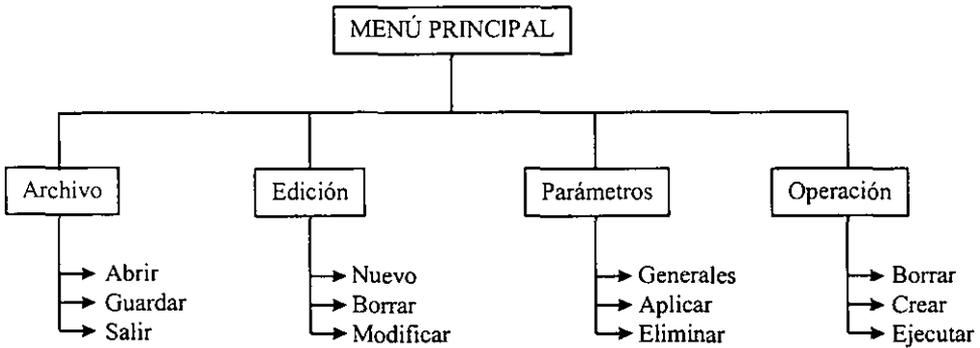


Fig. 4.6. Esquema de menús de la herramienta userglb.

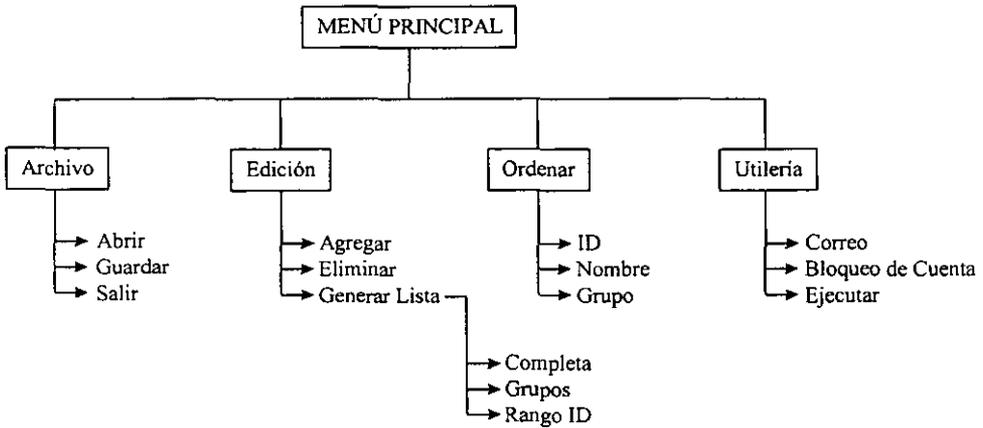


Fig. 4.7. Esquema de menús de la herramienta userlist.

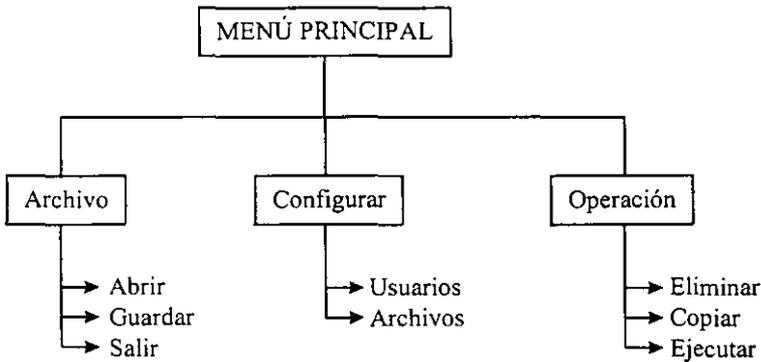


Fig. 4.8. Esquema de menús de la herramienta userarch.

