



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**“Sistema de Administración del
Tráfico de una Red de Cómputo”**

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE
Ingeniero en Computación

P R E S E N T A

ISSET GUERRERO GALACHE

DIRECTOR DE TESIS:

L.I. NEPTALÍ GONZÁLEZ GÓMEZ

CODIRECTOR DE TESIS:

ING. FERNANDO JAVIER MARTÍNEZ MENDOZA





Universidad Nacional
Autónoma de México



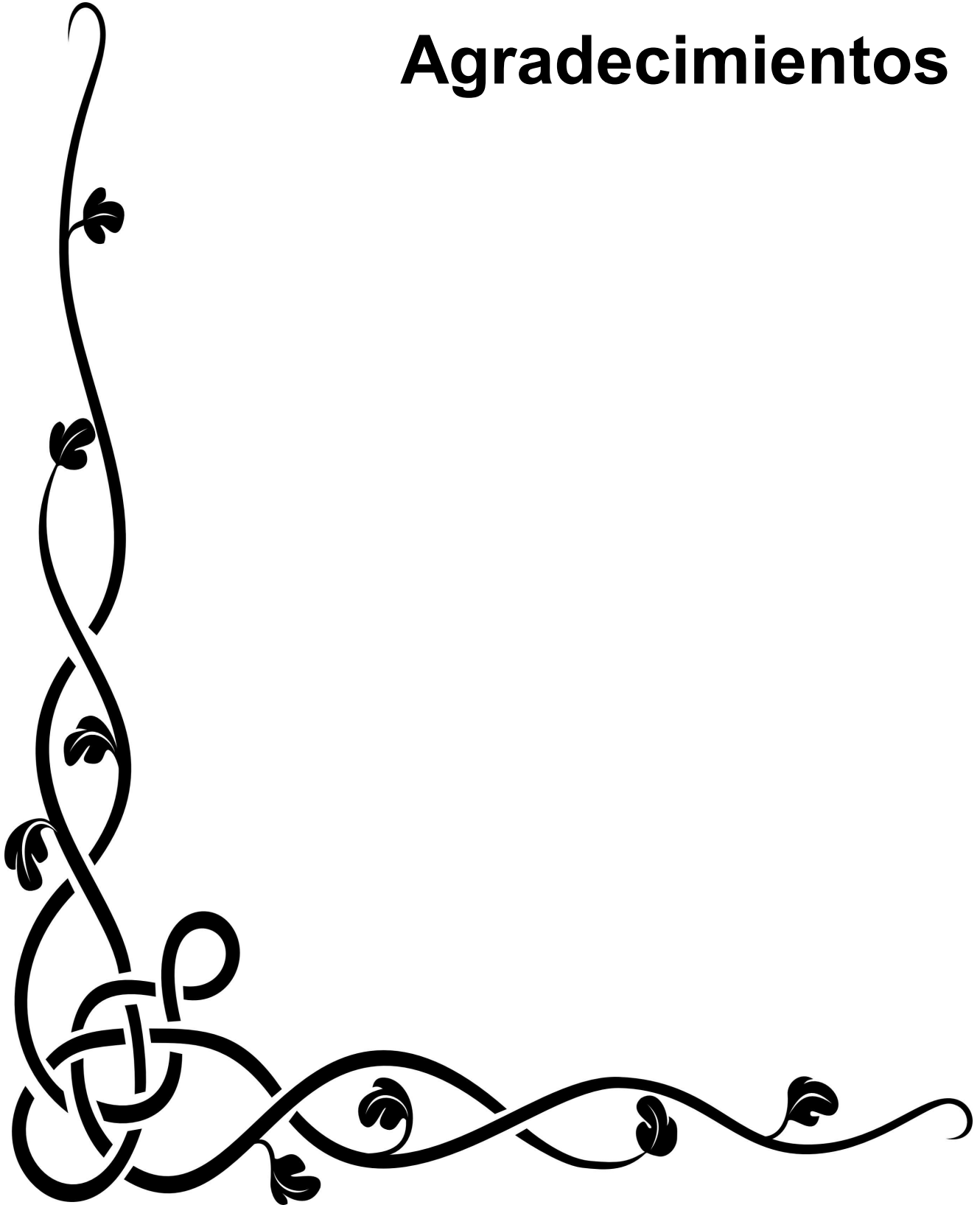
UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

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



Agradecimientos

A mis padres Marcelina Galache Vega y Alejandro Guerrero Fuentes por todo el cuidado y cariño que siempre nos brindaron a mis hermanos y a mí, por ofrecernos un hogar lleno de amor y que siempre tiene las puertas abiertas cuando necesitamos su consejo, por todos los valores y enseñanzas que nos inculcaron, por nunca defraudarnos y por mantenernos unidos en las buenas y en las malas. Este trabajo también es un logro de ustedes, gracias por dejarme desenvolverme en lo que más me gusta. Son uno de mis pilares principales y ejemplo a seguir en esta vida, los amo.

A mis hermanos Lorna, Iván y Juan Antonio por hacer que mi niñez fuera de lo más divertida, mi vida sin ustedes no sería lo mismo. Ahora que somos adultos y cada quien ha seguido su camino me alegro que sigamos tan unidos a pesar de las distancias, gracias por todas las risas y llantos, por todas las alegrías y tristezas y por hacer más grande a la familia con tan maravillosos sobrinos. Los quiero con todo mi corazón.

A Gustavo Gómez Macías por estar siempre a mi lado y apoyarme en los buenos y malos momentos, por amarme con todos mis defectos y virtudes. Eres mi luz en el camino y doy gracias por que seas parte de mi vida. Gracias por enseñarme de todo y alentarme a llegar siempre a lo más alto, por no dejarme caer y compartir tu vida conmigo, este logro también es gracias a ti. Te amo.

A mis queridos amigos Jacqueline Segura Ortiz y Juan Antonio Contreras Martínez por todos los años de desvelos, estudios, triunfos, tropiezos, risas y bromas que compartimos durante la carrera. Gracias por estar siempre ahí tanto en lo personal como en lo profesional. La facultad fue mejor de lo esperado gracias a su presencia y espero que sigamos compartiendo nuestras vidas

Al L.I. Neptalí González Gómez por ser mi maestro y amigo, gracias por todas las enseñanzas y consejos que me has dado. Por corregirme cuando lo he necesitado y por alentarme siempre a dar lo mejor en el trabajo. También por creer en mí y permitirme desarrollar Matlatl con tecnologías nuevas, fue un tiempo de gran aprendizaje y que he

aprovechado al máximo.

A mis sinodales M.C. Ma. Jacqueline López Barrientos, Ing. Laura Sandoval Montaña, Ing. Fernando Javier Martínez Mendoza e Ing. Eduardo Ramírez Sánchez por todas sus observaciones y consejos para mejorar el presente trabajo.

A la Facultad de Ingeniería por la formación que me brindó y por los maestros que compartieron sus conocimientos con nosotros. Ingeniería es y será una parte fundamental y muy importante en mi vida.

Al Instituto de Física de la UNAM por dejarme ingresar a sus filas, primero como estudiante y luego como colaboradora de trabajo, por permitirme desarrollar mis habilidades profesionalmente.

Y por último, pero no menos importante, a la UNAM por ser mi segundo hogar, porque es aquí donde conocí a muchos de los maestros, amigos, compañeros y académicos que han marcado mi existencia, gracias por todo lo que me has ofrecido.

Índice General

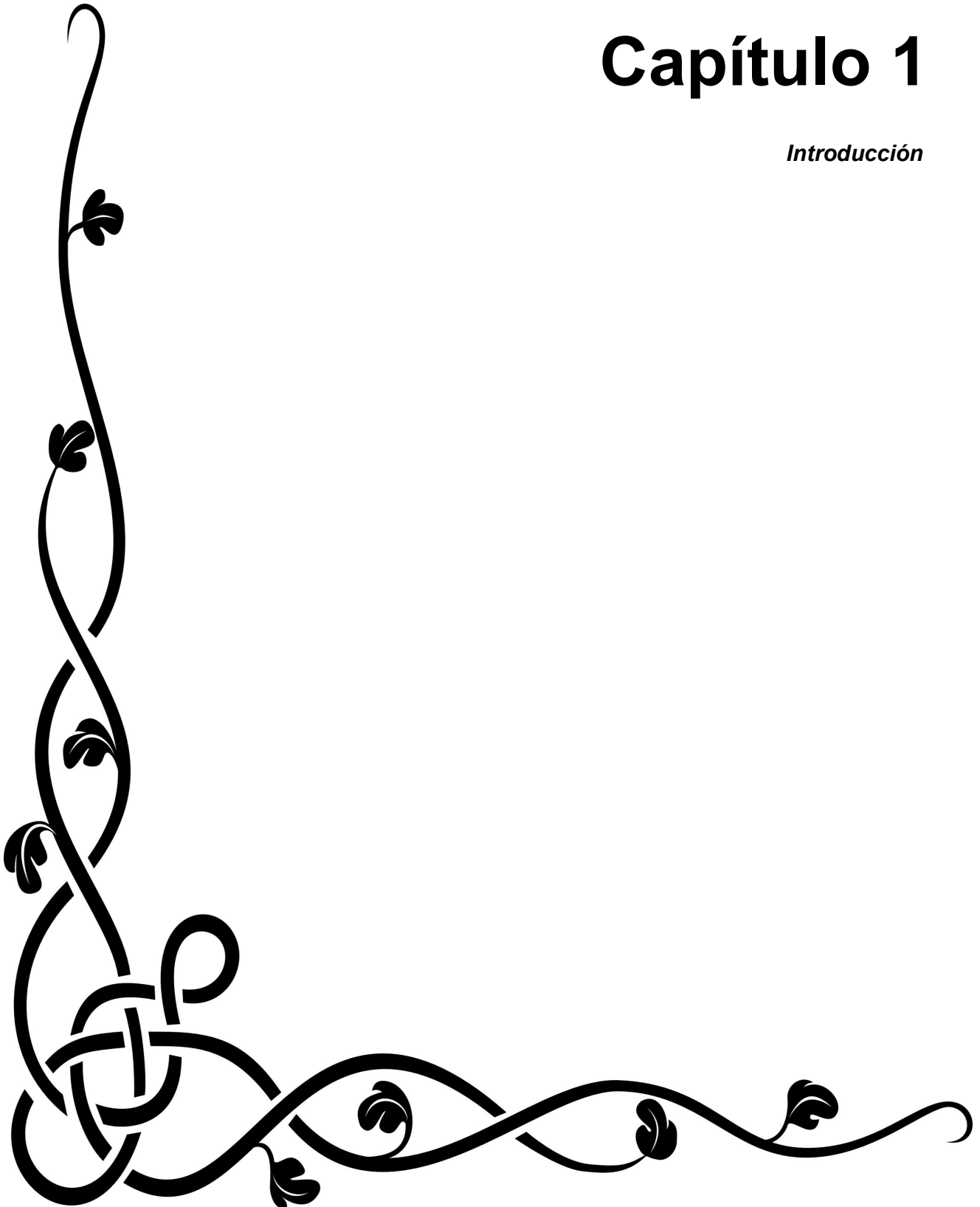
1. Introducción.....	9
1.1. Breve Historia.....	9
1.2 Problemática y Necesidades	11
1.3 Objetivo del Trabajo	12
1.4 Estructura General	13
2. Redes de Computadoras	16
2.1 Conceptos Básicos.....	16
2.1.1 Red de Computadoras	16
2.1.2 Clasificación de Redes.....	16
2.1.2.1 Clasificación por tecnología de transmisión	17
2.1.2.2 Clasificación por su extensión geográfica	17
2.1.3 Topologías de Red.....	19
2.1.4 Modelos de Referencia	24
2.1.4.1 Modelo OSI (Interconexión de Sistemas Abiertos).....	24
2.1.4.2 Modelo TCP/IP.....	27
2.2 Control de Acceso al Medio (MAC – Media Access Control).....	29
2.3 Dirección IP	30
2.4 El Modelo Cliente/Servidor	31
2.5 Protocolo SSH (Protocolo de conexiones seguras).....	32
2.6 Protocolos de Comunicación	33
2.6.1 Protocolo TCP (Protocolo de Control de Transmisión).....	33
2.6.2 Protocolo IP (Protocolo de Internet)	35
2.7 Servidor DHCP	37
2.7.1 Funcionamiento de un servidor DHCP	37
2.8 Sistema Operativo.....	39
2.8.1 GNU/Linux	39
2.8.2 Razones para usar GNU/Linux.....	41
3. Software para el Monitoreo de Redes.....	44
3.1 Elección del software	45
3.1.1 CACTI	45
3.1.2 CRICKET	46
3.1.3 MRTG.....	46
3.1.4 IPTRAF	47
3.2 IP Flow Meter (IPFM).....	47
3.3 Herramientas de Base de Datos Circulares (RRDTool).....	49
3.3.1 Bases de Datos Circulares (RRD – Round Robin Database)	49
3.3.2 Componentes de una RRD.....	52
3.3.3 Herramientas más útiles.....	55
4. Procedimientos para la Instalación y Configuración del Software.....	58
4.1 Instalación del Sistema Operativo	58

4.2	Instalación y configuración del servidor DHCP	76
4.3	Instalación y configuración del software para Monitoreo y Graficación	80
4.3.1	Instalación y configuración de IPFM.....	80
4.3.2	Instalación de RRDTool.....	81
4.4	Ejecución de scripts para el monitoreo de la red	81
4.4.1	Scripts para generar listas de hosts.....	81
4.4.2	Scripts para generar las Bases de Datos Circulares	83
4.4.3	Scripts para ingresar información a las bases de datos	86
4.4.4	Scripts para generar gráficas	91
4.5	Creación de un cron para ejecutar tareas programadas	104
5.	Sistema MATLATL para la Administración del Tráfico de la Red de Datos del Instituto de Física de la UNAM.....	108
5.1	Selección de metodología para el desarrollo de software	108
5.1.1	Métodos orientados al plan.....	109
5.1.2	Métodos ágiles	109
5.1.3	Diferencias entre metodologías ágiles y tradicionales.....	110
5.1.4	Razones para usar metodologías ágiles	111
5.2	Ruby on Rails	111
5.2.1	El lenguaje de programación Ruby.....	112
5.2.2	El framework Rails.....	113
5.3	Instalación de Ruby on Rails (RoR)	115
5.4	Implementación del Sistema Matlatl.....	117
5.4.1	Casos de uso de Matlatl.....	118
5.4.2	Creación de la Base de Datos.....	120
5.4.3	Presentación de los Datos.....	121
5.4.4	Instalación e implementación de Auto_Complete	122
5.4.5	Instalación e implementación de Will_Paginate.....	125
5.4.6	Instalación e implementación de PaperClip	127
5.4.7	Creación de una caja de búsqueda.....	130
5.4.8	Validaciones.....	133
5.4.9	Logeo y autenticación de usuarios.....	135
5.5	Liberar la versión de producción.....	143
	Conclusiones.....	148
	Anexo A - Manual de Usuario.....	150
1.	Ingreso al Sistema Matlatl.....	150
2.	Iniciar Sesión	151
3.	Funciones para la página de Direcciones IP.....	154
3.1.	Funciones básicas.....	155
3.2.	Funciones extras.....	160
4.	Funciones para la página de Usuarios de Red.....	163
4.1.	Funciones básicas.....	165
5.	Funciones para la página de Direcciones MAC	169
5.1.	Funciones básicas.....	170
5.2.	Funciones extras.....	172
6.	Funciones para la página de Roles.....	174
7.	Funciones para la página de Altas.....	175

8. Orden de Borrado.....	177
9. Terminar Sesión.....	178
Anexo B – Topología de Red IFUNAM.....	180
Referencias	182
Mesografía.....	183

Capítulo 1

Introducción



“Todavía hay mucha gente que ofrece contenidos por el mero placer de saber que la información puede resultar útil a otras personas.”
Vinton Cerf

Introducción

En la actualidad, las redes de computadoras han adquirido una gran importancia, no solo en el ámbito laboral y personal, también en el ámbito científico, ya que han derribado barreras geográficas permitiendo que ciencia y tecnología compartan recursos para el intercambio de información. Esto facilita el acceso a equipo de cómputo sofisticado y la colaboración de gente en todo el mundo aumentando la eficiencia de los proyectos. Para entender mejor este proceso es necesario conocer sus inicios.

1.1. Breve Historia

Las redes de computadoras tienen su origen en el proyecto ARPANET, el cual fue desarrollado durante el auge de la Guerra Fría. Su principal objetivo era crear un sistema de información descentralizado y así evitar la pérdida total de información valiosa en caso de un ataque.

Hay que tomar en cuenta que fue un proyecto que tardó años en desarrollarse, y que necesitó el esfuerzo de científicos y estudiantes para lograr sus objetivos. A continuación se hablará un poco de su historia y evolución.

ARPANET

El lanzamiento en 1957 del primer satélite artificial “*Sputnik*”, por parte de la Unión Soviética, orilló a los científicos e ingenieros de Estados Unidos a ser más competitivos, por lo que el presidente Eisenhower decidió crear la **Agencia de Proyectos de Investigación Avanzada (ARPA – Advanced Research Project Agency)** dentro del

Departamento de Defensa, para establecer su liderazgo en el área de la ciencia y la tecnología aplicadas a las fuerzas armadas. Su objetivo principal radicaba en superar el desafío de realizar enlaces e intercambio de información entre centros de investigación y el Departamento de Defensa.

Esta investigación adquirió mucha importancia debido a que el rol inicial de las computadoras como calculadoras aritméticas se expandía al de medios de comunicación.

El director de ARPA en 1968, Larry Roberts, se interesó en las redes, y decidió contactar a algunos expertos. Entre ellos se encontraba Wesley Clark, quien propuso la implementación de una subred de conmutación de paquetes, donde cada *host* contaría con su propio enrutador. De esta forma si un nodo desaparecía, la red no se vería afectada, ya que los datos se enviaban automáticamente por diferentes rutas.

La idea fue aceptada, cada subred estaría formada por minicomputadoras llamadas **Procesadores de Mensajes de Interfaz (IMP – Interface Message Processors)**, conectadas por lo menos a otras dos IMP's mediante líneas de transmisión de 56 kbps.

ARPA convoca a universidades y empresas para construir la subred. La empresa BBN de Cambridge, Massachusetts, gana la licitación y en diciembre de 1968 es contratada para construir la subred y desarrollar su software.

En diciembre de 1969 se crea la primera red de computadoras denominada ARPANET, compuesta de cuatro nodos situados en UCLA (Universidad de California en Los Ángeles), SRI (Stanford Research Institute), UCSB (Universidad de California de Santa Bárbara) y la Universidad de Utah. Cada *host* tenía características diferentes y eran incompatibles entre ellas para que el reto fuera más interesante. En 1970 se comienza a utilizar el protocolo *host-to-host* para poder comunicarse.

En 1971 Ray Tomlinson desarrolla el primer programa de correo electrónico para distribuir mensajes entre usuarios de ARPANET. En el siguiente año deciden utilizar el @ como separador entre el usuario y el nombre de la máquina que lo hospeda. Finalmente en 1973 logran realizar la primera comunicación internacional entre ARPANET y el Colegio

Universitario de Londres.

Posteriormente fueron agregándose más nodos pero los protocolos de transmisión comenzaron a ser inadecuados para utilizarse entre diversas redes, por lo que las investigaciones que se realizaron para mejorar éste aspecto los llevo a la creación del modelo y protocolo TCP/IP, desarrollado por Vinton Cerf y Robert Kahn en 1974.

Basándose en el sistema UNIX los investigadores en Berkeley desarrollan una interfaz para la red (*sockets*) que en conjunto con el protocolo TCP/IP y el sistema 4.2BSD fueron ideales para que las Universidades pudieran conectar las nuevas computadoras VAX y las LAN's que habían adquirido recientemente.

En la década de los 80's el escalamiento con *hosts* se volvió muy costoso por lo que se crearon los **Sistemas de Nombres de Dominios (DNS – Domain Name System)** para que las máquinas estuvieran dentro de un dominio y los nombres de los hosts se resolvieran en direcciones IP, además ARPANET establece que el protocolo TCP/IP será el único protocolo que utilizarán.

A partir de éste momento el desarrollo de tecnología y el crecimiento de redes es exponencial, dando origen a lo que hoy conocemos como Internet, la red de redes.

1.2 Problemática y Necesidades

La red de cómputo del IFUNAM es altamente dinámica, cuenta con muchos dispositivos y próximamente se adicionaran nuevos elementos a la red (VoIP mediante el uso de teléfonos SIP). Por lo que es necesario contar con un sistema de estadísticas y uso de la red.

Los requerimientos solicitados para el sistema son los siguientes:

- El sistema deberá estar basado en Software OpenSource.
- La interfaz de usuario sera vía Web.
- Proporcionara estadísticas individuales de cada computadora personal conectada a la

red.

- El acceso a estas estadísticas se podrá realizar verificando la dirección IP, el usuario o la dirección MAC.
- Proporcionara la ubicación de la computadora personal.
- Datos personales del usuario del equipo.
- Detalles de la configuración del dispositivo (dirección IP, dirección MAC y puerto del Switch al que esta conectado).
- Gráficas de utilización de la red.

Se requiere el monitoreo para:

- Observar el comportamiento del uso del ancho de banda.
- Obtener estadísticas de uso del ancho de banda de un usuario o elemento en la red.
- Determinar el consumo de ancho de banda en horarios laborales y no laborales.
- Verificar si una PC o un conjunto de PCs pertenecen a una *botnet*¹.
- Adicionalmente, el sistema también se usara para la administración de las asignaciones de direcciones IP.
- Registro de las direcciones MAC.
- Identificación y ubicación de los nodos de red.
- Nombre, ubicación, y datos para contactar a los usuarios.
- Identificar a que puerto en el switch esta conectada la PC o el dispositivo de red.

1.3 Objetivo del Trabajo

El objetivo principal es diseñar un sistema que aportará a la Secretaría Técnica de Cómputo y Telecomunicaciones información que permita la visualización de la carga de red de un usuario, lo que ayudará a identificar a los Investigadores con mayor demanda de utilización de red y ayudará a regular apropiadamente los requerimientos de sus aplicaciones. Esta regulación ayudará a reducir al mínimo los problemas de la red.

¹ Los *bots* o *botnets* son programa maliciosos que permite a un intruso tomar el control de un equipo infectado. El equipo víctima es comúnmente conocido como *zombie*, debido a que solo cumple ordenes del *bot*.

1.4 Estructura General

El presente trabajo se divide en cuatro capítulos donde se explican las etapas que se siguieron para desarrollar el sistema web, a continuación se presenta una breve descripción.

Capítulo 2

Para comprender cómo funciona una red de datos es importante abordar conceptos básicos tales como su clasificación, las topologías utilizadas, los protocolos de comunicación que permiten el intercambio de información y los modelos de referencia aplicables a su estudio y su implementación. También se habla de lo que es un servidor DHCP y las razones por las que se tomó la decisión de trabajar con un sistema operativo GNU/Linux.

Capítulo 3

En este apartado se presenta una comparación entre el software (de distribución libre) más usado en la actualidad y empleado en el monitoreo de red, la cual justifica la decisión tomada respecto a usar una aplicación en particular, además se aborda el tema de las bases de datos circulares para entender ¿Qué son? ¿Cómo se trabaja con ellas? ¿Porqué nos conviene usarlas?, así como de las herramientas que nos ayudan en el proceso de graficación.

Capítulo 4

En dicha sección se explica el procedimiento que se siguió para instalar el sistema operativo *Ubuntu Server Edition 8.04*, además de la instalación y configuración del servidor DHCP y del software de monitoreo y graficación. También se presentan los *scripts* diseñados para realizar la recolección de los datos obtenidos por IPFM, los encargados de almacenar los datos en las RRDs y los que generan las gráficas que se muestran en línea.

Capítulo 5

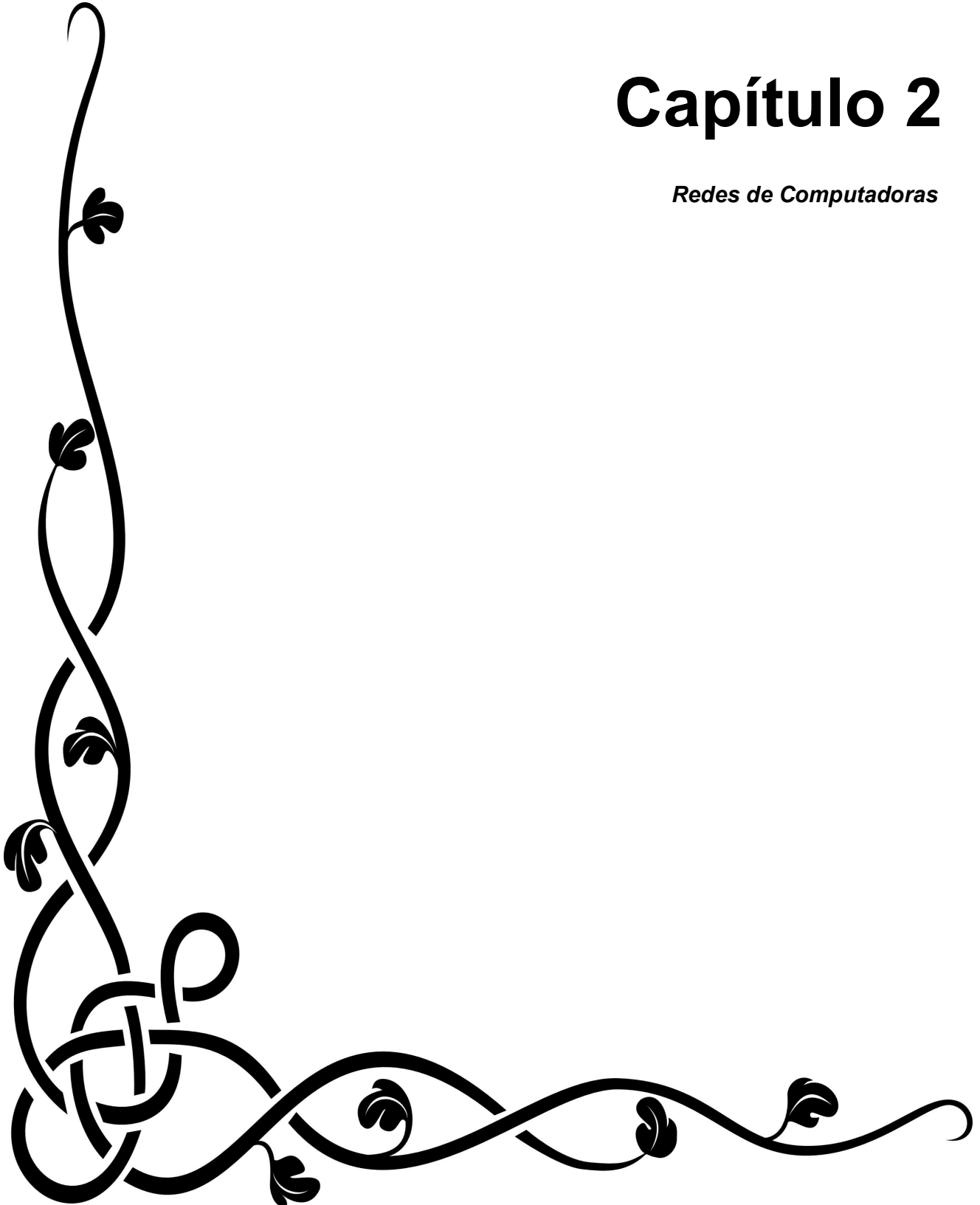
Dedicado al proceso de creación del Sistema Web Matlatl, utilizado por la Secretaría Técnica de Cómputo y Telecomunicaciones del IFUNAM para la Administración de la Red de Datos. Se explican las razones del porqué se decidió usar tecnologías ágiles para el desarrollo de la aplicación así como una breve descripción del *framework Ruby on Rails*. Se aborda el procedimiento de instalación, configuración y desarrollo de las herramientas utilizadas así como de la liberación de la versión final del sistema.

Conclusiones

Finalmente en este apartado se habla de las metas que se cumplieron y de las conclusiones a las que llegué después del proceso de implementación del Sistema Matlatl.

Capítulo 2

Redes de Computadoras



“¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil, nos aporta tan poca felicidad? La respuesta es esta, simplemente porque aún no hemos aprendido a usarla con tino.”
Albert Einstein

Redes de Computadoras

2.1 Conceptos Básicos

El concepto de redes no solo es aplicable al contexto de las computadoras, un conjunto de células o un conjunto de personas también son consideradas una red, pero sin importar la forma en que estén implementadas, tienen el mismo objetivo: la comunicación. Para lograr ésta comunicación es necesario un medio para transmitir la información, y un sistema que nos indique cómo transmitir esa información.

En este capítulo me enfocaré en los términos básicos para entender cómo se conforma una red de computadoras.

2.1.1 Red de Computadoras

Es un conjunto de dos o más computadoras independientes que se encuentran interconectadas entre sí con el propósito de intercambiar información y compartir recursos. Pueden configurarse usando diferentes topologías físicas. El medio para establecer la conexión varía de acuerdo a las necesidades del usuario, puede usarse cable de cobre, fibra óptica y aire (microondas, luz infrarroja y satélites de comunicación).

2.1.2 Clasificación de Redes

Para facilitar el estudio de las redes de computadoras, se han realizado diversas clasificaciones, dos de las más aceptadas son las siguientes:

1. Por tecnología de transmisión.
2. Por su extensión geográfica.

2.1.2.1 Clasificación por tecnología de transmisión

Las tecnologías más utilizadas actualmente son:

Redes de Difusión

Este tipo de redes tienen un solo canal de transmisión que comparten todos los equipos (nodos). El envío de un mensaje (paquete) es recibido por todos los nodos, cada paquete cuenta con un campo de dirección que especifica a quien va dirigido. La máquina que recibe el paquete verifica el campo de dirección y comprueba si esta dirigido a ella, si es así lo procesa, en caso contrario lo ignora.

Al colocar un código especial en el campo de dirección el sistema de difusión permite que todos los nodos procesen el paquete recibido, a este método se le conoce como difusión (broadcasting). Si la transmisión se realiza a un subconjunto de nodos se le llama multidifusión (multicasting).

Redes de Punto a Punto

Las redes punto a punto constan de muchas conexiones de pares de nodos. En este tipo de redes es importante encontrar la ruta correcta. Por eso al realizar una transferencia, es probable que el paquete tenga que visitar uno o más nodos. La transmisión de punto a punto entre emisor y receptor se conoce como unidifusión (unicasting).

En general se recomienda usar difusión cuando se implementa una red dentro de una misma área geográfica, o punto a punto cuando es una red más grande.

2.1.2.2 Clasificación por su extensión geográfica

Otra forma de clasificar las redes es considerando la extensión física que ocupan sus componentes, se dividen en 5 categorías.

Redes de Área Personal (PANs)

Son las redes destinadas para uso personal, la comunicación se realiza entre distintos dispositivos (computadoras con mouse y teclado inalámbrico, impresoras, PDAs, teléfonos celulares apoyados en funciones como Bluetooth, dispositivos de audio, cámaras fotográficas, etc.) que se encuentran cerca.

Redes de Área Local (LANs)

Son las redes de propiedad privada, su extensión comprende desde un edificio hasta un campus. Su mayor aplicación es la interconexión de PC's con estaciones de trabajo en oficinas, fábricas y universidades para compartir recursos (impresoras, faxes) e intercambiar datos y aplicaciones (bases de datos, software, ancho de banda). Hay tres aspectos importantes que distinguen a las LANs de otro tipo de redes:

Extensión física.

Tecnología de transmisión.

Topología.

Redes de Área Metropolitana (MANs)

Redes de comunicación de alta velocidad, permiten la interconexión de varias LANs cercanas geográficamente ofreciendo cobertura en un área más extensa, también proporcionan la integración de múltiples servicios mediante la transmisión de datos, voz y video. El ejemplo más común de una MAN es la televisión por cable.

Redes de Área Amplia (WANs)

Redes diseñadas para satisfacer áreas geográficas tan extensas como países y continentes. Compuesta por un conjunto de máquinas dedicadas a ejecutar aplicaciones de usuarios. A cada usuario se le llama host y estos están conectados por una subred (compañía telefónica o proveedor del servicio de Internet) la cual tiene la función principal de conducir mensajes de un *host* a otro.

Internet

Internet es un conjunto global de diferentes redes de comunicación que utilizan la familia de protocolos TCP/IP. Se encarga de interconectar computadoras y estaciones de trabajo en todo el mundo con la finalidad de intercambiar datos e información.

En la Tabla 2.1 podemos observar la extensión que se recomienda para cada tipo de red.

Distancia entre computadoras	Ubicación	Ejemplo
1 m	Metro cuadrado	Red de Área Personal
10 m	Cuarto	Red de Área Local
100 m	Edificio	
1 Km	Campus	
10 Km	Ciudad	Red de Área Metropolitana
100 Km	País	Red de Área Amplia
1,000 Km	Continente	
10,000 Km	Planeta	Internet

Tabla 2.1 Clasificación de computadoras interconectadas por escala²

2.1.3 Topologías de Red

Cuando hablamos de topología nos referimos a la disposición física de los dispositivos o nodos de una red, para lograr la interconexión entre ellos a través de un medio de comunicación.

Existen cuatro topologías básicas:

1. Topología de Bus
2. Topología de Anillo
3. Topología de Estrella
4. Topología de Malla

A continuación se explica brevemente en qué consiste cada una de estas configuraciones.

² Fuente: Andrews S. Tanenbaum "Redes de Computadoras"

Topología de Bus

Se caracteriza porque todos sus nodos comparten el mismo canal de comunicación a través de un medio multipunto (Figura 2.1). La información que se transmite se propaga por todo el bus y llega a todos los nodos, cada nodo es capaz de reconocer cuando la información está destinada a él. Tiene un ancho de banda de 10 Mbps.

Ventajas

- Es de fácil implementación y crecimiento.
- Económica

Desventajas

- La longitud máxima del bus se encuentra entre 185 y 200 m.
- Si hay problemas con el cable o llega a romperse, toda la red dejará de funcionar.
- Se requieren terminadores en cada extremo del bus.
- Hay colisiones cuando varios nodos quieren transmitir al mismo tiempo, ya que solo hay un canal de transmisión.

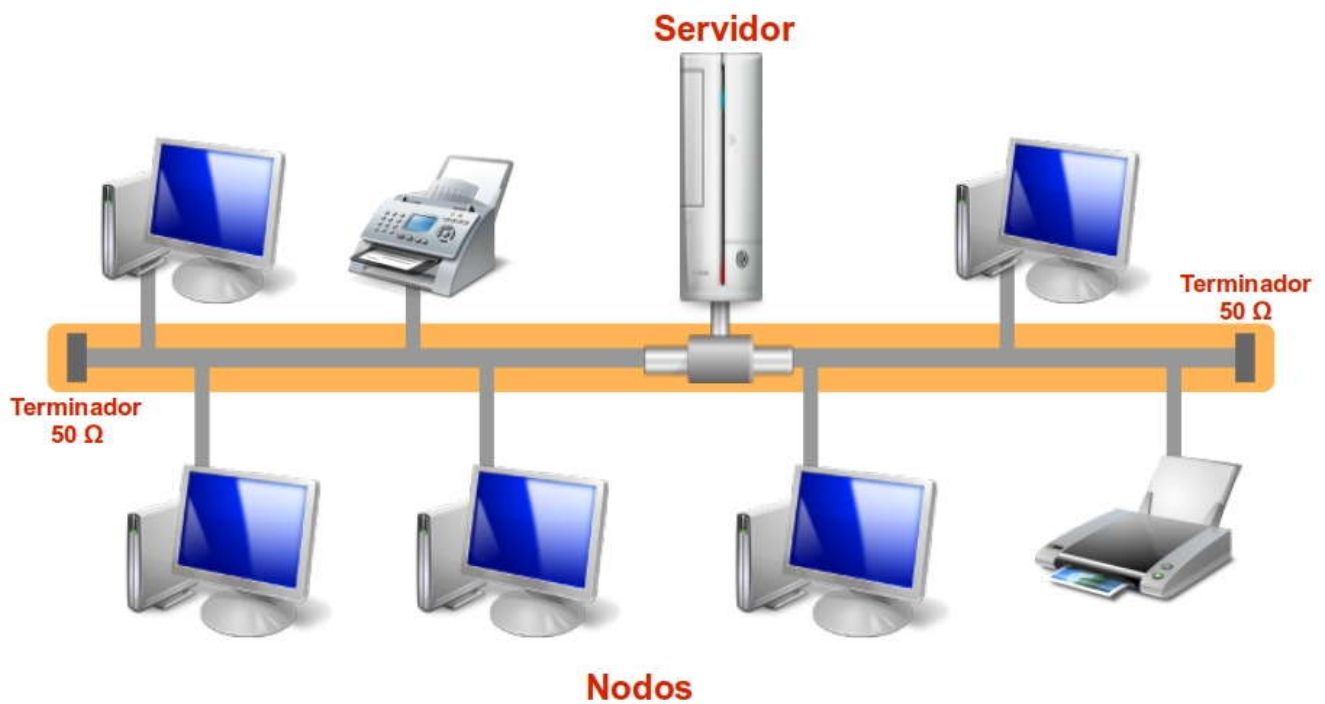


Figura 2.1 Topología de Bus

Topología de Anillo

Los nodos de la red están interconectados entre si mediante un anillo cerrado (Figura 2.2). Cada nodo consta de un receptor y un transmisor que hacen la función de repetidor. El conjunto de repetidores están unidos mediante enlaces punto a punto.

La información se transmite en un solo sentido, el acceso al medio de la red se realiza por medio de un *token* o testigo. Si un nodo quiere enviar información crea un *token* y lo envía a través del anillo pasando por todos los nodos hasta llegar al nodo destino, quien enviará un mensaje al nodo origen para indicar que recibió la información. Tiene un ancho de banda de 4 a 16 Mbps.

Ventajas

- Es fácil aumentar o disminuir la cantidad de nodos.
- Se puede implementar un anillo doble para transmitir información en dos sentidos.

Desventajas

- Es difícil localizar los problemas en el canal de comunicación lo que puede paralizar o bloquear a toda la red.
- Entre mayor sea el flujo de información, la velocidad de respuesta de la red será menor.

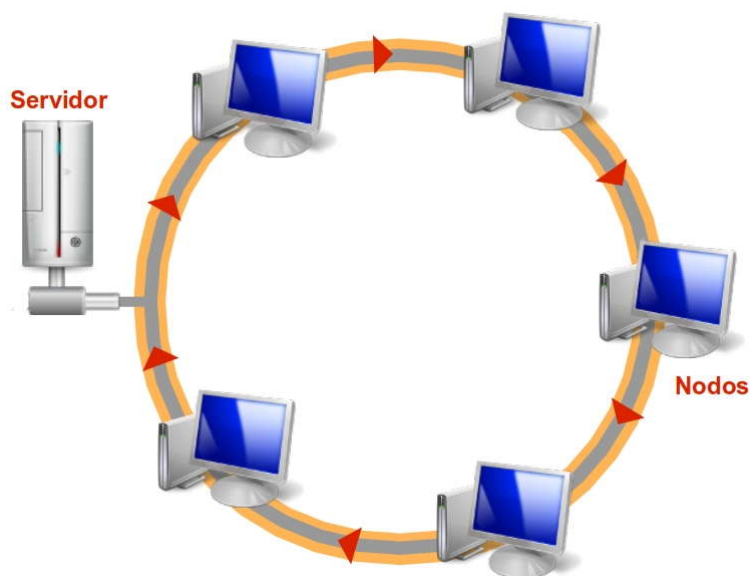


Figura 2.2 Topología de Anillo

Topología de Estrella

En esta red los nodos están directamente conectados a un nodo central (*hub*, *switch* o *router*) común mediante un enlace punto a punto dedicado y un puerto de entrada/salida para transmisión y recepción. Se puede observar una representación en la Figura 2.3.

Si un nodo quiere enviar información a otro equipo, primero debe enviarla al nodo central y éste se encargará de enviara al nodo destino. En la actualidad el ancho de banda puede ser hasta de 10Gbps.

Ventajas

- Es fácil agregar nodos y reconfigurar la red.
- Si un nodo se desconecta o tiene un fallo, no afecta en el funcionamiento de toda la red.

Desventajas

- La red se desconectará si el nodo central deja de funcionar.
- Es más costosa debido al equipo empleado como nodo central y al uso de mayor cantidad de cable.

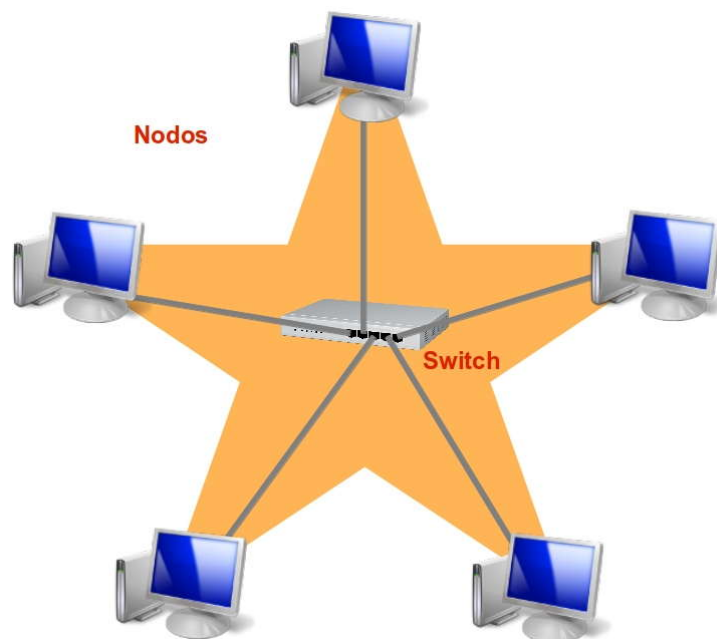


Figura 2.3 Topología de Estrella

Topología de Malla

En este tipo de topología cada nodo de la red está conectado a todos los demás nodos mediante enlaces punto a punto. Esta configuración facilita el envío de información ya que se puede enviar por diferentes rutas. En la Figura 2.4 se ilustra un ejemplo.

Ventajas

- Es tolerante a fallas gracias a la redundancia de sus conexiones.
- Si un cable se rompe no inhabilita a toda la red.
- Se pueden agregar nuevos nodos sin afectar a los que ya existen.

Desventajas

- Su implementación y configuración son complicadas debido a la gran cantidad de conexiones.
- Es muy cara ya que requiere más cable que cualquier otra topología.

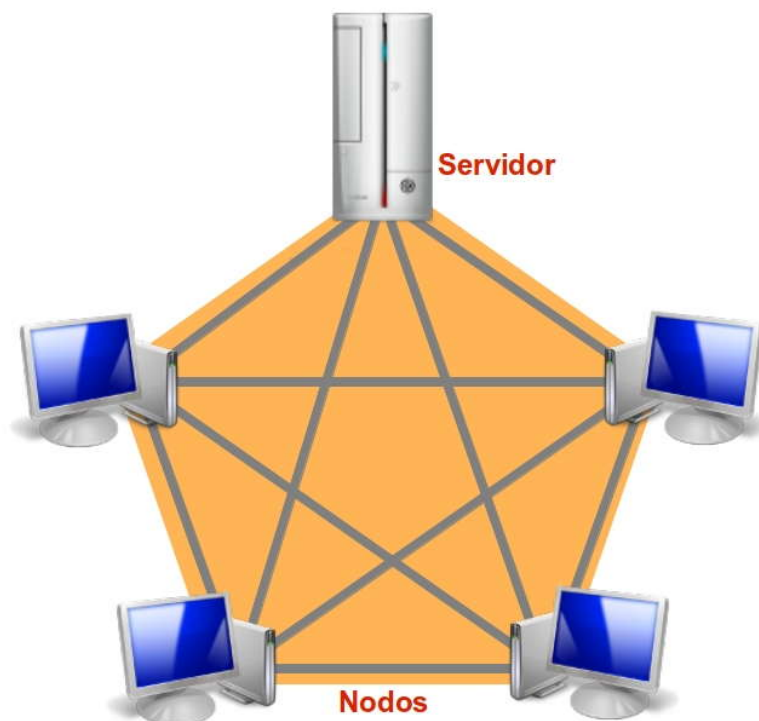


Figura 2.4 Topología de Malla

2.1.4 Modelos de Referencia

Debido al crecimiento acelerado de las telecomunicaciones y la informática los fabricantes de equipos comenzaron a tener problemas, ya que lograr una comunicación con equipos de diferentes fabricantes era muy complicado o casi imposible debido a la cantidad de especificaciones e implementaciones en la tecnología de red. La necesidad de crear estándares para lograr una interconexión entre redes informáticas se hizo evidente y en 1977 la ISO (Organización Internacional para la Estandarización) realizó una investigación para tener un conjunto de reglas aplicables en todas las redes, así creó el modelo OSI (Interconexión de Sistemas Abiertos) el cual fue adoptado como estándar internacional hasta 1984.

Por otro lado, el modelo TCP/IP fue desarrollado por los científicos Vinton Cerf y Robert Khan en 1974 durante su participación en el proyecto ARPANET para lograr conectar múltiples redes independientemente del hardware con el que se implementaran, crearon una arquitectura de protocolos de comunicación que facilitaba el intercambio de información y que aún en la actualidad sigue siendo la más utilizada.

