



UNIVERSIDAD NACIONAL DE LOJA

**Área de la Energía las Industrias y los Recursos
Naturales no Renovables.**

TÍTULO:

**“ANÁLISIS, DISEÑO E IMPLEMENTACION DE UNA
HERRAMIENTA WEB, EN PLATAFORMA LINUX, CON
ESTÁNDARES DE QoS, PARA EL CONTROL Y MONITOREO
DE TRÁFICO EN LA RED DEL ÁREA DE ENERGÍA,
INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES,
DE LA UNIVERSIDAD NACIONAL DE LOJA”**

*TESIS PREVIA A LA OBTENCIÓN DEL
GRADO DE INGENIERO EN SISTEMAS*

AUTOR

Jorge Cástulo Guerrón Eras.

DIRECTOR DE TESIS

Ing. Wilson Augusto Vélez Ludeña.

Loja – Ecuador

2009

Wilson Augusto Vélez Ludeña

Docente de la carrera de Sistemas, del Área de la Energía las Industrias y los Recursos Naturales no Renovables.

C E R T I F I C A

Que el trabajo de investigación, con el tema: **“ANÁLISIS, DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA WEB, EN PLATAFORMA LINUX, CON ESTÁNDARES DE QoS, PARA EL CONTROL Y MONITOREO DE TRÁFICO EN LA RED DEL ÁREA DE ENERGÍA, INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES, DE LA UNIVERSIDAD NACIONAL DE LOJA”**, presentado por el egresado: Jorge Cástulo Guerrón Eras, ha sido verificado en su totalidad por lo que cumple con los requisitos reglamentarios, autorizando su presentación y defensa correspondiente.

Loja, Noviembre del 2009.

F_____

Ing. Wilson Augusto Vélez Ludeña

DIRECTOR DE TESIS

SESIÓN DE DERECHOS

Yo, Jorge Cástulo Guerrón Eras, autor de éste trabajo de tesis, certifico la propiedad intelectual a favor de la Universidad Nacional de Loja, la cual podrá hacer uso del mismo con la finalidad académica que estime conveniente, así mismo declaro que el software se encuentra bajo la licencia GPL (Generic Public License), la cuál obliga a dar una copia del código fuente acompañando a los programas, además permite hacer las modificaciones necesarias a los mismos, si el programa modificado se distribuye, su código también, así mismo la documentación se encuentra bajo licencia GFDL (Generic Free Document License), que ha sido diseñada para usarla en manuales de software libre, ya que éste necesita documentación libre, todo ésto es con la finalidad de que el material producido en ésta investigación pueda ser distribuido y modificado libremente de acuerdo a los términos especificados por las licencias.

AUTORÍA

Las ideas, versiones, criterios y comentarios vertidos en la presente tesis de Ingeniero en Sistemas, denominada: **“ANÁLISIS, DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA WEB, EN PLATAFORMA LINUX, CON ESTÁNDARES DE QoS, PARA EL CONTROL Y MONITOREO DE TRÁFICO EN LA RED DEL ÁREA DE ENERGÍA, INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES, DE LA UNIVERSIDAD NACIONAL DE LOJA”**, son absoluta autoría y responsabilidad del Autor.

F_____

Jorge Cástulo Guerrón Eras

AGRADECIMIENTO

Al concluir el presente trabajo investigativo, quiero dejar constancia de mi profundo y sincero agradecimiento a la Universidad Nacional de Loja, en especial al Área de la Energía las Industrias y los Recursos naturales no Renovables, a quienes la dirigen y han tenido la certeza de acoger a catedráticos de excelencia, los mismos que al impartir sus enseñanzas, me brindaron en ellas las bases necesarias para mi superación académica, logrando así esta distinguida Institución integrar al medio profesionales de alta competencia.

Además quiero expresar mi gratitud al Ing. Wilson Augusto Vélez Ludeña, distinguido catedrático quien con sabiduría, paciencia y profesionalismo ha dirigido el presente trabajo.

DEDICATORIA

El presente trabajo investigativo deseo dedicarlo a mis padres, Señora Blanca Mariana Eras Gaona y Señor Jorge Gerardo Guerrón Santelí, con su constante amor supieron guiar mis pasos, con sus consejos crearon sabiduría, y su ejemplo me han enseñado que trabajando duro, un hombre puede ser lo que sueña.

A mis hermanos un verdadero ejemplo de superación; Luisita y Jorgito Galarza Guerrón, su inocencia y alegría ilumina a quienes miran; Paulina Pineda Ordóñez en adelante debo compensar su amor, confianza y apoyo, y finalmente a Oswaldo Xavier Huertas Noboa (+), una verdadera amistad no conoce de distancia y las barreras de la vida.

RESUMEN

El presente trabajo lleva como objetivo principal diseñar una herramienta que sea de ayuda a los administradores de una red, de tal modo que al finalizar la interacción estén en la capacidad de mejorar el rendimiento de la conexión a internet, realizando un control en diferentes niveles sean éstos por protocolos, direcciones ip de equipos de la red o subredes, permitiendo el acceso a la herramienta según privilegios de usuario, almacenado la información en una base de datos, y al finalizar realizar un análisis del desempeño de la herramienta, en base a estandares de QoS.

ÍNDICE

Certificado	
Sesión de Derechos	I
Autoría	II
Agradecimiento	III
Dedicatoria	IV
Resumen	V
Introducción	

CAPÍTULO I: CONOCIENDO LINUX

1	SERVICIOS EN LINUX	1
1.1	SEGURIDAD (FIREWALL)	1
1.1.1	Tablas y Cadenas	2
1.1.2	Comandos de Ubicación	4
1.1.3	Comandos de Especificación	4
1.1.4	Estado de la Conexión	5
1.1.5	Objetivos/Criterios/Acciones/Políticas	5
1.1.6	Sintaxis del Comando Iptables	6
1.2	PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS FTP	7
1.2.1	Diagrama de conexiones FTP	7
1.2.2	¿Qué necesitamos para conectarnos a un servidor FTP?	8
1.2.3	¿Cuál es la diferencia entre FTP, SFTP Y FTPS?	9
1.3	Servicio de sistema de domino de nombres DNS	10

1.3.1	Diagrama de servicio DNS	10
1.3.2	¿Cómo traduce el DNS mi dominio?	11
1.3.3	Componentes del DNS	11
1.3.4	Tipos de servidores DNS	12
1.3.5	Tipos de consultas	13
1.4	SERVICIO WEB	14
1.4.1	Diagrama de conexiones al servicio WEB	16
1.4.2	¿Como es el funcionamiento de un servidor WEB?	16
1.5	SERVICIO DE CORREO	17
1.5.1	Diagrama de servicio mail.	19
1.5.2	Correo entrante	19
1.5.3	Correo saliente	20

CAPÍTULO II: CONOCIENDO REDES

2	REDES	21
2.1	INTRODUCCIÓN	21
2.2	ARQUITECTURA DE UNA RED	21
2.3	DIRECCIONAMIENTO IP Y SUBNETTING	22
2.3.1	Introducción	22
2.3.2	Entendiendo las direcciones IP	22
2.3.3	Entendiendo subnetting	24

CAPITULO III: LAS REDES EN LAS EMPRESAS

3	REDES CORPORATIVAS	32
3.1	INTRODUCCIÓN	32
3.2	USO	32
3.3	RIESGOS	33
3.4	ORGANIZACIONES DE ESTÁNDARES DE CABLEADO	33
3.5	CONSUMO DE ANCHO DE BANDA	34
3.5.1	Introducción	34
3.5.2	Gestionando el consumo	34
3.5.3	Monitoreo	36
3.5.3.1	Puertas abiertas	37
3.5.3.2	Vigilando el consumo de la red	38

CAPÍTULO IV: METODOLOGÍA

4	METODOLOGÍA DE LO USADO	40
4.1	MÉTODOS	40
4.2	METODOLOGÍA ICONIX PARA EL DESARROLLO DE SOFTWARE	40
4.3	LENGUAJE	41
4.3.1	Introducción	41
4.3.1.1	¿Qué son los Servlets Java?	41
4.3.1.2	¿Cuáles son las Ventajas de los Servlets sobre el CGI "Tradicional"?	41
4.3.2	Introducción a JSP	43

4.3.2.1	JSP contra ASP	43
4.3.3	ELEMENTOS DE JSP	45
4.3.3.1	Código Java	45
4.3.3.1.1	Expresiones	45
4.3.3.1.2	Scriptlets	46
4.3.3.1.3	Declaraciones	47
4.3.3.2	Directivas	48
4.3.3.2.1	page	48
4.3.3.2.2	include	49
4.3.3.2.3	taglib	50
4.3.3.3	Acciones	50
4.3.3.3.1	Inclusión de páginas	50
4.3.3.3.2	Transferencia de control	51
4.3.4	Objetos implícitos	52
4.3.4.1	Request	52
4.3.4.2	Response	53
4.3.4.3	Out	53
4.3.4.4	Session	53
4.3.4.5	Application	53
4.3.4.6	Config	54
4.3.4.7	PageContext	54
4.3.4.8	Page	55
4.3.5	Comunicación entre formularios HTML y páginas JSP	55
4.3.5.1	Conceptos básicos	55

4.3.5.2	GET y POST	56
4.3.6	Introducción a JSTL	56
4.3.7	CSS y JSP	57
4.3.8	Internacionalización	58
4.4	BASE DE DATOS	59
4.5	FRAMEWORKS	60
4.5.1	Spring MVC Modelo Vista Controlador	61
4.6	IDE	62
4.6.1	Historia	62
4.6.2	¿Como trabaja NetBeans?	63
4.6.3	NetBeans Add-on Packs	64
4.6.4	¿Como se realiza la liberación de versiones de NetBeans?	65
4.6.5	Versiones de NetBeans.	67
4.7	IPROUTE2	68
4.7.1	Nuevos Comandos de iproute2	68
4.7.2	CBQ	70
4.7.3	Parámetros de CBQ	71
4.7.3.1	Parámetro DEVICE	71
4.7.3.2	Parámetro de clase RATE	71
4.7.3.3	Parámetro de clase PRIO	71
4.7.3.4	Parámetro de clase PARENT	72
4.7.3.5	Parámetro de clase LEAF	72
4.7.3.6	Parámetro de clase BOUNDED	73
4.7.3.7	Parámetro de clase ISOLETED	73

4.7.3.8	Parámetros de filtración	74
---------	--------------------------	----

CAPÍTULO V: DESARROLLO DE LA PROPUESTA ALTERNATIVA

5	ANÁLISIS Y DISEÑO DEL PROYECTO DE TESIS	75
5.1	REQUERIMIENTOS DE SOFTWARE	75
5.1.1	REQUERIMIENTOS FUNCIONALES	75
5.1.2	REQUERIMIENTOS NO FUNCIONALES	78
5.2	MODELAMIENTO DE LOS CASOS DE USO	79
5.3	REQUERIMIENTOS	83
5.4	ORGANIZAR LOS CASOS DE USO	85
5.5	DIAGRAMA DE CASOS DE USO	88
5.6	PROTOTIPO DE PANTALLAS	89
5.7	DESCRIPCIÓN DE CASOS DE USO	113
5.8	DIAGRAMAS DE PAQUETES	133
5.8.1	Paquetes Herramienta Web	133
5.8.2	Paquetes Applet Calculadora Ip	133
5.8.3	Paquetes Applet Escaner Puertos	134
5.9	DIAGRAMAS DE ROBUSTEZ	135
5.9.1	DR01: Administrar Configuración por Defecto	135
5.9.2	DR02: Ingresar a la Herramienta	136
5.9.3	DR03: Administrar Dispositivos de Red	137
5.9.4	DR04: Administrar Ip de Equipo	138
5.9.5	DR05: Administrar Ip de Subred	139
5.9.6	DR06: Administrar Script	140
5.9.7	DR07: Calcular Direcciones Ip	142
5.9.8	DR08: Escanear Puertos	143
5.10	DIAGRAMAS DE SECUENCIA	144
5.10.1	DS01: Administrar Configuración por Defecto	144
5.10.2	DS02: Ingresar a la Herramienta	145
5.10.3	DS03: Administrar Dispositivos de Red	146
5.10.4	DS04: Administrar Ip de Equipo	147
5.10.5	DS05: Administrar Ip de Subred	148
5.10.6	DS06: Administrar Script	149
5.10.7	DS07: Calcular Direcciones Ip	150
5.10.8	DS08: Escanear Puertos	151
5.11	DIAGRAMAS DE CLASES	152
5.11.1	DIAGRAMA DE CLASES APLET CALCULADORA IP	156
5.11.2	DIAGRAMA DE CLASES APLET ESCANER	158
5.12	DIAGRAMA DE COMPONENTES	160
5.13	DIAGRAMA DE NAVEGABILIDAD	161
5.14	DIAGRAMA DE CAPAS DE LA APLICACIÓN	162
5.15	DIAGRAMA DE BASE DE DATOS	162
5.16	SCRIPT DE BASE DE DATOS	163
6	CONCLUSIONES	167
7	RECOMENDACIONES	168

8	BIBLIOGRAFÍA	169
9	ANEXOS	172
	ANTEPROYECTO DE TESIS	
	LICENCIA GPL	
	LICENCIA GFDL	

INTRODUCCIÓN

El presente trabajo de tesis lleva como título **“ANÁLISIS, DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA WEB, EN PLATAFORMA LINUX, CON ESTÁNDARES DE QoS, PARA EL CONTROL Y MONITOREO DE TRÁFICO EN LA RED DEL ÁREA DE ENERGÍA, INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES, DE LA UNIVERSIDAD NACIONAL DE LOJA”**, el mismo surgió de necesidad de conocer los métodos que nos ofrece el sistema operativo Linux, para poder realizar el control del ancho de banda con el que cuenta una institución y de como si usamos correctamente éstos métodos podemos mejorar el uso de nuestro acceso a Internet.

En la actualidad tener una conexión a internet en nuestras empresas es de suma importancia puesto que nos permite mantener una comunicación constante con nuestros proveedores, clientes y con nuestros empleados, sin embargo la mayoría de las veces los recursos con los que contamos no son bien utilizados por los nodos que conforman la red.

El sistema operativo Linux nos ofrece varias posibilidades para poder realizar un control de acceso a internet, entre otros usados están squid, éste es un proxy que como conocemos por su funcionalidad nos proporciona un caché de acceso y un control basado en reglas que aunque nos puede ayudar en el control, no nos ayuda en la segmentación de la red.

Otra de las posibilidades es hacer uso de las reglas de control que nos da netfilter y su frontral iptables, para mediante la implementación de un firewall hacer el redireccionamiento de los paquetes que ingresan y salen de la red, esto nos ayudará pero el mantenimiento es muy complicado y además debemos tener un conocimiento en el manejo de las tablas de iptables y en especial de nat, hacer marcado de paquetes y además agregarlas en colas para realizar la segmentación, y si a eso le agregamos el mantener al día los accesos a puertos, red, etc, se nos complica verdad, además de tener un script demasiado largo vamos a terminar algunas veces equivocándonos, y estamos dando a iptables tareas que no le corresponden.

¿Que podemos hacer?, a partir de los kernel 2.2 y 2.4 el sistema operativo Linux incluye un nuevo paquete conocido como iproute2, este nuevo paquete, nuevo en teoría ya es

más una reescritura de algunas de las sentencias antes conocidas, nos ayuda a manejar de un mejor modo el control de nuestro ancho de banda, implementa nuevas sentencias junto a las conocidas y algunos algoritmos que ya estaban disponibles como cbq, la herramienta que es objeto de la presente tesis nos ayuda entregando luego de unas cuantas configuraciones e información que agrega el administrador de la red, los scripts necesarios para realizar la segmentación de nuestro ancho de banda.

Para el desarrollo de la presente tesis se aplicó una metodología acorde a los requerimientos básicos, entre ellos el método científico, deductivo – inductivo y analítico, se usó además la técnica del fichaje para la recopilación de la información necesaria, esto junto a la metodología ICONIX, para el desarrollo de software y la implementación de frameworks como spring e hibernate, ha permitido desarrollar la herramienta, según los requerimientos obtenidos.

CAPÍTULO I

CONOCIENDO LINUX

1 SERVICIOS EN LINUX

En el presente capítulo vamos a hablar de los mas conocidos y usados servicios que podemos toparnos cuando hacemos uso de un servidor en linux, o cuando tenemos una conexión a internet y la usamos para trabajar con los diferentes servicios que a continuación nombramos, no vamos a topar temas de instalación pero si de conocimiento general, que nos pueden ayudar en algún momento.

Cada servicio implementado ya tiene asignado un puerto mediante el cual se puede comunicar, la asignación de los puertos que debe utilizar cada servicio son regulados por la IANA.

1.1 SEGURIDAD (FIREWALL)

En el presente apartado vamos a introducirnos en la implementación de seguridades dentro del sistema operativo linux, el modo mas popular de controlar el acceso a nuestro sistema operativo, es mediante la ejecución de un script que agrega las reglas de control de un Firewall.

Ahora bien denominamos **Firewall** a un hardware específico con un sistema operativo o una IOS que filtra el tráfico de la red (TCP/UDP/ICMP/./IP) y decide si un paquete pasa, se modifica, se convierte o se descarta. Para que un firewall pueda funcionar correctamente debe tener al menos dos tarjetas de red.

Nota: en el presente no se indica como realizar la instalación del firewall, asumo que el usuario del presente documento, ya tiene instalados los paquetes necesarios.

- Netfilter: es un firewall que forma parte del kernel de Linux. Examina las cabeceras de cada paquete que llega a las interfaces de red y decide qué hacer con él:

1. Por un lado, decide si modificar o no el paquete alterando su origen y/o destino, según las reglas NAT (*Network Address Translation*). Esto es lo que hacemos, por ejemplo, al redirigir puertos en el router.

2. Por otro lado, decide si lo deja pasar, lo rechaza o lo ignora.

- Iptables: es un frontal para Netfilter (reemplazó a **IPchains**). El comando iptables nos permite definir las reglas que aplica Netfilter.

Iptables es uno de los comandos mas complejos de entender por lo tanto explicaré como se estructura y como debemos implementarlo, lo mejor y que más llama la atención en iptables es la flexibilidad que se permite para que cada persona pueda escribir sus soluciones o reglas, sin interferir con el netfilter o el código de iptables.

Para poder empezar a utilizar iptables es necesario iniciar entendiendo sus partes.

1.1.1 Tablas y Cadenas

Cuando un paquete ingresa estos atraviesan diferentes tablas y cadenas en un orden, es valioso saber esto para cuando deseamos escribir nuestras propias reglas. Así como los momentos en que ciertos componentes, también dependientes del núcleo, ingresa (decisiones de enrutado).

Cuando un paquete entra al firewall (cortafuegos), este empieza a recorrer una serie de etapas antes de ser enviado a la aplicación adecuada (localmente), reenviada hacia otro host, o cualquier otra operación.

En iptables existen tres tablas ya definidas, las cuales contienen cadenas predeterminadas y son:

- **Filter** (Tabla de filtros) contiene las cadenas que pueden tener reglas de filtrado de los paquetes como:
 - INPUT (entrada).-filtra todos los paquetes entrantes.
 - OUTPUT (salida).- filtra los paquetes salientes.
 - FORWARD (redirección).- filtra los paquetes que están redirigidos a otras interfaces SOLO atraviesan la tabla FORWARD los paquetes que llegan desde la red, teniendo como destino el firewall.
- **Nat** (Tabla de traducción de direcciones de red) contiene las cadenas de reescritura de direcciones o de puertos de los paquetes como:
 - PREROUTING (preruteo) los paquetes entrantes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.

- **POSTROUTING** (postruteo) los paquetes salientes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.
- **OUTPUT** (salida) filtra los paquetes salientes.

Nota: MASQUERADE es un caso especial de SNAT, (Traducción del Direcciones en el Origen)

- *REDIRECT es un caso especial de DNAT (Traducción de Direcciones en el Destino).*
- **Mangle** (Tabla de destrozo) esta tabla contiene todas las cadenas predefinidas y todos los paquetes pasan por ella. Esta ajusta algunas opciones de los paquetes.
- **PREROUTING** (preruteo) los paquetes entrantes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.
- **INPUT** (entrada) filtra todos los paquetes entrantes.
- **FORWARD** (redirección) filtra los paquetes que atraviesan.
- **OUTPUT** (salida) filtra los paquetes salientes.
- **POSTROUTING** (postruteo) los paquetes salientes pasan a través de esta cadena antes de que sea consultada la tabla de ruteo.

1.1.2 Comandos de Ubicación

- **-t** Indica en que tabla ubicar la regla, si se omite este comando se agrega la regla a la tabla Filter Table.
- **-A** Agrega una regla al final de la lista de reglas de la cadena.
- **-I** Inserta una regla dentro de la cadena especificada en la posición.
- **-D** Borra la regla en la posición especificada
- **-R** Reemplaza una regla en la posición indicada por otra.

1.1.3 Comandos de Especificación

Tabla N° 1: Indica las condiciones que se pueden usar.

-s	Especifica el origen sea dirección ip / máscara de origen
-d	Especifica una dirección de destino dirección ip / máscara de destino
-p	Especifica un protocolo (tcp/udp/icmp)
-i	Especifica una interfaz de entrada (eth0/eth1/ppp12/..)
-o	Especifica una interfaz de salida (eth0/eth1/ppp12/..)
-j	Especifica la acción a ejecutar sobre el paquete
--sport	Puerto de origen
--dport	Puerto de destino
--mac-source	Dirección (ethernet) MAC
-m state --state	Estado de la conexión
-m multiport --dports port,port[,...]	Para especificar varios puertos de destino.

1.1.4 Estado de la Conexión

Los posibles estados de la conexión, también se encuentran incluidos dentro de los comandos de especificación, (condiciones) son:

- **NEW:** conexiones nuevas desde o hacia el cortafuegos.
- **ESTABLISHED:** conexiones ya establecidas (intercambiaron paquetes anteriormente).

- **RELATED:** conexión nueva relacionada con otra ya establecida. Por ejemplo una conexión que necesita abrir un socket pero que guarda relación con otra conexión.

Ej:FTP

- **INVALID:** inválidas por problemas de comunicación, cabeceras incorrectas.

1.1.5 Objetivos/Criterios/Acciones/Políticas

- **DROP** Elimina el paquete
- **DENY** Deniega el paso al paquete
- **REJECT** Rechaza el paquete
- **ACCEPT** Acepta el paquete
- **QUEUE** Encola el paquete
- **RETURN** Termina la cadena actual y retorna a la inicial.

Como escogemos la política que va manejar nuestro firewall? Bueno la política accept como lo podemos deducir mantiene abierto todos los puertos en nuestra máquina y simplemente nosotros vamos cerrando lo que no necesitamos, por otro lado la política drop cierra todos los puertos en nuestro equipo y nosotros tenemos que ir abriendo lo que necesitamos explicitar cada conexión permitida en los ambos sentidos (salida y entrada), además debemos estar consientes que vamos a mantener abierto y vamos a bloquear, es mas complicado de mantener pero nos da mayor seguridad.

Que debemos tener presente, antes de empezar a escribir nuestras reglas iptables:

1. Saber sobre que tabla queremos operar.
2. Que cadena de esa tabla queremos tratar
3. Que tipo de operación vamos a realizar.
4. Que comparaciones queremos utilizar.
5. Que objetivos (política) queremos aplicar.

1.1.6 Sintaxis del Comando Iptables

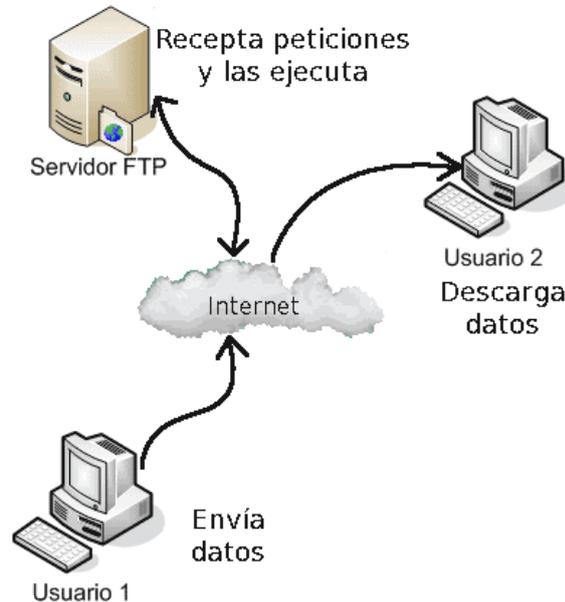
Las siguientes líneas con un ejemplo de como se debe ubicar los diferentes comandos que serán reconocidos por iptables:

```
iptables [-t|--table tabla] -command [cadena] [-i interfaz] [-p protocolo] [-s dirección [port[:puerto]]] [-d dirección [port[:puerto]]] -j política.
```

iptables [tabla] [cadena] [condiciones] [qué_hacer]

1.2 PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS FTP

Cuando deseamos enviar nuestros datos a un nodo alojado en un punto distante a nuestra infraestructura física, necesitamos muchas de las veces hacer uso del servicio FTP, conocido así por sus siglas en inglés (File Transfer Protocol), este servicio nos permite conectar a dos nodos distantes, manteniendo una comunicación constante, para poder hacer el envío de datos entre los dos nodos.



1.2.1 Diagrama de conexiones FTP

Figura N° 1: Esquema básico de una sesión FTP

1.2.2 En que se utilizan los servidores FTP

El uso que le podemos dar a un servidor FTP es variado, el ejemplo más común es cuando los webmaster o encargado de una página web, usan los servidores FTP para subir sus páginas web y su contenido a Internet.

Además podemos tener configurado un servidor para compartir los archivos en nuestra empresa, y permitir el acceso a los miembros de nuestra red, no es necesario que tengamos una conexión a internet para hacer esto ya que este servicio bien configurado es de mucha ayuda, ya que igualmente lo podemos utilizar como un backup de los archivos importantes que pueda tener una empresa.

1.2.3 ¿Qué necesitamos para conectarnos a un servidor FTP?

Si ya contamos con el acceso a un servidor FTP, necesitamos conectarnos mediante *Cientes FTP*, que son los programas que nos permiten establecer la conexión al

servidor, existen una gran cantidad de clientes ftp, entre los mas conocidos y utilizados están:

- FileZilla; es un cliente gratuito, libre distribuido bajo la licencia (GNU). Soporta los protocolos FTP, SFTP y FTPS, inicialmente diseñado para Windows, gracias al uso de wxWidgets es multiplataforma desde la versión 3.0.3

NOTA: FileZilla Server es el servidor FTP desarrollado como parte del mismo proyecto, éste funciona bajo Windows NT, 2000 y XP.

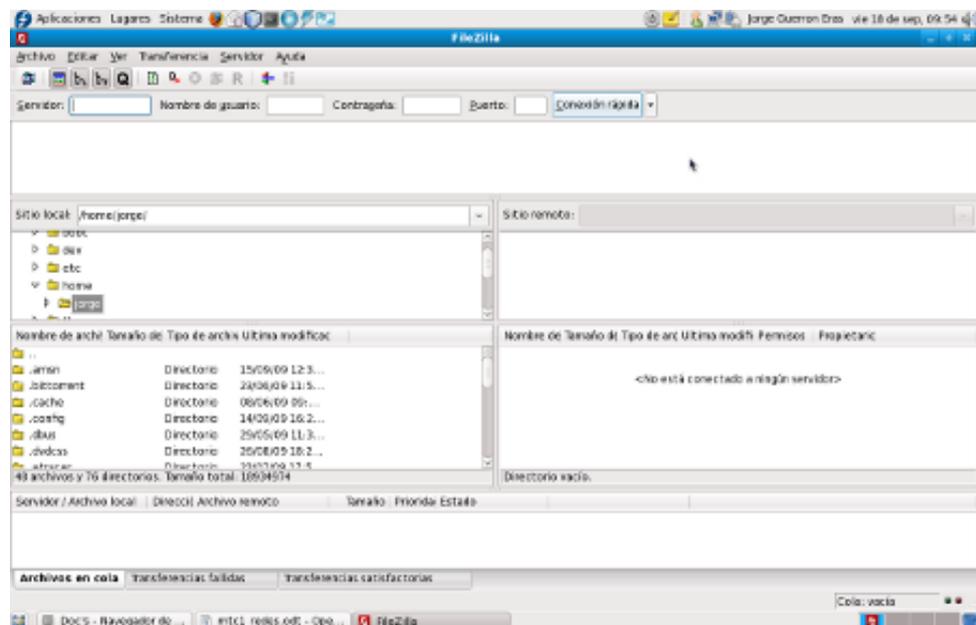


Figura N° 2: Captura de la vista principal de FileZilla.

- gFTP; éste es uno de los clientes de transferencia FTP, más versátiles ya que nos permite realizar transferencias tanto desde gftp-gtk, una GUI o desde gftp-text, cliente de línea de comandos, éste cliente al igual que FileZilla, soporta los múltiples protocolos asociados a FTP.

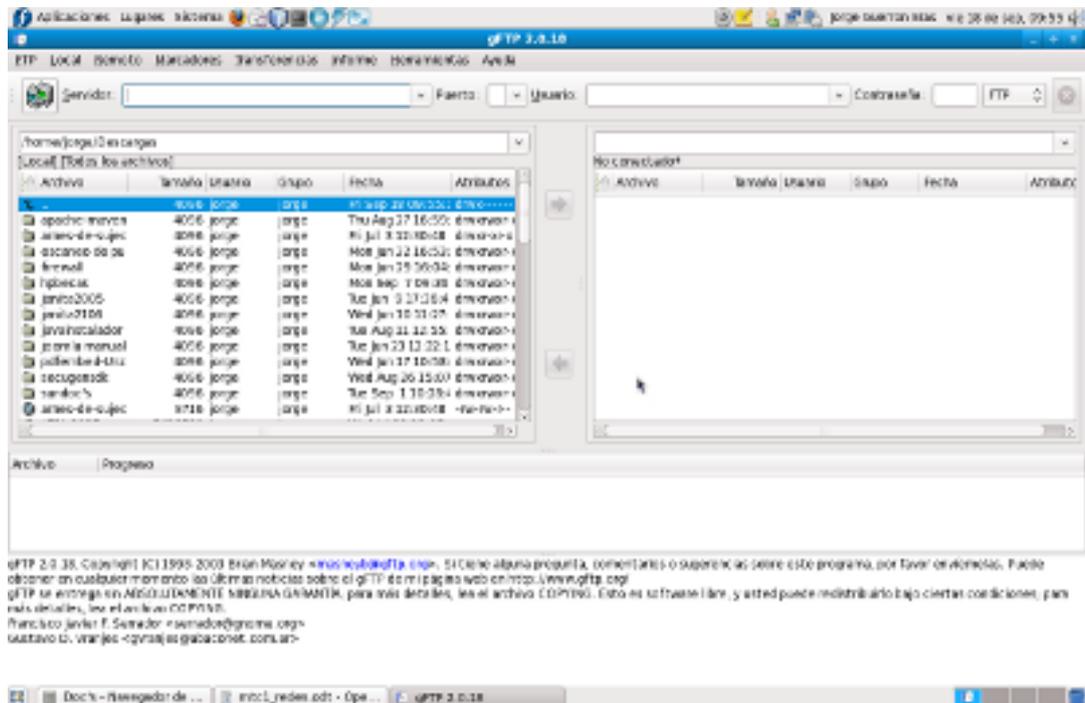


Figura N° 3: Captura de la vista principal de gftp.

- Para sistemas operativos windows tenemos varias alternativas de clientes FTP, entre las mas conocidas son wsFTP, FTP Voyager, SmartFTP.

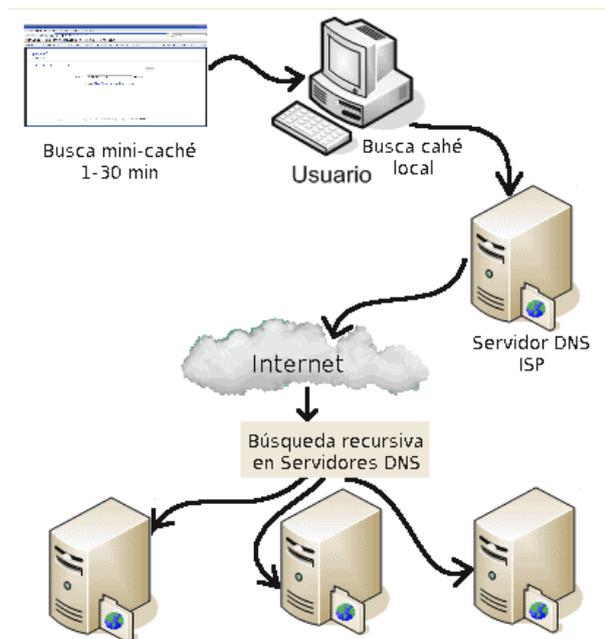
1.2.4 ¿Cuál es la diferencia entre FTP, SFTP Y FTPS?

FTPS es una variante de FTP basada en la protección SSL por sus siglas en inglés (Secure Sockets Layer) o Protocolo de Capa de Conexión Segura, éste protocolo encripta las conexiones de control, es decir que transmite comandos FTP con sus parámetros como contraseñas con texto cifrado, pero no cifra ninguno de los ficheros, en los últimos tiempos también se ha estado trabajando en TLS (Transport Layer Security) o Seguridad de la Capa de Transporte, y está pensado para que sea el sucesor de SSL, y al igual que SSL son protocolos criptográficos que proporcionan comunicaciones seguras por una red.

Las transferencias seguras con SFTP, son bastantes útiles si estamos subiendo datos a nuestro sitio web, ya que ha diferencia de FTP y FTPS, SFTP cifra la contraseña y los datos, sólo únicamente lo podemos hacer si el proveedor tiene acceso SSH, ya que usa éstas funciones para la realizar las conexiones.

1.3 Servicio de sistema de domino de nombres DNS

El servicio de DNS nos permite traducir el nombre de dominio a una dirección ip, y de éste modo ser localizados desde cualquier lugar del mundo únicamente mediante nuestro nombre de domino.



1.3.1 Diagrama de servicio DNS

Figura N° 4: Esquema básico de una búsqueda DNS

1.3.2 ¿Cómo traduce el DNS mi dominio?

El servidor DNS mantiene una base de datos distribuida, con información que se usa para traducir los nombres de dominio, fáciles de recordar y usar por las personas, en números de protocolo de Internet (IP) que es la forma en la que las máquinas pueden encontrarse una a otra en Internet.

El trabajo que realizan los servidores es transparente al usuario general, ya que la traducción de los nombres se realiza por las aplicaciones del cliente (por ejemplo, navegadores, clientes de correo y otras aplicaciones que usan Internet), de este modo facilitan el recordar un nombre a recordar una dirección IP, otra de las facilidades que tenemos al utilizar un servidor DNS, es que una dirección numérica puede cambiar por cualquier circunstancia, sin que ésto afecte el nombre seleccionado, los siguientes pasos ilustran como se realiza la resolución de nombres.

1.3.3 Componentes del DNS

Para poder funcionar correctamente, el DNS utiliza tres componentes principales:

- **Cientes DNS** (*resolvers*). Los clientes DNS envían las peticiones de resolución de nombres a un servidor DNS. Las peticiones de nombres son preguntas de la forma: ¿Qué dirección IP le corresponde al nombre *www.midominio.com.ec*?
- **Servidores DNS** (*name servers*). Los servidores DNS contestan a las peticiones de los clientes consultando su base de datos. Si no disponen de la dirección solicitada pueden reenviar la petición a otro servidor.
- **Espacio de nombres de dominio** (*domain name space*). Se trata de una base de datos distribuida entre distintos servidores.

1.1.1 Tipos de servidores DNS

1.1.2 Servidores primarios (*primary name servers*). Estos servidores almacenan la información de su zona en una base de datos local. Son los responsables de mantener la información actualizada y cualquier cambio debe ser notificado a este servidor.

1.1.3 Servidores secundarios (*secondary name servers*). Son aquellos que obtienen los datos de su zona desde otro servidor que tenga autoridad para esa zona. El proceso de copia de la información se denomina *transferencia de zona*.

1.1.4 Servidores maestros (*master name servers*). Los servidores maestros son los que transfieren las zonas a los servidores secundarios. Cuando un servidor secundario arranca busca un servidor maestro y realiza la transferencia de zona. Un servidor maestro para una zona puede ser a la vez un servidor primario o secundario de esa zona. Estos servidores

extraen la información desde el servidor primario de la zona. Así se evita que los servidores secundarios sobrecargen al servidor primario con transferencias de zonas.

1.1.5 Servidores locales (*caching-only servers*). Los servidores locales no tienen autoridad sobre ningún dominio: se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Estos servidores mantienen una *memoria caché* con las últimas preguntas contestadas. Cada vez que un cliente DNS le formula una pregunta, primero consulta en su memoria caché. Si encuentra la dirección IP solicitada, se la devuelve al cliente; si no, consulta a otros servidores, apunta la respuesta en su memoria caché y le comunica la respuesta al cliente.

1.3.4 Tipos de consultas

Existen tres tipos de consultas que un cliente puede hacer a un servidor DNS:

- **Recursiva:** las consultas recursivas consisten en la mejor respuesta que el servidor de nombres pueda dar. El servidor de nombres consulta sus datos locales (incluyendo su caché) buscando los datos solicitados.
- **Iterativa:** las consultas iterativas, o resolución iterativa el servidor no tiene la información en sus datos locales, por lo que busca un servidor raíz y repite el mismo proceso básico (consultar a un servidor remoto y seguir a la siguiente referencia) hasta que obtiene la respuesta a la pregunta.

El proceso de resolución normal se da de la siguiente manera:

- El servidor A recibe una consulta recursiva desde el cliente DNS.
- El servidor A envía una consulta iterativa a B.
- El servidor B refiere a A otro servidor de nombres, incluyendo a C.
- El servidor A envía una consulta iterativa a C.
- El servidor C refiere a A otro servidor de nombres, incluyendo a D.
- El servidor A envía una consulta iterativa a D.
- El servidor D responde.
- El servidor A regresa la respuesta al resolver.

- El resolver entrega la resolución al programa que solicitó la información.
- **Inversas:** las consultas inversas son cuando se envía una petición para conocer el nombre de dominio dada una dirección IP. Para evitar una búsqueda exhaustiva por todo el espacio de nombres de dominio, se ha creado un dominio especial llamado *in-addr.arpa*. Cuando un cliente DNS desea conocer el nombre de dominio asociado a la dirección IP *192.168.0.1*, formula una pregunta inversa a *1.0.168.192.in-addr.arpa*. La inversión de los bytes es necesaria debido a que los nombres de dominio son más genéricos por la derecha, al contrario que ocurre con las direcciones.

1.4 SERVICIO WEB

Imaginemos que el internet es como una pequeña ciudad, los sitios serían como locales dentro de un centro comercial, cuando un usuario visita la ciudad busca entre los centros comerciales sus necesidades, y son los encargados de los locales quienes tienen la tarea de decir lo que tienen y ofrecen a los visitantes.

Un servidor web vendría a ser nuestro centro comercial, éste servidor web es un programa que se ejecuta continuamente en un ordenador, y que se mantiene a la espera de peticiones por parte de sus clientes nosotros los conocemos como *navegador web*, cuando el servidor responde al cliente enviando el código de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla, ésto mediante lo que conocemos como una página web la cual será exhibida en el navegador. Éste programa implementa el protocolo HTTP por sus siglas en inglés (*HyperText Transfer Protocol*), y se diseñó para transferir lo que llamamos hipertextos, páginas web o páginas HTML (*HyperText Markup Language*), el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar el tipo de letra, los colores, la disposición de los textos, imágenes y demás objetos de la página.

Debemos tener en claro que cuando hacemos peticiones al servidor web, las aplicaciones se pueden ejecutar de los 2 lados es decir del lado del servidor y del lado de las aplicaciones.

- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.
- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo java o javascript; el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

Algunos servidores web importantes son:

- Apache, Tomcat, Glassfish, Jboss.

Si deseamos hacer pruebas podemos configurar un servidor web nosotros mismo esto nos permitirá, hacer test en nuestra red antes de subirlo al hosting, uno de los más usados es:

- Xampp.

Servidores como Apache nos permiten, por su arquitectura modular, tener un servidor con las últimas actualizaciones en cuanto a los protocolos, además nos permiten responder a diferentes lenguajes de programación, lo único que debemos hacer es instalar los módulos dependiendo del lenguaje que queremos tener soporte.

1.4.1 Diagrama de conexiones al servicio WEB

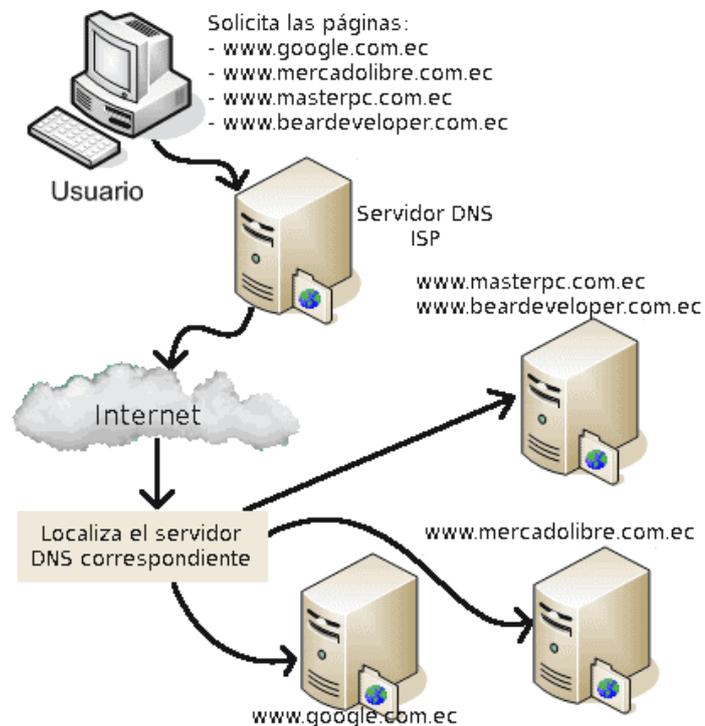


Figura N° 5: Esquema de conexiones a un servidor WEB

1.4.2 ¿Como es el funcionamiento de un servidor WEB?

Ahora se explicará el funcionamiento básico del servidor web apache y como éste responde a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada). Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

- Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
- Recibe una petición.
- Busca el recurso.
- Envía el recurso utilizando la misma conexión por la que recibió petición.
- Vuelve al segundo punto.

A partir de este esquema básico se han diseñado y desarrollado todos los servidores HTTP que existen, variando sólo el tipo de peticiones a las que responden páginas estáticas, CGIs, Servlets, etc., que pueden atender, en función de que sean o no sean multi-proceso o multi-hilo, etc., ahora es posible gracias a módulos combinar varios servidores web a fin de que sea uno el que recibe la petición y la responde y otro el que la procesa, son dos de los posibles casos, que se pueden presentar un ejemplo es cuando necesitamos que sea nuestro servidor web primario (Apache), el que reciba las peticiones de servidores secundarios (Tomcat, Glassfish, etc), de tal manera que no es el primario el que resuelve como servir las páginas, en sí este únicamente las redirecciona al servidor destinado por el administrador (lo de servidor primario y secundario es una ejemplificación se puede o no dar el caso).

1.5 SERVICIO DE CORREO

El funcionamiento que le damos al correo electrónico es similar al que tenemos cuando hacemos uso de un correo privado o postal, básicamente cuando usamos el correo postal indicamos el destinatario de la carta, para poder saber a quien debe llegar nuestro mensaje, con la innovación tecnológica esto se redujo a enviar datos electrónicos desde diferentes partes del mundo y de éste modo poder mantener una comunicación mas ágil, es así que podemos definir al servicio de correo electrónico como la aplicación informática que nos permite enviar mensajes a través de nuestras redes computacionales.

Para poder hacer uso del correo electrónico debemos hacer uso de dos protocolos diferentes, uno lo usaremos en el envío y otro en la recepción, aunque nuestros mensajes se receptorán en el mismo servidor de correo, el envío y la recepción se realizan gracias a la implementación de diferentes protocolos.

Para obtener los mensajes del servidor, los usuarios se sirven de clientes, es decir, programas que implementan un protocolo de envío de correo, éstos clientes pueden ser ejecutados directamente desde los equipos de los usuarios, entre los clientes mas conocidos están Thunderbird, Evolution, Microsoft Outlook, a su vez es posible que el cliente no esté instalado en el equipo del usuario y que el cliente deba acceder al correo

vía web, como es el caso de Hotmail, Gmail, Yahoo Mail, OpenWebmail, SquirrelMail, RoundCube WebMail, Horde.

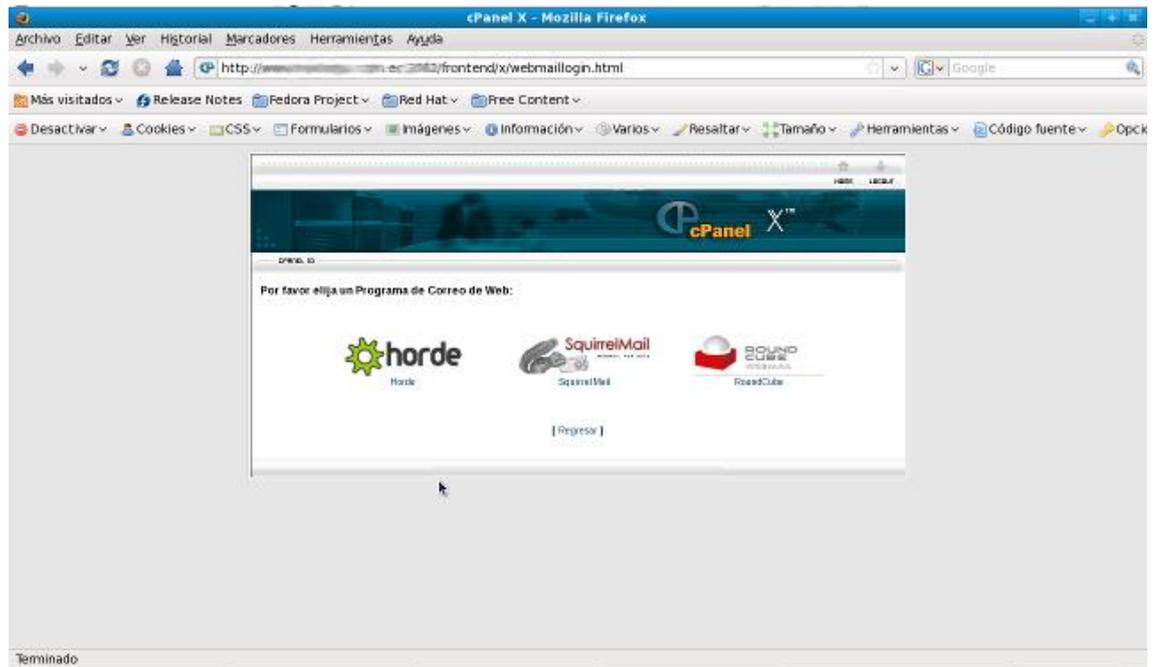
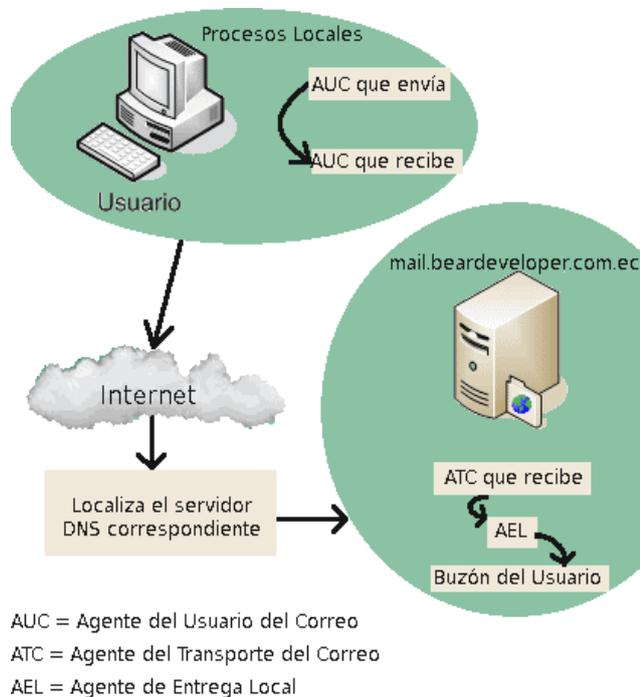


Figura N° 6: Opciones para acceder a un servidor mail via Web



1.5.1 Diagrama de servicio mail.

Figura N° 7: Esquema básico del proceso de envío de correo

1.5.2 Correo entrante

POP: Post Office Protocol: Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario, los servidores Post Office Protocol (POP) almacenan los mensajes hasta el inicio de sesión en el sistema, al iniciar la sesión en un servidor POP, se descargan todos los mensajes en el sistema local y se eliminan del servidor. De esta forma se trabaja de forma en ql equipo con los mensajes. Por lo general, un servidor POP posee una conexión permanente con Internet y los clientes se conectan de forma ocasional.

IMAP: Internet Message Access Protocol: Su finalidad es la misma que la de POP, pero el funcionamiento y las funcionalidades que ofrecen son diferentes, ya que éste protocolo permite recibir y almacenar los mensajes, pero al iniciar la sesión en un

servidor IMAP se puede visualizar la información de cabecera de los mensajes. Se pueden abrir los mensajes que se deseen leer y también se puede crear y trabajar con carpetas creadas en el servidor, ésta opción es importante si desea acceder al correo electrónico desde varios tipos de clientes, ya que los mensajes no son eliminados del servidor.

1.5.3 Correo saliente

SMTP: Simple Mail Transfer Protocol: Es el protocolo que se utiliza para el envío de correo ya sea desde un servidor de correo a otro, o bien, desde un cliente de correo al servidor. El servidor se encarga de enviar los mensajes, éste protocolo es uno de los mas comunes en todos los sistemas operativos ya que a SMTP realmente no le concierne como viaja el correo, solo en como entregar el mensaje a su destino , es ésta flexibilidad la que le permite trabajar bajo diferentes ambientes de red.

Sendmail/Postfix: Sendmail y Postfix son en realidad servicios en los sistemas operativos mas comunmente linux, cuando utilizamos el programa sendmail para enviar el correo, es nuestro sistema el que se encarga de establecer la conexión y el mismo sistema el encargado de enviar, los mensajes.