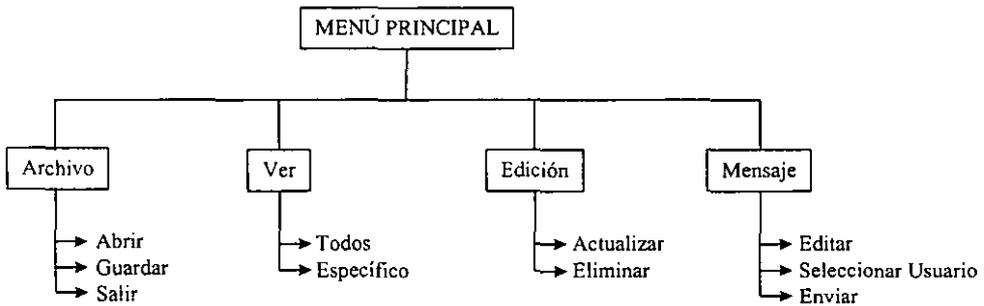


Fig. 4.9. Esquema de menús de la herramienta userprc.

II) Pantallas.

Cuando se realiza un programa con interfaz gráfica, en la mayoría de los casos se requiere demasiado tiempo para darle la presentación adecuada y más aún cuando no se tiene un programa que presente de forma inmediata lo que se quiere dibujar en pantalla. En la etapa de investigación se encontraron lenguajes (los cuales son mencionados a continuación) que salvaban este problema, pero que presentaban otros, tales como:

- ◆ Java Workshop y Workshop Visual. El costo del lenguaje es muy alto.
- ◆ SpecTCL. Genera demasiado código innecesario.
- ◆ Visual TCL. No funciona adecuadamente; su código contiene errores.

Dados los problemas anteriores, se decidió diseñar las pantallas manualmente, siguiendo un proceso de prueba y error y utilizando el packer como manejador de geometría.

Al final del diseño, las pantallas lucen como a continuación se presentan:

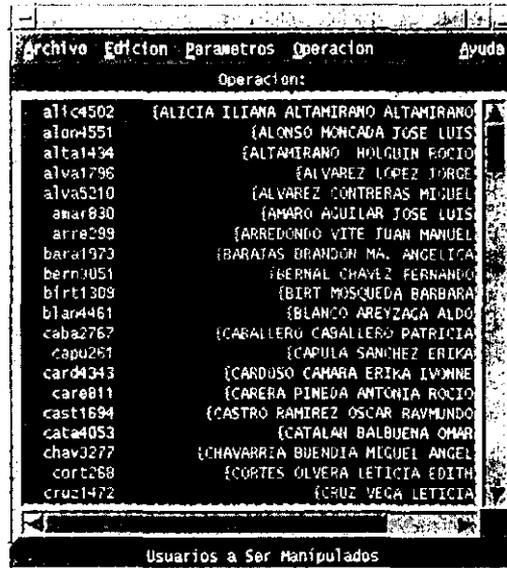


Fig. 4.10. Apariencia de la herramienta userglb.

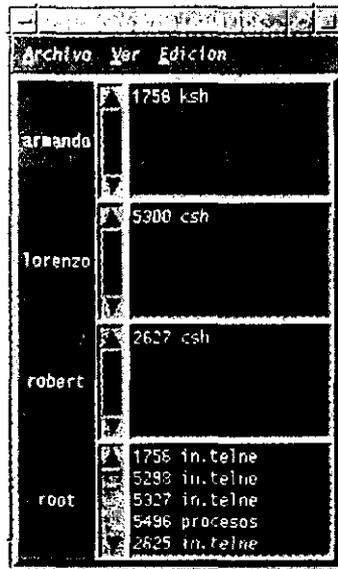


Fig. 4.11. Apariencia de la herramienta userprc.

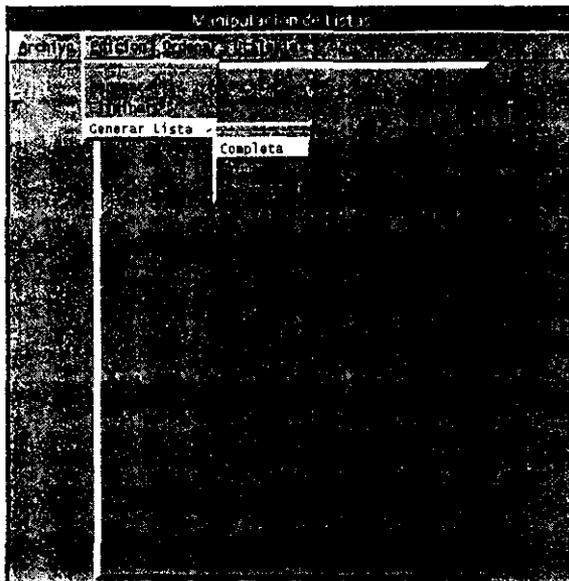


Fig. 4.12. Apariencia de la herramienta userlist.