2.1.4.1 Modelo OSI (Interconexión de Sistemas Abiertos)

El modelo OSI, ilustrado en la Figura 2.5, está basado en la idea de que el proceso de comunicación entre dos nodos puede dividirse en 7 capas:

La capa física

En esta capa se encuentran todos los medios físicos que intervienen en la comunicación, además se encarga de controlar los puertos ó circuitos de los dispositivos utilizados para garantizar la transmisión de información. Las características más importantes a considerar son las propiedades físicas y especificaciones de los conectores, cómo representar un bit y cómo interactúa el medio físico con el medio de transmisión.

La capa de enlace de datos

El trabajo principal de esta capa es hacer que el enlace físico sea fiable, y que la información que se transmite llegue sin errores a su destino, además determina cuando y quien puede acceder al medio.

Los datos son segmentados en tramas y se emplea el *direccionamiento físico*³ para garantizar que llegarán a su destino. Esta capa se caracteriza por tener un control de flujo y un manejo de errores.

La capa de red

El objetivo de ésta capa es hacer que los datos lleguen de un origen a un destino a través de una red de comunicación. Tanto el emisor como el receptor pueden estar configurados de forma diferente (pueden usar distintos protocolos), por lo que ésta capa resuelve los problemas de compatibilidad. Por medio del *direccionamiento lógico*⁴ se determina la ruta de los datos y su receptor final.

Cuenta con un control de la congestión de red, elemento necesario para saber cuando la saturación de un nodo puede llegar a bloquear la red.

La capa de transporte

Esta capa es el corazón de la comunicación, ya que aísla a las capas superiores de los cambios en la tecnología del hardware y además garantiza la entrega de los datos sin errores, en orden y sin pérdidas.

Recibe los datos de las capas superiores y los segmenta para transmitirlos a la capa de red y los reensambla en el nodo destino. En esta capa se determina el tipo de servicio que se proporciona a la capa de sesión para ofrecer un servicio de calidad al usuario final.

La capa de sesión

Esta capa permite que se establezcan sesiones entre diferentes máquinas. Inicia, mantiene y termina la comunicación. Los principales servicios que ofrece es la de control de diálogo, la *sincronización*⁵ y la administración de concurrencia.

³ Dirección física que seguirán las tramas, es única ya que identifica al fabricante y al hardware de red.

⁴ Dirección IP que identifica a un equipo de cómputo en una red interna o externa.

⁵ Establecer puntos de referencia para recuperar una transferencia en caso de que sea interrumpida.

La capa de presentación

Esta capa se encarga de la forma en la que se presenta la información. Está relacionada con el significado, la interpretación y la coordinación de los datos. Tiene la función de que la información enviada por el nodo inicial sea recibida por el nodo final.

La capa de aplicación

Es la capa más cercana al usuario y ofrece la posibilidad de acceder a los servicios de las otras capas. Se define la interfaz para la comunicación entre el software y la red. Los usuarios no tienen trato directo con ésta capa, son los programas los que interactúan con ella. Algunos ejemplos de aplicaciones de uso común son la transferencia de archivos, el correo electrónico y el acceso a computadoras por medio de terminales.

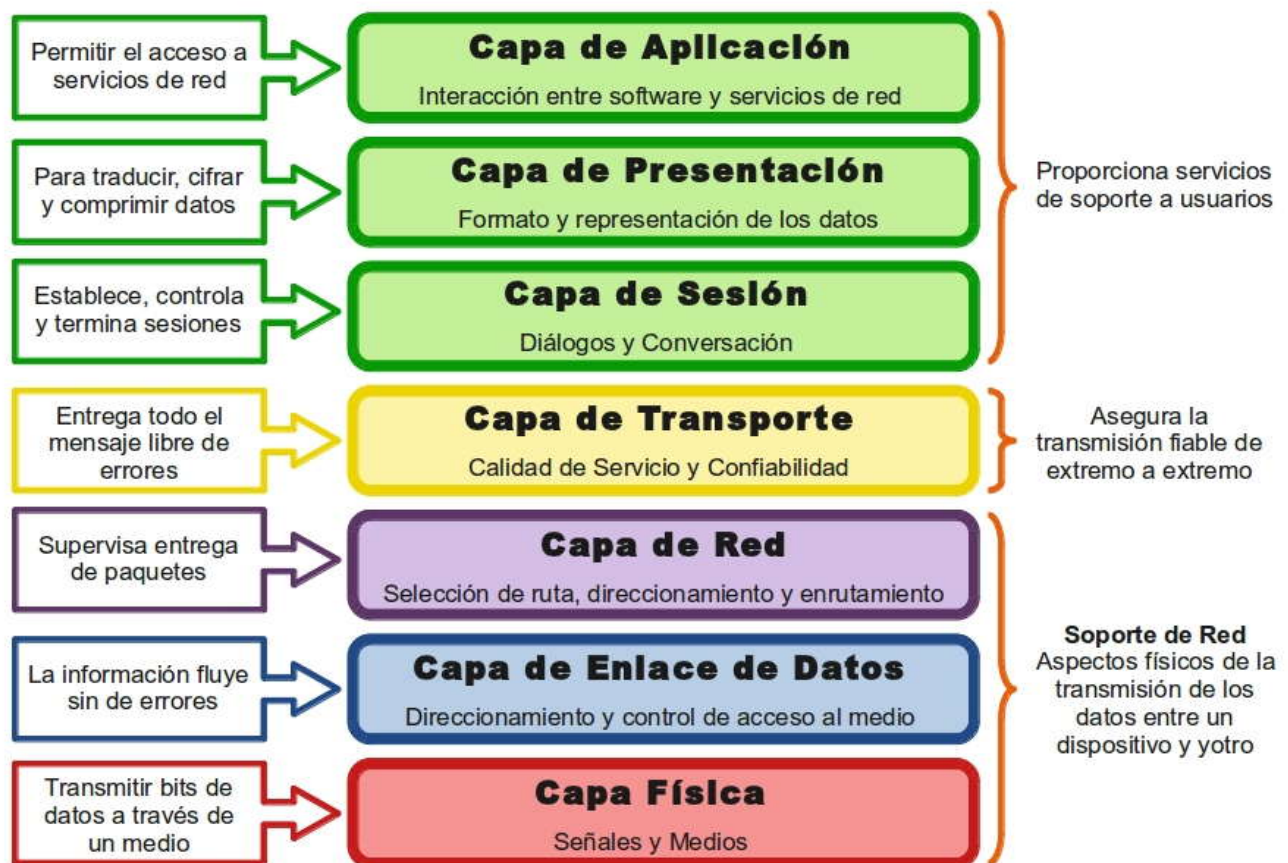


Figura 2.5 Las capas del modelo de referencia OSI

2.1.4.2 Modelo TCP/IP

Llamado así por los dos protocolos más importantes que lo componen, el TCP (Protocolo de Control de Transmisión) y el IP (Protocolo de Internet), pero en realidad está compuesto por un grupo extenso de protocolos. Podemos observar una representación del modelo en la Figura 2.6. Está dividido en 4 capas:

La capa de host a red

En este nivel el *host* establece una conexión con los medios físicos de la red mediante el protocolo TCP/IP para enviar paquetes IP. Éste protocolo puede variar de *host* a *host* ya que no está definido.

La capa de interred

La capa de interred es la que mantiene unida a toda la arquitectura y se considera similar en funcionalidad a la capa de red del modelo OSI. En ésta capa se permite el traslado independiente y se gestiona la entrega de los paquetes IP a su destino. El protocolo principal que se define y trabaja en ésta capa es el IP. Un aspecto muy importante a considerar es el enrutamiento, ya que tiene como propósito evitar la congestión en la red.

La capa de transporte

Esta capa permite que un *host* origen establezca una conversación con el *host* destino (similar al comportamiento de la capa de transporte en el modelo OSI). Se definen dos protocolos de transporte importante, el primero de ellos es el TCP, el cual es confiable y está orientado a la conexión por lo que el mensaje debe entregarse sin errores, el flujo de bytes se segmenta en pequeños mensajes y los pasa a la capa de interred, en el *host* destino el protocolo TCP se encarga de reensamblar los mensajes. El TCP también cuenta con un control de flujo.

El segundo protocolo es el UDP (Protocolo de Datagrama de Usuario), no es confiable y no está orientado a la conexión. Tiene un amplio uso de consultas únicas de tipo cliente-servidor y aplicaciones de entrega puntual tales como voz y video.

La capa de aplicación

En esta capa se realiza la comunicación entre programas de red usando protocolos de alto nivel, además de manejar los aspectos de representación, codificación y control del diálogo. Algunos ejemplo de las aplicaciones más utilizadas por los usuarios con sus respectivos protocolos son:

- La terminal remota (SSH - Interprete de Comandos Seguro).
- Transferencia de archivos (FTP - Protocolo de Transferencia de Archivos).
- El correo electrónico (SMTP - Protocolo Simple de Transferencia de Correo).
- La administración de red (SNMP – Protocolo Simple de Administración de Red).
- Y la resolución de nombres (DNS – Sistema de Nombres de Dominio).

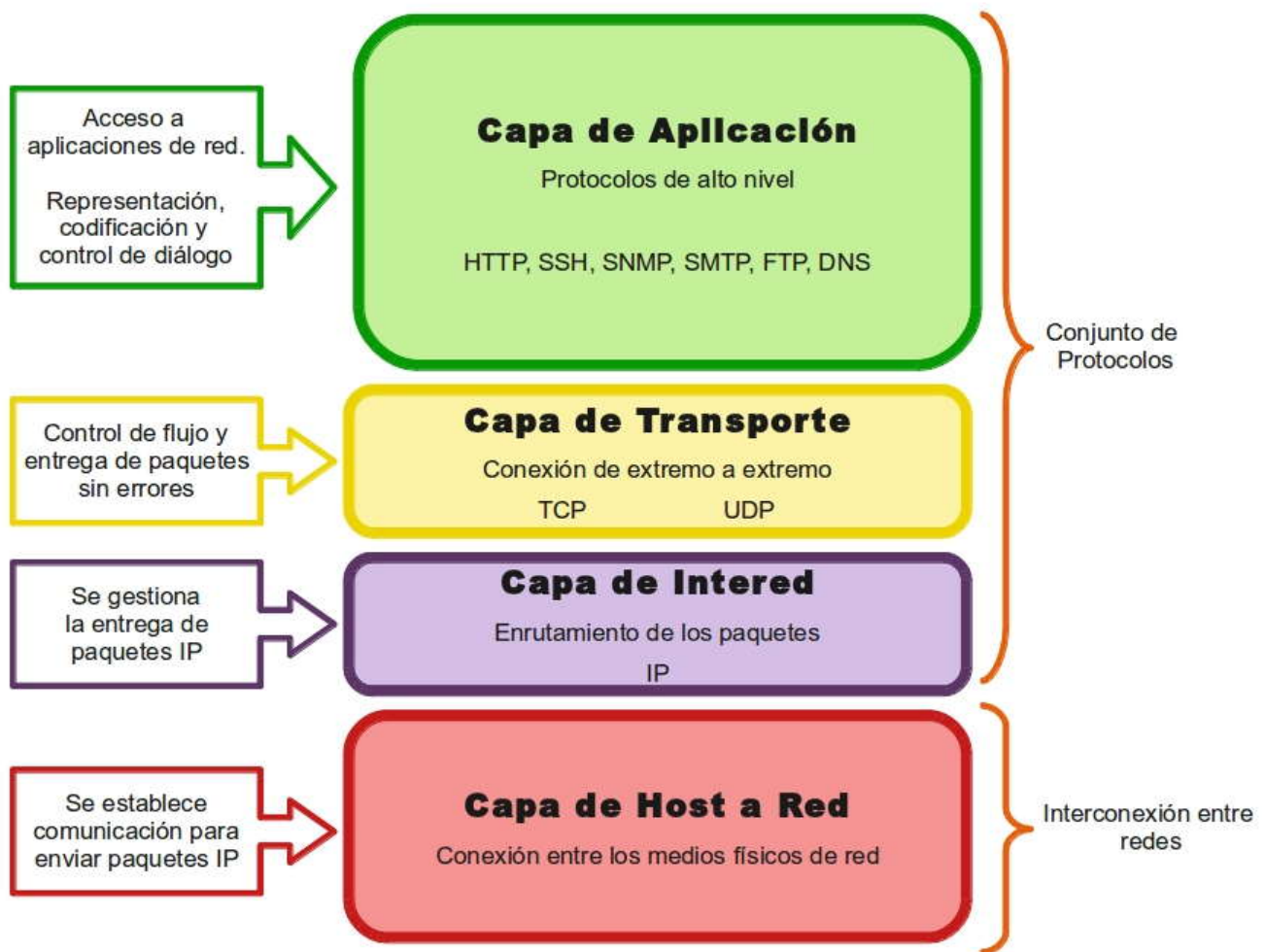


Figura 2.6 Las capas del modelo de referencia TCP/IP

2.2 Control de Acceso al Medio (MAC – Media Access Control)

El administrador de la red de datos del Instituto de Física solicitó un registro de las direcciones MAC's de los usuarios de la red, por lo que es importante explicar en qué consisten dichas direcciones físicas. A continuación se da una breve descripción.

La dirección MAC es un identificador único compuesto de 12 dígitos hexadecimales (48 bits de longitud), éste es asignado y escrito de forma binaria directamente en los dispositivos de red durante su fabricación con el objetivo de lograr el reconocimiento de cada dispositivo a nivel mundial.

Por convención, el formato de escritura que sigue la dirección MAC puede ser de las siguientes dos formas:

- MM:MM:MM:SS:SS:SS
- MM-MM-MM-SS-SS-SS

Los primeros 6 dígitos (24 bits) sirven para identificar al fabricante del dispositivo, éstos son asignados por la **OUI (Organizationally Unique Identifier - Identificador Único Organizacional)**, quien a su vez compra bloques, con sus posibles derivados de direcciones, administrados por la **IEEE (Institute of Electrical and Electronics Engineers – Instituto de Ingeniería Eléctrica y Electrónica)** para identificar mundialmente a cada empresa u organización y poder asegurar la singularidad de cada dispositivo. Los últimos 6 dígitos (24 bits) son asignados por el fabricante de forma secuencial durante su fabricación en el momento del Quemado de las Direcciones (**BIA – Burned-In Address**).

Cada tarjeta de red cuenta con un conjunto de chips encargados de administrar la comunicación con los *medios físicos*⁶ controlando las *señales de comunicación*⁷ de dichos medios.

⁶ Cable, fibra óptica, aire, etc.

⁷ Electricidad, luz o frecuencia de radio

La MAC se encarga de proporcionar los medios para acceder al medio físico usado en la comunicación. Su principal objetivo es hacer que la información fluya libre de errores entre dos máquinas conectadas.

2.3 Dirección IP

Una dirección IP es un identificador numérico único que permite referirse lógicamente a una interfaz de red.

*Cada host y enrutador de Internet tiene una dirección IP, que codifica su número de red y su número de host. La combinación es única: no hay dos máquinas que tengan la misma dirección IP.*⁸

El objetivo de estas direcciones es permitir la comunicación entre hosts, ya que al ser irrepetibles es fácil identificar a cada uno de ellos dentro de una red.

En la actualidad existen 2 tipos de direcciones, a continuación se realiza una breve descripción de ellas.

Direcciones IPv4

Las direcciones IPv4 se expresan con una notación decimal y tienen una longitud de 32 bits (2^{32} direcciones), se representan por 4 octetos separados por un "." y pueden estar comprendidos entre 0 a 255.

Ejemplo de IPv4: 132.248.7.15

Direcciones IPv6

Debido al incremento del uso de Internet en el mundo, la gran demanda de direcciones IP provocó que las direcciones IPv4 fueran insuficientes, por lo que se creó una nueva generación para otorgar una cantidad significativa de direcciones.

⁸ Fuente: Andrews S. Tanenbaum "Redes de Computadoras"

Las direcciones IPv6 tiene una expresión hexadecimal, con una longitud de 128 bits (2^{128} direcciones) representados por 8 grupos de 16 bits y separados por ":", pueden estar comprendidos entre 0000 a FFFF

Ejemplo de IPv6: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Estas direcciones no son sensibles a mayúsculas/minúsculas. Están conformadas por 2 partes lógicas, la primera parte es un prefijo con los 64 bits más significativos, la segunda parte con los 64 bits menos significativos se genera automáticamente a partir de las direcciones mac.

2.4 El Modelo Cliente/Servidor

El modelo Cliente/Servidor es una arquitectura distribuida donde las transacciones se dividen en procesos independientes que cooperan entre si para intercambiar servicios, recursos o información. El cliente inicia el diálogo solicitando un servicio y el servidor envía uno o más mensajes respondiendo a la petición.

El servidor presenta a los clientes una interfaz bien definida, los clientes solo conocen la interfaz externa del servidor y no dependen de su ubicación física, del equipo físico que lo conforma ni del sistema operativo con el que trabaja.

Servidor

Un servidor es una máquina encargada de proporcionar servicios dentro de una red, con el propósito de compartir sus recursos con usuarios a los que se les conoce como clientes. Hay diversos tipos de servidores que satisfacen las necesidades de los usuarios, algunos ejemplos más conocidos son los servidores de correo, los servidores web, los servidores de archivos, los servidores de aplicaciones y los servidores de impresión.

Cliente

Un cliente es una máquina o una aplicación que se utiliza para acceder a los servicios que

ofrece un servidor. Es el proceso que permite al usuario formular una solicitud al servidor para consultar datos externos, interactuar con otros usuarios, compartir su información o para utilizar recursos de los que no dispone en su máquina local.

2.5 Protocolo SSH (Protocolo de conexiones seguras)

El protocolo SSH fue creado por el investigador finlandés Tatu Ylönen en 1995 después de que su red sufrió un ataque de *sniffing*⁹. Su implementación fue liberada como *freeware*¹⁰ y tuvo tanto éxito que llegó a reemplazar a los protocolos *telnet*, *rlogin* y *rsh*, ya que no garantizaban seguridad y confidencialidad. Este protocolo opera en la capa de aplicación, facilita una comunicación segura entre dos sistemas usando una arquitectura cliente/servidor y realizando la conexión de forma remota, es decir, no es necesario estar físicamente frente al equipo, se controla a través de una interfaz de línea de comandos. El intercambio de datos se realiza usando un canal seguro entre dos dispositivos de red. SSH cifra toda la información (incluyendo las contraseñas) para evitar que terceros la intercepten ya que solo verán un conjunto de símbolos ininteligibles.



Figura 2.7 Ejemplo de conexión usando el protocolo SSH

Tuvo mucho éxito gracias a que sus principales aplicaciones son el logueo, la autenticación

⁹ Interceptar la información que circula por una red informática.

¹⁰ Distribución libre o gratuita de programas.

de usuarios, la transferencia de archivos, la ejecución de comandos y consideradas como las más importantes, la codificación del mensaje, la autenticación de equipos y la integridad del mensaje.

En nuestro caso usaremos la distribución libre OpenSSH, la cual es desarrollada por el proyecto OpenBSD, cuyo código puede usarse libremente bajo la licencia BSD.

Los programas integrados a OpenSSH son:

- ssh: programa cliente que sirve para loguearse a una máquina remota y nos permite ejecutar comandos.
- scp: nos permite realizar copias seguras de documentos entre *hosts*.
- sftp: programa de transferencia segura de archivos, muy similar a ftp, el cual realiza las operaciones de transporte de datos sobre un canal cifrado.
- sshd: es el demonio del programa ssh y se ejecuta constantemente en el servidor, esperando que soliciten una conexión.
- ssh-keygen: se encarga de generar, controlar y convertir las llaves autenticadas.

2.6 Protocolos de Comunicación

La configuración, el funcionamiento y el contenido de las capas puede variar de red en red por lo que para lograr una comunicación entre equipos, que pueden ser de fabricantes diferentes y usar diversos sistemas operativos, se crearon los protocolos de comunicación, que pueden definirse como el conjunto de reglas y estándares que especifican como debe realizarse la comunicación y el intercambio de datos entre dos sistemas.

2.6.1 Protocolo TCP (Protocolo de Control de Transmisión)

El protocolo de comunicación TCP fue desarrollado basándose en los conceptos descritos en 1974 por los científicos Vinton Cerf y Robert Khan, se diseñó para permitir una conexión de extremo a extremo entre redes fiables y no fiables y proveer las funciones necesarias para una aplicación de transporte de flujo de bytes sin errores.

El protocolo TCP fragmenta la información en paquetes, llamados segmento TCP, a los que les añade un encabezado con formato fijo de 20 bytes seguido de 0 o más bytes de datos. Este protocolo decide el tamaño de los segmentos, cuándo enviar los datos y cuándo bloquear la información.

Para lograr que la comunicación entre los *hosts* sea confiable el protocolo realiza operaciones específicas con mecanismos de diversas áreas, estos mecanismos y sus funciones principales en cada una de ellas son:

Transferencia básica de datos: TCP es capaz de transferir un flujo continuo de octetos empaquetados en segmentos para su transmisión a través de una red. Si el usuario especifica un bloque de datos como urgente TCP marcará el final de dicho bloque como un puntero de urgente y lo enviará en el flujo de datos ordinario.

Fiabilidad: TCP debe recuperar la información corrupta, perdida, duplicada o desordenada, para lograrlo asigna una secuencia a cada octeto que transmite y exige un acuse de recibo (ACK) del receptor, si no lo recibe en cierto tiempo los datos se retransmiten. El receptor usa los números de secuencia para ordenar los segmentos y así evitar la duplicidad. Para descartar los segmentos dañados se añade una suma de control (checksum) a cada segmento transmitido que se comprueba en el receptor. TCP es capaz de recuperarse cuando hay errores en la comunicación.

Flujo de control: Con ayuda de TCP el receptor puede controlar la cantidad de datos enviados por el emisor, en cada ACK envía una “ventana” que indica la cantidad de octetos que el emisor tiene permitido transmitir antes de recibir el próximo permiso.

Multiplexamiento: Para permitir el uso simultáneo de los procesos de comunicación en un solo *host*, el módulo TCP proporciona una serie de direcciones o puertos dentro de cada *host*, que combinada con la dirección IP y la dirección local del host conforman lo que conocemos como *socket*¹¹, lo que permite identificar de forma única a cada conexión y de esta forma realizar múltiples conexiones. La asignación de los puertos a cada proceso se controla independientemente en cada *host*.

¹¹ Interfaz de conexión entre un proceso cliente y un proceso servidor. Es el punto final de comunicación.

Conexiones: TCP debe realizar un informe de estado para cada flujo de datos al iniciar una conexión. Una conexión se establece de forma única entre dos sockets que corresponden en ambos extremos. Cuando dos procesos desean comunicarse, sus módulos TCP establecen la conexión y la cierran al completarse. Para evitar inicializaciones erróneas de conexiones en sistemas de comunicación no fiables, se usa un mecanismo de números de secuencia basados en tiempos de reloj.

Prioridad y Seguridad: Cada usuario tiene la opción de indicar el nivel de prioridad y seguridad que desea para sus comunicaciones. Cuando no se especifican estas características se usan valores por defecto.

2.6.2 Protocolo IP (Protocolo de Internet)

El protocolo de Internet IP fue diseñado en un principio para la interconexión de redes. Proporciona los medios necesarios para transportar los bloques de datos (datagramas) de un origen a un destino, sin importar que los *hosts* se encuentren en la misma red, o haya otras redes entre ellos.

El datagrama IP se conforma de dos secciones, la primera es un encabezado que cuenta con una parte fija de 20 bytes y una parte opcional de longitud variable, la segunda es una parte de texto. El protocolo de internet trata a los datagramas como una entidad independiente sin ninguna relación con otro datagrama.

Este protocolo proporciona un servicio de mejor esfuerzo, es decir que hará lo mejor posible pero no garantiza la fiabilidad de datos entre los extremos, tampoco cuenta con un mecanismo de control de flujo por lo que no puede determinar si los datos llegaron a su destino, si fueron duplicados o si están dañados.

El protocolo de internet implementa dos funciones básicas:

Direccionamiento: Se hace uso de las direcciones que se encuentran en la cabecera para poder transmitir los datagramas hacia su destino. La elección de un camino para la transmisión se llama encaminamiento.

El encaminamiento se trata de la búsqueda de la mejor ruta posible mediante el uso de una tabla de encaminamiento en la que se especifica para cada posible red destino el siguiente dispositivo de encaminamiento al que se enviará el datagrama.

Fragmentación: Cuando es necesario el módulo IP hace uso de la cabecera para fragmentar o segmentar los datagramas en unidades más pequeñas para transmitirlos a redes que limitan el tamaño de los paquetes. Al llegar a su destino los datagramas son reensamblados.

Cada *host* involucrado en una red cuenta con un módulo IP, estos módulos tienen reglas similares que se encargan de interpretar las direcciones en las cabeceras y así ser capaces de fragmentar y reensamblar los datagramas. También tienen procedimientos para tomar decisiones de encaminamiento y otras funciones.

El protocolo de internet usa cuatro mecanismos esenciales para poder ofrecer sus servicios:

Tipo de servicio: Utilizado para indicar la calidad del servicio requerido, se basa en los parámetros que presentan los servicios seleccionados en las redes que conforman la Internet.

Tiempo de vida: Es el límite superior del periodo de vida de un datagrama. Esta marca es especificada por el remitente y se implementa un contador de saltos que reducirá el tiempo de vida cada vez que pase a través de un dispositivo de encaminamiento, si el tiempo de vida llega a cero antes de que el datagrama llegue a su destino, el datagrama se descarta.

Opciones: Proporcionan los recursos necesarios para marcas de tiempo, seguridad y encaminamiento especial.

Suma de control de cabecera: Sirve para proporcionar una verificación de que la información ha sido transmitida correctamente al procesar el datagrama. Si los datos contienen errores, al realizarse la suma de control de cabecera el datagrama será descartado ya que se encontrará una falla.

2.7 Servidor DHCP

El Protocolo de Configuración de Host Dinámico (DHCP - Dynamic Host Configuration Protocol) es utilizado para asignar automáticamente parámetros a un equipo cliente dentro de una red TCP/IP. El servidor DHCP centralizado proporcionará la configuración de red, que consiste en la dirección IP, la *máscara de subred*¹², la *puerta de enlace*¹³ y los *servidores DNS*¹⁴, cuando un cliente DHCP se comunique con él.

Este protocolo de red está basado en una arquitectura cliente/servidor en el que generalmente el servidor almacena en memoria una lista de direcciones IP en la que buscará y asignará la dirección disponible cuando el cliente la solicite.

El servidor DHCP puede configurarse para asignar direcciones IP en tres formas diferentes:

Asignación Automática: Designará permanentemente una dirección arbitraria al cliente.

Asignación Dinámica: Designará una dirección durante un determinado tiempo.

Asignación Estática: Designará una dirección reservada en función de la dirección física de la tarjeta de red del cliente.

2.7.1 Funcionamiento de un servidor DHCP

La interacción entre un cliente y un servidor se lleva a cabo mediante el uso de un sistema de comunicación conformado por 8 tipos de mensajes de difusión:

DHCPDISCOVER: Este mensaje es emitido por un cliente para localizar a los servidores disponibles.

DHCPOFFER: Es el mensaje de respuesta que envían los servidores activos ofreciendo al cliente sus parámetros de configuración.

DHCPREQUEST: El cliente elige un servidor y responde solicitando los parámetros de red, esta acción le informa a los otros servidores que rechaza sus ofertas.

DHCPACK: Mensaje que envía el servidor para confirmar los parámetros incluyendo la dirección de red.

DHCPNACK: Este mensaje es enviado al cliente por el servidor para informarle que la

¹² Conjunto de bits que permiten identificar la red y el host en una dirección IP.

¹³ Dispositivo para interconectar redes, realiza la traducción necesaria cuando se usan protocolos diferentes.

¹⁴ Encargados de traducir los nombres de dominio en sus respectivas direcciones IP.

dirección de red que solicita no es válida para la subred en la que se encuentra o que su *arrendamiento*¹⁵ ha expirado.

DHCPDECLINE: El cliente envía este mensaje al servidor para informarle que esta usando la dirección.

DHCPRELEASE: Este mensaje es enviado por el cliente para informarle al servidor que no usará más la dirección IP y da por terminado su arriendo.

DHCPINFORM: Mensaje enviado por el cliente cuando ya está configurado, únicamente le pregunta al servidor sobre los parámetros de configuración local.

Cabe destacar que para lograr una comunicación entre cliente y servidor no son suficientes los mensajes anteriormente descritos, es necesario seguir una secuencia de eventos para iniciar el servicio DHCP:

El cliente DHCP emite un mensaje de descubrimiento (DHCPDISCOVER), en este mensaje se incluye la dirección MAC (Control de Acceso al Medio) que sirve como identificador único del cliente.

Los servidores que puedan brindar este servicio responden a la petición con el mensaje DHCP OFFER ofreciendo al cliente una dirección IP basándose en la interfaz de red y en la subred de donde proviene la petición, también se debe tener en consideración si la dirección IP está asociada específicamente a una dirección MAC o se le puede asignar una de las direcciones disponibles. Mientras esté en uso, la dirección asignada es reservada temporalmente por el servidor para evitar ofrecerla por segunda ocasión.

El cliente selecciona la mejor oferta y para confirmar los datos envía una respuesta (DHCPREQUEST) solicitando los parámetros de configuración del servidor que eligió, esta emisión le indica a los servidores restantes que el cliente ha seleccionado un servidor y pueden cancelar las reservaciones de las direcciones de red.

El servidor envía al cliente un mensaje de reconocimiento (DHCPACK) que contiene los parámetros de configuración de red y almacena en una base de datos la información del cliente y la IP asignada.

El cliente usa la información enviada por el servidor para configurar su interfaz de red. También debe supervisar el tiempo que usará la dirección IP, al transcurrir un tiempo

¹⁵ Técnica utilizada para asignar direcciones IP, el servidor DHCP emite un préstamo por un tiempo determinado, al terminar el cliente debe solicitar una renovación o dejar de usar la dirección.

determinado éste envía al servidor un nuevo mensaje para renovar y aumentar el tiempo de permiso.

2.8 Sistema Operativo

El sistema operativo es un conjunto de programas lógicos encargados de administrar los dispositivos de hardware y los recursos de una computadora mediante la ejecución de procesos de control. Realiza operaciones ocultas de bajo nivel por lo que el usuario no se da cuenta de los complejos procedimientos que se ejecutan, además ofrece a los programas una interfaz sencilla para que el usuario logre comunicarse con el *hardware*. Su principal tarea es la repartición ordenada y controlado de los recursos entre el *hardware* y el *software* que compiten por obtenerlos.

La administración de los recursos del sistema se realiza apoyándose en dos tipos de comportamientos:

Multiplexaje en el tiempo: Este comportamiento se caracteriza por el hecho de que los programas o los usuarios toman turnos para poder hacer uso de los recursos. El sistema operativo decide de quien es el turno siguiente y el tiempo óptimo de uso.

Multiplexaje en el espacio: En este caso se realiza una repartición del recurso para cada cliente. El ejemplo más claro es cuando la memoria principal es asignada a diferentes programas en ejecución en lugar concentrarla en una sola tarea.

2.8.1 GNU/Linux

En 1971 Richard Stallman, estudiante del primer año de física en la Universidad de Harvard, se convirtió en un especialista de informática gracias a su trabajo en el laboratorio de Inteligencia Artificial del MIT. En ese entonces el código fuente de los programas se compartían libremente ya que en esa época no existía el concepto de *software* libre o *software* propietario.

Conforme pasaba el tiempo la industria del *software* se dio cuenta de que era un negocio rentable lo que originó que a principios de los 80's el *software* ya no fuera cooperativo si no propietario, las compañías obligaron a los usuarios a pagar licencias y a firmar acuerdos de "no revelar" para poder usarlo, y más importante aún, no se podía tener acceso al código fuente.

Esta situación fue la que orilló a Stallman a buscar otras alternativas, ya que consideraba al *software* propietario como antisocial, abandona el MIT y en 1983 inicia el proyecto GNU, agregando: "Cualquier usuario debe poder modificar un programa para ajustarlo a sus necesidades y cualquier usuario debe poder compartir el software, porque ayudarnos unos a otros es la base de la sociedad".

Este proyecto consistía en desarrollar un conjunto de aplicaciones y un sistema operativo que fueran completamente libres y compatibles con UNIX para su portabilidad. El nombre que se eligió fue GNU que significa GNU's Not Unix (GNU no es UNIX). Su primer logro fue el compilador de c llamado gcc, el cual sigue siendo de vital importancia para el desarrollo de *software* UNIX.

A principios de los 90's el sistema estaba casi terminado, solo faltaba un *kernel*¹⁶, se inició el desarrollo de GNU Hurd, sin embargo hasta el día de hoy no está listo. Afortunadamente otro *kernel* estaba disponible: Linux.

En 1991 Linus Torvalds era un estudiante de segundo año de informática en la Universidad de Helsinki, se caracterizaba por ser un programador autodidacta. El 25 de agosto del mismo año, a los 21 años de edad, envía un correo al grupo de noticias de *MINIX*¹⁷:

"Estoy haciendo un sistema operativo (gratis, solo un hobby, no será nada grande ni profesional como GNU) para clones AT 386(486). Llevo en ello desde abril y está empezando a estar listo. Me gustaría saber su opinión sobre las cosas que les gustan o disgustan en minix, ya que mi SO tiene algún parecido con él [...] Actualmente

¹⁶ Software fundamental en un sistema operativo ya que es el encargado de administrar los recursos y dispositivos de la máquina mediante llamadas al sistema, también se encarga de gestionar la comunicación entre los programas y el hardware.

¹⁷ Sistema Operativo desarrollado por el académico Andrews Tanenbaum para enseñar a sus alumnos el funcionamiento interno de un sistema operativo real, fue muy importante ya que su código fuente estaba disponible.

he portado bash(1.08) y gcc(1.40), y parece que las cosas funcionan. Esto implica que tendré algo práctico dentro de unos meses..."

Para mediados de septiembre fue lanzada la primera versión 0.01 de Linux, la comunidad lo descargó y envió sus modificaciones a Linus, incluso recibió un correo de parte de Tanenbaum donde pronosticaba su fracaso, esta situación no lo desanimó y siguió adelante con su proyecto.

Las nuevas versiones betas siguieron publicándose con las mejoras del sistema bajo una licencia propia de Torvalds en la cual se especificaba que el código fuente podía compartirse pero quedaba estrictamente prohibido el uso comercial, fue a principios de 1992 cuando Linus decidió adoptar la licencia GNU GPL (Licencia Pública General de GNU) para su distribución lo que permitía que cualquier persona tenía acceso al código fuente para utilizarlo, modificarlo o distribuirlo libremente. Esta decisión permitió que a mediados de 1992 la combinación de GNU/Linux resultara en un sistema operativo libre, completo y funcional.

Es importante mencionar que el desarrollo de GNU/Linux se debe en gran parte al apoyo de una gran comunidad de programadores y usuarios que aportan su tiempo y sus ideas.

2.8.2 Razones para usar GNU/Linux

Hay muchas razones para usar GNU/Linux, pero las más importantes y las que consideramos esenciales para el desarrollo de nuestro proyecto son:

Estabilidad y Confiabilidad: GNU/Linux se desarrolla basándose en la arquitectura UNIX, la cual ha sido probada y refinada durante más de 35 años demostrando ser muy eficaz, robusta y segura. Además se ha diseñado desde cero para crear un sistema operativo estable y resistente a los fallos de sistema, por lo que rara vez es necesario reiniciarlo.

Seguridad: GNU/Linux es prácticamente inmune a los virus, troyanos y gusanos ya que fue diseñado desde un comienzo con la seguridad en mente. Se trabaja con permisos de administración para proteger los principales archivos del sistema, es decir, se protege desde adentro y no con aplicaciones adicionales al sistema.

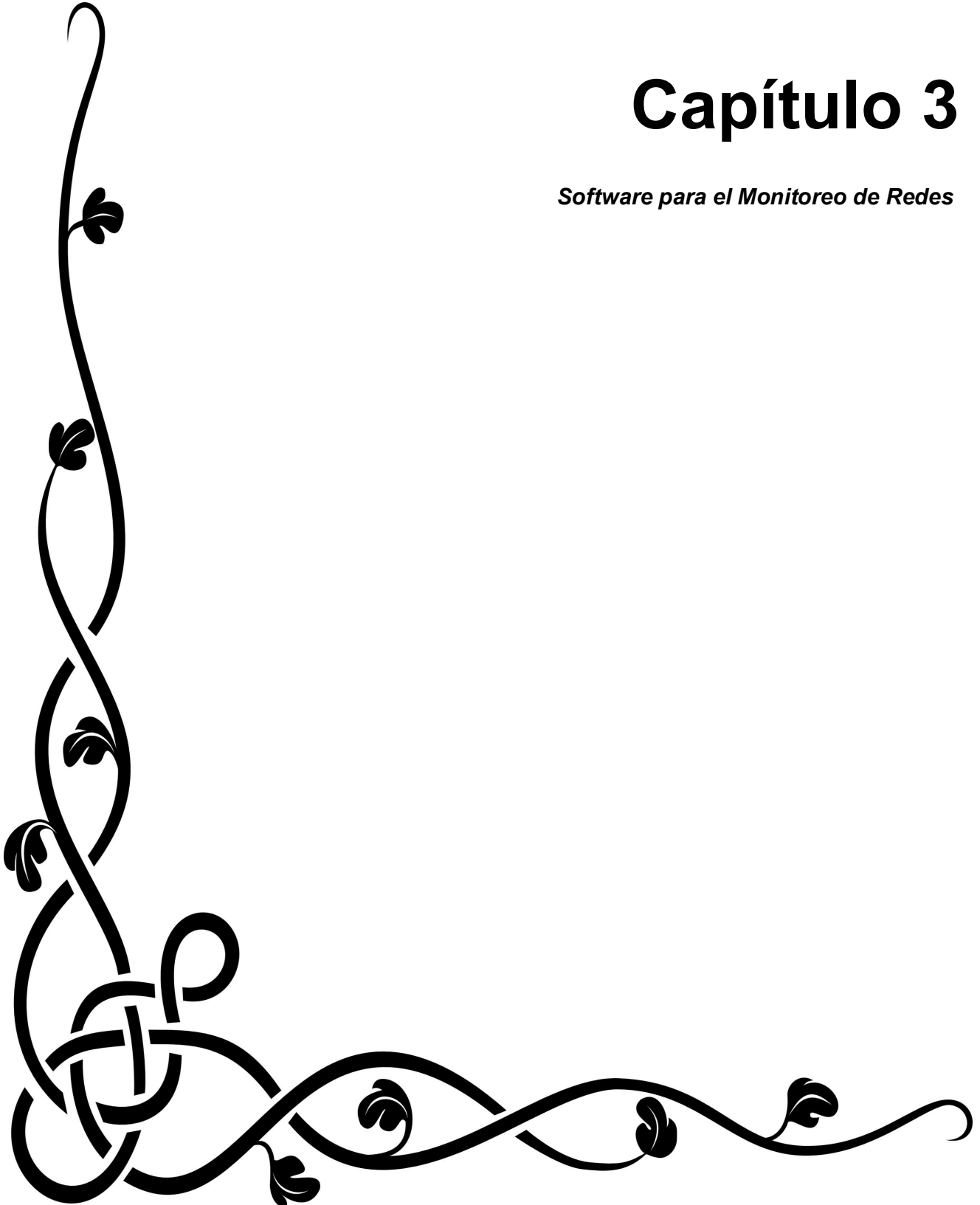
Flexibilidad: GNU/Linux nos da la libertad de configurar el sistema para adaptarlo a nuestras necesidades, ya que puede ser optimizado para estación de trabajo, para servidor o para su uso en una PC.

Libertad: Gracias a que GNU/Linux es distribuido bajo una licencia GPL tenemos la libertad de modificar, editar y redistribuir nuestro sistema. Podemos realizar modificaciones sencillas o cambiar complejas líneas de código para hacer que trabaje adecuándolo a nuestras necesidades.

Costo: No es necesario pagar por una licencia ya que la mayoría de las distribuciones de GNU/Linux son gratuitas, incluso puede descargarse de internet e instalarlo sin restricciones en cuantas máquinas se desee.

Capítulo 3

Software para el Monitoreo de Redes



*“No basta saber, se debe también aplicar. No es suficiente querer, se debe también hacer.”
Johann Wolfgang Goethe*

Software para el Monitoreo de Redes

El estilo de trabajo en la actualidad ha encausado a muchas de sus funciones en el uso de redes de computadoras, por mencionar algunos casos están las instituciones bancarias, los centros de investigación, el comercio electrónico, la educación a distancia, los buscadores de información, entre muchos otros.

Conforme las redes de computadoras han incrementado en extensión y en demanda de servicios, el monitoreo de las mismas a adquirido mucha importancia ya que es necesario mantener un control de las actividades que se realizan y de la cantidad de información que circula en ella para ofrecer un servicio confiable de comunicación, debido a que en la actualidad muchas personas dependen de una red para poder desarrollar su trabajo o incluso existen redes de apoyo a equipo médico de soporte vital.

Es por esto que el tener un apoyo visual de los datos recolectados durante el monitoreo y vigilancia de la red es de gran utilidad para determinar las causas de los problemas que se presenten como las caídas de red, los cuellos de botella y algunas otras cuestiones.

El monitoreo consiste en observar y recolectar información referente al comportamiento de la red y para el IFUNAM es importante considerar los siguientes aspectos.

La utilización de enlaces

Obtener la cantidad de ancho de banda utilizada por cada una de las interfaces de red de las computadoras personales, servidores, estaciones de trabajo y otros elementos que forman parte de la red de cómputo del IFUNAM.

Porcentaje de transmisión y recepción de información

Identificar los equipos de cómputo personales, servidores, estaciones de trabajo y otros elementos de la red que mas solicitudes hacen y atienden.

3.1 Elección del software

Antes de encontrar un software que se adecuara a nuestras necesidades fue necesario realizar un análisis de las aplicaciones que ofrecen actualmente los sistemas disponibles para el monitoreo de redes.

Durante este procedimiento se buscaron aplicaciones consideradas como software libre que se distribuyeran bajo la Licencia Pública General de GNU y, principalmente, que pudieran ejecutarse en sistemas operativos GNU/Linux.

Es importante mencionar que hay una gran variedad de opciones en software que pueden adecuarse a cada usuario, pero solo se mencionaran los programas que se investigaron debido a sus características.

3.1.1 CACTI

Software desarrollado con PHP que proporciona una interfaz de usuario muy completa para aprovechar las funcionalidades de las herramientas RRDTool. Es capaz de almacenar información para generar gráficas con datos obtenidos de una base de datos MySQL, además cuenta con soporte SNMP (Simple Network Management Protocol)¹⁸ para la recolección de datos del tráfico de red.

Cacti recolecta los datos que el usuario necesita cada determinado tiempo para almacenarlos en la base MySQL y en los archivos de *planificación Round Robin*¹⁹, este procedimiento permite que las fuentes de datos se actualicen constantemente y permita generar gráficas

¹⁸ El Protocolo Simple de Administración de Red es un conjunto de estándares que facilita el intercambio de información de tráfico entre diferentes dispositivos de red.

¹⁹ La planificación Round Robin es un algoritmo utilizado para seleccionar un grupo de datos, que coinciden en tiempo y lugar, de forma concurrente.

con datos confiables.

Una característica muy importante de Cacti es que ofrece la funcionalidad de manejo de usuarios, por lo que se podrán dar de alta a usuarios con diferentes privilegios. Esto permite tener usuarios con permisos para modificar parámetros en las gráficas y otros que solo puedan consultarlas.

3.1.2 CRICKET

Software de alto rendimiento encargado de dar seguimiento a los datos y desarrollado para ayudar a los administradores a visualizar y entender el tráfico de la red. Cuenta con un sistema de configuración jerárquico, lo que evita que la información se repita. Fue escrito en su totalidad con Perl.

Compuesto por un colector y un graficador. El colector de datos se ejecuta cada determinado tiempo y almacena la información en Bases de Datos Circulares (RRD - Round Robin Database), posteriormente con ayuda de las herramientas RRDTool y una Interfaz de Entrada Común (CGI - Common Gateway Interface)²⁰ puede visualizar las gráficas de los datos recolectados.

3.1.3 MRTG

Herramienta escrita en C y en Perl para supervisar la carga de tráfico en determinadas interfaces de red. MRTG (Multi Router Traffic Grapher) se diseñó para recolectar contadores de tráfico SNMP y el registro de datos para convertirlos en gráficas, que posteriormente serán embebidas en una página web.

Se pueden generar gráficas que muestren el tráfico diario, semanal y anual de prácticamente cualquier dispositivo conectado a la red. Esto ayuda a tener una visualización rápida cuando hay problemas en la red.

²⁰ Mecanismo de comunicación entre el servidor web y una aplicación externa. Las aplicaciones CGI permiten crear contenido dinámico en páginas web.

3.1.4 IPTRAF

Programa que nos sirve para generar estadísticas de red, es una aplicación que se ejecuta en la terminal utilizando una interfaz de usuario desarrollada con la librería *curse*²¹. IPTRAF es un monitor que muestra información del tráfico IP que pasa a través de la red.

Permite elegir entre diferentes servicios como el monitoreo local, especificación de dispositivos que se desean controlar, filtrado de paquetes de diferentes protocolos, generar archivos de registros, búsqueda de DNS inversa para ofrecer mayor información.

Cada una de los programas anteriores ofrecía opciones que solucionarían parte del problema expuesto en dicho trabajo de Tesis, pero la Secretaría Técnica de Cómputo del IFUNAM buscaba un software más específico, por lo que se llegó a la decisión de utilizar dos herramientas:

RRDTool como el programa graficador.