En cualquier caso, los protocolos SMTP/POP/IMAP son inseguros en cuanto los mensajes viajan en claro por la red, es decir, es fácil obtener nuestros mensajes y contraseñas. Para ello se suele añadir una capa SSL, que como ya lo explicamos en los temas anteriores nos permiten implementar un método de cifrado tanto en el servidor como el cliente.

CAPÍTULO II

CONOCIENDO REDES

2 REDES

2.1 INTRODUCCIÓN

Desde que el ser humano vio la necesidad de comunicarse, éste inventó diversas maneras de romper las barreras que las distancias y la geografía le impusieron para poder estar al tanto de los acontecimientos que se daban en su entorno, es así que con el nacimiento de la Red Telegráfica inventada por Alexander Graham Bell, se da el punto inicial para las redes de comunicaciones, y a través de los años ésta fue evolucionando.

En la actualidad contamos con diversos medios que nos permiten pasar todo tipo de datos que necesitemos y saber en segundos lo que está sucediendo al otro lado del mundo.

2.2 ARQUITECTURA DE UNA RED

Una red de computadoras, también llamada red de ordenadores o red informática, es un conjunto de equipos conocidos como nodos, éstos pueden ser computadoras y/o dispositivos conectados por medio de aire, cables, señales, ondas o cualquier otro método de transporte de datos, que comparten información (archivos), recursos (cd-rom, impresoras, etc.) y servicios (acceso a internet, mail, chat), etc.

Para simplificar la comunicación entre programas (aplicaciones) de distintos equipos, se definió el Modelo OSI -es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones-, éste fue definido por ISO **I**nternational **O**rganization for **S**tandardization, (*Organización Internacional para la Estandarización*), y especifica 7 distintas capas de abstracción. Con ello, cada capa desarrolla una función específica con un alcance definido.



Figura N° 8: Modelo de la pila OSI

2.3 DIRECCIONAMIENTO IP Y SUBNETTING

2.3.1 Introducción

Esta sección nos va a dar una información básica que se necesitará para configurar, la cantidad de direcciones que se disponen en nuestra red, las IP's que se desaprovechan, hasta cuantas subredes disponemos para trabajar.

2.3.2 Entendiendo las direcciones IP

Una dirección IP consiste en cuatro octetos (1 octeto = 8 bits = 1 byte), 32 bits o 4 bytes. El valor en cada octeto varía en el rango de 0 a 255 en numeración decimal, o de 0000000 a 11111111 en numeración binaria. Es muy recomendable que tener en claro como es el sistema de numeración binaria, dado que se lo utiliza mucho y le será de

mucha ayuda estar familiarizado con él. A continuación un recordatorio de cómo se convierte un octeto binario a decimal:

$$\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} = (128+64+32+16+8+4+2+1=255)$$

A continuación ejemplo de la conversión de un octeto:

$$\begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} = (0+64+0+0+0+0+0+1=65)$$

Y lo que sigue es la representación binaria y decimal de una dirección IP :

10.	1.	23.	19	(decimal)
00001010.	00000001.	00010111.	00010011	(binario)

Los octetos están divididos para proveer un esquema de direccionamiento que sea flexible y se pueda ajustar tanto a grandes como pequeñas redes. Existen 5 clases de redes IP diferentes, de la A a la E, pero únicamente hablaremos de las clases A a la C, dado que las clases D y E son reservadas para multicast y con fines experimentales. Para determinar el tipo de red a la que pertenecemos basta con observar el primer octeto:

Clase A: 1 – 126

Clase B: 128 – 191

Clase C: 192 – 223

Tabla N° 2: Clases de Direcciones IP					
Clase	Primer byte	Identificación de red	Identificación de hosts	Número de redes	Número de hosts
A	1 .. 126	1 byte	3 byte	126	16.387.064
B	128 .. 191	2 byte	2 byte	16.256	64.516
C	192 .. 223	3 byte	1 byte	2.064.512	254

2.3.3 Entendiendo subnetting

La división en subredes o subnetting nos da la posibilidad de crear múltiples redes, partiendo de una sola dirección de red clase A, B o C.

Una máscara de subred está definida para cada dirección IP. La máscara de subred identifica que segmento de los cuatro octetos de la IP, está determinando red y subred y que porción está determinando hosts, si vamos a dividir en subredes, debemos usar las mascararas de subred por default para cada clase:

Clase A: 255.0.0.0

Clase B: 255.255.0.0

Clase C: 255.255.255.0

Ahora si nos surge la duda ¿Puedo crear una subred con un número mínimo de hosts por IP?, la respuesta es afirmativa, podemos crear subredes con un número determinado de host dentro de nuestra subred, calculando la sub máscara dependiendo del número específico de hosts, que vamos a habilitar en nuestra red.

Entonces hacemos lo siguiente:

Si queremos crear una subred con un mínimo de 10 hosts, lo que debemos hacer es crear la sub máscara que nos permita controlarlo:

Elevamos 2^x donde x es igual 0...n,

a.) h = cantidad mínima requerida, en este ejemplo 10 host.

$2^0 = 1$	$1 < 10$	No es suficiente
$2^1 = 2$	$2 < 10$	No es suficiente
$2^2 = 4$	$4 < 10$	No es suficiente
$2^3 = 8$	$8 < 10$	No es suficiente
$2^4 = 16$	$16 > 10$	Bien

Ahora colocamos los últimos 4 bits de la sub máscara en 0.

255.	255.	255.	240	(decimal)
11111111.	11111111.	11111111.	11110000	(binario)

Con esta sub máscara se tenemos un mínimo de 10 hosts en la subred, de este modo controlamos las direcciones IP que pueden funcionar en la red.

¿Que otros datos son necesarios conocer respecto a los datos de la máscara de red o netmask, y las direcciones IP?

CIDR: Classless Inter-Domain Routing (Encaminamiento Inter-Dominios sin Clases), conocido como **Resumen de Rutas**, ya que es la simplificación de varias direcciones de redes o subredes en una sola dirección IP o un patrón que cubra todo ese esquema de direccionamiento IP.

Wildcard: Una máscara **wildcard** sencillamente una agrupación de 32 bits dividida en octetos, similar a una máscara de subred, pero solo la apariencia, es la única similitud, ya que la máscara wildcard se emplean junto a un valor IP, para seleccionar direcciones IP, esto se debe a que la máscara wildcard indica con sus ceros y unos cuáles de los bits se tomarán en cuenta para compararse o no, un cero indica que el bit debe ser comparado, por su lado un uno indica que el bit debe dejarse a un lado.

Broadcast: es un modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

Ejemplos:

Cálculo del CIDR: 172.16.3.0/24

1	0	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	
255								255								255								0										

192.168.0.0/28

11000000.10101000.00000000.00000000

1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
255								255								255								240									

Retornando al ejemplo anterior en el que debíamos crear una red con un mínimo de 10 host nos daría lo siguiente:

Netmask: 255255255240 11111111.11111111.11111111.1111 0000

Wildcard: 0.0.0.15 00000000.00000000.00000000.0000 1111

Hosts/Net: 14 Class C.

CIDR: 192.168.0.0/28 11111111.11111111.11111111.1111 0000
 HostMin: 192.168.0.1 11000000.10101000.00000000.0000 0001
 HostMax: 192.168.0.14 11000000.10101000.00000000.0000 1110
 Broadcast: 192.168.0.15 11000000.10101000.00000000.0000 1111

¿Si deseamos crear una sub máscara con mínimo 70 hosts por IP?

1. Elevamos números a la potencia de dos, donde la cantidad mínima requerida es 70.

$2^6 = 64$	$64 < 70$	No es suficiente
$2^7 = 128$	$128 > 70$	Bien

2. Ahora coloca los últimos 7 bits de la submáscara en 0.

11111111.	11111111.	11111111.	10000000
255.	255.	255.	128

Cálculo del Broadcast dada la subred

Para la subred 172.16.64.0/20, determinar el Broadcast?

1. CIDR= /20 significa 255.255.240.0

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
255								255								240								0							

Como se sabe, una dirección IP esta conformada por 4 octetos de 8 bits cada uno, para un total de 32 bits.

2. La sub máscara está en el tercer octeto el cuál es: 11110000

Cada bit es un múltiplo de dos, por lo que se tiene:

1	1	1	1	0	0	0	0
128	64	32	16	0	0	0	0

Con lo cuál el último de los 1 corresponde al 16, donde se establece los saltos de la subred, de 16 en 16.

3. Ahora se escribe las distintas subredes partiendo de la que nos dan, con saltos de 16 en 16.

Subred1:172.16.64.0 - 172.16.79.255 -->Broadcast

Subred2:172.16.80.0 - 172.16.95.255 -->Broadcast

Subred3:172.16.96.0-172.16.111.255 -->Broadcast

Subred4:172.16.112.0-172.16.127.255 ->Broadcast

Ya que la siguiente SUBRED a la que nos dan es la 172.16.80.0 el Broadcast es el número anterior a ella, donde todos los Bits de hosts están en 1, o sea 172.16.79.255 donde en binario es: 1 0 1 0 1 1 0 0 . 0 0 0 1 0 0 0 0 . 0 1 0 0 1 1 1 1 . 1 1 1 1 1 1 1 1 1

Dada la siguiente subred 172.16.64.0 /26, Cuál es la dirección IP de broadcast?

Se aplica el mismo procedimiento donde /26 significa: 255.255.255.192

Donde 192 es el número binario

11000000 --> 128 64 32 16 8 4 2 1

El último 1 se encuentra en el valor 64, donde se establece el número de saltos de 64 en 64.

Subred 1 : 172.16.64.0 - 172.16.64.63 -->Broadcast

Subred 2 : 172.16.64.64 - 172.16.64.127 -->Broadcast

Subred 3 : 172.16.64.128 - 172.16.64.191 -->Broadcast

Subred 4 : 172.16.64.192 - 172.16.64.257 -->Broadcast

Cálculo de la dirección IP del primero y último valor de host para una subred dada

Si se tiene la dirección IP 192.168.20.32 /27

La primera dirección esta reservada por defecto por el Gateway.

Entonces cuál es la primera dirección de host?

1) Primer paso:

/27 significa 255.255.255.224

2) Segundo paso:

224 significa 11100000 -->128 64 32 16 8 4 2 1

Por lo cuál el último 1 se encuentra en el valor 32, estableciendo los saltos en 32 para la siguiente subred.

3) Tercer paso:

Dado que el salto de la subred es de 32:

Subred 1 : 192.168.20.0 - 172.168.20.31 -->Broadcast

Subred 2 : 192.168.20.32 - 172.168.20.63 -->Broadcast

Subred 3 : 192.168.20.64 - 172.168.20.91 -->Broadcast

Donde el rango de Hosts va desde 33 al 62, pero como la primera dirección esta reservada, quedaria desde:

192.168.20.34 --> primera dirección de host

.....

192.168.20.62 --> última dirección de host

***Nota:** La calculadora de subredes que se encuentra como un applet en el proyecto, le ayudará a calcular nuevas subredes de direcciones IP, a partir de una IP dada, y el CIDR o la máscara d*

red, además entrega los datos del wildcard, broadcast, ip máximo, ip mínimo, y la red a la que pertenece.

CAPITULO III

LAS REDES EN LAS

EMPRESAS

1 REDES CORPORATIVAS

1.1 INTRODUCCIÓN

Podemos definir como una red corporativa, al enlace que permite conectar diferentes localizaciones de una empresa, de un modo permanente, privado, y seguro a través de diferentes medios, y nos faculta enviar y recibir los datos que se necesitan en diferentes puntos de la empresa.

1.2 USO

Las redes corporativas son usadas por la mayoría de empresas, que necesitan compartir su información, al igual que otros servicios, entre estos pueden estar una conexión a internet, impresoras, documentos importantes, etc.

Para poder implementar una red corporativa, se debe primero tener realizada la conexión de los puntos que deseamos sean parte de la red, mediante una topología, la cual debe ser escogida dependiendo de las necesidades de la empresa, y que debe estar definida por estándares que nos van a ir dirigiendo en el modo en el que debemos *diseñar, construir y administrar*, nuestra red, a éste tipo de implementación de una red se llama *Cableado Estructurado*.

Una conexión de cableado estructurado (CE) es una serie de estándares definidos por la TIA/EIA que definen todo el cableado y componentes instalados en una red basados en un orden lógico y organizado, es decir, que el sistema está diseñado en bloques que tienen características de desempeño muy específicas.

1.3 RIESGOS

Cuando decidimos realizar la implementación de una red corporativa o red local LAN por sus siglas en inglés (Local Area Network), debemos estar conscientes que vamos a estar vulnerables frente a algunas circunstancias de las que no vamos poder escaparnos,

entre las mas comunes están los ataques de hackers, o de personas que intentan infiltrarse a nuestra red para obtener información valiosa de la empresa, para poder evitar ésto es necesario tener configurado un buen firewall, que nos defienda de la mayor cantidad de puertas abiertas o puertos, y cubra nuestras espaldas mientras nos ocupamos de los problemas que se puedan suceder al interno de la empresa.

Algunos otros inconvenientes se dan dentro de nuestra propia red, esto debido muchas de las veces a la falta de conocimientos por parte de ciertos usuarios de la red, ya que algunos de los usuarios sin el mayor ánimo de causar daños a la empresa, pueden alterar documentos que están compartidos en la red o en su defecto sobre escribirlos por documentos mas antiguos, algunas veces también se puede dar fugas de información desde el interno de la red, cualquiera que sea el caso siempre hay que manejar políticas internas que nos permitan evitar cualquier inconveniente, implementando diferentes niveles de acceso, sea éste por claves o por acceso a la red de nuevos usuarios luego de un tiempo de prueba, y de una pequeña introducción a las cosas a las que puede acceder y los beneficios de estar en la red local de nuestra compañía.

1.4 ORGANIZACIONES DE ESTÁNDARES DE CABLEADO

Hay muchas organizaciones involucradas en el cableado estructurado en el mundo. En Estados Unidos es la ANSI, TIA e EIA, Internacionalmente es la ISO (International Standards Organization). El propósito de las organizaciones de estándares es formular un conjunto de reglas comunes para todos en la industria, en el caso del cableado estructurado para própositos comerciales es proveer un conjunto estándar de reglas que permitan el soporte de múltiples marcas o fabricantes. Existen varias referencias en SCE alrededor del mundo, tales como:

- **EIA/TIA 568A/B** El primer estándar de cableado estructurado Publicado en EUA por la EIA/TIA en 1991
- **ISO/IEC 11801** Versión internacional del estándar 568
- **CENELEC EN 50173** Estándar de cableado estructurado británico
- **CSA T529** Estándar de cableado estructurado Canadiense.

1.5 CONSUMO DE ANCHO DE BANDA

1.5.1 Introducción

Cuando estamos conectados a una red, se nos da acceso a hacer uso de todos los recursos que cada nodo nos puede ofrecer, entendemos por recurso al servicio que nos puede brindar el nodo al cual nos estamos conectando, cuando hacemos una petición a un nodo de nuestra red, éste nos responde con un mensaje de aceptación o de negación, una vez que el nodo acepta la petición en ese momento establecemos la conexión entre los nodos.

1.5.2 Gestionando el consumo

Cuando realizamos la implementación de una red corporativa, usamos varios medios de comunicación que son los que nos permiten realizar las transferencias de datos entre nodos de nuestra red, a estos métodos de comunicación se los conoce como protocolos, al cual podemos definir como: “*En informática, un **protocolo** es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.*”¹, estos protocolos nos permiten, como dice la definición, saber como comunicarnos con nuestros vecinos, que hacer y cuando vamos a dejar de hablar con ellos.

Cuando establecemos una comunicación con otros nodos, estamos haciendo un consumo de los recursos que implementamos en la red, un ejemplo de esto es cuando tenemos instalado un software de control de personal y que hace consultas a un servidor de MySQL, cada vez que nos conectamos al servidor y realizamos una petición hacemos uso del protocolo tcp hacia el puerto 3306 (a menos que lo hayamos configurado diferente), al hacer esto nuestro ancho de banda en la red local va a disminuir por unos segundos mientras realizamos la conexión.

¹http://es.wikipedia.org/wiki/Protocolo_de_red

Otro ejemplo es cuando tenemos una conexión a internet y decidimos compartirla con algunos de los miembros de la red, en ese caso el ancho de banda a la conexión de internet, puede ser diferente al ancho de banda que manejamos en nuestra red local, porque en la red local el ancho de banda está dado por los dispositivos en nuestra red (a menos que nosotros mismo limitemos uno diferente), en cambio el ancho de banda con el que sale y recibimos información de internet va a estar limitado por nuestro proveedor de internet o ISP por sus siglas en inglés (Internet Service Provider), el ancho de banda es medido en bits por segundo, entonces definimos como ancho de banda a “*La cantidad de datos que podemos transmitir en una unidad de tiempo*”, por ejemplo si tenemos una conexión de 512 kbps quiere decir vamos a poder enviar y recibir datos a un máximo de 512000 bits en un segundo, claro que todo esto depende de la conexión que tengamos, cable utp, fibra óptica, wireless, además de factores externos como las interferencias de otras señales en un mismo canal, la corriente eléctrica, etc., los cuales van a influir en el estado de nuestra conexión.

Cuando tenemos compartido el ancho de banda en nuestra red local, debemos estar conscientes que el ancho de banda que fue asignado por nuestro ISP, va a servir de entrada y salida para todos los equipos que se van a conectar, si nosotros ocupamos varios servicios en nuestra red (ftp, web, correo, MySQL, etc), todos éstos servicios van a realizar un consumo según sus necesidades creando conflictos entre unos y otros, por tal motivo podremos darnos cuenta cómo se pone lento nuestro servicio, si ha esto le sumamos las descargas de archivos que algunos usuarios hacen, o el envío de documentos, fotos, música o videos entre miembros de la red local o corporativa, va a ser mucho mas visible la lentitud en nuestro servicio, el cual no es causado por la conexión que ha realizado nuestro ISP, es mas bien por el mal consumo que nosotros le estamos dando dentro de nuestra red.

1.5.3 Monitoreo

Cuando hemos llegado al punto en el que no sabemos, porqué no tenemos el rendimiento que esperábamos de nuestra red, y existe el consumo excesivo de recursos, es cuando debemos pensar no en la ampliación de ancho de banda, mas bien debemos

iniciar a pensar en la optimización de los recursos con los que actualmente contamos en nuestra red, y como podemos mejorar nuestra conexión, los servicios a los que accede cada nodo, y el desperdicio de banda que es causado por malos hábitos de los usuarios.

Es aquí donde tiene una fundamental importancia el manejo de una forma que nos permita monitorear a donde quien y cuanto tiempo se conectan los nodos de nuestra red, para poder hacer ésto es necesario tener implementados 2 software que nos permitan saber las falencias de nuestra red.

1.5.3.1 Puertas abiertas

Cuando realizamos la instalación de una red es posible que personas externas a la empresa quieran ingresar a nuestra red para obtener información o causar simplemente daños, es aquí importante que el administrador de la red decida implementar algunos métodos de protección entre uno de los básicos está la instalación de un ScannPort o escaner de puertos, la principal función de un escaner de puertos es poder saber por que puertos pueden existir infiltraciones en nuestra red, el funcionamiento básico de éstos es emitir una trama de paquetes hacia cada uno de los puertos que escogemos los mas conocidos o en su defecto todos los puertos de nuestro equipo, de éste modo de los puertos de los que se recibe una respuesta de entrada son los puertos que se encuentran habilitados y permiten un acceso a nuestra red, si alguien logra entrar, a través de nuestro firewall, tengamos por seguro que será mas fácil poder obtener información de nuestra red o en su defecto puede causar daños de configuración, una vez que alguien encuentra una puerta abierta, es fácil que inicie a sacar información no en una sola conexión si no en varias en las que vaya obteniendo poco a poco lo que necesita.

Nosotros como administradores de nuestra red estamos en la posibilidad de evitar éstos problemas y debemos hacer constantes escaneos a nuestro servidor firewall, para ésto podemos utilizar programas como Nmap, que son programas especializados en el escaneo de puertos de un equipo, no de toda la red, si queremos hacer un mejor control debemos ir realizando un escaneo por cada equipo que consideremos esencial o prioritario, dependiendo del servicio que nos brinde.

1.5.3.2 **Vigilando el consumo de la red**

Cuando realizamos la implementación de una red de datos, una de las mayores seguridades que podemos instalar es un programa que se llama *Sniffer*.

Se instalará en nuestro proxy o en el equipo que gestiona el acceso a internet un sniffer, lo que nos permite este programa es capturar las tramas de red, o los datos que están viajando a través de las tarjetas de red.

Es algo común que por necesidad de los fines de la empresa, el medio de transmisión, sea compartido a varias computadoras y dispositivos de red, lo que hace posible que uno o varios usuarios de la red usen desmedidamente o en su defecto hagan uso de los servicios que se tienen implementados en la empresa para fines que no tienen nada que ver con el flujo de trabajo, el sniffer implementado en nuestra red permite vigilar constantemente los paquetes y el consumo realizado por cada persona que se le permite el acceso a la red, no solo del acceso a internet también a los servicios internos, (correo, MySQL, etc), además de fines productivos como monitorizar las redes para detectar y analizar fallos o ingeniería inversa de protocolos de red, también es habitual su uso para fines maliciosos, como robar contraseñas, interceptar mensajes de correo electrónico, espiar conversaciones de chat, ver documentos enviados adjuntos, etc, para conseguir esto el sniffer pone la tarjeta de red o NIC, en un estado conocido como “modo promiscuo”, en el cual en la capa de enlace de datos de los niveles OSI, no son descartadas las tramas no destinadas a la MAC address de la tarjeta; de esta manera se puede capturar (sniff, esnifar, olfatear) todo el tráfico que viaja por la red.

```

administrador@beaserver:/home/Administrador
~/Archivo [editar] Ver Terminal Solapas Ayuda
run 0:00:21 device eth0 pkt[filter: none
[control: on [bits-bytes/s [local agr.: none [remote agr.: none
[quit [help [starting [packets [.] pause [0-9] switch device
LOCAL <-> REMOTE
[IP] PORT PROTO [IP] PORT TXBPS RXBPS TOTALBPS
204.2.241.136 <-> 150.190.0.3 1923 1.17k/s 195b/s 1.36k/s
204.2.241.136 GET /nugget/26688_6832715 80 TCP 150.190.0.3 3.51k 586b 4.00k
150.190.0.1 <-> 150.190.0.2 59268 891b/s 66b/s 957b/s
150.190.0.1 22 TCP 150.190.0.2 13.1k 1.48k 14.5k
snlmsg2020100.phx.gbl <-> 150.190.0.3 2094 23b/s 198b/s 213b/s
65.54.50.159 1863 TCP 150.190.0.3 619b 1.86k 2.46k
dostoja.telconet.net <-> 150.190.0.3 1845 136b/s 25b/s 161b/s
288.53.221.17 53 UDP 150.190.0.3 489b 77b 480b
64.4.35.253 <-> 150.190.0.3 2816 15b/s 31b/s 46b/s
64.4.35.253 7001 UDP 150.190.0.3 62b 124b 186b
srv1.telconet.net <-> 150.190.0.2 49586 8b/s 37b/s 37b/s
288.53.152.148 53 UDP 150.190.0.2 392b 75b 467b
UNKNOWNv4 <-> UNKNOWNv4 0 8b/s 12b/s 28b/s
0.0.0.0 0 ARP 0.0.0.0 42b 60b 102b
dostoja.telconet.net <-> 150.190.0.2 37715 8b/s 8b/s 8b/s
288.53.221.17 53 UDP 150.190.0.2 215b 75b 298b
snlmsg1828112.phx.gbl <-> 150.190.0.2 58673 8b/s 8b/s 8b/s
65.54.50.36 1863 TCP 150.190.0.2 431b 283b 634b
dostoja.telconet.net <-> 150.190.0.2 52728 8b/s 8b/s 8b/s
288.53.221.17 53 UDP 150.190.0.2 8b 75b 75b
snlmsg1818514.phx.gbl <-> 150.190.0.10 35970 8b/s 8b/s 8b/s
65.54.48.38 1863 TCP 150.190.0.10 74b 137b 211b
TOTAL 2.22k/s 556b/s 2.76k/s
23.2k 5.48k 28.7k

```

Figura N° 9: Captura trabajo jnettop.

Entre los más conocidos Sniffers, están jnettop, Wireshark (Ethereal), éste sniffer es de fácil instalación y actualización se lo puede instalar en cualquier distribución de sistema operativo Linux, y brinda interesantes opciones de configuración y reportes, éste es el sniffer con el que trabajó la validación en cuanto al monitoreo.

CAPÍTULO IV

METODOLOGÍA

2 METODOLOGÍA DE LO USADO

En el presente capítulo vamos a internarnos en los métodos y las herramientas usadas para el desarrollo del proyecto, vamos a ir viendo en cada uno de los puntos especificados, como cada una de las tecnologías empleadas ayudaron a mejorar el rendimiento y la calidad del proyecto, de este modo justifico el uso de las herramientas y como éste proyecto aporta a la optimización del manejo de los recursos de una red privada.

2.1 MÉTODOS

Para la elaboración del proyecto se usó el método científico, integrado por el método inductivo y deductivo, éste método permitió realizar el análisis y la síntesis a partir de la observación de hechos y fenómenos ocurridos, además sirve de apoyo a los conocimientos científicos y teóricos, que permiten aceptar la factibilidad del proyecto, de éste modo se parte del planteamiento del problema, para luego formular la hipótesis, generar los objetivos a cumplir, y entorno a esto desarrollar la base teórica.

2.2 METODOLOGÍA ICONIX PARA EL DESARROLLO DE SOFTWARE

Aunque esta metodología es más compleja que la metodología Xp y Rup, el soporte para expresar el funcionamiento es mas simple, ya que para generar su representación se basa únicamente en los casos de uso, además nos ayuda de mejor manera para representar los procesos de desarrollo de objetos, utilizando ésta metodología en cualquiera de las etapas.

2.3 LENGUAJE

2.3.1 Introducción

2.3.1.1 ¿Qué son los Servlets Java?

Los Servlets son la respuesta de la tecnología Java a la programación CGI (El sistema más antiguo que existe para la programación de páginas dinámicas. Los CGI se escriben habitualmente en el lenguaje Perl, sin embargo, otros lenguajes como C, C++ o Visual Basic pueden ser también empleados para construirlos). Son programas que se ejecutan en un servidor Web y construyen páginas Web. Construir páginas Web rápidamente es útil (y comúnmente usado) por un número de razones:

- La página Web está basada en datos enviados por el usuario.
- Las páginas Web que usan información desde bases de datos corporativas u otras fuentes.

2.3.1.2 **¿Cuáles son las Ventajas de los Servlets sobre el CGI "Tradicional"?**

Los Servlets Java son más eficientes, fáciles de usar, más poderosos, más portables, y más baratos (en cuanto al uso de recursos) que el CGI tradicional y otras muchas tecnologías del tipo CGI. Con CGI tradicional, se arranca un nuevo proceso para cada solicitud HTTP. Si el programa CGI hace una operación relativamente rápida, la sobrecarga del proceso de arrancada puede dominar el tiempo de ejecución. Con los Servlets, la máquina Virtual Java permanece arrancada, y cada petición es manejada por un thread Java de peso ligero, no un pesado proceso del sistema operativo. De forma similar, en CGI tradicional, si hay N peticiones simultáneas para el mismo programa CGI, el código de este problema se cargará N veces en memoria. Sin embargo, con los Servlets, hay N threads pero sólo una copia de la clase Servlet. Los Servlet también tienen más alternativas que los programas normales CGI para optimizaciones como los cachés de cálculos previos, mantener abiertas las conexiones de bases de datos, etc. , los Servlets tienen una gran infraestructura para análisis automático y decodificación de datos de formularios HTML, leer y seleccionar cabeceras HTTP, manejar cookies, seguimiento de sesiones, y muchas otras utilidades.

Los Servlets Java nos permiten fácilmente hacer muchas cosas que son difíciles o imposibles con CGI normal. Por algo, los servlets pueden hablar directamente con el servidor Web. Esto simplifica las operaciones que se necesitan para buscar imágenes y

otros datos almacenados en situaciones estándar. Los Servlets también pueden compartir los datos entre ellos, haciendo las cosas útiles como almacenes de conexiones a bases de datos fáciles de implementar. También pueden mantener información de solicitud en solicitud, simplificando cosas como seguimiento de sesión y el caché de cálculos anteriores.

Los Servlets están escritos en Java y siguen un API bien estandarizado. Consecuentemente, los servlets están soportados directamente o mediante plug-in en la mayoría de los servidores Web.

2.3.2 Introducción a JSP

JavaServer Pages (JSP) es una tecnología basada en el lenguaje Java que permite incorporar contenido dinámico a las páginas web. Los archivos JSP combinan HTML con etiquetas especiales y fragmentos de código Java.²

Esto quiere decir que Java Server Pages (JSP) es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente .

2.3.2.1 JSP contra ASP

JSP y ASP ofrecen funciones similares. Ambos utilizan etiquetas para permitir código embebido en una página HTML, seguimiento de sesión, y conexión a bases de datos. Algunas de las diferencias mas significativas son:

- Las páginas ASP están escritas en VBScript y las páginas JSP están escritas en lenguaje Java. Por lo tanto, las páginas JSP son independientes de la plataforma y las páginas ASP no lo son.
- Las páginas JSP usan tecnología JavaBeans como arquitectura de componentes y las páginas ASP usan componentes ActiveX.

²(<http://java.sun.com/jsp>)

Más allá de estas diferencias triviales, hay varias diferencias importantes, que podrían ayudarnos a elegir la tecnología para nuestras aplicaciones:

- **Velocidad y Escalabilidad:** Aunque las páginas ASP son cacheadas, siempre son interpretadas, las páginas JSP son compiladas en Servlets Java y cargadas en memoria la primera vez que se las llama, y son ejecutadas para todas las llamadas siguientes. Esto le da a las páginas JSP la ventaja de la velocidad y escalabilidad sobre las páginas ASP.
- **Etiquetas Extensibles:** Las páginas JSP tiene una característica avanzada conocida como etiquetas extensibles. Este mecanismo permite a los desarrolladores crear etiquetas personalizadas. En otras palabras, las etiquetas extensibles nos permiten extender la sintaxis de las etiquetas de las páginas JSP. No podemos hacer esto en ASP.
- **Libertad de Elección:** Las páginas ASP sólo trabajan con Microsoft IIS y Personal Web Server. El uso de páginas ASP requiere un compromiso con los productos de Microsoft, mientras que las páginas JSP no nos imponen ningún servidor web ni sistema operativo. Las páginas JSP se están convirtiendo en un estándar ampliamente soportado.

El código fuente de una página JSP puede contener:

- **Directivas:** Indican información general de la página, como puede ser importación de clases, página a invocar ante errores, si la página forma parte de una sesión, etc.
- **Declaraciones:** Sirven para declarar métodos o variables.
- **Scriptlets:** Código Java embebido.
- **Expresiones:** Expresiones Java que se evalúan y se envían a la salida.
- **Tags JSP:** Etiquetas especiales que interpreta el servidor.

Un ejemplo simple de archivo JSP sería:

```
<% @page import="java.util.*"%>
```

```
<%! String cadena="Bienvenidos a JSP"; %>

<html>

    <body>

        <%= cadena %>

        <br>

        <% out.println(new Date()); %>

    </body>

</html>
```

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

2.3.3 ELEMENTOS DE JSP

2.3.3.1 Código Java

Podemos insertar código Java dentro de JSP de tres formas: Expresiones, scriptlets y declaraciones.

2.3.3.1.1 **Expresiones:** Son fragmentos de código Java, con la forma `<%= expresión %>` que se evalúan y se muestran en la salida del navegador. En general, dentro de una expresión podemos usar cualquier cosa que usaríamos dentro de un `System.out.print(expr);`

Ejemplos:

```
<%= "Tamaño de vector: "+vector.size() %>
```

```
<%= new java.util.Date() %>
```

2.3.3.1.2 **Scriptlets:** Son fragmentos de código Java con la forma `<% código %>`, en general, podemos insertar cualquier código que pudiéramos usar dentro de una función Java. Para acceder a la salida del navegador, usamos el objeto implícito `out`.

Ejemplos:

```
<table>

    <% for (int i=0;i<10;i++) { %>

        <tr><td> <%=i%> </td></tr>

        <% }

    %>

</table>
```

Dentro de un scriptlet podemos usar cualquier librería de Java, incluyendo las propias, lo cual hace que resulte muy sencillo construir interfaces web de entrada y salida para nuestras clases.

```
<% String parametro1=request.getParameter("parametro1");

String parametro2=request.getParameter("parametro2");

MiClase objetodemiclase=new MiClase();

String salida=miClase.procesa(parametro1, parametro2);

%>

<%= salida %>
```

Para introducir comentarios en JSP, usaremos las marcas `<%-- comentario --%>`, dentro de un scriptlet o declaración podemos usar comentarios siguiendo la sintaxis de Java.

```
<%-- Comentario JSP --%>
```

```
<!-- Comentario HTML -->
```

```
<%
```

```
    // Comentario
```

```
    /* Comentario */
```

```
%>
```

2.3.3.1.3 **Declaraciones:** Contienen declaraciones de variables o métodos, con la forma `<%! declaración %>`. Estas variables o métodos serán accesibles desde cualquier lugar de la página JSP.

Hay que tener en cuenta que el servidor transforma la página JSP en un servlet, y éste es usado por múltiples peticiones, lo que provoca que las variables conserven su valor entre sucesivas ejecuciones.

Ejemplos:

```
<%! int numeroAccesos=0; %>
```

```
<html>
```

```
    <body>
```

```
        <%=
```

```
            "La página ha sido accedida "+(++numeroAccesos)+ " veces desde el  
            arranque del servidor"
```

```
        %>
```

```
    </body>
```

```
</html>
```

```
<%! java.util.Date primerAcceso=new java.util.Date(); %>
```

```
<html>
```

```
  <body>
```

El primer acceso a la página se realizo en:

```
  <%= primerAcceso %>
```

```
  </body>
```

```
</html>
```

2.3.3.2 Directivas

Las directivas son elementos que proporcionan información al motor JSP, e influirán en la estructura del servlet generado. Hay tres tipos de directivas: page, taglib e include.

2.3.3.2.1 **page:** Se indica con la forma `<%@ page atributo="valor">`. Tiene diversos usos, entre los usados están:

- **Importar clases.** Importar código, de la misma forma que se realiza en un programa en Java, se indica con el atributo import.

Ejemplo:

```
<%@page import="java.io.*, miPackage.miClase"%>
```

- **Indicar si la página tendrá acceso a la sesión.** Se especifica con el atributo session. El uso de sesiones se verá con más detalle en el apartado de objetos implícitos.

Ejemplo:

```
<%@page session="true" import="java.util.ArrayList"%>
```

- **Gestión de errores.** Permite redireccionar a una página cuando se produzca un error, se indica con los atributos errorPage y isErrorPage.

Ejemplos:

```

<% @page errorPage="error.jsp">

<% @page isErrorPage="yes">

<html>

    <body>

        Error, contacte con el administrador [...]

    </body>

</html>

```

2.3.3.2.2 **include:** Permite incluir un archivo en el lugar donde se especifique, al contrario que con la acción `<jsp:include>` que veremos más adelante, la directiva `include` simplemente copia el contenido del archivo byte a byte, siendo el resultado similar a si copiáramos el texto del archivo incluido y lo pegáramos en el JSP.

Ejemplo:

```

<html>

    <head>

        <% @ include file="titulo.txt"%>

    </head>

    <body>

        <% @ include file="cuerpoPagina.jsp"%>

    </body>

</html>

```

2.3.3.2.3 **taglib:** Se emplea para indicar que se van a emplear librerías de etiquetas. Se verá con más detalle en el apartado de JSTL.

Ejemplo:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

2.3.3.3 **Acciones** : Las acciones tienen la forma `<jsp:accion [parámetros]/>`, y tienen diversos usos, entre los que destacan la inclusión de páginas y transferencia de control.

2.3.3.3.1 **Inclusión de páginas**

Se realiza con la acción `<jsp:include page="pagina.jsp">`. Incluye la salida de otra página JSP en la actual, al contrario que con la directiva `<%@include file="fichero.ext"%>` la página incluida se ejecuta y su salida se inserta en la página que la incluye, con la directiva se incluye el contenido del archivo (no su salida) y se ejecuta conjuntamente con la página principal.

La página incluida tiene acceso a los parámetros enviados a la principal, y podemos enviarle nuevos parámetros con la subetiqueta `<jsp:param name="nombre" value="valor"/>`.

Ejemplo:

```
<html>

  <head>

    <jsp:include page="cabecera.jsp"/>

  </head>

  <body>

    <jsp:include page="cuerpo.jsp">

      <jsp:param name="tipo" value="paginaPrincipal"/>

    </jsp:include>
```

```
</body>  
  
</html>
```

2.3.3.3.2 **Transferencia de control**

Se realiza con la acción `<jsp:forward page="pagina.jsp"/>`. La petición es redirigida a otra página, y la salida de la actual se descarta. Al igual que con la inclusión, la página a la que se redirige tiene acceso a los parámetros pasados a la actual, y es posible el envío de nuevos parámetros.

Ejemplo:

```
<jsp:forward page="principal.jsp">  
  
    <jsp:param name="titulo" value="Principal"/>  
  
</jsp:forward>
```

2.3.4 **Objetos implícitos .**

En JSP disponemos de algunos objetos implícitos, que nos permitirán acceder a diferente información y realizar diversas acciones. En JSP tenemos los siguientes objetos implícitos: request, response, out, session, application, config, pageContext, y page.

2.3.4.1 **Request** : Es un objeto de la clase `HttpServletRequest`, su uso principal es el acceso a los parámetros de la petición. Destacaremos las siguientes funciones:

- `String getParameter(String name)` , devuelve el valor de un parámetro.

- Enumeration `getParameterNames()` , devuelve una enumeración con los nombres de todos los parámetros de la petición.
- `String[] getParameterValues(String name)` , los parámetros pueden tener valor múltiple, con esta función recuperamos un array con todos los valores para un nombre dado.
- `String getRemoteAddr()` , devuelve la IP del host desde el que se realiza la petición .
- `String getRemoteHost()` , Devuelve el nombre del host desde el que se realiza la petición.

2.3.4.2 **Response** : Es un objeto de la clase `HttpServletResponse`, que asiste al servlet en su generación de la respuesta para el cliente, contiene funciones para manejo de cabeceras, códigos de estado, cookies y transferencia de control.

2.3.4.3 **Out** : Es un objeto de la clase `JspWriter`, es el que nos permite acceder a la salida del navegador desde los scriplet.

2.3.4.4 **Session** : Es un objeto de la clase `HttpSession`. Nos permite acceder a la sesión asociada a la petición. A través de este objeto podemos, entre otras cosas, guardar objetos que serán accesibles desde cualquier JSP de la sesión o invalidarla.

Para guardar y recuperar información usaremos:

```
Object session.getAttribute("clave");
```

```
void session.setAttribute("clave", Object objeto);
```

Y para invalidar la sesión:

```
void session.invalidate();
```

2.3.4.5 **Application** : Es un objeto de la clase ServletContext. Este objeto es común para toda la aplicación web y, entre otras cosas, nos permite almacenar información que será accesible desde todas las páginas de la aplicación web, independientemente de la sesión.

Para guardar y recuperar valores:

```
Object application.getAttribute("clave");
```

```
void application.setAttribute("clave", Object objeto);
```

2.3.4.6 **Config** : Es un objeto de la clase ServletConfig. Permite acceder a parámetros de inicialización del servlet y a su contexto.

2.3.4.7 **PageContext** : Es un objeto de la clase PageContext. Entre otras cosas, nos permite almacenar información localmente a la página.

Para guardar y recuperar valores:

```
Object pageContext.getAttribute("clave");
```

```
void pageContext.setAttribute("clave", Object objeto);
```

También podemos usar PageContext para almacenar y recuperar información en sesión y en aplicación:

Almacenar en contexto de página:

```
PageContext.setAttribute("clave", obj, PageContext.PAGE_SCOPE);
```

```
PageContext.setAttribute("clave", obj);
```

Almacenar en contexto de sesión:

```
PageContext.setAttribute("clave",obj, PageContext.SESSION_SCOPE);
```

```
session.setAttribute("clave", objeto);
```

Almacenar en contexto de aplicación:

```
PageContext.setAttribute("clave",obj,PageContext.APPLICATION_SCOPE);
```

```
application.setAttribute("clave",objeto);
```

2.3.4.8 **Page** : Es un sinónimo de this, no tiene utilidad en el estado actual de la especificación.

2.3.5 Comunicación entre formularios HTML y páginas JSP

En esta sección se hará un breve repaso a los formularios HTML, y veremos las distintas formas de enviar parámetros desde un formulario y de recibirlos desde un JSP. Dejaremos al margen algunos tipos como botones y envío de archivos, por tener relevancia sólo en lenguajes de script del lado del cliente o quedar fuera del alcance de este tutorial.

2.3.5.1 Conceptos básicos

Un formulario HTML tiene la forma:

```
<form action="destino" method="método">
```

Elementos de formulario

```
</form>
```

En destino especificaremos la página que recibe los datos del formulario, en el atributo method podemos indicar dos valores diferentes GET y POST. Si no se especifica el valor de los atributos, los valores por defecto son la página actual para action y GET para method .

2.3.5.2 GET y POST

Cuando usamos GET, la información se codifica directamente en la URL, con la forma:

```
http://url?param1=valor1&param2=valor2...&paramN=valorN
```

Con GET no podemos manejar grandes cantidades de información, y existe la desventaja de que el servidor o el navegador guarden en caché la página llamada. Hay que tener en cuenta que los logs del servidor y el historial del navegador guardarán el acceso incluyendo los parámetros, lo cual hace desaconsejable GET para el envío de información privada. Por otro lado, al visualizarse en la URL los parámetros, facilita el desarrollo y depurado de la aplicación web, y en algunos casos, es imprescindible para realizar estadísticas basadas en los logs del servidor.

Con POST la información se envía directamente al servidor, no se codifica en la URL, y además permite el envío de grandes cantidades de información, como podrían ser archivos.

2.3.6 Introducción a JSTL

En JSP es posible definir librerías de etiquetas personalizadas, estas etiquetas no son más que clases Java que heredan de determinadas clases (p.e. BodyTagSupport). Estas clases se agrupan en librerías mediante un archivo descriptor TLD (Taglib descriptor). Existen innumerables librerías, comerciales y gratuitas, que implementan las más diversas funciones.

A partir de JSP 1.2. se introduce un conjunto de librerías en la especificación, pasando a ser estándar, es la librería JSTL (JavaServer pages Standard Tag Library).

JSTL consta de los siguientes grupos de etiquetas:

Tabla N° 3: JSTLs mas usados y conocidos

TIPO	URI (identificador)	PREFIJO
Core	http://java.sun.com/jsp/jstl/core	c
Xml	http://java.sun.com/jsp/jstl/xml	x
Internacionalización y formato	http://java.sun.com/jsp/jstl/fmt	fmt
SQL	http://java.sun.com/jsp/jstl/sql	sql

2.3.7 CSS y JSP

La tecnología CSS (Cascading Style Sheets) se considera una de las mayores evoluciones en las formas de desarrollar sitios web, las CSS son hojas de estilo de diseño gráfico para páginas Web que permiten cambiar simultáneamente la apariencia de todas las páginas relacionadas con el mismo estilo, esto es algo similar a insertar un look and feel, en una aplicación desktop.

La idea es simplificar el desarrollo del sitio web ahorrando tiempo en la presentación, ya que en vez de crear un formato para cada página del sitio, se crea una única página que contiene todo los estilos usados, así cuando se requiera cambiar algún estilo, bastará con modificar un único archivo.

Las hojas de estilo son posibles incluirlas en páginas HTML, por lo tanto tenemos la facilidad de incluirlas en páginas JSP, de éste modo brindamos a nuestros portales Web, interesantes presentaciones que atraen a los visitantes.

2.3.8 Internacionalización

Cuando hacemos el desarrollo de un sitio Web, es interesante tener todos los mensajes (normales o de error) en un archivo de propiedades, además sería útil que tuviésemos un archivo de propiedades por cada idioma al que deseamos traducir nuestro sitio, de éste

modo se cargará de forma automática los mensajes en función del idioma predefinido en el sistema operativo.

Este tipo de soluciones al manejo de textos estáticos es posible en jsp, ya se nos permite incluir archivos properties en los directorios de nuestro proyecto de tal manera que el archivo properties es utilizable en cualquier lugar de nuestro proyecto, este uso de “mensajes” es bastante útil ya que nos permite modificar el texto sin la necesidad de buscar la página en la que se encuentra, reutilizar los mensajes no solo en párrafos, también en elementos como links, botones, tablas, etc, en jsp es posible gracias a la utilización de la interface MessageSource.

La implementación más manejada de la interface MessageSource es ResourceBundleMessageSource, y es posible utilizarla no únicamente en el desarrollo de aplicaciones web, también es implementable en aplicaciones de desktop y de consola.

2.4 BASE DE DATOS

En el desarrollo de aplicaciones que pretenden reducir costos al momento de incluir una base de datos se recomienda el uso de una base de datos que aplique la Licencia GNU- GPL, una de las alternativas mas conocidas es MySQL la base de datos open source que goza de mayor popularidad y, posiblemente, mejor del mundo en su ámbito. En la actualidad MySQL está bajo la dirección de Sun Microsystems, en todas las versiones de MySQL, se ofrece el acceder mediante 2 ediciones Enterprise, la cual tiene soporte 24/7 y por la cual se paga, y la edición Community la cual es la versión libre. Actualmente se encuentra la versión 4.0, 5.0, 5.1, 6.0, además Sun pone a disposición conectores para los diferentes lenguajes de programación como JAVA, .NET, C, C++, así como herramientas para administración consulta y mantenimiento de MySQL.

Características

✂ Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.

✂ Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.

Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.

2.5 FRAMEWORKS

Una de las innovaciones en el presente proyecto es la aplicación del Framework Spring 2.5, este Framework promueve las buenas prácticas en el desarrollo de aplicaciones, tanto en el diseño como en la programación, maneja patrones de diseño como Factory, Abstract Factory, Builder, Decorator, Service Locator, los cuales son ampliamente aplicados dentro del desarrollo de aplicaciones web y desktop,.

Spring Framework fue escrito inicialmente por Rod Jonhson, quien lo lanzó junto a la publicación de su libro *Expert One-on-One Java EE Design and Development*, por el editorial de libros computacionales Wrox Press, en octubre de 2002, aunque inicialmente fue pensado para el desarrollo de aplicaciones JAVA, en la actualidad hay una versión para la plataforma .Net, Spring.net.

Spring se enfoca principalmente en el manejo de objetos de negocio, dentro de una arquitectura por capas, aunque Spring no obliga para su utilización un modelo en específico, se está promoviendo en la actualidad y en mayor número entre los desarrolladores, utilizar para la realización de proyectos web, el patrón Modelo-Vista-Controlador, una de las principales ventajas de Spring es su modularidad, entre las principales características que nos ofrece Spring están:

- **Core Container o Contenedor de Inversión de Control (Inversion of Control, IoC):** es el núcleo del sistema. Responsable de la creación y configuración de los objetos.
- **Aspect-Oriented Programming Framework:** que trabaja con soluciones que son utilizadas en numerosos lugares de una aplicación, lo que se conoce como asuntos transversales (cross-cutting concerns).
- **Data Access Framework:** que facilita el trabajo de usar un API como JDBC, Hibernate, etc.
- **Transaction Management Framework:** una de las razones mas importantes por las que se promueve Spring es el soporte global a transacciones, entre los principales beneficios de éste segmento de Spring están:
 - Proveer un modelo de programación consistente a través de las API de transacciones diferentes, tales como JTA, JDBC, Hibernate, JPA, y JDO.
 - Proporciona un API sencillo para la gestión de transacciones de una serie de APIs complejos como JTA.
 - Se integra muy bien con las abstracciones de acceso a datos.
- **Remote Access framework:** Facilita la existencia de objetos en el servidor que son exportados para ser usados como servicios remotos.

2.5.1 Spring MVC Modelo Vista Controlador

El patrón MVC es un patrón que nos ayuda a mantener por separado, la capa de presentación de la lógica de negocio, en una aplicación MVC todas las peticiones son manejadas por los controladores. Un "controlador" es responsable de interpretar la petición del usuario e interactuar con los objetos de la capa de negocio de la aplicación, a fin de cumplir con la petición. Estos objetos de negocio están representados como el "modelo" dentro del patrón MVC. Basándose en los resultados de la ejecución de la solicitud, el controlador decide que "mostrar", la vista utiliza los datos en el modelo para crear la presentación que se devuelve al usuario.

Si hemos trabajado en el desarrollo de aplicaciones web, lo más probable es que hayamos trabajado con algunos frameworks como Struts o JSF y aunque estos frameworks se han extendido por su uso entre los programadores Java, el Framework Spring MVC promete proporcionar una alternativa más sencilla para el uso y aplicación de diferentes frameworks.

2.6 IDE

2.6.1 Historia ³

NetBeans comenzó como un proyecto estudiantil en Republica Checa (originalmente llamado Xelfi), en 1996 bajo la tutoría de la Facultad de Matemáticas y Física en la Universidad de Charles en Praga. La meta era escribir un entorno de desarrollo integrado (IDE) para Java parecida a la de Delphi. Xelfi fue el primer entorno de desarrollo integrado escrito en Java, con su primer pre-release en 1997.

Xelfi fue un proyecto divertido para trabajar, ya que las IDEs escritas en Java eran un territorio desconocido en esa época. El proyecto atrajo suficiente interés, por lo que los estudiantes, después de graduarse, decidieron que lo podían convertir en un proyecto comercial. Prestando espacios web de amigos y familiares, formaron una compañía alrededor de esto. Casi todos ellos siguen trabajando en NetBeans.

Tiempo después, ellos fueron contactados por Roman Stanek, un empresario que ya había estado relacionado con varias iniciativas en la Republica Checa. Él estaba buscando una buena idea en que invertir, y encontró en Xelfi una buena oportunidad. Ellos se reunieron, y el negocio surgió.