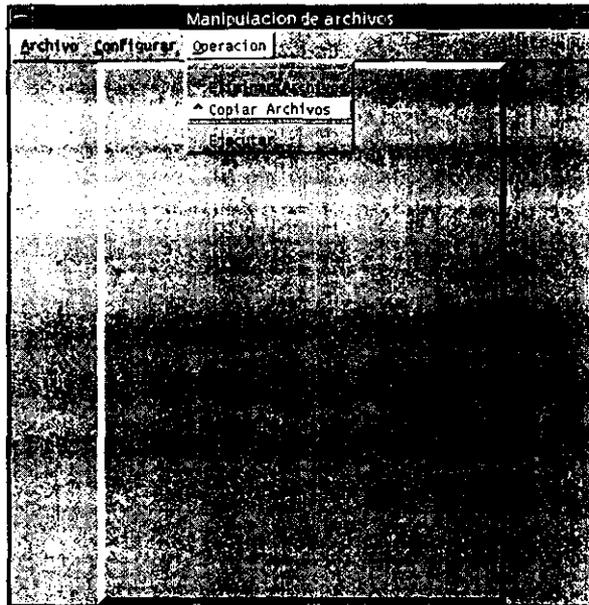


Fig. 4.13. Apariencia de la herramienta userprc.

III) Procedimientos.

La estructura de las herramientas se hizo de tal forma que fueran lo más independientes entre sí como fuera posible, por lo que cada herramienta cuenta con sus propios procedimientos. Ya fue descrito que los procedimientos se manejan de forma muy similar a como lo hace C, sin embargo, lo que se pretendió fue crear secciones dentro del código del programa, tanto para los eventos como para los procedimientos, teniendo su lugar estos últimos en la parte superior del mencionado código.

Las tareas que realiza cada herramienta difieren tanto, que prácticamente no existen procedimientos iguales o parecidos, por lo que no se presenta el problema de duplicidad de código entre las herramientas, y si en un futuro cuando se les de mantenimiento se requiere de procedimientos iguales, TCL da la oportunidad de crear algo conocido como “cabeceras”, en las cuales se pueden incluir procedimientos que después pueden ser llamados desde los programas de TCL.



4.4. IMPLEMENTACIÓN DE LAS HERRAMIENTAS

4.4.1. Pruebas realizadas

Cuando se diseña cualquier clase de programa, es necesario hacer pruebas para garantizar la confiabilidad de éste; el caso de las herramientas generadas no fue la excepción.

Realizar las pruebas de las herramientas de forma directa sobre el servidor implicaba demasiado riesgo, ya que el mismo da servicio las 24 horas del día¹¹; sin embargo se instaló un servidor de NIS+ sobre un host llamado “vulcano” y fue en esa máquina donde se realizaron las pruebas de las herramientas.

Se consideraron varios puntos en las pruebas las cuales se agruparon en las 3 categorías siguientes:

Tiempo de Respuesta. No tenía caso desarrollar herramientas que no agilizaran las tareas del administrador, es decir, el tiempo de respuesta con las herramientas debía ser considerablemente menor al método manual.

Seguridad. Dado que la mayoría de las tareas de un administrador son delicadas, no cualquier usuario debe poder acceder (ejecutar) las herramientas o alguno de sus procedimientos. Las responsabilidades de seguridad fueron delegadas al S.O., ya que proporciona métodos de validación muy adecuados.

Confiabilidad de la información. La confiabilidad de la información es vital para el buen funcionamiento del S.O. y en particular de NIS+.

4.4.2 Ajustes necesarios

Una vez satisfechas las pruebas pertinentes sobre el host vulcano, se prosiguió a instalar el conjunto de herramientas sobre el servidor apolo, en el cual se hicieron los cambios necesarios de las herramientas como el cambio de dominio de NIS+¹² y algunas rutas de acceso.