IP Flow Meter como el programa recolector de información.

A continuación se hablará de cada uno de ellos más detalladamente.

3.2 IP Flow Meter (IPFM)

IPFM es una herramienta encargada de analizar el ancho de banda, es decir que mide la cantidad de datos que son enviados y recibidos a través de un enlace de internet para determinados *hosts* dentro de una red.

IPFM fue desarrollado usando la librería *libpcap*, la cual fue escrita en C para facilitar su portabilidad y proporciona al usuario una interfaz que facilita la captura, mediante funciones, de paquetes en la capa de red. Este sistema es utilizado para la monitorización de redes de bajo nivel.

²¹ Librería de programación para el control de terminales en sistemas UNIX.

Ventajas de IPFM

- IPFM genera archivos de texto cada determinado tiempo, los cuales contienen la información en bytes del consumo de ancho de banda por cada *host* configurado. A continuación se muestra un ejemplo del formato de cada archivo generado:

```
# IPFMv0.11.5 yyyy/mm/dd 00:00:00 (local time) -- dump every 0d00:00:00 -- listening on eth0
# Host          In (bytes)      Out (bytes)      Total (bytes)
host1.dominio.com 2575494         0                2575494
host2.dominio.com 0               2407417         2407417
host3.dominio.com 0               69312           69312
# end of dump yyyy/mm/dd 00:00:00
```

- Se definen los *hosts* de los que deseamos generar un registro de actividad.
- Podemos definir el intervalo de tiempo de salida, es decir cada cuánto determinado tiempo generaremos un archivo de registro.
- Se definen los nombres de los archivos de registro, en los cuales se incluye la fecha para tener una mejor referencia.
- Se puede configura la *resolución inversa de DNS*²².
- Se realiza la clasificación de bytes en IN (la cantidad de paquetes recibidos), OUT (la cantidad de paquetes enviados) y TOTAL (la suma de IN y OUT es la cantidad total de datos transferidos).

Desventajas de IPFM

- IPFM se diseñó para realizar el registro de actividades de un solo dispositivo de red.
- Hay poca documentación actualizada.

²² Traducción de una dirección IP a su nombre de dominio.

3.3 Herramientas de Base de Datos Circulares (RRDTool)

RRDTool es un programa de alto rendimiento utilizado para el registro de datos (tráfico de red, carga de servidores, temperatura, etc.) y su representación mediante gráficas en diferentes intervalos de tiempo. Los datos son almacenados en bases de datos compactas que tienen la característica de no crecer con el tiempo.

3.3.1 Bases de Datos Circulares (RRD – Round Robin Database)

En la actualidad el uso de las bases de datos se ha convertido en una actividad muy común, ya que la independencia física de la información ha favorecido el desarrollo de sistemas encargados de administrar los datos almacenados.

Una base de datos es un conjunto de archivos o datos relacionados y existe una colección de programas diseñado para crear y administrar bases de datos llamados sistema de bases de datos. Un sistema de base de datos es básicamente un sistema computarizado cuya finalidad general es almacenar información y permitir que los usuarios puedan recuperar y actualizar dicha información con base en peticiones.²³

En la Figura 3.1 podemos apreciar la relación que existe entre una base de datos y un sistema de bases de datos.

Las bases de datos pueden clasificarse en diversos rubros, de acuerdo a su función o de acuerdo a su modelo de administración, aunque al final el objetivo es el mismo, almacenar información. Pero debido a la naturaleza del problema planteado en la presente tesis fue necesaria la búsqueda de soluciones diferentes a las que conocemos generalmente. Es así como se optó por utilizar las Round Robin Database - Bases de Datos Circulares.

²³ Fuente: C. J. Date “Introducción s los Sistema de Bases de Datos”

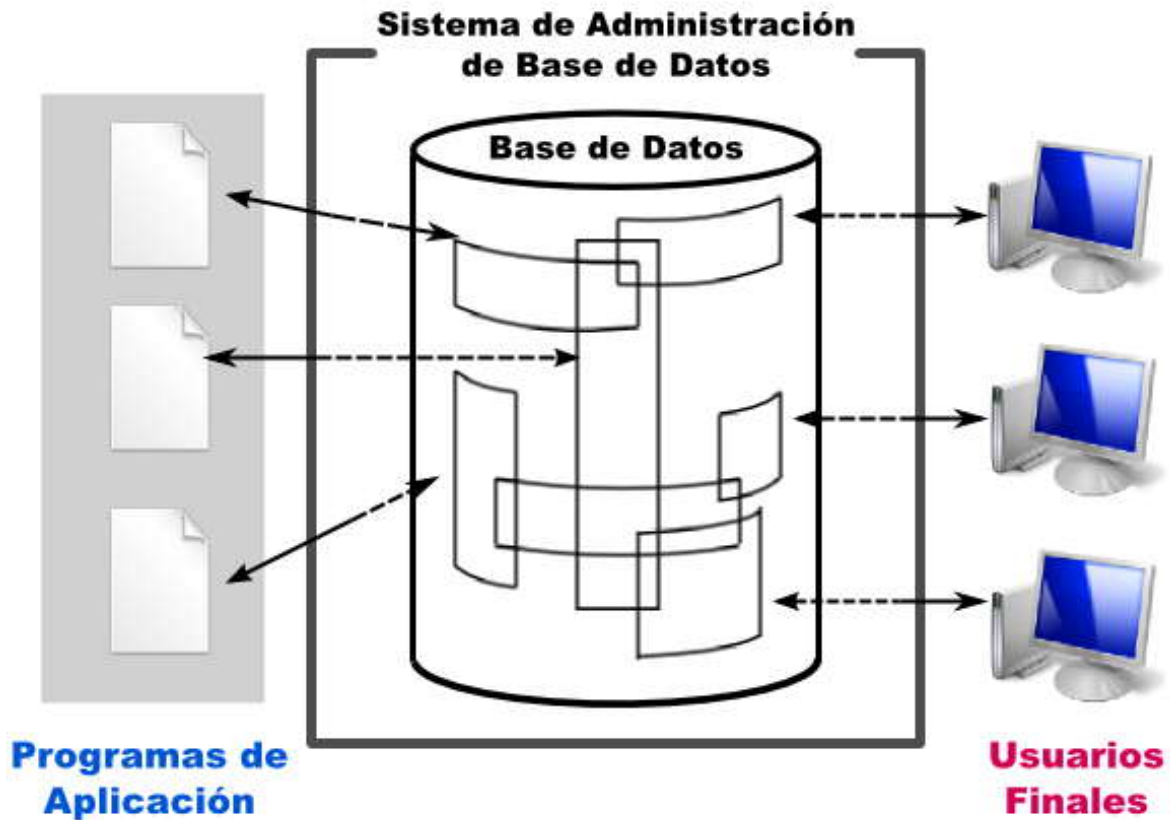


Figura 3.1 Sistema de Base de Datos²⁴

Base de Datos Circular

Es un tipo de base de datos muy específica, orientada al almacenamiento de secuencias de datos medidos en determinados periodos de tiempo, espaciadas de forma uniforme y ordenadas cronológicamente. Trabaja con una cantidad fija de datos que es especificada en el momento de su creación y con un apuntador al dato más actual. Debido a su estructura sencilla son usadas principalmente como herramientas de monitoreo.

Razones para usar una Base de Datos Circular

Una RRD se caracteriza porque siempre contendrá la misma cantidad de datos, esto se logra gracias a que al llegar al límite máximo de almacenamiento, los datos se sobrescribirán en los más antiguos.

Para comprender mejor el concepto imaginemos un círculo dividido en varios sectores como

²⁴ Fuente: C. J. Date "Introducción s los Sistema de Bases de Datos"

se muestra en la Figura 3.2, dichos sectores representan la cantidad máxima de datos que se podrán almacenar en la RRD, cada determinado tiempo se ingresará información en uno de los sectores, la flecha nos sirve como apuntador para indicarnos cual es el dato más reciente, conforme la base vaya llenándose de datos llegará el momento en que se complete una vuelta del círculo. Al llegar nuevamente al inicio, los datos más nuevos comenzarán a sobrescribirse en los sectores más antiguos para continuar el siguiente ciclo.

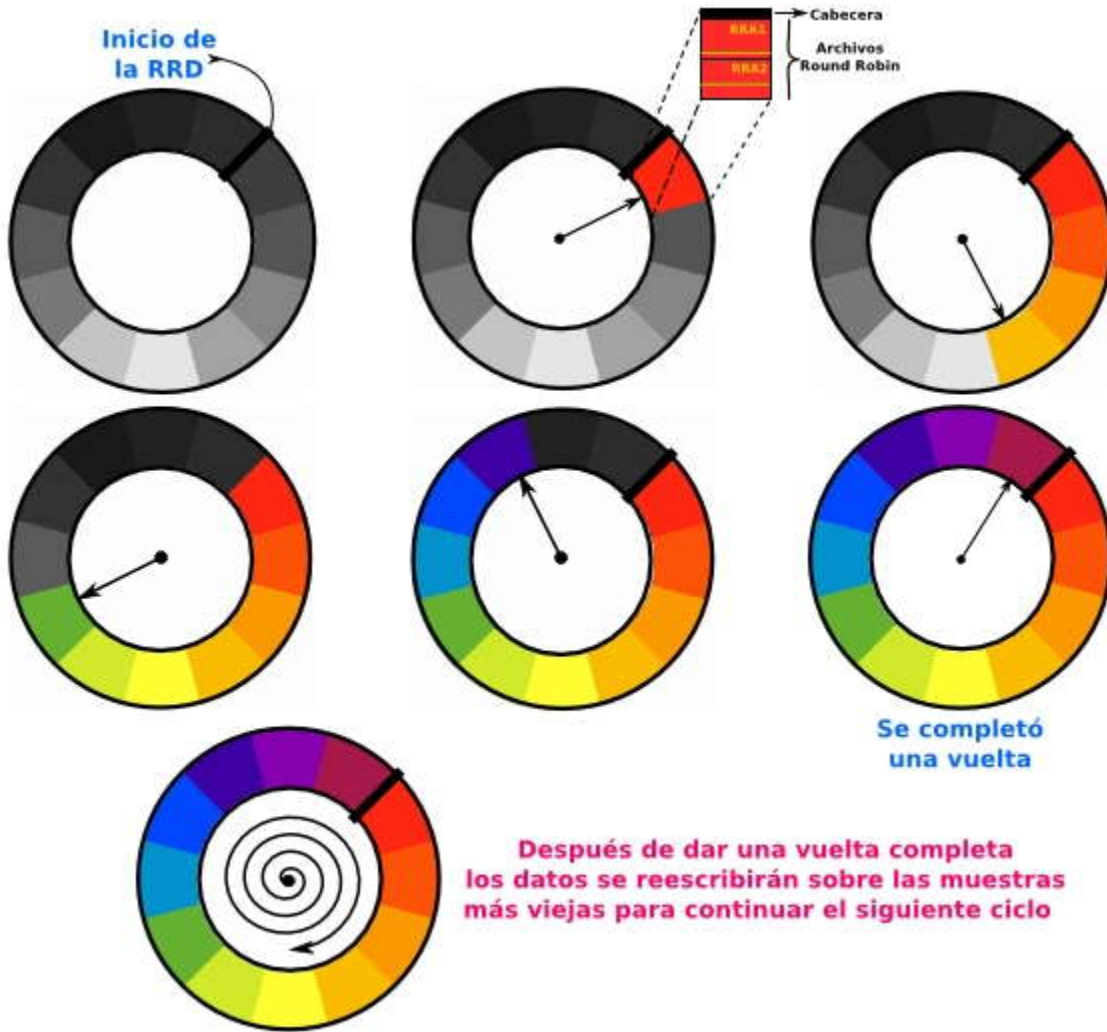


Figura 3.2 Base de Datos Circular

Este principio es la razón por la que se decidió usar las bases de datos circulares ya que al mantener un número fijo de datos (logrando que la base no crezca con el tiempo), evitamos depender físicamente del espacio de almacenamiento necesario para dicha información.

3.3.2 Componentes de una RRD

Las bases de datos circulares están organizadas de tal forma que cada componente cuenta con acciones específicas para su creación, a continuación se comentarán las características principales de cada uno de ellos.

Fuentes de Datos (DS)

Las fuentes de datos son las “muestras” o el origen de nuestros datos, cada RRD puede aceptar aportaciones de diversas fuentes de datos. En la Tabla 3.1 se describen los cuatro tipos de DS existentes:

Tipo DS	Descripción
GAUGE	Indicador de diferentes valores en el tiempo, registra los valores tal como lo medimos
COUNTER	Contador con incremento constante, se encarga de registrar el <i>incremento / intervalo de tiempo</i>
DERIVE	Contador que acepta valores negativos
ABSOLUTE	Contador que realiza un reset después de su lectura

Tabla 3.1 Tipos de Fuentes de Datos

Cuando configuramos una fuente de datos podemos definir las propiedades básicas de cada DS que deseamos almacenar en nuestra RRD. Dichas propiedades son el nombre, el tipo, el tiempo máximo que puede pasar antes de considerar los datos como desconocidos y los valores máximos y mínimos que pueden registrarse.

Funciones de Consolidación (CF)

Una función de consolidación consiste en un cálculo matemático o una selección lógica de los datos recolectados, se define en el momento de crear una RRD. Es necesario especificar el intervalo de tiempo en que ocurrirá dicha consolidación y que CF debe utilizarse para realizar la cuenta de los valores recolectados.

El uso de diferentes CF nos permite almacenar exactamente el tipo de dato que nos interesa, por lo que se cuenta con 4 tipos (Tabla 3.2):

CF	Tipo	Descripción
AVERAGE	Promedio	Se toma el promedio aritmético de los datos recolectados
LAST	Último valor leído	Se toma el último valor recolectado
MIN	Mínimo valor leído	Se toma el valor más pequeño recolectado
MAX	Máximo valor leído	Se toma el valor más grande recolectado

Tabla 3.2 Tipos de Funciones de Consolidación

Al crear una RRD se puede realizar una combinación de diferentes configuraciones de consolidación, por ejemplo, para el caso del Instituto de Física de la UNAM se recolectan datos cada 5 minutos y se definieron intervalos de consolidación para construir la cuenta de los valores cada 5 minutos, 24 horas, 1 mes y 1 año mediante las funciones AVERAGE, MIN y MAX para cada uno de ellos. Los datos consolidados son almacenados en los llamados Archivos Round Robin.

En la Figura 3.3 podemos observar un ejemplo sencillo de cómo se realiza el muestreo y la consolidación de los datos.

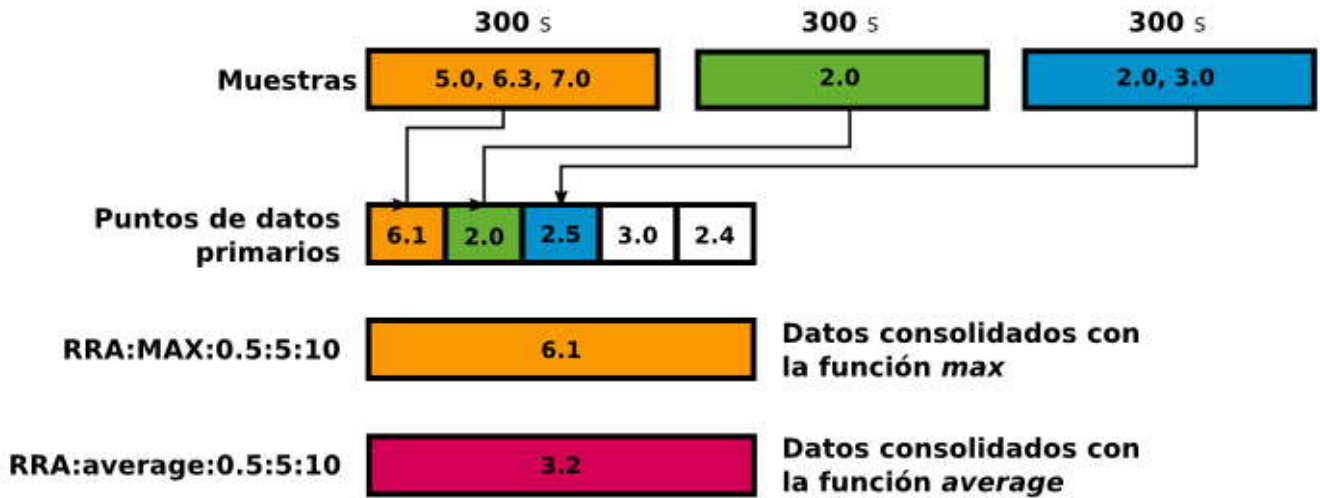


Figura 3.3 Muestreo y Consolidación de Datos

Cada 300 segundos (5 minutos) se recolectan datos y se establecen los puntos de datos primarios calculando el promedio de las muestras, por ejemplo, la primera muestra de la Figura 3.3 recolectó tres datos los cuales son sumados y divididos entre el número total de datos de la muestra $[(5.0 + 6.3 + 7.0) / 3 = 6.1]$, así obtenemos el primer punto de datos

primario, posteriormente se calcula el promedio de la siguiente muestra $[2.0 / 1 = 2.0]$ para el segundo punto de datos primario y así continuamente con todas las muestras obtenidas. A continuación se consolidan los datos de acuerdo a las funciones de consolidación que fueron definidas, para seguir con el ejemplo se puede observar en la figura dos tipos de funciones, la primera es la función MAX, la cual obtendrá de los puntos de datos primarios el máximo valor leído (6.1), la segunda es la función AVERAGE la cual dará como resulta el promedio de los puntos de datos primarios (3.2). Estos datos consolidados serán almacenados en los *Archivos Round Robin*.

Archivos Round Robin (RRA)

Los RRA son archivos configurados para indicar que tipo de datos queremos registrar, por lo que es necesario usar una función de consolidación. Estos archivos contienen un número limitado de datos consolidados para cada una de las fuentes de datos (DS) definidas.

Los datos adquiridos durante la recolección de información deben ser ingresados en una RRD cada determinado intervalo de tiempo, dichos intervalos son definidos en segundos. Estos se conocen como los puntos de datos primarios. Los datos deben pasar por el proceso de consolidación mediante las funciones que fueron especificadas en el momento de su creación.

En la figura 3.4 se observa un ejemplo sencillo de cómo se genera un RRA. Cuando se obtienen las muestras y se establecen los puntos de datos primarios, es necesario consolidar los datos, este proceso genera un RRA el cual contiene un número fijo de muestras durante un periodo de tiempo establecido. Cada RRA es almacenado en la base de datos circular, la cual también tiene un tamaño fijo, si ésta se encuentra en su límite el RRA se escribirá sobre el RRA más viejo, es pocas palabras, se descartará la muestra más vieja.

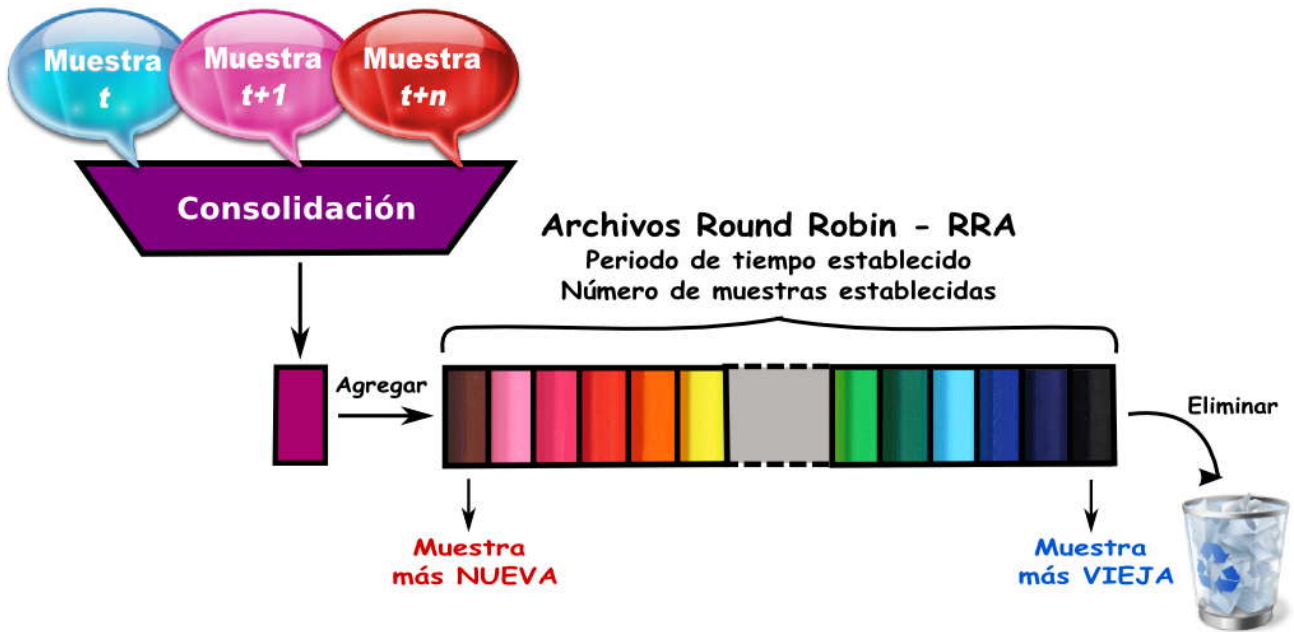


Figura 3.4 Archivos Round Robin

3.3.3 Herramientas más útiles

Las herramientas proporcionadas por RRDTool nos servirán principalmente para interactuar con la base de datos. Las funciones más utilizadas son:

rrdtool create

Esta función nos permite crear un nuevo archivo RRD. El archivo es creado con su tamaño completo y llenado con datos tipo UNKNOWN (desconocido).

rrdtool update

Lleva a cabo la actualización de los archivos RRD almacenando un conjunto de datos nuevos. Los datos son procesados de acuerdo a la configuración de las bases de datos.

rrdtool graph

Genera gráficos a partir de los datos de uno o más archivos RRD para mostrar la información en una forma que sea fácil de entender para las personas. Su principal objetivo es crear gráficos pero también puede generar reportes numéricos.

rrdtool fetch

Permite extraer información de un determinado periodo de tiempo. Esta instrucción es utilizada internamente por la función graph para obtener los datos de los archivos RRD, analiza dichos archivos e intenta recuperar la información en el tiempo solicitado.

rrdtool info

Nos proporciona la información de la cabecera de los archivos RRD en un formato amigable.

rrdtool rdcgi

Genera páginas web dinámicas en las que se incluyen las gráficas RRD. Su objetivo es ejecutar un programa CGI y generar una plantilla web que contenga etiquetas especiales, es decir, la impresión de una página web que incluya las cabeceras CGI necesarias.

Ventajas de RRDTool

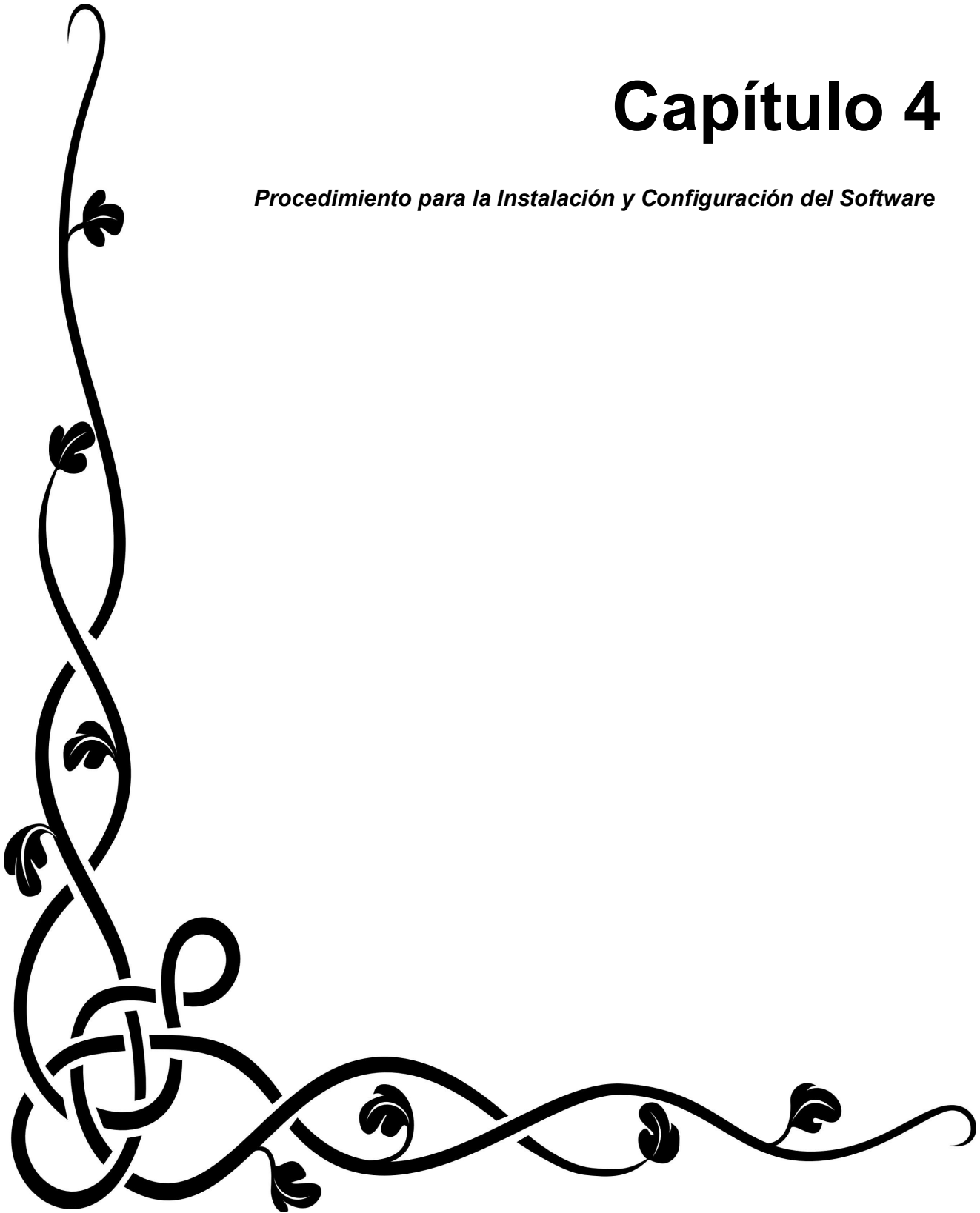
- Se requiere poco espacio de almacenamiento ya que las bases de datos tienen un tamaño fijo y no crecen con el tiempo.
- Se pueden generar gráficas (.png y .gif) con atributos que son definidos por el usuario.
- Las gráficas pueden presentar reportes numéricos.
- Cada gráfica puede presentar información de una o varias fuentes de datos.
- Debido a que diversos programas administradores de red utilizan dicha herramienta, es fácil encontrar documentación que nos enseñe a utilizarla.

Desventajas de RRDTool

- Es necesario utilizar procesos externos para recolectar la información e insertar los datos en las RRDs correspondientes.
- Si no se configuran adecuadamente los intervalos de tiempo para la recolección de datos habrá información que se perderá.

Capítulo 4

Procedimiento para la Instalación y Configuración del Software



“Las obras de conocimiento deben ser libres, no hay excusas para que no sea así”
Richard Stallman

Procedimientos para la Instalación y Configuración del Software

4.1 Instalación del Sistema Operativo

Se realizó la instalación del sistema operativo *Ubuntu Server Edition 8.04*, el cual puede ser solicitado o descargado desde el sitio web oficial <http://www.ubuntu.com/>. A continuación se mostrará el procedimiento que se siguió para su instalación.

El primer paso fue ingresar en el BIOS del servidor para indicar que el arranque se realizaría desde un CD. Después de guardarse los cambios se inserto el CD de instalación de Ubuntu y se reinició el equipo.

Al comenzar a ejecutarse fue necesario proporcionar algunos datos y seguir una serie de pasos para comenzar la configuración e instalación del sistema.

Paso 1.

Como se muestra en la Figura 4.1 lo primero que se nos presenta en la pantalla es una lista gráfica con los idiomas disponibles, el elegido será el idioma que se usará durante todo el proceso de instalación y configuración, seleccionamos *English* (Inglés).



Figura 4.1 Elegir idioma para instalación

Paso 2.

A continuación se muestra en la pantalla un menú con diferentes opciones (Figura 4.2) donde podemos elegir la instalación, verificar el CD en busca de errores, recuperar un sistema dañado, comprobar la memoria o iniciar desde otro sistema operativo, en nuestro caso seleccionamos la opción *Install Ubuntu Server* (Instalar Ubuntu Server) para comenzar la instalación.

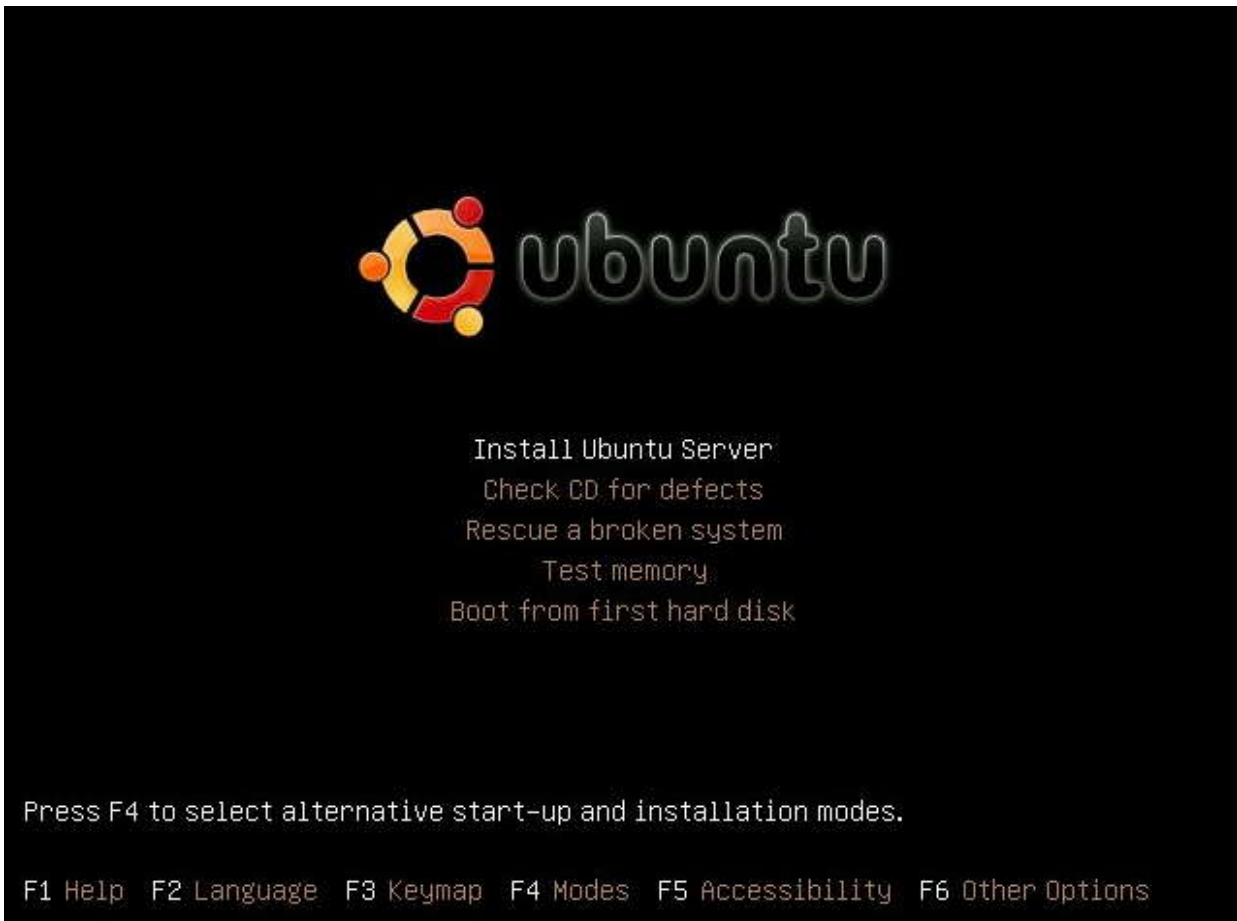


Figura 4.2 Instalar Ubuntu Server

Paso 3.

A partir de este momento el proceso de instalación se realiza en modo texto. En la siguiente pantalla se presenta otro menú para seleccionar el idioma predeterminado que tendrá el sistema operativo (Figura 4.3). Elegimos la opción *English*.



Figura 4.3 Elegir idioma del sistema

Paso 4.

Elegir el país, territorio o área en el que nos encontramos (Figura 4.4), en nuestro caso seleccionamos México.

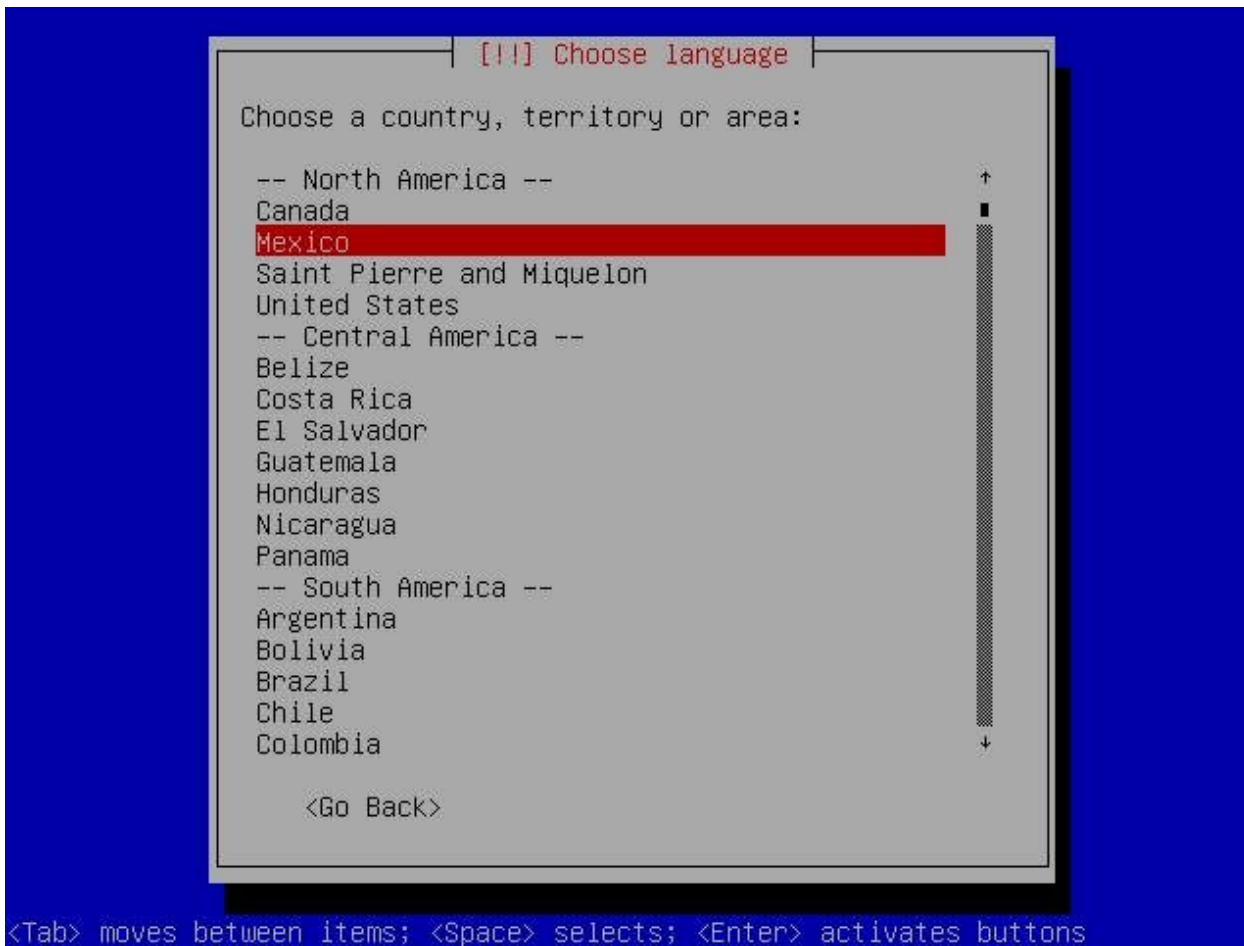


Figura 4.4 Elegir país, territorio o área

Paso 5.

A continuación debemos seleccionar la distribución del teclado. Primero se solicita indicar el origen del teclado (Figura 4.5) el cual se realiza de acuerdo a nuestro región, en el menú elegimos la opción *Latin American*.

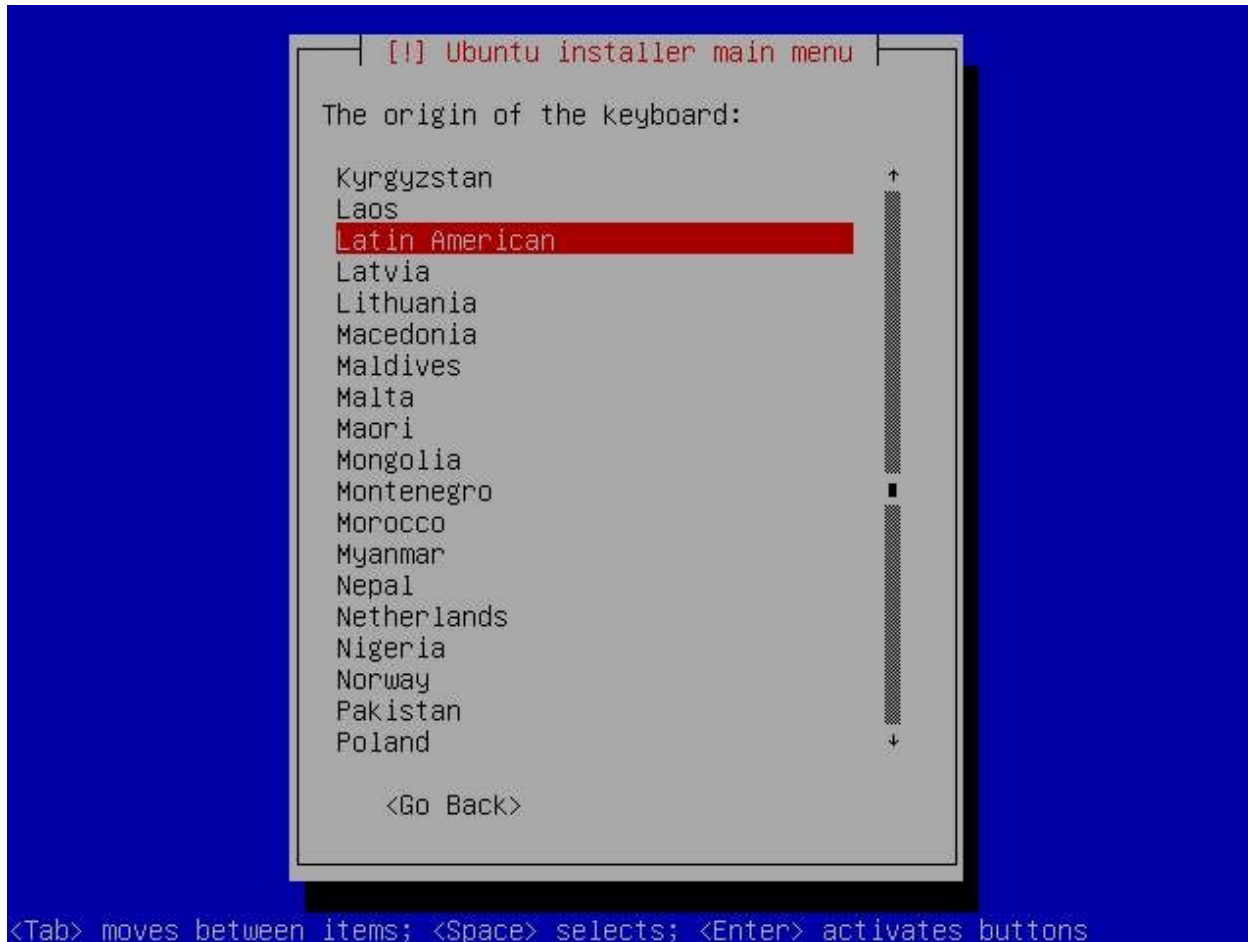


Figura 4.5 Elegir idioma del teclado

Paso 6.

Posteriormente seleccionamos su distribución, Latin American (Figura 4.6).

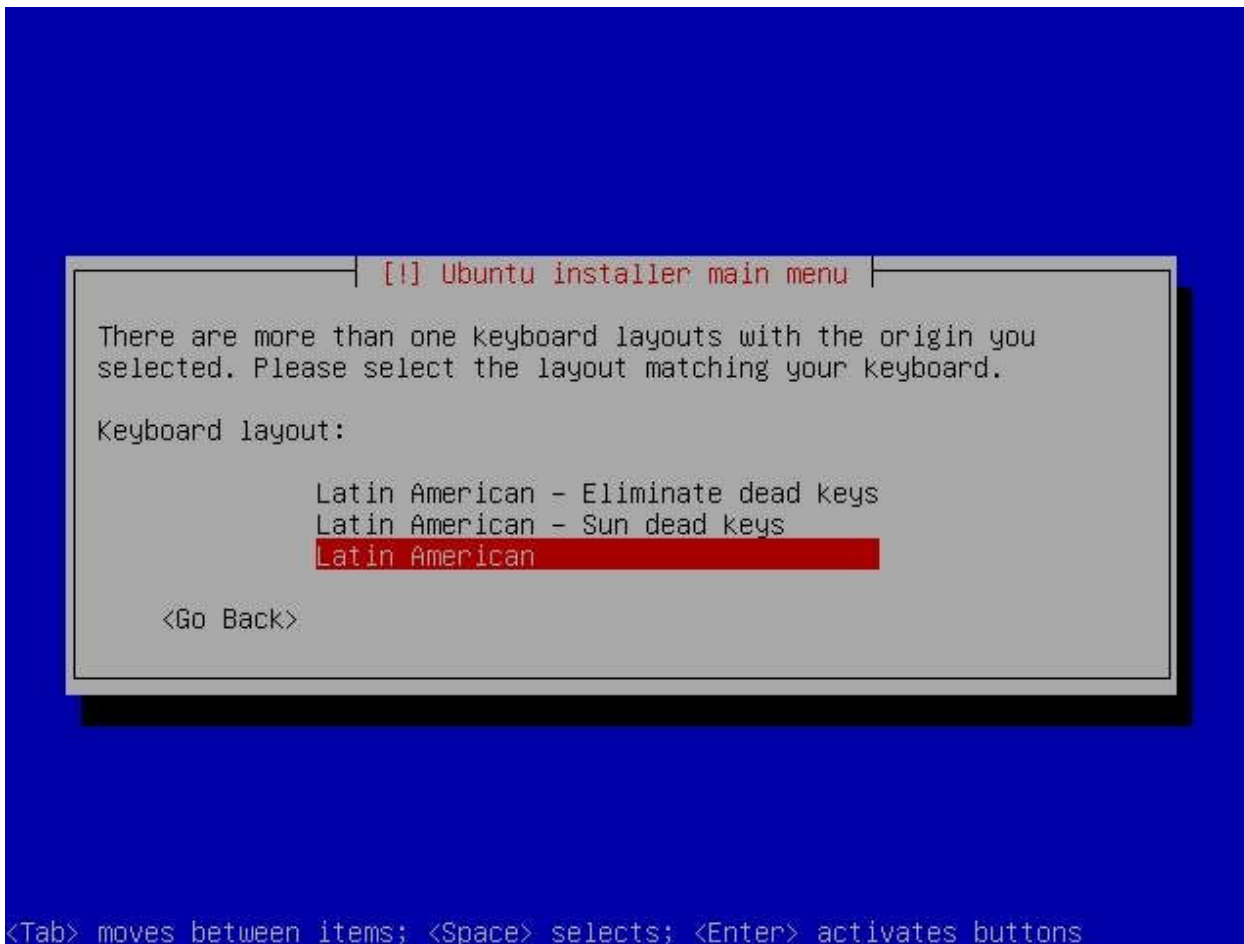


Figura 4.6 Elegir tipo de teclado

Paso 7.

A continuación se solicita elegir un nombre para el *host* (Figura 4.7). Por solicitud del administrador de red del instituto se le puso el nombre de **quarks**.