El plan original era desarrollar unos componentes JavaBeans para redes. Jarda Tulach, quien diseñó la arquitectura básica de la IDE, surgió con la idea de llamarlo NetBeans, con el fin de describir lo que ellos harían. Cuando las especificaciones de los Enterprise JavaBeans salieron, ellos decidieron trabajar con este estándar, ya que no tenía sentido competir con él, sin embargo el nombre de NetBeans se quedó.

³<http://es.wikipedia.org/wiki/NetBeans>

En la primavera de 1999, Netbeans DeveloperX2 fue lanzado, soportando Swing. Las mejoras de rendimiento que llegaron con el JDK 1.3, lanzado en otoño de 1999, hicieron a NetBeans una alternativa realmente viable para el desarrollo de herramientas. En el verano de 1999, el equipo trabajó duro para rediseñar a DeveloperX2 en un NetBeans más modular, lo que lo convirtió en la base de NetBeans hoy en día.

Algo más paso en el verano de 1999. Sun Microsystems quería una mejor herramienta de desarrollo de Java, y comenzó a estar interesado en NetBeans. En otoño de 1999, con la nueva generación de NetBeans en Beta, el acuerdo fue realizado.

Sun adquirió otra compañía de herramientas al mismo tiempo, Forté, y decidió renombrar NetBeans a Forté for Java. El nombre de NetBeans desapareció de vista por un tiempo.

Seis meses después, se tomó la decisión de hacer a NetBeans open source. Mientras que Sun había contribuido considerablemente con líneas de código en varios proyectos de código abierto a través de los años, NetBeans se convirtió en el primer proyecto de código abierto patrocinado por ellos. En Junio del 2000 NetBeans.org fue lanzado.

2.6.2 ¿Como trabaja NetBeans?

El IDE NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial llamado manifest, que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

NetBeans es patrocinado principalmente por SunMicroSystems, pero recibe sus mayores beneficios de la comunidad que colabora en sacar adelante y beneficiar al proyecto.

2.6.3 NetBeans Add-on Packs

Aunque NetBeans es un proyecto pensado para JAVA, éste proyecto se ha ido extendiendo según se le ha ido dando funcionalidad al lenguaje, al ser un proyecto modular Netbeans, permite ir agregando mas componentes que le permiten extender su funcionalidad y en la actualidad cuando descargamos NetBeans podemos ya tener integrados algunos componentes adicionales, entre los mas comunes y conocidos tenemos:

- C/C++ Native Development Module: nos permite ejecutar un programa escrito en C++.
- NetBeans Enterprise Pack: Provee Soporte para la creación de aplicaciones Orientadas a Servicios (SOA), incluyendo herramientas de esquemas XML, un editor WSDL, y un editor BPEL para web services.
- Mobile Pack: ésta es una herramienta que nos permite desarrollar aplicaciones que se ejecutan en dispositivos móviles, puede ser usado para escribir, probar, y depurar aplicaciones para la plataforma Java ME, tecnología existente en dispositivos móviles.
- Profiler de NetBeans: es una herramienta que nos ayuda a optimizar nuestras aplicaciones, encontrando cuellos de botella en la memoria y a maximizar la velocidad de las aplicaciones.
- El Visual Web Pack: incorpora soporte para construir aplicaciones web que usan componentes JSF, además da soporte para las demás tecnologías web como javascript, Jsp, Ajax.

Además de los componentes ya nombrados existe soporte para Ruby, PHP, Python, CVS, Mercurial, Subversion, JavaFX, y por sus propiedades da soporte para mas frameworks que se desarrollan como Maven, Spring, Hibernate, etc.

2.6.4 ¿Como se realiza la liberación de versiones de NetBeans?

La propuesta del proyecto NetBeans, es como en el Proyecto Ubuntu, realizar lanzamientos de sus versiones cada seis meses, además de siempre estar pensando en la nueva versión

La agenda de liberación de una nueva versión consiste en:⁴

1. Congelamiento de características: Esto se realiza después de un tiempo que no serán introducidas nuevas funcionalidades. Solo serán permitidos en el CVS cambios en el código fuente a causa de corrección de errores.

- Se crea una rama en el CVS para agregar correcciones de errores y cambios en la documentación. La convención para el nombre de esta rama será en el CVS será releaseNN donde NN es la versión pendiente a liberar.
- después del congelamiento de características, los cambios en la interfaz de usuario deben ser mínimos. Cualquier cambio pendiente en la interfaz de usuario debe ser comunicado por adelantado y solo debe hacerse para corrección de errores.
- Durante la etapa de estabilización, los binarios son marcados como NB X.Y beta. Construcciones diarias estarán disponibles para bajar y todos los usuarios están invitados a probar el software. Se espera que los desarrolladores respondan rápidamente a los reportes de errores. Los reportes de errores deben ser preferiblemente llenados en la base de datos de errores, pero los desarrolladores deben también monitorear los usuarios de las listas de distribución y responder las recomendaciones aquí escritas.
- Durante la etapa de estabilización, el README y las notas de la versión son escritas. Los primeros borradores de estos dos documentos críticos deben preferiblemente hacerlos antes que después.

2. Etapa de estabilización: En la cual los errores son localizados y arreglados.

⁴<http://es.wikipedia.org/wiki/NetBeans>

3. Release candidate 1: Si ningún problema serio es encontrado la última versión candidate a ser liberada se convierte en la versión final. Debe haber por lo menos una semana entre la última versión candidata y la versión final.

- Después que la primera versión candidate es hecha, todos los cambios en el código fuente deben ser negociados por adelantado escribiendo un post en la lista de distribución de nbdev@netbeans.org nbdev 40 netbeans.org.

4. Release candidate 2: La segunda versión candidate.

5. Final release: La final, versión pública de la nueva versión de NetBeans.

2.6.5 Versiones de NetBeans.

Tabla N° 4: Versiones y Fechas de lanzamiento de NetBeans

Versión	Fecha de Lanzamiento
NetBeans 6.7.1	27 de Julio de 2009
NetBeans 6.7	29 de Junio de 2009
NetBeans 6.5	Noviembre 25 de 2008
NetBeans 6.1	Abril 28 de 2008
NetBeans 6.0	Diciembre 3 de 2007
NetBeans 5.5.1	Mayo 24 de 2007
NetBeans 5.5	Octubre 30 de 2006

NetBeans 5.0	Enero 2006
NetBeans 4.1	Mayo 2005
NetBeans 4.0	Diciembre 2004
NetBeans 3.6	Abril 2004
NetBeans 3.5	Junio 2003

2.7 IPRROUTE2

Con la expansión del sistema operativo Linux, se ha estado intentando hacer crecer a sus posibilidades muy extensamente, y dando a ciertos módulos prioridades que no le son competentes para las que fueron creadas ej, firewall.

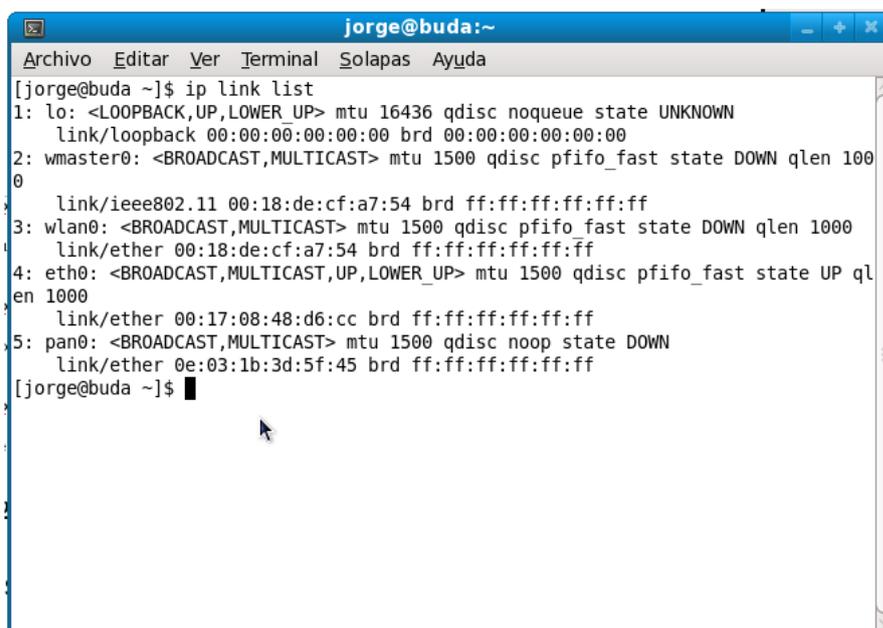
Es así que desde los kernel 2.2 y 2.4, se incluye como un paquete básico de instalación en las distribuciones a iproute2, este paquete permite integrar, varias cualidades que se encontraban sueltas, o que su funcionamiento, muestra un comportamiento extraño, entre otros se recogen a comandos como arp, ifconfig y route, es así que se vio la necesidad de incluir un subsistema de red completamente rediseñado. Este nuevo código de red proporciona a Linux un rendimiento y características con poca competencia en el panorama general de los SO. En realidad, el nuevo código de enrutado, filtrado y

clasificación tiene más posibilidades que el que porporcionan muchos enrutadores y cortafuegos dedicados y productos de control de tráfico.

2.7.1 Nuevos Comandos de iproute2

En el presente segmento vamos a ver algunos de los nuevos comandos que podemos utilizar, y que podemos usarlos en cualquier SO Linux que tenga un kernel superior a 2.2, como veremos algunos de los comandos están pensados para reemplazar a comandos que anteriormente se los utilizaba, un ejemplo es el comando **ifconfig**, su salida es igual a **ip address show**, básicamente ambos nos permiten visualizar los dispositivos de red instalados en nuestro equipo.

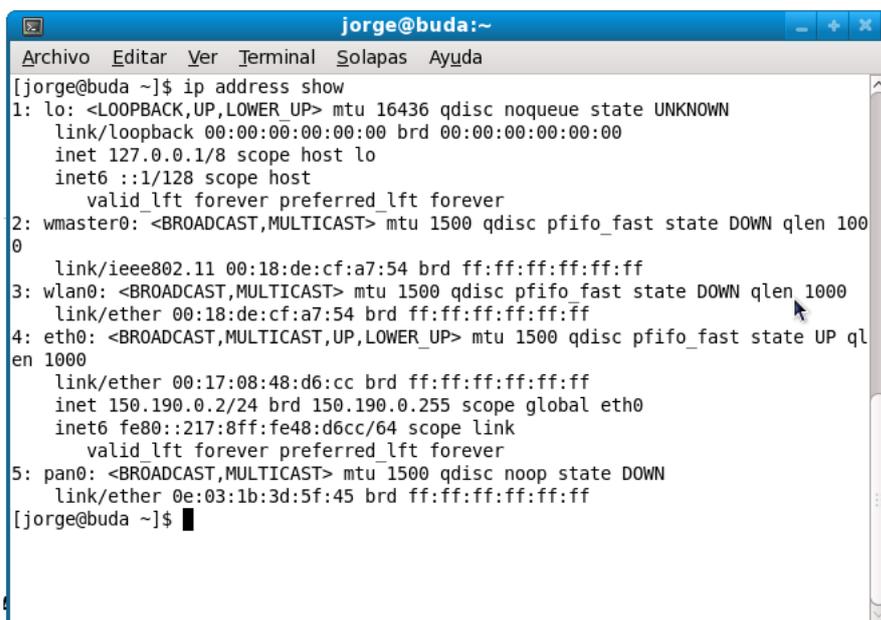
ip link list: permite ver los dispositivos existentes en nuestro equipo.

A screenshot of a terminal window titled 'jorge@buda:~'. The window contains the command 'ip link list' and its output. The output lists five network interfaces: 'lo' (loopback), 'wmaster0' (wireless master), 'wlan0' (wireless LAN), 'eth0' (ethernet), and 'pan0' (powerline). Each interface entry includes its name, state, MTU, queue discipline, and MAC address.

```
[jorge@buda ~]$ ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: wmaster0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
    link/ieee802.11 00:18:de:cf:a7:54 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
    link/ether 00:18:de:cf:a7:54 brd ff:ff:ff:ff:ff:ff
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:17:08:48:d6:cc brd ff:ff:ff:ff:ff:ff
5: pan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
    link/ether 0e:03:1b:3d:5f:45 brd ff:ff:ff:ff:ff:ff
[jorge@buda ~]$
```

Figura N° 10: Captura del comando ip link list

ip address show: permite visualizar la configuración ip de los dispositivos eth en nuestro equipo.



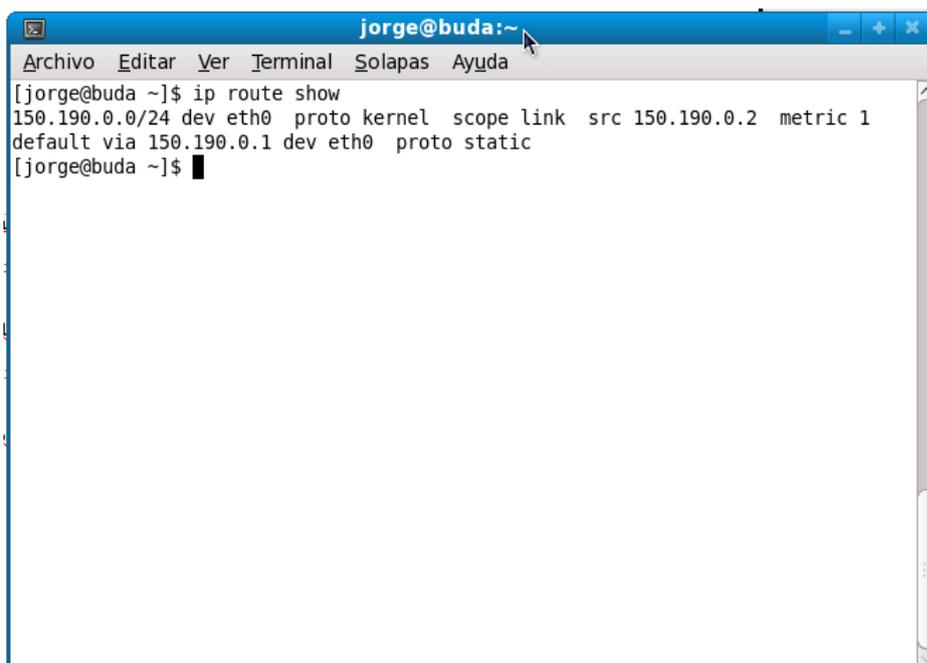
```

jorge@buda:~
Archivo Editar Ver Terminal Solapas Ayuda
[jorge@buda ~]$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: wmaster0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
   link/ieee802.11 00:18:de:cf:a7:54 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
   link/ether 00:18:de:cf:a7:54 brd ff:ff:ff:ff:ff:ff
4: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
   link/ether 00:17:08:48:d6:cc brd ff:ff:ff:ff:ff:ff
   inet 150.190.0.2/24 brd 150.190.0.255 scope global eth0
   inet6 fe80::217:8ff:fe48:d6cc/64 scope link
       valid_lft forever preferred_lft forever
5: pan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
   link/ether 0e:03:1b:3d:5f:45 brd ff:ff:ff:ff:ff:ff
[jorge@buda ~]$

```

Figura N° 11: Captura del comando ip address show

ip route show: nos muestra como se está realizando el ruteo desde nuestro equipo, para poder salir al mundo.



```

jorge@buda:~
Archivo Editar Ver Terminal Solapas Ayuda
[jorge@buda ~]$ ip route show
150.190.0.0/24 dev eth0 proto kernel scope link src 150.190.0.2 metric 1
default via 150.190.0.1 dev eth0 proto static
[jorge@buda ~]$

```

Figura N° 12: Captura del comando ip route show

2.7.2 CBQ

Las qdisc con clases son muy útiles si tiene diferentes tipos de tráfico a los que quiere dar un tratamiento separado. Una de las qdisc con clases se denomina «CBQ», «Class Based Queueing», y se la menciona tan frecuentemente que la gente suele identificar el encolado con clases sólo con CBQ, pero éste no es el caso.⁵

CBQ es meramente la que lleva más tiempo (y también la más compleja). No siempre es la que necesita. Esto puede ser un trauma para los partidarios del «efecto sendmail», que nos enseña que cualquier tecnología compleja que no viene documentada debe ser lo mejor que hay disponible.

Cuando deseamos utilizar o iniciar a configurar CBQ puede ser un poco intimidante, especialmente si todo lo que deseamos es ajustar el tráfico de algunos computadores de nuestra red hacia un router. CBQ.init puede ayudarnos a configurar Linux con una sintaxis simplificada, fácil de entender y utilizar además CBQ usa de una forma simplificada los mandatos **ip** y **tc** para su funcionamiento, así como algoritmos complejos que nos ayudan a controlar nuestra red.

2.7.3 Parámetros de CBQ

2.7.3.1 Parámetro DEVICE

Es un parámetro obligatorio. Se determina los valores con el nombre de la interfaz, ancho de banda y peso de esta interfaz. Este último valor, que es opcional en este parámetro, se calcula dividiendo el ancho de banda de la interfaz entre diez.

Si se dispone de una interfaz eth0 conectada a un modem ADSL de 2048 kbps de tráfico entrante o de bajada, el peso será de 204 kbps.

2.7.3.2 Parámetro de clase RATE

Es un parámetro obligatorio. Se refiere al ancho de banda a asignar a la clase. El tráfico que pase a través de esta clase será modificado para ajustarse a la proporción definida.

⁵ Linux Advanced Routing & Traffic Control HOWTO , BERT Hubert, Netherlabs BV

2.7.3.3 Parámetro de clase **PRIO**

Es un parámetro opcional que se utiliza para especificar que prioridad tendrá sobre otras reglas de control de ancho de banda. Mientras más alto sea el valor, menos prioridad tendrá sobre otras reglas. Se recomienda utilizar el valor 5 que funcionará para la mayoría de los casos.

2.7.3.4 Parámetro de clase **PARENT**

Cuando se utilizan reglas que se requiere estén jerarquizadas, se utiliza para establecer la identidad de clase padre a la que pertenecen. Puede llevar cualquier valor. Cuando se trata de una clase padre, se define junto con el parámetro **LEAF** con el valor **none**. En el siguiente ejemplo se establece la identidad 100 en una clase padre.

2.7.3.5 Parámetro de clase **LEAF**

Es un parámetro opcional y se utiliza para determinar que política se utilizará para utilizar el ancho de banda de una clase padre. Si se utiliza el valor **tbf**, que es el valor predeterminado, se utilizará el algoritmo **TBF (Token Bucket Filter)**, el cual impide que la clase tome ancho de banda de la clase padre, los valores que podemos hacer uso son:

Tabla N° 4: Opciones del parámetro LEAF

BUFFER	<p>Determina el tamaño máximo de ráfaga (<i>maximal burst size</i>) que la clase puede enviar, y puede llevar como parámetro opcional la longitud de los intervalos en bytes. El valor predeterminado es 10Kb/8. es decir, ráfagas de 10Kb en intervalos de 8 bytes.</p> <p>LEAF=tbf BUFFER=10Kb/8</p>
LIMIT	<p>Determina el tamaño máximo de las reservas (<i>backlog</i>). Si la cola de datos por procesar contiene más de los especificados por LIMIT, los siguientes paquetes que lleguen serán descartados. La longitud de las reservas determina la latencia (tiempo de recuperación de datos) de la cola en caso de presentarse una congestión. El valor predeterminado es 15kb.</p> <p>LEAF=tbf LIMIT=15kb</p>

PEAK	<p>Determina el pico máximo para una ráfaga de tráfico de corto plazo que una clase puede enviar. Considerando que un ancho de banda de 256 kbps envía 256 Kbit por segundo, en un momento dado se puede dar el caso de el envío de 512 Kbit en 0.50 segundos, o 1 Mbit en 0.25 segundos. En el siguiente ejemplo se establece el el pico máximo para ráfagas de 1024 Kbit:</p> <p>LEAF=tbf PEAK=1024Kbit</p>
MTU	<p>Determina la máxima cantidad de datos que se pueden enviar al mismo tiempo en un medio físico. Es un parámetro obligatorio si se utiliza el parámetro PEAK. En el caso de una interfaz <i>Ethernet</i>, el valor predeterminado es igual al MTU de la propia interfaz (1500).</p> <p>LEAF=tbf PEAK=1024Kbit MTU=1500</p>

El valor **sfq**, que corresponde al algoritmo **SFQ** (Stochastic Fairness Queueing), hace que sea compartido el ancho de banda de la clase padre aproximadamente en la **misma** proporción de ancho de banda entre anfitriones dentro de la misma clase. El valor **none** permite utilizar libremente el ancho de banda disponible, siempre que el valor del parámetro **BOUDED** sea igual a **no**.

2.7.3.6 Parámetro de clase **BOUNDED**

Es un parámetro opcional. Si el valor es **yes**, que es el valor predeterminado, la clase no tendrá permitido utilizar ancho de banda de la clase padre. Si el valor es **no**, la clase podrá hacer uso del ancho de banda disponible en la clase padre. Si se establece con valor **no**, es necesario utilizar **none** o bien **sfq** en el parámetro **LEAF**.

2.7.3.7 Parámetro de clase **ISOLETED**

Es un parámetro opcional. Si se establece con el valor **yes**, la clase no prestará ancho de banda a las clases hijas. Si se utiliza el valor **no**, que es el valor predeterminado, se permitirá prestar el ancho de banda disponible a las clases hijas.

2.7.3.8 Parámetros de filtración

Son las reglas de filtración que se utilizan para seleccionar tráfico en cada una de las clases.

En general la interpretación sigue cuatro simples principios:⁶

- Cualquier dirección IP o red que se coloque **antes de la coma** se considera dirección IP o red de **origen**.
- Cualquier dirección IP o red que se coloque **después de la coma** se considera dirección IP o red de **destino**.
- Cualquier puerto **antes de la coma** se considera el puerto de **origen**.
- Cualquier puerto especificado **después de la coma** se considera puerto de **destino**.

⁶<http://www.alcancelibre.org/staticpages/index.php/como-cbq>

CAPÍTULO V

DESARROLLO DE LA

PROPUESTA ALTERNATIVA

1 ANÁLISIS Y DISEÑO DEL PROYECTO DE TESIS

1.1 REQUERIMIENTOS DE SOFTWARE

1.1.1 REQUERIMIENTOS FUNCIONALES

Gestión de Administrador

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
RF001	Ingreso a la herramienta mediante un nombre de usuario y una clave	Evidente
RF002	Validar ingreso del usuario a la herramienta	Sistema

Gestión de Configuración por Defecto

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
RF003	Ingresar datos de configuración por defecto.	Evidente
RF004	Validar datos ingresados en los campos	Sistema
RF005	Actualizar datos de configuración por defecto.	Sistema

Gestión de Dispositivos de Red

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
------	-------------	--------

RF006	Crear interfaces de red .	Evidente
RF007	Validar datos de dispositivos.	Sistema
RF008	Presentar datos de dispositivos existentes	Evidente
RF009	Cargar datos de dispositivos.	Evidente
RF010	Actualizar datos de dispositivo	Evidente
RF011	Eliminar dispositivos que no corresponden.	Evidente

Gestión de Direcciones Ip de Equipos a Controlar

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
RF012	Crear direcciones ip.	Evidente
RF013	Validar datos de direcciones ip.	Sistema
RF014	Presentar datos de direcciones ip.	Evidente
RF015	Cargar datos de direcciones ip.	Evidente
RF016	Actualizar datos de direcciones ip.	Evidente
RF017	Eliminar direcciones ip que ya no se deseen controlar	Evidente

Gestión de Direcciones Ip de Subredes

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
------	-------------	--------

RF018	Crear direcciones ip.	Evidente
RF019	Validar datos de direcciones ip.	Sistema
RF020	Presentar datos de direcciones ip.	Evidente
RF021	Cargar datos de direcciones ip.	Evidente
RF022	Actualizar datos de direcciones ip.	Evidente
RF023	Eliminar direcciones ip que ya no se controlen	Evidente

Gestión de Relaciones de Scripts

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
RF024	Crear relaciones de scripts.	Evidente
RF025	Validar que el script tenga una única relación.	Sistema
RF026	Presentar datos de relaciones.	Evidente
RF027	Eliminar relaciones de scripts.	Evidente

Gestión de Generación de Scripts

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
RF028	Crear scripts de control.	Evidente
RF029	Validar datos del script.	Sistema
RF030	Presentar scripts.	Evidente

RF031	Cargar datos de script.	Evidente
RF032	Actualizar datos de script.	Evidente
RF033	Crear regla de control de script.	Evidente
RF034	Validar regla del script.	Sistema
RF035	Presentar reglas del script.	Evidente
RF036	Eliminar relaciones de scripts.	Evidente
RF037	Eliminar reglas que pertenecen a script	Sistema - Evidente
RF038	Eliminar script	Evidente

Gestión de Monitoreo.

La herramienta permitirá:

CÓD.	DESCRIPCIÓN	ESTADO
RF039	Calcular los datos de direcciones de red.	Sistema
RF040	Escanear los puertos de los equipos	Sistema

1.1.2 REQUERIMIENTOS NO FUNCIONALES.

CÓD.	DESCRIPCIÓN
RF001	Usare como lenguaje de programación JSP
RF002	Usaré como IDE NetBeans 6.5

RF003	La herramienta utilizará a MySQL 5.0 como base de datos.
RF004	La herramienta utilizará a Hibernate 3.0 y Spring MVC 2.5 como frameworks.
RF005	La herramienta contará con una interfaz de fácil uso.
RF006	La herramienta contará con documentación para su fácil actualización .
RF007	Para la creación de la herramienta se hará uso únicamente de programas open source.

1.2 MODELAMIENTO DE LOS CASOS DE USO

La herramienta permitirá:

ACTORES	META	UC	
Administrador – Monitor	Cargar datos de configuración por defecto	Administrar configuración por defecto	CD
Administrador – Monitor	Ingreso a la herramienta mediante un nombre de usuario y una clave	Ingresar al sistema	I
Administrador	Administrar los datos de la configuración por defecto	Administrar configuración por defecto	CD
	Crear una nueva interfaz de red	Administrar dispositivos de red.	ADR

<p>Validar que los datos del dispositivo de red que se desea crear no se encuentren existentes</p> <p>Presentar datos de dispositivos existentes</p> <p>Seleccionar un dispositivo de red para modificar</p> <p>Actualizar datos de dispositivo</p> <p>Eliminar dispositivos de red que ya no se utilicen.</p>		
<p>Crear una Ip perteneciente a un equipo de la red</p> <p>Validar que los datos de la Ip que se desea crear no se encuentren existentes</p> <p>Presentar Ips existentes</p> <p>Seleccionar una Ip de red para modificar</p> <p>Actualizar datos de Ip seleccionada</p> <p>Eliminar ip que ya no se</p>	<p>Administrar Ip de equipos</p>	<p>AIE</p>

desea controlar.		
<p>Crear una Ip perteneciente a una subred</p> <p>Validar que los datos de la Ip que se desea crear no se encuentren existentes</p> <p>Presentar Ips existentes</p> <p>Seleccionar una Ip de red para modificar</p> <p>Actualizar datos de Ip seleccionada</p> <p>Eliminar ip que ya no se desea controlar.</p>	Administrar Ip de subred	AIR
<p>Crear script que permita realizar control de ancho de banda.</p> <p>Validar que los datos del script, no se repitan.</p> <p>Presentar scripts que se usarán para el control.</p> <p>Seleccionar un script para poder modificar sus datos.</p>	Administrar Scripts	AS

<p>Actualizar datos de script seleccionado.</p> <p>Seleccionar un script e ingresar las reglas que usará.</p> <p>Validar que la regla no se repita en los script.</p> <p>Eliminar regla de un script</p> <p>Eliminar el script y sus reglas.</p> <p>Seleccionar scripts que se van a relacionar para compartir banda.</p> <p>Validar que el script se relacione solo una ves.</p> <p>Presentar las relaciones entre scripts.</p> <p>Eliminar relaciones de scripts.</p>		
---	--	--

1.3 REQUERIMIENTOS

Para iniciar a trabajar con la herramienta es necesario tener instalado un servidor que interprete las páginas jsp, puedes leer la sección **SERVICIO WEB**, en este caso puede ser Tomcat, GlassFish o Jboss, en cualquiera de los casos que selecciones básicamente se necesita:

Instalar y configurar java 1.6.x: <http://www.java.com/es/download/>

Instalar y configurar MySQL 5.0.x:

<http://dev.mysql.com/downloads/mysql/5.1.html>

El servidor seleccionado:

- Tomcat 6.0.x: <http://tomcat.apache.org/>
- GlassFish: <https://glassfish.dev.java.net/>
- Jboss: <http://jboss.org/jbossas/downloads/>

Como hardware para la instalación es recomendable:

Memoria: 128MB o 256MB

Procesador de 750 MHz o 1 Ghz

Disco Duro: 60 Gb

Sistema operativo que recomiendo es cualquier distribución de Linux, cuyo kernel sea una mayor a 2.4.

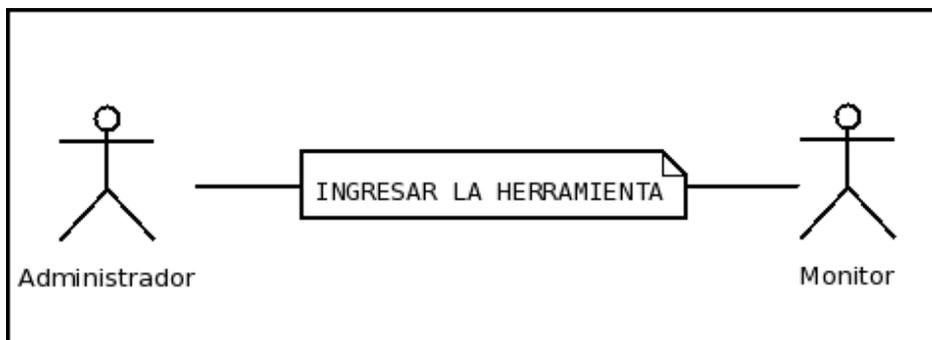
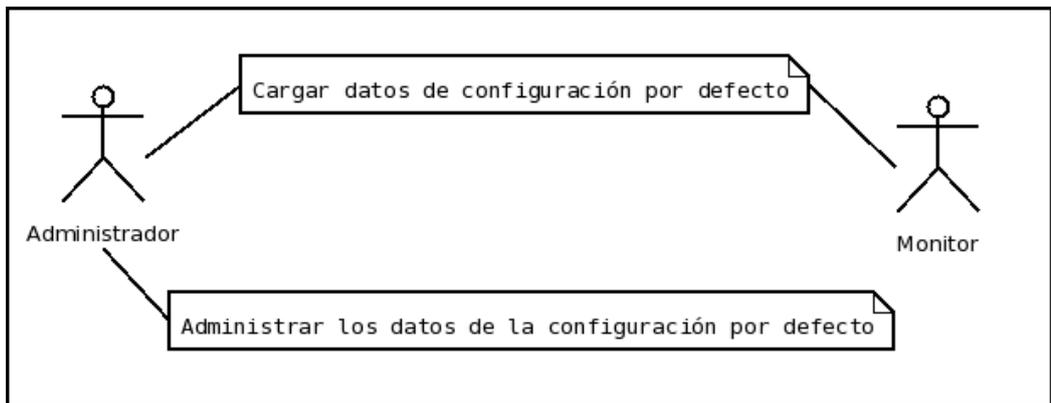
Una vez que tienes instalado todo esto solo se debe crear la base de datos en MySQL que se llame *bdcwr*, y crear un usuario que se llame *cwr* cuya clave sea *cwr* y tenga permisos de acceso local, además se debe copiar el archivo *cwr.war* en la carpeta del servidor, e iniciar a trabajar con la herramienta.

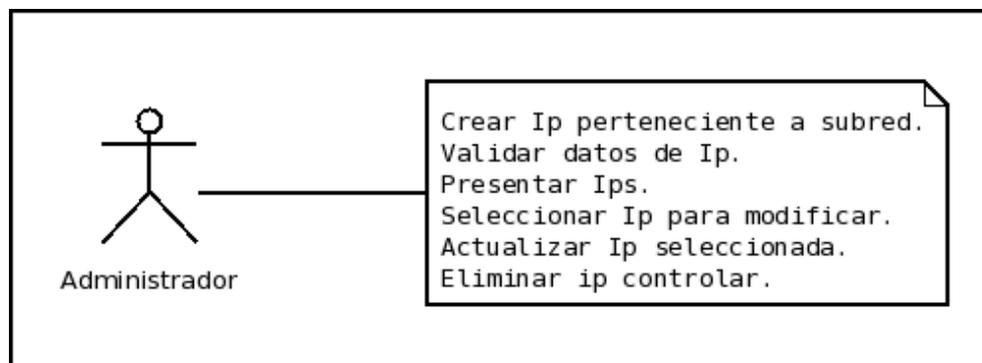
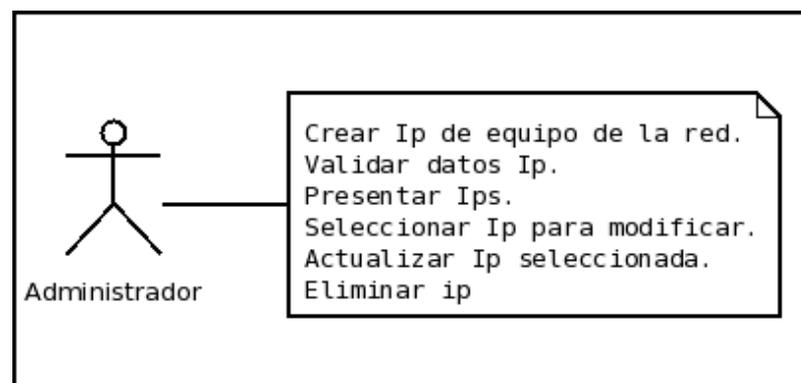
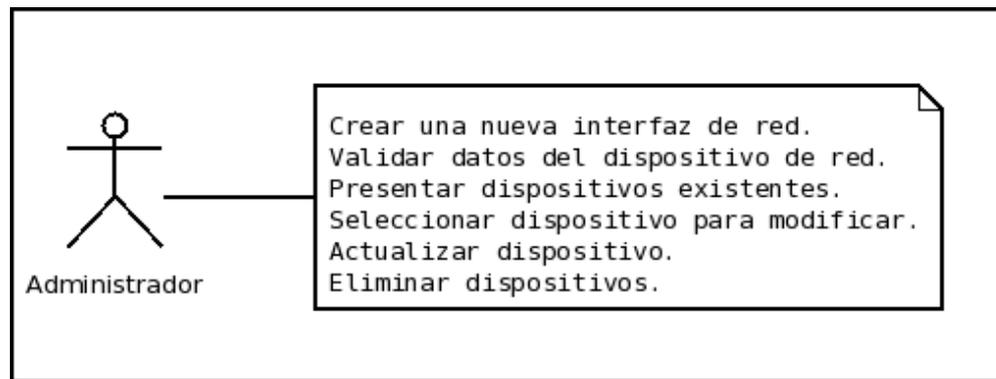
En caso de que quieras cambiar el nombre de la base de datos o el usuario con el que se realiza la conexión lo puedes hacer modificando los datos del archivo *jdbc.properties* que se encuentra en el proyecto, para poder abrir y recompilarlo es necesario que tengas

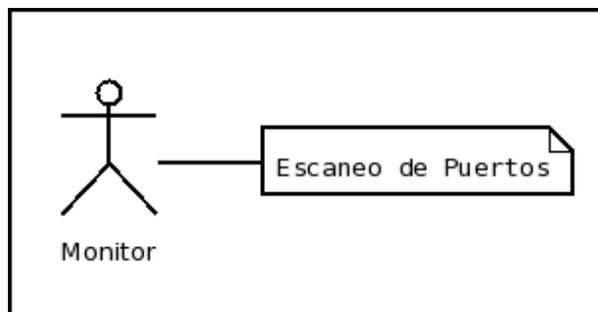
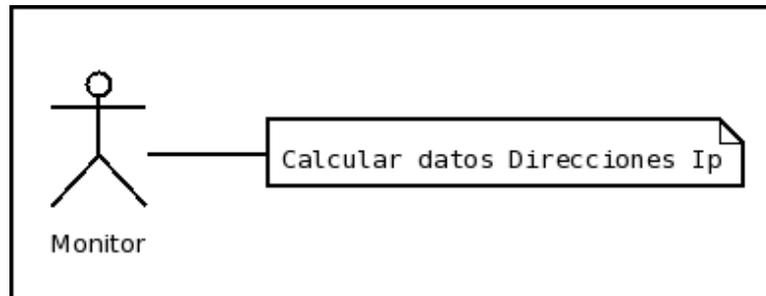
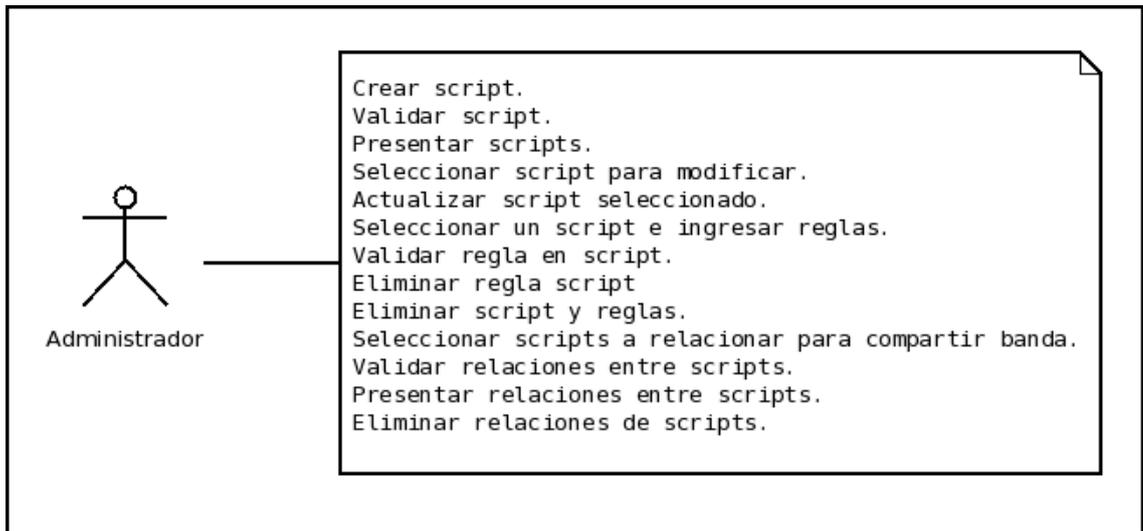
instalado NetBeans 6.5, si deseas cambiar los mensajes que se ven en las páginas puedes modificar el archivo messages.properties.

Una vez que tienes todo esto listo solo debes ejecutar la aplicación no es necesario ejecutar ningún otro script ya que las tablas de la base de datos y sus relaciones son creadas automáticamente por la herramienta, bueno este trabajo se lo delego a hibernate, otro de los frameworks que se usaron en el desarrollo de la aplicación, luego de eso puedes continuar trabajando con normalidad.

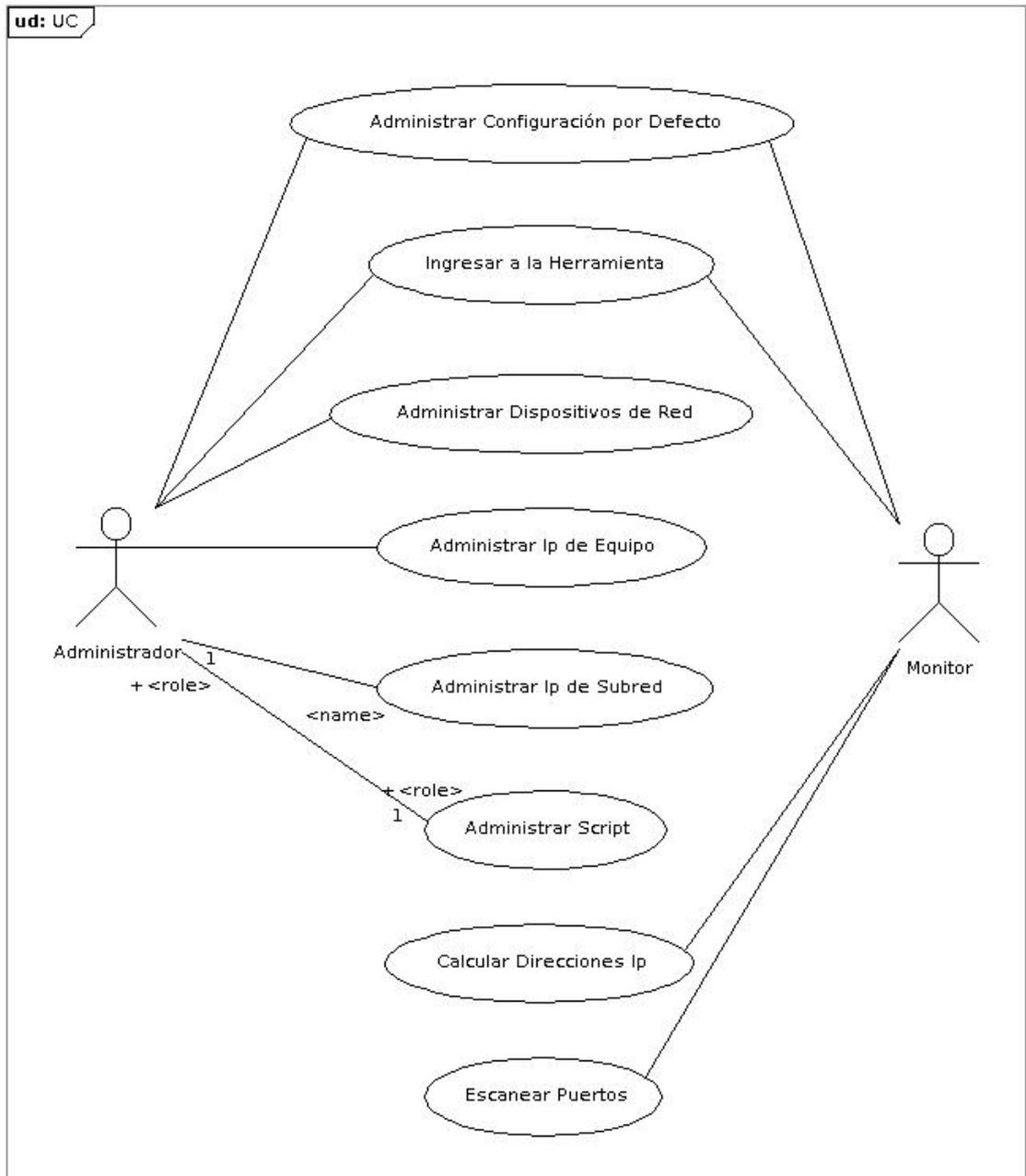
1.4 ORGANIZAR LOS CASOS DE USO







1.5 DIAGRAMA DE CASOS DE USO



1.6 PROTOTIPO DE PANTALLAS

Nombre Vista: index.htm
Referencia Requerimientos: RF003
Caso de Uso: Administrar Configuración por defecto





Inicio
Acceso

Iniciemos con el control de redes!

viernes, 16 octubre 2009, 16:21
Escrito por Jorge Guerrón Eras

Como lo indica el nombre, este proyecto lo que nos va a permitir es realizar varios controles sobre nuestro servidor, configurar mediante `cbq`, para realizar la segmentación nuestro ancho de banda, mediante la calculadora `ip` que se incluye analizar los diferentes tipos de redes que se pueden configurar, y todas esto comprobarlo mediante un cliente sniffer a nuestra elección, además verificar los puertos del servidor mediante un escaner de puertos.

Como inicio?

Iniciamos creando la base de datos y el usuario que necesitamos para realizar la conexión, este código te va a ayudar:

1. `create database bdcwr;`
2. `grant all privileges on bdcwr.* to cwr@'localhost';`
3. `flush privileges;`
4. `set password for 'cwr'@'localhost' = password('cwr');`

luego, da un click [aquí](#)

Ten en cuenta este apartado

Este software lo dejo libre para que cumpla las cuatro libertades que debe cumplir todo software libre.

Libertades del Software Libre

1. "La libertad de usar el programa, con cualquier propósito"
2. "La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades"
3. "La libertad de distribuir copias, con lo que puedes ayudar a tu vecino"
4. "La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie"

Acceso

- Acceso Usuarios

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#); Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: start.htm

Referencia Requerimientos: RF005

Caso de Uso: Administrar Configuración por Defecto



Nombre Vista: error.htm
Referencia Requerimientos: RF005
Caso de Uso: Administrar Configuración por Defecto

Error
página de error

Inicio

La información por defecto ya ha sido cargada, si deseas iniciar nuevamente sigue éstos pasos:

sábado, 17 octubre 2009, 21:29
Escrito por Jorge Guerrón Eras

1. *Saca un backup de la base de datos. (para respaldar tu información).*
2. *Realiza un drop de la base de datos, drop database bdcwr;*
3. *Crear nuevamente la base de datos, create database bdcwr;*
4. *Carga nuevamente los datos por defecto;*
5. *Del backup que sacaste, ve subiendo de la información que necesitas de cada tabla, no todo el script.*

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#); Imágenes [Zero Signs tonozero@hotmail.com](#), Diseño: [Free CSS Templates](#)

Nombre Vista: inadm.htm

Referencia Requerimientos: RF001

Caso de Uso: Ingresar a la Herramienta

CWR
control web de redes

Inicio

Sección de Administrador

martes, 13 octubre 2009, 15:04
Escrito por Jorge Guerrón Eras

Esta sección está destinada al administrador, va a poder ingresar nuevos datos de red, ips, agregar y eliminar controles, en fin administrar de manera eficiente los recursos de la red

Menú Usuarios:

Usuario:

Clave:

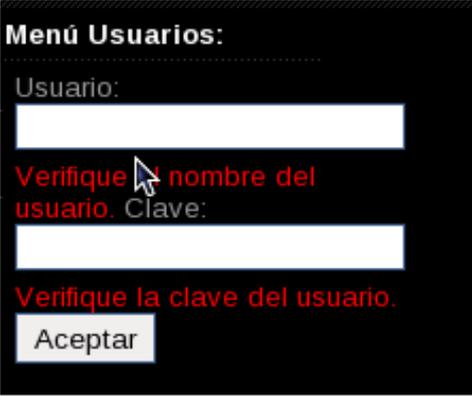
Aceptar

Sección de Monitoreo

martes, 13 octubre 2009, 15:04
Escrito por Jorge Guerrón Eras

Esta sección está destinada al monitor, va a poder visualizar dos applets que les van a ayudar a conocer como trabajan las direcciones de red, y además algunos consejos de utilidad.

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#), Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: inadm.htm
Referencia Requerimientos: RF002
Caso de Uso: Ingresar a la herramienta


Nombre Vista: general.htm
Referencia Requerimientos: RF002
Caso de Uso: Ingreso a la herramienta

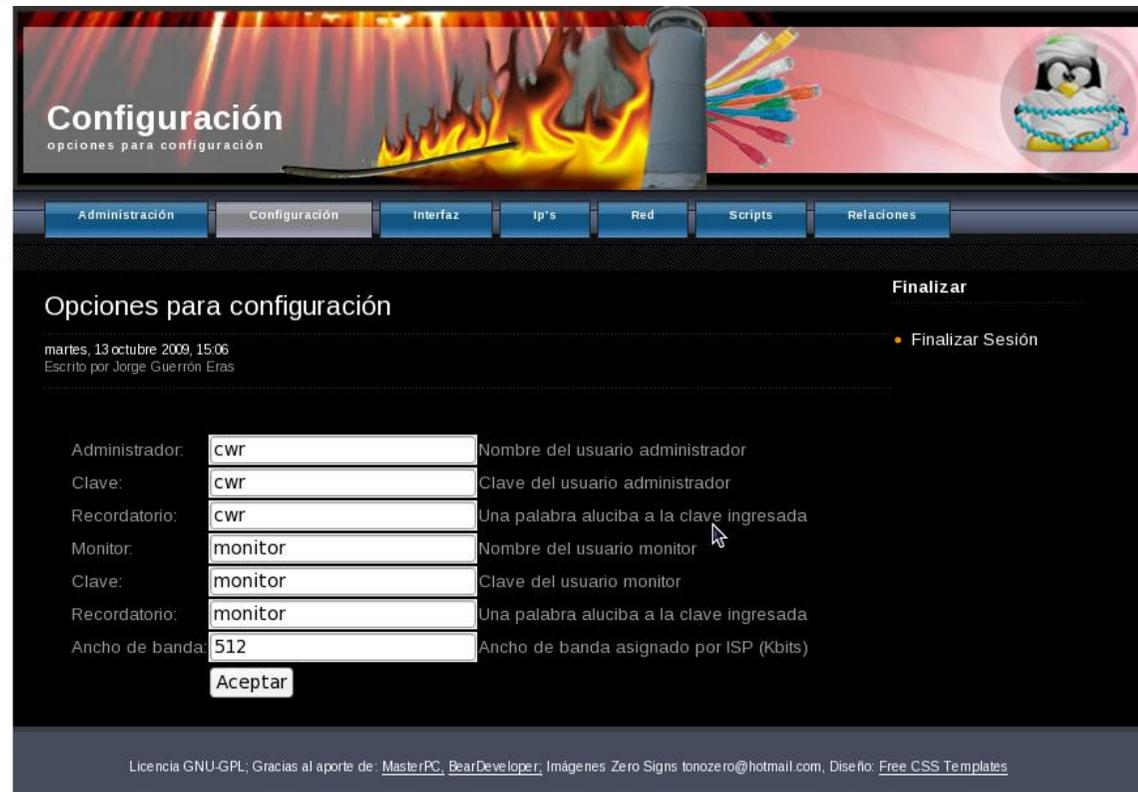


The screenshot shows a web administration interface. At the top, there is a banner with the word "Administración" and the subtitle "opciones del administrador". The banner features a background of flames, a server tower, and a penguin character. Below the banner, there is a navigation bar with a button labeled "Administración". The main content area is titled "Opciones del Administrador" and includes the date "martes, 13 octubre 2009, 15:06" and the author "Escrito por Jorge Guerrón Eras". A "Menú Usuarios:" section contains a "Finalizar Sesión" link. A list of options is provided: "1. Configuración", "2. Interfaz", "3. Ip's", "4. Red", "5. Scripts", and "6. Relaciones". The footer contains a license notice: "Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#); Imágenes: Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)".