¹¹ El servicio que ofrece la red SUN y en particular el servidor abarca prácticamente las 24 horas del día, ya que de 8:00 a 21:00 se puede hacer uso del equipo en el D.S.C. y en las noches es consultado para revisar los correos electrónicos de los profesores o demás usuarios via internet.

¹² En una LAN no pueden existir dos dominios de NIS+ con el mismo nombre, por lo que cuando se instaló el servidor de prueba vulcano se manejó un dominio distinto al de apolo.



4.5 REPORTE DE RESULTADOS

Las herramientas en su primer etapa (motivo de este trabajo) han cumplido satisfactoriamente con los objetivos establecidos para cada una. El tiempo de respuesta para la atención de los usuarios ha disminuido, parte de la administración de la red se ha facilitado de sobremanera. Prueba fiel de lo mencionado es que las herramientas están siendo utilizadas en la actualidad y el ahorro de recursos principalmente humanos conllevan a brindar un mejor servicio.

Por citar algunos ejemplos mencionaremos que, para crear usuarios al principio del semestre se requería de al menos dos personas capturando la información durante un lapso de tiempo no menor a tres horas, tomando en cuenta además que el promedio de error en la captura aproximadamente era de dos usuarios de cada cien (ocasionado por el factor humano). En la actualidad por medio de la herramienta **userglb** se garantiza que la tasa de error se ha reducido prácticamente a cero, ya que ha sido utilizada dos veces, la primera en el semestre 97-1 y la segunda en el 98-2, y aunque en ambas ocasiones no ha presentado problemas no se puede considerar que la tasa de error es cero, ya que ha sido ocupada en pocas ocasiones; los problemas que presentan las cuentas no son en sí del S.O., sino de captura de ventanilla¹³. En el caso de **userrrpt** ha sido utilizada para monitorear la cantidad de usuarios que alberga la red y de esta forma ayudar a la toma de decisiones para crear o borrar usuarios del sistema, así como para crear reportes utilizados para otras tareas administrativas del D.S.C. Respecto a **userpre** su uso es diario, ya que los procesos de los usuarios tienden a revisarse de forma constante para el buen funcionamiento del S.O. y así evitar el mal uso de recursos de la red. Por último mencionaremos que **userarch** ha sido utilizada en momentos críticos, ya que de no hacer limpieza de archivos en las cuentas de los usuarios, el S.O. se pudo caer por la saturación del disco duro.

¹³ Todo alumno que desee utilizar la red SUN primero tiene que darse de alta en el sistema de ventanilla, el cual maneja la base de datos de los usuarios del D.S.C. El proceso para dar de alta usuarios en el sistema de ventanilla es totalmente independiente del manejo de la red SUN, sin embargo los datos que son utilizados para la creación de los usuarios dentro de la red son proporcionados por la base de datos de ventanilla, así que si existe un error en el proceso de captura, ocasionará un mal flujo de información reflejado también en la creación de los usuarios dentro de la red.

CONCLUSIONES

A través del trabajo se observó que la administración de usuarios utilizando NIS+ es un proceso complejo, que gracias a las herramientas gráficas con que cuenta el S.O. se puede volver "amigable" para el administrador. Sin embargo las herramientas que se proporcionen siempre serán insuficientes dada la variedad de tareas que realiza el administrador, por lo que el diseño y creación de herramientas auxiliares siempre será necesario. Cabe aclarar que las necesidades de los administradores varían considerablemente y que si una tarea es vital para una red, para otra puede resultar menos importante; en este tenor es conveniente señalar que las herramientas propuestas y creadas en este trabajo para la red SUN Microsystems de la ENEP Acatlán nacieron debido a que la manipulación de los "usuarios" es una de las cosas de mayor importancia.

Existe un vínculo muy importante entre la forma en que trabaja el sistema operativo con la información y las facilidades que nos brindan las herramientas gráficas; este vínculo desafortunadamente se pierde con mucha frecuencia ya que en la mayoría de las ocasiones, a un administrador lo único que le interesa es que el sistema trabaje bien sin darle importancia a como es que se realizan las operaciones; en este sentido el trabajo fue desarrollado con base en ese vínculo y se hizo hincapié en hacer ver como es que se lleva a cabo el manejo y transacción de la información de NIS+ tanto para un mejor entendimiento como para ayudar, en caso de ser necesario, a la creación de nuevas herramientas administrativas.