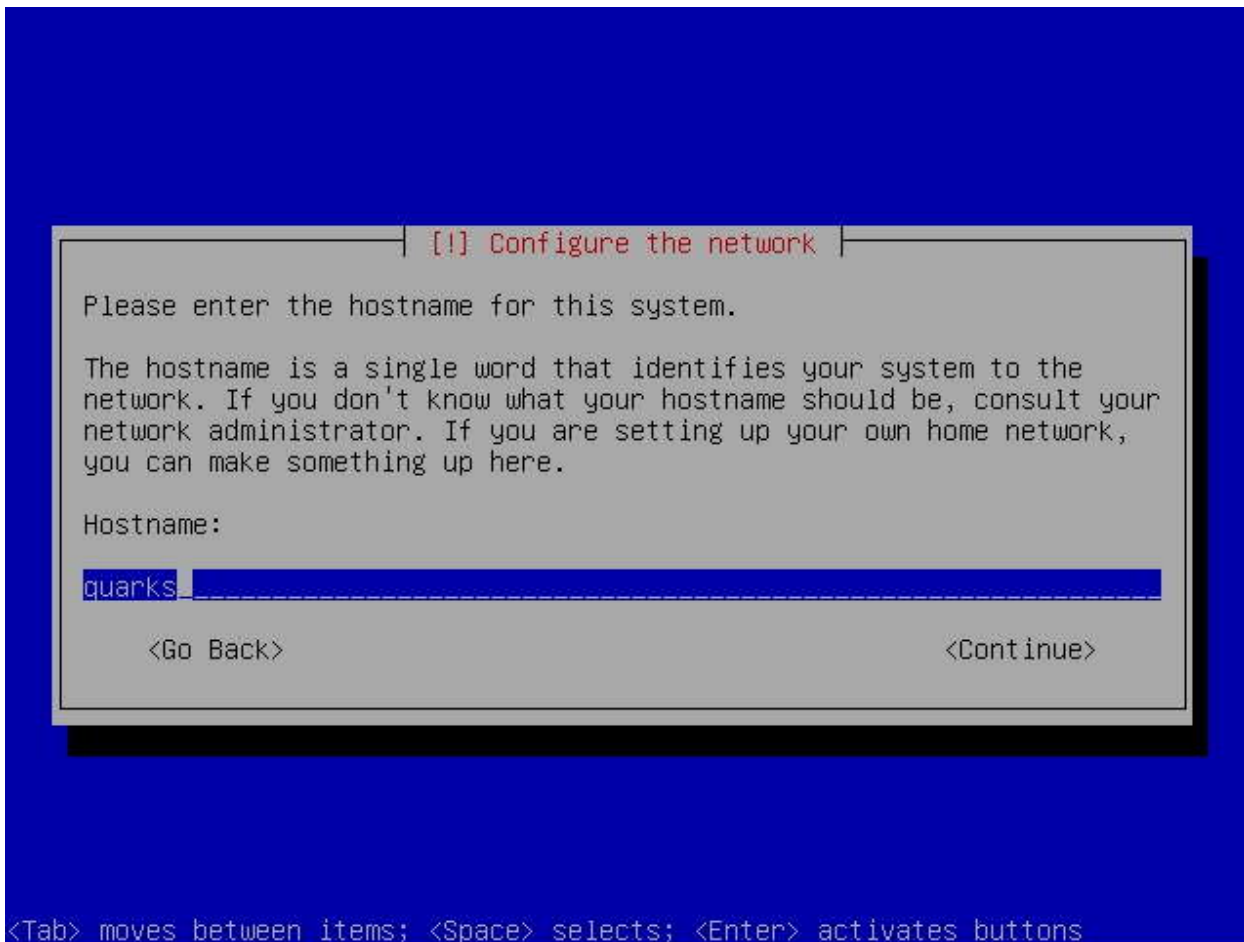


Figura 4.7 Elegir nombre para el host

Paso 8.

Para configurar el reloj solo debemos seleccionar la zona horaria (Figura 4.8), por nuestra ubicación la opción a elegir es *Central*.

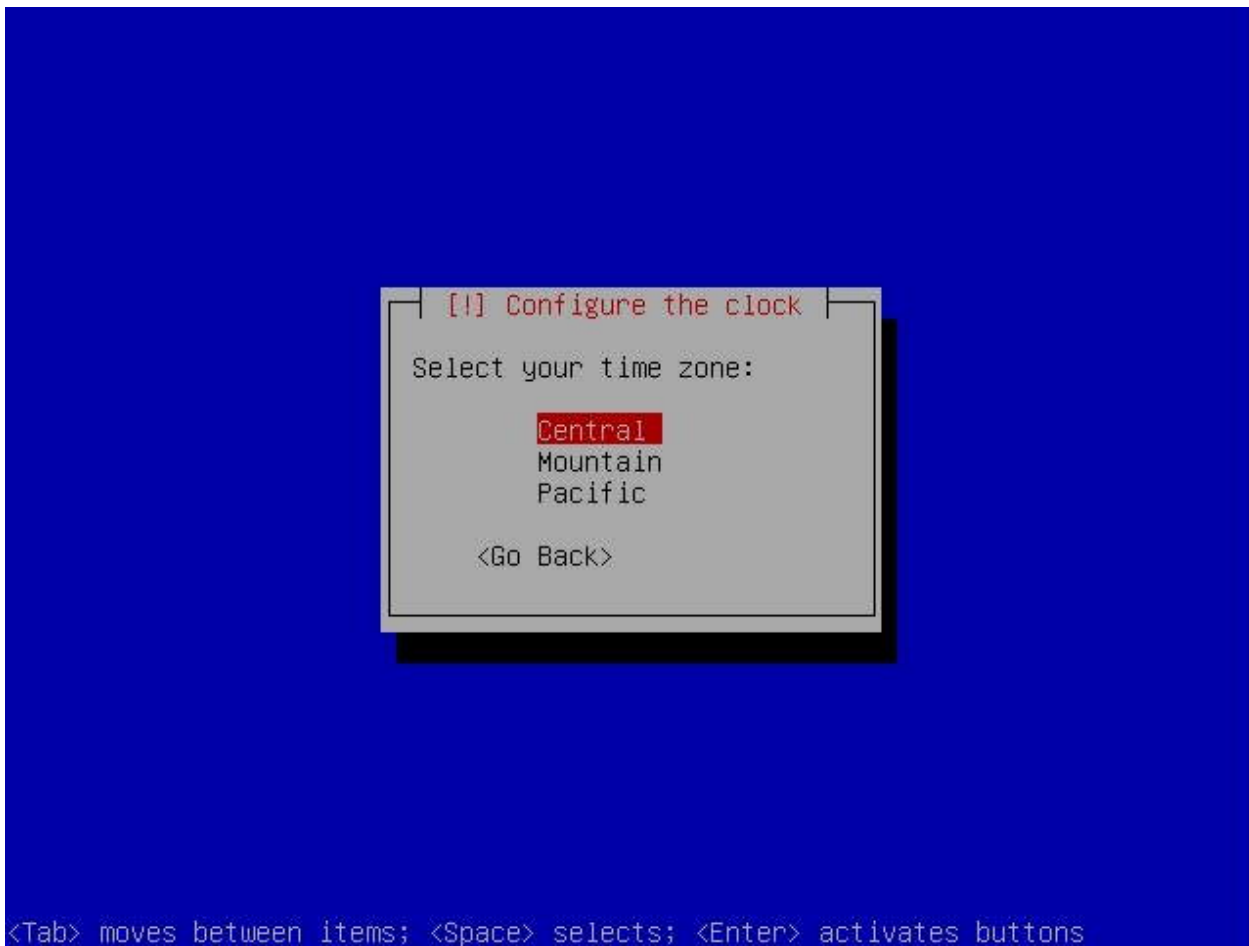


Figura 4.8 Elegir zona horaria

Paso 9.

El siguiente paso es el proceso de particionar²⁵ el disco (Figura 4.9), por lo que se nos presenta una tabla de particiones inicial con el espacio disponible. Esta partición se realiza para que el sistema operativo crea que hay varios discos cuando solo hay uno físico.

²⁵ Realizar una división en el disco duro

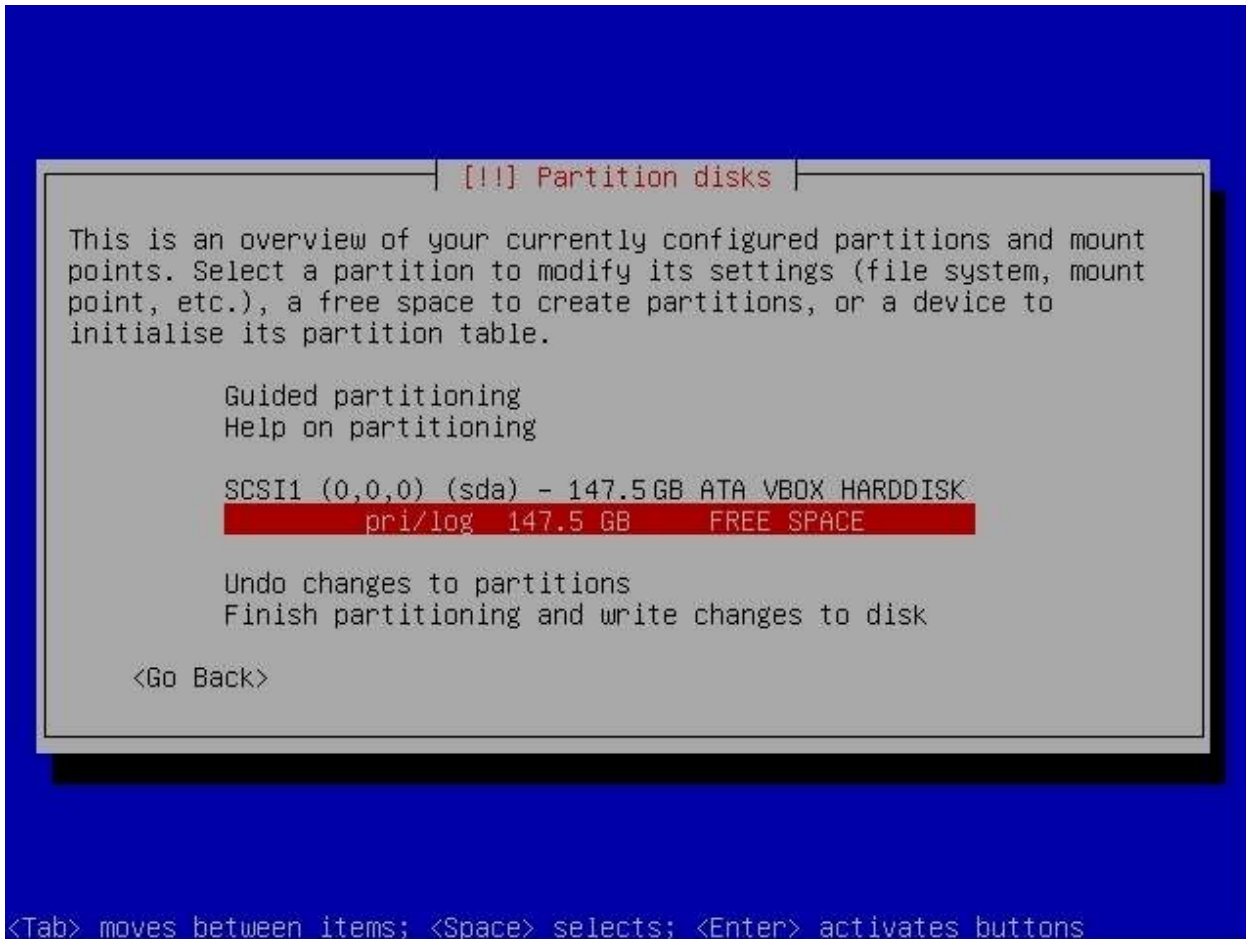


Figura 4.9 Tabla inicial de particiones del disco duro

Seleccionamos *FREE SPACE* (Espacio Libre) para comenzar la partición. A continuación nos pregunta como deseamos usar este espacio y elegimos la opción de *Create a new partition* (Crear una nueva partición). Ver figura 4.10.

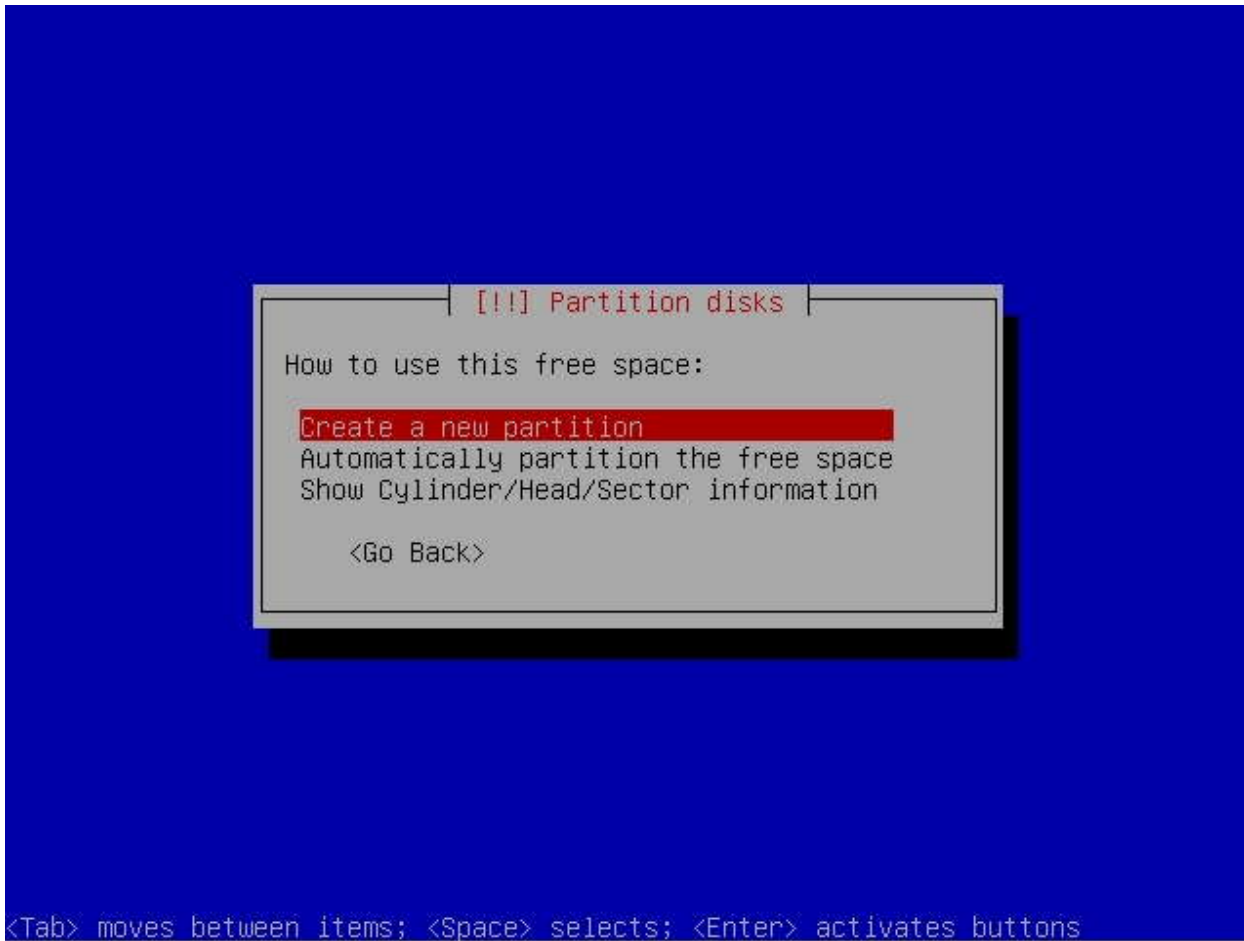


Figura 4.10 Crear una nueva partición de disco

A partir de este momento se solicitan las características que tendrá cada partición a configurar. Para el servidor quarks se realizaron 4 particiones principales:

boot: Directorio que contiene todos los archivos necesarios para el arranque del sistema.

/: La raíz es el directorio principal o maestro que contiene todos los directorios del sistema.

swap: Espacio utilizado como una extensión de memoria virtual en el sistema.

/home: Directorio que contiene los directorios iniciales de los usuarios.

En la Figura 4.11 se presenta la tabla de particiones final del servidor.

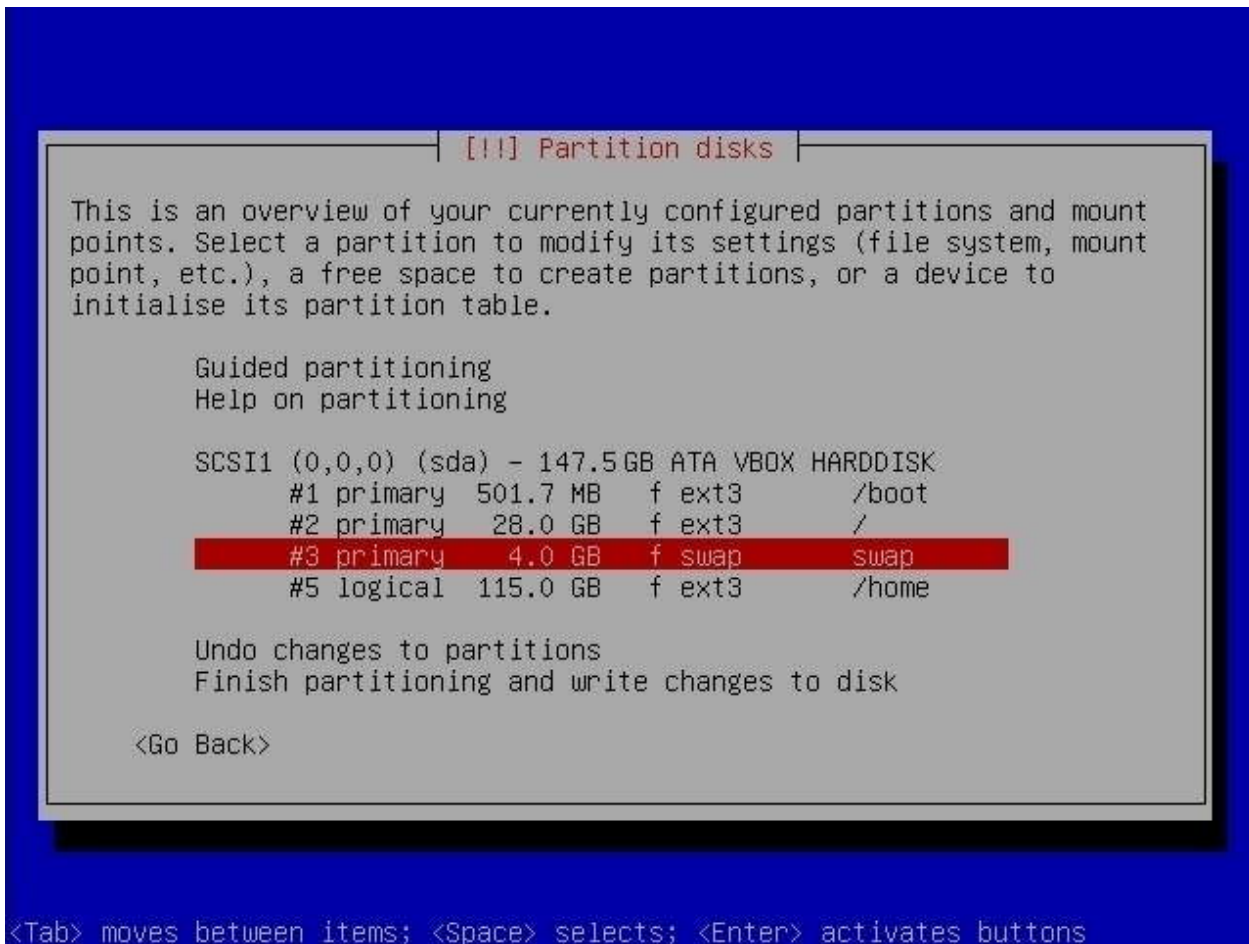


Figura 4.11 Tabla final de particiones del disco duro

Paso 10.

Para continuar elegimos la opción de terminar la partición y guardar los cambios en el disco, como el cambio borrará cualquier dato que este almacenado en el disco, pide la confirmación para guardar estos cambios, seleccionamos la opción Yes (Figura 4.12).

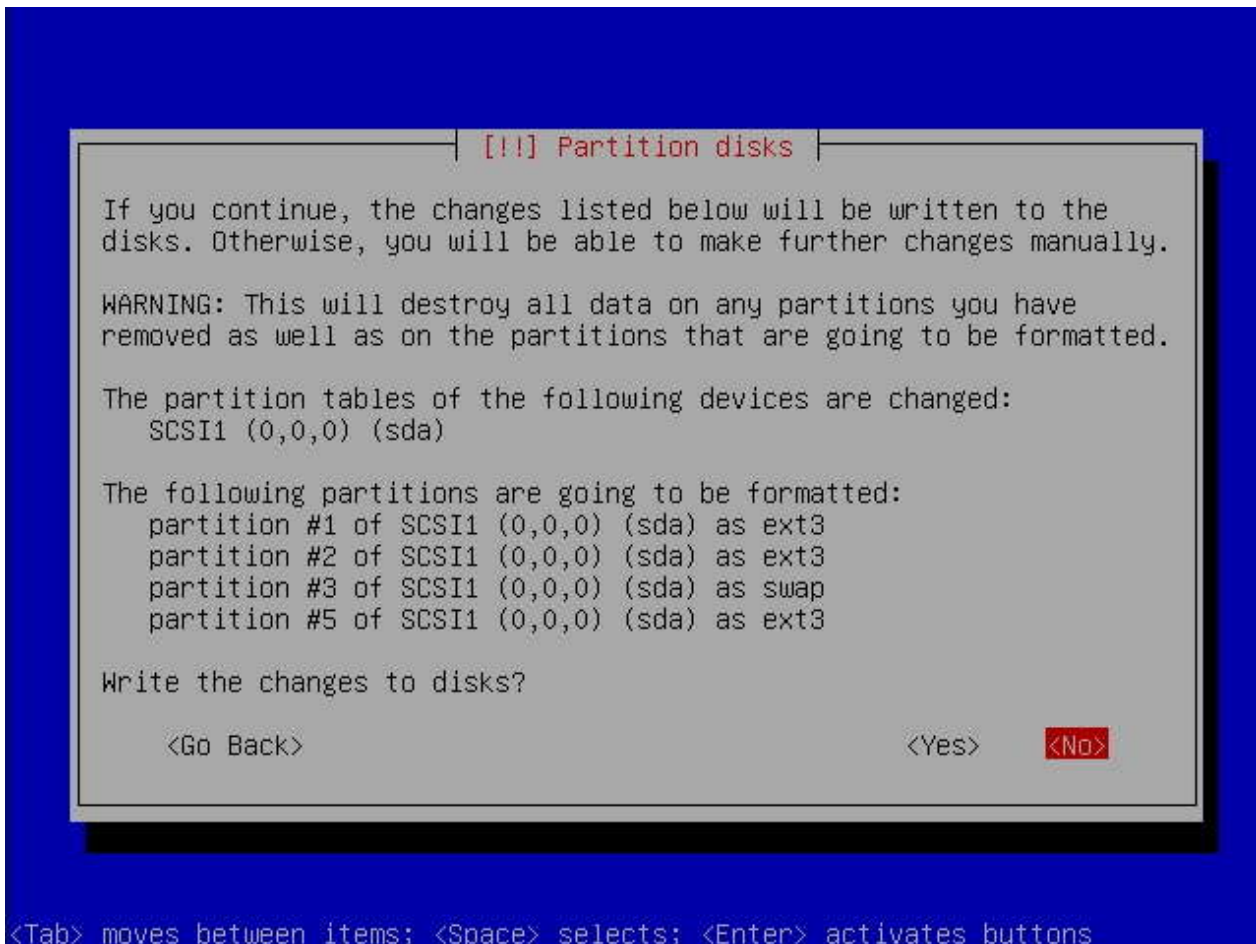


Figura 4.12 Confirmar particiones del disco duro

Paso 11.

El siguiente paso es configurar la cuenta del usuario que será el superusuario o administrador del sistema, se solicita proporcionar el nombre completo del usuario (Figura 4.13).

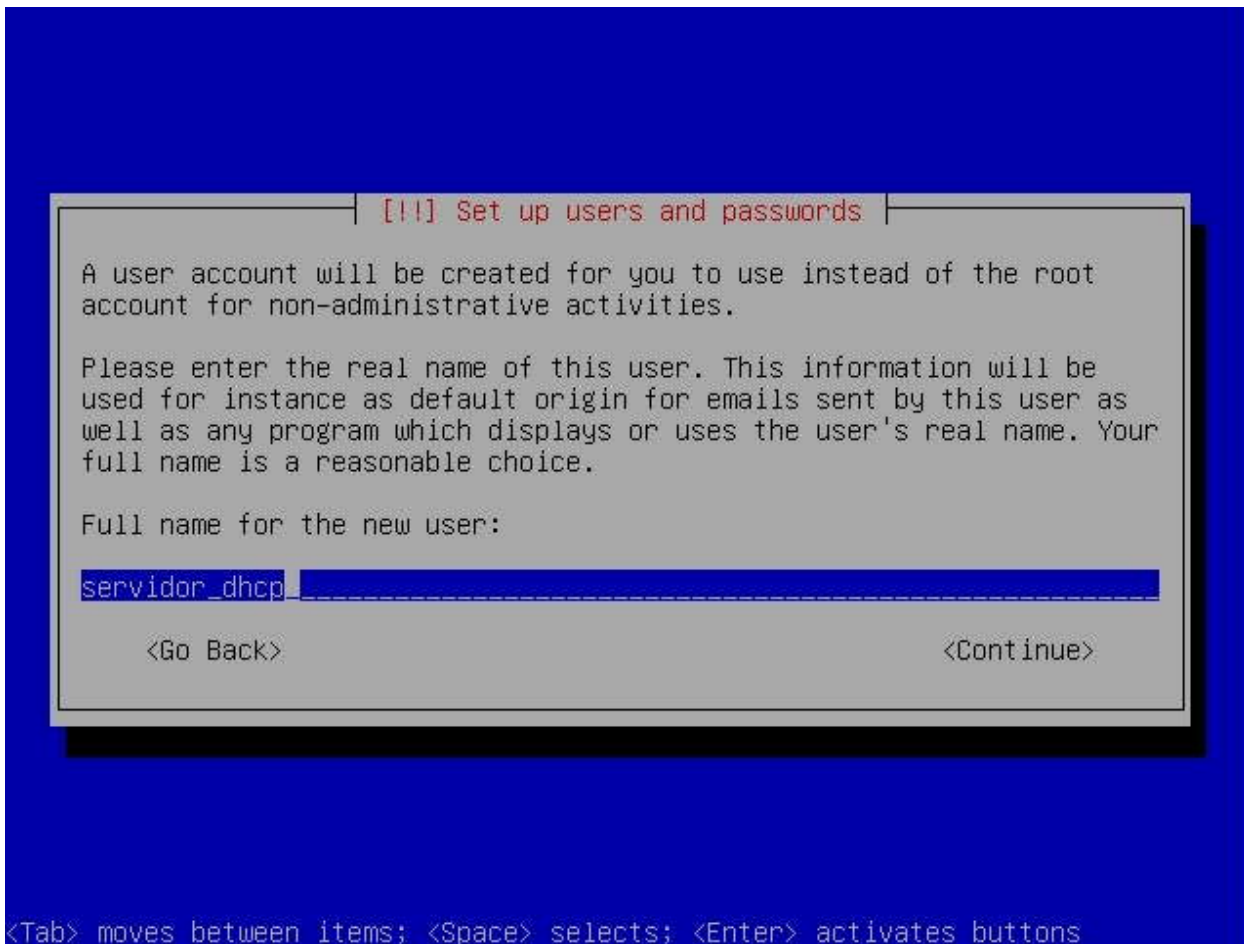


Figura 4.13 Indicar nombre completo del usuario

Paso 12.

A continuación se solicita crear un nombre de usuario para la cuenta, éste debe comenzar con una letra minúscula seguida de cualquier combinación de números y letras minúsculas (Figura 4.14), no se permiten espacios en blanco ni caracteres especiales.

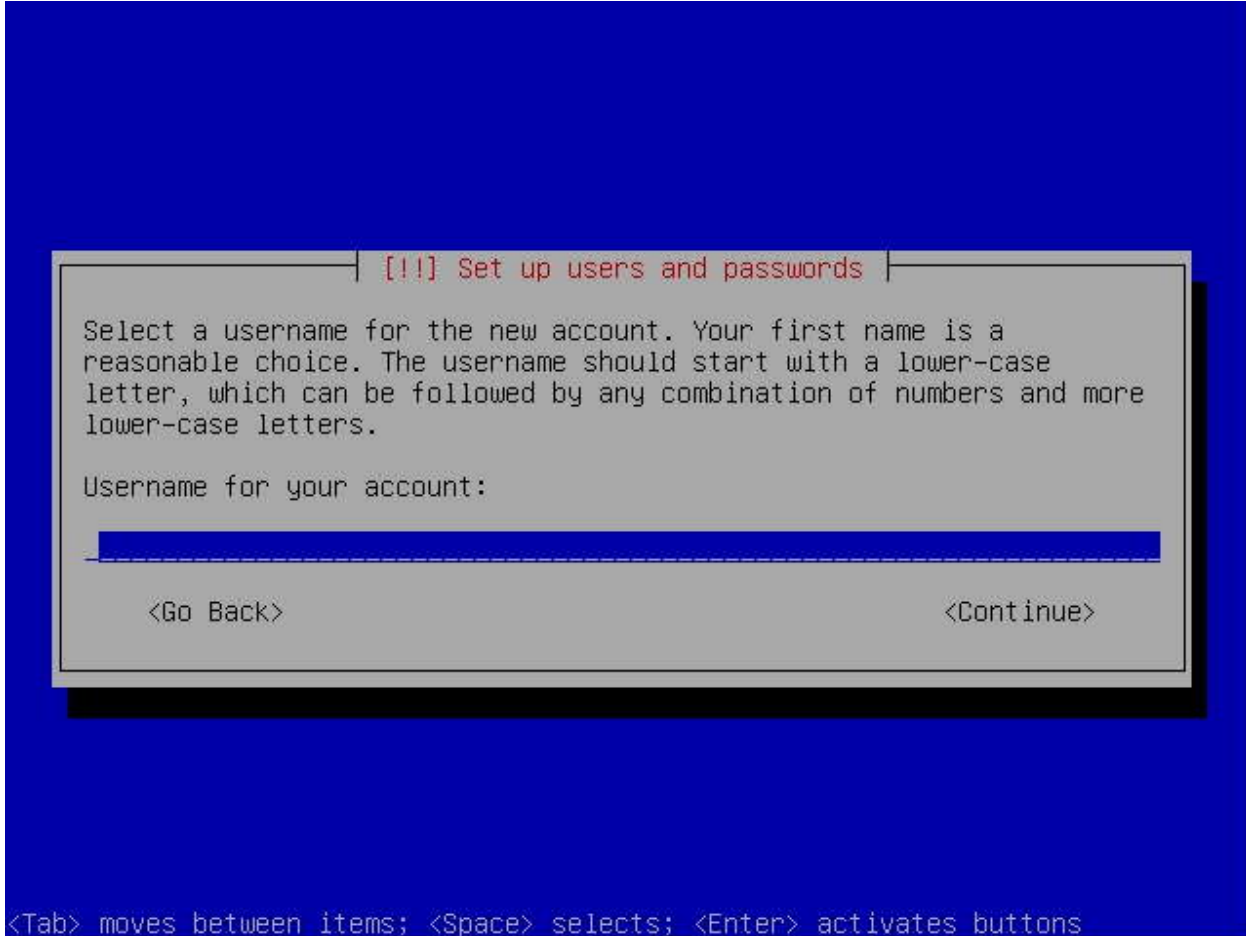


Figura 4.14 Indicar un nombre de usuario

Paso 13.

Por último se debe proporcionar una *password* (contraseña) que solo el usuario debe conocer para ingresar al servidor, ver Figura 4.15. Para mayor protección y seguridad se recomienda crear una contraseña robusta, es decir una combinación de letras, números y signos de puntuación que resulten difíciles de adivinar. Esto nos permite proteger los archivos de cualquier otro usuario u intento de ataques.

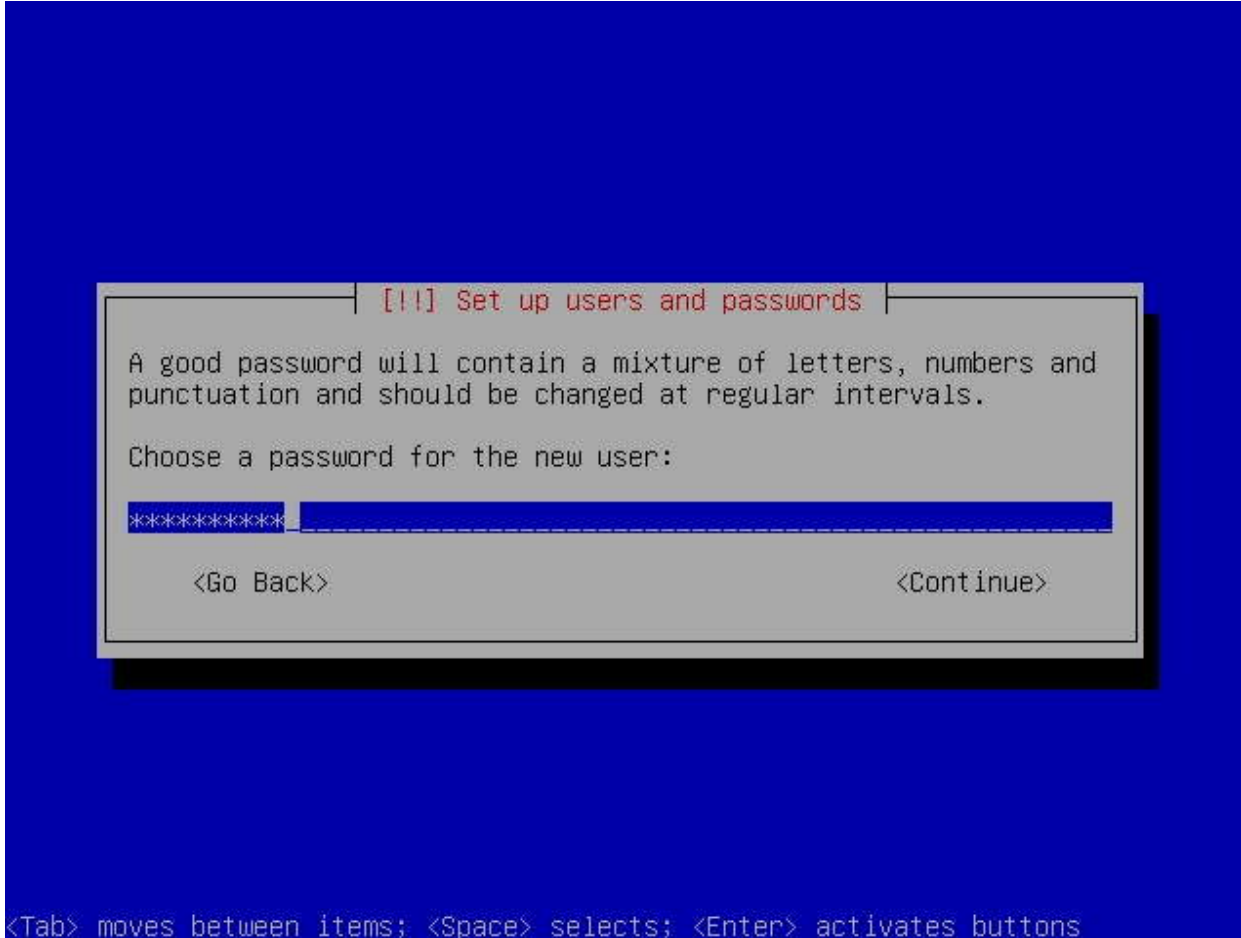


Figura 4.15 Establecer contraseña para el usuario

Paso 14.

El siguiente paso nos permite seleccionar algunos programas predefinidos, seleccionamos la opción del servidor OpenSSH para facilitar la administración del servidor de forma remota desde cualquier parte sin la necesidad de estar físicamente frente a el y le dimos *Continue* (Figura 4.16).

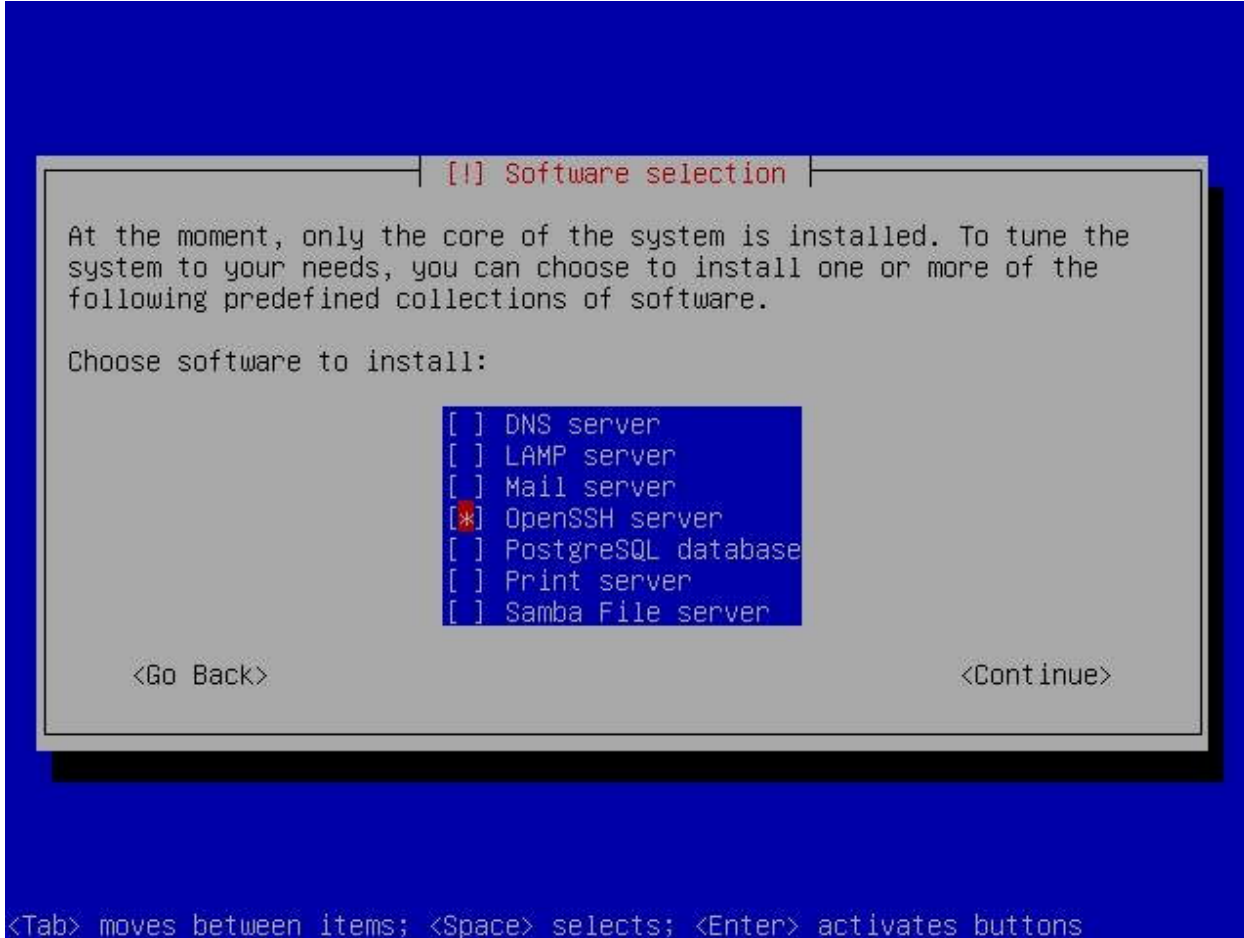


Figura 4.16 Instalar el servidor OpenSSH

Paso 15.

Se muestra la leyenda de que la instalación ha finalizado. Seleccionamos la opción de *Continue* (Figura 4.17) para terminar.

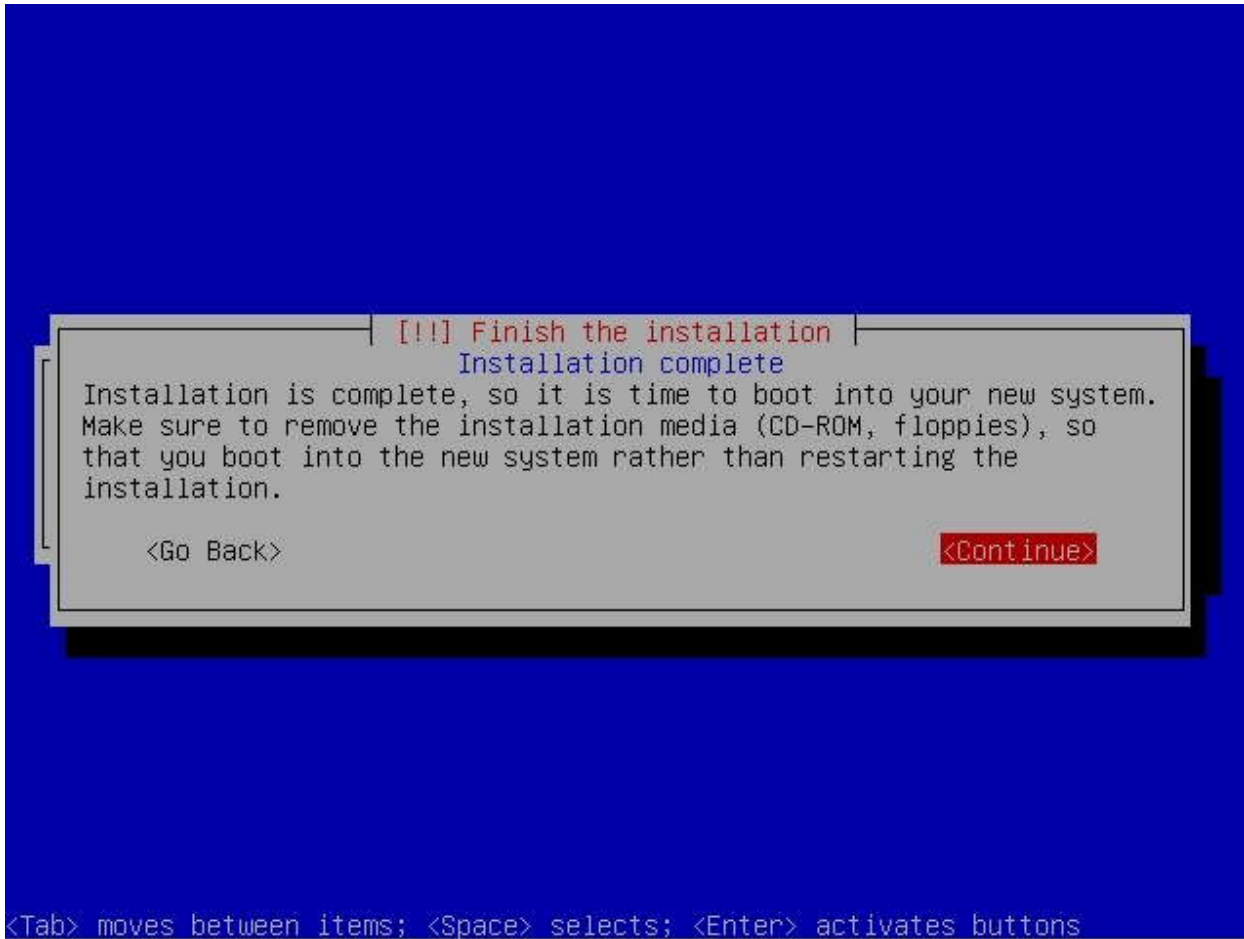


Figura 4.17 Instalación Finalizada

Al finalizar fue necesario retirar el disco de instalación y reiniciar el equipo. Antes de continuar con cualquier otro procedimiento, por cuestiones de estabilidad y seguridad, se recomienda realizar una actualización del sistema a través de la red. Se deben tener permisos de root y escribir en la terminal las siguientes instrucciones:

a) Para actualizar el índice de los paquetes y sincronizar la descarga de las actualizaciones.

```
root@quarks:~# apt-get update
```

b) Para instalar la versión más actualizada de los paquetes.

```
root@quarks:~# apt-get dist-upgrade
```

4.2 Instalación y configuración del servidor DHCP

Es importante señalar que el diseño del servidor QUARKS fue realizado por el encargado de la red del IFUNAM, mi trabajo consistió en actualizar tanto el hardware como el software. Para conocer con más detalle la topología de la Red IFUNAM puede consultar el diagrama del Anexo B.

El equipo utilizado está conformado por el siguiente hardware:

Modelo	Dell Optiplex 360 Minitower
Procesador	Intel Core 2 Duo E7400 a 2.8 GHz
Disco Duro	160 GB 7200 RPM SATA
Memoria RAM	2.0 GB DDR2 Non-ECC SDRAM 800 MHz
Memoria Caché	3 MB L2 Caché
Tarjeta de video	Intel GMA3100
Tarjeta de red	3Com Gigabit 3C200-T

Tabla 4.1 Características físicas del servidor Quarks

El servidor QUARKS es el encargado de proporcionar la configuración de red a los usuarios del Instituto de Física de la UNAM y se ha configurado para realizar una asignación estática de direcciones IP.