Nombre Vista: config.htm

Referencia Requerimientos: RF003, RF005

Caso de Uso: Administrar Configuración por Defecto



The screenshot shows a web interface for configuration management. At the top, there is a banner with the title "Configuración" and the subtitle "opciones para configuración". Below the banner is a navigation menu with tabs for "Administración", "Configuración", "Interfaz", "Ip's", "Red", "Scripts", and "Relaciones". The "Configuración" tab is selected. The main content area is titled "Opciones para configuración" and includes a "Finalizar" button. The form contains the following fields and labels:

Administrador:	<input type="text" value="cwr"/>	Nombre del usuario administrador
Clave:	<input type="text" value="cwr"/>	Clave del usuario administrador
Recordatorio:	<input type="text" value="cwr"/>	Una palabra aluciba a la clave ingresada
Monitor:	<input type="text" value="monitor"/>	Nombre del usuario monitor
Clave:	<input type="text" value="monitor"/>	Clave del usuario monitor
Recordatorio:	<input type="text" value="monitor"/>	Una palabra aluciba a la clave ingresada
Ancho de banda:	<input type="text" value="512"/>	Ancho de banda asignado por ISP (Kbits)

Below the form is an "Aceptar" button. In the top right corner of the main content area, there is a "Finalizar" button with a sub-item "Finalizar Sesión". At the bottom of the page, there is a footer with the text: "Licencia GNU-GPL; Gracias al aporte de: MasterPC, BearDeveloper; Imágenes Zero Signs tonozero@hotmail.com, Diseño: Free CSS Templates".

Nombre Vista: config.htm

Referencia Requerimientos: RF004**Caso de Uso: Administrar Configuración por Defecto**

Opciones para configuración

martes, 13 octubre 2009, 15:10
Escrito por Jorge Guerrón Eras

Administrador:	<input type="text"/>	Nombre del usuario administrador
	Verifique el nombre del usuario.	
Clave:	<input type="text"/>	Clave del usuario administrador
	Verifique la clave del usuario.	
Recordatorio:	<input type="text"/>	Una palabra aluciba a la clave ingresada
	Ingrese un dato.	
Monitor:	<input type="text"/>	Nombre del usuario monitor
	Verifique el nombre del usuario.	
Clave:	<input type="text"/>	Clave del usuario monitor
	Verifique la clave del usuario.	
Recordatorio:	<input type="text"/>	Una palabra aluciba a la clave ingresada
	Ingrese un dato.	
Ancho de banda:	<input type="text"/>	Ancho de banda asignado por ISP (Kbits)
	Failed to convert property value of type [java.lang.String] to required type [int] for property bw, nested exception is java.lang.NumberFormatException: Zero length string	
	<input type="button" value="Aceptar"/>	

Nombre Vista: eth.htm

Referencia Requerimientos: RF006, RF008, RF009, RF010, RF011

Caso de Uso: Administrar Dispositivos de Red

Interfases del Servidor
dispositivos disponibles

Administración Configuración Interfaz Ip's Red Scripts Relaciones

Dispositivos disponibles Finalizar

martes 13 octubre 2009, 15:12
Escrito por Jorge Guerrón Eras

Finalizar Sesión

Interfaz: Nombre de la tarjeta de red

Dirección Ip: Dirección ip

Máscara de Red: Máscara de red

Puerta de Enlace: Puerta de enlace predeterminada

Detalle: Descripción de la tarjeta de red

Peso: El ancho de banda total de la interfaz

Guardar Actualizar

Interfases de Red Creadas

Listado de las interfaces existentes en el servidor, éstas son creadas por el administrador.

Refrescar

	Interfaz	Ip	Máscara de Red	Puerta de Enlace	Detalle	Peso		
1	eth0	192.168.0.24	24	192.168.0.1	salida a internet prueba	1024	Cargar	Borrar
7	eth1	150.190.0.1	24	192.168.0.24	eth para red interna	1024	Cargar	Borrar

Licencia GNU-GPL. Gracias al aporte de: [MasterPC](#), [BearDeveloper](#), [Imágenes Zero Signs](#) konozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: eth.htm

Referencia Requerimientos: RF007

Caso de Uso: Administrar Dispositivos de Red

Dispositivos disponibles

martes 13 octubre 2009, 15:14
Escrito por Jorge Guerrón Eras

Interfaz:	<input type="text"/>	Nombre de la tarjeta de red
Dirección Ip:	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	Dirección ip
Máscara de Red:	<input type="text"/>	Máscara de red
Puerta de Enlace:	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	Puerta de enlace predertminada
Detalle:	<input type="text"/>	Descripción de la tarjeta de red
Peso:	<input type="text"/>	El ancho de banda total de la interfaz

Failed to convert property value of type [java.lang.String] to required type [int] for property cidr; nested exception is java.lang.NumberFormatException: Zero length string

Failed to convert property value of type [java.lang.String] to required type [int] for property weigth; nested exception is java.lang.NumberFormatException: Zero length string

Ingrese un dato.

Ingrese un dato.

Nombre Vista: ip.htm

Referencia Requerimientos: RF012, RF014, RF015, RF016, RF017

Caso de Uso: Administrar Ip de Equipo

Equipos de la red
ips de equipos disponibles

Administración Configuración Interfaz Ip's Red Scripts Relaciones

Ips de equipos disponibles Finalizar

martes 13 octubre 2009, 15:15
Escrito por Jorge Guerrero Eras Finalizar Sesión

Dirección Ip: Dirección ip
Máscara de Red: Máscara de red
Detalle: Descripción de la tarjeta de red

Ips, de nuestra red.

Listado de ips agregadas por el administrador, éstas ips pertenecen a equipos individuales y van a ser controladas, permitiendo o negando el acceso, y además cuanto pueden consumir de banda.

	Ip	Máscara de Red	Detalle		
1	150.190.0.1	24	Servidor	<input type="button" value="Cargar"/>	<input type="button" value="Borrar"/>
2	150.190.0.2	24	Portatil	<input type="button" value="Cargar"/>	<input type="button" value="Borrar"/>
3	150.190.0.3	24	Desarrollo 1	<input type="button" value="Cargar"/>	<input type="button" value="Borrar"/>

Licencia GNU-GPL. Gracias al aporte de: [MasterPC_BearDeveloper](#); Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: ip.htm

Referencia Requerimientos: RF017

Caso de Uso: Administrar Ip de Equipo

Ips de equipos disponibles

martes 13 octubre 2009, 15:16

Escrito por Jorge Guerrón Eras

Dirección Ip:

Dirección ip

Máscara de Red:

Máscara de red

Debe especificar un número mayor a 0!

Detalle:

Descripción de la tarjeta de red

Ingrese un dato.

Guardar

Actualizar

Nombre Vista: net.htm
Referencia Requerimientos: RF018, RF020, RF021, RF022, RF023
Caso de Uso: Administrar Ip de Subred

Subredes existentes!
ips de las subredes disponibles

Administración Configuración Interfaz Ip's Red Scripts Relaciones

Ips de las subredes disponibles Finalizar

Finalizar Sesión

martes 13 octubre 2009, 15:18
Escrito por Jorge Guerrón Eras

Dirección ip: Dirección ip

Máscara de Red: Máscara de red

Detalle: Descripción de la tarjeta de red

Guardar Actualizar

Subredes, de nuestra red.

Listado de ips agregadas por el administrador, éstas ips pertenecen a subredes que pertenecen a nuestra red y van a ser controladas, permitiendo o negando el acceso, y además cuanto pueden consumir de banda. Refrescar

	Ip	CIDR	Detalle		
4	150.190.0.0	24	Red Software	Cargar	Borrar

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#); Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: net.htm

Referencia Requerimientos: RF019

Caso de Uso: Administrar Ip de Subred

Ips de las subredes disponibles

.....

martes 13 octubre 2009, 15:17
Escrito por Jorge Guerrón Eras

.....

Dirección Ip:	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	Dirección ip
Máscara de Red:	<input type="text" value="33"/>				Máscara de red
Detalle:	<input type="text"/>				Descripción de la tarjeta de red

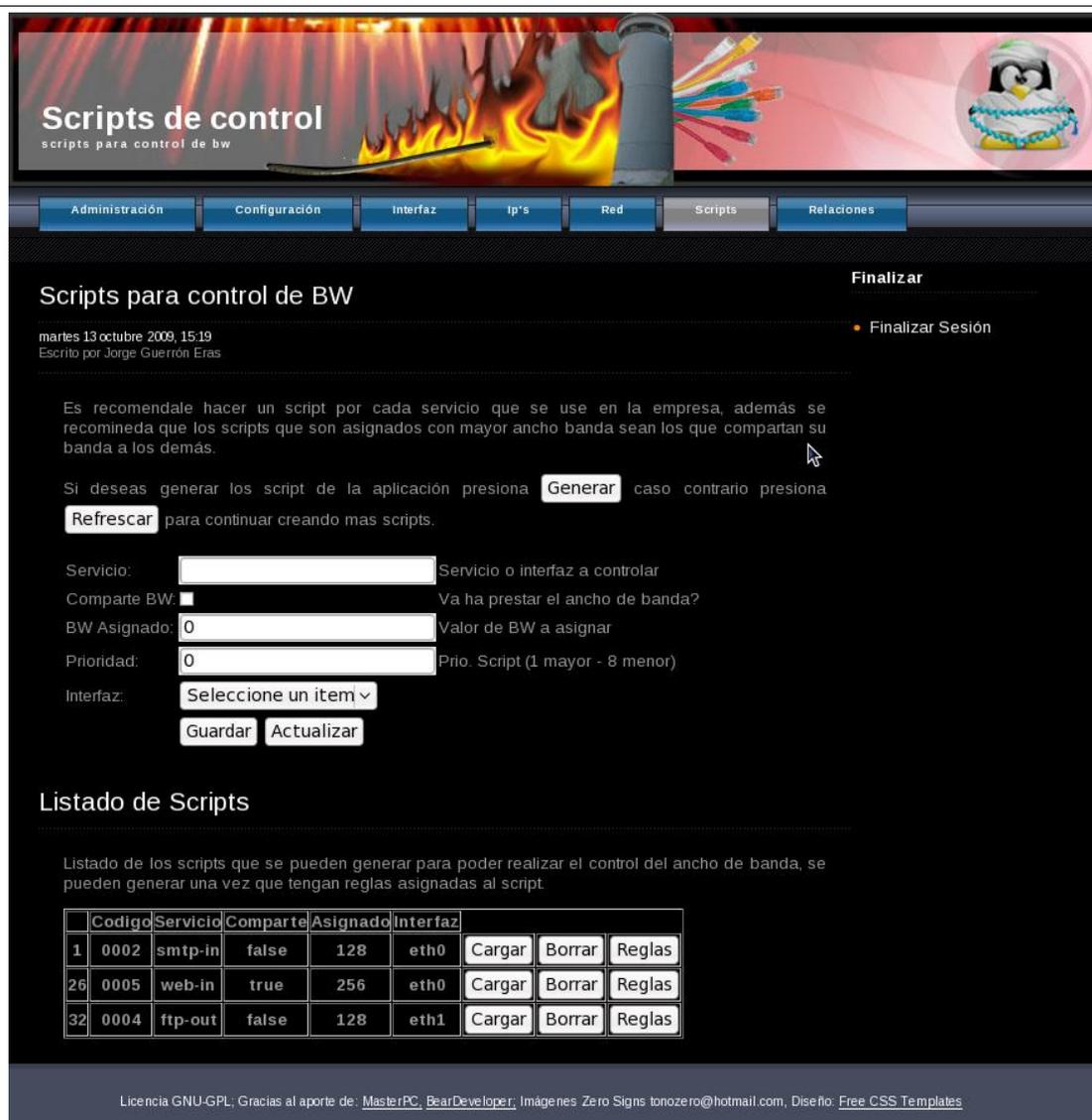
Debe especificar un numero menor a 32!

Ingrese un dato.

Nombre Vista: file.htm

Referencia Requerimientos: RF028,RF030,RF031,RF032,RF038

Caso de Uso: Administrar Script



Scripts de control
scripts para control de bw

Administración Configuración Interfaz Ip's Red Scripts Relaciones

Scripts para control de BW Finalizar

martes 13 octubre 2009, 15:19
Escrito por Jorge Guerrón Eras

• Finalizar Sesión

Es recomendable hacer un script por cada servicio que se use en la empresa, además se recomienda que los scripts que son asignados con mayor ancho banda sean los que compartan su banda a los demás.

Si deseas generar los script de la aplicación presiona **Generar** caso contrario presiona **Refrescar** para continuar creando mas scripts.

Servicio: Servicio o interfaz a controlar
Comparte BW: Va ha prestar el ancho de banda?
BW Asignado: Valor de BW a asignar
Prioridad: Prio. Script (1 mayor - 8 menor)
Interfaz: **Guardar** **Actualizar**

Listado de Scripts

Listado de los scripts que se pueden generar para poder realizar el control del ancho de banda, se pueden generar una vez que tengan reglas asignadas al script.

	Codigo	Servicio	Comparte	Asignado	Interfaz			
1	0002	smtp-in	false	128	eth0	Cargar	Borrar	Reglas
26	0005	web-in	true	256	eth0	Cargar	Borrar	Reglas
32	0004	ftp-out	false	128	eth1	Cargar	Borrar	Reglas

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#); Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: file.htm

Referencia Requerimientos: RF029

Caso de Uso: Administrar Scripts

Scripts para control de BW

martes 13 octubre 2009, 15:30
Escrito por Jorge Guerrón Eras

Es recomendable hacer un script por cada servicio que se use en la empresa, además se recomienda que los scripts que son asignados con mayor ancho banda sean los que compartan su banda a los demás.

Si deseas generar los script de la aplicación presiona **Generar** caso contrario presiona **Refrescar** para continuar creando mas scripts.

Servicio:	<input type="text"/>	Servicio o interfaz a controlar
	Ingrese un dato.	
Comparte BW:	<input type="checkbox"/>	Va ha prestar el ancho de banda?
BW Asignado:	<input type="text" value="0"/>	Valor de BW a asignar
	Debe especificar un número mayor a 1!	
Prioridad:	<input type="text" value="9"/>	Prio. Script (1 mayor - 8 menor)
	Debe especificar un número menor a 8! Debe especificar un número mayor a 1!	
Interfaz:	<input type="text" value="Seleccione un item v"/>	
	Ingrese un dato.	
	<input type="button" value="Guardar"/> <input type="button" value="Actualizar"/>	

Nombre Vista: rule.htm
Referencia Requerimientos: RF033,RF035,RF037
Caso de Uso: Administrar Script

Reglas de control
reglas para control de bw

Scripts

Reglas de control Finalizar

Reglas para control de BW Finalizar Sesión

Código: Id del Script

Script: Nombre del Archivo

Ip de Origen:

Puerto de Origen:

Ip de Destino:

Puerto de Destino:

Reglas del script

Las opciones que a continuación se presentan permiten escoger una dirección ip perteneciente a un equipo o a una red ingresada con anterioridad, o en su defecto realizar el control en un puerto.

Ip Origen	Puerto Origen	Ip Destino	Puerto Destino	
2	25	150.190.0.0	25	<input type="button" value="Borrar"/>

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#); Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: rule.htm

Referencia Requerimientos: RF034

Caso de Uso: Administrar Script

Reglas de control

Reglas para control de BW

Código:	<input type="text" value="0"/>	Id del Script
Script:	<input type="text"/>	Nombre del Archivo
Ip de Origen:	<input type="text" value="Seleccione un item v"/>	
	No puede seleccionar la misma Ip.	
Puerto de Origen:	<input type="text" value="-1"/>	
	Debe especificar un número mayor a 0!	
Ip de Destino:	<input type="text" value="Seleccione un item v"/>	
	No puede seleccionar la misma Ip.	
Puerto de Destino:	<input type="text" value="700000"/>	
	Debe especificar un número menor a 65.535!	
		<input type="button" value="Guardar"/>

Nombre Vista: association.htm

Referencia Requerimientos: RF024,RF0,RF009,RF010, RF011

Caso de Uso: Administrar Script

Asociación de Scripts
asociación de scripts padres.

Administración Configuración Interfaz Ip's Red Scripts Relaciones

Asociación de Scripts Padres. Finalizar

martes 13 octubre 2009, 19:24
Escrito por Jorge Guerrón Eras

Finalizar Sesión

Refrescar

Nombre del Script: cbq-0005.web-in Script que va ha compartir

Nombre del Script: cbq-0002.smtp-in Script que va ha consumir

Guardar

Listado de asociaciones entre scripts.

Se presenta una lista con las diferectes asociaciones que ha almacenado el administrador

	Padre	Asociado	
1	cbq-0005.web-in	cbq-0002.smtp-in	Borrar

Licencia GNU-GPL. Gracias al aporte de: [MasterPC](#), [BearDeveloper](#), Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: inmon.htm

Referencia Requerimientos: RF002

Caso de Uso: Ingreso a la herramienta

CWR
control web de redes

Monitor Calculadora IP Escaner

Sección de Monitor

martes, 13 octubre 2009, 15:35
Escrito por Jorge Guerrón Eras

En ésta sección el monitor, va ha poder consultar (calip) los datos de una red, ip minima, ip máxima, cidr, broadcast, wildcard, etc, tiene un scanner de puertos al estilo de nmap (una aplicación básica), y unos cuantos consejos para el manejo de una red.

Menú Usuarios:
Finalizar Sesión

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#), Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: calip.htm

Referencia Requerimientos: RF039

Caso de Uso: Calcular direcciones ip

Calculadora IP
utilidades

Monitor **Calculadora IP** Escaner

Esta sección nos permite calcular a partir de una dirección ip, los datos que podemos utilizar en la configuración de nuestra red. Como lo hacemos? en el campo **Dirección IP**: escribimos la dirección ip a utilizar; tenemos 3 opciones para **GENERAR**, escribir la **Máscara de Red**: que si esta mal nos retorna un valor por defecto, o escribir el **CIDR**: que es un número entre 1 y 32, o finalmente, escribir el **Nº Hosts**: que vamos a habilitar en nuestra red, es decir la ip con cualquiera de los valores antes mencionados, y precionamos **GENERAR**

Dirección IP :

Máscara Red :

CIDR : Nº Hosts:

Red :

Broadcast :

Wlidcard :

IP Min :

IP Max :

Tipo :

Consejos.

Si deseas aprender sobre redes te recomiendo iniciar con estos tutoriales.

1. <http://www.netfilter.org/documentation/HOWTO/es/>
2. <http://www.vtc.com/products/Cisco-CCNA/ICND-Espanol-tutorials.htm>
3. <http://www.manualespdf.es/manual-datos/>

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#), Imágenes Zero Signs tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: calip.htm

Referencia Requerimientos: RF039

Caso de Uso: Calcular direcciones ip



Nombre Vista: scan.htm

Referencia Requerimientos: RF040

Caso de Uso: Escanear Puertos

Escaner de Puertos
utilidades

Monitor Calculadora IP Escaner

Esta sección nos permite verificar si tenemos o no puertos abiertos en nuestro servidor, de este modo poder configurar a nuestra conveniencia el acceso por medio de puertos a nuestra pc.

Dirección IP :

Todos Conocidos

Consejos.

Mis consejos personales para administrar una red son:

1. *Investiga bien tu estructura de red.*
2. *Verifica los servicios que tienes implementados.*
3. *Investiga documentación antes de hacer pruebas.*
4. *Mantén actualizada tu base de conocimientos de la red y de servicios*

Si deseas aprender sobre redes te recomiendo iniciar con estos tutoriales.

1. <http://www.pello.info/>
2. <http://www.manualespdf.es/manual-linux/>

Licencia GNU-GPL; Gracias al aporte de: [MasterPC](#), [BearDeveloper](#), Imágenes [Zero Signs](#) tonozero@hotmail.com, Diseño: [Free CSS Templates](#)

Nombre Vista: scan.htm

Referencia Requerimientos: RF040

Caso de Uso: Escanear Puertos



1.7 DESCRIPCIÓN DE CASOS DE USO

Identificador del Use Case: UC01
Nombre de Use Case: ADMINISTRAR CONFIGURACIÓN POR DEFECTO
Actor: Administrador - Monitor
Propósito: Ingresar y actualizar la información que usa la herramienta.
Descripción Breve: La herramienta permite al administrador o monitor, cargar los datos iniciales con los que iniciará a funcionar, luego solo al administrador modificar los datos de la configuración inicial.
Pre-Condición: <ul style="list-style-type: none"> • El <i>usuario</i> ingresa a la vista principal.
Post-Condición: <ul style="list-style-type: none"> • La herramienta debe actualizar los datos de configuración por defecto.
Ref. Requerimientos: RF003, RF004, RF005
Curso Normal
1.- El usuario elige el link aquí de la sección <i>Como inicio?</i> , de la vista <i>index.htm</i> .
2.- La herramienta verifica que no existan datos ya ingresados en la base de datos.
3.- La herramienta guarda la configuración por defecto.
4.- La herramienta presenta la vista <i>start.htm</i>
5.- El usuario elige el link Inicio de la barra de menús.
6.- La herramienta redirecciona a la vista <i>index.htm</i> .
7.- El usuario ingresa a la herramienta con la clave de administrador.
8.- El administrador elige el link Configuración , de la sección <i>Opciones del Administrador</i> , de la vista <i>general.htm</i> .

9.- La herramienta consulta los datos de Configuración
10.- La herramienta presenta la vista <i>config.htm</i>
11.- El administrador modifica los campos que desea cambiar y elige el botón [Aceptar]
12.- La herramienta verifica los datos ingresados
13.- La herramienta modifica los datos
14.- La herramienta presenta la vista <i>general.htm</i>
15.- Fin del caso de uso.
<i>Cursos Alternos</i>
<p>A. EXISTEN DATOS DE CONFIGURACIÓN</p> <p>A3. La herramienta presenta la vista <i>error.htm</i></p> <p>A4. El caso de uso continúa en el paso 5 del Curso Normal.</p>
<p>B. DATOS DE CONFIGURACIÓN INCORRECTOS.</p> <p>B12. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>B13. El caso de uso continúa en el paso 11 del Curso Normal.</p>

Identificador del Use Case: UC02
Nombre de Use Case: INGRESAR A LA HERRAMIENTA
Actor: Administrador - Monitor
Propósito: Ingresar a la herramienta para ver las opciones del usuario.
Descripción Breve: La herramienta permite al administrador o monitor, ingresar para poder realizar operaciones dependiendo de sus privilegios.
Pre-Condición: El <i>usuario</i> ingresa a la vista principal.
Post-Condición: Ingresar a las opciones de la herramienta.
Ref. Requerimientos: RF001, RF002
Curso Normal
1.- El usuario selecciona el link Acceso Usuarios en la sección <i>Acceso</i> , o link Acceso de la barra de menús de la vista <i>index.htm</i>
2.- La herramienta redirecciona a la vista <i>inadm.htm</i>
3.- El usuario ingresa los datos de Usuario y Clave , de la sección Menú Usuarios y elige el botón [Aceptar]
4.- La herramienta verifica los datos ingresados.
5.- La herramienta presenta la vista que le corresponde al usuario.
6.- Fin del Use Case.

<i>Cursos Alternos</i>
<p>A. USUARIO Y CLAVE INCORRECTOS.</p> <p>A4. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>A5. El caso de uso continúa en el paso 3 del Curso Normal.</p>

Identificador del Use Case: UC03
Nombre de Use Case: ADMINISTRAR DISPOSITIVOS DE RED
Actor: Administrador
Propósito: Ingresar y actualizar la información correspondiente a los dispositivos de red que se encuentran instalados y configurados en el servidor que va a realizar el control de ancho de banda.
Descripción Breve: La herramienta permite al administrador, ingresar la información de los dispositivos que se encuentran instalados en el servidor, actualizarlos y eliminarlos dependiendo de sus necesidades.
<p>Pre-Condición:</p> <ul style="list-style-type: none"> • El administrador ingresa con su usuario y clave a la herramienta.
<p>Post-Condición:</p> <ul style="list-style-type: none"> • La herramienta debe almacenar los datos de los dispositivos de red ingresados.
Ref. Requerimientos: RF006, RF007, RF008, RF009, RF010, RF011

<i>Curso Normal</i>
1.1.- El administrador elige el link Interfaz , de la vista <i>general.htm</i> .
1.2.- La herramienta presenta la vista <i>eth.htm</i>
1.3.- El administrador completa los campos y elige el botón [Aceptar]
1.4.- La herramienta valida los datos ingresados
1.5.- La herramienta almacena los datos del dispositivo
1.6.- La herramienta consulta los dispositivos creadas
1.7.- La herramienta presenta los dispositivos creados en la sección Interfaces de Red Creadas.
2.1.- El administrador elige el botón [Refrescar] de la sección Interfaces de Red Creadas.
2.2.- La herramienta consulta los dispositivos creados
2.3.- La herramienta presenta los dispositivos en la sección Interfaces de Red Creadas
2.4.- El administrador selecciona una interfaz de red a modificar y escoge el botón [Cargar]
2.5.- La herramienta carga los datos del dispositivo
2.6.- El administrador cambia los campos a modificar y elige el botón [Actualizar]
2.7.- La herramienta verifica los datos ingresados
2.8.- La herramienta actualiza los datos del dispositivo seleccionado
2.9.- La herramienta consulta los dispositivos creados
2.10.- La herramienta presenta los dispositivos creados en la sección Interfaces de Red Creadas.

3.1.- El administrador elige el botón [Refrescar] de la sección Interfaces de Red Creadas.
3.2.- La herramienta consulta los dispositivos creados
3.3.- La herramienta presenta los dispositivos en la sección Interfaces de Red Creadas.
3.4.- El administrador selecciona una interfaz de red a eliminar y escoge el botón [Borrar]
3.5.- La herramienta elimina el dispositivo seleccionado
3.6.- La herramienta consulta los dispositivos creados
3.7.- La herramienta presenta los dispositivos creados en la sección Interfaces de Red Creadas.
<i>Cursos Alternos</i>
<p>A. EXISTEN DATOS DE DISPOSITIVO A CREAR.</p> <p>A1.4. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>A1.5. El caso de uso continúa en el paso 1.3 del Curso Normal.</p>
<p>B. EXISTEN DATOS DE DISPOSITIVO A MODIFICAR.</p> <p>B2.7. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>B2.8. El caso de uso continúa en el paso 2.6 del Curso Normal.</p>

Identificador del Use Case: UC04
Nombre de Use Case: ADMINISTRAR IP DE EQUIPO
Actor: Administrador
Propósito: Ingresar y actualizar la información correspondiente a las direcciones Ip de nodos (equipos) de la red que van a ser incluidos en el control de ancho de banda.
Descripción Breve: La herramienta permite al administrador, ingresar la información de las direcciones Ip de nodos que tienen acceso a internet y que van a ser controlados para segmentar el consumo de ancho de banda, además permitirá actualizar y eliminar direcciones Ip

dependiendo de las necesidades del administrador de la red.
<p>Pre-Condición:</p> <ul style="list-style-type: none"> • El administrador ingresa con su usuario y clave a la herramienta.
<p>Post-Condición:</p> <ul style="list-style-type: none"> • La herramienta debe almacenar los datos de las direcciones Ip que van a ser objeto de control.
Ref. Requerimientos: RF012, RF013, RF014, RF015, RF016, RF017
Curso Normal
1.1.- El administrador elige el link Ip's , de la vista <i>general.htm</i>
1.2.- La herramienta presenta la vista <i>ip.htm</i>
1.3.- El administrador completa los campos y elige el botón [Aceptar]
1.4.- La herramienta valida los datos ingresados
1.5.- La herramienta almacena la dirección
1.6.- La herramienta consulta las direcciones creadas
1.9.- La herramienta presenta las direcciones creadas en la sección Ips, de nuestra red
2.1.- El administrador elige el botón [Refrescar] de la sección Ips, de nuestra red
2.2.- La herramienta consulta las direcciones existentes
2.3.- La herramienta presenta las direcciones en la sección Ips, de nuestra red
2.4.- El administrador selecciona una dirección a modificar y escoge el botón [Cargar]
2.5.- La herramienta carga los datos de la dirección

2.6.- El administrador cambia los campos a modificar y escoge el botón [Actualizar]
2.7.- La herramienta verifica los datos ingresados
2.8.- La herramienta actualiza los datos de la dirección seleccionada
2.9.- La herramienta consulta las direcciones creadas
2.10.- La herramienta presenta las direcciones creadas en la sección Ips, de nuestra red
3.1.- El administrador elige el botón [Refrescar] de la sección Ips, de nuestra red
3.2.- La herramienta consulta las direcciones creadas
3.3.- La herramienta presenta las direcciones en la sección Ips, de nuestra red
3.4.- El administrador selecciona una dirección a eliminar y escoge el botón [Borrar]
3.5.- La herramienta elimina la dirección seleccionada
3.6.- La herramienta consulta las direcciones creadas
3.7.- La herramienta presenta las direcciones creadas en la sección Ips, de nuestra red
<i>Cursos Alternos</i>
<p>A. DATOS DE IP A CREAR MAL INGRESADOS.</p> <p>A1.4. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>A1.5. El caso de uso continúa en el paso 1.3 del Curso Normal.</p>
<p>B. DATOS DE IP A MODIFICAR MAL INGRESADOS</p> <p>B2.7. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>B2.8. El caso de uso continúa en el paso 2.6 del Curso Normal.</p>

Identificador del Use Case: UC05
Nombre de Use Case: ADMINISTRAR IP DE SUBRED
Actor: Administrador
Propósito: Ingresar y actualizar la información correspondiente a las direcciones Ip de subredes que componen la red y que van a ser incluidas en el control de ancho de banda.
Descripción Breve: La herramienta permite al administrador, ingresar la información de las direcciones Ip de subredes que forman parte de la red y tienen acceso a internet y que van a ser controladas para segmentar el consumo de ancho de banda, además permitirá actualizar y eliminar direcciones Ip dependiendo de las necesidades del administrador de la red.
Pre-Condición: <ul style="list-style-type: none"> • El administrador ingresa con su usuario y clave a la herramienta.
Post-Condición: <ul style="list-style-type: none"> • La herramienta debe almacenar los datos de las direcciones Ip que van a ser objeto de control.
Ref. Requerimientos: RF018, RF019, RF020, RF021, RF022, RF023
Curso Normal
1.1.- El administrador elige el link Red , de la vista <i>general.htm</i>
1.2.- La herramienta presenta la vista <i>net.htm</i>
1.3.- El administrador completa los campos y elige el botón [Aceptar]
1.4.- La herramienta valida los datos ingresados

1.5.- La herramienta almacena la dirección	
1.6.- La herramienta consulta las direcciones creadas	
1.7.- La herramienta presenta las direcciones creadas en la sección Subredes, de nuestra red	
2.1.- El administrador elige el botón [Refrescar] de la sección Subredes, de nuestra red	
2.2.- La herramienta consulta las direcciones existentes	
2.3.- La herramienta presenta las direcciones en la sección Subredes, de nuestra red	
2.4.- El administrador selecciona una dirección a modificar y escoge el botón [Cargar]	
2.5.- La herramienta carga los datos de la dirección	
2.6.- El administrador cambia los campos a modificar y escoge el botón [Actualizar]	
2.7.- La herramienta verifica los datos ingresados	
2.8.- La herramienta actualiza los datos de la dirección seleccionada	
2.9.- La herramienta consulta las direcciones creadas	
2.10.- La herramienta presenta las direcciones creadas en la sección Subredes, de nuestra red	
3.1.- El administrador elige el botón [Refrescar] de la sección Subredes, de nuestra red	
3.2.- La herramienta consulta las direcciones creadas	
3.3.- La herramienta presenta las direcciones en la sección Subredes, de nuestra red	
3.4.- El administrador selecciona una dirección a eliminar y escoge el botón [Borrar]	

3.5.- La herramienta elimina la dirección seleccionada
3.6.- La herramienta consulta las direcciones creadas
3.7.- La herramienta presenta las direcciones creadas en la sección Subredes, de nuestra red
<i>Cursos Alternos</i>
<p>A. DATOS DE IP A CREAR MAL INGRESADOS.</p> <p>A1.4. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>A1.5. El caso de uso continúa en el paso 1.3 del Curso Normal.</p>
<p>B. DATOS DE IP A MODIFICAR MAL INGRESADOS</p> <p>B2.7. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>B2.8. El caso de uso continúa en el paso 2.6 del Curso Normal.</p>

<i>Identificador del Use Case:</i> UC06
<i>Nombre de Use Case:</i> ADMINISTRAR SCRIPT
<i>Actor:</i> Administrador
<i>Propósito:</i> Ingresar y actualizar la información correspondiente a los scripts que se van a ejecutar para realizar el control de ancho de banda.
<i>Descripción Breve:</i> La herramienta permite al administrador, ingresar la información correspondiente a los scripts que se van a generar y nos van a ayudar en el control de acceso a internet, así como sus reglas y sus relaciones.

<p>Pre-Condición:</p> <ul style="list-style-type: none"> • El administrador ingresa con su usuario y clave a la herramienta.
<p>Post-Condición:</p> <ul style="list-style-type: none"> • La herramienta debe almacenar los datos de los scripts, reglas y relaciones entre scripts, así como generar los scripts con sus datos.
<p>Ref. Requerimientos: RF024, RF025, RF026, RF027, RF028, RF029, RF030, RF031, RF032, RF033, RF034, RF035, RF036, RF037, RF038.</p>
<p style="text-align: center;">Curso Normal</p>
<p>1.1.- El administrador elige el link Scripts, de la vista <i>general.htm</i></p>
<p>1.2.- La herramienta presenta la vista <i>file.htm</i></p>
<p>1.3.- El administrador elige el botón [Refrescar] de la sección Scripts para control de BW</p>
<p>1.4.- La herramienta consulta las interfaces de red a utilizar</p>
<p>1.5.- La herramienta consulta los scripts creados</p>
<p>1.6.- La herramienta carga las interfaces en el campo Interfaz: y los scripts en la sección Listado de Scripts</p>
<p>1.7.- El administrador completa los campos y elige el botón [Aceptar]</p>
<p>1.8.- La herramienta valida los datos ingresados</p>
<p>1.9.- La herramienta almacena los datos del script</p>
<p>1.10.- La herramienta consulta las interfaces de red a utilizar</p>
<p>1.11.- La herramienta consulta los scripts creados</p>
<p>1.12.- La herramienta carga las interfaces en el campo Interfaz: y los scripts en la</p>

sección Listado de Scripts
2.1.- El administrador selecciona un script a modificar y escoge el botón [Cargar]
2.2.- La herramienta carga los datos del script
2.3.- El administrador cambia los campos a modificar y elige el botón [Actualizar]
2.4.- La herramienta verifica los datos ingresados
2.5.- La herramienta actualiza los datos del script seleccionado
2.6.- La herramienta consulta las interfaces de red a utilizar y los scripts creados
2.7.- La herramienta carga las interfaces en el campo Interfaz: y los scripts en la sección Listado de Scripts
3.1.- El administrador selecciona un script a eliminar y escoge el botón [Borrar]
3.2.- La herramienta elimina las reglas del script seleccionado
3.3.- La herramienta elimina las relaciones del script seleccionado
3.4.- La herramienta elimina el script seleccionado
3.5.- La herramienta consulta las interfaces de red a utilizar y los scripts creados
3.6.- La herramienta carga las interfaces en el campo Interfaz: y los scripts en la sección Listado de Scripts
3.7.- El administrador elige el botón [Generar]
3.8.- La herramienta genera los scripts que contengan reglas.
4.1.- El administrador elige el botón [Reglas] de la sección Listado de Scripts
4.2.- La herramienta consulta las direcciones ip de equipos y subred
4.3.- La herramienta consulta las reglas del script

4.4.- La herramienta redirecciona a la vista <i>rule.htm</i>
4.5.- La herramienta carga las direcciones en los campos Ip de Origen: e Ip de Destino:, los datos del script en los campos Código: y Script:
4.6.- El administrador completa los campos y elige el botón [Guardar]
4.7.- La herramienta valida los datos ingresados
4.8.- La herramienta almacena los datos de la regla para el script
4.9.- La herramienta consulta las direcciones ip de equipos y subred
4.10.- La herramienta consulta las reglas del script
4.11.- La herramienta redirecciona a la vista <i>rule.htm</i>
5.1.- El administrador elige el botón [Borrar]
5.2.- La herramienta elimina la regla seleccionada del script.
5.3.- La herramienta consulta las direcciones ip de equipos y subred
5.4.- La herramienta consulta las reglas del script
5.5.- La herramienta redirecciona a la vista <i>rule.htm</i>
5.6.- La herramienta carga las direcciones en los campos Ip de Origen: e Ip de Destino:, los datos del script en los campos Código: y Script:
6.1.- El administrador elige el link Scripts
6.2.- La herramienta redirecciona a la vista <i>file.htm</i>
6.3.- El administrador elige el link Relaciones , de la barra de menús
6.4.- La herramienta redirecciona a la vista <i>association.htm</i>
6.5.- El administrador elige el botón [Refrescar]

6.6.- La herramienta consulta los scripts
6.7.- La herramienta consulta las relaciones existentes
6.8.- La herramienta carga los scripts en los campos Nombre del Script: , y las relaciones en la sección Listado de Scripts
6.9.- El administrador selecciona como relacionar los scripts y elige el botón [Guardar]
6.10.- La herramienta valida los datos seleccionados
6.11.- La herramienta almacena la relación
6.12.- La herramienta consulta los scripts
6.13.- La herramienta consulta las relaciones existentes
6.14.- La herramienta carga los scripts en los campos Nombre del Script: , y las relaciones en la sección Listado de Scripts
6.15.- El administrador elige el botón [Borrar]
6.16.- La herramienta elimina la regla seleccionada del script.
6.17.- La herramienta consulta los scripts
6.18.- La herramienta consulta las relaciones existentes
6.19.- La herramienta carga los scripts en los campos Nombre del Script: , y las relaciones en la sección Listado de Scripts
<i>Cursos Alternos</i>
A. DATOS DE SCRIPT A CREAR MAL INGRESADOS
A1.7. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.

A1.8. El caso de uso continúa en el paso 1.6 del Curso Normal.
<p>B. DATOS DE SCRIPT A MODIFICAR MAL INGRESADOS</p> <p>B2.4. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.</p> <p>B2.5. El caso de uso continúa en el paso 2.3 del Curso Normal.</p>
<p>C. DATOS DE REGLA DE SCRIPT A CREAR MAL INGRESADOS</p> <p>C4.5. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido. C4.6. El caso de uso continúa en el paso 4.4 del Curso Normal.</p>

Identificador del Use Case: UC07
Nombre de Use Case: CALCULAR DIRECCIONES IP
Actor: Monitor
Propósito: Calcular los datos de una Dirección Ip.
Descripción Breve: La herramienta permite al monitor, ingresar los datos de una dirección ip y máscara de red, o cidr, o número de host a incluir en una red, y a partir de éstos datos, calcular datos complementarios como broadcast, wildcard, red, ip mínima, ip máxima y el tipo de red al que pertenece.
<p>Pre-Condición:</p> <ul style="list-style-type: none"> • El <i>monitor</i> ingresa a la herramienta.
<p>Post-Condición:</p> <ul style="list-style-type: none"> • Obtener los datos complementarios de una dirección ip
Ref. Requerimientos: RF039

<i>Curso Normal</i>
1.- El monitor selecciona el link Calculadora IP de la barra de menús de la vista <i>inadm.htm</i>
2.- La herramienta redirecciona a la vista <i>calip.htm</i>
3.- El monitor ingresa la dirección ip a calcular
4.- El monitor ingresa la máscara de red/cidr/n° hosts y elige el botón [Generar]
5.- La herramienta verifica los campos
6.- La herramienta calcula los valores
7.- La herramienta presenta los datos
8.- El monitor elige el botón [Actualizar]
9.- La herramienta encera los campos
10.- Fin del Use Case.
<i>Cursos Alternos</i>
<p>A. NO SE HA INGRESADO DIRECCIÓN IP.</p> <p>A5. La herramienta muestra un mensaje de error indicando que el campo Dirección IP: no puede estar vacío</p> <p>A6. El caso de uso continúa en el paso 3 del Curso Normal.</p>
<p>B. NO SE HA INGRESADO MÁSCARA DE RED/CIDR/N° HOST.</p> <p>B6. La herramienta muestra un mensaje de error indicando que por lo menos uno de los tres campos no puede estar vacío</p> <p>B7. El caso de uso continúa en el paso 4 del Curso Normal.</p>

Identificador del Use Case: UC08
Nombre de Use Case: ESCANEAR PUERTOS
Actor: Administrador - Monitor
Propósito: Ingresar a la herramienta para ver las opciones del usuario.
Descripción Breve: La herramienta permite al monitor, ingresar una dirección ip y verificar si existen o no puertos abiertos.
Pre-Condición: <ul style="list-style-type: none"> • El <i>monitor</i> ingresa a la herramienta.
Post-Condición: <ul style="list-style-type: none"> • EL monitor puede verificar que existen puertos abiertos en un equipo.
Ref. Requerimientos: RF040
Curso Normal
1.- El monitor selecciona el link Escanear de la barra de menús de la vista <i>inadm.htm</i>
2.- La herramienta redirecciona a la vista <i>scan.htm</i>
3.- El monitor ingresa la dirección ip a escanear
4.- El monitor elige entre las opciones Todos ó Conocidos, y elige el botón [Iniciar]
5.- La herramienta verifica los campos
6.- La herramienta escanea los puertos de la ip escogida
7.- La herramienta presenta los resultados
8.- La monitor escoge la opción [Detener]
9.- Fin del Use Case

Cursos Alternos

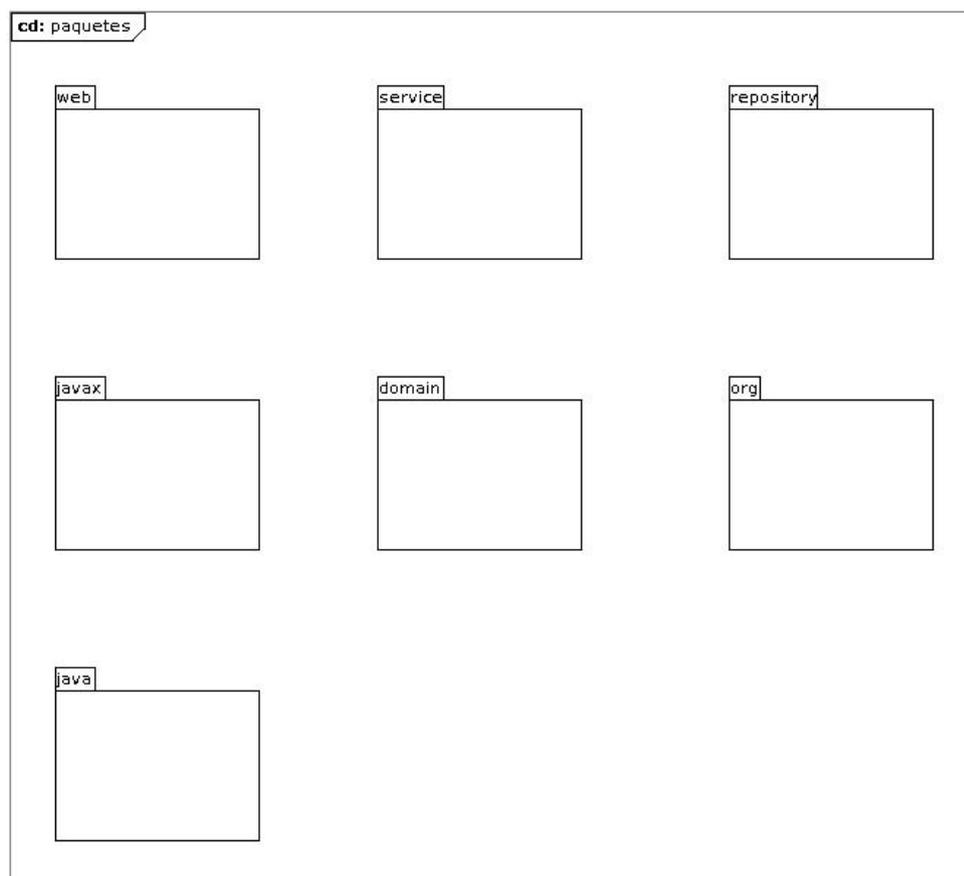
A. USUARIO Y CLAVE INCORRECTOS.

A5. La herramienta muestra un mensaje de error en cada campo, indicando el error cometido.

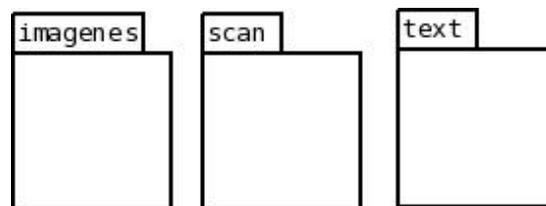
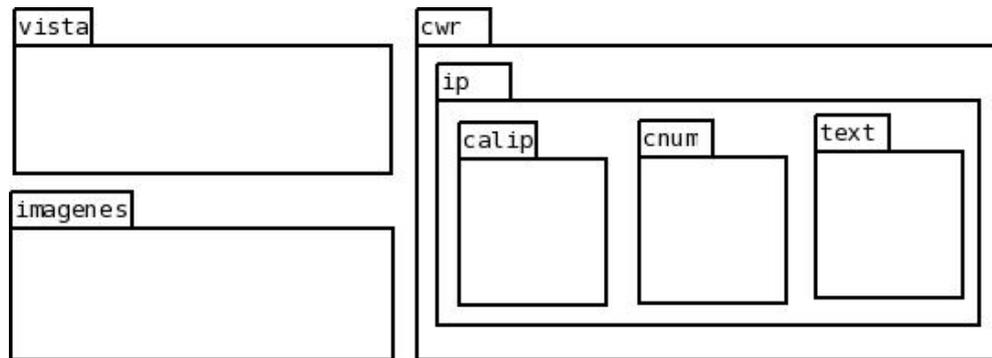
A6. El caso de uso continúa en el paso 3 del Curso Normal.