Ahora bien, dado que las herramientas propuestas en este trabajo ya están implementadas y actualmente están funcionando, el objetivo de las mismas se cumplió satisfactoriamente. El desarrollar las herramientas ha ayudado enormemente a realizar parte de las tareas de administración; y el tiempo de respuesta se redujo considerablemente pues, en el peor de los casos, éste es la mitad del que se utilizaba cuando las tareas se realizaban manualmente. La sistematización de diversas tareas permite a los administradores libertad para investigar e innovar en otros aspectos, por lo que la importancia de las herramientas creadas es relevante.

Independientemente de las herramientas, el presente trabajo ha superado las expectativas que de él se tenían, ya que para algunos estudiantes ha servido como punto de partida para incursionar en el ámbito de las redes, aún antes de que éste fuera terminado. Así mismo el trabajo es pionero en el conocimiento de otras redes, tal es el caso de la red RS6000 en la que el estudio realizado en la sección de servicio de nombres (capítulo 2) ayudó a comprender como se podía instalar y manejar NIS sobre equipos POWER PC de IBM.

A partir del desarrollo de este trabajo, TCL/TK se está dando a conocer en las redes UNIX dentro del D.S.C. como un lenguaje accesible, de fácil uso y que de ocuparse inteligentemente puede ser muy poderoso. Un claro ejemplo de lo antes mencionado es que en redes que en la actualidad se encuentran en boga, se utiliza TCL/TK como el lenguaje de

CONCLUSIONES

desarrollo para herramientas de configuración del sistema; tal es el caso de LINUX Red Hat en su versión 5.2.

Algunos razonamientos que quedaron del presente trabajo son los siguientes:

- La formación de la carrera nos ayuda a enfocar la mayoría de los problemas a una solución matemática a través de la optimización de recursos u otros medios, utilizando la computadora como una herramienta potencial de trabajo.
- La sistematización de cualquier tarea es uno de los muchos retos a los que se enfrenta un egresado de la Lic. en Matemáticas Aplicadas y Computación; sin embargo, en el desarrollo de la sistematización uno se va dando cuenta de los beneficios extras que se van obteniendo y se percata del por qué en la carrera se imparten materias que no son cien por ciento de matemáticas.
- Resulta interesante observar y comprender como se ven involucradas las matemáticas de forma aplicada en la arquitectura y funcionamiento de la computadora.
- Finalmente, es necesario decir que es satisfactorio el enfrentarse a un problema del ambiente laboral y contar con los elementos necesarios para resolverlo.

*“ A veces sentimos que lo que hacemos
es una gota en el mar, pero el mar
sería menos si le faltase esa gota “*

Teresa de Calcuta

ANEXO A

COMANDOS BÁSICOS MÁS COMUNES DENTRO DEL SISTEMA OPERATIVO UNIX

Inicio y término de una sesión

login	Petición de acceso al S.O.
logout	Petición de salida del sistema

Información del sistema

date	Despliega la hora y fecha
who	Despliega información de los usuarios conectados al sistema
whoami	Despliega el nombre del usuario que esta ocupando la cuenta
man	Despliega ayuda de un comando
whereis	Indica la ruta de acceso de un comando

Manipulación de archivos

ls	Muestra el contenido de un directorio especificado
cat	Despliega el contenido de un archivo
wc	Muestra el número de líneas, palabras y caracteres de un archivo
cp	Copia un archivo a la ruta indicada
mv	Cambia el nombre de un archivo
more	Despliega el contenido de un archivo, una pantalla completa a la vez.
rm	Borra un archivo
head	Despliega la primera parte de un archivo
tail	Despliega la última parte de un archivo
diff	Presenta diferencias entre dos archivos o directorios
find	Localiza archivos dentro del sistema de archivos
grep	Busca un patrón (cadena de caracteres) en un archivo
sort	Ordena y/o mezcla archivos

Manipulación de directorios

cd	Cambia la posición actual del directorio
mkdir	Crea un directorio
rmdir	Elimina el directorio indicado, siempre y cuando se encuentre vacío
pwd	Indica la trayectoria del directorio actual