Para instalar el servidor fue necesario tener permisos de root y ejecutar en la línea de comandos la siguiente instrucción:

```
root@quarks:~# apt-get install dhcp3-server
```

El siguiente paso al finalizar la instalación fue editar el archivo de configuración, el cual se ubica en /etc/dhcp3/dhcpd.conf. A continuación se muestra un extracto de dicho archivo:

```
# Archivo de configuración del servidor QUARKS
authoritative;
log-facility local7;
default-lease-time 600;
max-lease-time 7200;
```

Subred 7

```
subnet 20.20.7.0 netmask 255.255.255.0 {
option subnet-mask 255.255.255.0;
option broadcast-address 20.20.7.255;
option routers 20.20.7.254;
option domain-name-servers 20.20.204.1, 20.20.10.2;
option domain-name "fisica.unam.mx";

    host 20.20.7.1 {
        hardware ethernet 00:01:03:DC:70:8E;
        fixed-address 20.20.7.1;
    }
    host 20.20.7.2 {
        hardware ethernet 00:DE:AB:BE:EF:00;
        fixed-address 20.20.7.2;
    }
    host 20.20.7.3 {
        hardware ethernet 00:DE:AC:4E:E1:00;
        fixed-address 20.20.7.3;
    }
    host 20.20.7.4 {
        hardware ethernet 00:80:24:61:4F:70;
        fixed-address 20.20.7.4;
    }
    host 20.20.7.5 {
        hardware ethernet 00:04:76:D1:88:19;
        fixed-address 20.20.7.5;
    }
    host 20.20.7.6 {
        hardware ethernet 00:D0:58:63:3E:80;
        fixed-address 20.20.7.6;
    }
    host 20.20.7.7 {
        hardware ethernet 00:04:76:D1:77:9D;
        fixed-address 20.20.7.7;
    }
    host 20.20.7.8 {
        hardware ethernet 00:08:02:BD:C5:EC;
        fixed-address 20.20.7.8;
    }
    host 20.20.7.9 {
        hardware ethernet 00:DE:AD:BE:EF:00;
        fixed-address 20.20.7.9;
    }
    host 20.20.7.10 {
        hardware ethernet 00:DE:AD:12:EF:00;
        fixed-address 20.20.7.10;
    }
```

```
#Subred 209
```

```
subnet 20.20.209.0 netmask 255.255.255.0 {
option subnet-mask 255.255.255.0;
option broadcast-address 20.20.209.255;
option routers 20.20.209.254;
option domain-name-servers 20.20.204.1, 20.20.10.2;
option domain-name "fisica.unam.mx";

    host 20.20.209.1 {
        hardware ethernet 00:04:76:D1:8B:EF;
        fixed-address 20.20.209.1;
    }
    host 20.20.209.2 {
        hardware ethernet 00:DE:AD:BE:EF:00;
        fixed-address 20.20.209.2;
    }
    host 20.20.209.3 {
        hardware ethernet 00:0D:87:35:16:E9;
        fixed-address 20.20.209.3;
    }
    host 20.20.209.4 {
        hardware ethernet 00:DE:AD:BE:EF:00;
        fixed-address 20.20.209.4;
    }
    host 20.20.209.5 {
        hardware ethernet 00:DE:AD:B2:51:00;
        fixed-address 20.20.209.5;
    }
    host 20.20.209.6 {
        hardware ethernet 00:DE:AD:BE:EF:00;
        fixed-address 20.20.209.6;
    }
    host 20.20.209.7 {
        hardware ethernet 00:DE:AD:BE:EF:00;
        fixed-address 20.20.209.7;
    }
    host 20.20.209.8 {
        hardware ethernet 00:13:D4:4F:D9:56;
        fixed-address 20.20.209.8;
    }
    host 20.20.209.9 {
        hardware ethernet 00:DE:11:B3:EF:00;
        fixed-address 20.20.209.9;
    }
    host 20.20.209.10 {
        hardware ethernet 00:DE:AD:B2:F5:00;
        fixed-address 20.20.209.10;
    }
```

A continuación se editó el archivo `/etc/default/dhcp3-server` para indicar las interfaces de red que deben ser escuchada. Se buscó la línea en la que se leía `INTERFACES=""` para realizar el siguiente cambio:

```
INTERFACES="eth0 eth1"
```

En este caso la interfaz `eth0` es asignada a la subred 7 y la interfaz `eth1` a la subred 209, el siguiente paso es editar el archivo `/etc/network/interfaces` para especificar las direcciones IP que se asignarán a los dispositivos de red disponibles en el sistema. Se agregaron las siguientes líneas:

```
auto eth0
iface eth0 inet static
address 20.20.7.247
netmask 255.255.255.0
gateway 20.20.7.254
```

```
auto eth1
iface eth1 inet static
address 20.20.209.248
netmask 255.255.255.0
gateway 20.20.209.254
```

Por último es necesario editar el archivo `/etc/resolv.conf` el cual permite resolver los nombres de dominio en Internet, se agregaron las siguientes líneas:

```
search fisica.unam.mx
nameserver 20.20.204.1
nameserver 20.20.10.2
```

Se guardaron los cambios realizados, y para finalizar la configuración del servidor DHCP fue necesario reiniciar el *demonio*²⁶ ejecutando los siguientes comandos:

a) Para detener el servidor.

```
root@quarks:~# /etc/init.d/dhcp3-server stop
```

²⁶ Un *demonio* o *daemon* es, por sus siglas en inglés, acrónimo de *Disk And Execution MONitor*, el cual es un proceso que se ejecuta en background, es decir que no requiere interacción directa con el usuario.

b) Para iniciar el servidor.

```
root@quarks:~# /etc/init.d/dhcp3-server start
```

4.3 Instalación y configuración del software para Monitoreo y Gráfica

En el capítulo anterior se explicó el funcionamiento y el objetivo de cada uno de los programas a utilizar, por lo que a continuación solo se detallará el procedimiento de instalación y configuración a seguir para cada uno de ellos.

4.3.1 Instalación y configuración de IPFM

Teniendo los permisos de root se ejecutó el siguiente comando en la terminal:

```
root@quarks:~# apt-get install ipfm
```

Al terminar la instalación se editó el archivo de configuración localizado en /etc/ipfm.conf como se muestra a continuación:

```
# Archivo de configuracion de IPFM para el servidor QUARKS

##### PRIMER REGISTRO DE CONFIGURACION #####

# Iniciar monitoreo en la red 20.20.7.0
LOG 20.20.7.0/255.255.255.0

FILENAME "/home/nglez/ipfm-dump/20.20.7.0/%Y_%d_%m/%H_%M"
DUMP EVERY 5 minutes
# Limpiar estadísticas a las 00:05 UTC
CLEAR EVERY 1 day AFTER 5 minutes
SORT TOTAL
NORESOLVE

##### SEGUNDO REGISTRO DE CONFIGURACION #####

NEWLOG

# Iniciar monitoreo en la red 20.20.209.0
LOG 20.20.209.0/255.255.255.0
```

```
FILENAME "/home/nglez/ipfm-dump/20.20.209.0/%Y_%d_%m/%H_%M"  
  
DUMP EVERY 5 minutes  
# Limpiar estadísticas a las 00:05 UTC  
CLEAR EVERY 1 day AFTER 5 minutes  
SORT TOTAL  
NORESOLVE
```

La configuración anterior se encarga de realizar las siguientes tareas:

1. Monitorear el tráfico de red cada 5 minutos.
2. Guardar el archivo que se generó con la información que representa el tráfico de cada host en la red.
3. Para evitar que el proceso de producción de cada archivo de salida sea lento, se activo el DNS inverso, es decir, que no se resuelve el nombre de dominio.

4.3.2 Instalación de RRDTool

La instalación de esta herramienta fue muy sencilla, ya que no fue necesario realizar ninguna configuración especial, solo se ejecutó en la terminal el siguiente comando:

```
root@quarks:~# apt-get install rrdtool librrd4
```

Es necesario instalar la librería librrd4 ya que permite la manipulación y el acceso a los archivos RRD.

4.4 Ejecución de scripts para el monitoreo de la red

Después de instalar y configurar el software necesario, se procedió con la ejecución de los scripts desarrollados para realizar la comunicación entre el programa colector IPFM y el programa graficador RRDTool.

4.4.1 Scripts para generar listas de hosts.

Se crearon diversos scripts en bash para administrar las tareas por separado de los dos sectores de red que administra la Secretaría Técnica de Cómputo y Telecomunicaciones del Instituto de Física de la UNAM.

El primer paso fue generar una lista de los hosts que originan tráfico en la red, nos ubicamos en las carpetas correspondientes donde están los scripts y teniendo permisos de root se ejecutaron los siguientes comandos:

a) Para la red 7

```
root@quarks:~# ./create_hostlist_7
```

b) Para la red 209

```
root@quarks:~# ./create_hostlist_209
```

Éstos se encargaron de leer cada uno de los archivos de texto generados por IPFM y de separar la información en columnas, las cuales son identificadas al estar separadas por cada espacio presente antes del salto de línea. Como ejemplo podemos observar la Figura 4.18, una línea de datos termina hasta que el intérprete encuentra un salto de línea, los datos son separados en columnas al encontrar uno o más espacios en blanco entre ellos. Cada línea de datos es separada en columnas de acuerdo a los espacios en blanco encontrados, al llegar a un salto de línea comienza nuevamente el conteo de columnas para la siguiente línea. En este caso se puede observar que se encontraron 4 columnas.

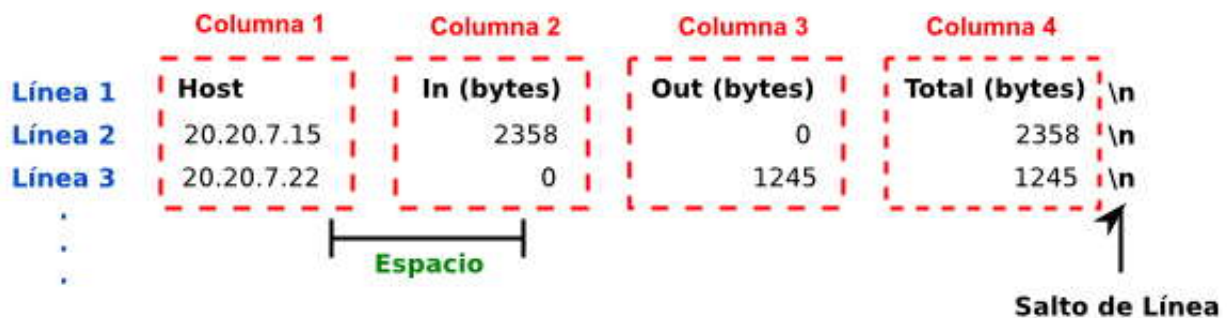


Figura 4.18 Ejemplo de la división en columnas de un texto

Para nuestro propósito los datos de la primera columna eran los de interés, por lo que se copió la primera columna de cada archivo y se guardó en un archivo de texto temporal. Posteriormente los datos recolectados fueron organizados alfabéticamente y comparados para eliminar la información duplicada, al final se realiza una última revisión para eliminar símbolos de comentarios.

El archivo de texto final nos muestra una lista de los hosts que fueron localizados durante el monitoreo de la red, la cual es de utilidad para otros procedimientos.

Código del script create_hostlist_7

```
#!/bin/bash

# ooooooooooooooooooooooooooooo
# ° Crear lista de hosts °
# ooooooooooooooooooooooooooooo

RED7=`find /home/nglez/ipfm-dump/20.20.7.0/ -type f` ; # lee archivos de directorio especificado
for i in $RED7; do
  DIR=`awk '{print $1}' $RED7 >> tmphost1` ; # extrae la primera columna de todos los archivos
  sort -u tmphost1 > tmphost2 ; # ordena alfabeticamente y elimina repetidos
  sed '/^\#/d' tmphost2 > hostlist_7 ; # elimina simbolos de comentario
done
rm tmphost* ;
```

Código del script create_hostlist_209

```
#!/bin/bash

# ooooooooooooooooooooooooooooo
# ° Crear lista de hosts °
# ooooooooooooooooooooooooooooo

RED209=`find /home/nglez/ipfm-dump/20.20.209.0/ -type f` ; # lee archivos de directorio especificado
for i in $RED209; do
  DIR=`awk '{print $1}' $RED209 >> tmphost1` ; # extrae la primera columna de todos los archivos
  sort -u tmphost1 > tmphost2 ; # ordena alfabeticamente y elimina repetidos
  sed '/^\#/d' tmphost2 > hostlist_209 ; # elimina simbolos de comentario
done
rm tmphost* ;
```

4.4.2 Scripts para generar las Bases de Datos Circulares

El siguiente paso fue crear la base de datos para cada host existente en la lista generada en el paso anterior. Nuevamente se ingresó a las carpetas correspondientes y se ejecutaron los siguientes comandos:

a) Para la red 7

```
root@quarks:~# ./create_bases7
```

b) Para la red 209

```
root@quarks:~# ./create_bases209
```

En cada base de datos circular se definió:

- Un tiempo de volcado de información de 300 segundos, es decir que se ingresarán datos cada 5 minutos.
- Se especificaron 3 fuentes de datos (DS), para los bytes que entran (In), los que salen (Out) y la suma de ellos (Total).
- Se definen los archivos round robin (RRA) y sus parámetros para indicar el tipo de datos que queremos almacenar.

Es importante recordar que cada base de datos es creada para almacenar la información de un año, al finalizar éste periodo, los datos se sobrescribirán en los más viejos.

Código del script create_bases7

```
#!/bin/bash

# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Seccion para crear la base RRD de los hosts nuevos °
# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

HOST=`awk '{print $1}' hostlist_7` ;      # se extrae el nombre de cada host

mkdir -p -m 755 7_rrd_host ;           # crea un directorio si no existe

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host" ;

for i in $HOST; do

  rrdtool create $HOSTDIR/$i.rrd -s 300 \
    DS:In:GAUGE:600:0:U \
    DS:Out:GAUGE:600:0:U \
    DS:Total:GAUGE:600:0:U \
    RRA:AVERAGE:0.5:1:600 \
    RRA:AVERAGE:0.5:6:700 \
```

```
RRA:AVERAGE:0.5:24:744 \
RRA:AVERAGE:0.5:288:797 \
RRA:MIN:0.5:1:600 \
RRA:MIN:0.5:6:700 \
RRA:MIN:0.5:24:775 \
RRA:MIN:0.5:288:797 \
RRA:MAX:0.5:1:600 \
RRA:MAX:0.5:6:700 \
RRA:MAX:0.5:24:775 \
RRA:MAX:0.5:288:797
```

done

Código del script create_bases209

```
#!/bin/bash

# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Seccion para crear la base RRD de los hosts nuevos °
# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

HOST=`awk '{print $1}' hostlist_209` ;      # se extrae el nombre de cada host

mkdir -p -m 755 209_rrd_host ;           # crea un directorio si no existe y le da permisos

HOSTDIR="/home/nglez/scripts-ipfm/209_red/209_rrd_host" ;

for i in $HOST; do

    rrdtool create $HOSTDIR/$i.rrd -s 300 \
        DS:In:GAUGE:600:0:U \
        DS:Out:GAUGE:600:0:U \
        DS:Total:GAUGE:600:0:U \
        RRA:AVERAGE:0.5:1:600 \
        RRA:AVERAGE:0.5:6:700 \
        RRA:AVERAGE:0.5:24:744 \
        RRA:AVERAGE:0.5:288:797 \
        RRA:MIN:0.5:1:600 \
        RRA:MIN:0.5:6:700 \
        RRA:MIN:0.5:24:775 \
        RRA:MIN:0.5:288:797 \
        RRA:MAX:0.5:1:600 \
        RRA:MAX:0.5:6:700 \
        RRA:MAX:0.5:24:775 \
        RRA:MAX:0.5:288:797

done
```

4.4.3 Scripts para ingresar información a las bases de datos

Los siguientes scripts se encargan de llenar con información las bases de datos, debido a la frecuencia necesaria para su ejecución fue necesario automatizar esta tarea mediante la programación de un cron, dicho procedimiento se explicará más adelante.

Los scripts en bash `rrd_create_7` y `rrd_create_209` son los más importantes para el correcto funcionamiento del presente sistema, deben ejecutarse cada 5 minutos y se encargan de realizar diversas tareas:

1. La primera sección se encarga de crear una nueva lista de hosts con la información obtenida del archivo generado por IPFM durante los últimos 5 minutos. Posteriormente se comparan las dos listas (la lista principal que generamos en un principio y la actual) y se obtienen los nombres de nuevos *hosts*, si es que los hay, para agregarlos a la lista principal.
2. La segunda sección realiza la tarea de crear las bases de datos circulares de los nuevos hosts encontrados. Si no hay nuevos continúa con la siguiente parte.
3. La tercera sección está encargada de ingresar la información recolectada a cada una de las bases de datos circulares correspondientes a los *hosts* que generaron tráfico en los últimos 5 minutos. En este caso se hace uso de las columnas restantes del archivo, ya que representan las fuentes de datos que se especificaron con anterioridad (columna 2 = In, columna 3 = Out, Columna 4 = Total).
4. Por último se ejecuta el script encargado de crear las gráficas que muestran la actividad de los *hosts*.

Código del script rrd_create_7

```
#!/bin/bash

# ooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Seccion para crear datos de cada host °
# ooooooooooooooooooooooooooooooooooooooooooooooooooooo

BUSCA=`find /home/nglez/ipfm-dump/20.20.7.0/ -type f -mmin -5` ; # lee los archivos del directorio
# especificado

for i in $BUSCA; do

    DIR=`awk '{print $1}' $BUSCA >> tmphost1` ; # extrae la primera columna de todos los archivos
    sort -u tmphost1 > tmphost2 ; # ordena alfabeticamente y elimina repetidos
    sed '/^\#/d' tmphost2 > tmphost3 ; # elimina simbolos de comentario

done

diff --suppress-common-lines hostlist_7 tmphost3 > dif ; # compara para encontrar hosts nuevos
awk '/^>/ {print}' dif > dif1 ; # copia en un archivo los hosts
nuevos
sed 's/> //' dif1 > dif ; # elimina caracteres extras

cat hostlist_7 dif > tmphost1 ; # agrega los hosts nuevos a la lista
sort -u tmphost1 > hostlist_7 ; # ordena alfabeticamente los hosts

# ooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Seccion para crear la base RRD de los hosts nuevos °
# ooooooooooooooooooooooooooooooooooooooooooooooooooooo

NEWHOST=`awk '{print $1}' dif` ; # se extrae el nombre de cada host nuevo

mkdir -p -m 755 7_rrd_host ; # crea un directorio si no existe

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host" ;

for i in $NEWHOST; do
    rrdtool create $HOSTDIR/$i.rrd -s 300 \
        DS:In:GAUGE:600:0:U \
        DS:Out:GAUGE:600:0:U \
        DS:Total:GAUGE:600:0:U \
        RRA:AVERAGE:0.5:1:600 \
        RRA:AVERAGE:0.5:6:700 \
        RRA:AVERAGE:0.5:24:744 \
        RRA:AVERAGE:0.5:288:797 \
        RRA:MIN:0.5:1:600 \
        RRA:MIN:0.5:6:700 \
```



```

RRA:MIN:0.5:24:775 \
RRA:MIN:0.5:288:797 \
RRA:MAX:0.5:1:600 \
RRA:MAX:0.5:6:700 \
RRA:MAX:0.5:24:775 \
RRA:MAX:0.5:288:797
done

# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Sección para la Introduccion de datos a las RRD °
# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

HOST=`awk '{print $1}' hostlist_7` ; # Obtenemos el nombre de cada uno de los hosts

for i in $HOST; do # Guardamos en un archivo los datos que extraemos de cada host
  grep -r "$i" "$BUSCA" >> datos ;
done

BASE=`awk '{print $1}' datos` ;

CONT=1 ; # Inicio contador
LINEAS=`wc -l datos | awk '{print $1}'` ; # obtener número total de líneas en el archivo texto

while [ $CONT -le $LINEAS ]; do # El contador debe ser menor o igual que el total de líneas

  j=`head -n $CONT datos | tail -1 > doc` ; # Obtengo línea por línea del texto y la guardo en doc

  NOM=`awk '{print $1}' doc` ;
  I=`awk '{print $2}' doc` ;
  O=`awk '{print $3}' doc` ;
  T=`awk '{print $4}' doc` ;

  for i in $HOST; do

    if [[ $i == "$NOM" ]]; then

      rrdtool update "$HOSTDIR/$i".rrd N:$I:$O:$T ; # Para actualizar la RRD de cada host

    fi

  done

  let CONT=CONT+1 ;

done

rm tmp* ;
rm dif* ;
rm datos ;

```

```
rm doc ;

sleep 10 ;

./home/nglez/scripts-ipfm/7_red/graficas_7 ;           # Ejecuta el scripts para generar las graficas
```

Código del script rrd_create_209

```
#!/bin/bash

# oooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Seccion para crear datos de cada host °
# oooooooooooooooooooooooooooooooooooooooooooooooooooooo

BUSCA=`find /home/nglez/ipfm-dump/20.20.209.0/ -type f -mmin -5` ; # lee los archivos del
                                                                    # directorio especificado

for i in $BUSCA; do

    DIR=`awk '{print $1}' $BUSCA >> tmphost1` ;           # extrae la primera columna de todos los archivos
    sort -u tmphost1 > tmphost2 ;                       # ordena alfabeticamente y elimina repetidos
    sed '/^#\#/d' tmphost2 > tmphost3 ;                 # elimina simbolos de comentario

done

diff --suppress-common-lines hostlist_209 tmphost3 > dif ; # encontrar hosts nuevos
awk '/^>/ {print}' dif > dif1 ;                         # copia en un archivo los hosts
nuevos
sed 's/> //' dif1 > dif ;                               # elimina caracteres extras

cat hostlist_209 dif > tmphost1 ;                       # agrega los hosts nuevos a la lista
sort -u tmphost1 > hostlist_209 ;                      # ordena alfabeticamente los hosts

# oooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Seccion para crear la base RRD de los hosts nuevos °
# oooooooooooooooooooooooooooooooooooooooooooooooooooooo

NEWHOST=`awk '{print $1}' dif` ;                       # se extrae el nombre de cada host

mkdir -p -m 755 209_rrd_host ;                        # crea un directorio si no existe

HOSTDIR="/home/nglez/scripts-ipfm/209_red/209_rrd_host" ;

for i in $NEWHOST; do
```

```
rrdtool create $HOSTDIR/$i.rrd -s 300 \
  DS:In:GAUGE:600:0:U \
  DS:Out:GAUGE:600:0:U \
  DS:Total:GAUGE:600:0:U \
  RRA:AVERAGE:0.5:1:600 \
  RRA:AVERAGE:0.5:6:700 \
  RRA:AVERAGE:0.5:24:744 \
  RRA:AVERAGE:0.5:288:797 \
  RRA:MIN:0.5:1:600 \
  RRA:MIN:0.5:6:700 \
  RRA:MIN:0.5:24:775 \
  RRA:MIN:0.5:288:797 \
  RRA:MAX:0.5:1:600 \
  RRA:MAX:0.5:6:700 \
  RRA:MAX:0.5:24:775 \
  RRA:MAX:0.5:288:797
```

done

```
# oooooooooooooooooooooooooooooooooooooooooooooooooooooo
# ° Sección para la Introduccion de datos a las RRD °
# oooooooooooooooooooooooooooooooooooooooooooooooooooooo
```

```
HOST=`awk '{print $1}' hostlist_209` ; # obtenemos el nombre de cada uno de los hosts
```

```
for i in $HOST; do # guardar en un archivo los datos extraidos de cada host
  grep -r "$i" "$BUSCA" >> datos ;
done
```

```
BASE=`awk '{print $1}' datos` ;
```

```
CONT=1 ; # inicio contador
LINEAS=`wc -l datos | awk '{print $1}'` ; # obtengo el número total de líneas en el archivo texto
```

```
while [ $CONT -le $LINEAS ]; do # el contador debe ser menor o igual que el total de líneas
```

```
  j=`head -n $CONT datos | tail -1 > doc` ; # obtengo línea por línea del texto y la guardo en doc
```

```
  NOM=`awk '{print $1}' doc` ;
  I=`awk '{print $2}' doc` ;
  O=`awk '{print $3}' doc` ;
  T=`awk '{print $4}' doc` ;
```

```
for i in $HOST; do
```

```
  if [[ $i == "$NOM" ]]; then
    rrdtool update "$HOSTDIR/$i".rrd N:$I:$O:$T ; # para actualizar la RRD de cada host
  fi
```

```
done

let CONT=CONT+1 ;

done

rm tmp* ;
rm dif* ;
rm datos ;
rm doc ;

sleep 10 ;

./home/nglez/scripts-ipfm/209_red/graficas_209 ; # ejecuta el script para crear las graficas
```

4.4.4 Scripts para generar gráficas

Las gráficas que solicita la Secretaría Técnica de Cómputo y Telecomunicaciones deben permitir la visualización diaria, semanal, mensual y anual de la actividad en la red del Instituto de Física de la UNAM.

Para poder realizar dicha tarea fue necesario crear una variedad de scripts que se ejecutaran en periodos de tiempo distintos.

Gracias a que RRDTool nos da la facilidad de configurar hasta los detalles más sencillos en las gráficas es posible mostrar en cada una de ellas información relevante y de importancia para nosotros como:

- Nombre del *host*.
- Tipo de gráfica (diaria, semanal, mensual o anual).
- En casos específicos mostrar la fecha y hora de creación.
- El promedio y la cantidad máxima de bytes que entran.
- El promedio y la cantidad máxima de bytes que salen.
- El promedio y la cantidad máxima de bytes totales.

Cada gráfica generada es un archivo de imagen en formato GIF, y para crearlas fue necesario especificar características especiales como el ancho y la altura de cada imagen, el rango del tiempo que queremos mostrar, el archivo RRA de donde proviene la información,

etiquetas como el título y nombres de los ejes, el color de los elementos (marco, fondo, ejes, cuadrícula, flechas). También se definieron las funciones para recuperar la información, el tipo de imagen a generar para crear el reporte (en este caso una línea) y su color, así como la leyenda que representará a cada línea y el formato en el que queremos presentar los datos.

Los scripts que a continuación se presentan, se encargan de crear gráficas en la que es posible visualizar la actividad de cada host durante las últimas 8 horas. Dichas gráficas deben actualizarse cada 5 minutos.

Código del script graficas_7

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/netgraphs/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/7_red/hostlist_7`;

for i in $HOST; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-480m --width 600 --height 250 \
--title "HOST $i" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\n" \
COMMENT:"Actualizando...\r"

done
```

Código del script graficas_209

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/209_red/209_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/netgraphs/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/209_red/hostlist_209` ;

for i in $HOST ; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-480m --width 600 --height 250 \
--title "HOST $i" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\n" \
COMMENT:"Actualizando...\r"

done
```

Para representar la información correspondiente a la actividad de los hosts en 1 día se realizan 5 gráficas. Cuatro de ellas representan un cuarto de día con los datos de las 6 horas anteriores a su creación, es decir que se tendrá una gráfica con información de las 0 a las 6 horas, otra con información de las 6 a las 12 horas, una tercera con información de las 12 a las 18 horas y la cuarta con información de las 18 a las 24 horas. La gráfica restante representa los datos de las últimas 24 horas.

Los siguientes *scripts* generan las gráficas que muestran datos cada 6 horas. Adicionalmente y para evitar confusiones se muestra la información de la hora y la fecha en que fueron creadas.

Código del script grafica1_x6horas para la red 7

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/daily_h/1/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/7_red/hostlist_7`;

HORA=`date +%D--%R`;

for i in $HOST; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-6h --width 600 --height 250 \
--title "HOST $i - $HORA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In" \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out" \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total" \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...r"

done
```

Código del script grafica2_x6horas para la red 7

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/daily_4/4/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/7_red/hostlist_7`;

HORA=`date +%D--%R`;

for i in $HOST; do
```

```

rrdtool graph $GRAPHDIR/$i.png --end now --start end-6h --width 600 --height 250 \
--title "HOST $i - $HORA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In          " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out
" \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...r"

done

```

Código del script grafica3_x6horas para la red 7

```

#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/daily_h/3/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/7_red/hostlist_7`;

HORA=`date +%D--%R`;

for i in $HOST; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-6h --width 600 --height 250 \
--title "HOST $i - $HORA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In          " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out
" \

```



```
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...r"
```

done

Código del script grafica4_x6horas para la red 7

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/daily_4/4/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/7_red/hostlist_7`;

HORA=`date +%D--%R`;

for i in $HOST; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-6h --width 600 --height 250 \
--title "HOST $i - $HORA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...r"
```

done

Código del script grafica1_x6horas para la red 209

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/209_red/209_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/daily_h/1/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/209_red/hostlist_209`;

HORA=`date +%D--%R`;

for i in $HOST ; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-6h --width 600 --height 250 \
--title "HOST $i - $HORA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\n" \
COMMENT:"Actualizando...\r"

done
```

Código del script grafica2_x6horas para la red 209

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/209_red/209_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/daily_h/2/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/209_red/hostlist_209`;

HORA=`date +%D--%R`;

for i in $HOST ; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-6h --width 600 --height 250 \
```

```
--title "HOST $i - $HORA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%" \
GPRINT:hostIn:AVERAGE:"Average %lg%\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%" \
GPRINT:hostOut:AVERAGE:"Average %lg%\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%" \
GPRINT:hostTotal:AVERAGE:"Average %lg%\n" \
COMMENT:"Actualizando...\r"
```

done

Código del script grafica3_x6horas para la red 209

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/209_red/209_rrd_host";
GRAPHDIR="/var/www/monitor/public/images/daily_h/3/";
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/209_red/hostlist_209`;

HORA=`date +%D-%R`;

for i in $HOST ; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-6h --width 600 --height 250 \
--title "HOST $i - $HORA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%" \
GPRINT:hostIn:AVERAGE:"Average %lg%\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%" \
GPRINT:hostOut:AVERAGE:"Average %lg%\n" \
```

```
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...\\r"
```

done

Código del script grafica4_x6horas para la red 209

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/7_red/7_rrd_host" ;
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/7_red/hostlist_7` ;

#####
## Instruccion para crear las graficas correspondientes a un dia ##
#####

GRAPHDIR="/var/www/monitor/public/images/daily/" ;

for i in $HOST ; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-1d --width 600 --height 250 \
--title "HOST $i - GRAFICA DIARIA" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...\\r"

done
```

```
#####  
## Instruccion para crear las graficas correspondientes a una semana ##  
#####
```

```
GRAPHDIR="/var/www/monitor/public/images/weekly/";
```

```
for i in $HOST ; do
```

```
rrdtool graph $GRAPHDIR/$i.png --end now --start end-1w --width 600 --height 250 \  
--title "HOST $i - GRAFICA SEMANAL" --vertical-label "Bits por segundo" --watermark ZETAGG \  
--color CANVAS#000000 \  
--color BACK#000000 \  
--color FRAME#000000 \  
--color FONT#FFFFFF \  
--color ARROW#E70C0C \  
--lower-limit=0 \  
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \  
GPRINT:hostIn:MAX:"Max %lg%s" \  
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \  
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \  
GPRINT:hostOut:MAX:"Max %lg%s" \  
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \  
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \  
GPRINT:hostTotal:MAX:"Max %lg%s" \  
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \  
COMMENT:"Actualizando...\\r"
```

```
done
```

```
#####  
## Instruccion para crear las graficas correspondientes a un mes ##  
#####
```

```
GRAPHDIR="/var/www/monitor/public/images/monthly/";
```

```
for i in $HOST ; do
```

```
rrdtool graph $GRAPHDIR/$i.png --end now --start end-1m --width 600 --height 250 \  
--title "HOST $i - GRAFICA MENSUAL" --vertical-label "Bits por segundo" --watermark ZETAGG \  
--color CANVAS#000000 \  
--color BACK#000000 \  
--color FRAME#000000 \  
--color FONT#FFFFFF \  
--color ARROW#E70C0C \  
--lower-limit=0 \  
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \  
GPRINT:hostIn:MAX:"Max %lg%s" \  
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \  
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \  
GPRINT:hostOut:MAX:"Max %lg%s" \  
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \  
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \  
GPRINT:hostTotal:MAX:"Max %lg%s" \  
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \  
COMMENT:"Actualizando...\\r"
```

```
GPRINT:hostIn:AVERAGE:"Average %lg%os\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out
" \
GPRINT:hostOut:MAX:"Max %lg%os" \
GPRINT:hostOut:AVERAGE:"Average %lg%os\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%os" \
GPRINT:hostTotal:AVERAGE:"Average %lg%os\\n" \
COMMENT:"Actualizando...r"
```

done

```
#####
## Instruccion para crear las graficas correspondientes a un anio ##
#####
```

```
GRAPHDIR="/var/www/monitor/public/images/yearly/" ;
```

```
for i in $HOST ; do
```

```
rrdtool graph $GRAPHDIR/$i.png --end now --start end-1y --width 600 --height 250 \
--title "HOST $i - GRAFICA ANUAL" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%os" \
GPRINT:hostIn:AVERAGE:"Average %lg%os\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out
" \
GPRINT:hostOut:MAX:"Max %lg%os" \
GPRINT:hostOut:AVERAGE:"Average %lg%os\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%os" \
GPRINT:hostTotal:AVERAGE:"Average %lg%os\\n" \
COMMENT:"Actualizando...r"
```

done

Código del script graficas_completas para la red 209

```
#!/bin/bash

HOSTDIR="/home/nglez/scripts-ipfm/209_red/209_rrd_host" ;
HOST=`awk '{print $1}' /home/nglez/scripts-ipfm/209_red/hostlist_209` ;

#####
## Instruccion para crear las graficas correspondientes a un dia ##
#####

GRAPHDIR="/var/www/monitor/public/images/daily/" ;

for i in $HOST ; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-1d --width 600 --height 250 \
  --title "HOST $i - GRAFICA DIARIA" --vertical-label "Bits por segundo" --watermark ZETAGG \
  --color CANVAS#000000 \
  --color BACK#000000 \
  --color FRAME#000000 \
  --color FONT#FFFFFF \
  --color ARROW#E70C0C \
  --lower-limit=0 \
  DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In" \
  GPRINT:hostIn:MAX:"Max %lg%s" \
  GPRINT:hostIn:AVERAGE:"Average %lg%s\n" \
  DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out" \
  " \
  GPRINT:hostOut:MAX:"Max %lg%s" \
  GPRINT:hostOut:AVERAGE:"Average %lg%s\n" \
  DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total" \
  GPRINT:hostTotal:MAX:"Max %lg%s" \
  GPRINT:hostTotal:AVERAGE:"Average %lg%s\n" \
  COMMENT:"Actualizando...\r"

done

#####
## Instruccion para crear las graficas correspondientes a una semana ##
#####

GRAPHDIR="/var/www/monitor/public/images/weekly/" ;

for i in $HOST ; do
```

```

rrdtool graph $GRAPHDIR/$i.png --end now --start end-1w --width 600 --height 250 \
--title "HOST $i - GRAFICA SEMANAL" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...\\r"

```

done

```

#####
## Instruccion para crear las graficas correspondientes a un mes ##
#####

```

```

GRAPHDIR="/var/www/monitor/public/images/monthly/" ;

```

```

for i in $HOST ; do

```

```

rrdtool graph $GRAPHDIR/$i.png --end now --start end-1m --width 600 --height 250 \
--title "HOST $i - GRAFICA MENSUAL" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \

```



```

COMMENT:"Actualizando...\r"

done

#####
## Instruccion para crear las graficas correspondientes a un anio ##
#####

GRAPHDIR="/var/www/monitor/public/images/yearly/" ;

for i in $HOST ; do

rrdtool graph $GRAPHDIR/$i.png --end now --start end-1y --width 600 --height 250 \
--title "HOST $i - GRAFICA ANUAL" --vertical-label "Bits por segundo" --watermark ZETAGG \
--color CANVAS#000000 \
--color BACK#000000 \
--color FRAME#000000 \
--color FONT#FFFFFF \
--color ARROW#E70C0C \
--lower-limit=0 \
DEF:hostIn=$HOSTDIR/$i.rrd:In:AVERAGE LINE1:hostIn#0881C6:"Bytes In " \
GPRINT:hostIn:MAX:"Max %lg%s" \
GPRINT:hostIn:AVERAGE:"Average %lg%s\\n" \
DEF:hostOut=$HOSTDIR/$i.rrd:Out:AVERAGE LINE1:hostOut#B80D77:"Bytes Out " \
GPRINT:hostOut:MAX:"Max %lg%s" \
GPRINT:hostOut:AVERAGE:"Average %lg%s\\n" \
DEF:hostTotal=$HOSTDIR/$i.rrd:Total:AVERAGE LINE1:hostTotal#508B1D:"Bytes Total " \
GPRINT:hostTotal:MAX:"Max %lg%s" \
GPRINT:hostTotal:AVERAGE:"Average %lg%s\\n" \
COMMENT:"Actualizando...\r"

done

```

4.5 Creación de un cron para ejecutar tareas programadas

Como se mencionó anteriormente, para automatizar las tareas de nuestro sistema fue necesario utilizar un *cron*²⁷, el cual permite a los usuarios de Linux/Unix programar *scripts* o instrucciones para ejecutarse automáticamente a una hora o fecha específica.

²⁷ Un cron es un demonio o proceso del sistema que ejecuta tareas programadas.

Un cron se compone de dos elementos importantes (Figura 4.19):

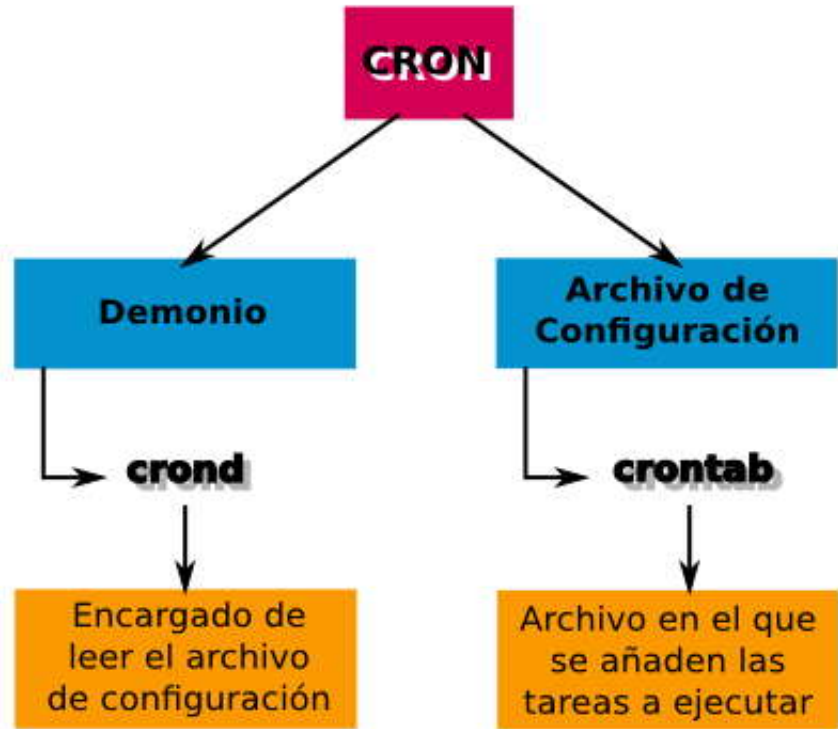


Figura 4.19 Elementos de un Cron en GNU/Linux

Un demonio llamado `crond`, el cual es un servicio que se ejecuta en background, es decir, que no requiere una interfaz directa con el usuario y por ser un demonio puede iniciarse en dos formas, al arrancar el sistema o cuando el usuario lo inicia.

Un archivo de configuración llamado `crontab`, en el cual se añaden las tareas que queremos programar.

Para poder editar el archivo de configuración ubicado en `/etc/crontab` es necesario ser `root` y ejecutar en una terminal el siguiente comando:

```
root@quarks:~# crontab -e
```

Con esta instrucción se abrirá un editor de texto con el archivo *crontab* vacío (Figura 4.20). Posteriormente el usuario define una línea por tarea/trabajo a ejecutar, el cual debe respetar el siguiente formato:



Figura 4.20 Formato principal de crontab

El archivo de texto inicial solo nos indica el orden en que debemos acomodar los datos para poder ser ejecutados.

El archivo *crontab* usado en QUARKS es el siguiente:

```
# m h dom mon dow  command

*/5 * * * * /bin/bash && /home/nglez/scripts-ipfm/7_red/rrd_create_7
*/5 * * * * /bin/bash && /home/nglez/scripts-ipfm/209_red/rrd_create_209

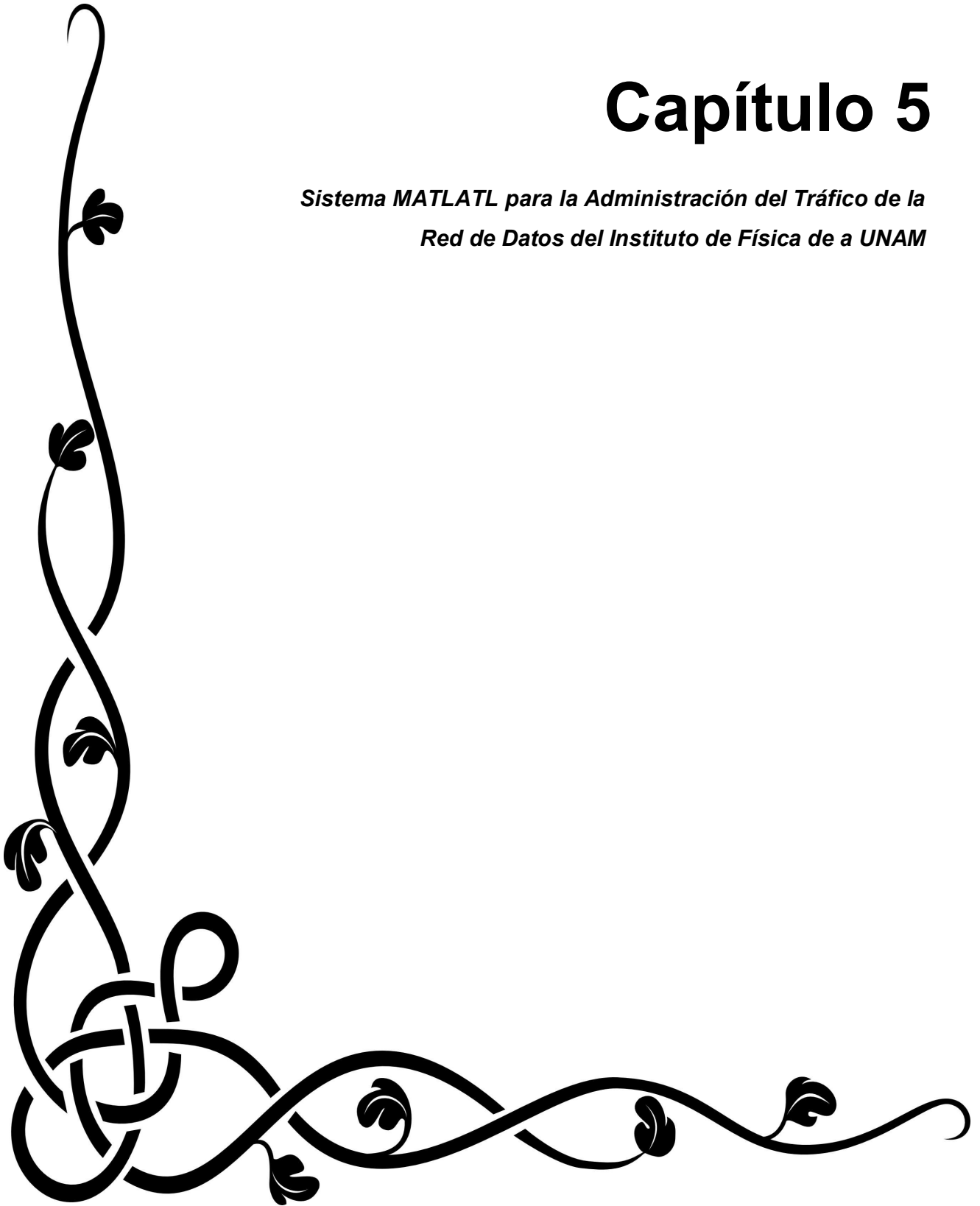
5 0 * * * /bin/bash && /home/nglez/scripts-ipfm/7_red/graficas_completas
5 0 * * * /bin/bash && /home/nglez/scripts-ipfm/209_red/graficas_completas

0 0 * * * /bin/bash && /home/nglez/scripts-ipfm/7_red/grafica1_x6horas
0 6 * * * /bin/bash && /home/nglez/scripts-ipfm/7_red/grafica2_x6horas
0 12 * * * /bin/bash && /home/nglez/scripts-ipfm/7_red/grafica3_x6horas
0 18 * * * /bin/bash && /home/nglez/scripts-ipfm/7_red/grafica4_x6horas