1.8 DIAGRAMAS DE PAQUETES

1.8.1 Paquetes Herramienta Web



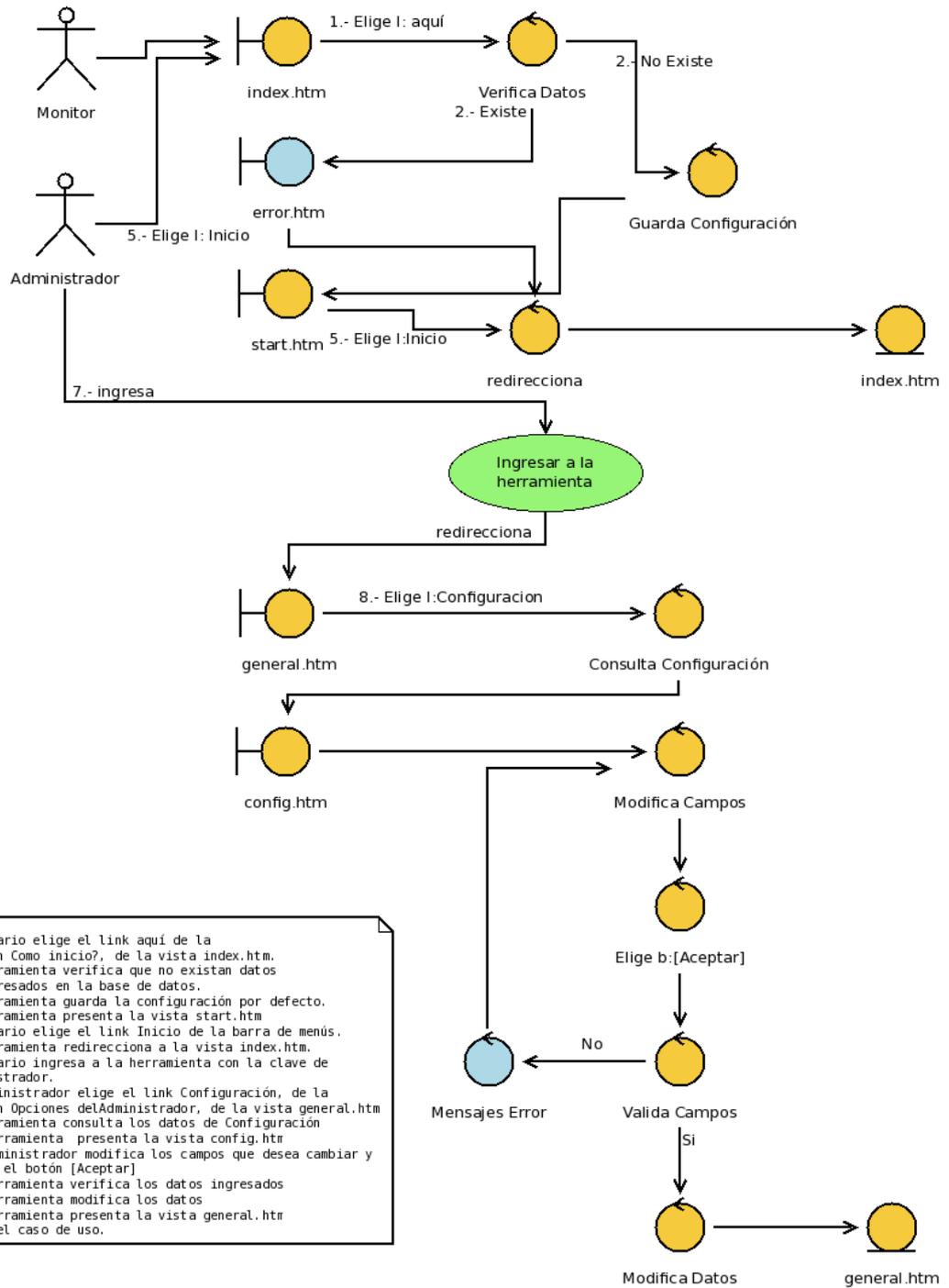
1.8.2 Paquetes Applet Calculadora Ip



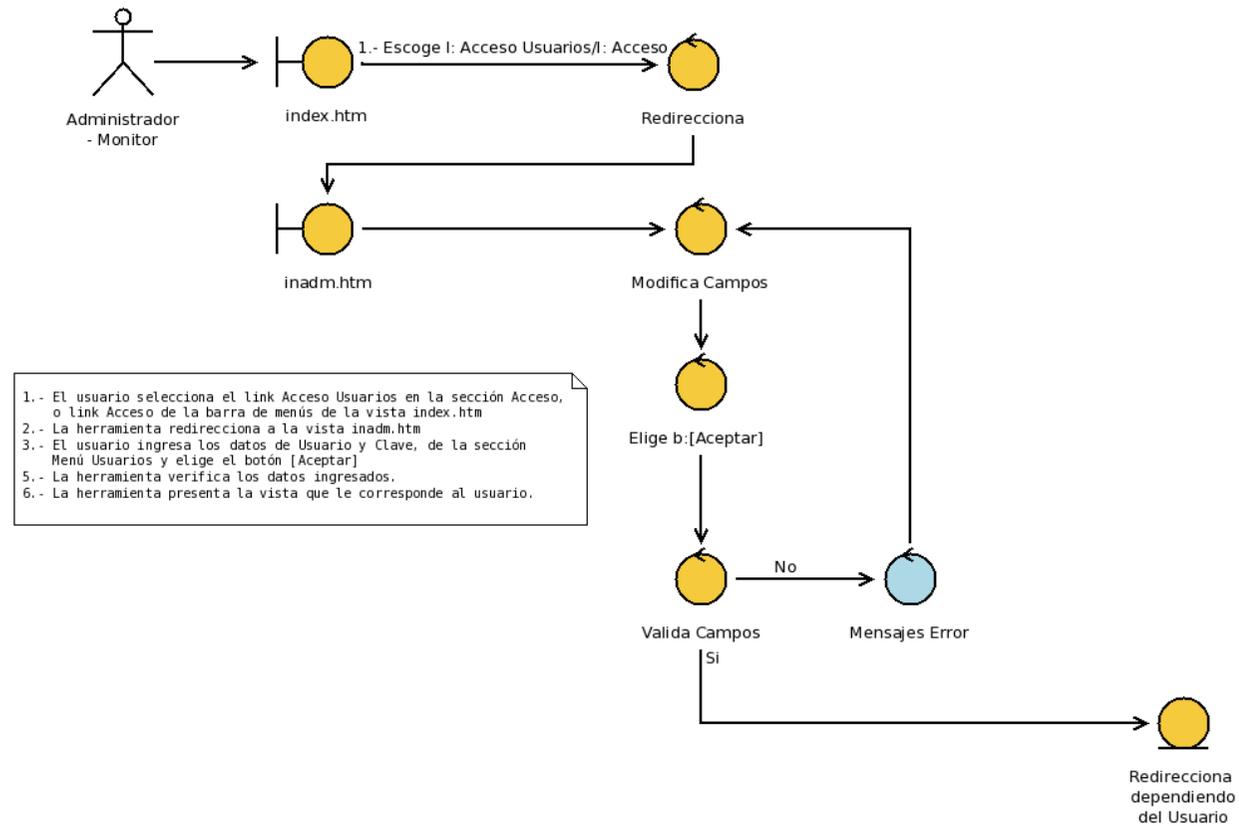
1.8.3 Paquetes Applet Escaner Puertos

1.9 DIAGRAMAS DE ROBUSTEZ

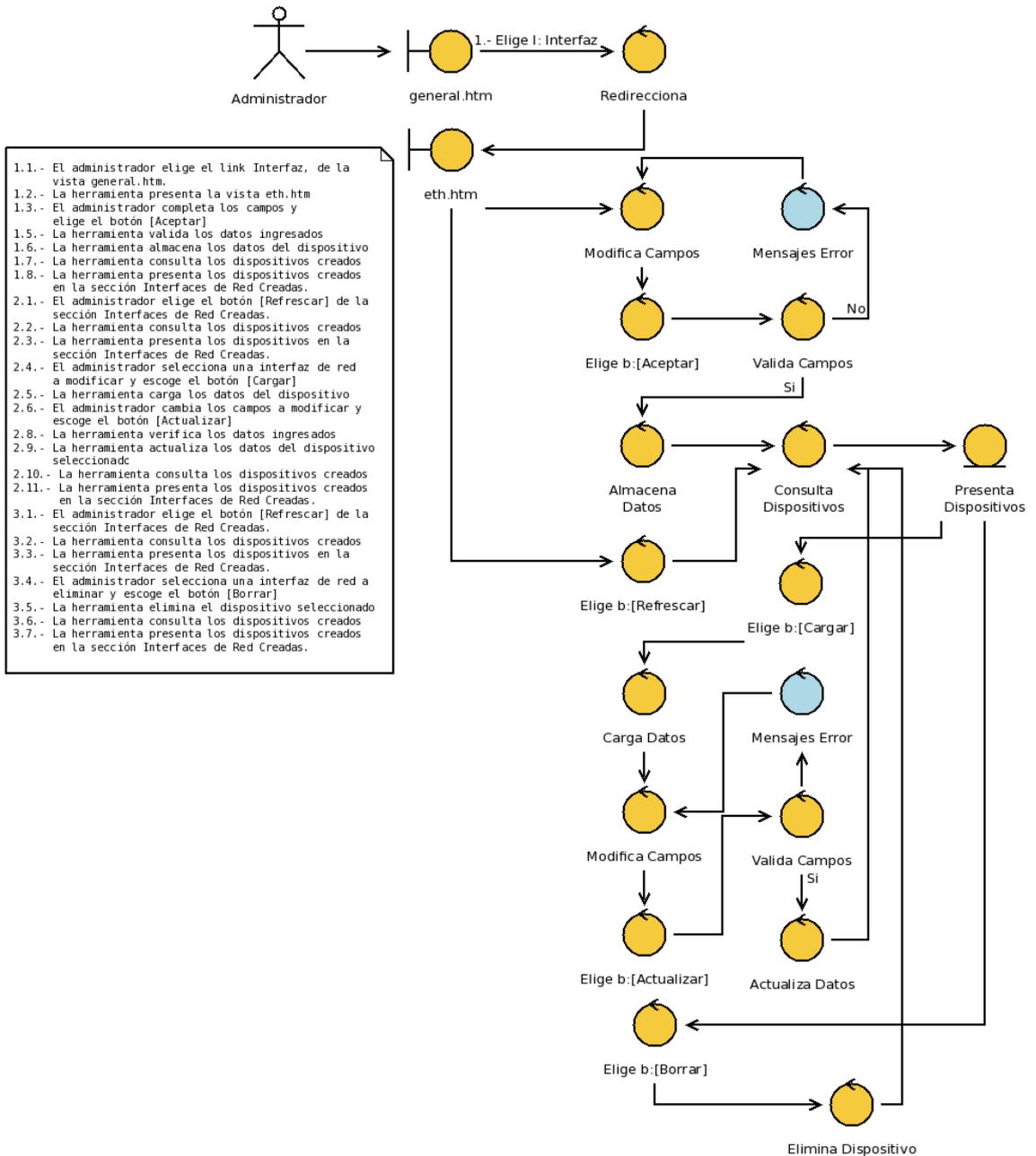
1.9.1 DR01: Administrar Configuración por Defecto



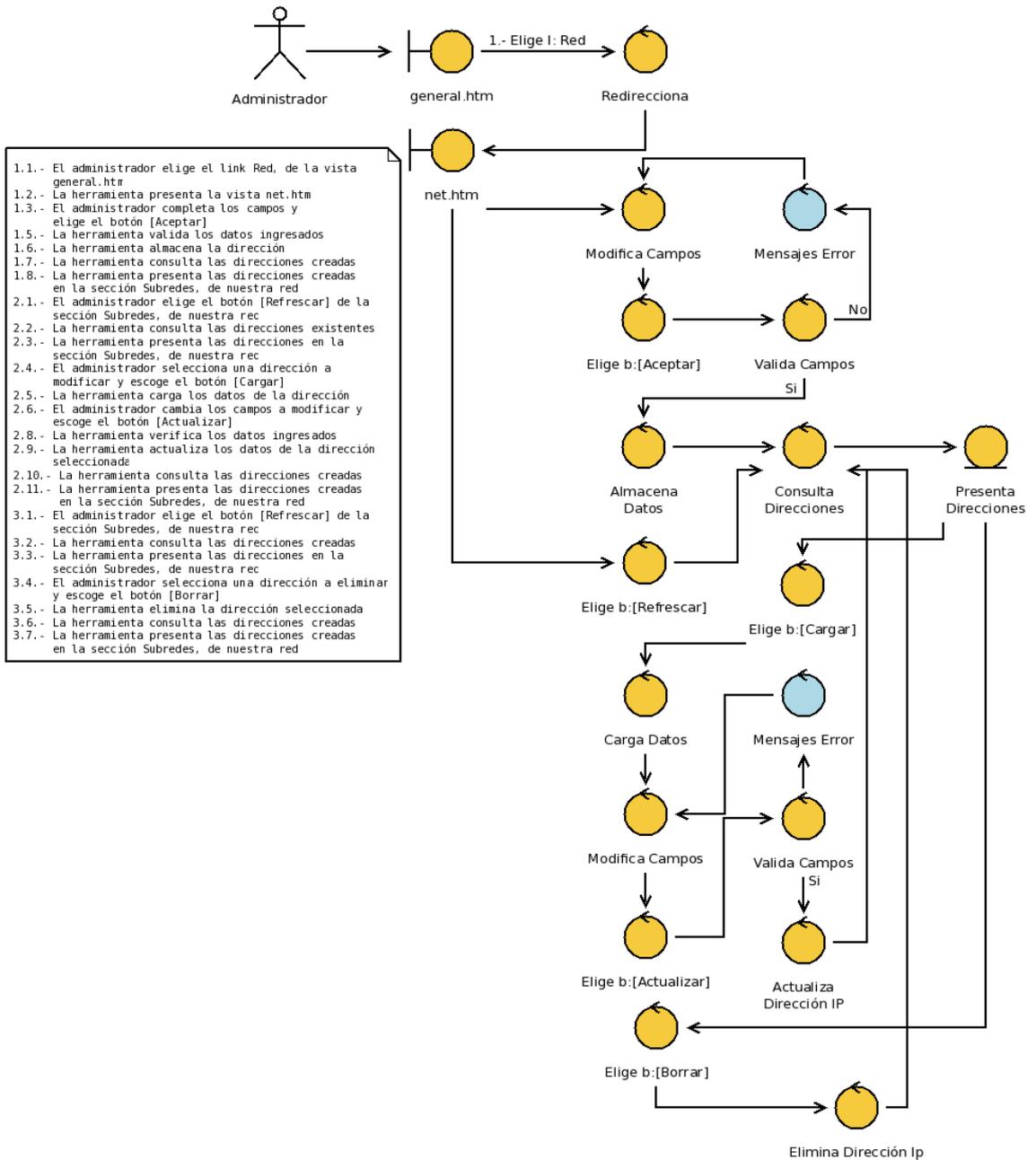
1.9.2 DR02: Ingresar a la Herramienta



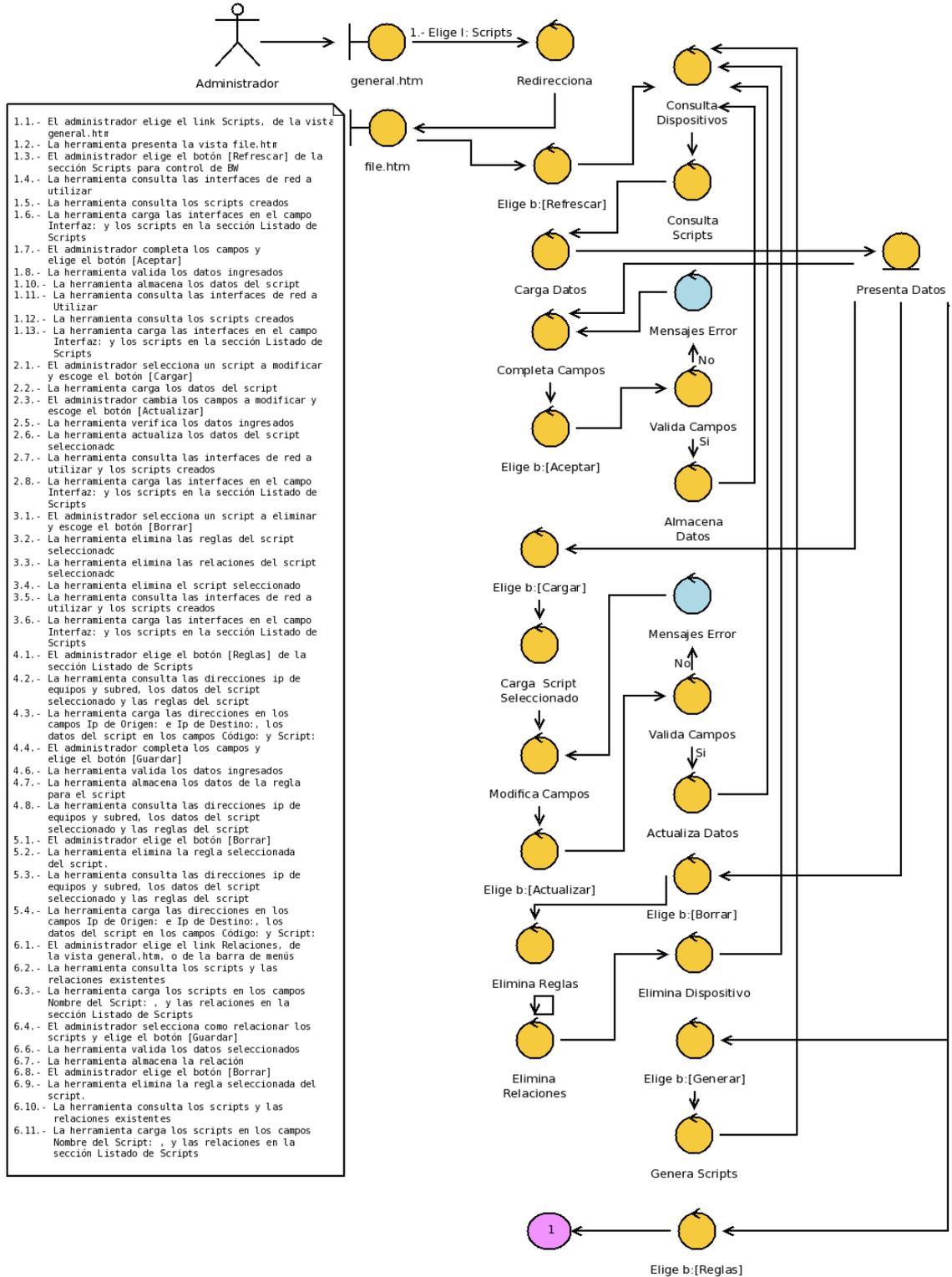
1.9.3 DR03: Administrar Dispositivos de Red



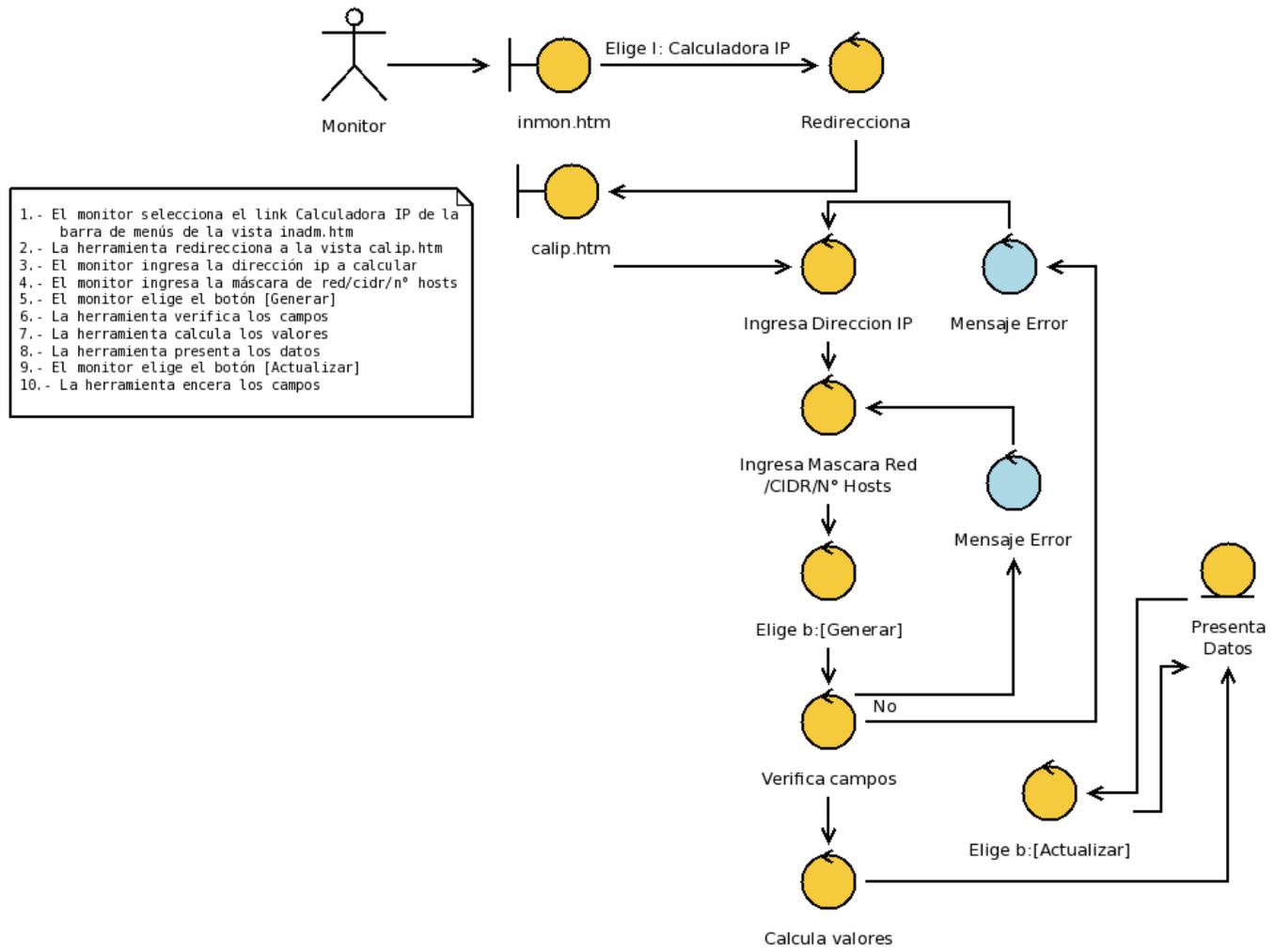
1.9.5 DR05: Administrar Ip de Subred



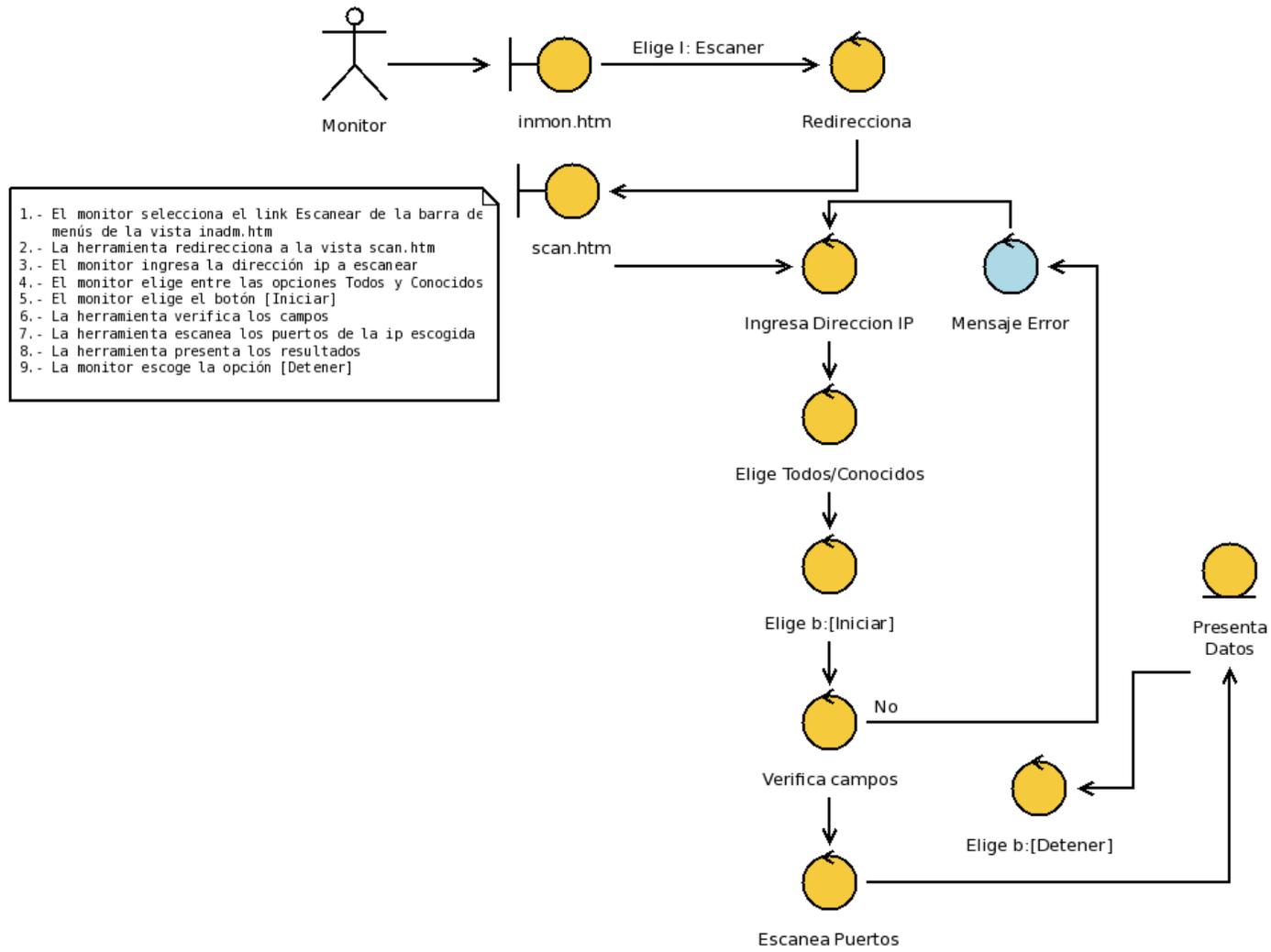
1.9.6 DR06: Administrar Script



1.9.7 DR07: Calcular Direcciones Ip

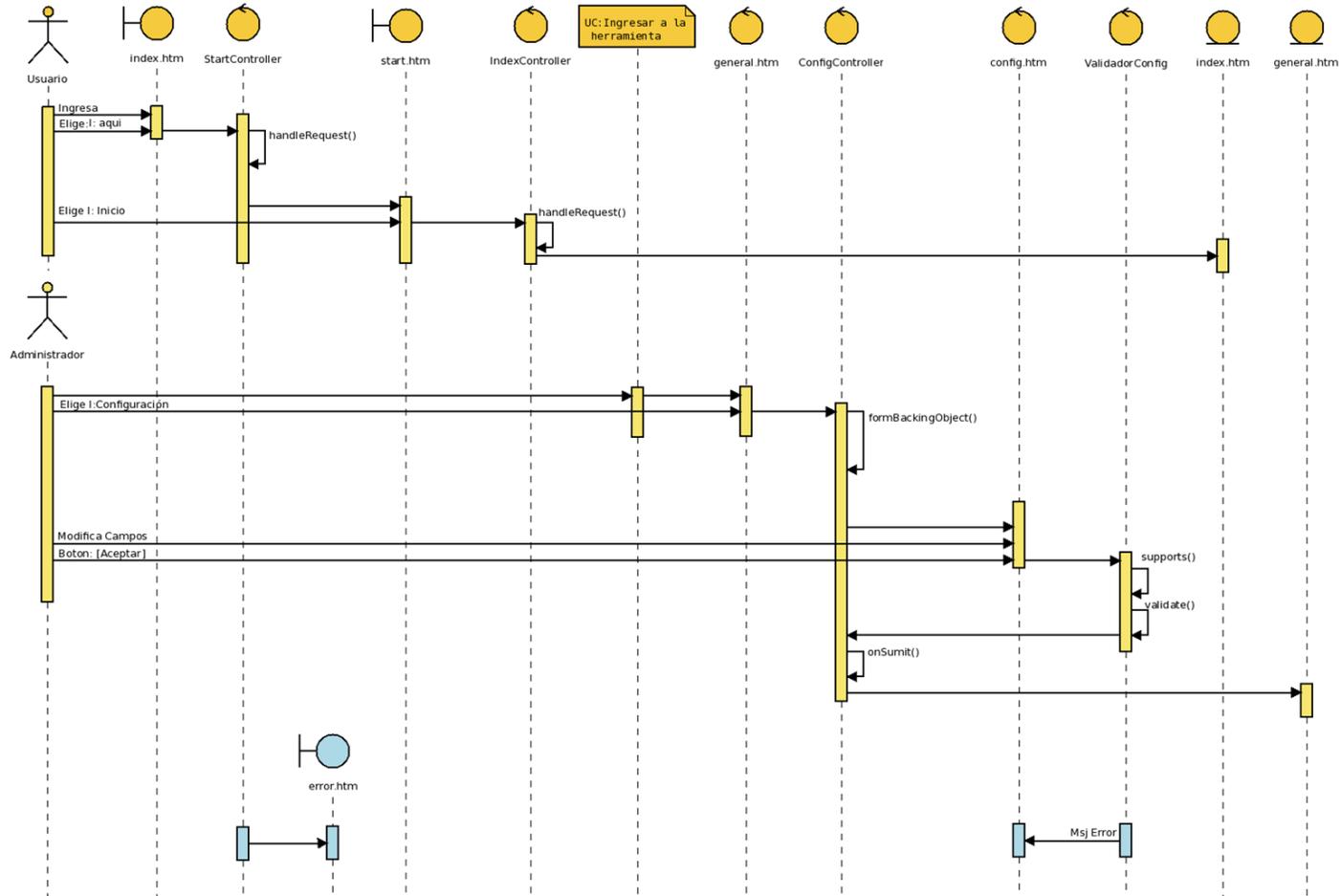


1.9.8 DR08: Escanear Puertos

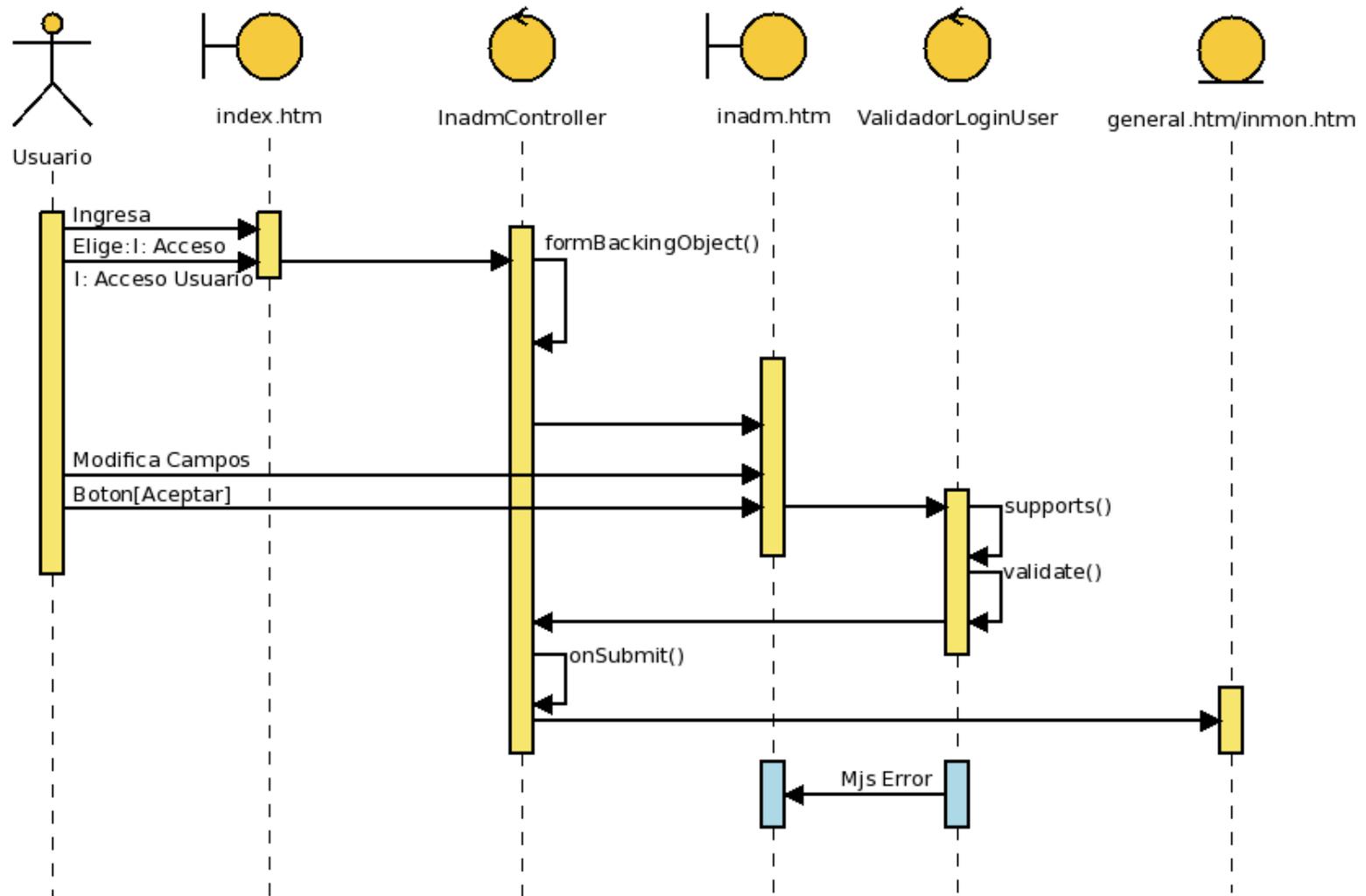


1.10 DIAGRAMAS DE SECUENCIA

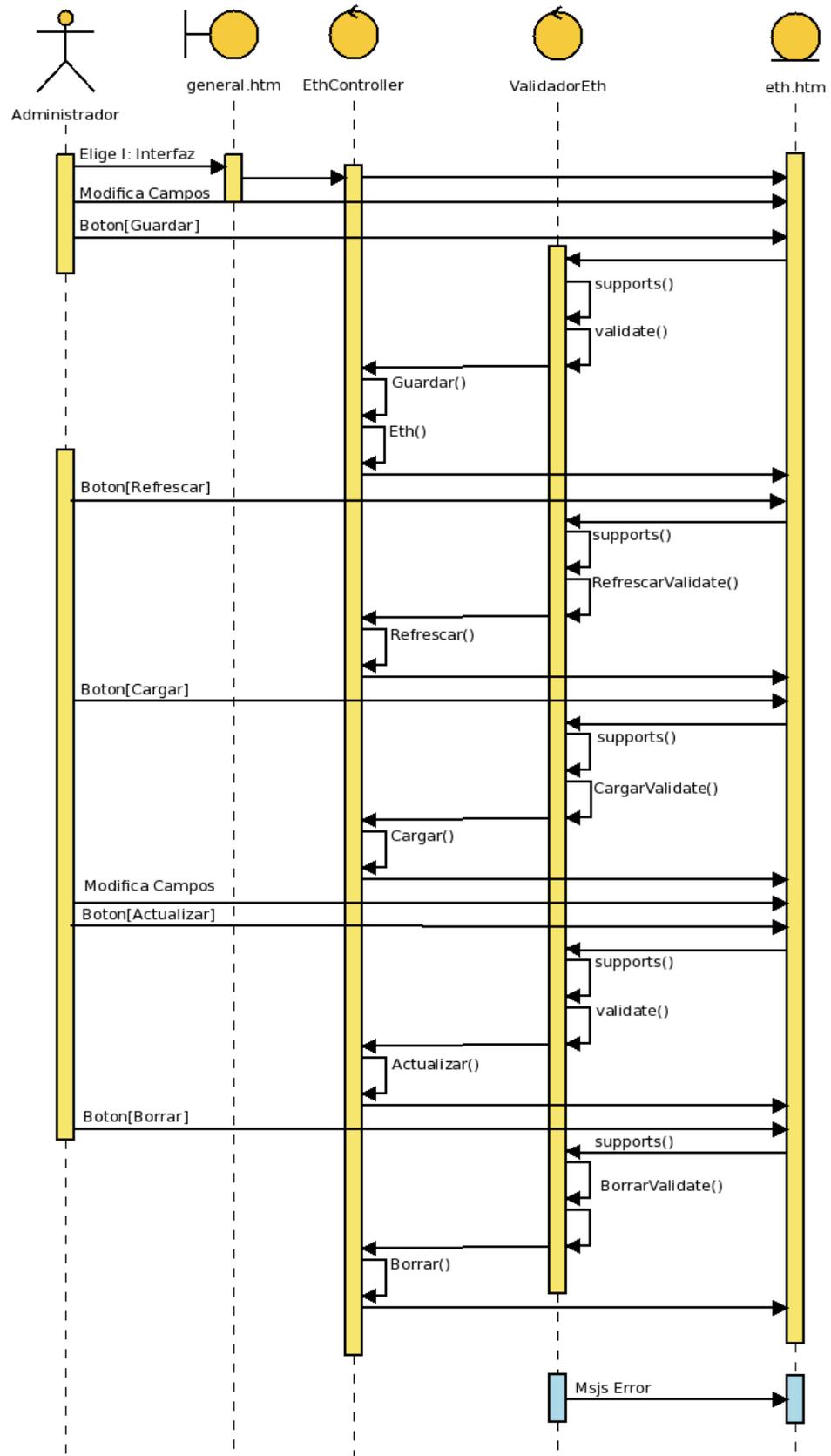
1.10.1 **DS01: Administrar Configuración por Defecto**



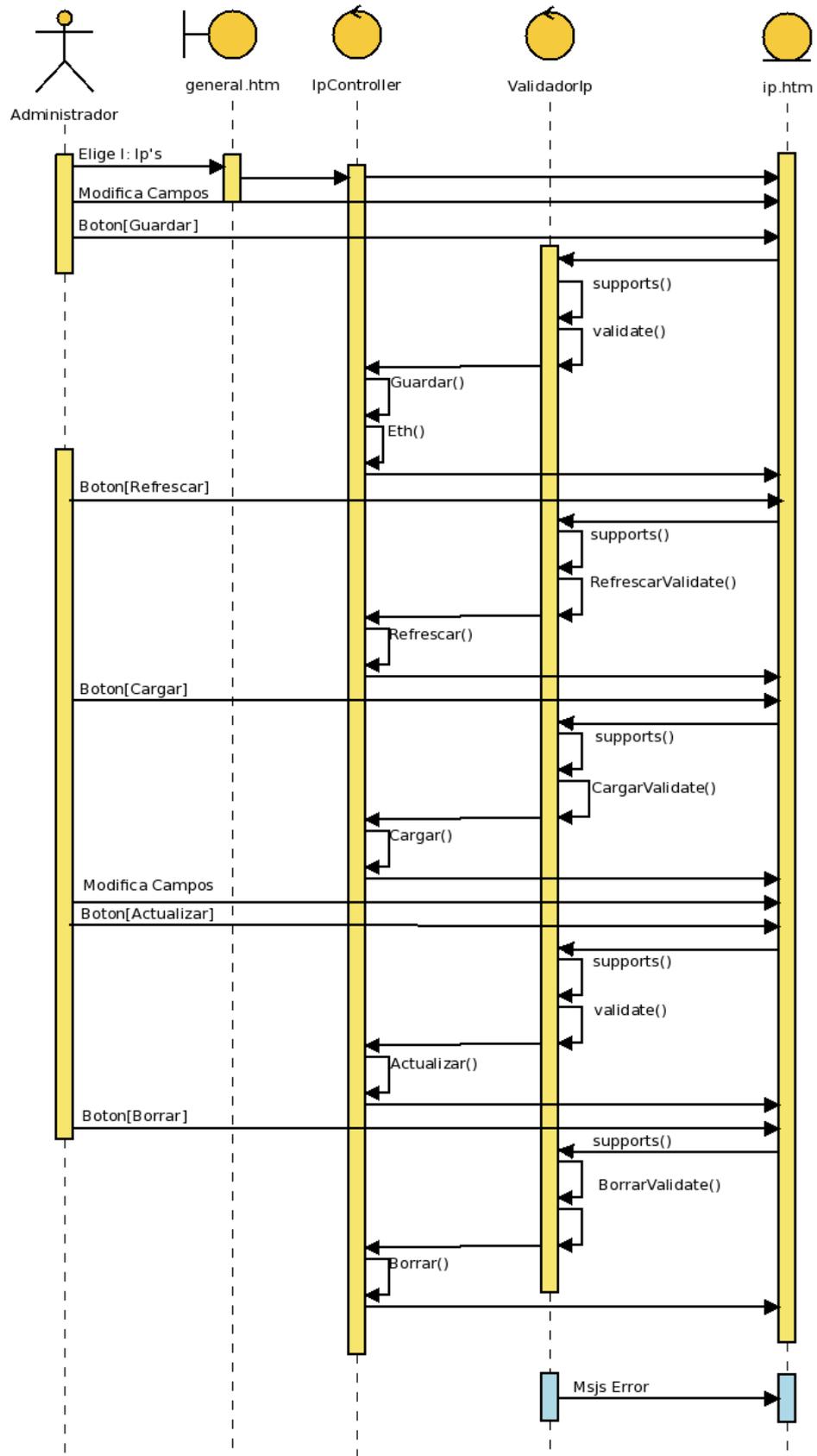
1.10.2 DS02: Ingresar a la Herramienta



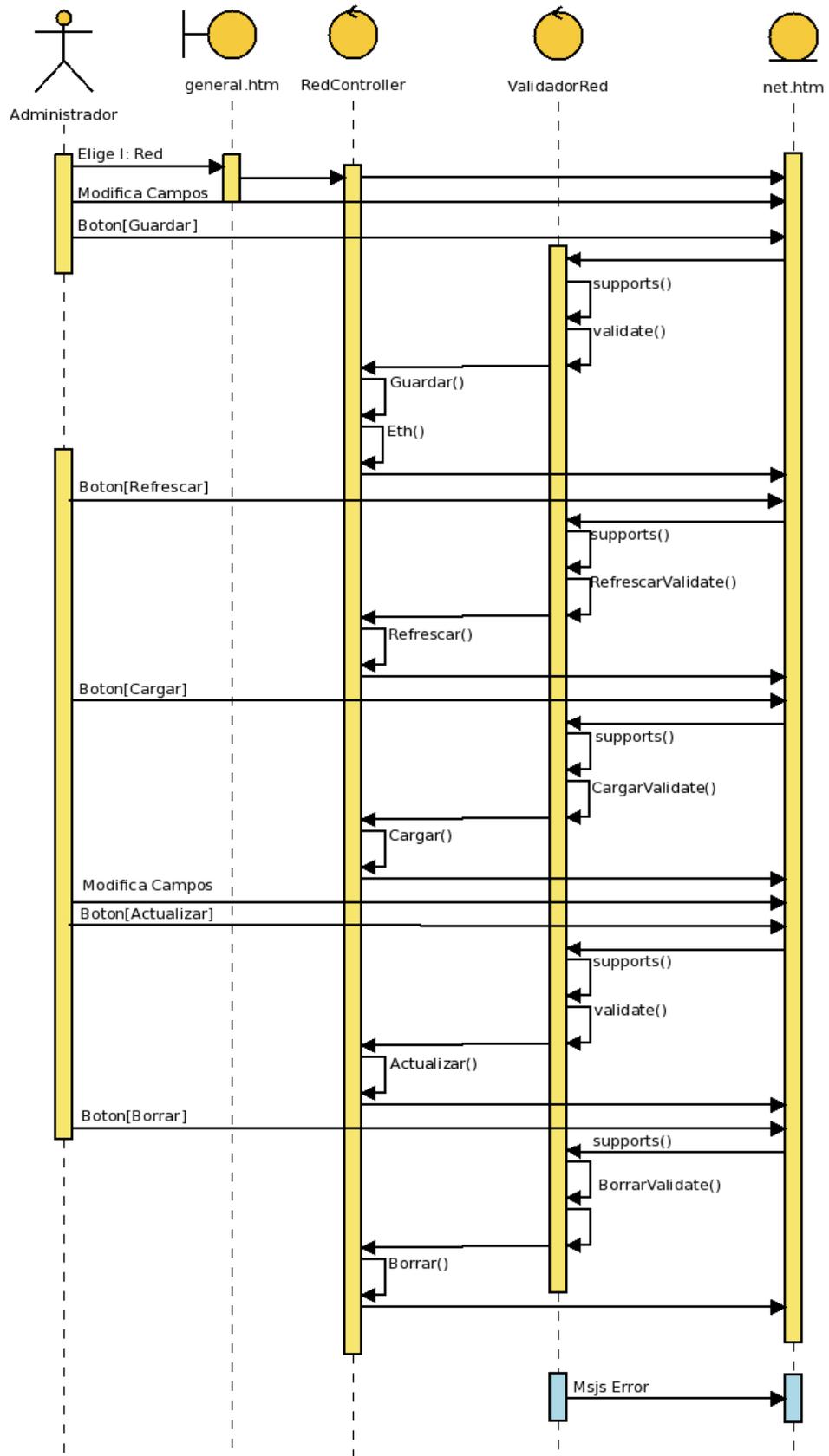
1.10.3 **DS03: Administrar Dispositivos de Red**