Utilerías de comunicación	
mail	Envía o recibe correo
write	Envía un mensaje a otro usuario
mesg	Habilita o inhabilita la recepción de mensajes

Utilerías diversas	
alias	Asigna un nombre secundario a un comando
ps	Muestra el estado de los procesos
kill	Elimina un proceso
cut	Extrae campos de cada línea de un archivo
sleep	Pone a "dormir" un proceso
clear	Limpia pantalla

Seguridad del sistema	
passwd	Cambia el password
chmod	Cambia los permisos de los archivos
chown	Cambia archivos de dueño
chgrp	Cambia archivos de grupo

GLOSARIO

<i>SIGLAS</i>	<i>SIGNIFICADO</i>
<i>Aplicación</i>	Programa que puede correr sobre el sistema X-windows.
<i>ARP</i>	Address Resolution Protocol. Protocolo encargado de codificar direcciones IP a direcciones Ethernet.
<i>Autenticación</i>	Proceso de validación de requisiciones hacia el servidor.
<i>Autorización</i>	Proceso de aceptación de los clientes que hacen una requisición hacia el servidor.
<i>Basic</i>	Lenguaje de programación.
<i>Browser</i>	Software utilizado para la visualización de hojas HTML a través de la Internet.
<i>BUS</i>	Cable por el cual son transmitidos los datos de la red.
<i>Button</i>	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
<i>Canvas</i>	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
<i>Caracter especial</i>	Es un caracter que tiene un significado especial para el lenguaje de programación.
<i>CDE</i>	Common Desktop Environment. Interfáz gráfica utilizable dentro del S.O. Solaris.
<i>Checkbutton</i>	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
<i>Cliente</i>	Es una máquina que has sido preparada para recibir los servicios de NIS+.
<i>Comando</i>	Tarea que entiende el intérprete del S.O. (shell) y que debe realizar.
<i>Concentrador</i>	Dispositivo físico que controla el flujo de datos a través de una red.
<i>Credencial</i>	Identificador de un cliente de NIS+ que sirve para confirmar la identidad de las requisiciones de acceso hacia los objetos de NIS+.
<i>Cursor</i>	Apuntador pequeño que aparece en la pantalla de una máquina indicando donde se empezara a escribir.
<i>DES</i>	Tipo de credencial que permite hacer requisiciones desde un cliente.
<i>DGSCA</i>	Departamento General de Servicios de Cómputo Académico.
<i>Directorio</i>	Objeto de NIS+ que conforma el esqueleto del namespace.
<i>Display</i>	Pantalla de un host en el cual será desplegada la aplicación.
<i>DNS</i>	Domain Name Service. Servicio de Nombres de Dominio, que es ocupado por el protocolo TCP/IP.
<i>Dominio</i>	Conjunto de directorios, tablas y grupos dentro de un namespace.
<i>D.S.I.</i>	Departamento de Servicios de Información.
<i>D.S.C.</i>	Departamento de Servicios de Cómputo.
<i>E-MAIL</i>	Identificar de correo electrónico de los usuarios de un sistema con

	capacidad de conexión a internet.
Encriptar	Codificación de caracteres en base a un algoritmo que permite ocultar su valor real.
Entrada	Identificador para referenciar el renglón de una tabla de NIS+.
Entré	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
Estación de Trabajo	Máquina conectada a red que tiene asignados un Host Name y una IP Address.
Esclavo	Terminó utilizado para describir una ventana de TK, que depende de otra ventana.
Ethernet Address	Dirección asignada a una tarjeta de comunicación expresada en numeración exadecimal.
Evento	Suceso que ocurre mientras esta corriendo una aplicación (click del mouse, opresión de una tecla en específico, etc).
FNS	Federal Name Service. Servicio de información de nombres que puede ser utilizado por Solaris.
Fortran	Lenguaje de programación.
Frame	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
Grupo	Son asociaciones de NIS+ principals agrupados por conveniencia del administrador del sistema.
GUI	Graphic User Interface. Interfáz gráfica para usuario.
Home Directory	Directorio personal que es asignado a cada usuario y en el cual es depositado cuando se accesa al sistema.
Host	Máquina que se encuentra conectada a la red.
Host Name	Identificador alfanumérico asignado a una máquina.
HTML	Hiper Text Markup Lenguaje. Lenguaje de marcaje de hipertexto enfocado para la creación de páginas presentadas en la Internet.
Icono	Pequeño dibujo simbolico que representa una aplicación.
ICMP	Internet Control Message Protocol. Protocolo encargado de llevar los mensajes de error de la red.
Instrucción	Acción a realizar dentro de un lenguaje de programación.
Interfáz Gráfica	Concepto relacionado con el poder de despliegue en pantalla no solo de tipo carcater, sino que tambien de tipo gráfico.
Internet	La red más grande del mundo en la que se encuentra la mayor cantidad de información existente.
IP	Internet Protocol. Protocolo de internet responsable de el envío de paquetes de información.
IP Address	Internet Protocol Address. Identificador que es ocupado por el protocolo TCP/IP para ubicar una máquina que se encuentre conectada a Internet.
JAVA	Lenguaje de programación cuya finalidad es poder correr programas a travez del browser.
Label	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I)