0 0 * * * /bin/bash && /home/nglez/scripts-ipfm/209_red/grafica1_x6horas
0 6 * * * /bin/bash && /home/nglez/scripts-ipfm/209_red/grafica2_x6horas
0 12 * * * /bin/bash && /home/nglez/scripts-ipfm/209_red/grafica3_x6horas
0 18 * * * /bin/bash && /home/nglez/scripts-ipfm/209_red/grafica4_x6horas
```

Capítulo 5

*Sistema MATLATL para la Administración del Tráfico de la
Red de Datos del Instituto de Física de a UNAM*



*“Hazlo todo tan simple como sea posible, pero no más simple.”
Albert Einstein*

Sistema MATLATL para la Administración del Tráfico de la Red de Datos del Instituto de Física de la UNAM

5.1 Selección de metodología para el desarrollo de software

Al comenzar a desarrollar un software es importante tener en mente la metodología que se utilizará, ya que nos indicará el camino a seguir para la correcta aplicación de los elementos involucrados en el proyecto.

Los procesos utilizados en el desarrollo de un software hacen referencia a los paradigmas en el desarrollo del software, los cuales en la actualidad se dividen en dos ramas, aquellos que utilizan metodologías tradicionales y los que utilizan metodologías livianas, pero ¿cuál es el método correcto a seguir y porqué?

Ambos métodos tienen sus ventajas y desventajas pero nuestra elección dependerá de las necesidades que tengamos.

5.1.1 Métodos orientados al plan.

Estos métodos también son conocidos como metodologías tradicionales, las cuales son desarrolladas basándose en la teoría de la Ingeniería de Software, lo que nos proporciona técnicas que permiten la mejora de procesos para crear un software de calidad. Su objetivo principal es tener un demo funcional y sin errores al final de cada iteración, así como una documentación comprehensiva y muy completa.

Utilizar dichas metodologías implica seguir una secuencia de pasos por cada iteración (unidad de tiempo) programada, tales como:

- Planificación
- Análisis de requerimientos
- Diseño
- Codificación
- Revisión o pruebas de unidad
- Documentación

5.1.2 Métodos ágiles

Las metodologías ágiles, o también llamadas metodologías livianas, son aquellas enfocadas en el *desarrollo de software iterativo e incremental* ²⁸ y se caracterizan por estar enfocados en la gente y en los resultados. Es decir que se realiza un trabajo de colaboración entre desarrollador-cliente, y se le da más importancia a la comunicación cara a cara que a la documentación. Su principal objetivo es la entrega de software funcional dentro de un periodo corto de tiempo antes de continuar con la siguiente etapa.

Las metodologías ágiles tienen como principios básicos promover los siguientes valores:

- Desarrollo
- Simplicidad
- Trabajo en equipo
- Colaboración
- Dar la bienvenida a los cambios.

²⁸ Entregas pequeñas de software en cortos periodos de tiempo.

A pesar de que las metodologías livianas se utilizaban desde principios de los 90's, el término "ágil" fue formalmente establecido en febrero del 2001, durante una reunión realizada en Utah, EEUU en la que participaron 17 expertos de la industria del software. En dicha reunión se creó La Alianza Ágil (The Agil Alliance) y tuvo como consecuencia el origen del *Manifiesto Ágil*²⁹, documento que resume el conjunto de ideologías de cada integrante.

Manifiesto para el desarrollo ágil de software:

Estamos descubriendo mejores formas de desarrollar software, haciéndolo y ayudando a otros a hacerlo. A través de este trabajo hemos aprendido a valorar:

- Al individuo y a las interacciones antes que a los procesos y herramientas.
- Al software funcional antes que a una buena documentación.
- A la colaboración con el cliente antes que a las negociaciones de contrato.
- Respuestas al cambio antes que seguir un plan.

5.1.3 Diferencias entre metodologías ágiles y tradicionales

Para facilitar el proceso de comparación se muestra a continuación la tabla donde se aprecian las diferencias existentes entre ambos métodos:

	Métodos ágiles	Métodos tradicionales
Enfoque	Adaptabilidad de los procesos	Planificación de los procesos
Tamaño del Proyecto	Pequeño	Grande
Administración	Descentralizada	Autócrata
Perspectivas al Cambio	Preparados para el cambio	Cierta resistencia la cambio
Cultura	Liderazgo-Colaboración Libertad para lograr objetivos	Comando-Control Roles y tareas definidas
Documentación	Poca	En grandes cantidades
Énfasis	Orientado a la gente	Orientado al proceso
Ciclos	Numerosos	Limitados
Dominios	Impredecible/Exploratorio	Predecibles
Planificación	Mínima	La evolución del proyecto está basada en le planificación
Recuperación de la Inversión	Temprano en el proyecto	Al finalizar el proyecto
Tamaño del equipo	Pequeño/Creativo	Grande

*Tabla 5.1 Diferencias entre metodologías ágiles y metodologías tradicionales*³⁰

²⁹ Fuente: Sitio web Agile Manifesto.

³⁰ Fuente: M. A. Awad "A comparision between Agil and Traditional Software Development Methodologies"

5.1.4 Razones para usar metodologías ágiles

Las metodologías tradicionales han estado presentes por mucho tiempo, y han demostrado ser necesarias en proyectos de gran tamaño, pero en la actualidad el desarrollo de software ha evolucionado considerablemente, y el entorno tan cambiante de los sistemas impide que un enfoque “tradicional” y estricto en sus procesos sea flexible en la implementación de software de alta calidad en periodos cortos de tiempo.

Las metodologías ágiles surgen como una solución a este problema, ofreciendo la posibilidad de incorporar nuevas herramientas y buenas prácticas para satisfacer al cliente, el cual es el objetivo de todo proyecto. Por tal motivo, y comparando las características de las metodologías antes mencionadas, se llegó a la conclusión de que los procesos ágiles se adecuaban a nuestras necesidades para desarrollar el Sistema Matlatl debido a las siguientes razones:

1. El grupo de trabajo era pequeño, estaba conformado por el programador y el administrador de red, por lo que era importante mantenerse en comunicación constante.
2. Se buscaba la solución para un problema en concreto, consultar y actualizar (vía web) la información de usuarios de la red de datos del Instituto de Física.
3. Se deseaba usar nueva tecnología que permitiera implementar aplicaciones enfocadas al manejo de información en tiempo real.
4. Las entregas de los avances debían ser constantes.
5. Debía existir la posibilidad de realizar cambios cuando fueran solicitados.

5.2 Ruby on Rails

Si bien es cierto que durante los últimos años Ruby on Rails ha adquirido cierta popularidad, la decisión de utilizar dicha plataforma se basó principalmente en su capacidad de desarrollar aplicaciones web de una forma más rápida, simple y flexible. Además de contar con librerías que proveen la independencia de la base de datos, permiten la transmisión de datos, operaciones básicas CRUD (create, read, update y delete), búsquedas avanzadas, relación

de modelos, etc. En pocas palabras, facilitan la conexión entre nuestra aplicación y la base de datos para el intercambio de información.

5.2.1 El lenguaje de programación Ruby

En 1995 el programador japonés Yukihiro “Matz” Matsumoto presentó públicamente la primera versión de Ruby, un lenguaje de programación dinámico y orientado a objetos, de código abierto y distribuido bajo la licencia GPL. Su sintaxis es una combinación de lenguajes como Perl, Smalltalk, Eiffel, Ada y Lisp, lo que facilita el uso de *programación funcional* ³¹ y *programación imperativa* ³².

Yukihiro ha comentado que “Ruby está diseñado para la productividad y diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario. El diseño de sistemas necesita enfatizar las necesidades humanas más que en las de las máquinas.”

Las ventajas más destacables de Ruby son:

- Todo es considerado un objeto.
- Su sintaxis es natural, clara y sencilla.
- El código es más sencillo de entender por lo que se reduce la probabilidad de errores.
- Es portable ya que puede correr en la mayoría de plataformas tales como GNU/Linux, Unix, Mac OS X, Windows 95/98/Me/2000/XP, etc.
- No es necesario declarar las variables y éstas siempre hacen referencias a objetos, no son los objetos mismos.
- Soporta herencia simple para extender su funcionalidad y evitar duplicidad de código.
- Buena funcionalidad al manejar expresiones regulares.

Desventajas:

- El tiempo de ejecución es más lento ya que cada instrucción debe ser interpretada al momento de ejecutarse.

³¹ Programación basada en el uso de funciones aritméticas (Haskell, Miranda, Lisp, Scheme, ML, Ocaml, SAP).

³² Programación basada en un conjunto de instrucciones que indican al equipo los pasos a realizar para conseguir un objetivo (ASP, Basic, C, Fortran, Pascal, Perl, PHP, Java).

5.2.2 El framework Rails

Entorno de programación (framework) diseñado para facilitar el desarrollo de aplicaciones web usando Ruby. Creado por el programador danés David Heinemeier Hansson y liberado en 2004 como software de código libre.

Las aplicaciones de Rails están basadas en la arquitectura Modelo-Vista-Controlador (MVC), ver Figura 5.1, la cual es considerada como la columna central de todo proyecto y que separa los datos, la interfaz de usuario y la lógica de control en tres componentes:

Modelo: Responsable del *comportamiento*³³ y el uso de los *datos*³⁴ en la aplicación.

Vista: Responsable de generar la interfaz de usuario con los datos de la aplicación.

Controlador: Responde a las acciones del usuario e indica qué se debe hacer.

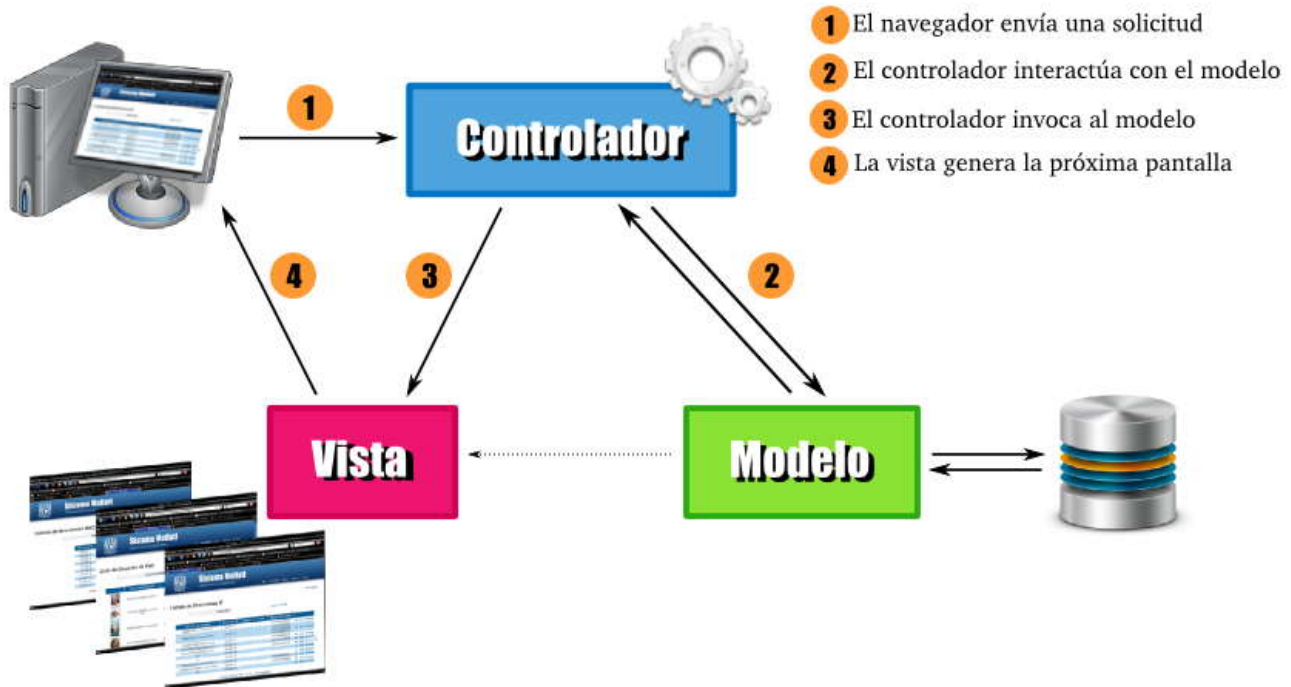


Figura 5.1 Arquitectura Modelo-Vista-Controlador³⁵

³³ Responde a instrucciones del controlador.

³⁴ Responde a peticiones de la vista.

³⁵ Fuente: Sam Ruby, Dave Thomas "Agile Web Development with Rails"

El flujo de control del MVC que se muestra en la figura 5.1 es el siguiente: Se muestra una aplicación en el navegador y el usuario decide hacer una solicitud, por ejemplo dando click a un botón, el controlador interactúa con el modelo para notificar los datos solicitados por la acción del usuario y le encarga a la vista la tarea de mostrar los datos, la vista a su vez obtiene los datos del modelo y así puede generar la próxima pantalla.

Algo que caracteriza a Rails de otros entornos de trabajo son los principios de desarrollo en los que está basado:

DRY - Don't Repeat Yourself (No te repitas).

Cada definición debe realizarse una sola vez, es decir, especificas lo que necesitas en un solo lugar y sigues con el trabajo. Ruby y la arquitectura MVC serán los encargados de interpretar estas definiciones.

Convención sobre configuración.

Al diseñar una aplicación con Rails se escribirá menos código si se siguen las convenciones, pero existe la posibilidad de configurar un nuevo comportamiento sin afectar el desempeño del sistema.

Las ventajas de Rails son:

- Es ágil, por lo que facilita la interacción con los desarrolladores.
- Si se utilizan las convenciones del framework, podemos ahorrarnos mucho trabajo.
- Se pueden desarrollar aplicaciones muy completas usando poco código.
- Utiliza un sistema de plugins para permitir modificaciones y ampliar los proyectos conforme lo vayan requiriendo.
- Utiliza la técnica Mapeo Objeto-Relacional (ORM - Object-Relational Mapping) lo que nos permite expresar fácilmente cada una de las relaciones ya que se trata a cada uno de los registros como objetos.

Desventajas:

- En México es difícil encontrar un *hosting* capaz de soportar aplicaciones de RoR.
- Rails consume mucha memoria.
- Poner en producción una aplicación es muy complicado.

Es importante mencionar que el presente trabajo no se ve directamente afectado por dichas desventajas ya que:

- El instituto cuenta con servidor propio, por lo que no existe la necesidad de contratar un *hosting* y tenemos la ventaja de poder instalar el software que se necesite.
- A pesar de consumir muchos recursos de memoria, el sistema difícilmente será sobresaturado con consultas ya que es para uso interno de la Secretaría Técnica de Cómputo, Telecomunicaciones y Fotografía del Instituto de Física de la UNAM, por lo que solo los encargados de las áreas de Red, Seguridad, Informática, Supercómputo y Desarrollo de Aplicaciones tendrán acceso a él (aproximadamente 10 usuarios), ningún estudiante o investigador del instituto tendrá permisos para ingresar en el sistema. Además el servidor cuenta con 2 GB en RAM.
- Hay programadores que consideran difícil poner en producción aplicaciones de RoR porque no están acostumbrados a trabajar con líneas de comandos y a modificar los archivos de configuración de las aplicaciones, en nuestro caso es diferente ya que se tiene experiencia trabajando en sistemas operativos que utilizan dichos recursos.

5.3 Instalación de Ruby on Rails (RoR)

Antes de iniciar cualquier instalación es recomendable actualizar el sistema operativo, teniendo los permisos de *root* ejecutamos en una terminal las siguientes instrucciones:

```
# apt-get update  
# apt-get dist-upgrade
```

1. Instalamos la librería que nos permite compilar paquetes.

```
# apt-get install build-essential
```

2. Instalamos los paquetes necesarios para comenzar a usar RoR.

```
# apt-get install ruby ri rdoc ruby1.8-dev irb1.8 libdbi-perl libnet-daemon-perl libplrpc-perl libreadline-  
ruby1.8 libruby1.8 rdoc1.8 ri1.8 ruby1.8 irb libopenssl-ruby libopenssl-ruby1.8 libhtml-template-perl  
libreadline5 psmisc
```

3. RubyGems es un gestor de paquetes utilizado por Ruby para la instalación de gemas (gems), este sistema favorece la distribución de aplicaciones o librerías en un formato estándar, se utiliza un servidor para almacenar las gemas y poder descargarlas desde cualquier máquina que tenga instalado RubyGems. Se descargó la última versión y procedimos con su instalación.

```
# wget http://rubyforge.org/frs/download.php/69365/rubygems-1.3.6.tgz  
# tar xzvf rubygems-1.3.6.tgz  
# cd rubygems-1.3.6  
# ruby setup.rb
```

4. Para evitar el tener que escribir la ubicación completa de los programas durante el desarrollo de un proyecto creamos enlaces simbólicos que apunten a éstos.

```
# ln -s /usr/bin/gem1.8 /usr/local/bin/gem  
# ln -s /usr/bin/ruby1.8 /usr/local/bin/ruby  
# ln -s /usr/bin/rdoc1.8 /usr/local/bin/rdoc  
# ln -s /usr/bin/ri1.8 /usr/local/bin/ri  
# ln -s /usr/bin/irb1.8 /usr/local/bin/irb
```

5. Apoyándonos en el gestor rubygems instalamos Rails.

```
# gem install rails
```

6. Se instaló SQLite, como herramienta de base de datos, la cual es una librería que nos ofrece las características necesarias para trabajar con una base de datos relacional y permitir consultas SQL. Creada por D. Richard Hipp y de dominio público. Es una herramienta ágil y robusta, ya que la definición de las tablas, los índices y los propios datos son guardados en un archivo local al que se accede mediante llamadas a subrutinas o funciones de la propia librería.

```
# apt-get install sqlite3 swig libsqlite3-ruby libsqlite3-ruby1.8 libsqlite3-dev  
# gem install sqlite3-ruby
```

5.4 Implementación del Sistema Matlatl



Figura 5.2 Nombre del Sistema

El nombre del sistema (Figura 5.2) tiene su origen en la palabra náhuatl MATLATL, la cuál significa RED, y a pesar de que los antiguos aztecas designaban esta palabra para redes de pesca o caza, se usó este nombre no solo por el hecho de hacer una alusión a la conexión entre puntos, si no porque se realizaría una mezcla de nuestro pasado con el presente, nuestras raíces y cultura con la tecnología actual.

Técnicamente hablando, Matlatl es un sistema web creado para administrar la información de los usuarios de la red de datos del IFUNAM, así como gestionar la asignación de direcciones IP, y por seguridad, llevar un registro de las direcciones MAC asociadas a cada IP. Permite la creación, actualización y eliminación de datos en tiempo real y ofrece el apoyo visual (gráficas) para observar el tráfico de red por usuario. Desarrollado en su totalidad con software de distribución libre para uso exclusivo de la Secretaría Técnica de Cómputo, Telecomunicaciones y Fotografía del Instituto de Física.

Para comprender que es lo que se necesita en un sistema, se recomienda identificar los casos de uso, los cuales son una declaración sencilla de cómo un usuario utiliza el sistema.

5.4.1 Casos de uso de Matlatl

Se identifican 2 tipos de usuarios: el administrador y el usuario.

1. El Administrador utiliza Matlatl para consultar una lista de los usuarios de red, una lista de las direcciones ips asignadas y una lista de las direcciones macs registradas, además puede actualizar los datos o ingresar nuevos. También tiene la opción de registrar nuevos usuarios que puedan acceder al sistema.

2. El Usuario utiliza Matlatl para consultar y buscar las ips o macs asignadas a los usuarios de red.

Para ver el diagrama de flujo de páginas del sistema Matlatl observe la Figura 5.3

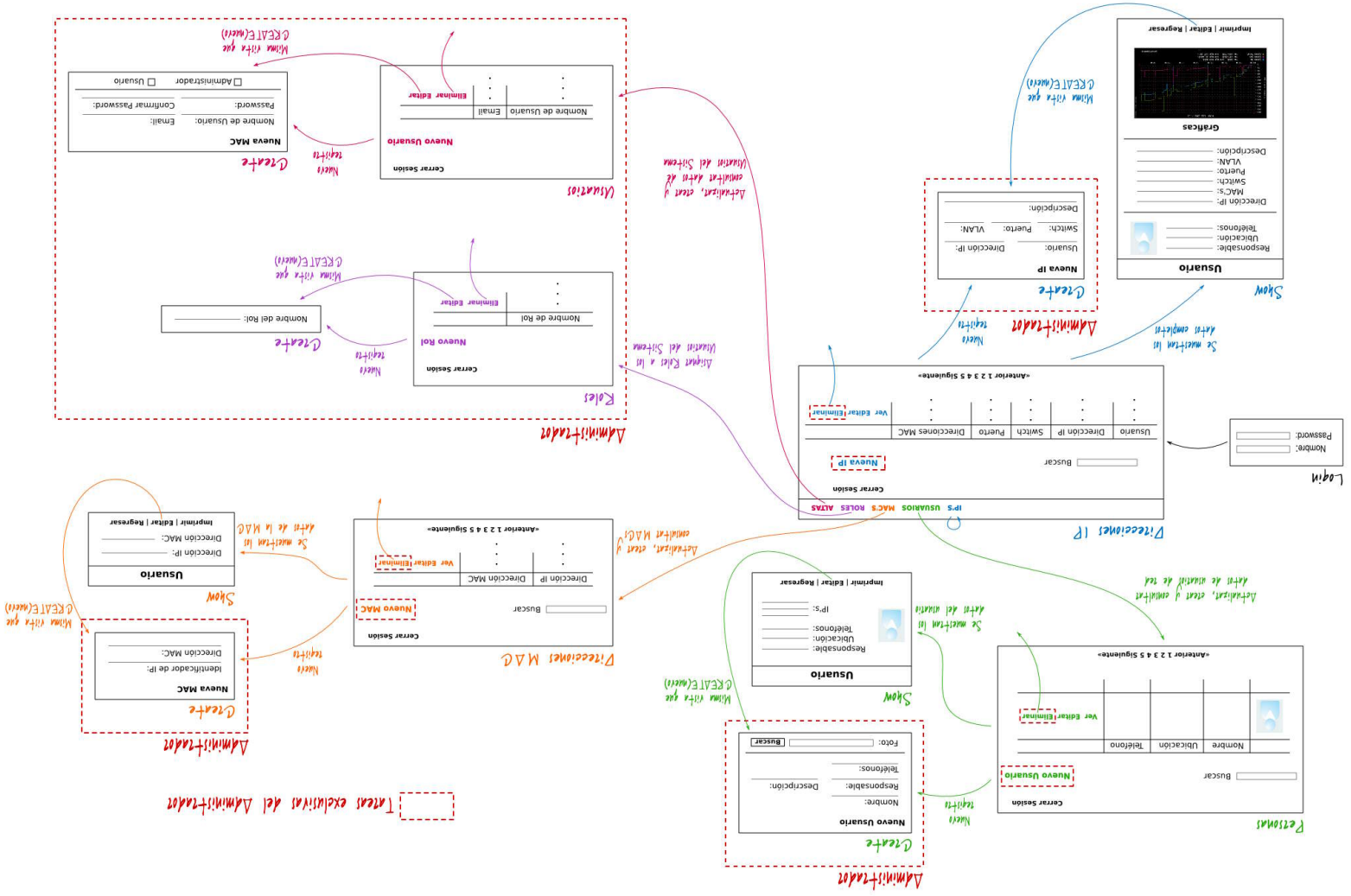


Figura 5.3 Diagrama de flujo de páginas del sistema Matlatl

5.4.2 Creación de la Base de Datos.

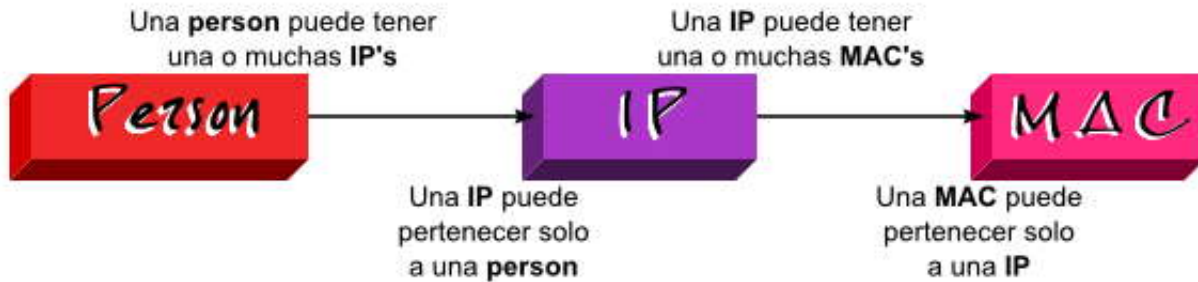


Figura 5.4 Las 3 tablas principales de la Base de Datos

1. Se creó un nuevo proyecto:

```
$ rails monitor_redes
```

2. Para generar la estructura principal del proyecto creamos los controladores, los modelos y las vistas utilizando la instrucción scaffold:

- Para *People*

```
$ ruby script/generate scaffold person person_name:string responsible:string location:text tel_person:string
```

- Para *IP's*

```
$ ruby script/generate scaffold ip person_id:integer ip_address:string port:integer switch:string vlan:string description:text
```

- Para *MAC's*

```
$ ruby script/generate scaffold mac ip_id:integer mac_address:string
```

3. Ejecutamos el comando de migración para crear las tablas:

```
$ rake db:migrate
```

4. Para indicar las relaciones existentes entre las clases, indicamos las asociaciones en cada uno de los modelos:

- En el modelo person.rb agregamos la siguiente línea:

```
has_many :ips # una persona puede tener muchas ip's
```

- En el modelo ip.rb agregamos las siguientes líneas

```
belongs_to :person # una ip pertenece a una sola persona
has_many :macs    # una ip puede tener muchas mac's
```

- En el modelo mac.rb agregamos la siguiente línea:

```
belongs_to :ip # una mac pertenece a una sola ip
```

5.4.3 Presentación de los Datos.

Para presentar los datos con un formato estándar se realizaron los siguientes pasos:

1. Se creó una hoja de estilos y se ubicó en:

```
/public/stylesheets/estilo_redes.css
```

2. Se indicó en el *tag head* de cada layout la nueva hoja de estilos a utilizar, ya que RoR genera una predeterminada.

```
<%= stylesheet_link_tag ('estilo_redes', :media => :screen) %>
```

3. Para centrar la plantilla que se mostrará en el navegador web se usó el método de los *div's*, los cuales son declarados en la hoja de estilos, y se agregó a cada layout dentro del *tag body* las siguientes sentencias:

```
<div class="contenedor">
  <div class="contenido">

    Contenido de la página

  </div>
</div>
```

4. Para imprimir los datos de los usuarios usamos *javascript* con hojas de estilo, agregamos en el *tag head* de cada layout donde usemos esta función la siguiente instrucción:

```
<%= stylesheet_link_tag ('estilo_redes', :media => :print) %>
```

- Se agregó un link a la función "Imprimir" con la siguiente instrucción:

```
<%= link_to_function("Imprimir", "javascript:print()") %>
```

5.4.4 Instalación e implementación de Auto_Complete

Plugin oficial de Rails que sirve para desplegar información en un campo de texto, utiliza librerías de JavaScript para proporcionar efectos visuales en interfaces de usuario. Su principal uso es para autocompletar datos en un formulario.

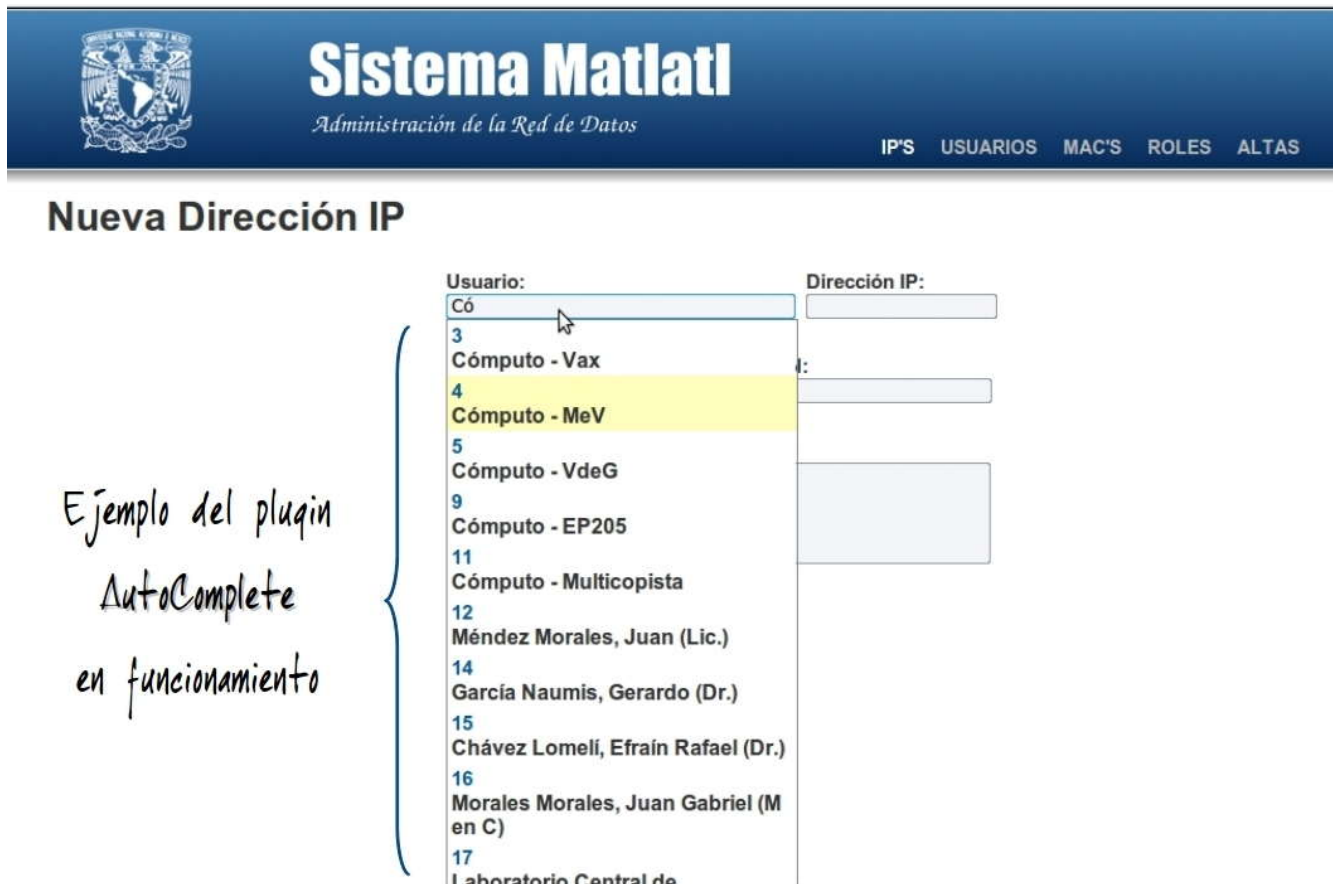


Figura 5.5 Ejemplo del plugin AutoComplete usado en el sistema

La decisión de usar dicha extensión en nuestro sistema se debió a la dificultad que presentaba el tener que consultar los identificadores para poder relacionar datos en tiempo real, por ejemplo, para asociar una ip con un usuario era necesario verificar:

1. Si el usuario ya existía.
2. Si existía, era necesario consultar su id.
3. Si el usuario no existía debía crearse para poder tener un id.

4. Por último se relaciona el id del usuario con la ip.

Usando *auto_complete* en lugar de indicar el id nosotros comenzamos a escribir en el campo de texto, el *plugin* ejecuta una instrucción de consulta a la base de datos para presentarnos una lista de los nombres que coincidan con la secuencia de letras que vamos ingresando, al realizar nuestra selección guarda el id correspondiente al usuario elegido. De esta forma evitamos tener que navegar entre diferentes pantallas consultando datos.

Al configurarlo, el desarrollador debe indicar los datos que se deben mostrar y los datos que serán almacenados.

- Descargar el *plugin* del repositorio github.

```
$ wget http://github.com/rails/auto_complete/tree/master/rails-auto_complete.tar.gz
```

- Se descomprimió el paquete y se renombró la carpeta que se generó.

```
$ tar -xvf rails-auto_complete.tar.gz
$ mv -rf rails-auto_complete auto_complete
```

- Copiar la carpeta dentro de la ubicación /vendor/plugins/

```
$ cp -rf auto_complete/ monitor_redes/vendor/plugins/
```

Procedimiento para configurar auto_complete entre ip-person

- Se agregó en el archivo /app/views/layouts/ips.html.erb dentro de la etiqueta *head* la siguiente instrucción:

```
<%= javascript_include_tag :defaults %>
```

- Se buscó en el archivo /app/views/ips/new.html.erb la línea correspondiente a la instrucción del campo de texto para *person_id*

```
<%= f.text_field :person_id %>
```

- Dicha instrucción se modificó para tener un campo de autocompletado de datos, además se indicó el parámetro que debe almacenar (*person_id*):

```
<%= text_field_with_auto_complete 'ip', 'person_id', {}, :skip_style => false %>
```

- Dentro del controlador /app/controller/ips_controller.rb se agregó la siguiente instrucción:

```
skip_before_filter :verify_authenticity_token, :only => [:auto_complete_for_ip_person_id]
```

- En el mismo controlador se definió la función que permite realizar el autocompletado:

```
def auto_complete_for_ip_person_id()
  personname = params[:ip][:person_id]
  @people = Person.find(:all , :conditions=>"person_name like '%" + personname.downcase + "%")
  render :partial => 'personname'
end
```

- Ubicamos en /app/views/ips/ el *partial*³⁶_personname.html.erb:

```
<ul class="allusers">
  <% for person in @people do %>
    <li class="thisuser">
      <div class="useremail">
        <%=h person.id %>
      </div>

      <div class="username">
        <span class="informal">
          <%=h person.person_name %>
        </span>
      </div>
    </li>
  <% end %>
</ul>
```

Procedimiento para configurar auto_complete entre mac-ip

El procedimiento es similar al descrito con anterioridad, solo se cambiaron algunos parámetros.

- Se agregó en el archivo /app/views/layouts/mac.html.erb dentro de la etiqueta *head* la siguiente instrucción:

```
<%= javascript_include_tag :defaults %>
```

- Se modificó la instrucción del campo de texto en el archivo /app/views/mac/new.html.erb para tener un campo de autocompletado, se indicó el parámetro que debe almacenar (*ip_id*):

```
<%= text_field_with_auto_complete 'mac', 'ip_id', {}, :skip_style => false %>
```

³⁶ Método utilizado para crear sub-vistas que requieran diferentes parámetros a los establecidos dentro de un controlador. También se usan cuando un mismo código será utilizado por diferentes vistas.

- Se agregó en el controlador `/app/controller/mac_controller.rb` la siguiente instrucción:

```
skip_before_filter :verify_authenticity_token, :only => [:auto_complete_for_mac_ip_id]
```

- En el mismo controlador se definió la función de autocompletado.

```
def auto_complete_for_mac_ip_id()
  ipaddress = params[:mac][:ip_id]
  @ips = Ip.find(:all , :conditions=>"ip_address like '%" + ipaddress.downcase + "%'")
  render :partial => 'ipaddress'
end
```

- Se creó el *partial* `ipaddress.html.erb` ubicado en `/app/views/macs/`

```
<ul class="allusers">
  <% for ip in @ips do %>
    <li class="thisuser">
      <div class="useremail">
        <%=h ip.id %>
      </div>
      <div class="username">
        <span class="informal">
          <%=h ip.ip_address %>
        </span>
      </div>
    </li>
  <% end %>
</ul>
```

5.4.5 Instalación e implementación de Will_Paginate

Plugin para Rails que tiene como función paginar los resultados en la interfaz de usuario, es decir, que limita a un número preestablecido la cantidad de datos que se mostrarán por página (Figura 5.6).

Hasta este momento en el Instituto se cuentan con 510 direcciones IP, 207 usuarios de red y 410 direcciones MAC's registrados. Presentar dichos datos en una sola pantalla sería impráctico y poco presentable, por lo que se decidió usar un método de paginación para reducir la cantidad de datos que se presentaban durante una consulta.

The screenshot shows the 'Sistema Matlatl' web interface. At the top, there is a header with the system name and navigation links for 'IP'S', 'USUARIOS', 'MAC'S', 'ROLES', and 'ALTAS'. Below the header, there is a search bar labeled 'Buscar por MAC' and a link for 'Ingresar nueva MAC'. The main content area displays a table with columns for 'Dirección IP', 'Dirección MAC', and three action buttons: 'Ver', 'Editar', and 'Eliminar'. A red box highlights a portion of the table with the text 'Muestra del resultado obtenido al usar el plugin WillPaginate'. Below the table, a pagination control is shown, which is circled in red and contains the text '« Anterior 1 2 3 4 5 6 7 8 9 ... 42 43 Siguiente »'.

Figura 5.6 Ejemplo del plugin WillPaginate usado en el sistema

- Instalamos el código fuente de *GemCutter* para proporcionar nuevas fuentes de gemas:

```
$ sudo gem install gemcutter
```

- Las extensiones en RoR pueden ser instaladas como *plugin* o como gema, en este caso se instaló la gema *will_paginate* con la siguiente instrucción:

```
$ sudo gem install will_paginate
```

- Para habilitar la librería en el proyecto se agregó en el archivo `/config/environment.rb` el método de configuración de gemas:

```
require "will_paginate"
```

- Se agregó en cada controlador, dentro de la función `index`, la instrucción para indicar que se usará el método de paginación, además de establecer la cantidad de datos que se presentarán por página:

1. Para /app/controllers/people_controller.rb

```
@people = Person.paginate(:page => params[:page], :per_page => 10)
```

2. Para /app/controllers/ips_controller.rb

```
@ips = Ip.paginate(:page => params[:page], :per_page => 10)
```

3. Para /app/controllers/mac_controller.rb

```
@macs = Mac.paginate(:page => params[:page], :per_page => 10)
```

- Por último se agregó a la interfaz de usuario los enlaces a las páginas para navegar entre ellas:

1. Para /app/views/people/index.html.erb

```
<%= will_paginate @people, :prev_label => "« Anterior", :next_label => "Siguiente »" %>
```

2. Para /app/views/ips/index.html.erb

```
<%= will_paginate @ips, :prev_label => "« Anterior", :next_label => "Siguiente »" %>
```

3. Para /app/views/mac/index.html.erb

```
<%= will_paginate @macs, :prev_label => "« Anterior", :next_label => "Siguiente »" %>
```

5.4.6 Instalación e implementación de PaperClip

Plugin para Rails cuya principal función es adjuntar o almacenar archivos (Figura 5.7), tiene la capacidad de gestionar la presencia y el tamaño de ellos así como validar el tipo de archivo que se está cargando. No hay necesidad de crear nuevas tablas en la base de datos ya que se anexa como un campo en una ya existente.



Figura 5.7 Ejemplo del plugin PaperClip usado en el sistema

Para el procesamiento de imágenes se requiere la instalación de la librería *ImageMagick*³⁷ lo que nos facilita la creación de miniaturas.

En el sistema Matlatl se sube la imagen de los usuario, y de cada original se generan dos tamaños, una miniatura para presentar en la lista general y una mediana para la presentación de los datos generales.

- Instalamos la gema de Paperclip

```
$ sudo gem install paperclip
```

- Usando el método de configuración de gemas se agregó el archivo /config/environment.rb la siguiente línea:

³⁷ Software para editar, crear y componer imágenes.

```
config.gem 'paperclip', :source => 'http://gemcutter.org'
```

- Después se ejecutaron en la terminal la siguiente instrucción:

```
$ rake gem:install
```

- Al terminar la instalación de la gema se procedió a ejecutar su generador para agregar un campo de adjuntar documentos, en nuestro caso una imagen, al modelo Person:

```
$ ruby script/generate paperclip person photo
```

- Se creó la migración `20100205012445_add_attachments_photo_to_person.rb`, la cual generó automáticamente cuatro nuevos campos en el modelo, un campo para el nombre, un campo para el tipo de documento, un campo para el tamaño y un campo para la fecha de actualización:

```
add_column :people, :photo_file_name, :string
add_column :people, :photo_content_type, :string
add_column :people, :photo_file_size, :integer
add_column :people, :photo_updated_at, :datetime
```

- Para actualizar los cambios en la tabla people de la base de datos se ejecutó la siguiente instrucción:

```
$ rake db:migrate
```

- Se realizaron modificaciones al código del modelo para indicar que se adjuntaría un documento. Se agregó a `/app/models/person.rb` la siguiente línea:

```
has_attached_file :photo
```

- El código de las vistas también se vio modificado, por lo que en los archivos `/app/views/people/new.html.erb` y `/app/views/people/edit.html.erb` dentro de la sentencia `form_for` se agregaron los atributos que permiten la opción de adjuntar archivos:

```
<% form_for(@person, :html => { :multipart => true }) do |f| %>
```

- Se agregó la instrucción para el campo de búsqueda del archivo:

```
<%= f.file_field :photo %>
```

El método `has_attached_file` que se agregó en el modelo `person.rb` tiene diferentes opciones, una de ellas se llama `style`, lo que nos permite redimensionar el tamaño de las imágenes, en nuestro caso nos interesaba tener una vista en miniatura para presentar en el índice de

usuarios y otra mediana para presentar en los datos generales de cada usuario, por lo que se realizaron algunos cambios:

- Antes de continuar fue necesario instalar en el servidor la librería ImageMagic:

```
$ sudo apt-get install imagemagick
```

- En el archivo `/app/models/person.rb` se modificó la instrucción agregada con anterioridad:

```
has_attached_file :photo
```

- Se especificaron los atributos para redimensionar las imágenes:

```
has_attached_file :photo, :styles => { :medium => "160x160>", :small => "80x80>" }
```

- En la vista `/app/views/people/index.html.erb` usamos el estilo `:small` para el índice de usuarios:

```
<%= image_tag person.photo.url(:small) %>
```

- En la vista `/app/views/ips/show.html.erb` usamos el estilo `:medium` para mostrar los datos generales del usuario:

```
<%= image_tag @person.photo.url(:medium) %>
```

- Una opción más que nos ofrece *PaperClip* es la posibilidad de validar los archivos adjuntos, era importante especificar el tamaño límite y el tipo de archivo a usar, por lo que se agregó al modelo `/app/models/person.rb` las instrucciones que solo permiten archivos con extensiones `.jpg` o `.png` y que no rebasen un tamaño mayor a 1 MB:

```
validates_attachment_size :photo, :less_than => 1.megabytes  
validates_attachment_content_type :photo, :content_type => ['image/jpeg', 'image/png']
```

5.4.7 Creación de una caja de búsqueda.

La secretaría técnica de cómputo del IFUNAM quería tener la posibilidad de realizar búsquedas dentro del sistema, para localizar direcciones MAC, direcciones IP y usuarios.

Para dicho propósito se generaron cajas de búsqueda en los tres modelos principales: *Person*, *Ip* y *Mac* (Figura 5.8). A continuación se explican los pasos que se siguieron para crearlas.

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

Cerrar sesión

Lista de Usuarios de Red

Buscar por nombre Ingresar nuevo usuario

	Nombre de Usuario	Ubicación	Teléfono	
	Ortiz Salazar, Ma. Esther (Dra.)	Acel 5.5, 1° p. C-3 Jefe Fis. Exp.	562-25057	Ver Editar Eliminar
	García Naumis, Gerardo (Dr.)	EP, PB, C-3 Jefe Fis. Exp.	562-25174	Ver Editar Eliminar
	Chávez Lomelí, Efraín Rafael (Dr.)	Acel 5.5, 1° p. C-3 Jefe Fis. Exp.	562-25070	Ver Editar Eliminar
	Morales Morales, Juan Gabriel (M en C)	Colisur, 1° p. C-FG-33 Fis. Exp.	562-25003; 562-25013; 562-25054	Ver Editar Eliminar

Campo de Búsqueda por dirección MAC

Figura 5.8 Ejemplo de una caja de búsqueda usada en el sistema

Forma de búsqueda por persona.

- Primero se modificó la instrucción de paginación ya que dicha sentencia cambia al usar formas de búsqueda, por lo que en el controlador `/app/controllers/people_controller.rb` se buscó la siguiente instrucción:

```
@people = Person.paginate(:page => params[:page], :per_page => 5)
```

- Y se realizó el cambio para juntar las funciones de búsqueda con paginación:

```
@people = Person.search(params[:search]).paginate :per_page => 5, :page => params[:page]
```

- En el modelo `/app/models/person.rb` se definieron los patrones de búsqueda agregando la siguiente función:

```
def self.search(search)
  if search
    find(:all, :conditions => ['person_name LIKE ?', "%#{search}%"])
  else validates_presence_of :ip_id, :mac_address
    find(:all)
  end
end
```

Dicha función busca el nombre de la persona(s) que coincidan con los caracteres tecleados en el campo de texto, si no se ingresa ningún texto muestra los datos de todos los usuarios.

- Por último se agregó a la vista /app/views/people/index.html.erb la instrucción para mostrar el campo de texto y el botón de búsqueda.

```
<% form_tag people_path, :method => 'get' do %>
  <%= text_field_tag :search, params[:search] %>
  <%= submit_tag "Buscar por nombre", :person_name => nil %>
<% end %>
```

Forma de búsqueda por dirección IP.

- Se indicó en el controlador /app/controllers/ips_controller.rb la instrucción que permite juntar las funciones de paginación y búsqueda:

```
@ips = Ip.search(params[:search]).paginate :per_page => 5, :page => params[:page]
```

- Se definieron los parámetros de búsqueda en el modelo /app/models/ip.rb

```
def self.search(search)
  if search
    find(:all, :conditions => ['ip_address LIKE ?', "%#{search}%"])
  else
    find(:all)
  end
end
```

- Se agregó a la vista /app/views/ips/index.html.erb la instrucción para mostrar el campo de texto y el botón de búsqueda.

```
<% form_tag ips_path, :method => 'get' do %>
  <%= text_field_tag :search, params[:search] %>
  <%= submit_tag "Buscar por IP", :ip_address => nil %>
<% end %>
```

Forma de búsqueda por dirección MAC.

- Se indicó en el controlador /app/controllers/mac_controller.rb la instrucción que une las funciones de paginación y búsqueda.

```
@macs = Mac.search(params[:search]).paginate :per_page => 5, :page => params[:page]]
```

- Se definieron los parámetros de búsqueda en el modelo `/app/models/mac.rb`

```
def self.search(search)
  if search
    find(:all, :conditions => ["mac_address LIKE ?", "%#{search}%"])
  else
    find(:all)
  end
end
```

- Se agregó a la vista `/app/views/mac/index.html.erb` la instrucción para mostrar el campo de texto y el botón de búsqueda.