1.10.4 **DS04: Administrar Ip de Equipo**

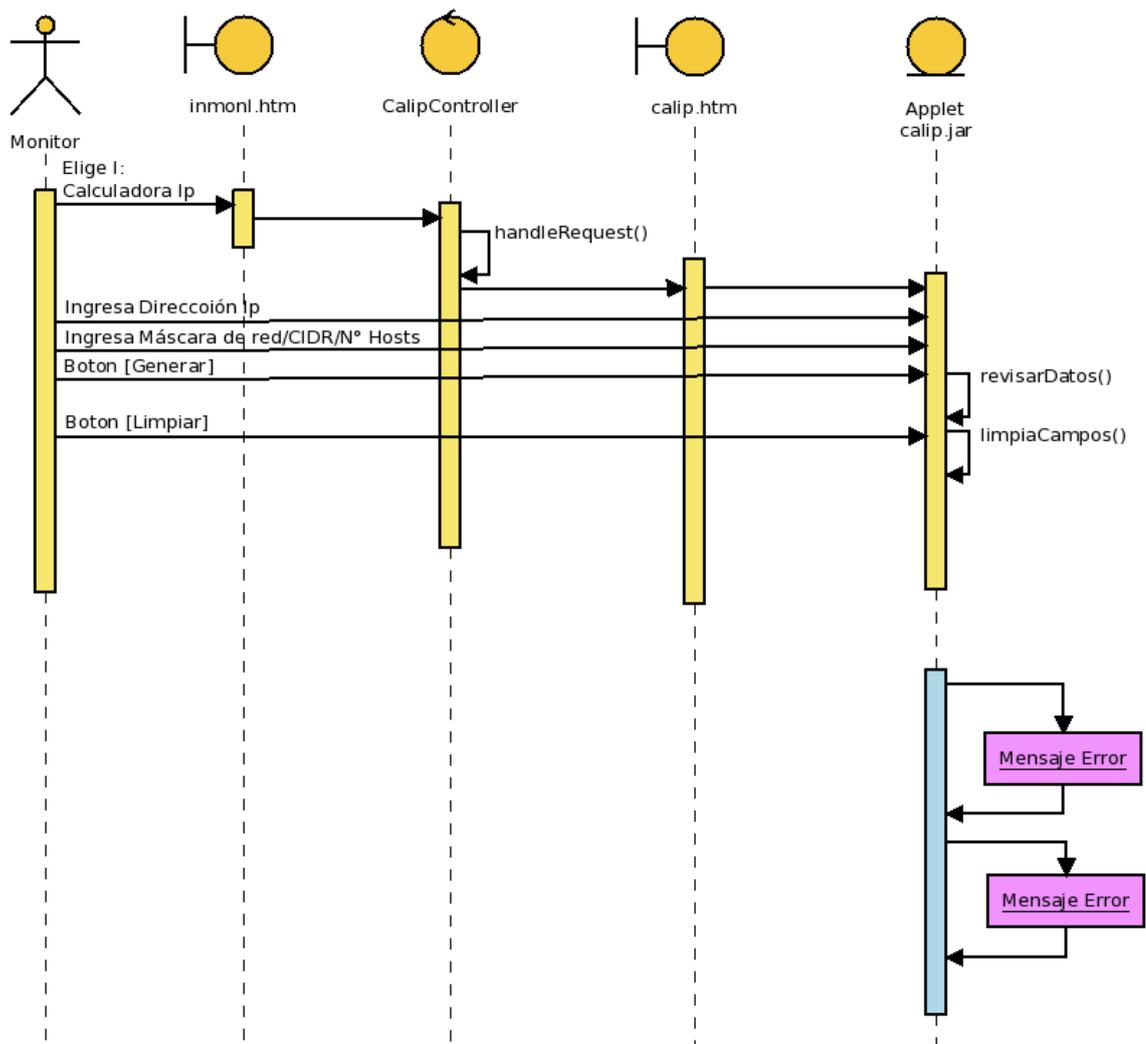


1.10.5 **DS05: Administrar Ip de Subred**

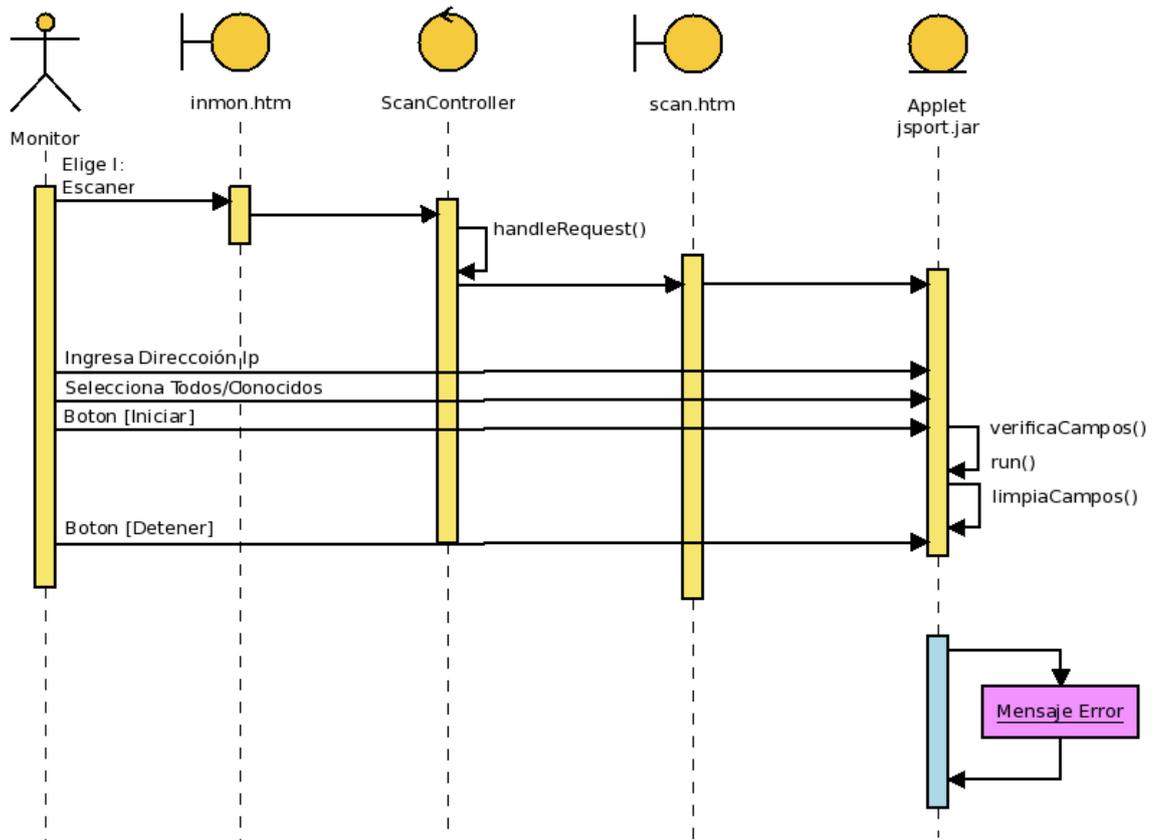


1.10.6 **DS06: Administrar Script**

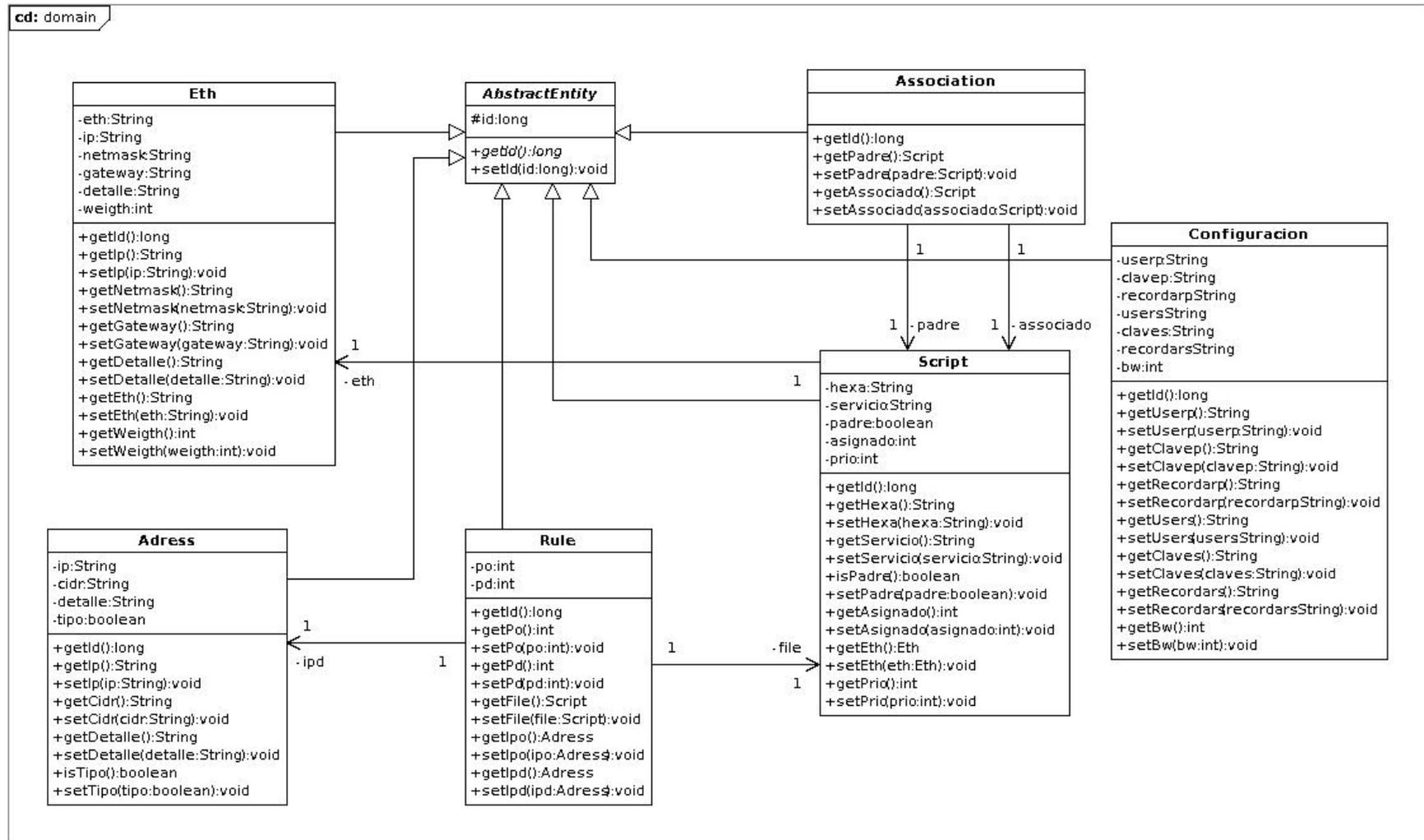
1.10.7 DS07: Calcular Direcciones Ip

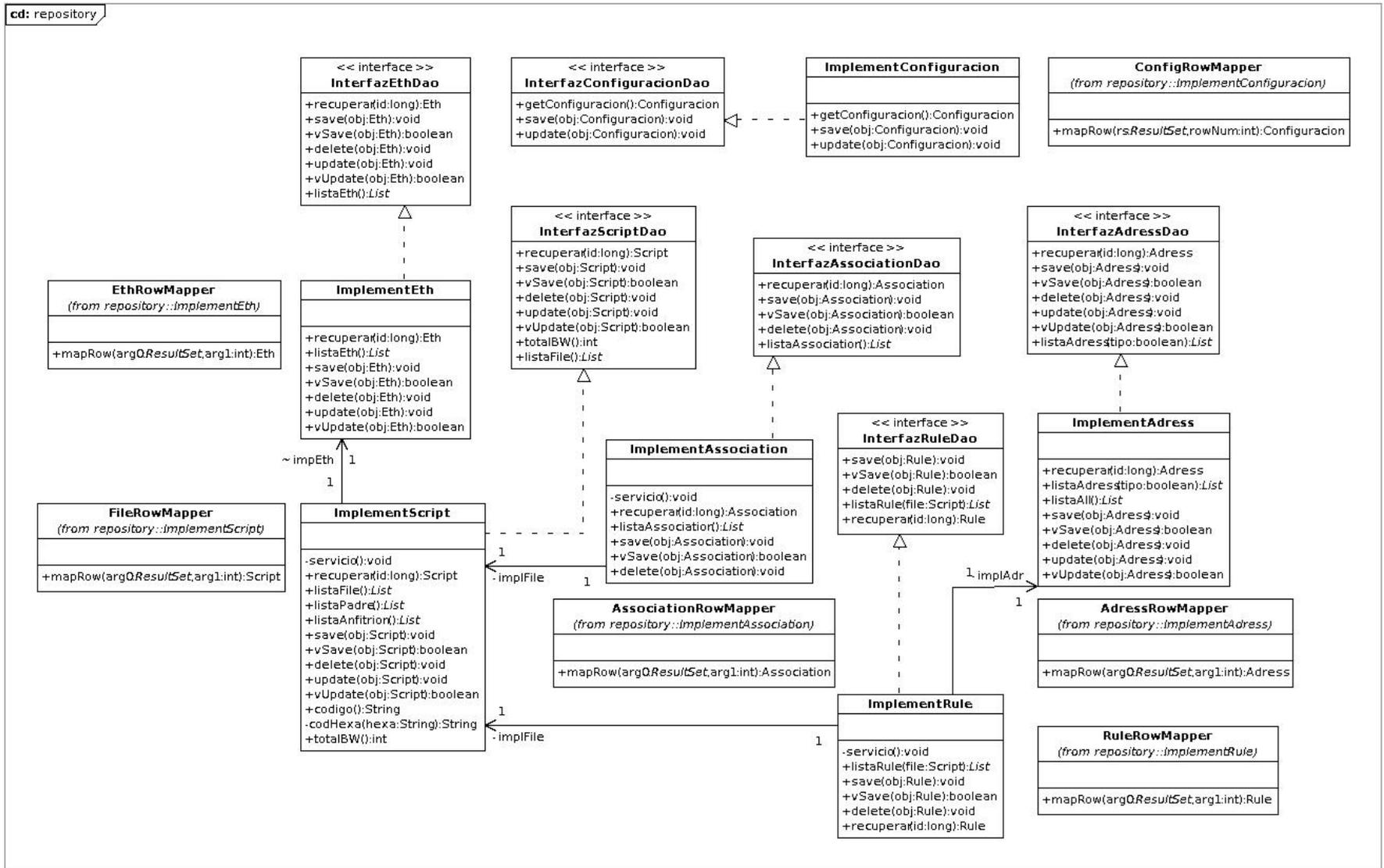


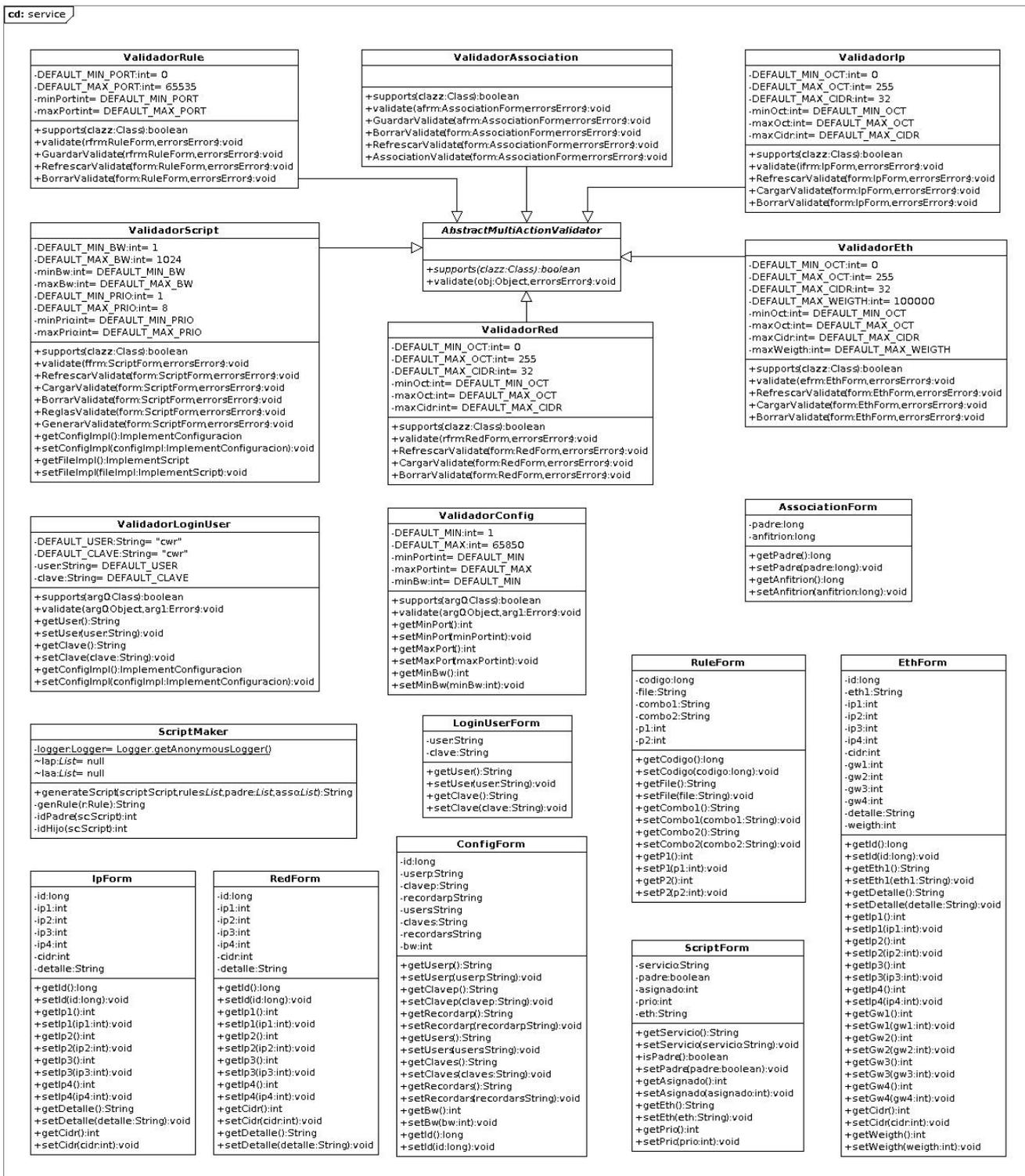
1.10.8 DS08: Escanear Puertos

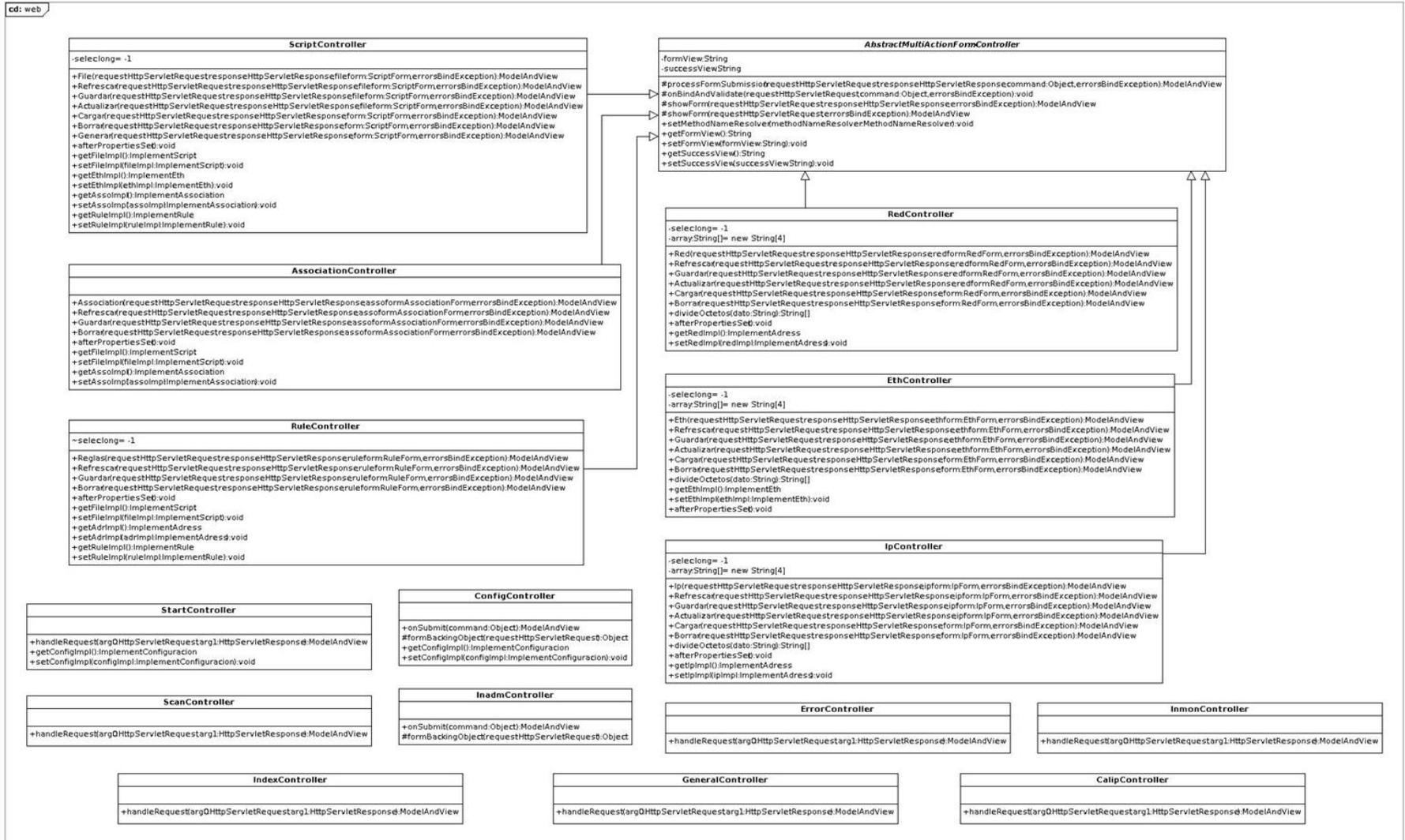


1.11 DIAGRAMAS DE CLASES

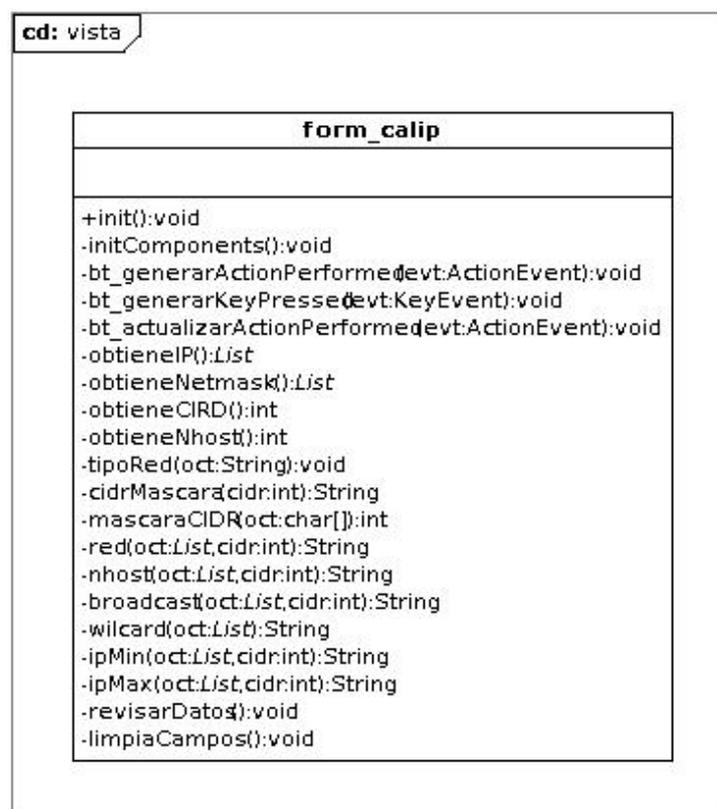
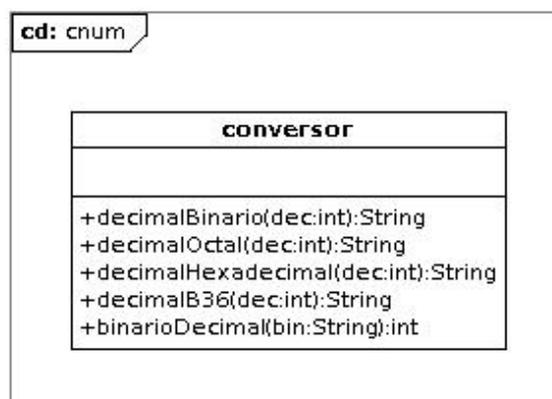




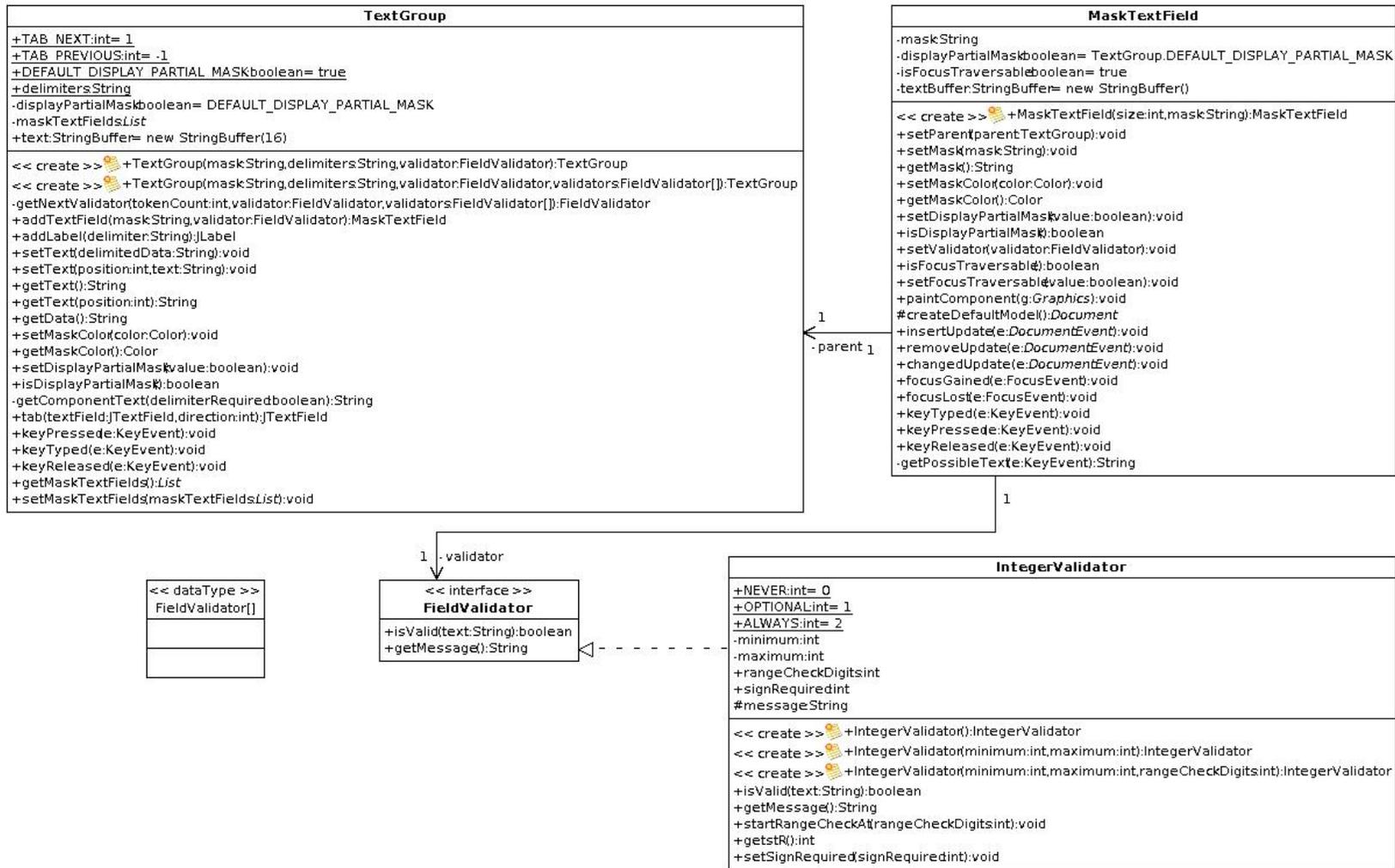


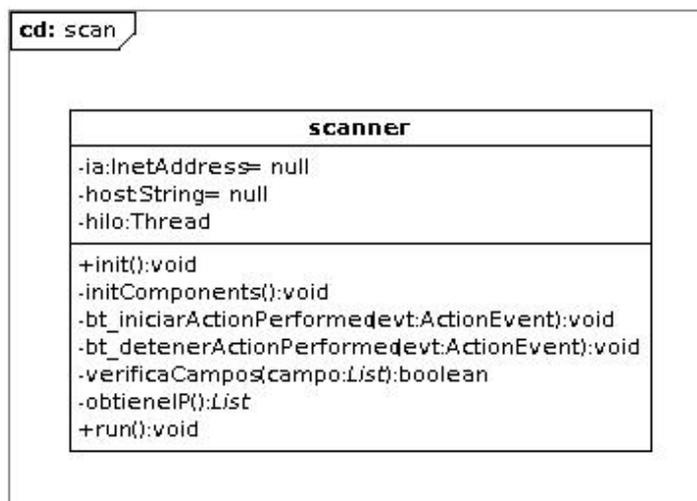


1.11.1 DIAGRAMA DE CLASES APLET CALCULADORA IP

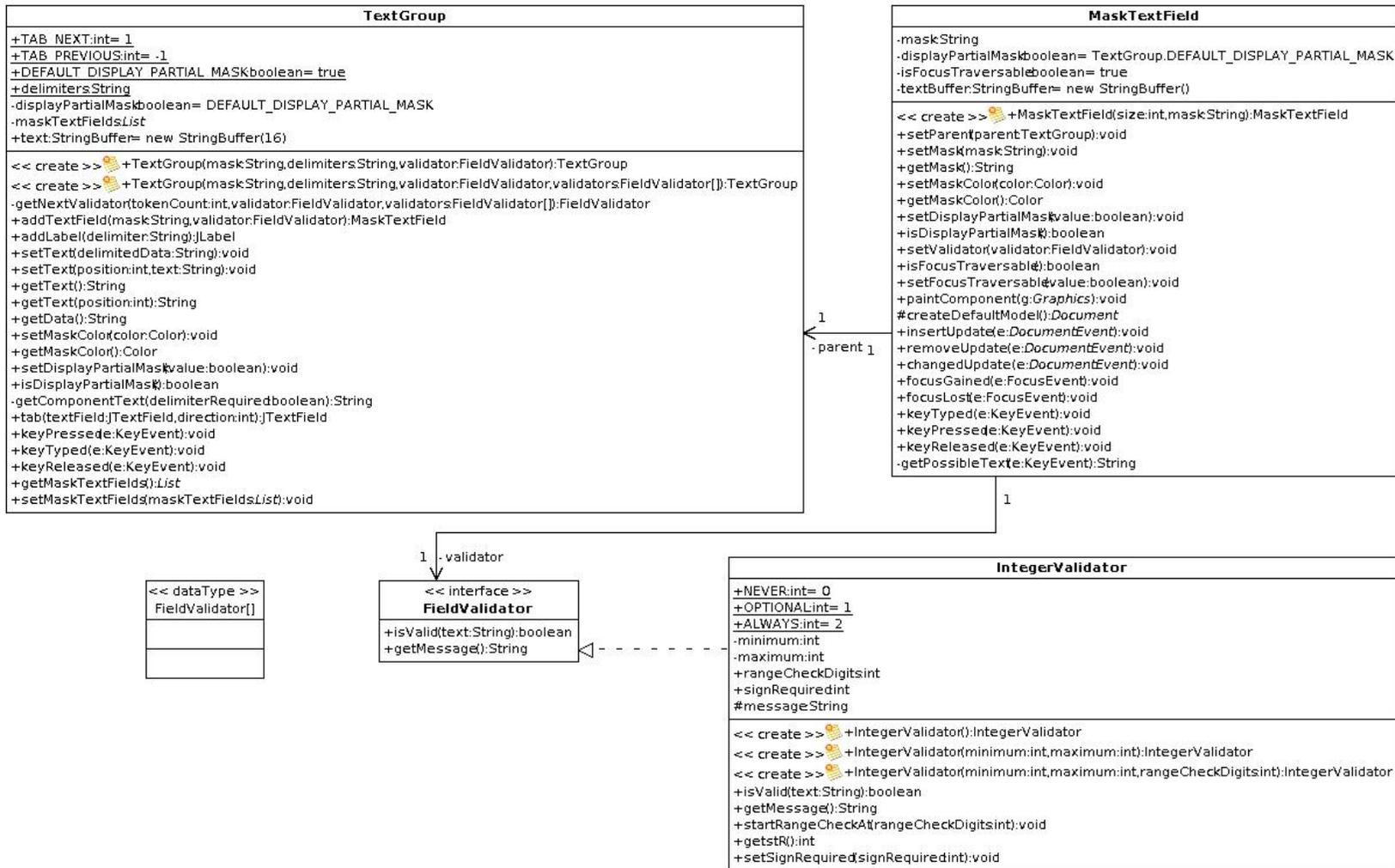


cd: text

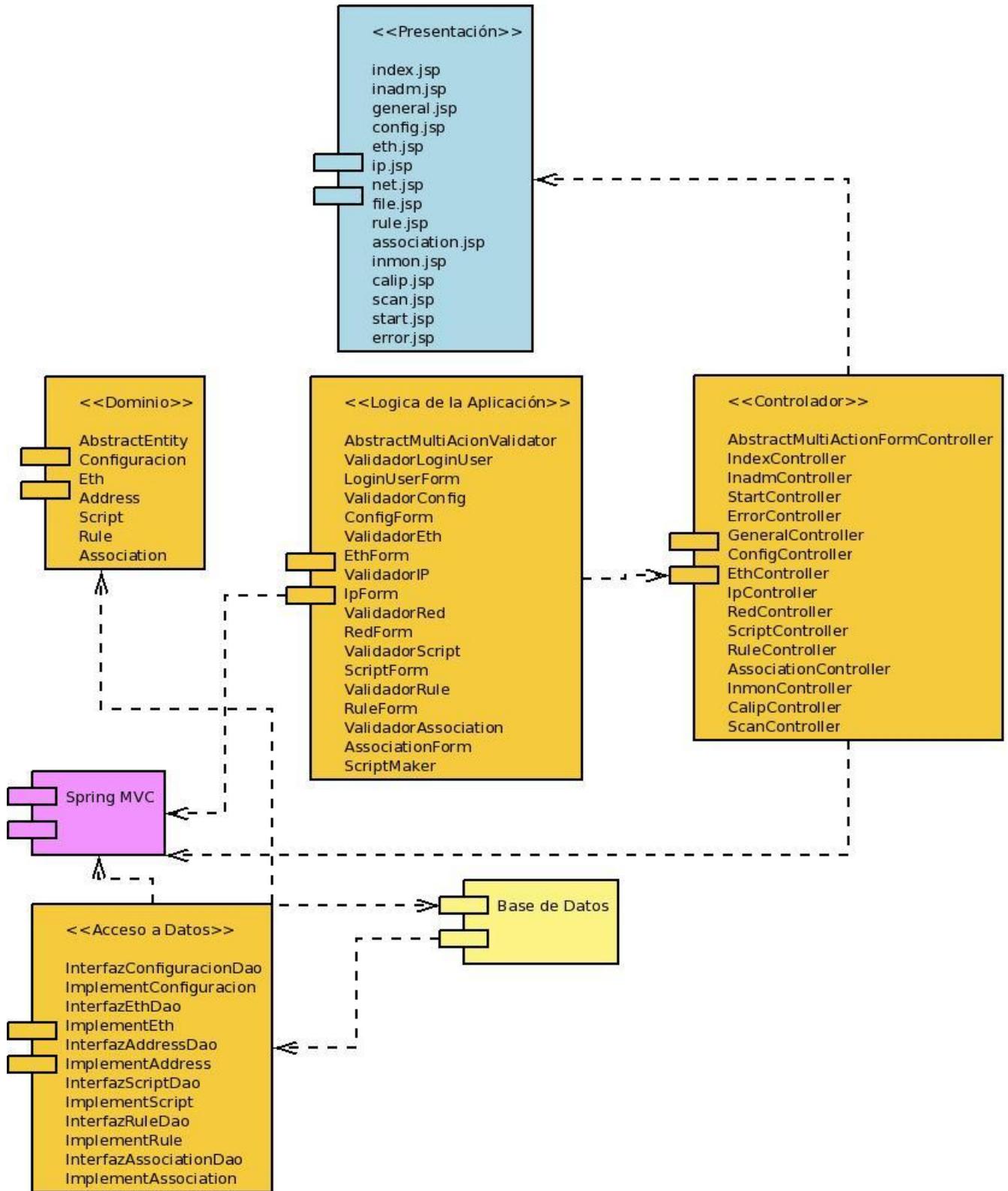


1.11.2 **DIAGRAMA DE CLASES APLET ESCANER**

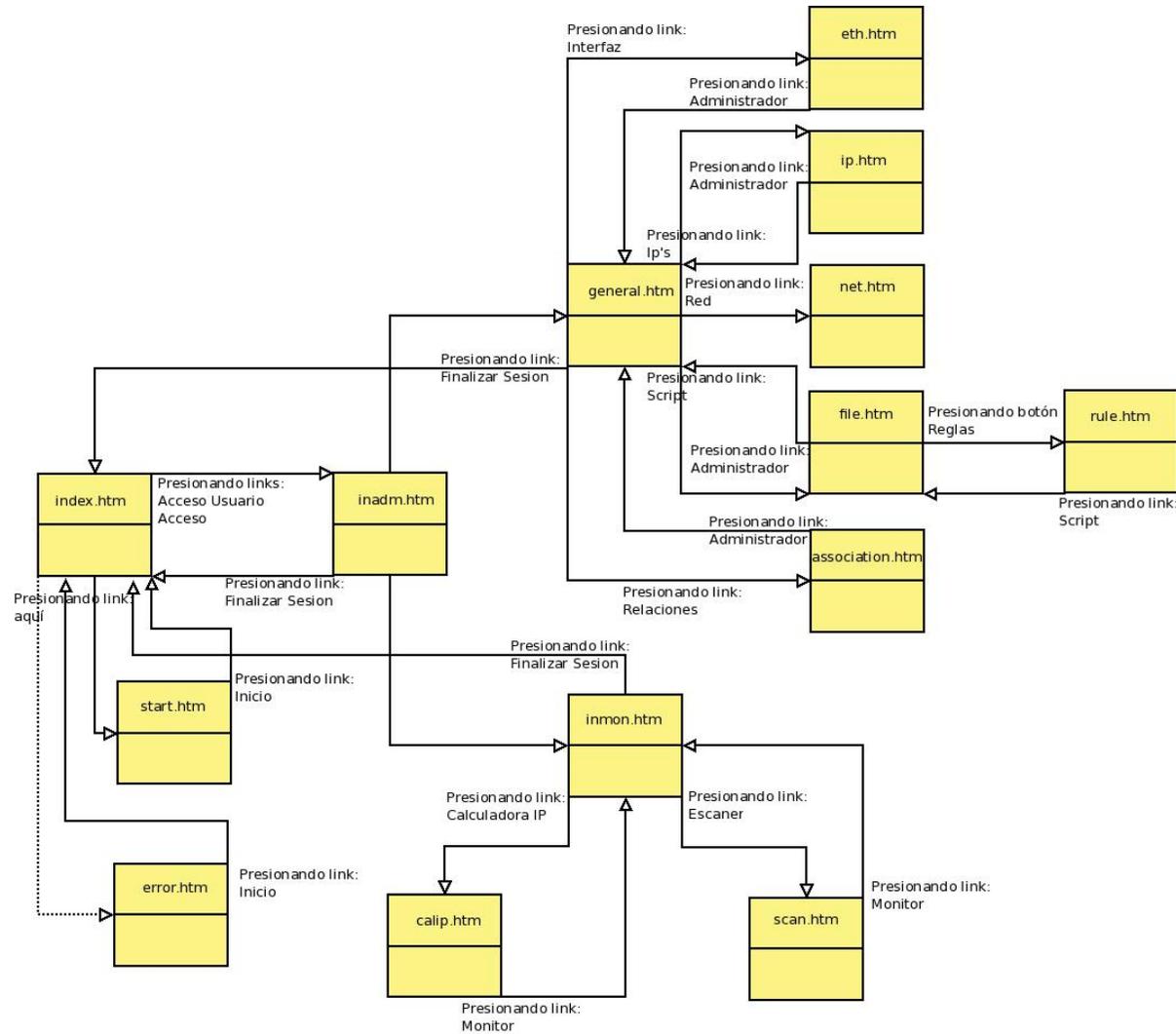
cd: text



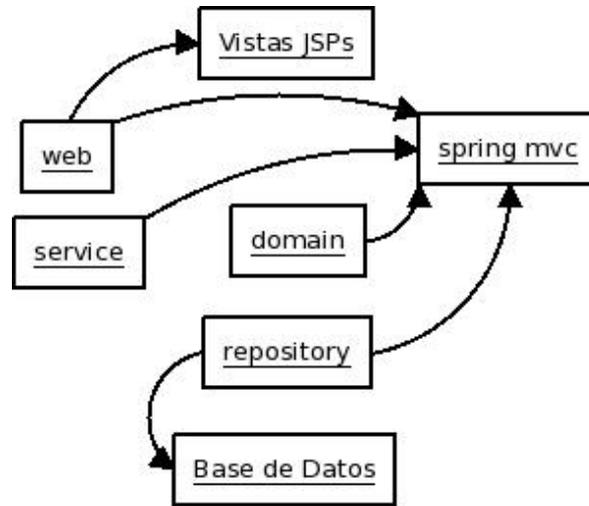
1.12 DIAGRAMA DE COMPONENTES



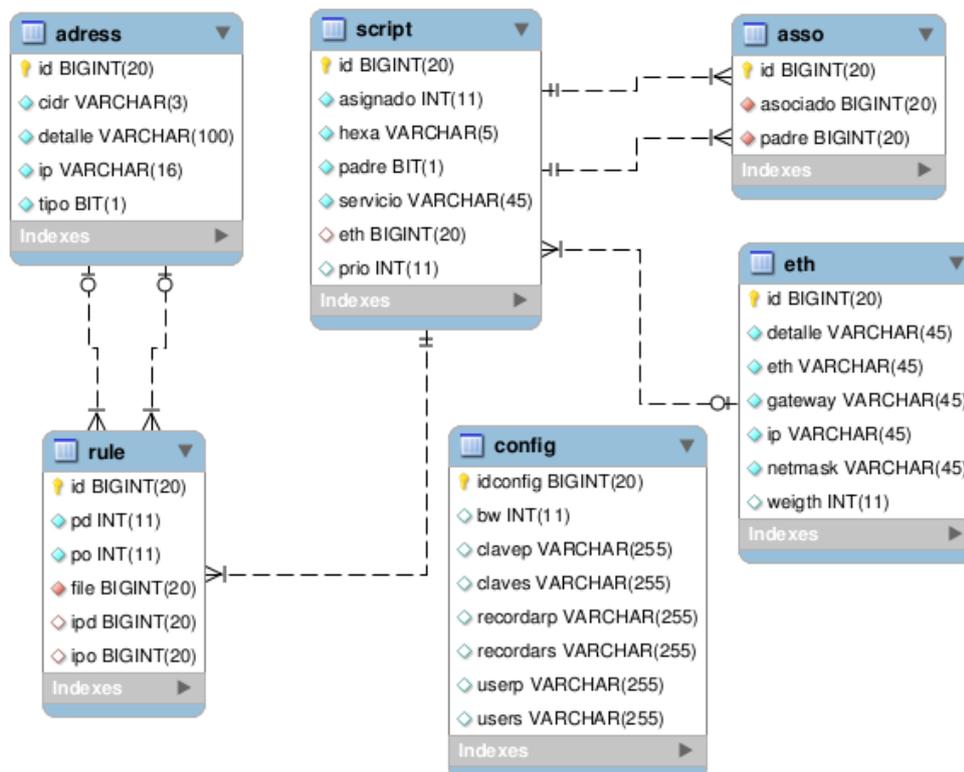
1.13 DIAGRAMA DE NAVEGABILIDAD



1.14 DIAGRAMA DE CAPAS DE LA APLICACIÓN



1.15 DIAGRAMA DE BASE DE DATOS



1.16 SCRIPT DE BASE DE DATOS

```
CREATE DATABASE IF NOT EXISTS bdcwr;
```

```
USE bdcwr;
```

```
DROP TABLE IF EXISTS `bdcwr`.`adress`;
```

```
CREATE TABLE `bdcwr`.`adress` (
```

```
  `id` bigint(20) NOT NULL auto_increment,
```

```
  `cidr` varchar(16) NOT NULL,
```

```
  `detalle` varchar(100) NOT NULL,
```

```
  `ip` varchar(3) NOT NULL,
```

```
  `tipo` bit(1) NOT NULL,
```

```

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `bdcwr`.`asso`;

CREATE TABLE `bdcwr`.`asso` (
  `id` bigint(20) NOT NULL auto_increment,
  `asociado` bigint(20) NOT NULL,
  `padre` bigint(20) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `FK2DD60E43D4F5B` (`padre`),
  KEY `FK2DD60EE0B23F7E` (`asociado`),
  CONSTRAINT `FK2DD60E43D4F5B` FOREIGN KEY (`padre`) REFERENCES
`script` (`id`),
  CONSTRAINT `FK2DD60EE0B23F7E` FOREIGN KEY (`asociado`)
REFERENCES `script` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

DROP TABLE IF EXISTS `bdcwr`.`config`;

CREATE TABLE `bdcwr`.`config` (
  `idconfig` bigint(20) NOT NULL auto_increment,
  `bw` int(11) default NULL,
  `clavep` varchar(255) default NULL,
  `claves` varchar(255) default NULL,
  `recordarp` varchar(255) default NULL,