	para mayor referencia.
<i>LAN</i>	Red de área local.
<i>Listbox</i>	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso 1) para mayor referencia.
<i>LOCAL</i>	Tipo de credencial que permite que un cliente pueda hacer requisiciones desde el master server.
<i>Login</i>	Identificador ocupado por el sistema para ubicar al usuario que esta o pretende conectarse al sistema.
<i>Manejador de Geometría</i>	Es un concepto utilizado para describir la parte del lenguaje de programación que se encarga de manejar la apariencia de los widgets.
<i>Mapeos o Mapas</i>	Son una forma de almacenamiento de la información del servicio de nombres de NIS.
<i>Master Server</i>	Servidor principal que almacena la información de NIS+.
<i>Mbp</i>	Mega Bit por segundo. Unidad de medida para la transmisión de datos.
<i>Menubutton</i>	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso 1) para mayor referencia.
<i>Message</i>	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso 1) para mayor referencia.
<i>Mirror</i>	Máquinas de respaldo utilizadas para crear resguardos de la información de NIS+ contenida en el Master Server.
<i>MOTIF</i>	Conjunto de librerías utilizables por los sistemas X-Windows.
<i>Namespace</i>	Es el espacio de nombres que abarca el servicio de NIS+. Es la forma en que esta estructurado el servicio de NIS+.
<i>NIS</i>	Network Information Service. Servicio de información de nombres que puede ser utilizado por Solaris.
<i>NIS+</i>	Network Information Service plus. Servicio de información de nombres que puede ser utilizado por Solaris y que es mejor a NIS.
<i>NIS+ Principals</i>	Es el identificador con el cual se diferencia entre un cliente de tipo usuario y el de máquina.
<i>Nodo</i>	Lugar en donde se realiza la conexión física de una máquina para comunicarse con la red.
<i>Objetos de NIS+</i>	Son los elementos que forman parte del NIS+ Namespace (directorios, tablas y grupos).
<i>Packer</i>	Manejador de geometría de TK.
<i>Palabra Reservada</i>	Como su nombre lo dice es una palabra que no puede ser utilizada para declarar una función o una variable dentro de un programa, debido a que tiene un significado especial para el lenguaje que se este manejando.
<i>Pascal</i>	Lenguaje de Programación.
<i>Password</i>	Identificador secreto asociado al login sin el cual no se puede acceder al sistema.
<i>Permisos de NIS+</i>	Son aquellos permisos que pueden ser asignados a los objetos de NIS+ para el acceso o no acceso a la información.
<i>Placer</i>	Manejador de geometría de TK.