```
<% form_tag ips_path, :method => 'get' do %>
  <%= text_field_tag :search, params[:search] %>
  <%= submit_tag "Buscar por IP", :ip_address => nil %>
<% end %>
```

Como se puede apreciar, el procedimiento para agregar la caja de búsqueda es el mismo en los tres modelos, la diferencia entre ellos radica en las variables y la definición de los parámetros.

5.4.8 Validaciones

Una de las grandes ventajas al desarrollar sistemas en RoR es que podemos implementar validaciones de una forma rápida y con poco código.

*“La capa del modelo es el guardián entre el mundo del código y la base de datos”*³⁸ por lo que es recomendable colocar las validaciones en el modelo, ya que nada que tenga que ver con nuestro sistema sale de la base de datos o se almacena en ella sin antes ser verificado por el modelo. Este método como medida de seguridad es perfecto para mantener la base de datos libre de datos basura además de prevenir al usuario de guardar datos corruptos.

Algunas de las validaciones más utilizadas son:

1. Validaciones de campos vacíos: `validate_presence_of`

³⁸ Fuente: Sam Ruby, Dave Thomas “Agile Web Development with Rails”

2. Validaciones de singularidad: `validate_uniqueness_of`
3. Validaciones de campos numéricos: `validate_numericality_of`
4. Validaciones de formatos para email o nombres: `validate_format_of`

En el presente trabajo se realizaron las validaciones para evitar que ciertos campos de texto estuvieran vacíos, ya que al guardar datos nulos se generaban errores que impedían acceder a la información. En la Figura 5.9 se muestra un ejemplo de las validaciones de datos ya que nos indica cuales son los datos que no pueden estar en blanco.



Figura 5.9 Ejemplo de validación de datos usado en el sistema

- Para comprobar la presencia del nombre de usuario y el responsable en el modelo `/app/models/person.rb` se agregó la siguiente validación:

```
validates_presence_of :person_name, :responsible
```

- Para comprobar la presencia del id del usuario y la dirección ip en modelo `/app/models/ip.rb` se agregó la siguiente validación:

```
validates_presence_of :person_id, :ip_address
```

- Para comprobar la presencia del id de la dirección ip y la dirección mac en el modelo /app/models/mac.rb se agregó la siguiente validación:

```
validates_presence_of :ip_id, :mac_address
```

5.4.9 Logeo y autenticación de usuarios.

Con el motivo de mantener un control en la edición y consulta de datos, se solicitó realizar una verificación de los usuarios que ingresen al sistema.

Instalación e implementación de Authlogic.

Plugin para Rails que ofrece una solución para autenticación de usuarios en un sistema web, de código simple, limpio y discreto. Se caracteriza por que es fácil de extender, ya que hay otros plugins que pueden agregarle funcionalidades.

Authlogic no genera vistas o controladores, se trabaja directamente con el código por lo que nos ofrece mayor flexibilidad y control en la función encargada de administrar la autenticación de usuarios.

A continuación se describe el procedimiento que se siguió para implementar dicho plugin en el sistema Matlatl.

- Se instaló el plugin como gema, por lo que indicamos en /config/environment.rb el método de configuración:

```
config.gem "authlogic"
```

- Ejecutamos en la terminal la instrucción:

```
rake gems:install
```

- Al terminar la instalación procedimos a crear el modelo y el controlador para administrar a los usuarios. Ubicándonos en la carpeta donde se almacena el proyecto se generó el modelo llamado User:


```
$ ruby script/generate model user
```

- Para configurar las propiedades del modelo editamos la migración generada, es importante mencionar que *Authlogic* trabaja buscando columnas con un determinado nombre dentro de la base de datos, por lo que éstos no son arbitrarios, son específicos para ser usados por el plugin y pueden consultarse en su documentación. A continuación se muestra el archivo de migración después de agregar los parámetros requeridos:

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table :users do |t|
      t.string :username
      t.string :email
      t.string :crypted_password
      t.string :password_salt
      t.string :persistence_token
      t.timestamps
    end
  end
  def self.down
    drop_table :users
  end
end
```

- Para crear la nueva tabla en la base de datos usamos la siguiente instrucción:

```
$ rake db:migrate
```

- Para que el modelo User trabaje con las funcionalidades de *Authlogic* es importante añadir en `/app/models/user.rb` la siguiente instrucción:

```
acts_as_authentic
```

- Se generaron el controlador users y las vistas index, new y edit:

```
$ ruby script/generate controller users new edit
```

- Dentro del controlador `/app/controllers/users_controller.rb` se definió la función que permite crear nuevos usuarios:

```
def create
  @user = User.new(params[:user])
  if @user.save
    flash[:notice] = "Registro realizado"
    redirect_to :controller => "users", :action => "index"
  end
end
```

```
else
  render :action => 'new'
end
end
```

Si al ingresar los datos hay un error de validación (realizadas por *authlogic*), se mostrará nuevamente el formulario para nuevos usuarios, en caso contrario se creará el usuario y se almacenará en la base de datos.

- Debido a que el formulario para registrar o editar los datos del usuario es el mismo en ambos casos se creó el *partial* llamado `_user.html.erb` y éste es invocado desde las vistas `/app/views/users/new.html.erb` y `/app/views/users/edit.html.erb` con la instrucción:

```
<%= render @user %>
```

- *Authlogic* proporciona un script generador llamado `UserSession` el cual nos permite crear un modelo para administrar toda la lógica de sesión:

```
$ ruby script/generate session user_session
```

- En este caso tampoco se generaron automáticamente el controlador y las vistas necesarias, por lo que creamos el controlador `UserSessions` y la vista `new`:

```
$ ruby script/generate controller UserSessions new
```

- En el controlador se definieron las funciones para crear, iniciar y terminar las sesiones:

```
def new
  @user_session = UserSession.new
end

def create
  @user_session = UserSession.new(params[:user_session])
  if @user_session.save
    flash[:notice] = "Sesión Iniciada"
    redirect_to :controller => "ips", :action => "index"
  else
    render :action => 'new'
  end
end

def destroy
  @user_session = UserSession.find
  @user_session.destroy
  flash[:notice] = "Sesión Finalizada"
```

```
  redirect_to :controller => "user_sessions", :action => "new"  
end
```

- Se editó el archivo de rutas ubicado en /config/routes.rb para que el controlador sea considerado un recurso y presentar URL's limpias. El formulario de logeo se estableció como la página de inicio, si no se inicia sesión para autenticarse como usuario registrado no se podrá acceder al sistema:

```
map.root :controller => 'user_sessions', :action => 'new'  
map.resources :user_sessions  
map.login 'login', :controller => 'user_sessions', :action => 'new'  
map.logout 'logout', :controller => 'user_sessions', :action => 'destroy'
```

- Para tener en todos los controladores el bloqueo de usuarios se definieron los siguientes métodos en el controlador global /app/controllers/application_controller.rb:

```
helper_method :current_user  
  
def current_user_session  
  return @current_user_session if defined?(@current_user_session)  
  @current_user_session = UserSession.find  
end  
  
def current_user  
  @current_user = current_user_session && current_user_session.record  
end  
  
def require_user  
  unless current_user  
    store_location  
    flash[:notice] = "Es necesario iniciar sesión"  
    redirect_to new_user_session_url  
    return false  
  end  
end  
def require_no_user  
  if current_user  
    store_location  
    flash[:notice] = "Iniciar sesión"  
    #redirect_to account_url  
    return false  
  end  
end  
  
def store_location  
  session[:return_to] = request.request_uri
```

```
end

def redirect_back_or_default(default)
  redirect_to(session[:return_to] || default)
  session[:return_to] = nil
end
```

- Por último para restringir el acceso a usuarios no registrados y hacer cumplir los métodos anteriores, fue necesario establecer un filtro para realizar la autenticación antes de ejecutar cualquier otra acción, se escribió la siguiente instrucción en los controladores que lo requerían:

```
before_filter :require_user
```

Instalación e implementación de Declarative_Authorization

Plugin para *Rails* que permite especificar autorizaciones basadas en roles. Debido a que las reglas de autorización se declaran en un archivo, el desarrollador debe indicar qué roles tienen permitido acceder a ciertos controladores.

Para el caso del sistema Matlatl se contaban con las siguientes características:

Se debían contar con dos modelos *Role* y *User*.

Un usuario puede tener asignado de uno a muchos roles.

Un rol puede tener asignado de uno a muchos usuarios.

Como se contaba con una relación muchos a muchos fue necesario crear un modelo de asignaciones (*Assignment*) para dividir esta relación. Ver Figura 5.10

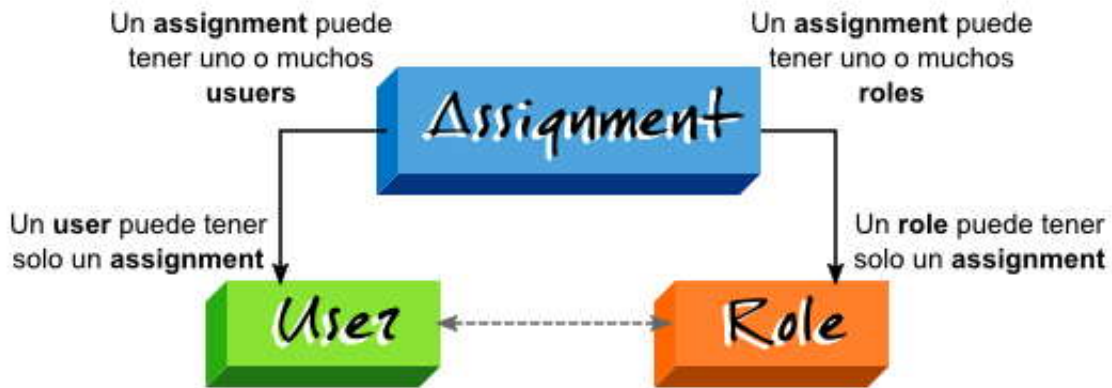


Figura 5.10 Creación del modelo de Asignación para dividir relación muchos a muchos

- Usando la instrucción scaffold se generó la vista, el controlador y el modelo de *Role*:

```
$ ruby script/generate scaffold model role name:string
```

- Se generó el modelo *Assignment*:

```
$ ruby script/generate model assignment user_id:integer role_id:integer
```

- Se actualizaron las tablas:

```
$ rake db:migrate
```

- Se editaron los modelos involucrados en la relación muchos a muchos para establecer sus asociaciones:

1. Para users.rb:

```
has_many :assignments  
has_many :roles, :through => :assignments
```

2. Para assignment.rb:

```
belongs_to :role  
belongs_to :user
```

3. Para role.rb:

```
has_many :assignments  
has_many :users, :through => :assignments
```

- A partir de este momento será posible asociar un rol a cualquier usuario. Se procedió a

editar la vista `/app/views/users/_user.html.erb` para agregar el código correspondiente a un `check_box_tag`, el cual nos permitirá asignar un tipo de usuario a cada registro que se realice:

```
<% for role in Role.find(:all) %>
  <%= check_box_tag "user[role_ids][]", role.id, @user.roles.include?(role) %>
  <%= role.name %>
<% end %>
```

- Para permitir un array vacío antes de hacer las asignaciones y poder actualizar los datos se agregó al controlador `/app/controllers/users_controller.rb`, al principio de la acción `update`, la siguiente instrucción:

```
params[:user][:role_ids] ||= []
```

- A continuación se indicó en `/config/environment.rb` la instrucción de configuración de gemas para `declarative_authorization`:

```
config.gem "declarative_authorization", :source => "http://gemcutter.org"
```

- Ejecutamos en la terminal:

```
rake gems:install
```

Los usuarios que se crearon son Administrador y Usuario, ya que fueron los solicitados por la Secretaría Técnica de Cómputo del IFUNAM, y los permisos asignados a cada uno de ellos son:

1. Administrador: Crear, ver, editar y eliminar en los controladores `people`, `ips`, `macs`, `roles` y `users`, es decir que tendrá todos los permisos.
2. Usuario: Ver en los controladores `people`, `ips` y `macs`. Este tipo de usuario tiene permisos limitados, solo podrá consultar información.

- En la carpeta `/config` se creó el archivo llamado `authorization_rules.rb` donde se definieron los permisos para cada rol, a continuación se muestra el archivo final:

```
authorization do
  role :administrador do
    has_permission_on [:people, :ips, :macs, :roles, :users], :to => [:index, :show, :new, :create, :edit, :update, :destroy, :auto_complete_for_ip_person_id, :auto_complete_for_mac_ip_id]
  end
end
```

```
role :usuario do
  has_permission_on [:people, :ips, :macs], :to => [:index, :show]
end
end
```

- Para que *declarative_authorization* sea capaz de reconocer los permisos de los roles asignados es necesario crear un nuevo metodo llamado `role_symbols` en el modelo `/app/models/user.rb`:

```
def role_symbols
  roles.map do |role|
    role.name.underscore.to_sym
  end
end
```

- Se agregó un filtro en `/app/controllers/application_controller.rb` para poder identificar al usuario de la sesión:

```
helper :all
protect_from_forgery
before_filter { |c| Authorization.current_user = c.current_user }
```

- Se agregó en cada uno de los controladores un filtro para restringir el acceso:

```
filter_resource_access
```

- Al finalizar se descubrió que al usar *declarative_authorization* la gema *auto_complete* no funcionaba adecuadamente ya que no tenía permisos asignados, por lo que fue necesario denominarlas como una colección adicional al plugin y para ello se editaron los siguientes controladores:

1. Para `/app/controllers/ips_controller.rb` el filtro se definió:

```
filter_resource_access :additional_collection => :auto_complete_for_ip_person_id
```

2. Para `/app/controllers/macs_controller.rb` el filtro se definió:

```
filter_resource_access :additional_collection => :auto_complete_for_mac_ip_id
```

5.5 Liberar la versión de producción.

1. Instalamos el módulo Phusion Passenger, el cual permite poner en marcha las aplicaciones web desarrolladas en RoR.

- Primero instalamos las librerías requeridas:

```
# apt-get install libc6 libpcre3 libpcre3-dev libpcrecpp0 libssl0.9.8 libssl-dev zlib1g zlib1g-dev lsb-base
```

- Se creó un directorio para alojar nuestra aplicación:

```
# mkdir -p /var/www/monitor/
```

- Fué necesario cambiar de dueño y grupo al directorio con todo su contenido.

```
# chown -R www-data:www-data /var/www/monitor/
```

- Finalizamos instalando la gema de *passenger* y el módulo para el servidor web nginx.

```
# gem install passenger  
# passenger-install-nginx-module
```

Al ejecutar el instalador de Nginx elegimos la primera opción, para permitir que Passenger realice el trabajo de instalación y conservamos el directorio de ubicación predeterminado `/opt/nginx`.

Nginx es un servidor *http* y *proxy* inverso de alto rendimiento y muy liviano, desarrollado por el Ruso Igor Sysoev y liberado en el 2004 con licencia *Open Source*. Nginx se ha dado a conocer por su estabilidad, su gran conjunto de características, por su configuración sencilla y su bajo consumo de recursos. En la actualidad alberga cerca del 6% de todos los dominios del mundo³⁹.

2. Fue necesario modificar el archivo de configuración predeterminado de nginx para agregar las opciones que requería el sistema. El archivo se encuentra ubicado en `/opt/nginx/conf/nginx.conf` y al realizar los cambios se obtuvo lo siguiente:

```
user www-data www-data;  
worker_processes 1;
```

³⁹ Fuente: Sitio Web Nginx.


```

events {
    worker_connections 1024;
}

http {
    passenger_root /usr/lib/ruby/gems/1.8/gems/passenger-2.2.10;
    passenger_ruby /usr/bin/ruby1.8;

    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    gzip on;

    keepalive_timeout 65;

    server {
        listen 80;
        server_name localhost;
        root /var/www;
        passenger_enabled on;
        passenger_base_uri /matlatl;
        client_max_body_size 50M;

        if (-f $document_root/system/maintenance.html) {
            rewrite ^(.*)$ /system/maintenance.html break;
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}

```

- Se creó un enlace simbólico, al que llamamos matlatl, que apunta al directorio public/ de nuestra aplicación, para proporcionar un URI⁴⁰ (Uniform Resource Identifier – Identificador Uniforme de Recursos) a nuestro sistema web:

```
# ln -s /var/www/monitor/public/ matlatl
```

3. Al instalar nginx no se crea el archivo init, por lo que se descargó el script de la página oficial, el cual funciona para iniciar, detener y reiniciar el servidor, pero principalmente para

⁴⁰ Un “fragmento” o cadena corta de caracteres que permite identificar una parte del recurso principal, es nuestro caso la dirección web del sistema.

asegurar que inicia automáticamente al arrancar el sistema operativo.

- Teniendo los permisos de root se creó en el directorio /etc/init.d/ el archivo llamado nginx y se copió el siguiente contenido:

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:      nginx
# Required-Start:  $all
# Required-Stop:  $all
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: starts the nginx web server
# Description:    starts nginx using start-stop-daemon
### END INIT INFO

PATH=/opt/nginx/sbin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/opt/nginx/sbin/nginx
NAME=nginx
DESC=nginx

test -x $DAEMON || exit 0

# Include nginx defaults if available
if [ -f /etc/default/nginx ] ; then
    . /etc/default/nginx
fi

set -e

case "$1" in
    start)
        echo -n "Starting $DESC: "
        start-stop-daemon --start --quiet --pidfile /opt/nginx/logs/$NAME.pid \
            --exec $DAEMON -- $DAEMON_OPTS
        echo "$NAME."
        ;;
    stop)
        echo -n "Stopping $DESC: "
        start-stop-daemon --stop --quiet --pidfile /opt/nginx/logs/$NAME.pid \
            --exec $DAEMON
        echo "$NAME."
        ;;
    restart|force-reload)
        echo -n "Restarting $DESC: "
        start-stop-daemon --stop --quiet --pidfile \
            /opt/nginx/logs/$NAME.pid --exec $DAEMON
```

```
sleep 1
start-stop-daemon --start --quiet --pidfile \
    /opt/nginx/logs/$NAME.pid --exec $DAEMON -- $DAEMON_OPTS
echo "$NAME."
;;
reload)
    echo -n "Reloading $DESC configuration: "
    start-stop-daemon --stop --signal HUP --quiet --pidfile    /opt/nginx/logs/$NAME.pid \
        --exec $DAEMON
    echo "$NAME."
    ;;
*)
    N=/etc/init.d/$NAME
    echo "Usage: $N {start|stop|restart|reload|force-reload}" >&2
    exit 1
    ;;
esac

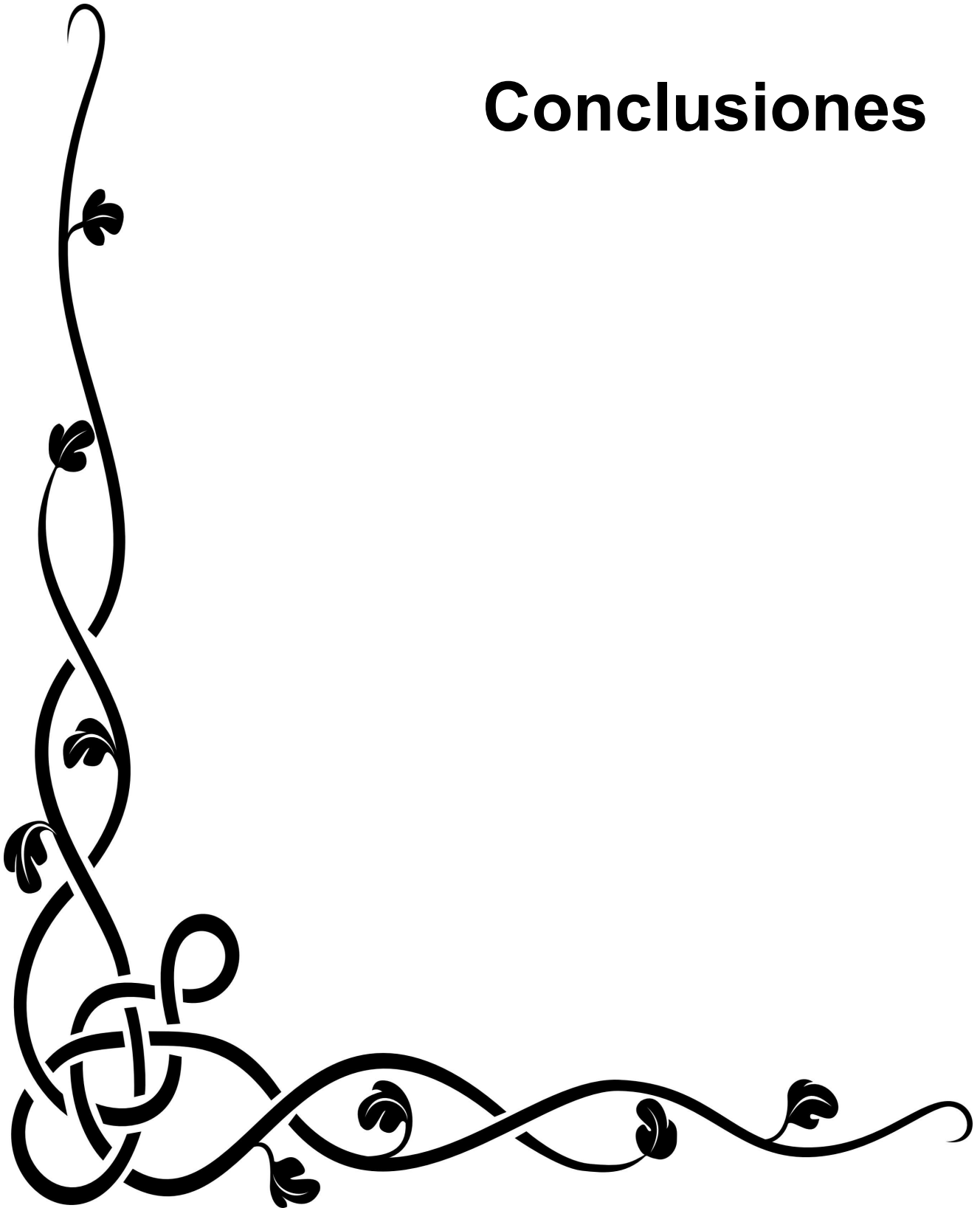
exit 0
```

- Se cambiaron los permisos del archivo y se indicó que debía arrancar al momento de iniciar el sistema:

```
# sudo chmod +x /etc/init.d/nginx
# sudo /usr/sbin/update-rc.d -f nginx defaults
```

A partir de este momento el sistema se encuentra liberado y es posible realizar consultas vía web.

Conclusiones



Conclusiones

Después de pasar por diversas etapas en la realización del presente trabajo, pudimos llegar a la conclusión de que es posible crear sistemas, apoyándonos en el uso de una gran variedad de software, que se adapten a las demandas y necesidades de los usuarios, por esta razón el Sistema Matlat para la Administración del Tráfico de la Red de Datos será de gran utilidad para la Secretaría Técnica de Cómputo y Telecomunicaciones del Instituto de Física de la UNAM, ayudando a agilizar el proceso de actualización de información así como presentando los datos de una forma ordenada.

Al pertenecer a una institución de investigación científica como el IFUNAM, en constante crecimiento y desarrollo, crece la demanda de contar con una conexión a Internet confiable. Es por esto que el encargado de la red necesita realizar un monitoreo constante para mantener el servicio en condiciones óptimas.

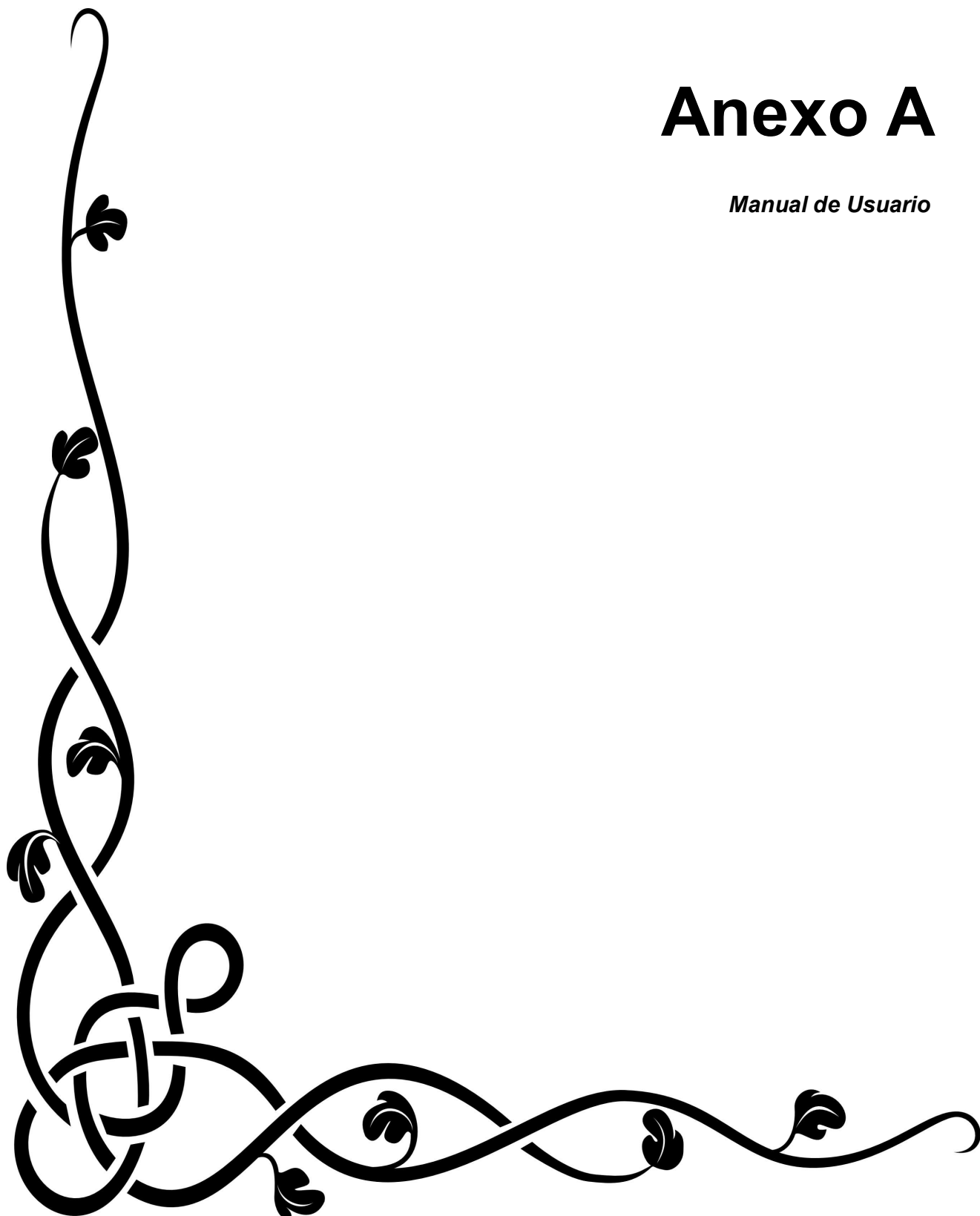
Anteriormente el registro de datos se apoyaba en libretas y en hojas de cálculo, lo que permitía que el rango de error en la actualización de la información fuera grande, ya que no siempre se portaba el documento electrónico o las hojas donde se apuntaban las modificaciones realizadas podían perderse.

En la actualidad, con la implementación del Sistema Matlatl, es posible registrar vía web cualquier modificación o actualización que se realice en tiempo real y desde cualquier posición dentro del instituto, además de permitir la consulta de datos de los usuarios y contar con el apoyo visual al generar gráficas que muestran el tráfico de red de cada usuario en determinado tiempo, facilita la identificación de usuarios de alta demanda y los problemas de la red generados por el tráfico. Su diseño permite la creación de nuevos módulos si son requeridos en el futuro. Por esto concluyo que se dio solución a los requisitos iniciales y deja una puerta abierta para facilitar el crecimiento o expansión del mismo proyecto.

También se concluye que el uso de software libre es una gran alternativa para la solución de problemas, ya que se puede ofrecer un producto de excelente calidad sin la necesidad de realizar una gran inversión y permite combinar el uso de diferentes herramientas de software.

Anexo A

Manual de Usuario



ANEXO A

Manual de Usuario

El sistema Matlatl es una aplicación que permite gestionar la información de la red de datos del *Instituto de Física de la UNAM (IFUNAM)*.

El objetivo que se persigue con el presente manual es:

- Dar a conocer a los usuarios finales las características y la forma de uso del Sistema Matlatl.

1. Ingreso al Sistema Matlatl

Para ingresar al sistema, es necesario solicitar previamente una cuenta de acceso al administrador y pertenecer a la Secretaría Técnica de Cómputo, Telecomunicaciones y Fotografía del IFUNAM. Posteriormente con ayuda del navegador web de su preferencia dirigirse (usando la barra de direcciones como se muestra en la Figura 1) al sitio ubicado en la siguiente dirección electrónica:

<http://132.248.7.247/matlatl>



Figura 1. Dirección web del Sistema Matlatl

Al dar un enter automáticamente se visualizará la página de logeo (Figura 2):



Figura 2. Autenticación de usuarios.

2. Iniciar Sesión

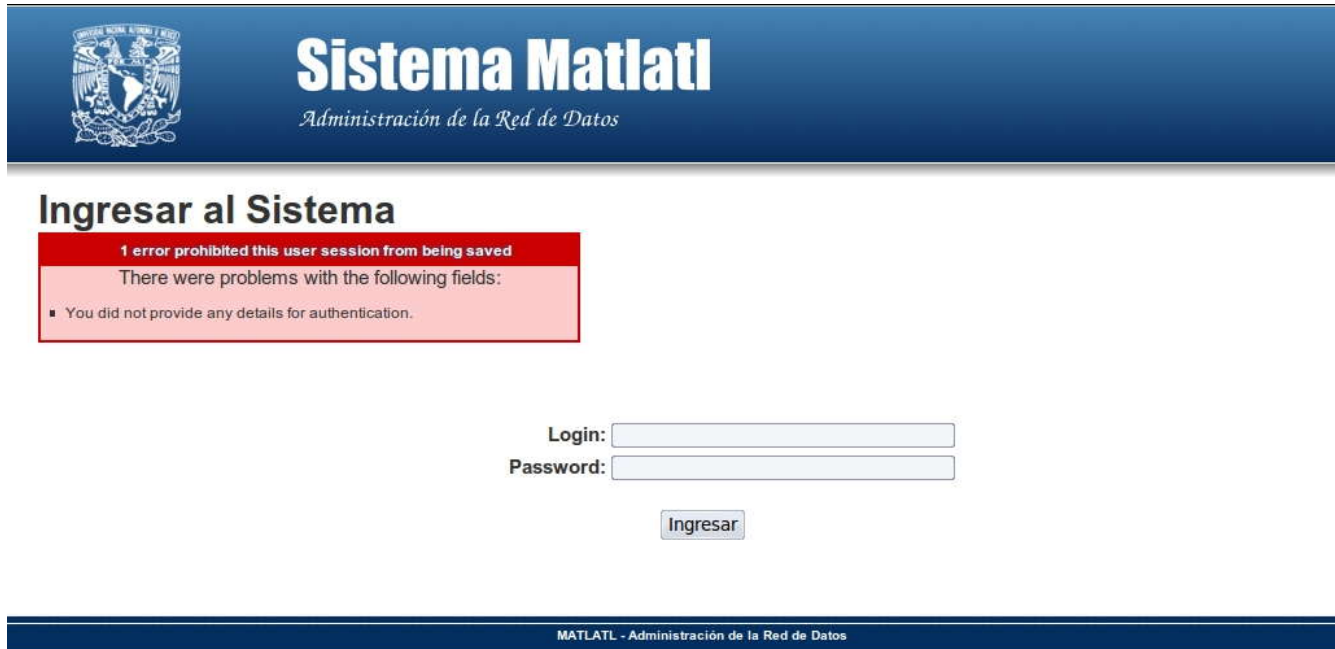
La página de logeo, Figura 3, nos permitirá el acceso al sistema para consultar los datos de red, por lo que deberá ingresar los siguientes datos:

1. **Login:** El nombre de usuario que eligió.
2. **Password:** Su contraseña.
3. Para finalizar de un click con el mouse en el botón “**Ingresar**”.



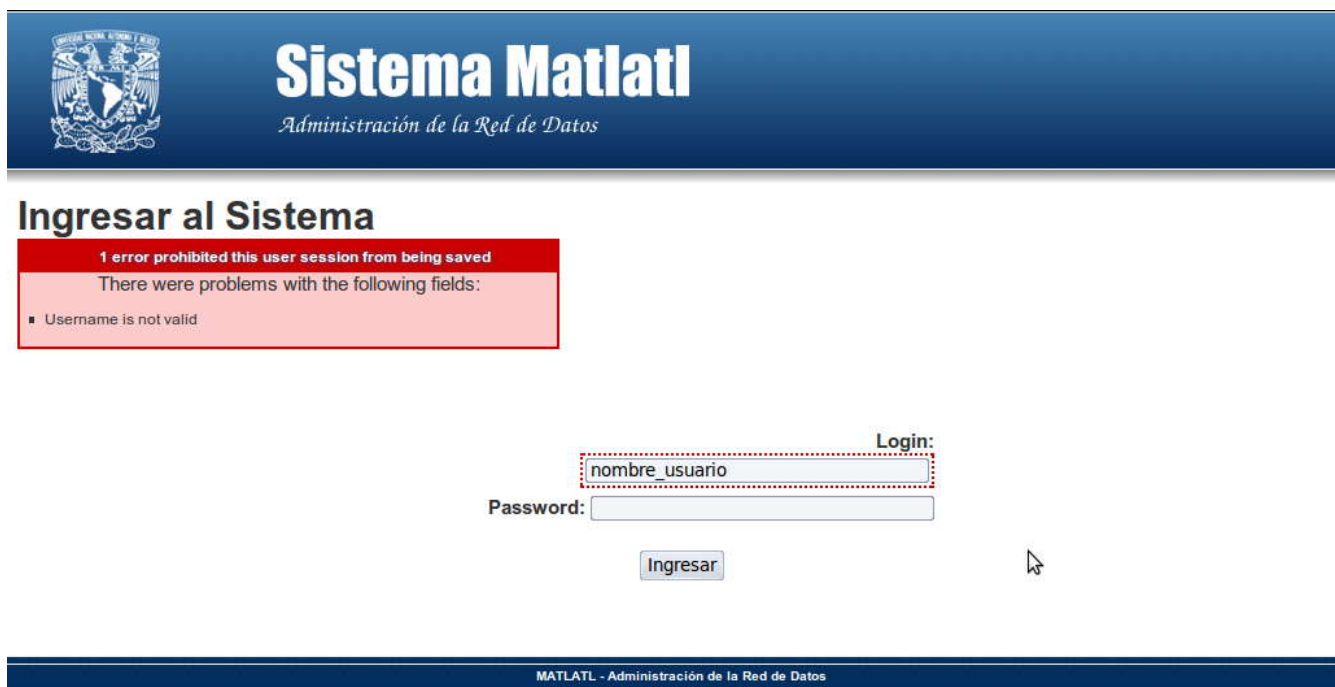
Figura 3. Pasos a seguir para ingresar al sistema Matlatl

Este proceso es necesario para verificar que es un usuario registrado, si sus datos son incorrectos observará una leyenda en un recuadro rojo indicándole cuál es su posible error. Ver Figura 4 y Figura 5.



The screenshot shows the login interface for 'Sistema Matlatl' (Administración de la Red de Datos). A red error box at the top left contains the message: '1 error prohibited this user session from being saved. There were problems with the following fields: You did not provide any details for authentication.' Below this, the 'Login:' and 'Password:' fields are empty, and the 'Ingresar' button is visible. A footer bar at the bottom reads 'MATLATL - Administración de la Red de Datos'.

Figura 4. Ejemplo de error de autenticación.



The screenshot shows the same login interface as Figure 4. The red error box now displays: '1 error prohibited this user session from being saved. There were problems with the following fields: Username is not valid'. The 'Login:' field contains the text 'nombre_usuario' and is highlighted with a red dashed border. The 'Password:' field is empty. The 'Ingresar' button is visible. A footer bar at the bottom reads 'MATLATL - Administración de la Red de Datos'.

Figura 5. Ejemplo de error de llenado de datos.

NOTA

A partir de este momento las tareas permitidas dentro del sistema varían de acuerdo al tipo de usuario que le fue asignado, por el momento solo se crearon dos tipos:

- **Usuario Normal:** Tiene acceso a las páginas de consulta de direcciones IP, direcciones MAC y a la información de los usuarios de la red de datos del instituto. No tiene permisos para editar, crear o borrar.
- **Administrador:** Además de tener los permisos de un usuario normal, es capaz de crear, editar y borrar cualquier información almacenada en el sistema, así como dar de alta a los usuarios que podrán ingresar y crear nuevos tipos de usuarios.

Si aprueba la autenticación podrá visualizar la página principal del sistema (Figura 6) y observará la leyenda **"Sesión Iniciada"** con letras verdes en la parte superior central.

The screenshot shows the 'Sistema Matlatl' web interface. At the top, there is a navigation menu with 'IP'S', 'USUARIOS', 'MAC'S', 'ROLES', and 'ALTAS'. Below the menu, there is a search bar with the text 'Buscar por IP' and a button 'Ingresar nueva IP'. The main content area is titled 'Listado de Direcciones IP' and contains a table with columns: 'Nombre de Usuario', 'Dirección IP', 'Switch', 'Puerto', 'Direcciones MAC', and actions ('Ver', 'Editar', 'Eliminar'). The table lists various users and their associated IP and MAC addresses. At the bottom of the table, there is a pagination control showing '« Anterior 1 2 3 4 5 6 7 8 9 ... 50 51 Siguiente »'. The footer of the page reads 'MATLATL - Administración de la Red de Datos'.

Nombre de Usuario	Dirección IP	Switch	Puerto	Direcciones MAC	Ver	Editar	Eliminar
Cómputo - Vax	20.20.7.1			00:04:76:D1:88:12	Ver	Editar	Eliminar
Cómputo - Vax	20.20.7.2				Ver	Editar	Eliminar
SD	20.20.7.3				Ver	Editar	Eliminar
Cómputo - Acelerador .7 MeV	20.20.7.4			00:11:88:53:94:27 00:11:88:3C:8D:8C	Ver	Editar	Eliminar
SD	20.20.7.5				Ver	Editar	Eliminar
Cómputo - Acelerador 2 MeV	20.20.7.6			00:11:88:53:94:6F 00:01:F4:83:B6:0B	Ver	Editar	Eliminar
SD	20.20.7.7				Ver	Editar	Eliminar
SD	20.20.7.8				Ver	Editar	Eliminar
Cómputo - Acelerador 5.5 MeV	20.20.7.9			00:11:88:60:0D:B2 00:11:88:53:94:27	Ver	Editar	Eliminar
Servidor Shadowwall	20.20.7.10			00:0D:56:D4:32:F3	Ver	Editar	Eliminar

Figura 6. Índice de direcciones IP.

Un elemento importante que nos permitirá movernos entre las diferentes páginas del sistema es el **menú de navegación**, el cual se encuentra en la parte superior derecha (Figura 7). Este menú se muestra constantemente para facilitar la transición de una página a otra. Solo es necesario dar un click con el mouse en el tema que desee consultar.



Figura 7. Menú de navegación

3. Funciones para la página de Direcciones IP

El "**Listado de Direcciones IP**" es la página de inicio del sistema (Figura 8), por lo que siempre que se ingrese a él será la primera que se muestre. Sus elementos principales son:

1. Buscador de IPs.
2. Listado de datos por dirección IP.
3. Funciones de la aplicación.
4. Paginador.

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

Cerrar sesión

Listado de Direcciones IP

1 Buscar por IP

3 Ingresar nueva IP

Nombre de Usuario	Dirección IP	Switch	Puerto	Direcciones MAC	Ver	Editar	Eliminar
Cómputo - Vax	20.20.7.1			00:04:76:D1:88:12	Ver	Editar	Eliminar
Cómputo - Vax	20.20.7.2				Ver	Editar	Eliminar
SD	20.20.7.3				Ver	Editar	Eliminar
Cómputo - Acelerador .7 MeV	20.20.7.4			00:11:88:53:94:27 00:11:88:3C:8D:8C	Ver	Editar	Eliminar
SD	20.20.7.5				Ver	Editar	Eliminar
Cómputo - Acelerador 2 MeV	20.20.7.6			00:11:88:53:94:6F 00:01:F4:83:B6:0B	Ver	Editar	Eliminar
SD	20.20.7.7				Ver	Editar	Eliminar
SD	20.20.7.8				Ver	Editar	Eliminar
Cómputo - Acelerador 5.5 MeV	20.20.7.9			00:11:88:60:0D:B2 00:11:88:53:94:27	Ver	Editar	Eliminar
Servidor Shadowall	20.20.7.10			00:0D:56:D4:32:F3	Ver	Editar	Eliminar

2

4 « Anterior 1 2 3 4 5 6 7 8 9 ... 50 51 Sigulente »

MATLATL - Administración de la Red de Datos

Figura 8. Elementos de la página principal de direcciones IP

3.1. Funciones básicas.

Con el objetivo de facilitar y agilizar la consulta de datos se agregó un buscador, todo usuario puede realizar la búsqueda de una dirección en específico, solo debe ingresar en el cuadro de texto el dato completo o parcial y dar un click en el botón **“Buscar por IP”**, el resultado mostrará todos los datos que coincidan con el texto.

NOTA

Solo buscará **direcciones IP**, es decir que busca datos con un formato similar a **123.123.123.123**

Otra opción para navegar entre las páginas que muestran los datos de direcciones IP es el paginador, ubicado en la parte inferior central (Figura 9). Se puede navegar de dos formas diferentes:

1. Dar un click en la opción **Anterior** para retroceder páginas o en la opción **Siguiente** para avanzar páginas.
2. Dar un click en el número de página al que se desear ir.

Para identificar la página actual se puede observar en el paginador que el tamaño de fuente es más grande y de color negro.



Figura 9. Paginador del sistema

Para consultar la información completa por dirección IP dar un click con el mouse en la opción **Ver**, ubicada en la última columna del listado, lo cual nos lleva a la página que muestra los datos relacionados a la IP seleccionada. Esta información es la siguiente:

1. Los datos personales del usuario al que se le asignó la dirección IP.
2. Los datos técnicos de la dirección IP.
3. Las gráficas de monitoreo del tráfico de red.

En la Figura 10 se observa la presentación de la información del usuario seleccionado.

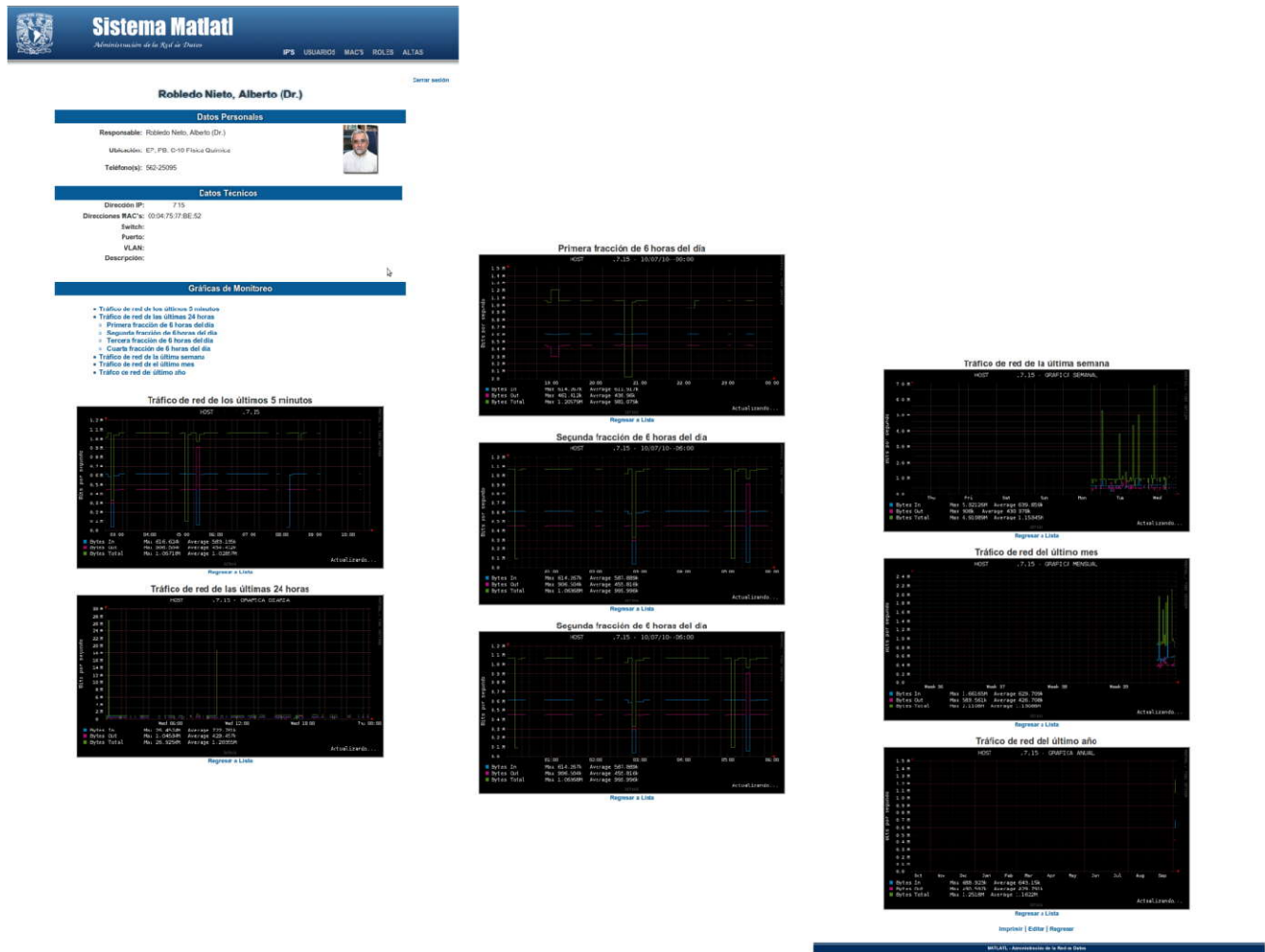


Figura 10. Página para ver datos de la dirección IP


NOTA

Como usuario normal no podrá usar las opciones de **Editar** y **Eliminar** ya que no tiene los permisos necesarios, si da un click en cualquiera de estas funciones será redirigido a la página de inicio y observará en la parte superior central la leyenda “**Lo sentimos pero no tiene los permisos necesarios para realizar esta acción**” en letras verdes.

Los datos personales y técnicos son secciones creadas para consulta de información.

La sección de “**Gráficas de Monitoreo**” se creó como apoyo visual, en ella se podrá observar gráficas que muestran el tráfico de red en el instituto, debido a la gran cantidad de dichas imágenes que se crean se cuenta con un menú que nos enlaza directamente a la gráfica que deseamos consultar. Es necesario dar un click con el mouse a una opción de la lista y automáticamente nos dirige a la imagen.

Si deseamos regresar al listado dar click a la opción “**Regresar a Lista**” que se encuentra debajo de cada gráfica y regresaremos al inicio del menú (Figura 11).



Sistema Matlatl

Administración de la Red de Datos

IP'S
USUARIOS
MAC'S
ROLES
ALTAS

[Cerrar sesión](#)


Robledo Nieto, Alberto (Dr.)

Datos Personales

Responsable: Robledo Nieto, Alberto (Dr.)

Ubicación: EP, PB. C-10 Física Química

Teléfono(s): 562-



Datos Técnicos

Dirección IP: 7.15

Direcciones MAC's: 00:04:75:77:BE:52

Switch:

Puerto:

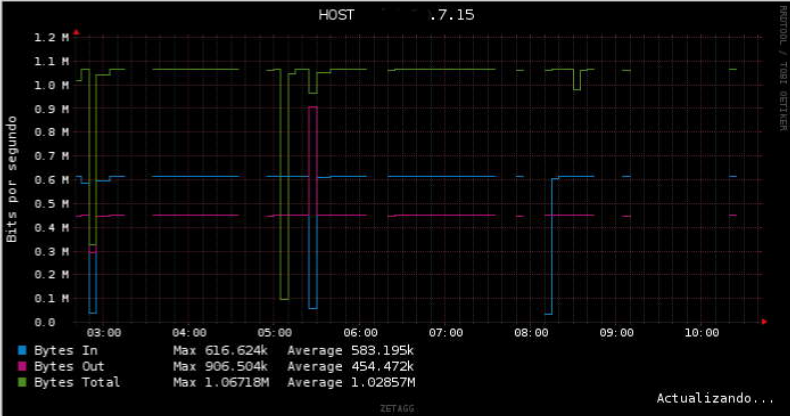
VLAN:

Descripción:

Gráficas de Monitoreo

- [Tráfico de red de los últimos 5 minutos](#)
- [Tráfico de red de las últimas 24 horas](#)
 - [Primera fracción de 6 horas del día](#)
 - [Segunda fracción de 6 horas del día](#)
 - [Tercera fracción de 6 horas del día](#)
 - [Cuarta fracción de 6 horas del día](#)
- [Tráfico de red de la última semana](#)
- [Tráfico de red de el último mes](#)
- [Tráfico de red del último año](#)

Tráfico de red de los últimos 5 minutos



HOST . 7.15		
Bytes In	Max 616.624k	Average 583.195k
Bytes Out	Max 906.504k	Average 454.472k
Bytes Total	Max 1.06718M	Average 1.02857M

ZETAAG Actualizando...

[Regresar a Lista](#)

Figura 11. Índice de gráficas

En la parte inferior de la página se cuenta con tres funciones:

1. Imprimir
2. Editar
3. Regresar

Al dar un click en la opción **Imprimir** se mostrará la ventana que solicita los datos necesarios para imprimir la página actual. Estas opciones pueden variar de acuerdo a la configuración de cada usuario.

Al dar un click en la opción **Editar** seremos redirigidos a la página de inicio y se mostrará en la parte superior la leyenda **“Lo sentimos pero no tiene los permisos necesarios para realizar esta acción”** en letras verdes ya que solo el administrador puede realizar esta acción.

NOTA

Si encuentra un error o desea modificar algún dato deberá enviar un correo electrónico al administrador indicando los cambios y el motivo de dichos cambios.

Al dar un click en la opción **Regresar** se mostrará la página inicial del Listado de Direcciones IP.

3.2. Funciones extras.

Esta sección es dedicada exclusivamente al administrador del Sistema Matlatl. Las acciones que puede realizar incluyen las funciones básicas y las que a continuación explicaremos.

En la página de inicio se cuenta con las opciones de:

1. Agregar nueva IP.
2. Editar.
3. Eliminar.

Al dar click con el mouse en la opción **Agregar nueva IP** se mostrará el siguiente formulario:

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

[Cerrar sesión](#)

Nueva Dirección IP

Ingresar los datos solicitados

Usuario: Dirección IP:

Switch: Puerto: VLAN:

Descripción:

Click

MATLATL - Administración de la Red de Datos

Figura 12. Formulario para ingresar nueva dirección IP

Después de ingresar los datos solicitados dar un click en el botón **Crear nueva IP** para guardar la información (Figura 12). Podrá consultar los datos haciendo uso del buscador o consultando la última página del listado, ya que está organizado por orden de creación.

Al dar click en la opción **Editar** se mostrará el formulario para modificar la información (Figura 13), al ingresar el dato de Usuario se mostrará una lista con los nombres que coinciden con el texto, debe elegir a la persona de la lista para guardar su identificador, de otro modo se mostrará un error.

Después de realizar los cambios dar click en el botón **Guardar cambios**.

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

Cerrar sesión

Editar Dirección IP

Modificar los datos necesarios

Usuario: 3 Dirección IP: 20.20.7.1

Switch: Puerto: VLAN:

Descripción:

Guardar cambios Click

Regresar

MATLATL - Administración de la Red de Datos

Figura 13. Formulario para editar una dirección IP

Si la persona buscada no aparece en la lista quiere decir que no se encuentra dentro del sistema, por lo que deberá ir a la página de Usuarios y dar click en la opción **“Ingresar nuevo usuario”**, dar los datos que se solicitan y crearlo.

NOTA

Es necesario que exista el usuario para poder asignarle una dirección IP.

Al dar click en la opción **Eliminar** se mostrará una ventana de confirmación para borrar el registro (Figura 14), si está seguro de querer eliminarlo dar click en la opción **OK**, si no es así puede dar click en la opción **Cancel**.

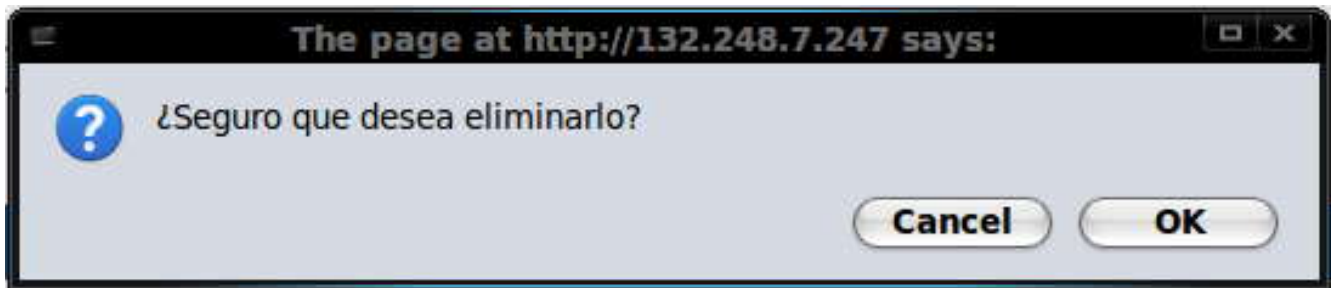



Figura 14. Confirmación para eliminar un registro

4. Funciones para la página de Usuarios de Red

Para ingresar a esta sección usar el menú de navegación ubicado en la parte superior derecha y dar un click en **Usuarios**. La página "**Lista de Usuarios de Red**", ver Figura 15, tiene los siguientes elementos:

1. Buscador de Usuarios.
2. Listado de usuarios de red del instituto.
3. Funciones de la aplicación.
4. Paginador.



Sistema Matlatl











Administración de la Red de Datos

[IP'S](#) [USUARIOS](#) [MAC'S](#) [ROLES](#) [ALTAS](#)

[Cerrar sesión](#)

Lista de Usuarios de Red

1
Buscar por nombre
3 Ingresar nuevo usuario

	Nombre de Usuario	Ubicación	Teléfono	
	Ortiz Salazar, Ma. Esther (Dra.)	Acel 5.5, 1° p C-3 Jefe Fis. Exp.		Ver Editar Eliminar
	García Naumis, Gerardo (Dr.)	EP, PB. C-37 Física Química		Ver Editar Eliminar
	Chávez Lomelí, Efraín Rafael (Dr.)	Acel 5.5, 1° p C-4 Fis. Exp.		Ver Editar Eliminar
	Morales Morales, Juan Gabriel (M en C)	Colisur, 1° p. C-FG-33 Fis. Exp.		Ver Editar Eliminar
	Laboratorio Central de Microscopía	Laboratorio Central de Microscopía Electrónica	552-25064	Ver Editar Eliminar
	Bunge Molina, Carlos Federico (Dr.)	EP, 2° p. C-226 Física Teórica		Ver Editar Eliminar
	Sistemas Complejos	Edificio Principal, 2° p.		Ver Editar Eliminar
	Menchaca Rocha, Arturo A. (Dr.)	Acel 5.5, 1° p. C-1 Fis. Exp		Ver Editar Eliminar
	Santamaría Ortiz, Rubén (Dr.)	EP, 2° p. C-222 Física Teórica		Ver Editar Eliminar
	Magaña Zavala, Carlos R. (Dr.)	EP, PB. C-16 Lab. Microscopía		Ver Editar Eliminar

4 « Anterior 1 2 3 4 5 6 7 8 9 ... 20 21 Siguiente »

MATLATL - Administración de la Red de Datos

Figura 15. Partes del índice de usuarios

4.1. Funciones básicas.

Todo usuario puede realizar la búsqueda por nombre, apellidos o nombre completo de algún usuario de red en específico, solo debe ingresar en el cuadro de texto la información y dar un click en el botón “**Buscar por Usuario**”, el resultado mostrará todos los datos que coincidan con el texto.

NOTA

Solo buscará **nombres** que coincidan con el texto ingresado.

En esta página se cuenta, al igual que en la página de IPs, con un paginador ubicado en la parte inferior central. Se puede navegar de dos formas diferentes:

1. Dar un click en la opción **Anterior** para retroceder páginas o en la opción **Siguiente** para avanzar páginas.
2. Dar un click en el número de página al que se desea ir.

Para identificar la página actual se puede observar en el paginador que el tamaño de fuente es más grande y de color negro.

Para consultar la información completa de cada usuario de red dar un click con el mouse en la opción **Ver**, ubicada en la última columna del listado, para ir a la página que muestra los datos del usuario seleccionado (Figura 15). Esta información es la siguiente:

1. Nombre.
2. Responsable.
3. Ubicación en el instituto.
4. Teléfonos para localizarlo.
5. Direcciones IP que tiene asignadas.

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

[Cerrar sesión](#)

Menchaca Rocha, Arturo A. (Dr.)



Responsable: Menchaca Rocha, Arturo A. (Dr.)
Ubicación: Acel 5.5, 1° p. C-1 Fis. Exp
Teléfono(s):

IP's: 20.20.7.60 [Ver](#)
20.20.7.78 [Ver](#)
20.20.7.134 [Ver](#)
20.20.209.138 [Ver](#)

[Imprimir](#) | [Editar](#) | [Regresar](#)

MATLATL - Administración de la Red de Datos

Figura 15. Vista de la página de datos de un Usuario

NOTA

Como usuario normal no podrá usar las opciones de **Editar** y **Eliminar** ya que no tiene los permisos necesarios.

Al costado derecho de cada dirección IP se observará la opción **Ver**, el cual es un enlace a la página de información de la dirección correspondiente, dar un click para ir a esta página.

En la parte inferior de la página se cuenta con tres funciones:

1. Imprimir
2. Editar
3. Regresar

Al dar un click en la opción **Imprimir** se mostrará la ventana que solicita los datos necesarios para imprimir la página actual. Estas opciones pueden variar de acuerdo a la configuración de cada usuario.

Al dar un click en la opción **Editar** seremos redirigidos a la página de inicio y se mostrará en la parte superior la leyenda “**Lo sentimos pero no tiene los permisos necesarios para realizar esta acción**” en letras verdes ya que solo el administrador puede realizar esta acción.

NOTA

Si encuentra un error o desea modificar algún dato deberá enviar un correo electrónico al administrador indicando los cambios y el motivo de dichos cambios.

Al dar un click en la opción **Regresar** se mostrará la página de la Lista de Usuarios de Red.

4.2. Funciones extras.

Las acciones que puede realizar el administrador de Matlatl en esta sección incluyen las funciones básicas y las que a continuación explicaremos.

En la página de inicio se cuenta con las opciones de:

1. Ingresar nuevo usuario.
2. Editar.
3. Eliminar.

Al dar click con el mouse en la opción **Ingresar nuevo usuario** se mostrará el siguiente formulario (Figura 16):



Figura 16. Formulario para ingresar nuevo usuario.

Ingresar la información solicitada y posteriormente dar click en el botón **Examinar...** Se abrirá una ventana que nos ayudará a buscar en el equipo una fotografía para el nuevo usuario. Después de elegir la imagen dar click en **Crear Usuario** para guardar la información. Podrá consultar los datos haciendo uso del buscador o consultando la última página del listado, ya que está organizado por orden de creación.

Al dar click en la opción **Editar** se mostrará el formulario para modificar la información (Figura 17), después de realizar los cambios dar click en el botón **Guardar cambios**.

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

Cerrar sesión

Editar Datos del Usuario

Nombre: Ortiz Salazar, Ma. Esther (Dra.)

Ubicación: Acel 5.5. 1° p C-3 Jefe Fís. Exp.

Responsable: Ortiz Salazar, Ma. Esther (Dra.)

Teléfono(s):

Foto: Browse...

Guardar cambios

Ver | Regresar **Click**

MATLATL - Administración de la Red de Datos

Modificaz los datos necesazios

Figura 17. Formulario para editar datos de usuario

Al dar click en la opción **Eliminar** se mostrará una ventana de confirmación para borrar el registro, si está seguro de querer eliminarlo dar click en la opción **OK**, si no es así puede dar click en la opción **Cancel**.

5. Funciones para la página de Direcciones MAC

Para ingresar a esta sección usar el menú de navegación ubicado en la parte superior derecha y dar un click en MAC's. La página "**Listado de direcciones MAC's**", ver Figura 18, tiene los siguientes elementos:

1. Buscador de direcciones MAC.
2. Listado de direcciones MAC.
3. Funciones de la aplicación.
4. Paginador.

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

Cerrar sesión

Listado de direcciones MAC's

1 Buscar por MAC

3 Ingresar nueva MAC

Dirección IP	Dirección MAC	Ver	Editar	Eliminar
20.20.7.1	00:04:76:D1:88:12	Ver	Editar	Eliminar
20.20.7.4	00:11:88:53:94:27	Ver	Editar	Eliminar
20.20.7.4	00:11:88:3C:8D:8C	Ver	Editar	Eliminar
20.20.7.6	00:11:88:53:94:6F	Ver	Editar	Eliminar
20.20.7.6	00:01:F4:83:B6:0B	Ver	Editar	Eliminar
20.20.7.9	00:11:88:60:0D:B2	Ver	Editar	Eliminar
20.20.7.9	00:11:88:53:94:27	Ver	Editar	Eliminar
20.20.7.11	00:11:43:CB:52:41	Ver	Editar	Eliminar
20.20.7.11	00:1D:09:9D:D5:14	Ver	Editar	Eliminar
20.20.7.13	00:0A:95:81:BC:C8	Ver	Editar	Eliminar

2

4 « Anterior 1 2 3 4 5 6 7 8 9 ... 42 43 Siguiente »

MATLATL - Administración de la Red de Datos

Figura 18. Partes del índice de direcciones MAC

5.1. Funciones básicas.

Como se ha explicado en las 2 secciones anteriores todo usuario puede realizar búsquedas, en este caso de direcciones mac, ingresar en el cuadro de texto la información y dar un click en el botón **“Buscar por mac”**, el resultado mostrará todos los datos que coincidan con el texto.

Podemos observar que esta página también cuenta con un paginador ubicado en la parte inferior central. Se puede navegar de dos formas diferentes:

1. Dar un click en la opción **Anterior** para retroceder páginas o en la opción **Siguiente** para avanzar páginas.
2. Dar un click en el número de página al que se desear ir.

Para identificar la página actual se puede observar en el paginador que el tamaño de fuente es más grande y de color negro.

Para consultar la información de cada dirección MAC dar un click con el mouse en la opción **Ver**, ubicada en la última columna del listado, para consultar la información de la MAC seleccionada. Esta información es la siguiente:

1. Dirección IP.
2. Dirección MAC.

NOTA

Como usuario normal no podrá usar las opciones de **Editar** y **Eliminar** ya que no tiene los permisos necesarios.

En la parte inferior de la página se cuenta con dos funciones:

1. Editar
2. Regresar

Al dar un click en la opción **Editar** seremos redirigidos a la página de inicio y se mostrará en la parte superior la leyenda **“Lo sentimos pero no tiene los permisos necesarios para realizar esta acción”** en letras verdes ya que solo el administrador puede realizar esta acción.

NOTA

Si encuentra un error o desea modificar algún dato deberá enviar un correo electrónico al administrador indicando los cambios y el motivo de dichos cambios.

Al dar un click en la opción **Regresar** se mostrará la página Lista de direcciones MAC's.

5.2. Funciones extras.

El administrador de Matlatl tiene las funciones de un usuario normal y las que a continuación explicaremos.

En la página de inicio se cuenta con las opciones de:

1. Ingresar nueva MAC.
2. Editar.
3. Eliminar.

Al dar click con el mouse en la opción **Ingresar nueva MAC** se mostrará el siguiente formulario:

The screenshot displays the 'Sistema Matlatl' interface. At the top, there is a blue header with the system logo and name 'Sistema Matlatl' and the subtitle 'Administración de la Red de Datos'. Navigation links for 'IP'S', 'USUARIOS', 'MAC'S', 'ROLES', and 'ALTAS' are visible. A 'Cerrar sesión' link is located in the top right corner. The main content area is titled 'Nueva Dirección MAC'. It features two input fields: 'Identificador de IP:' and 'Dirección MAC:'. A red handwritten note 'Ingresar los datos necesarios' is placed to the left of these fields. Below the input fields are two buttons: 'Crear nueva MAC' and 'Regresar'. A red box with the word 'Click' and a hand icon points to the 'Crear nueva MAC' button. The footer of the page contains the text 'MATLATL - Administración de la Red de Datos'.

Figura 19. Formulario para ingresar nueva dirección MAC

Se solicita el identificador de la dirección IP, solo tiene que escribir en el cuadro de texto la dirección y aparecerá un listado que contiene la información que coincide con los datos que está proporcionando, debe seleccionar la dirección de la lista para obtener el identificador, de cualquier otra forma se mostrará un error.

Dar click en el botón **Crear nueva MAC** para almacenar la información.

NOTA

Es necesario que exista la dirección IP para poder asignarle una dirección MAC.

Al dar click en la opción **Editar** se mostrará el formulario para modificar la información (Figura 20), después de realizar los cambios dar click en el botón **Guardar cambios**.

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

Cerrar sesión

Editar Dirección MAC

Modificar los datos necesarios

Identificador de IP:
1

Dirección MAC:
00:04:76:D1:88:12

Guardar cambios

Ver | Regresar

Click

MATLATL - Administración de la Red de Datos

Figura 20. Editar información de una dirección MAC

Al dar click en la opción **Eliminar** se mostrará una ventana de confirmación para borrar el registro, si está seguro de querer eliminarlo dar click en la opción **OK**, si no es así puede dar click en la opción **Cancel**.

6. Funciones para la página de Roles.

Para ingresar a esta sección usar el menú de navegación ubicado en la parte superior derecha y dar un click en **Roles**. Ver Figura 21.

A esta sección solo podrá ingresar el administrador y tiene como objetivo facilitar la creaciones de roles para diferenciar los tipos de usuarios que pueden existir en el sistema.

En la página de inicio se cuenta con las siguientes opciones:

1. Ingresar nuevo Rol.
2. Editar.
3. Eliminar.

The screenshot shows the 'Sistema Matlatl' interface. At the top, there is a navigation menu with options: IP'S, USUARIOS, MAC'S, **ROLES**, and ALTAS. Below the menu, there is a 'Cerrar sesión' link. The main heading is 'Listado de Roles'. To the right of the heading is a link 'Ingresar nuevo Rol'. Below this is a table with the following content:

Nombre de Rol		
Administrador	Editar	Eliminar
Usuario	Editar	Eliminar

At the bottom of the page, there is a footer with the text 'MATLATL - Administración de la Red de Datos'.

Figura 21. Índice de Roles

Al dar click en la opción **Ingresar nuevo Rol** se mostrará una caja de texto donde deberá ingresar el nombre que le quiere asignar al nuevo Rol, posteriormente de click en el botón **Crear Rol** para guardar la nueva información (Figura 22).

Sistema Matlatl
Administración de la Red de Datos

IP'S USUARIOS MAC'S ROLES ALTAS

Nuevo Rol

Nombre del Rol:

Crear Rol

[Regresar](#)

MATLATL - Administración de la Red de Datos

Figura 22. Crear nuevo Rol

En la parte inferior central de la página se encuentra la opción de **Regresar**, al dar click con el mouse en ella regresará a la página principal de Roles.

Si da click en la opción **Editar** podrá modificar el nombre del rol seleccionado, después de realizar los cambios de click en el botón **Guardar Cambios**.

Al dar click en la opción **Eliminar** se mostrará una ventana de confirmación para borrar el registro, si está seguro de querer eliminarlo dar click en la opción **OK**, si no es así puede dar click en la opción **Cancel**.

7. Funciones para la página de Altas.

Para ingresar a esta sección usar el menú de navegación ubicado en la parte superior derecha y dar un click en **Altas**. Ver Figura 23.

Esta sección es exclusiva del administrador y tiene como objetivo dar de alta a los usuarios que tendrán permiso para ingresar al sistema.

En la página principal podrá observar un listado de los usuarios registrados y las siguientes opciones:

1. Ingresar nuevo usuario.
2. Editar.
3. Eliminar.

[Cerrar sesión](#)

Lista de Usuarios

[Ingresar nuevo Usuario](#)

Nombre de Usuario	Email		
nglez	nglez@fisica.unam.mx	Editar	Eliminar
goz	goz@fisica.unam.mx	Editar	Eliminar
naena	naena@fisica.unam.mx	Editar	Eliminar
zetagg	issetgg@fisica.unam.mx	Editar	Eliminar

MATLATL - Administración de la Red de Datos

Figura 23. Índice de Altas

Al dar click en la opción **Ingresar nuevo Usuario** se mostrará un formulario, el cual deberá ser llenado con la información que proporcione la persona (Figura 24). Observe que se muestran dos *checkbox*, se utilizarán para indicar el tipo de usuario que desea crear, esta información es generada con los datos almacenados en Roles. Si crea o elimina conceptos en esta parte se verá el resultado habiendo más o menos opciones. Por ejemplo, si crea un nuevo rol llamado Estudiante tendrá la opción de elegir entre Administrador Estudiante o Usuario.

Al finalizar de click en el botón **Crear Usuario** para guardar la información.

The screenshot shows the 'Nuevo Perfil de Usuario' (New User Profile) form in the 'Sistema Matlatl' interface. The header includes the system logo and name, the subtitle 'Administración de la Red de Datos', and navigation links for IP'S, USUARIOS, MAC'S, ROLES, and ALTAS. The form itself has four input fields: 'Login:', 'Email:', 'Password:', and 'Confirmar Password:'. Below these fields are two radio buttons: 'Administrador' (selected) and 'Usuario'. At the bottom of the form are two buttons: 'Crear Usuario' and 'Regresar'. A footer bar at the bottom of the page reads 'MATLATL - Administración de la Red de Datos'.

Figura 24. Formulario para ingresar nuevo usuario del sistema.

Si da click en la opción **Editar** podrá modificar la información del usuario, después de realizar los cambios de click en el botón **Crear Usuario**.

Al dar click en la opción **Eliminar** se mostrará una ventana de confirmación para borrar el registro, si está seguro de querer eliminarlo dar click en la opción **OK**, si no es así puede dar click en la opción **Cancel**.

8. Orden de Borrado.

Con el fin de evitar errores en el sistema, es importante seguir un determinado orden de borrado para eliminar todo un registro con sus asignaciones.

Paso 1.

Eliminar la o las direcciones MAC asignadas a una dirección IP.

Paso 2.

Eliminar la o las direcciones IP asignadas a un usuario.

Paso 3.

Después de seguir los pasos anteriores se han eliminado las asignaciones y es posible eliminar al usuario.

9. Terminar Sesión.

Es recomendable que al terminar la consulta o la actualización de datos cierre su sesión, podrá hacerlo al dar click en la opción **Cerrar Sesión** que se encuentra en la parte superior derecha.

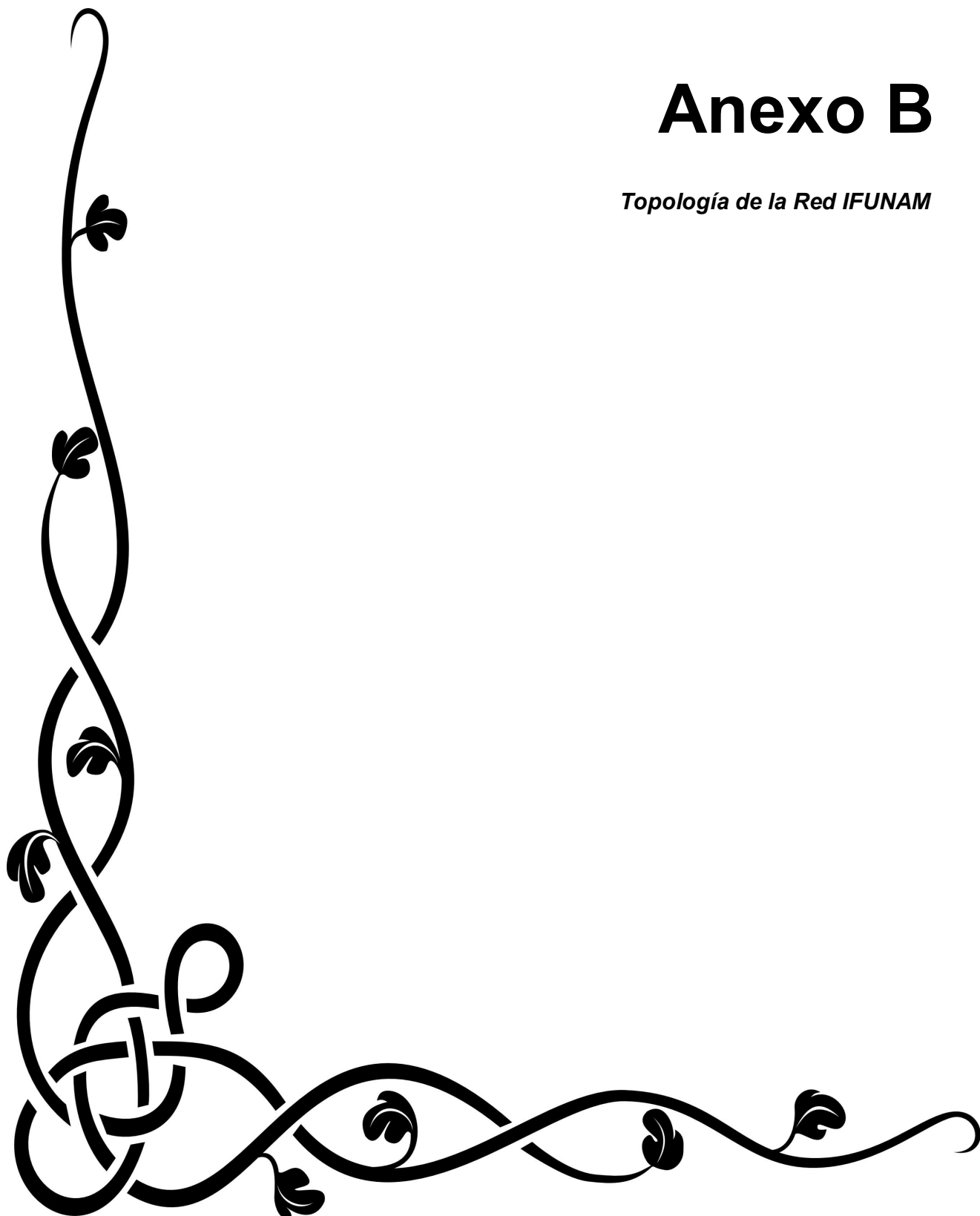
Este elemento se muestra en todas las páginas dentro del sistema y al dar click en él será redireccionado a la página de logeo y observara en la parte superior central la leyenda **“Sesión Finalizada”** en letras verdes (Figura 25).



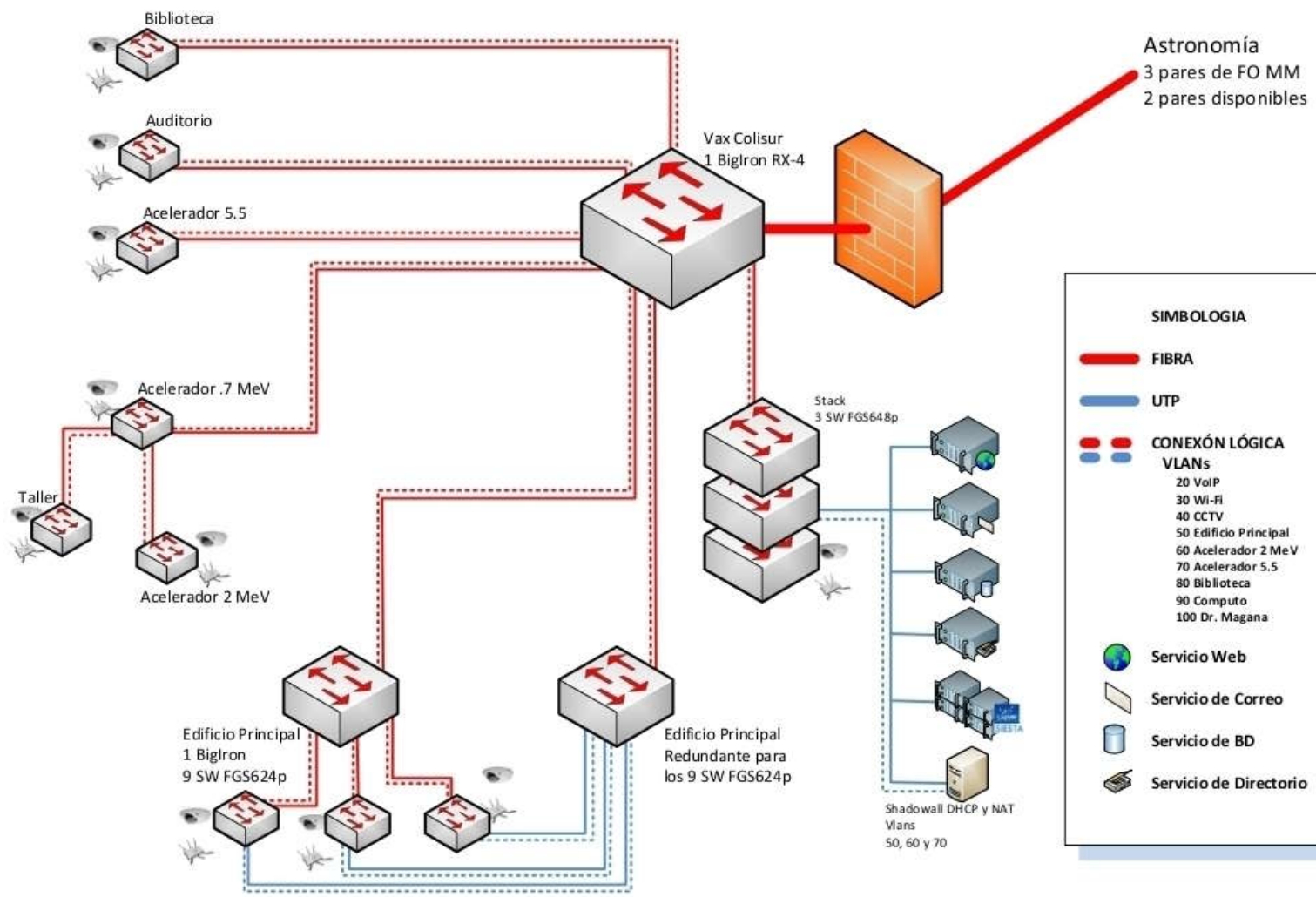
Figura 25. Finalizar sesión

Anexo B

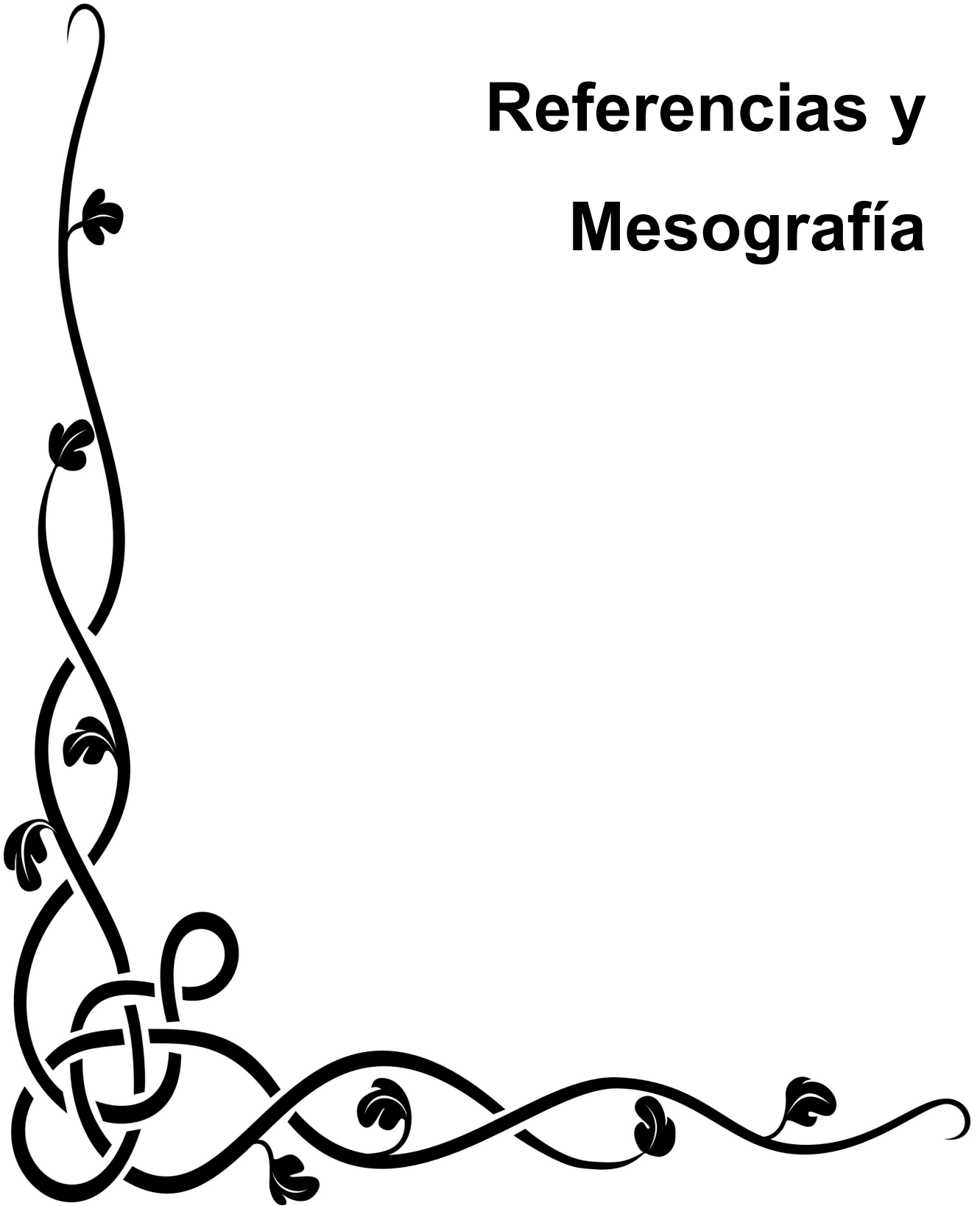
Topología de la Red IFUNAM



Topología de la Red IFUNAM



Referencias y Mesografía



Referencias

AWARD, M. A., (2005). *A comparison between Agile and Traditional Software development methodologies*. Honours Programme of the School of Computer Science and software Engineering, The University of Western Australia.

DATE, C. J., (2000). *Introducción a los Sistemas de Bases de Datos*. México. Pearson Educación de México.

DROMS, RALPH; Lemon, Ted, (2003). *The DHCP Handbook*. Estados Unidos. Sams Publishing.

HAGEN, WILLIAM VON, (2007). *Ubuntu Linux Bible*. Estados Unidos. Wiley Publishing.

KRETCHMAR, JAMES M., (2003). *Open Source Network Administration*. Estados Unidos. Prentice Hall PTR.

LANE, JONATHAN, Moscovitz, Meitar, Lewis, Joseph R., (2008). *Foundation Website Creation with CSS, XHTML, and JavaScript*. Estados Unidos. Apress.

MARSHALL, KEVIN, Pytel, Chad, Yurek, Jon, (2007). *Pro Active Record for Ruby: Databases with Ruby and Rails*. Estados Unidos. Apress.

NAUGLE, MATTHEW G., (1998). *Illustrated TCP/IP – A Graphic Guide to the Protocol Suite*. Estados Unidos. John Wiley & Sons.

RUBY, SAM, Thomas, Dave, Heinemeier Hansson, David, et al., (2007). *Agile Web Development with Rails*. Estados Unidos. The Pragmatic Programmers.

SCHRODER, CARLA, (2008). *Linux Networking Cookbook*. Estados Unidos. O'Reilly Media.

TANENBAUM, ANDREW S., (2003). *Redes de Computadoras*. México. Pearson Educación de México.

ZHANG, PENG, (2008). *Industrial control Technology: A Handbook for Engineers and Researchers*.

Mesografía

Gráficas RRD basadas en IPFM [*Consulta: Noviembre 28, 2008*]

<< <http://people.via.ecp.fr/~lool/rrd/ipfm/> >>

El verdadero origen del Internet [*Consulta: Enero 5, 2009*]

<< <http://www.tuobra.unam.mx/publicadas/010815132146-Title.html> >>

The history of amateur radio [*Consulta: Enero 5, 2009*]

<< <http://www.astrosurf.com/luxorion/qs1-ham-history11.htm> >>

Tipos de Redes [*Consulta: Enero 14, 2009*]

<< <http://elqui.dcsc.utfsm.cl/apuntes/redes/2001/pdf/0-2-Tipos-de-Redes.pdf> >>

Redes de Difusión [*Consulta: Enero 15, 2009*]

<< <http://monarrez.itchihuahua.edu.mx/red-co/trab1/trab1.htm> >>

The Internet Protocol Architecture and Programming [*Consulta: Enero 19, 2009*]

<< http://agenda.ictp.trieste.it/agenda_links/smr1335/networking/node1.html >>

Topology [*Consulta: Enero 20, 2009*]

<< <http://fcit.usf.edu/Network/chap5/chap5.htm> >>

Topologías de Red [*Consulta: enero 26, 2009*]

<< <http://www.eveliux.com/mx/topologias-de-red.php> >>

Topología de Red [*Consulta: Enero 20, 2009*]

<< <http://www.bloginformatico.com/topologia-de-red.php> >>

Modelo OSI [*Consulta: Febrero 4, 2009*]

<< http://docencia.izt.uam.mx/sgm8/Archivos/osi_nuevo.ppt >>

Apuntes de Redes [*Consulta: Febrero 10, 2009*]

<< <http://www.utomde.com/asigna/redes/MisApuntes.pdf> >>

OpenSSH [*Consulta: Febrero 25, 2009*]

<< <http://www.openssh.com/es/index.html> >>

Red Hat Enterprise Linux 4: Manual de referencia [*Consulta: Febrero 25, 2009*]

<< <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html> >>

Protocolo de Control de Transmisión [*Consulta: Marzo 2, 2009*]

<< <http://www.rfc-es.org/rfc/rfc0793-es.txt> >>

Internet Protocol [*Consulta: Marzo 4, 2009*]
<< <http://www.rfc-es.org/rfc/rfc0791-es.txt> >>

El servicio DHCP en GNU/Linux [*Consulta: Marzo 17, 2009*]
<< <http://fferrer.dsic.upv.es/cursos/Linux/Avanzado/HTML/ch06.html> >>

Cliente servidor [*Consulta: Marzo 17, 2009*]
<<http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf >>

Monitoreo de redes de computadoras [*Consulta: Marzo 27, 2009*]
<< http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/solano_v_h/capitulo1.pdf >>

Historia del Sistema GNU/Linux [*Consulta: Marzo 31, 2009*]
<< <http://www.alu.ua.es/p/psp4/historia/index.html> >>

25 razones para migrar a GNU/Linux [*Consulta: Abril 2, 2009*]
<< <http://www.luchoedu.org/descargas/25-razones-para-migrar-a-gnulinux/> >>

RRD World [*Consulta: Abril 16, 2009*]
<< <http://ftp.iasi.roedu.net/mirrors/people.ee.ethz.ch/%257Eoetiker/webtools/rrdtool-1.0.x/rrdworld/index.html> >>

Página web de Cricket [*Consulta: Abril 16, 2009*]
<< <http://cricket.sourceforge.net/> >>

Aprendiendo a programar con Libpcap [*Consulta: Mayo 14, 2009*]
<< <http://www.e-ghost.deusto.es/docs/2005/conferencias/pcap.pdf> >>

Hacking 9 – Programming with Libpcap [*Consulta: Mayo 14, 2009*]
<< <http://recursos.aldabaknocking.com/libpcapHakin9LuisMartinGarcia.pdf> >>

Página web del proyecto IPFM [*Consulta: Mayo 14, 2009*]
<< <http://robert.cheramy.net/ipfm/> >>

Introducción a la librería libpcap [*Consulta: Mayo 18, 2009*]
<< <http://www.escomposlinux.org/lfs-es/blfs-es-6.0/general/libpcap.html> >>

Librería PCAP [*Consulta: Mayo 18, 2009*]
<< http://www.tcpdump.org/pcap3_man.html >>

Creación de gráficas con RRDTools [*Consulta: Mayo 28, 2009*]
<< <http://www.bulma.net/body.phtml?nIdNoticia=1284> >>

Manual básico de CRON [*Consulta: Agosto 25, 2009*]
<< http://www.linuxtotal.com.mx/index.php?cont=info_admon_006 >>

Referencia rápida para CRON [*Consulta: Agosto 25, 2009*]
<< <http://adminschoice.com/crontab-quick-reference> >>

Preguntas Frecuentes de IEEE Standards Association [*Consulta: Septiembre 22, 2009*]
<< <http://standards.ieee.org/faqs/OUI.html#q1> >>

Instalación de Ruby on Rails en Ubuntu [*Consulta: Diciembre 8, 2009*]
<< <http://www.hackido.com/2009/11/install-ruby-on-rails-on-ubuntu-karmic.html> >>

Ejemplo de AutoCompleter para Ruby on Rails [*Consulta: Enero 28, 2010*]
<< <http://share-facts.blogspot.com/2009/02/autocompleter-example-in-ruby-on-rails.html> >>

Página web de Rubygems [*Consulta: Febrero 6, 2010*]
<< <http://rubygems.org/> >>

Video tutorial de la gema Will_Paginate [*Consulta: Febrero 8, 2010*]
<< <http://railscasts.com/episodes/51-will-paginate> >>

Página web oficial de la gema Will_Paginate [*Consulta: Febrero 8, 2010*]
<< https://github.com/mislav/will_paginate/wiki/ >>

Video tutorial de la gema PaperClip [*Consulta: Febrero 15, 2010*]
<< <http://railscasts.com/episodes/134-paperclip> >>

Tutorial de la gema PaperClip [*Consulta: Febrero 15, 2010*]
<< <http://asciicasts.com/episodes/134-paperclip> >>

Video tutorial de la gema Authlogic [*Consulta: Febrero 22, 2010*]
<< <http://railscasts.com/episodes/160-authlogic> >>

Tutorial de la gema Authlogic [*Consulta: Febrero 22, 2010*]
<< <http://asciicasts.com/episodes/160-authlogic> >>

Video tutorial de la gema Declarative_Authorization [*Consulta: Febrero 23, 2010*]
<< <http://railscasts.com/episodes/188-declarative-authorization> >>

Tutorial de la gema Declarative_Authorization [*Consulta: Febrero 23, 2010*]
<< <http://asciicasts.com/episodes/188-declarative-authorization> >>

Manifiesto Agil [*Consulta: Abril 20, 2010*]
<< http://en.wikipedia.org/wiki/Agile_Manifesto >>

¿Cómo usar el Modelo-Vista-Controlador? [*Consulta: Abril 30, 2010*]
<< <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html> >>