```

```
`recordars` varchar(255) default NULL,  
  
`userp` varchar(255) default NULL,  
  
`users` varchar(255) default NULL,  
  
PRIMARY KEY (`idconfig`)  
  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;  
  
DROP TABLE IF EXISTS `bdcwr`.`eth`;  
  
CREATE TABLE `bdcwr`.`eth` (  
  
  `id` bigint(20) NOT NULL auto_increment,  
  
  `detalle` varchar(45) NOT NULL,  
  
  `eth` varchar(45) NOT NULL,  
  
  `gateway` varchar(45) NOT NULL,  
  
  `ip` varchar(45) NOT NULL,  
  
  `netmask` varchar(45) NOT NULL,  
  
  `weigth` int(11) NOT NULL,  
  
  PRIMARY KEY (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
DROP TABLE IF EXISTS `bdcwr`.`rule`;  
  
CREATE TABLE `bdcwr`.`rule` (  
  
  `id` bigint(20) NOT NULL auto_increment,  
  
  `pd` int(11) NOT NULL,  
  
  `po` int(11) NOT NULL,  
  
  `file` bigint(20) NOT NULL,
```

```
`ipd` bigint(20) default NULL,  
  
`ipo` bigint(20) default NULL,  
  
PRIMARY KEY (`id`),  
  
KEY `FK3596FCDF3DCF28` (`ipo`),  
  
KEY `FK3596FCDF3DCF1D` (`ipd`),  
  
KEY `FK3596FCFE155ED1` (`file`),  
  
CONSTRAINT `FK3596FCDF3DCF1D` FOREIGN KEY (`ipd`) REFERENCES  
`adress` (`id`),  
  
CONSTRAINT `FK3596FCDF3DCF28` FOREIGN KEY (`ipo`) REFERENCES  
`adress` (`id`),  
  
CONSTRAINT `FK3596FCFE155ED1` FOREIGN KEY (`file`) REFERENCES  
`script` (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
DROP TABLE IF EXISTS `bdcwr`.`script`;  
  
CREATE TABLE `bdcwr`.`script` (  
  
  `id` bigint(20) NOT NULL auto_increment,  
  
  `asignado` int(11) NOT NULL,  
  
  `hexa` varchar(5) NOT NULL,  
  
  `padre` bit(1) NOT NULL,  
  
  `prio` int(11) NOT NULL,  
  
  `servicio` varchar(45) NOT NULL,  
  
  `eth` bigint(20) NOT NULL,  
  
  PRIMARY KEY (`id`),
```

```
KEY `FKC9E5D0CBB5CB56E8` (`eth`),  
  
CONSTRAINT `FKC9E5D0CBB5CB56E8` FOREIGN KEY (`eth`) REFERENCES  
`eth` (`id`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2 CONCLUSIONES

Luego de finalizar con el desarrollo de la herramienta, en base a una extensa fundamentación teórica que comprende la presente tesis, y como producto de los conocimientos, experiencias, criterios y certezas que se ha podido obtener de la presente, he llegado a las siguientes conclusiones:

- La herramienta necesita usar las propiedades que nos da el Sistema Operativo Linux, cuyo Kernel, no puede ser menor a 2.4.x, en cualquier distribución.
- El proyecto de desarrollo de la presente tesis, permite controlar adecuadamente el ancho de banda con el que contamos, en nuestra institución, o empresa.
- Es necesario tener mejores conocimientos en la teoría como en el manejo de redes, en especial al tema relacionado con el subnetting, que a criterio personal es el eje central de redes.

- La herramienta desarrollada, sirve de apoyo a la labor de el o los administradores de red , así como también puede ser utilizada por el o los catedráticos de redes como una fuente de apoyo didáctico.
- La definición de QoS “Calidad de servicio es la capacidad de dar un buen servicio”, es un estándar que nos permite implementar, nuestros servicios, en el menor tiempo, y al mejor costo.
- El uso del Framework Spring 2.5, ha sido de gran ayuda en el desarrollo de la herramienta, ya que luego de estudiarlo y entenderlo permitió un desarrollo mas rápido y eficiente.
- El principal problema fue no haber encontrado suficiente documentación en español respecto al uso y aplicación de Spring 2.5.
- Las clases `AbstractMultiActionFormController.java` y `AbstractMultiActionValidator.java`, creadas por el estudiante Japonés Choon Whee y modificadas por el Inglés David Reepmeyer, me ayudaron a implementar una mejor solución en ciertas clases de la capa de control de la herramienta.
- La presente problemática me ha permitido enfocarme en una de las principales causas que se originan en el uso indebido del ancho de banda, y de las herramientas que nos permitan ayudar a monitorear y controlar que el uso que se está dando a nuestros recursos sean los planificados.

1 RECOMENDACIONES

Las recomendaciones que a continuación escribo, van dirigidas a los diferentes sectores que he involucrado en el desarrollo de la presente tesis.

- **A LOS ESTUDIANTES DE LA CARRERA DE INGENIERÍA EN SISTEMAS**, preocuparse por investigar e implementar los nuevos frameworks que presentan las comunidades de los diferentes lenguajes de programación, ya que éstos han sido pensados para ayudar a los programadores a crear soluciones que se adaptan mejor a los requerimientos de las empresas.
- A usar Spring, preocupándose por investigar las nuevas soluciones que se implementan al framework, para así manejar de un modo mas eficiente el desarrollo de sus proyectos.
- **A LOS DOCENTES**, que se invierta más tiempo en el control de calidad de los proyectos que los estudiantes desarrollan.
- A tener una base de conocimiento de los proyectos que ya se han desarrollado y cómo éstos pueden ser una fuente de apoyo para los nuevos proyectos.
- A solicitar a los estudiantes el estudio y la implementación de estándares de calidad al momento de desarrollar las aplicaciones software.

- **AL ADMINISTRADOR DE LA RED**, para que haga uso de la presente herramienta, que le ayudará a controlar de una mejor manera el ancho de banda con el que cuenta su empresa.
- **A LAS AUTORIDADES DE LA CARRERA DE INGENIERÍA EN SISTEMAS**, para que se inviertan los recursos necesarios que permitan a los estudiantes y docentes, tener un departamento de soporte en base a soluciones del Sistema Operativo Linux, con lo cual se va a mejorar la calidad de profesionales de nuestra Carrera y se va a dar una mejor opción en cuanto a costo y mantenimiento.
- A que se realicen los estudios y se inviertan los recursos que permitan, realizar la implementación de los conceptos de subnetting, en la red del Área de la Energía las Industrias y los Recursos Naturales no Renovables.
- **A LA UNIVERSIDAD NACIONAL DE LOJA, ESPECIALMENTE AL ÁREA DE LA ENERGÍA LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES**, para que los proyectos, como el presente, tengan una mayor incentivo, ya que se lo ha desarrollado haciendo uso de únicamente de software libre y en plataforma Linux, desde su proyecto hasta la culminación, encontrando herramientas de alta calidad y sin costo.

1 BIBLIOGRAFÍA

Libros:

Configurando el servidor de correo, Mail-Administrator-HOWTO Networking Complete , 2nd. Edition, Ed. Sybex.

TCP/IP, T. Parker and M. Sportack , Ed. SAMS, 2000, Linux Advanced Routing & Traffic Control HOWTO , BERT Hubert, Netherlabs BV.

Sitios Web:

BARRIOS DUEÑAS, Joel. 2009 Cómo utilizar CBQ. [en línea] España,[<http://www.alcancelibre.org/staticpages/index.php/como-cbq>], [Actualizado: 19 Octubre 2009].

FERNÁNDEZ GARRIDO, Daniel . 2008, Configuración de un firewall en Linux con iptables. [en línea] España, [<http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=14>] ,[Actualizado: 19 Octubre 2009].

LAGOS TORRES, Manuel. 2009, Introducción al diseño con patrones. [en línea] España, [<http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=29>], [[Actualizado: 19 Octubre 2009](#)].

LA GRECA, Pablo. 2009. Epidata Consulting. Base de Conocimiento Spring Web MVC. [en línea] Argentina. [<http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=Spring+Web+MVC>], [Actualizado: 19 Octubre 2009].

KEDEKE. 2008, Iptables y NetFilter el firewall de Linux. [en línea] España, [<http://www.estrellateyarde.es/equiv/firewall-en-linux>],[Actualizado: 19 Octubre 2009].

CHMIELEWSKI, Tomasz. 2001. Bandwidth Limiting HOWTO. [en línea], [<http://www.faqs.org/docs/Linux-HOWTO/Bandwidth-Limiting-HOWTO.html#CBQ>], [Actualizado: 19 Octubre 2009].

ASTUCIO, Sepio. 2002. Limitar el ancho de banda. [en línea], [http://es.geocities.com/sepio_astucio/documentos/ancho_banda.htm], [Actualizado: 19 Octubre 2009].

ANDERSON, Oskar. 2008. Linux & more. [en línea] Swenden, [<http://iptables-tutorial.frozentux.net/spanish/chunkyhtml/index.html>],[Actualizado: 19 Octubre 2009].

FUENTES, Alex, 2009. MVC en Srping [en línea] Chile, [http://www.jroller.com/afuentes/entry/mvc_en_spring],[Actualizado: 19 de Octubre 2009]

TCLGO. ROMERO JIJON, Juan Carlos. 2008. Estudio de Subnetting, Vlsn, CIDR y Comandos de Administración y Configuración de Routers [en línea] Machala, El Oro, Ecuador, [<http://www.monografias.com/trabajos35/subnetting-vlsm/subnetting-vlsm.zip>], [Actualizado: 19 Octubre 2009].

RUSELL, Paul. 2009, netfilter/iptables project, [en línea], [<http://www.netfilter.org>], [Actualizado: 19 Octubre 2009].

ALTADILL IZURRA, Pello Xabier. 2008. IPTABLES manual práctico, tutorial de iptables con ejemplos. [en línea] España, [<http://www.pello.info/>], [Actualizado: 19 Octubre 2009].

BARAJAS, Saulo. 2009. Curso de Protocolos TCP/IP. [en línea] España, [<http://www.saulo.net/pub/tcpip/b.htm>], [Actualizado: 19 Octubre 2009].

GOLUBEV, Paul. 2009. sourceforge CBQ.init traffic management script [en línea], [<http://sourceforge.net/projects/cbqinit/>], [Actualizado: 19 Octubre 2009].

Spring Framework. 2009. Interface 21, Capter 13 Web MVC Framework. [en línea], [<http://static.springframework.org/spring/docs/2.0.x/reference/mvc.html#mvc-controller-multiaction>], [Actualizado: 19 Octubre 2009].

Sun Microsystems, Inc. 2009. Front Controller. [en línea], [<http://java.sun.com/blueprints/patterns/FrontController.html>], [Actualizado: 19 Octubre 2009].

techFAQ. 2008. ¿Qué es una direccion Broadcast?. [en línea] USA, [<http://es.tech-faq.com/broadcast-address.shtml>], [Actualizado: 19 Octubre 2009].

WIKIPEDIA. 2009. Netfilter/Iptables [en línea], [<http://es.wikipedia.org/wiki/Netfilter/iptables>], [Actualizado: 19 Octubre 2009].

WIKIPEDIA. 2009. Packet Sniffer [en línea], [<http://es.wikipedia.org/wiki/Sniffer>], [Actualizado: 19 Octubre 2009].

HUANCA, Renan. 2009. Como usar el MultiActionController de Spring 2.5. [en línea] Bolivia, [<http://renidev.wordpress.com/>], [Actualizado: 19 Octubre 2009].

A N E X O S

UNIVERSIDAD NACIONAL DE LOJA

**Área de la Energía las Industrias y los Recursos
Naturales no Renovables.**

TÍTULO:

**“ANÁLISIS, DISEÑO E IMPLEMENTACION DE UNA
HERRAMIENTA WEB, EN PLATAFORMA LINUX, CON
ESTÁNDARES DE QoS, PARA EL CONTROL Y MONITOREO
DE TRÁFICO EN LA RED DEL ÁREA DE ENERGÍA,
INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES,
DE LA UNIVERSIDAD NACIONAL DE LOJA”**

ANTEPROYECTO DE TESIS

AUTOR

Jorge Cástulo Guerrón Eras.

DIRECTOR DE TESIS

Ing. Wilson Augusto Vélez Ludeña.

Loja – Ecuador

2009

2 PLANTEAMIENTO DEL PROBLEMA.

1.1 Tema.

“DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA WEB PARA EL CONTROL Y MONITOREO DE TRAFICO DE REDES LINUX, ALREDEDOR DE LA QUALITY OF SERVICE QoS”.

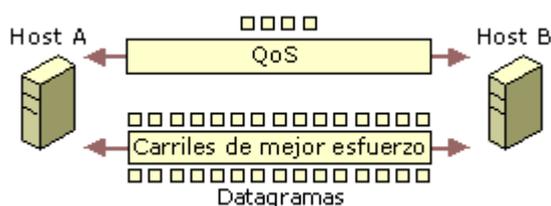
1.2 Situación Problemática.

1.2.1 Antecedentes.

El desarrollo y estandarización de los servicios de red ha sido conducido inicialmente por el IETF (Internet Engineer Task Force) a través de los diferentes RFC (Request For Comments) que involucran el uso de tecnologías para el aseguramiento de la calidad de los diferentes tipos de información que viajan en las redes de datos. El más relevante es el RFC 2386 el cual consiste en una recomendación para un marco de enrutamiento óptimo para contribuir al aseguramiento de la calidad en el trato de la información. Se destaca el RFC 2386 debido a que hoy en día en la red de redes - “*INTERNET*”, se maneja IP como protocolo de enrutamiento, a pesar que tiene en el encabezado de sus datagramas un campo para la identificación del tipo de información que transporta, no ofrece garantías de un trato adecuado de los datos al ser un protocolo no orientado a conexión. Además cabe destacar que IP nació antes de que las redes de datos tuvieran el suficiente ancho de banda para soportar los nuevos servicios que hoy en día requieren de una QoS para un funcionamiento adecuado y eficiente.

Calidad de servicio (QoS, Quality of Service) es un conjunto de requisitos de servicio que la red debe cumplir para asegurar un nivel de servicio adecuado para la transmisión de los datos. Estos requisitos de servicio se basan en estándares de funcionalidad QoS.

QoS permite que los programas en tiempo real optimicen el uso del ancho de banda de la red. Como QoS asegura cierto nivel de garantía de recursos de red suficientes, ofrece a una red compartida un nivel de servicio similar al de una red dedicada.



Una garantía de QoS indica un nivel de servicio que permite que un programa transmita datos a una velocidad especificada y los entregue en un periodo de tiempo dado.

Las tecnologías de QoS (*Quality of Service*, Calidad de Servicio) garantizan que se transmitirá cierta cantidad de datos en un tiempo dado (*throughput*). Una de las grandes ventajas de ATM (Asynchronous Transfer Mode – Modo de Transferencia Asíncrona) respecto de técnicas como el Frame Relay y Fast Ethernet, es que admite niveles de QoS. Esto permite que los proveedores de servicios ATM garanticen a sus clientes que el retardo de extremo a extremo no excederá un nivel específico de tiempo, o que garantizaran un ancho de banda específico para un servicio. Esto es posible de hacer marcando los paquetes que provengan de una IP determinada de los nodos conectados a un Gateway, (como por ejemplo la IP de un teléfono IP, según la puerta del router, etc...). Además que en los servicios satelitales da una nueva perspectiva en la utilización del ancho de banda, dando prioridades a las aplicaciones de extremo a extremo con una serie de reglas.

Una red IP esta basada en el envío de paquetes de datos, estos paquetes de datos tienen una cabecera que contiene información sobre el resto del paquete. Existe una parte del paquete que se llama TOS, en realidad esta parte esta pensada para llevar banderas o marcas. Lo que se puede hacer para darle prioridad a un paquete sobre el resto es marcar una de esas banderas (flags)

Para ellos el equipo que genera el paquete, por ejemplo un gateway de voz sobre IP, coloca una de esas banderas en un estado determinado y los dispositivos por donde pasa ese paquete luego de ser transmitido deben tener la capacidad para poder discriminar los paquetes para darle prioridad sobre los que no fueron marcados o los que se marcaron con una prioridad menor a los anteriores. De esta manera podemos generar prioridades altas a paquetes que requieren una cierta calidad de envío, como por ejemplo la voz o el video en tiempo real y menores al resto.

“QoS - Quality of Service - Calidad de servicio es la capacidad de dar un buen servicio”.

Iproute2 es un paquete de utilidades desarrollado por Alexey Kuznetsov. Este paquete reemplaza completamente las funcionalidades presentes en ifconfig, route, y arp y las extiende llegando a tener características similares a las provistas por dispositivos exclusivamente dedicados al ruteo y control de tráfico. Este paquete es un conjunto de herramientas muy potentes para administrar interfaces de red y conexiones en sistemas Linux. Este paquete lo podemos encontrar incluido en distribuciones de Debian y RedHat con versiones del kernel mayores a 2.2.

Muchas de las distribuciones Linux están preparadas para utilizar el paquete iproute2 gracias a los rediseñados subsistemas de red implementados en los núcleos 2.2 y superiores. Los viejos comandos 'ifconfig' y 'route' están obsoletos a causa de su comportamiento defectuoso e imprevisto en estos núcleos. Este nuevo código de encaminamiento (routing) y filtrado trae consigo muchas ventajas y funcionalidades de las que no se disponía antes.

1.2.2 Problemática.

Cuando tenemos mas de una maquina conectada a Internet a través de la misma salida suele pasar que si una de ellas empieza a consumir mucho ancho de banda (una descarga grande, p2pm, paginas web pesadas, etc), esta maquina acapara la conexión haciendo que las demás maquinas no tenga casi ancho de banda , o inclusive no tengan acceso al Internet, y la latencia (tiempo de respuesta de la conexión) sea penosa.

En GNU/Linux el control de ancho de banda tiene 2 partes: drivers o módulos en el kernel y herramientas en espacio de usuario. Estas herramientas se encuentran dentro del paquete iproute2, de las versiones con kernel 2.4 y 2.6.

En la actualidad y mas específicamente en nuestro entorno, no se conoce de herramientas web útiles para el control de trafico de redes en el sistema operativo Linux, que permitan a los usuarios en general o por lo menos a los usuarios que tienen conocimientos básicos de redes, aprender a utilizar un mecanismo de control y monitoreo para estas.

Este control debe estar habilitado para todos los usuarios que no tienen mucha experiencia con el sistema operativo Linux y que tampoco manejen la sintaxis de comandos tc (Traffic Control) del paquete iproute2, incorporado en las últimas versiones de los sistemas operativos Linux que trabajan con una versión de kernel 2.4 & 2.6.

Tomando en cuenta la necesidad de un control y optimización de los recursos de las redes, se propone el desarrollo del siguiente trabajo:

“DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA WEB PARA EL CONTROL Y MONITOREO DE TRAFICO DE REDES LINUX, ALREDEDOR DE LA QUALITY OF SERVICE QoS”.

1.3 Problema de Investigación.

No se ha difundido una herramienta web, creada en jsp, que permita realizar el control y monitoreo de trafico de redes en las versiones del Sistema Operativo Linux que traen las versiones de kernel 2.4 & 2.6.

1.3.1 Delimitación del Problema.

1.3.1.1 Espacio.

Redes LAN con acceso a Internet, que no tienen un control sobre el trafico de datos.

1.3.1.2 Tiempo.

El tiempo estimado para el investigar los requerimientos de la presente herramienta es de 6 meses.

1.3.1.3 Elementos de Observación.

Para realizar el desarrollo de la presente herramienta se tomará en cuenta los siguientes elementos de observación:

- Análisis de trafico de redes.
- Estudio y manejo de Iproute2.
- Estudio de estándares QoS, para herramientas de control.

- Estudio de los módulos del Sistema Operativo Linux que permiten realizar el control y monitoreo de redes.

1.4 Justificación.

La Universidad Nacional de Loja, es una entidad educativa de gran prestigio, a nivel nacional como en la región Sur del País, el sur de Colombia y el norte del Perú, la misma se a dividido en cinco áreas orientadas a la preparación de profesionales con un alto nivel crítico - investigativo que les permita aportar alternativas de solución en beneficio de la sociedad.

Entre las áreas que conforman esta prestigiosa Alma Mater, destacamos el Área de la Energía, Industrias y Recursos Naturales no Renovables cuyas actividades encaminan al crecimiento, enriquecimiento y desarrollo de entes sociales que reciben una educación científica, técnica y tecnológica en diferentes carreras tales como: Ingeniería en Sistemas, Geología Ambiental y Ordenamiento Territorial, Electromecánica, Electricidad y Electrónica.

La carrera de Ingeniería en Sistemas forma profesionales con conocimientos profundos de la estructura y particularidades del software, hardware, y para plasmar las enseñanzas vertidas durante los 5 años y 6 meses de estudio, me propongo poner en la practica los conocimientos impartidos, y planteo la ejecución de un proyecto que simplifique los procesos que se necesitan para realizar el control de trafico de redes en Linux.

1.4.1 Justificación Académica.

La presente herramienta de desarrollo satisface las expectativas académicas de nuestra Universidad ya que aplicaré todos los conocimientos adquiridos durante las diversas etapas de formación que he recibido a través de los años dentro de las aulas universitarias, como conocemos el objetivo de la Universidad Nacional de Loja a través del SAMOT, es propiciar el logro de aprendizajes significativos, en un proceso de vinculación permanente de la docencia, la investigación y la extensión, en base de las problemáticas y demandas de la sociedad.

Es así que, con el fin de demostrar que las técnicas y métodos implantados en nuestra universidad y aplicados por nuestros docentes, los dicentes nos encontramos en la capacidad de desarrollar proyectos que aportan gran desarrollo a la sociedad.

Además la documentación y herramienta que se pretende en el presente documento, dejará un rastro cognoscitivo que servirá de ayuda a alumnos, profesores y sin duda apoyará a futuras investigaciones. Dejando así comprobado que el desarrollo de la herramienta estoy en la capacidad de garantizar el título universitario que pretendo alcanzar, de la Carrera de Ingeniería en Sistemas, iniciando de este modo a escribir el nombre profesional que será la mejor carta de presentación al desempeñar las actividades profesionales.

1.4.2 **Justificación Técnica.**

Con el acelerado crecimiento de la ciencia informática, como el uso constante de mejores y efectivas herramientas que nos permitan manejar de un modo mas ágil y menos complicado los recursos con los que actualmente contamos, el desarrollo de la presente herramienta permitirá mejorar, el modo de aplicación de reglas tc (Traffic Control) para el manejo y control del trafico de redes Linux, por lo cual el presente proyecto es factible técnicamente ya que son varias las personas tanto jurídicas como naturales que cuentan con redes LAN y conexiones a Internet, pero que no conocen muy bien el manejo de los servicios que prestan en este caso sistemas operativos Linux, el software que servirá de base para la creación de la presente herramienta es totalmente gratuito, entre ellos enumeramos los siguientes:

- Sistema Operativo Linux, la distribución a utilizarse es Fedora 7 (<http://www.fedoraprojet.org>), la cual incluye la versión de kernel 2.6.x (<http://www.kernel.org>), y el paquete iproute2 que contiene las ordenes de tc (<http://www.developer.osdl.org/dev/iproute2>)
- Servidor-Web, con soporte para JSP(Java Server Page), Apache2, cooperando con servidor Tomcat, a través del módulo de conexión mod_jk 1.2 (<http://httpd.apache.org>).
- Jsdk 1.6, La misma que servira para el desarrollo de la aplicación, como para la inicialización del servidor Tomcat.

- Base de datos MySQL, se utilizará la versión 5.0.X (<http://www.mysql.com>)
- Framework Hibernate que servirá para el acceso a la base de datos.
- Framework JSF basado en el MVC (modelo, vista, controlador), que servirá para el desarrollo Web,.
- Umbrello, modelador UML.
- La utilización del citado software asegurará la rapidez en la programación, la seguridad en el manejo de datos, dando como resultado una herramienta de alta calidad.

1.4.3 Justificación Operativa.

Para la realización de la presente herramienta, cuanto con los elementos necesarios, documentación detallada en Internet, la capacidad para poder realizar el desarrollo, con los equipos, y la infraestructura necesaria, que asegure un desarrollo ininterrumpido de la herramienta.

1.4.4 Justificación Económica

Gran cantidad o un elevado número de proyectos presentan un reto económico al momento de ser presentados, es este el principal motivo, por el cual he realizado un estudio de factibilidad económica, del cual he logrado concluir que cuento con los suficientes recursos, tanto económicos, como humanos y bibliográficos, que me permitirán realizar con éxito la culminación del desarrollo de la presente herramienta. Ya que el software que voy a utilizar es un su totalidad es libre.

1.5 Objetivos.

1.5.1 Objetivo General.

Desarrollar e implementar una herramienta web que permita realizar el control de tráfico de redes Linux.

1.5.2 Objetivos Específicos.

- Recopilar y describir la información técnica referente al tema de Iproute2.
- Construir una interface amigable alrededor de la QoS.

- Construir un modulo de configuración de los servicios de control y monitoreo de la herramienta.
- Construir un modulo de control de seguridad de los servicios de la herramienta.
- Crear un modulo de usuarios, para la herramienta.
- Crear un modulo de monitoreo, para la presentación de los datos.
- Construir una Base de Datos que permita integrar nuevos módulos en el sistema, y que permita generar un historial de la misma.
- Realizar un análisis del desempeño de la herramienta.
- Implementar la herramienta en el Área de Energía, Industrias y Recursos Naturales no renovables, para su uso y provecho.

2 MARCO TEÓRICO.

2.1 Declaratoria Teórica.

1 Iproute2.

1.1 Introducción a iproute2

1.1.1 ¿Por qué iproute2?

La mayoría de las distribuciones de Linux, y la mayoría de los UNIX, usan actualmente los venerables comandos arp, ifconfig y route Aunque funcionan, muestran cierto comportamiento inesperado a partir de Linux 2.2. Por ejemplo, los túneles GRE son parte integral del enrutado hoy día, pero precisan herramientas completamente diferentes.

Con iproute2, los túneles son una parte integral del juego de herramientas. Los núcleos Linux 2.2 y superiores incluyen un subsistema de red completamente rediseñado. Este nuevo código de red proporciona a Linux un rendimiento y características con poca competencia en el panorama general de los SO. En realidad, el nuevo código de enrutado, filtrado y clasificación tiene más

posibilidades que el que proporcionan muchos enrutadores y cortafuegos dedicados y productos de control de tráfico.

Según se inventan nuevos conceptos de red, la gente encuentra maneras de emplastarlos encima de la infraestructura existente en los SO. Este continuo apilamiento de código no deseado ha llevado a código de red lleno de comportamientos extraños, muy parecido a lo que sucede con los idiomas humanos. En el pasado, Linux emuló la forma de SunOS de gestionar muchas de estas cosas, pero no era ideal.

Esta nueva infraestructura hace posible expresar claramente características que antes estaban más allá del alcance de Linux.

Linux tiene un sistema sofisticado para proporcionar ancho de banda llamado Traffic Control. Este sistema soporta varios métodos de clasificación, priorizado, compartición y limitación tanto de tráfico entrante como saliente.

1.1.1 Reglas (base de datos de normas de rutado).

Si se tiene una red grande, probablemente tengamos que encargarnos de las necesidades de diferentes personas, que deberían ser servidas de forma diferente. La base de datos de normas de rutado (routing policy database) nos permite hacerlo teniendo varios conjuntos de tablas de rutado.

Si se quiere usar esta característica, asegúremonos de que compila su núcleo con las opciones «IP: advanced router» e «IP: policy routing».

Cuando el núcleo necesita tomar una decisión de encaminamiento, busca la tabla que necesita consultar. Por defecto, hay tres tablas. La antigua herramienta "route" modifica las tablas principal y local, al igual que la herramienta ip (por defecto).

Las reglas por defecto:

```
[ahu@home ahu]$ ip rule      list
from all lookup      local
32766: from all lookup      main
```

```
32767: from all lookup      default
```

Aquí se lista la prioridad de todas las reglas. Vemos que todas son aplicables a todos los paquetes («from all»). Hemos visto anteriormente la tabla «main», mostrada por `ip route ls`, pero las tablas «local» y «default» son nuevas.

Si queremos hacer cosas interesantes, generaremos reglas que se refieran a diferentes tablas que nos permitirán saltarnos las reglas generales de rutado del sistema.

1.1.1.1 Normas de encaminamiento por origen sencillas

Tomando de nuevo un ejemplo real, tengo 2 cable módems (en realidad 3, pero acabé devolviendo uno), conectados a un router Linux NAT («masquerading»). La gente que vive aquí me paga por acceder a Internet. Suponga que uno de mis compañeros de casa sólo visita hotmail y desea pagar menos. Esto me parece bien, pero acabará usando el cable módem de menos prestaciones.

El cable módem «rápido» se conoce como 212.64.94.251 y es un enlace PPP a 212.64.94.1. El «lento» es conocido por varias IP, por ejemplo 212.64.78.148, y es un enlace a 195.96.98.253.

La tabla local:

```
[ahu@home ahu]$ ip route list table
```

```
local broadcast 127.255.255.255 dev lo proto kernel scope link src 127.0.0.1
```

```
local 10.0.0.1 dev eth0 proto kernel scope host src 10.0.0.1 broadcast 10.0.0.0
dev eth0 proto kernel scope link src 10.0.0.1
```

```
local 212.64.94.251 dev ppp0 proto kernel scope host src 212.64.94.251
broadcast 10.255.255.255 dev eth0 proto kernel scope link src 10.0.0.1
```

```
broadcast 127.0.0.0 dev lo proto kernel scope link src 127.0.0.1
```

```
local 212.64.78.148 dev ppp2 proto kernel scope host src 212.64.78.148
```

```
local 127.0.0.1 dev lo proto kernel scope host src 127.0.0.1
```

```
local 127.0.0.0/8 dev lo proto kernel scope host src 127.0.0.1
```

Montón de cosas obvias, pero que hace falta especificar en algún sitio. Bien, aquí están.

La tabla por defecto está vacía.

Veamos la tabla «main»:

```
[ahu@home ahu]$ ip route list table main
```

```
195.96.98.253 dev ppp2 proto kernel scope link src 212.64.78.148
```

```
212.64.94.1 dev ppp0 proto kernel scope link src 212.64.94.251
```

```
10.0.0.0/8 dev eth0 proto kernel scope link src 10.0.0.1
```

```
127.0.0.0/8 dev lo scope link
```

```
default via 212.64.94.1 dev ppp0
```

Ahora generaremos una nueva regla que llamaremos «John», por nuestro hipotético compañero. Aunque podemos trabajar con números, es mucho más sencillo añadir nuestras tablas a `/etc/iproute2/rt_tables`.

```
# echo 200 Buda.com.ec >> /etc/iproute2/rt_tables
```

```
# ip rule add from 10.0.0.10 table Buda.com.ec
```

```
# ip rule ls
```

```
0:    from all lookup local
```

```
32765: from 10.0.0.10 lookup John
```

```
32766: from all lookup main
```

```
32767: from all lookup default
```

Ahora todo lo que queda es generar la tabla John, y refrescar la caché de rutas:

```
# ip route add default via 195.96.98.253 dev ppp2 table Buda.com.ec
```

```
# ip route flush cache
```

1.1.1.2 Encaminamiento con varios enlaces de salida/proveedores

La siguiente es una configuración común, en la que hay dos proveedores que conectan una red local (o incluso una única máquina) a la gran Internet.

Normalmente surgen dos preguntas dada esta configuración.

1.1.1.2.1 Acceso dividido.

La primera es cómo enrutar respuestas a paquetes que vienen de un proveedor particular, por ejemplo Proveedor 1, de vuelta por el mismo camino. Primero establezcamos algunos nombres simbólicos. Digamos que \$IF1 es el nombre de la primera interfaz (if1 en la figura) y \$IF2 el nombre de la segunda. Sean entonces \$IP1 la dirección IP asociada con \$IF1 y \$IP2 la IP asociada con \$IF2. Luego, digamos que \$P1 es la dirección IP de la pasarela a Proveedor 1, y \$P2 la IP de la pasarela a Proveedor 2. Por último, \$P1_NET será la red IP donde está \$P1, y \$P2_NET la red IP donde está \$P2.

Creamos dos tablas de encaminamiento adicionales, llamémoslas T1 y T2. Las añadimos a /etc/iproute2/rt_tables. Entonces las configuramos de la siguiente manera:

```
ip route add $P1_NET dev $IF1 src $IP1 table T1
```

```
ip route add default via $P1 table T1
```

```
ip route add $P2_NET dev $IF2 src $IP2 table T2
```

```
ip route add default via $P2 table T2
```

Nada espectacular; simplemente hemos montado una ruta hacia una pasarela, y una ruta por defecto mediante ella, tal como sería el caso con un único proveedor, pero ponemos las rutas en tablas separadas, una por proveedor. Observe que basta la ruta hacia la red, ya que le indica cómo encontrar cualquier

máquina dentro de esa red, lo que incluye la pasarela, como se especificó anteriormente. Después configuramos la tabla de rutas principal (main). Es una buena idea encaminar las cosas a vecinos directos mediante la interfaz conectada a ese vecino. Observe las opciones «src», que se aseguran que se escoge la dirección IP correcta.

```
ip route add $P1_NET dev $IF1 src $IP1
```

```
ip route add $P2_NET dev $IF2 src $IP2
```

Luego, la ruta por defecto preferente:

```
ip route add default via $P1
```

A continuación, configuramos las reglas de encaminamiento. Estas son las que escogen qué tabla de rutas se usa. Querrá asegurarse de que encamina por una interfaz dada si ya tenemos una dirección de origen correspondiente:

```
ip rule add from $IP1 table T1
```

```
ip rule add from $IP2 table T2
```

Estas órdenes se aseguran de que todas las respuestas al tráfico proveniente de una interfaz en particular será contestado por esta interfaz.

Esta es la configuración más básica. Funcionará para todos los procesos que estén funcionando en el propio router, y para la red local, si está enmascarada, y si no, entonces puede que tenga un espacio IP de ambos proveedores, o que vaya a enmascarar la salida por uno de los proveedores. En ambos casos, querrá añadir reglas escogiendo por cual proveedor encaminar basándose en las direcciones IP de las máquinas en la red local.

1.1.1.3 Equilibrio de carga

La segunda pregunta es cómo equilibrar el tráfico que va por los dos proveedores. En realidad no es difícil si ya ha configurado un acceso dividido como se indicó previamente.

En lugar de escoger uno de los proveedores como la salida por defecto, configuraremos la ruta por defecto para que sea multicamino (multipath route). Por defecto, el núcleo equilibrará las rutas sobre los dos proveedores. Esto se hace como sigue (una vez más, nos apoyamos en el ejemplo de la sección sobre acceso dividido):

```
ip route add default scope global nexthop via $P1 dev $IF1 weight 1 \
nexthop via $P2 dev $IF2 weight 1
```

Esto equilibrará las rutas sobre ambos proveedores. Los parámetros weight se pueden modificar para favorecer a un proveedor sobre el otro. Tenga en cuenta que el equilibrio no será perfecto, ya que se basa en rutas, y las rutas están en caché. Esto significa que las rutas usadas más a menudo siempre irán sobre el mismo proveedor.

1.1.2 GRE y otros túneles

Hay tres tipos de túneles en Linux. Están los túneles IP sobre IP, los túneles GRE y túneles que se realizan fuera del núcleo (como por ejemplo, PPTP).

1.1.2.1 Breve inciso sobre los túneles:

Los túneles se pueden usar para hacer varias cosas poco usuales y bastante interesantes. También pueden hacer que las cosas vayan horriblemente mal si no los configura bien. No ponga su ruta por defecto sobre un dispositivo de túnel a menos que sepa EXACTAMENTE lo que está haciendo :-). Más aún, los túneles incrementan la carga, porque necesitan un juego extra de cabeceras IP. Normalmente, esto significa 20 bytes por paquete, de manera que si el tamaño normal de un paquete (MTU) en una red es de 1500 bytes, un paquete enviado por un túnel sólo puede ser de 1480 bytes como mucho.

Esto no es necesariamente un problema, pero asegúrese de leer algo sobre fragmentación/reensamblaje de paquetes IP si planea conectar redes grandes con túneles. Oh, y por supuesto, la manera más rápida de cavar un túnel es cavar desde los dos extremos.

1.1.2.2 Túneles IP sobre IP

Este tipo de túneles lleva disponible en Linux mucho tiempo. Precisa dos módulos del núcleo, `ipip.o` y `new_tunnel.o`

Digamos que tenemos tres redes: las redes internas A y B, y una red intermedia C (o Internet, por ejemplo). De manera que tenemos la red A:

```
network 10.0.1.0
```

```
netmask 255.255.255.0
```

```
router 10.0.1.1
```

El router tiene la dirección 172.16.17.18 en la red C. y la red B:

```
network 10.0.2.0
```

```
netmask 255.255.255.0
```

```
router 10.0.2.1
```

Cuyo router tiene la dirección 172.19.20.21 en la red C.

Hasta donde le concierne a C, asumimos que pasará cualquier paquete que vaya de A a B y viceversa. Incluso puede usar Internet para esto.

Esto es lo que haremos:

Primero, asegurarnos de que los módulos están instalados:

```
insmod ipip.o
```

```
insmod new_tunnel.o
```

Luego, en el router de la red A, hacemos lo siguiente:

```
ifconfig tunl0 10.0.1.1 pointopoint 172.19.20.21
```

```
route add -net 10.0.2.0 netmask 255.255.255.0 dev tunl0
```

Y en el de la red B:

```
ifconfig tunl0 10.0.2.1 pointopoint 172.16.17.18
```

```
route add -net 10.0.1.0 netmask 255.255.255.0 dev tunl0
```

Y si ha dejado de usar el túnel:

```
ifconfig tunl0 down
```

Listo, ya lo tiene. Sin embargo, no puede enviar tráfico de difusión (broadcast) o Ipv6 mediante un túnel IP-sobre-IP. Simplemente puede conectar dos redes IPv4 que normalmente no podrían comunicarse entre ellas; eso es todo. En lo que respecta a la compatibilidad, este código lleva ahí mucho tiempo, de manera que es compatible hasta con los núcleos 1.3. Los túneles IP-sobre-IP de Linux no funcionan con otros sistemas operativos o routers, hasta donde yo sé. Es sencillo, y funciona. Uselo si lo necesita, en cualquier otro caso, use GRE.

1.1.2.3 Túneles GRE

GRE es un protocolo de tunelizado que desarrolló Cisco originalmente, y que puede hacer unas cuantas cosas más que los túneles IP-sobre-IP. Por ejemplo, puede transportar tráfico multicast e IPv6 sobre un túnel GRE.

En Linux, necesitará el módulo ip_gre.o.

1.1.2.4 Túneles Ipv4

Primero hagamos un túnel Ipv4:

Digamos que tenemos 3 redes: las redes internas A y B, y una red intermedia C (por ejemplo, Internet).

De manera que tenemos la red A:

```
network 10.0.1.0
```

```
netmask 255.255.255.0
```

```
router 10.0.1.1
```

El router tiene la dirección 172.16.17.18 en la red C. Llamaremos a esta red neta (ok, no es muy original)

Y la red B:

```
network 10.0.2.0
```

```
netmask 255.255.255.0
```

```
router 10.0.2.1
```

El router tiene la dirección 172.19.20.21 en la red C. Llamemos a esta red netb (seguimos con nuestra originalidad) Hasta donde concierne a la red C, asumiremos que dejará pasar los paquetes enviados de A a B y viceversa. Cómo y por qué, no nos interesa.

En el router de la red A, haremos lo siguiente:

```
ip tunnel add netb mode gre remote 172.19.20.21 local 172.16.17.18 ttl 255
```

```
ip link set netb up
```

```
ip addr add 10.0.1.1 dev netb
```

```
ip route add 10.0.2.0/24 dev netb
```

Miremos esto con más atención. En la línea 1, hemos añadido un dispositivo de túnel, y le hemos llamado netb (bastante obvio porque es a donde queremos llegar). Más aún, le hemos dicho que use el protocolo GRE (mode gre), que la dirección remota es 172.19.20.21 (el router en el otro extremo), que nuestros paquetes de túnel deberían ser originados por 172.16.17.18 (lo que permite a nuestro router tener varias direcciones IP en la red C y decidir cual usar para el tunelizado) y que el campo TTL del paquete debería establecerse en 255 (ttl 255).

La segunda línea habilita el dispositivo.

En la tercera línea le hemos dado a la recién nacida interfaz netb la dirección 10.0.1.1. Esto está bien para redes pequeñas, pero cuando empiece una expedición de zapadores (MUCHOS túneles), quizá debiera considerar usar otro rango de IP para las interfaces de túneles (en este ejemplo, podría usar 10.0.3.0). En la cuarta línea hemos establecido la ruta hacia la red B. Fíjese la notación diferente para la máscara de red. Si no está familiarizado con esta notación, así es como funciona:

escriba la máscara de red en forma binaria, y cuente todos los unos. Si no sabe cómo hacerlo, límitese a recordar que 255.0.0.0 es /8, 255.255.0.0 es /16 y 255.255.255.0 es /24. Ah, y 255.255.254.0 es /23, en caso de que tuviera curiosidad. Pero ya hemos tenido bastante de esto; veamos el router de la red B.

```
ip tunnel add neta mode gre remote 172.16.17.18 local 172.19.20.21 ttl 255
```

```
ip link set neta up
```

```
ip addr add 10.0.2.1 dev neta
```

```
ip route add 10.0.1.0/24 dev neta
```

Y cuando vaya a eliminar el túnel del router A:

```
ip link set netb down
```

```
ip tunnel del netb
```

Por supuesto, puede cambiar netb por neta para el router B.

1.1.2.5 Túneles Ipv6

Vea la sección 6 si quiere una pequeña introducción a las direcciones Ipv6. Vamos con los túneles.

Asumamos que tiene la siguiente red IPv6, y que quiere conectarse a la 6bone, o con un amigo.

```
Red 3ffe:406:5:1:5:a:2:1/96
```

Su dirección IPv4 es 172.16.17.18, y el router 6bone tiene la dirección Ipv4 172.22.23.24. ip tunnel add sixbone mode sit remote 172.22.23.24 local 172.16.17.18 ttl 255

```
ip link set sixbone up
```

```
ip addr add 3ffe:406:5:1:5:a:2:1/96 dev sixbone
```

```
ip route add 3ffe::/15 dev sixbone
```

Vamos a comentar esto. En la primera línea, hemos creado un dispositivo de túnel llamado sixbone. Le hemos dado modo sit (que es un túnel IPv6 sobre IPv4) y le dijimos dónde debe ir (remote) y de dónde viene (local). TTL se pone al máximo, 255.

Después activamos el dispositivo (up). Tras esto, añadimos nuestra propia dirección de red, y establecemos una ruta para 3ffe::/15 (que actualmente es la totalidad de la 6bone) a través del túnel.

Los túneles GRE son actualmente el tipo preferido de túneles. Es un estándar que está ampliamente adoptado fuera de la comunidad de Linux, y por tanto una Cosa Buena.

1.1.2.6 Túneles en espacio de usuario

Literalmente, hay docenas de implementaciones de túneles fuera del núcleo. Los más conocidos por supuesto son PPP y PPTP, pero hay mucho más (algunos propietarios, algunos seguros, otros que ni siquiera usan IP) y que realmente están más allá del ámbito de este Cómo.

1.1.3 Túneles Ipv6

Las direcciones IPv6 son, comparadas con las IPv4, realmente grandes: 128 bits frente a 32. Y nos proporciona justo lo que necesitamos: muchas, muchas direcciones IP: 340.282.266.920.938.463.463.374.607.431.768.211.465, para ser precisos. Aparte de esto, se supone que IPv6 (o IPng, de IP Next Generation) traerá una reducción el tamaño de las tablas de rutas de los router principales de Internet, una configuración más sencilla para el equipamiento, mejor seguridad en el nivel de IP y mejor soporte para QoS (calidad de servicio).

Escribir direcciones IPv6 puede ser una gran molestia. Por tanto, nos facilitaremos la vida siguiendo algunas reglas:

- No use ceros sin significado (a la izquierda). Igual que hace en Ipv4.
- Use dos puntos (:) para separara cada grupo de 16 bits o dos bytes.

- Cuando tenga muchos ceros consecutivos, puede abreviarlos con ::. Sólo puede hacer esto una vez por cada dirección, y sólo en grupos de 16 bits.
- La dirección 2002:836b:9820:0000:0000:0000:836b:9886 puede escribirse 2002:836b:9820::836b:9886, que es algo más sencilla.

Se pretende que IPv6 sea el sucesor de la actual IPv4. Debido a que es una tecnología relativamente nueva, no hay ninguna red mundial IPv6 nativa. Se está introduciendo la 6bone para posibilitar la aceleración del cambio.

Las redes IPv6 nativas se conectan unas a otras encapsulando el protocolo IPv6 en paquetes IPv4 y enviándolos mediante la infraestructura IPv4 ya existente desde un sitio IPv6 a otro.

Aquí es justo donde entran los túneles. Para poder usar IPv6, deberíamos tener un núcleo que lo admita. Hay muchos documentos buenos sobre cómo conseguirlo. Pero todo se reduce a unos pocos pasos:

- ✂ Obtenga una distribución Linux reciente, con una glibc adecuada.
- ✂ Luego, consiga una fuente actualizada del núcleo.

Si lo tiene todo, puede seguir adelante y compilar un núcleo con capacidades de Ipv6:

Vaya a /usr/src/linux y escriba:

```
make menuconfig
```

Escoja «Networking Options»

Marque «The IPv6 protocol», «IPv6: enable EUI-64 token format», «IPv6: disable provider based addresses»

En otras palabras, compile IPv6 dentro del núcleo. Una vez hecho esto, guarde su configuración y compile el núcleo.

CONSEJO: Antes de hacerlo, considere la posibilidad de editar el Makefile: EXTRAVERSION = -x ; --> ; EXTRAVERSION = -x-IPv6. Hay un montón de documentación sobre cómo compilar e instalar un núcleo, sin embargo este

documento se refiere a otras cosas. Si tiene problemas en esta etapa, vaya a buscar algo de documentación sobre la compilación del núcleo de acuerdo con sus propias especificaciones.

El fichero `/usr/src/linux/README` podría ser un buen sitio donde empezar. Tras haber hecho todo esto, y reiniciado con su nuevo núcleo, puede que quiera ejecutar `"/sbin/ifconfig -a"` y ver el nuevo "sit0-device". SIT significa Simple Internet Transition.

Puede felicitarse a sí mismo; ahora está un paso más cerca de IP, la Próxima Generación,

Ahora vamos con el siguiente paso. Quiere conectar su máquina, o incluso su LAN entera a otras redes que puedan trabajar con IPv6. Puede ser la «6bone», que está configurada especialmente para este propósito particular.

Asumamos que tiene la siguiente red IPv6: `3ffe:604:6:8::/64` y que quiere conectar con la 6bone, o con un amigo. Fíjese que la notación `/64` de subred funciona igual que con cualquier dirección IP normal. Su dirección IPv4 es `145.100.24.181` y el router de 6bone tiene la dirección Ipv4 `145.100.1.5`

```
# ip tunnel add sixbone mode sit remote 145.100.1.5 [local 145.100.24.181 ttl 255]
```

```
# ip link set sixbone up
```

```
# ip addr add 3FFE:604:6:7::2/126 dev sixbone
```

```
# ip route add 3ffe::0/16 dev sixbone
```

Comentemos esto. En la primera línea, hemos creado un dispositivo de túnel llamado sixbone. Le hemos dado el modo sit (que es tunelizado IPv6 sobre IPv4) y le hemos dicho dónde debe ir (remote) y de dónde viene (local). El TTL se establece al máximo, 255. Después activamos el dispositivo (up). Tras esto, añadimos nuestra propia dirección de red, y establecemos una ruta para `3ffe::/15` (que actualmente es la totalidad de la 6bone) a través del túnel. Si la máquina en la que trabaja es su pasarela IPv6, quizá le interesa añadir estas líneas:

```
# echo 1 >/proc/sys/net/ipv6/conf/all/forwarding
```

```
# /usr/local/sbin/radvd
```

En la última arrancamos radvd, que es (como zebra) un demonio anunciador de rutas, para dar soporte a las capacidades de autoconfiguración de IPv6. Búsquelo en su motor de búsqueda favorito si lo desea. Puede comprobar las cosas así:

```
# /sbin/ip -f inet6 addr
```

Si tiene radvd ejecutándose en la pasarela IPv6 y arranca una máquina Linux con capacidades IPv6 en su LAN, podrá disfrutar de los beneficios de la autoconfiguración de IPv6:

```
# /sbin/ip -f inet6 addr1: lo: <LOOPBACK,UP> mtu 3924 qdisc noqueue inet6
::1/128 scope host 3: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc
pfifo_fast qlen 100 inet6 3ffe:604:6:8:5054:4cff:fe01:e3d6/64 scope global
dynamic valid_lft forever preferred_lft 604646sec inet6
fe80::5054:4cff:fe01:e3d6/10 scope link
```

Podría ir más allá y configurar el programa bind para que trabaje con direcciones Ipv6.

El tipo A tiene un equivalente para IPv6: AAAA. El equivalente de in-addr.arpa es:

ip6.int. Hay mucha información disponible sobre este tema.

Hay disponible un número creciente de aplicaciones que pueden trabajar con Ipv6, incluyendo secure shell, telnet, inetd, el navegador Mozilla, el servidor web Apache y muchos otros. Pero se sale del ámbito de este documento sobre Rutado.

1.1.4 IPsec: IP segura sobre Internet

FIXME: el puesto de editor esta vacante. Mientras tanto, vea: El proyecto FreeS/WAN. Otra implementación de IPsec para Linux es Cerberus, de NIST. Sin embargo, sus páginas web parecen no haberse actualizado en al menos dos

años, y su versión se ha quedado atrás con respecto al núcleo de Linux. USAGI, que es una implementación alternativa de IPv6 para Linux, también incluye una implementación de IPSec, pero sólo sería para Ipv6.

2 **Java**

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos. Pero no se queda ahí, Java está desarrollándose incluso para distintos tipos de dispositivos además del ordenador como móviles, agendas y en general para cualquier cosa que se le ocurra a la industria.

2.1 **Objetivos de Diseño de Java**

2.1.1 *Lenguaje familiar*

Java no sería un lenguaje totalmente nuevo, se parecería a lo que conocemos como C++, así que no le sería tan complicado recalcar en los programadores escépticos.

2.1.2 Lenguaje orientado a objetos

Para que un lenguaje pueda considerarse orientado a objetos debe soportar como mínimo las características de:

- Encapsulación.
- Herencia
- Polimorfismo
- Enlace dinámico.

2.1.3 Lenguaje robusto

Uno de los problemas más comunes en los lenguajes de programación es la posibilidad de escribir programas que pueden bloquear el sistema. Algunas veces este bloqueo puede ser inmediato, pero en otras ocasiones llega a aparecer inesperadamente porque, por ejemplo, la aplicación accede a zonas de memoria que no estaban siendo ocupadas por otros programas hasta ese momento. Un ejemplo claro de lenguaje no robusto es C. Al escribir código en C o C++ el programador debe hacerse cargo de la gestión de memoria de una forma explícita, solicitando la asignación de bloques a punteros y liberándolos cuando ya no son necesarios. En Java, los punteros, la aritmética de punteros y las funciones de asignación y liberación de memoria (`malloc()` y `free()`) no existen. En lugar de los punteros se emplean referencias a objetos, los cuales son identificadores simbólicos. El gestor de memoria de Java lleva una contabilidad de las referencias a los objetos. Cuando ya no existe una referencia a un objeto, éste se convierte en candidato para la recogida de basura (`garbage collection`).

2.1.4 Lenguaje de alto rendimiento (múltiples threads)

Una de las características del lenguaje es que soporta la concurrencia a través de threads. En ocasiones puede interesarnos dividir una aplicación en varios flujos de control independientes, cada uno de los cuales lleva a cabo sus funciones de

manera concurrente. Cuando los distintos flujos de control comparten un mismo espacio lógico de direcciones, se denominan threads.

2.1.5 Lenguaje Portable.

El principal objetivo de los diseñadores de Java, y dado el gran crecimiento de las redes en los últimos años, fue el de desarrollar un lenguaje cuyas aplicaciones una vez compiladas pudiesen ser inmediatamente ejecutables en cualquier máquina y sobre cualquier sistema operativo. Por ejemplo, un programa desarrollado en Java en una estación de trabajo Sun que emplea el sistema operativo Solaris, debería poderse llevar a un PC que utilice sistema operativo Windows NT.

2.1.6 Lenguaje lo más simple posible

Los diseñadores de Java trataron de mantener las facilidades básicas del lenguaje en un mínimo y proporcionar un gran número de extras con las librerías de clases.

2.1.7 Lenguaje seguro

Se pretendía construir un lenguaje de programación que fuese seguro, esto es, que no pudiera acceder a los recursos del sistema de manera incontrolada. Por este motivo se eliminó la posibilidad de manipular la memoria mediante el uso de punteros y la capacidad de transformación de números en direcciones de memoria (tal y como se hace en C) evitando así todo acceso ilegal a la memoria. Esto se asegura porque el compilador Java efectúa una verificación sistemática de conversiones.

2.2 Componentes

Librerías de Java, que son el resultado de compilar el código fuente desarrollado por quien implementa la JRE, y que ofrecen apoyo para el desarrollo en Java. Algunos ejemplos de estas librerías son:

- ✓ Las librerías centrales, que incluyen:
 - Una colección de librerías para implementar estructuras de datos como listas, arrays, árboles y conjuntos.
 - Librerías para análisis de XML.
 - Seguridad.
 - Librerías de internacionalización y localización.

- ✓ Librerías de integración, que permiten la comunicación con sistemas externos. Estas librerías incluyen:
 - La API para acceso a bases de datos JDBC (Java DataBase Connectivity).
 - La interfaz JNDI (Java Naming and Directory Interface) para servicios de directorio.
 - RMI (Remote Method Invocation) y CORBA para el desarrollo de aplicaciones distribuidas.

- ✓ Librerías para la interfaz de usuario, que incluyen:
 - El conjunto de herramientas nativas AWT (Abstract Windowing Toolkit), que ofrece componentes GUI (Graphical User Interface), mecanismos para usarlos y manejar sus eventos asociados.
 - Las librerías de Swing, construidas sobre AWT pero ofrecen implementaciones no nativas de los componentes de AWT.
 - APIs para la captura, procesamiento y reproducción de audio.

- Una implementación dependiente de la plataforma en que se ejecuta de la máquina virtual de Java (JVM), que es la encargada de la ejecución del código de las librerías y las aplicaciones externas.
- Plugins o conectores que permiten ejecutar applets en los navegadores Web.
- Java Web Start, para la distribución de aplicaciones Java a través de Internet.
- Documentación y licencia.

1 **FRAMEWORK WEB**

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que nos pueda ocurrir.

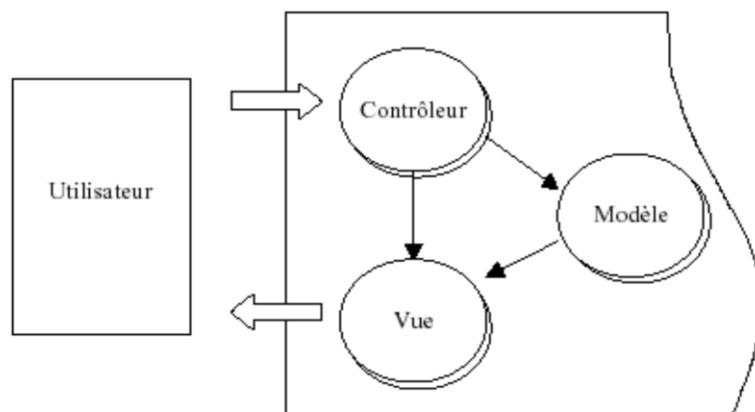
En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

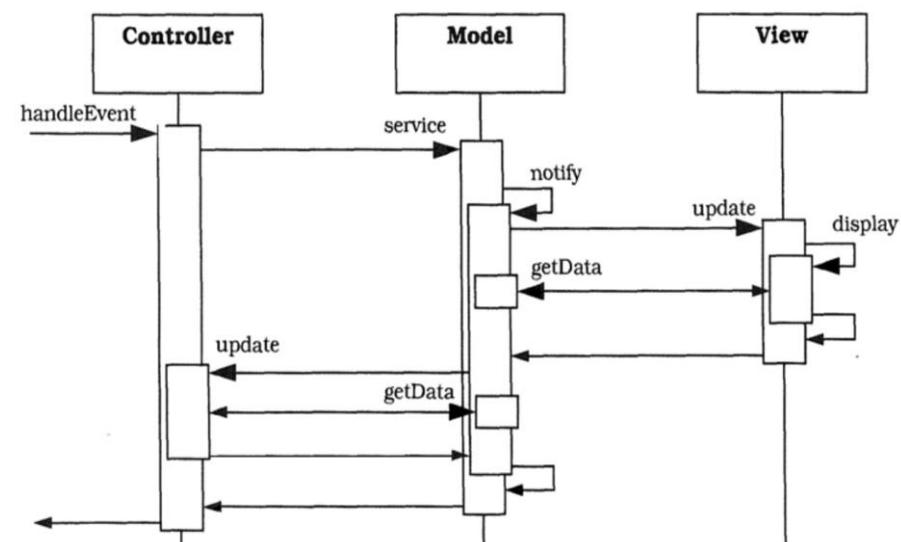
Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

1.1 **Patrón MVC y Model 2**

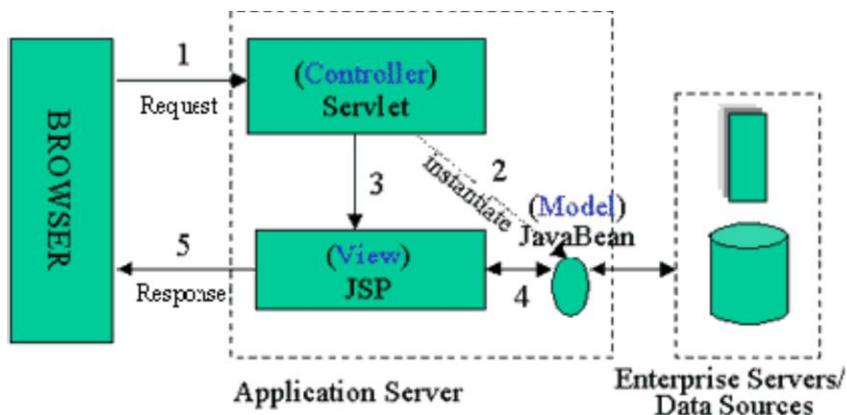
Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC.



El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema



La mayoría, por no decir todos, de los frameworks para Web implementan este patrón. Una aplicación de este patrón en entornos Java para programación Web es lo que se conoce con el nombre de arquitectura modelo 2.



Esta arquitectura consiste, a grandes rasgos, en la utilización de servlets para procesar las peticiones (controladores) y páginas JSP para mostrar la interfaz de usuario (vistas), implementando la parte del modelo mediante JavaBeans o POJOs.

1.2 Tipos de framework Web.

Existen varios tipos de frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Cocoon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC o con el modelo 2 de Servlets y JSP, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación.

1.3 Características.

A continuación enunciamos una serie de características que podemos encontrar en prácticamente todos los frameworks existentes:

- *Abstracción de URLs y sesiones.*

- No es necesario manipular directamente las URLs ni las sesiones, el framework ya se encarga de hacerlo.
- *Acceso a datos:* Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc..
- *Controladores:* La mayoría de frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
- *Autenticación y control:* Incluyen mecanismos para la identificación de usuarios *de acceso*. mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.
- *Internacionalización.* Separación entre diseño y contenido.

1 JAVA SERVER FACES

JavaServer Faces es un framework de componentes de interfaces de usuario para el desarrollo de aplicaciones Web basadas en tecnología Java™.

La tecnología JavaServer Faces surge como una solución a la separación entre la presentación y el comportamiento en una aplicación Web, de forma tal que las actividades de los autores de las páginas puedan separarse de las actividades de los desarrolladores de la lógica. Además, esto posibilita el desarrollo de entornos de desarrollos (IDE) que permitan trabajar en el diseño con funcionalidades del tipo drag-and-drop. Esta tecnología define un modelo de componentes UI que permiten definir la estructura del componente, diferentes formas de presentación, validaciones, la navegación entre las páginas y otros elementos.

1.1 Características

Tiene dos implicaciones importantes:

- a) Se trata de una tecnología que ejecuta del lado del servidor y no del lado del cliente.
- b) La interfaz de usuario es tratada como un conjunto de componentes UI.

Este tipo de tecnología es particularmente importante en los proyectos de tamaño mediano y grande, donde la separación de roles entre los integrantes del equipo resulta imprescindible para lograr un proyecto exitoso y mayor productividad. En este sentido, la separación entre la presentación y la lógica (o comportamiento) de la aplicación es una de las áreas más problemáticas.

Además, para una misma aplicación se suele trabajar con múltiples presentaciones (lo que depende del dispositivo donde se realizará la presentación, el navegador Web que se esté utilizando, cuestiones de localización de la aplicación, accesibilidad, etc.)

También, resulta importante trabajar, a nivel de la presentación, con componentes que puedan ser reutilizables o personalizados de acuerdo a la aplicación o usuario.

1.2 Ventajas de JSF.

La especificación JSF fue diseñada según los siguientes principios:

- **Amigable.** Por ejemplo, que los proveedores proporcionen soluciones mediante las que los desarrolladores puedan construir aplicaciones Web con drag-and-drop de componentes.
- **Independiente del Protocolo** y del dispositivo del cliente. Esto implica una correcta separación entre el modelo de componentes UI y la forma en que estos componentes se presentan en el dispositivo del cliente y bajo qué protocolo.
- Que pueda utilizarse con la tecnología **JSP**.
- Que pueda utilizarse independientemente de la tecnología JSP.
- Escalable.
- Que pueda utilizarse en el contexto de los **navegadores** Web actuales.

MySQL es la base de datos open source más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad están haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle.

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

1.1 Características.

✂ Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.

✂ Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.

✂ Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.

✂ Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.

1.2 Sintaxis de MySQL.