Principal	Máquina o usuario que esta haciendo una requisición al master server.
Protocolo	Software encargado de establecer los lineamientos o reglas para el envío y recepción de información a través de la red.
Puertas	Son sistemas de seguridad que validan el acceso hacia algún proceso en específico.
Radiobutton	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
Ratón	Dispositivo físico que sirve para apuntar en la pantalla.
RARP	Reverse Address Resolution Protocol. Protocolo encargado de codificar direcciones Ethernet a direcciones IP.
Red	Conjunto de máquinas interconectadas entre sí que comparten información.
Root Master	Cuenta de usuario especial que tiene los privilegios necesarios para modificar características del sistema operativo y que es asignada al administrador del sistema.
Scale	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
Scrollbar	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
Servidor	Máquina central encargada de procesar la información más importante (los servicios de NIS+)
Shell	Interprete de comandos del S.O. UNIX.
Solstice	Conjunto de herramientas que ayudan a la administración del S.O. Solaris.
SUN	Marca de un equipo cuyo S.O. es Solaris.
Switch	Dispositivo físico utilizable en la topología de una red.
Tablas	Objetos de NIS+ que se encargan de almacenar la información de usuarios y máquinas pertenecientes al dominio de NIS+.
TCL	Tool Command Language. Lenguaje de comandos de línea parecido al shell del S.O.
TCL/TK	Lenguaje de programación que conjunta el poder de TCL y TK para permitir la creación de programas visibles en sistemas X-windows.
TCP	Transport Control Protocol. Protocolo de internet responsable de verificar el buen envío y recepción de los paquetes de información.
TCP/IP	Transport Control Protocol / Internet Protocol. Protocolo utilizado por la internet para el envío y recepción de información.
Text	Es un widget con un objetivo específico. Véase la sección 4.2.2 Inciso I) para mayor referencia.
TK	Tool Kit. Lenguaje de manipulación de ventanas sobre sistemas X-windows y que es complemento de TCL.
Topología	Forma en que se encuentra interconectada una red.
UDP	User Datagram Protocol. Protocolo que cumple con funciones similares a TCP, pero siendo menos confiable.
UNIX	Sistema operativo cuya estructura esta basada en el manejo de archivos.

GLOSARIO

<i>userglb</i>	Herramienta creada con TCL/TK y comandos de UNIX para la creación y borrado de usuarios de forma masiva.
<i>userprt</i>	Herramienta creada con TCL/TK y comandos de UNIX para la creación de reportes de usuarios del sistema.
<i>userprc</i>	Herramienta creada con TCL/TK y comandos de UNIX para la manipulación de procesos de los usuarios.
<i>userarch</i>	Herramienta creada con TCL/TK y comandos de UNIX para la manipulación de archivos de los usuarios.
<i>Variable</i>	Término ocupado dentro de los lenguajes de programación que se refiere a un contenedor de información.
<i>Ventana</i>	Área limitada en donde una aplicación realiza la mayoría de sus funciones.
<i>WAN</i>	Red de trabajo estensa.
<i>Widget</i>	Nombre que reciben las ventanas predeterminadas que maneja TK (Label, Button, etc.).
<i>X-Windows</i>	Interfáz Gráfica utilizable dentro de los S.O. UNIX.

BIBLIOGRAFÍA

1. RAMSEY, Rick, All About Administering NIS+, Ed. Prentice-all, California E.U. 1994
2. WINSOR, Janice, Solaris Advanced System Administrator's Guide, Ed. Ziff-Davis Press, Emeryville California E.U., 1996
3. SCHATT, Stan, Understanding Local Area Networks, Ed. Prentice Hall, California E.U., 1994
4. OUSTERHOUT, John K., TCL and TK, Ed. Addison Wesley, California, 1995
5. CD de información del sistema Solaris, System Administrator Answerbook, Mountain View California, 1996

Direcciones de Internet:

Conceptos relacionados con redes computacionales.

- <http://emision.uson.mx/detodounpoco/glosario.htm>
- http://dir.yahoo.com/Computers_and_Internet/Communications_and_Networking/
- <http://netfactor.net.mx/soporte/glosario.html>
- <http://ns.uca.edu.sv/glosario.html>

Conceptos e Información relacionada con NIS+.

- <http://docs.sun.com>
- <http://www.sun.com>

Libros, manuales de referencia y ejemplos de TCL/TK.

- <http://sunscript.sun.com>
- http://dir.yahoo.com/Computers_and_Internet/Programming_Languages/Tcl_Tk/
- <http://www.neosoft.com/tcl/default.html>
- <http://http2.brunel.ac.uk:8080/~csstdm/TCL2/TCL2.html>
- <http://www.beedub.com/book/2nd/booktoc.mif.html>
- <http://www.elf.org/tcltk-man-html/contents.htm>
- <http://www.neuron.com/>