```
mysql> SELECT VERSION(), CURRENT_DATE;
```

```
| VERSION() | CURRENT_DATE |
```

```
| 3.23.41 | 2002-10-01 |
```

```
row in set (0.03 sec)
```

```
mysql>
```

Este comando ilustra distintas cosas acerca de mysql:

- Un comando normalmente consiste de una sentencia SQL seguida por un punto y coma.
- Cuando emitimos un comando, mysql lo manda al servidor para que lo ejecute, nos muestra los resultados y regresa el prompt indicando que está listo para recibir más consultas.
- Mysql muestra los resultados de la consulta como una tabla (filas y columnas). La primera fila contiene etiquetas para las columnas. Las filas siguientes muestran los resultados de la consulta. Normalmente las etiquetas de las columnas son los nombres de los campos de las tablas que estamos usando en alguna consulta. Si lo que estamos recuperando es el valor de una expresión (como en el ejemplo anterior) las etiquetas en las columnas son la expresión en sí.
- Mysql muestra cuántas filas fueron regresadas y cuanto tiempo tardó en ejecutarse la consulta, lo cual puede darnos una idea de la eficiencia del servidor, aunque estos valores pueden ser un tanto imprecisos ya que no se muestra la hora del CPU, y porque pueden verse afectados por otros factores, tales como la carga del servidor y la velocidad de comunicación en una red.

Las palabras clave pueden ser escritas usando mayúsculas y minúsculas. Las siguientes consultas son equivalentes:

```
mysql> SELECT VERSION(), CURRENT_DATE;
```

```
mysql> select version(), current_date;
```

```
mysql> SeLeCt vErSiOn(), current_DATE;
```

Aquí está otra consulta que demuestra como se pueden escribir algunas expresiones matemáticas y trigonométricas:

```
mysql> SELECT SIN(PI()/4), (4+1)*5;
```

```
/ SIN(PI()/4) | (4+1)*5 |
```

```
/ 0.707107 | 25 |
```

Además es posible escribir más de una sentencia por línea, siempre y cuando estén separadas por punto y coma:

```
mysql> SELECT VERSION(); SELECT NOW();
```

```
/ VERSION() |
```

```
/ 3.23.41 |
```

```
1 row in set (0.01 sec)
```

```
/ NOW() |
```

```
/ 2002-10-28 14:26:04 |
```

```
1 row in set (0.01 sec)
```

1.3 Ventajas.

- ✘ Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- ✘ Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- ✘ Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- ✘ Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

✂ El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).

1.4 Desventajas

- Depende de la aplicación
- No tiene tantas capacidades como otros gestores profesionales
- No es tan robusto como un ORACLE.
- Carece de soporte para transacciones, rollback's y subconsultas.
- El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
- No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

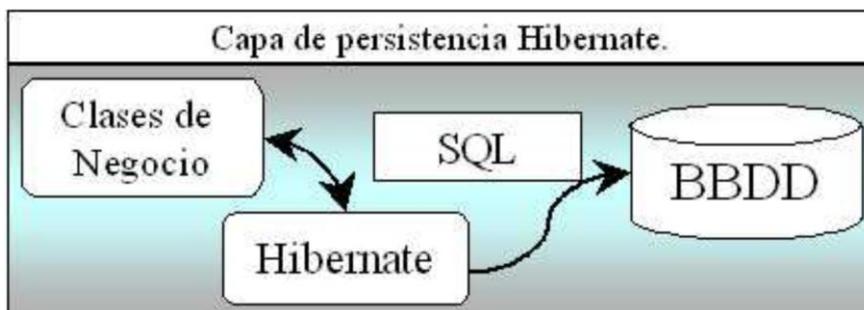
1 HIBERNATE

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias sql. Te permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada podremos generar BBDD en cualquiera de los entornos soportados: Oracle, DB2, MySql, etc. Y lo más importante de todo, es **open source**, lo que supone, entre otras cosas, que no tenemos que pagar nada por adquirirlo.

Uno de los posibles procesos de desarrollo consiste en, una vez tengamos el diseño de datos realizado, mapear este a ficheros XML siguiendo la DTD de mapeo de Hibernate.

Desde estos podremos generar el código de nuestros objetos persistentes en clases Java y también crear BBDD independientemente del entorno escogido.

Es decir, es el puente entre nuestra aplicación y la BBDD, sus funciones van desde la ejecución de sentencias SQL a través de JDBC hasta la creación, modificación y eliminación de objetos persistentes.



Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos. Una posible configuración básica de hibernate es la siguiente:



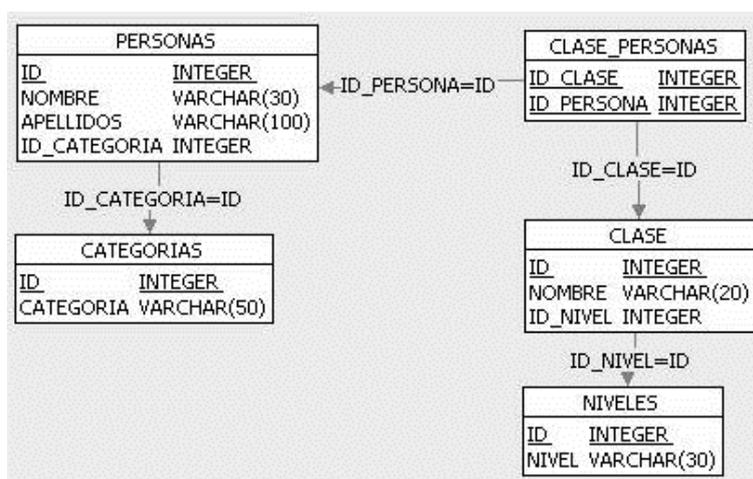
1.1 **Hibernate : Un casamentero**

Es muy común cuando se desarrollan proyectos con un concepto objetual llegar a la problemática de implementar una persistencia de objetos en un soporte relacional.

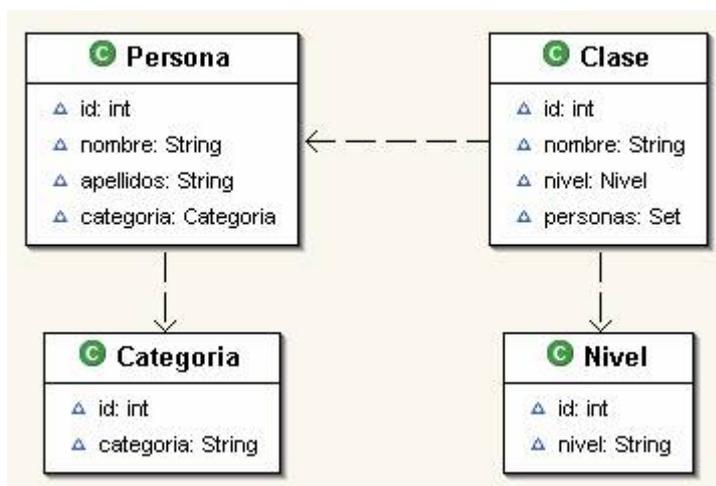
El problema es que hay un divorcio total entre la estructura relacional y la objetual. Esto hace que el desarrollador opte con frecuencia por soluciones de compromiso que le hacen perder de vista el mundo objetual y la aplicación acaba siendo un batiburrillo infumable.

Ejemplo: Intentamos diseñar una base de datos para un escuela. Por una parte, tenemos la lista de personas que pertenecen a la escuela. Cada persona tiene una categoría: profesor, alumno, personal, etc. En la escuela hay niveles: primero de EGB, segundo, etc. Además hay clases, cada clase pertenece a un nivel y tiene un nombre. Así, es posible que haya tres clases de primero de EGB: A, B y C. Lo único que nos queda es asignar personas a clases.

Así, podemos pensar en una definición de tablas similar a esta:



Pero, desgraciadamente el diseño objetual en el que nos gustaría trabajar no es un calco del diseño relacional, como mostramos en el siguiente diagrama UML:



Observamos las siguientes diferencias:

- Los nombres de los objetos están en singular mientras que los de las tablas están en plural. Así, entendemos que una instancia de un objeto se corresponde con un registro de la tabla. Luego una tabla es un conjunto de personas, y la clase Persona representa una sola persona.
- La clase Persona no tiene un entero que apunte a la tabla Categorías. Tiene un objeto de tipo Categoría. Así, vemos que los tipos de las propiedades de un objeto no siempre se corresponden con los tipos de las columnas de las tablas.
- En nuestro diseño del modelo relacional tenemos la tabla CLASE_PERSONAS que implementa la relación n:n de personas de una clase. Desde el punto de vista objetual, el planteamiento es distinto. Cada clase tiene un conjunto, Set, de personas. No necesitamos crear un objeto puente que relacione objetos con personas.

Es aquí donde entra Hibernate para ayudarnos. Se encarga de casar los dos modelos de manera que trabajemos desde Java como corresponde. Usando el modelo objetual.

Con Hibernate para añadir un registro a la tabla Categorías tendríamos que escribir algo como esto:

[...]

```
Configuration conf = new Configuration();
```

```
conf.addClass(Categoria.class);

SessionFactory sessionFactory = conf.buildSessionFactory();

Session session = sessionFactory.openSession();

Categoria categ = new Categoria();

categ.setCategoria("Profesor");

session.save(categ);

session.flush();

session.close();
```

[...]

La gran diferencia entre el código tradicional y el que escribimos con Hibernate es que: Ha desaparecido el SQL.

1.2 Qué ha de saber Hibernate para casar los dos modelos.

Quien esta detrás del modelo relacional. Gestor de base de datos. Forma de conexión.

Como se emparejan propiedades y campos de las tablas.

- Cual es la propiedad que se corresponde con la clave primaria de la tabla correspondiente.
- Que método deberá utilizar para generar un nuevo valor de la clave primaria
- Cómo empareja una propiedad con un campo de una tabla de la base de datos
- Como gestiona las relaciones entre tablas, uno-uno, uno-muchos, etc.

Para responder a estas preguntas se utilizan los siguientes archivos

- Archivo de propiedades de Hibernate (Hibernate .properties). Gestiona las conexiones con el gestor de la base de datos. Ejemplo de conexión a una BBDD de tipo HSQLDB:

⌘ *HypersonicSQL*

⌘ *hibernate.dialect net.sf.hibernate.dialect.HSQLDialect*

⌘ *hibernate.connection.driver_class org.hsqldb.jdbcDriver*

⌘ *hibernate.connection.username sa*

⌘ *hibernate.connection.password*

⌘ *hibernate.connection.url jdbc:hsqldb:hsq://localhost*

- Archivos que definen el emparejamiento de propiedades con tablas y columnas (*.hbm.xml). Ejemplo de archivo de emparejamiento simple entre la clase Categoría y la tabla Categorías

```
<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD//EN"

"http://hibernate.sourceforge.net/hibernate-mapping-
2.0.dtd">

<hibernate-mapping>

<class name="escuela.beans.Categoria" table="categorias">

<id name="id" type="integer" column="ID" unsaved-
value="-1">

<generator class="identity"/>

</id>
```

```

        <property name="categoria" column="CATEGORIA"
        type="string"
        unique="true"
        not-null="true"/>
    </class>
</hibernate-mapping>

```

- ✘ Con la etiqueta “class” emparejamos una clase con una tabla.
- ✘ Con la etiqueta “id” emparejamos una de las propiedades con la clave primaria de la tabla. No haría falta ni especificar el “type=integer” porque Hibernate analiza en tiempo de ejecución las clases Java y obtiene la información que necesita. El atributo “unsaved-value” permite especificar qué valor tiene el campo clave cuando aun no ha sido guardado en la BBDD. Este atributo es práctico porque la Session de Hibernate contiene un método “saveOrUpdate(Object)” que distingue si un objeto debe ser guardado por primera vez o bien modificado si ya está en la tabla según este valor.
- ✘ El subelemento “generator” permite definir en qué modo se crean las claves primarias, en nuestro ejemplo el gestor de BBDD HSQLDB soporta claves primarias de tipo IDENTITY.
- ✘ El elemento “property” emparejan aquellas propiedades que no forman parte de la clave primaria con las correspondientes columna(s) de la tabla(s).

1.3 Clases Persistentes – JavaBeans

Para que una clase pueda beneficiarse de las ventajas de Hibernate debe ser implementada como un JavaBean, vamos con los métodos get/set de toda la vida. Al menos las propiedades objetos de persistencia.

```

package escuela.beans;

public class Categoria {

    int id;

    public String getCategoria() {

        return categoria;

    }

    public int getId() {

        return id;

    }

    public void setCategoria(String nuevaCategoria) {

        categoria = nuevaCategoria;

    }

    public void setId(int nuevoId) {

        id = nuevoId;

    }
}

```

1.4 Altas , bajas , modificaciones (ABM) y consultas

En cualquier aplicación que use Hibernate aparecen cuatro objetos básicos :

- **Configuration.** es el objeto que contiene la información necesaria para conectarse con la BBDD. Se encarga de leerse el archivo Hibernate.properties y verificar los aparejamiento *.hbm.xml

```
Configuration conf = new Configuration();
```

```
conf.addClass(escuela.beans.Categoria.class);
```

- **SessionFactory.** Es una fábrica de Sessions.

```
SessionFactory sessionFactory = conf.buildSessionFactory();
```

- **Session.** El principal interface entre la aplicación Java e Hibernate. Es el que mantiene conversaciones entre la aplicación y la BBDD.

```
Session session = sessionFactory.openSession();
```

- **Transaction.** Se encarga de la transaccionalidad. Permite definir unidades de trabajo.

```
Transaction tx = session.beginTransaction(); [...]session.save(unaCategoria);  
tx.commit();  
[...]  
tx.rollback();
```

1 FASES DE SEGUIMIENTO DEL PROYECTO

Para el diseño, construcción e implementación del proyecto se usará el método ICONIX que es usado para empresas medianas y grandes, sigue los siguientes pasos:

1.1 PLANIFICACIÓN Y ESPECIFICACIÓN DE REQUERIMIENTOS

Esa fase preliminar en el desarrollo consiste en el análisis de los requisitos que presenta el usuario, es decir, las necesidades que debe satisfacer el producto que se desea construir.

Las actividades que se llevarán a cabo son las siguientes:

1.1.1 Definición del Plan Borrador

La fase de planificación y especificación de requisitos es una etapa preliminar al proceso de desarrollo. Consta de varias actividades que se deben completar para conseguir abordar la siguiente fase de forma óptima.

El plan borrador incluye un estudio de viabilidad para comprobar si el sistema es factible. Este estudio debe incluir costes y plazos estimados en términos de orden de magnitud, ya que un informe detallado implicaría un esfuerzo mayor que puede no ser necesario si el proyecto no es abordable.

Otra de las características que debe contemplar este estudio es una definición de los roles del equipo de desarrollo que va a participar en el proyecto.

También se define un plan acerca de cómo se va a desarrollar el sistema, incluyendo estimación de plazos, revisiones por parte del usuario, validación de requisitos, etc.

Las fases que contemplan en la definición de este plan son las siguientes:

Definición del programa.- Consiste en elaborar una planificación estimativa de las fases de las que va a constar el proyecto, incluyendo hitos relevantes, que servirá, para controlar si el proyecto transcurre dentro de los cauces estimativos, y establecer acciones correctoras en caso contrario.

Definición de recursos.- Es un aspecto importante en toda la gestión del proyecto. En esta fase se deben determinar los recursos que van a participar en el proyecto, y el rol que deben satisfacer en cada una de sus etapas. Algunos de los roles posibles son los siguientes:

Responsable de proyecto: es el responsable de la coordinación del mismo.

Responsable de calidad: es el responsable de que el proyecto cumple con las normas de calidad establecidas.

Responsable de soporte: es el encargado de cumplir con la gestión de configuración, control de versiones, mantenimiento de la línea base del proyecto, etc.

Analista: es el encargado de realizar el análisis y diseño.

Programador: es el encargado de llevar a cabo la implementación.

⌘ Estos roles son estimativos, y no tienen por qué participar todos, incluso se pueden tener otros no descritos.

Elaboración de presupuesto.- Es el documento contractual que establece una relación entre el cliente que ha encargado el sistema a construir y la empresa que se encarga de su implantación.

Determinación de riesgos.- Un riesgo es todo evento que puede hacer fracasar o retrasar el desarrollo del proyecto. Un plan de gestión de riesgos debe establecer la relación entre la posibilidad de ocurrencia de dicho riesgo y la pérdida que ocasiona.

⌘ **Informe de Investigación Preliminar.**

Sirve para analizar los motivos que han llevado a la empresa a realizar el proyecto, las alternativas de desarrollo, las necesidades de la empresa, tanto a nivel de recursos como de material (Hw, Sw, comunicaciones).

Plan de trabajo: es necesario, una vez que se haya dado la conformidad al proyecto, definir un plan de trabajo que riga la metodología a utilizar, el ciclo de desarrollo que tendrá la aplicación, definir puntos de control, etc.

Metodología: es básico definir la metodología que se va a usar para implementar el sistema. Habrá que hacer un estudio para determinar la metodología más adecuada a la semántica de la aplicación, debido a que un tipo de aplicación es más conveniente abordarla con una metodología o con otra. La metodología que se empleará en el desarrollo de éste sistema es orientado a objetos.

Definición de recursos: aquí se define o se especifica el tipo de plataforma, cuál va a ser el uso que se haga de la aplicación, el nivel de eficiencia que se precisa, si va a utilizar una base de datos, etc. Así mismo habrá que especificar la infraestructura de comunicaciones que debe contar el sistema según las necesidades de los usuarios.

⌘ **Especificación de Requisitos.**

Los requisitos son las funcionalidades que debe cumplir el sistema. Es una de las fases más críticas del proceso de desarrollo del software. Según se deriva de diversos estudios.

Por lo tanto se puede decir que una buena especificación de requisitos va a producir un software robusto y sin errores. No detectar incorrecciones en los requisitos incrementa los costes, y más cuanto más avanzada sea la fase en la

que se encuentra el sistema. La forma en la que se especifica los requisitos es en lenguaje natural, ya que es un documento abierto tanto al analista como a los usuarios. Se suele utilizar frases del estimo “el sistema hará”

☒ **Fases en la especificación de requisitos:**

El proceso para especificar requisitos consta de las siguientes fases:

☒ **Educción de requisitos:** Consiste en la captura de los requisitos. Para ello es necesario un conocimiento del dominio. Se usará las siguientes técnicas: **Brainstormin:** se reúne a un grupo de usuarios para que suelten ideas de lo que se pretende que haga el sistema. Los analistas decidirán cuales de ellas son relevantes para incluir en el mismo. **Observación y análisis de tareas:** el analista observa el trabajo del usuario para intentar implementar un sistema parecido. **Prototipado:** se realiza un prototipo y se analiza por parte del usuario. **Entrevistas:** se realizan preguntas a los usuarios. Pueden ser abiertas o cerradas.

☒ **Análisis de requisitos:** consiste en detectar conflictos entre requisitos y resolverlos y negociar aquellos que sean más conflictivos, es decir, es preferible ponerse de acuerdo acerca de cómo hacer algo, o imponerlos al cliente. La modelización conceptual puede servir de ayuda si el sistema es complejo e impide una correcta descripción de los objetivos. Como veremos, pueden utilizarse casos de uso para describir los requisitos.

☒ **Documentación:** este debe contener:

Información referente al problema

Propiedades y comportamiento del sistema.

Restricciones

Descripciones acerca de cómo el sistema ayudará a realizar las tareas.

Un documento de especificación de requisitos debe ser no ambiguo, es decir, determinar con claridad cada una de las restricciones que debe satisfacer. Tampoco será válida una especificación de requisitos que será completa y

correcta, o lo que es lo mismo, debe describir todo el sistema respondiendo a las necesidades del usuario.

Deberá utilizar un lenguaje comprensible por ambas partes (analista y cliente), para que pueda ser verificable e interpretable. Por este motivo, el documento deberá ser modificable si alguna de las partes encuentra un error en la especificación. Lo más importante es que el documento sea independiente de aspectos de diseño, ya que en la primera fase nos limitamos al modelado conceptual.

· **Elementos del estándar IEE** :el estándar es un documento que especifica los requisitos, y se divide en las siguientes partes:

⌘ Introducción

✘ Propósito y alcance

✘ Definiciones

✘ Referencias

✘ Visión general

⌘ Descripción

✘ Perspectiva

✘ Funciones

✘ Características de los usuarios

✘ Restricciones

✘ Suposiciones

✘ Requisitos específicos.

✘ **Validación:** la validación consiste en la aprobación por el usuario del documento de requisitos construido por el analista. El objetivo es encontrar fallos en los requisitos que deben ser corregidos en esta misma fase. Además convendrá revisar la calidad del documento y la adhesión a estándares. La

fórmula más empleada es la de listas de comprobaciones o ckechlist, que es un conjunto de preguntas que deberán de satisfacerse para aprobar el documento.

✧ **Glosario de Términos.**

Es una especie de diccionario donde se va a detallar todos los conceptos que puedan dar lugar a dudas. Se iniciará con la definición de los términos más importantes surgidos del análisis de requerimientos, así:

□ **M**

od	Nombre	Descripción
el	XXXXXXXX	XXXXXXXXXXXXXXXXXXXX

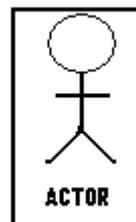
os

de casos de uso.

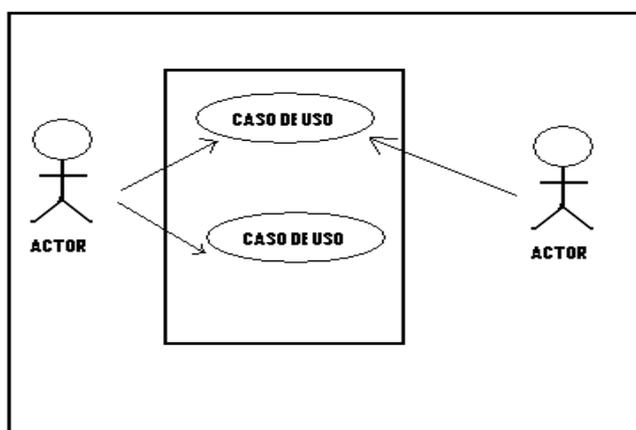
Un caso de uso es un documento que nos ayuda a comprender mejor los requisitos que debe cumplir el sistema, las necesidades determinadas por el usuario. Un caso de uso describe la secuencia de eventos, explicando la interacción del proceso con las entidades externas que intervienen en él. La notación par el caso de uso es le nombre de la funcionalidad rodeado por un óvalo, así:



En un caso de uso intervienen una serie d actores, que son las entidades que participan el proceso que describe el caso de uso. Un actor puede ser desde le clientes que acude a pagar a un terminar Punto de Venta, hasta el encargado de introducir dinero en un cajero automático. Una autor se representa así:



Para indicar que un actor interviene en un caso de uso se unen ambos con una recta. El sistema se representa con un rectángulo conteniendo los casos de uso:



Casos de uso de alto nivel

Hay que especificar lo siguiente:

- Nombre
- Tipo
- Actores
- Descripción breve de lo que hace el caso de uso

Nombre	Xxxxxxxxxx
Actores	Xxxxxxxxxx
Descripción	xxxxxxxxxxxxxxxxxxxx

Pasos para construir un modelo:

1. Listar las funciones del sistema, obtenidas de la especificación de los requisitos, e identificar actores y casos de uso.

2. Escribir los casos en formatos de alto nivel y clasificarlos por su opcionalidad.
3. Dibujar el diagrama de casos
4. Determinar los casos más importantes y escribirlos en formato esencial expandido
5. Para los casos descritos más concretos o para los que especifique el cliente, escribir los casos en formato real expandido.

Casos de uso de bajo nivel o expandidos.

Aquí se debe especificar las siguientes características:

- Nombre.
- Actores, hay que especificar el iniciador, es el actor que comienza el caso de uso.
- Propósito de proceso que se describe mediante el caso de uso.
- Visión general, descripción del proceso.
- Referencia a los requisitos que se contemplan en el caso y otros casos de uso.
- Curso normal de eventos.- indica la interacción entre el sistema y los actores, mientras que el curso.
- Curso alternativo de eventos.- especifica acciones que forman parte del curso normal, como errores, excepciones u otras opciones.

Nombre	XXXXXXXXXX
Actores	XXXXXXXXXX
Propósito	XXXXXXXXXXXXXXXXXXXXXX
Visión general	
Referencia	
Curso normal de eventos	
Curso alternativo de eventos	

□ **Modelo Conceptual.**

El objetivo del modelo conceptual es describir el dominio en un grado de abstracción alto, libre de decisiones de diseño, identificando conceptos, características y relaciones. Se denomina también *diagrama de estructura estática* y debe satisfacerse los requisitos expresados en los casos de uso. Las principales características del modelo conceptual son:

- Describe los conceptos del dominio mediante conceptos, asociaciones y atributos
- Estático
- Cíclico (un modelo conceptual por cada ciclo)
- Evolutivo.

Pasos en el modelo conceptual

Para construir un modelo conceptual se seguirá los siguientes pasos:

- Partir de los requisitos descritos en los casos de uso y del conocimiento del dominio
- Especificar nombres, conceptos y atributos
- Realizar una lista de conceptos comunes. Estos conceptos incluyen: objetos físicos, especificaciones o descripciones de objetos, lugares, transacciones, línea de una transacción, roles, contenedores, sistemas o dispositivos externos, conceptos abstractos, organizaciones, miembros de una organización, eventos, reglas o políticas, catálogos, documentos financieros o legales, manuales, libros, documentos.
- Realizar una lista de asociaciones típicas relevantes entre distintos conceptos, para las que interese almacenar la información de dicha relación. Las asociaciones que se pueden tener son:
 - A es parte de B
 - A es parte lógica de B
 - A está físicamente dentro de B

- A está lógicamente dentro de B
- A es una descripción de B
- A es un elemento de una transacción de B
- A es registrado, capturado o archivado en B
- A es miembro de B
- A es subunidad organizativa de B.
- A usa o gestiona B
- A comunica con B
- A está relacionado con una transacción B
- A es una transacción relacionado con otra transacción B
- A está junto a B
- A posee B

Elementos del diagrama de estructura estática:

Son los siguientes:

- **Conceptos.-** representa un término definido en los requisitos descritos en los casos de uso. Se representa con un cuadrado.
- **Atributo.-** característica de los conceptos que permite describirlos. Sus valores suelen pertenecer a tipos simples, aunque pueden pertenecer a tipos estructurados, en cuyo caso puede considerarse como un concepto aparte. Los atributos se colocan dentro del cuadrado que describe el concepto.
- **Asociaciones.-** representa relaciones entre dos conceptos. La multiplicidad indica el número de conceptos que se relaciona con cada uno de ellos, que puede ser un número exacto, un número indefinido o un rango de valores, dando la posibilidad de especificar opcionalidad.

Diagrama de secuencia.

El objetivo de los diagramas de secuencia es dar una visión dinámica, que complemente la visión estática del modelo conceptual, describiendo los eventos que hacen que los actores se comuniquen con el sistema, las

operaciones de éste y las repuestas a los eventos. Se construye a partir de los casos de uso en de bajo nivel o expandidos, representando sólo aquellos actores que generan eventos que llegan al sistema. Lo normal es realizar un diagrama de secuencia por el curso normal de eventos de cada caso de uso y para cada curso alterno.

Notación.- los eventos se indican con una flecha horizontal que parte de la línea correspondiente al actor que lo inicia y finaliza en el sistema, indicando para cada flecha el nombre del evento al que corresponde.

Pasos para elaborar un diagrama de secuencia:

- Trazar una línea que represente el sistema, como una caja negra
- Identificar los actores que operan en le sistema y trazar una línea para cada actor
- Identificar los eventos que son generados por el actor u mostrarlo en el diagrama
- A la izquierda del diagrama se puede mostrar el texto del caso de uso

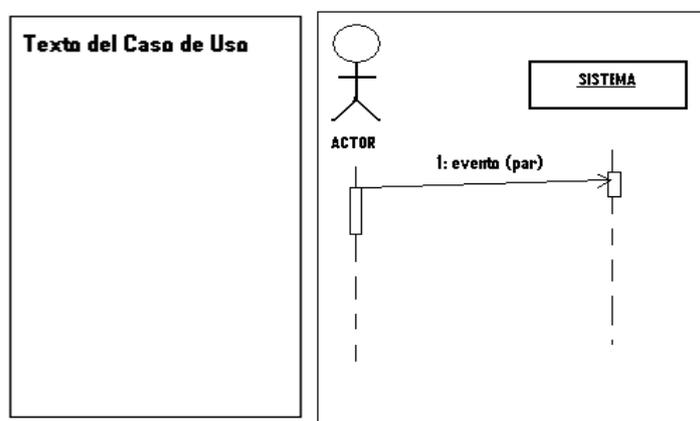


Diagrama de Robustez

Con este tipo de diagramas se obtendrá el diseño preliminar, además nos servirá para comprobar que los casos de uso estén correctos y completos.

Beneficios del análisis de robustez.

- Fuerza a escribir los casos de uso usando un estilo consistente.
- Fuerza a escribir los casos de uso en la voz correcta (presente)
- Este provee de un chequeo de sanidad y completitud para los casos de uso
- Este permite aplicar sintaxis a sus reglas de sus casos de uso. Pro ejemplo que un actor sólo pude hablar con objetos de frontera
- Los diagramas de robustez son rapidísimos para dibujar y facilísimos para leer que un diagrama de secuencia.
- Este ayuda a la capa de objetos dentro de un Framework (facilita la escritura de código) de interfaz gráfica, lógica y repositorio (base de datos) para sistemas cliente servidor.
- Este permite una reutilización de pasos a través de todos los casos de uso antes de iniciar el diseño
- Este provee un seguimiento entre que es lo que hace el sistema (casos de uso) y como el sistema trabaja(diagramas de secuencia) usando una vista conceptual de objetos
- Este mejora la semántica entre el análisis (casos de uso) y el diseño (diagramas de secuencia).

1.1.2 CONSTRUCCIÓN.

En esta fase se llevará a la práctica todo lo diseñado en fases anteriores. Si comparamos al software con la construcción de un edificio, la fase de análisis sería algo similar a realizar un estudio de viabilidad económica, el estudio topográfico de los terrenos, presupuestación, etc. La fase de diseño sería equivalente a realizar los planos arquitectónicos, mientras que la implementación corresponde a la construcción propiamente dicha del edificio. En la construcción, se codifica el diseño realizado con la herramienta de codificación escogida.

Definición de atributos y métodos: Lo primero que hay que tener en cuenta es crear una clase por cada una de las clases del diagrama de clases de diseño. Cada una de estas clases tendrán los tributos y métodos detallados en ella.

El siguiente paso es programar los métodos. Para ello tenemos en cuenta los diagramas de robustez que tiene como entrada dicho método.

Diagrama de clases de diseño

Este diagrama es una herramienta que servirá para definir el comportamiento del sistema, es decir, cuales son las clases, los atributos y los métodos que se van a implementar en el diseño de bajo nivel. Se construye como una ampliación del modelo conceptual, teniendo en cuenta los diagramas de secuencia, las operaciones y los contratos definidos en el diseño de alto nivel.

Pasos para construcción del diagrama:

- Identificar las clases en los diagramas de interacción, robustez, etc.
- Dibujarlas en el diagrama
- Colocar los atributos del modelo conceptual
- Colocar los métodos de los diagramas de robustez.
- Incorporar los tipos de los atributos y los métodos
- Agregar las asociaciones y dependencias entre atributos.- se puede indicar varios tipos de acceso o ámbito para los atributos o métodos: Privado(sólo puede ser accedido por métodos de su clase), Público (puede ser accedido por métodos de su clase y de las demás), Protegido (puede ser accedido por métodos de su clase y de los subtipos).

1.1.3 IMPLEMENTACIÓN Y PRUEBAS

Los planes de prueba son los que definen las pruebas que se harán al sistema par validar su funcionamiento. Se puede definir varias técnicas dentro de las cuales se encuentran los casos de uso.

Tipos de prueba

⌘ **Pruebas unitarias:** son las pruebas que verifican el funcionamiento de un procedimiento, sentencia o conjunto de instrucciones. Por ejemplo, comprobar que un bucle se ejecuta 2 veces, o que una función ejecuta las instrucciones deseadas.

⌘ **Pruebas de módulo:** sirven para comprobar el funcionamiento de un módulo. Por ejemplo, comprobar que el módulo de contabilidad funciona.

⌘ **Pruebas de integración:** sirven para determinar la correcta integración de los distintos procedimientos que componen el sistema. Prueban el sistema en su totalidad.

Como definir las pruebas

Las pruebas suelen seguir los estándares de la empresa, que a su vez pueden utilizar las distintas normalizaciones de los organismos internacionales como ISO. Así se pueden tener planes de calidad que hacen que el sistema cumpla con los requisitos de las normas. Además se tendrá en cuenta los siguientes aspectos:

- Las programadoras no deberán probar sus propios programas
- Se debe probar con valores límite y aleatorios; probar introducir valores en los rangos inferior y superior de variables array y algún valor representativo.
- Se debe probar el cero en las variables numéricas.
- Intentar seguir las secuencias de ejecución posibles.

Los fallos se resolverán directamente siguiendo las normas establecidas, además se seguirá los planes de control de cambio para mantener la consistencia de la aplicación y evitar duplicidades.

1.2 Tipo de Investigación

El presente proyecto será de tipo Practico-Científico ya sigue los pasos que requiere la investigación Exploratoria y de Campo y al final del proceso de investigación se procederá a su implantación y ejecución.

En la investigación no será de tipo experimental, porque no requiere de una experimentación en un laboratorio. Pero simplifica el análisis y descripción de la obtención de la información de campo.

3 DISEÑO METODOLÓGICO.

3.1 Metodología para la ejecución de la Investigación

La herramienta que se plantea permitirá realizar el control del tráfico de redes que tienen sistema operativo Linux, cuya versión de kernel se 2.4, 2.6 o superiores, permitirá realizar una configuración de la red, para de este modo realizar la visualización y consumo de banda en la red.

La investigación exploratoria tiene un diseño especial, que aplicado a esta disciplina científica nos servirá de mucho, puesto que su objetivo es realizar una investigación preliminar mediante la cual se realiza la observación inmediata del área y de los elementos constitutivos del objeto que va ser investigado, permitirá formular los problemas, aclarar conceptos, reunir información y familiarizarnos con el fenómeno que deseamos investigar.

Para esto utiliza el método científico porque nos permitirá establecer una relación teórica y práctica dentro de este proceso. Este método es el proceso mas objetivo, ya que nos ayudara a entender el proceso de los métodos inductivo y deductivo.

El **método inductivo**, crea leyes a partir de la observación de los hechos, mediante la generalización del comportamiento observado, nos servirá para la recolección de la información relacionada a las actividades, problemas, causas y posibles alternativas de solución referentes a los elementos de observación, **método deductivo** aspira a demostrar, mediante la lógica pura, la conclusión en su totalidad a partir de unas premisas, de manera que se garantiza la veracidad de las conclusiones.

Para realizar la observación de los elementos más importantes del objeto que se investiga, la investigación exploratoria necesita ayuda de una actividad de campo, en donde el mismo objeto de estudio nos servirá como fuente de información utilizando técnicas como:

- La observación directa que consiste en la inspección y estudio por medio de la observación de las características más sobresalientes, logrando la captación de la realidad natural, que implica, procesos de cada departamento, área e identificar y acudir a personas que tengan un conocimiento completo del funcionamiento, con la finalidad de recoger información valida para el análisis descriptivo de los hechos que nos confieren.
- La entrevista dirigida a empleados, usuarios y administrativos para obtener una información oral valida, clara y concisa como aporte en busca del correcto y normal funcionamiento.

Con la finalidad de que el proceso investigativo se desarrolle de la mejor manera, es que hemos creído conveniente usar una metodología que se amolde a las necesidades de este tipo de investigaciones, que con nociones básicas de metodologías “clásicas”, como las explicadas anteriormente y conocimientos generales del método inductivo, deductivo y las formas de producción de software es que hemos optado por la **Metodología Orientada a Objetos**.

MÉTODO ORIENTADO A OBJETOS: Es una de las metodologías de análisis y diseño orientados a objetos, más maduros y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software.

Se han concentrado muchos esfuerzos hacia el desarrollo de nuevos modelos para enriquecer este método de producción de software orientado a objetos con la expresividad necesaria para especificar características navegacionales y de presentación de información orientadas a las aplicaciones Web. Esta extensión del método OO con

capacidades navegacionales y de presentación es lo que se denomina OOWS (Object Oriented Web Solution).

Consta de las siguientes etapas: Análisis, Diseño, Construcción e Implementación.

- En la etapa de **Análisis** la recopilación de información se realizará mediante el planteamiento a los usuarios de cuestionarios y entrevistas, obtenida esta información se procederá a su análisis y evaluación.

- Posteriormente en la etapa de **Diseño** se especificarán los requisitos de funcionalidad, navegación y presentación de la aplicación Web. Siguiendo el proceso propuesto por el método OOWS en esta etapa debemos obtener una especificación completa del sistema software para la Web. Para ello esta etapa se divide en dos:

Especificación de los requisitos funcionales del sistema. Utilizando una aproximación basada en casos de uso se propone obtener los esquemas conceptuales (diagrama de clases, modelado funcional).

- ⌘ **Especificación de navegación y presentación de información:** Es decir se capturará la semántica navegacional de la aplicación Web (Modelo de navegación) y de presentación de la información.

- ⌘ **Modelado de la Navegación:** En este punto debemos contar con los requisitos de navegación de la aplicación. Para ello introduciremos el modelo de navegación sobre el diagrama de clases para cada tipo de usuario que pueda interactuar con el sistema. Esto nos proporcionará una estructura de accesibilidad (definiendo el conjunto posible de caminos de navegación) en función del tipo de usuario. Es decir realizamos el modelo de navegación de la aplicación Web en función de las necesidades de cada tipo de usuario. Para ello se realizará lo siguiente:

- ✘ Clasificación e identificación de usuarios. Aquí se estudiará los diferentes tipos potenciales de usuario que puedan interactuar con el sistema, que interrelaciones existen entre ellos y cual va a ser su modo de acceso al sistema.

- ✘ Construir el mapa navegacional. Para cada usuario detectado, se construirá su vista navegacional del sistema (basado en el diagrama de clases definido previamente). Seguidamente se realizará el modelado de la base de datos que se utilizará en el sistema.

✂ Modelado de la Presentación. Una vez definido el modelo de navegación se especificará las características de presentación del sistema. Para esto se introduce el Modelo de Presentación que complementa la información capturada en el Modelo de Navegación para la creación de interfaces Web, en este espacio debemos basarnos en el uso de patrones; los patrones de presentación de información que se pueden especificar son:

- Paginación de información. Cuando especificamos la paginación, el conjunto de instancias que deben ser presentados deberán ser “divididas” en “bloques” de manera que sólo uno de los bloques sea visible al mismo tiempo. Se proporcionará para avanzar y retroceder entre las distintas páginas obtenidas.
- Ordenación. Este patrón permite definir una ordenación de la población de una clase según el valor de uno o más atributos sobre los que se aplica.
- Presentación de instancias. Aquí se indicará como se presenta los elementos detalladamente.
- Ya concluida la etapa de diseño se pasará a la **Implementación** de la aplicación Web a partir del modelo conceptual. Para realizar la implementación de aplicaciones Web a partir del modelo conceptual realizado, se planteará una arquitectura de tres capas. Este tipo de arquitectura nos permite estructurar claramente las aplicaciones Web facilitando la adaptabilidad y extensibilidad de las mismas y ofrece una mayor escalabilidad. Las tres capas en las que se divide la arquitectura que se propone se describen a continuación:

✂ **Modelo:** Esta capa implementará la lógica de negocio y la funcionalidad de consulta.

✂ **Vista:** Incluye la interfaz gráfica de usuario (en este caso, páginas Web y componentes visuales) para interactuar con el usuario (visualizando información, proporcionando acceso a servicios y facilitando la navegación).

✂ **Controlador:** Esta capa implementará la persistencia y el acceso a datos ocultando a las capas superiores detalles de los repositorios de información. Básicamente esta etapa consistirá en la configuración del software necesario para empezar a la codificación de la aplicación. Luego se realizarán las respectivas pruebas

de validación del sistema con las carreras a distancia lo que permitirá comprobar el correcto funcionamiento de la aplicación, posteriormente se elaborarán los respectivos manuales así como la documentación y se adiestrarán a los usuarios.

3.2 Técnicas e Instrumentos.

Para la presente investigación se utilizará el método *inductivo-deductivo*, ya que es necesario primeramente llegar a comprender el fenómeno que lleva consigo el control de tráfico en redes Linux, como cuento con una base teórica, la técnica que se va a utilizar, para complementar el método escogido, es el fichaje, el cual permite referenciar los documentos en los que se basa la presente investigación, la aplicación de las presentes herramientas investigativas me permitirán poner en evidencia los conocimientos adquiridos, que junto con la base cognoscitiva y bibliográfica me permitirán llevar a conclusión el desarrollo e implementación de la herramienta que es tema de la presente tesis.

3.3 Elaboración o Redacción del Informe y de las alternativas de solución.

El informe será estructurado de tal manera que permita comprender, como se lleva el control de tráfico de red en el Área de Energía Industrias y Recursos Naturales no Renovables, cuales son los controles que actualmente se ejercen, que es el control de tráfico de redes Linux, como se realiza el control de tráfico, las configuraciones básicas de tc, los problemas que se presentan al no tener un controlador de tráfico de red.

- **ORGANIZACIÓN Y GESTIÓN DE LA INVESTIGACIÓN.**

Para lograr la culminación del presente proyecto se tomará en cuenta los recursos humanos, económicos, materiales, técnicos y tecnológicos. Además se realizará un cronograma de actividades que permitirá cumplir desde el inicio hasta el final con los objetivos planteados, los recursos serán detallados a continuación:

- **Recursos**

- **Humanos.**

El recurso humano necesario para la realización del presente trabajo se detalla a continuación:

- Director de Tesis.
- Asesor.
- Jorge Guerrón Eras.

■ **Económicos**

Tomando en cuenta los recursos humanos descritos anteriormente, como los recursos tecnológicos de hardware y software, presento una idea general de los costos que intervendrán en el desarrollo del presente proyecto.

<i>Factor</i>	<i>Costo</i>
Guía de asesor	\$ 250.00
Hardware y recursos	\$ 1522.00
Software	\$ 0.00
Recursos Materiales	\$ 320.00
Materiales de oficina	\$ 313.30
Transporte y Servicios Básicos	\$ 600.00
Imprevistos %5	\$ 150.27
Total	\$ 3,155.57

Las actividades que se cumplirán durante el proceso de desarrollo se lo explican detalladamente en el Cronograma de Actividades. (*Anexo A*)

Para tener una idea más detallada de los costos que implica la realización del proyecto, revisar el Plan Operativo que se anexa al final.(*Anexo B*)

■ **Materiales**

CANTIDAD	DESCRIPCION
RECURSOS	
MATERIALES	
<i>Bibliografía</i>	
1	Manual de JSF
1	Manual de MySQL
1	Manual de Apache-Tomcat
1	Manual de UML
2	Alquiler de Internet
1	Manual de Hibernate.
<i>Material de Oficina</i>	
3	Paquetes de 500 hojas de papel 75 g/m2 tamaño A4.
4	Cartuchos de tinta para impresora
5	Encuadernación
20	Cd
1	Memory flash
2	lápices
4	esferos
2	borrador

1000	Copias
<i>Otros</i>	Transporte
	Luz eléctrica, agua, teléfono

■ **Técnicos y Tecnológicos**

Los recursos Técnicos y Tecnológicos que se utilizarán están acorde a las necesidades y requerimientos de la Institución para así realizar el desarrollo del sistema. A continuación presentamos los siguientes:

CANTIDAD	DESCRIPCION
<i>HARDWARE</i>	
1	PORTÁTIL Centrino Duo, 1 Gb Ram, disco duro 100 GB
1	Impresora de inyección de tinta(Hp 845C)
1	Swicht(8 puertos)
6	Cable UTP cat 5
<i>SOFTWARE</i>	
1	Servidor Apache-Tomcat
1	Servicio MySQL
1	FrameWork JSF
1	NetBeans 5.5
1	MySQL Tools
1	Umbrello
1	FrameWork Hibernate

1	Open Office 2.2
1	Fedora 7 (kernel 2.6)

○ **Cronograma de Actividades**

○ **Bibliografía**

Libros:

1. Martín, James. Año. Análisis y Diseño Orientado a Objetos. 2da. Edición Prentice Hall Interamericana. México Pags. 51-59.
2. PETERSEN Richard, “Manual de Referencia Linux”, Segunda Edición, MacGraw-Hill, España, 2001
3. PIATTINI Mario, GARCÍA Félix, “Calidad en el desarrollo y mantenimiento del Software”, RA-MA Editorial, Mexico.

Internet:

1. <http://developer.osdl.org/dev/iproute2/signature.html>
2. <http://www.fedoraproject.org>
3. <http://www.hibernate.org>
4. <http://lartc.org/lartc.pdf>
5. <http://www.linuximq.net>
6. <http://www.mysql.com>
7. <http://wiki.nix.hu/cgi-bin/twiki/view/IMQ/HowToInstall>
8. <http://wiki.nix.hu/cgi-bin/twiki/view/IMQ/ImqFaq>

ANEXO B.

PLAN OPERATIVO

CANTIDAD	DESCRIPCION	DURACION	VALOR U (\$)	VALOR TOTAL (\$)
RRHH				
1	Director de tesis		0.00	0.00
2	Analista Programador	400 h	0.00	0.00
1	Asesor	50 h	5.00	250.00
Total				250.00
RECURSOS TECNICOS				
HARDWARE				
1	Portartil		1400.00	1400.00
1	Impresora de inyección de tinta		120.00	120.00
1	Switch			0.00
6	Cable UTP cat 5			2.00
Total				1522.00
SOFTWARE				

1	Servidor Apache Tomcat versión 5.0.9		0.00	0.00
1	MySQL 5.0		0.00	0.00
1	FrameWork JSF		0.00	0.00
1	Licencia NetBeans 5.5		0.00	0.00
1	Licencia Umbrello UML		0.00	0.00
1	Sistema Operativo Linux		0.00	0.00
Total				0.00
RECURSOS MATERIALES				
Bibliografía				
1	Manual de JSF		0.00	0.00
1	Manual de MySQL		0.00	0.00
1	Manual de Apache		0.00	0.00
1	Manual de UML		0.00	0.00
1	Manual de Hibernate		0.00	0.00
2	Alquiler de Internet	200 h	0.80	320.00
Total				320.00

<i>Material de Oficina</i>				
3	Paquetes de 500 hojas de papel 75 g/m2 tamaño A4		3.00	9.00
4	Cartuchos de tinta para impresora HP deskjet 845c		36.00	144.00
5	Encuadernación		20.00	100.00
20	Caja Cd		15.00	15.00
1	Memory flash		22.00	22.00
2	Lápices		1.50	3.00
4	Esferos		0.25	1.00
2	Borrador		0.15	0.30
1000	Copias		0.02	20.00
Total				313.30
Otros				
	Transporte		0.25	500.00
	Luz eléctrica, agua, teléfono			100.00
Total				600.00
IMPREVISTO				150.27
COSTO TOTAL DEL PROYECTO				3,155.57

ANEXO C.

Descripción y Alcance del Proyecto

La presente herramienta tiene como meta, convertirse en un apoyo para los administradores de redes y centros de cómputo, puesto que permitirá optimizar y agilizar las actividades diarias comprendidas en el control y monitoreo de tráfico de Redes, además será un apoyo para los estudiantes ya que podrá ser usado no solo como una guía para futuras investigaciones, sino como un instrumento de aprendizaje.

La herramienta que es mencionada en el presente trabajo será construida en interface Web, desarrollada e implementada con el framework JSF, que posee muchas ventajas ya que es libre y esta enfocado directamente con el patrón Modelo-Vista-Controlador, además me permite la integrar api's para el manejo y presentación de información. Se integrará una base de datos capaz de soportar la información que se manejará como es MySQL 5.0.

La presente herramienta estará integrada por los siguientes módulos que permitirán a los usuarios, manejar las especificaciones de mejor manera:

- Construir un modulo de control de seguridad de los servicios de la herramienta.
- Crear un modulo de usuarios, para la herramienta.
- Crear un modulo de monitoreo, para la presentación de los datos.
- Construir una Base de Datos que permita integrar nuevos módulos en el sistema, y que permita generar un historial de la misma.
- Realizar un análisis del desempeño de la herramienta.
- Implementar la herramienta en el Área de Energía, Industrias y Recursos Naturales no renovables, para su uso y provecho.

La primera parte de configuración de los servicios de control y monitoreo de la herramienta, se encargará de la administración los servicios, consistiendo estos en la creación, búsqueda e ingreso de las reglas para el manejo del ancho de banda de la red en la cual se desea llevar a cabo la implementación de la herramienta, de este modo los usuarios pueden definir sus propios conjuntos de reglas para manejar el flujo de la red, dentro de las principales aspiraciones de este proyecto, está la utilización de las reglas de control de trafico o TC, de iproute2 que viene incorporado en los kernel 2,4 y 2,6, de las versiones de linux mas actuales, al hacer uso de estas reglas se esta dando la posibilidad de hacer uso de todas las opciones que se nos presentan como son canales o tubos(pipes), filtros, y cadenas. Otro segmento interesante de la presente herramienta es la visualización del consumo de la red, en el cual se presentarán los datos correspondiente al consumo del ancho de banda de los diferentes equipos correspondientes a la red, de igual modo para poder llevar a cabo un control se consta con la implementación del módulo de usuarios el mismo que se encargará de restringir el acceso a ciertas partes de la herramienta,

Una vez que se ha creado cada módulo, se podrá integrar a un solo sistema que permitirá compartir información de acuerdo a las necesidades de los

usuarios, para de este modo poder realizar finalmente la evaluación de la herramienta.

Como último objetivo del presente proyecto es implementar la presente herramienta en el Área de La Energía las Industrias y los Recursos no Renovables, de la Universidad Nacional de Loja, para su uso, aprendizaje y explotación.

RESUMEN LICENCIA GPL

Es la más utilizada en el mundo Linux. Fue creada por la Free Software Foundation(FSF), en 1989 por Richard Stallman, para dar cabida al software libre.

Estipula los siguientes apartados:

- Se obliga a dar una copia del código fuente acompañado a los programas. Si nó, se debe facilitar algún método de distribución del código o incluso enviarlo por correo si es requerido.
- Se permite hacer todas la modificaciones necesarias de un programa sin que requiera costo alguno. Si es para uso empresarial o privado se pueden hacer públicos o nó, los cambios realizados, pero si el programa modificado se distribuye, su código también.
- Se puede cobrar por el software, modificado o nó, y no hay límites en el precio(desde un céntimo de Euro hasta unos miles).
- A quien se le venda o regale el software debe aplicar los puntos anteriores.
- La licencia de un programa propio pasa por la inclusión de una nota con un copyright propio aparte del que ha sido publicado bajo GPL, así como un archivo explicando la propia licencia GPL

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you

distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an

implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the

work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code

is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been

modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is

in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that

numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the

Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show
w'.
```

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.

For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

RESUMEN LICENCIA GFDL

La Licencia de documentación libre GNU(GNU Free Documentacion License), es una licencia copyleft para contenido libre, diseñada por la Fundación del Software Libre (FSF) para el proyecto GNU.

La licencia fue diseñada para la documentación del software y otras referencias y materiales institucionales.

Estipula los siguientes apartados:

- Cualquier copia del material, incluso de ser modificado, debe llevar la misma licencia.
- Dichas copias pueden ser vendidas pero, de ser producidas en cantidad, deben ser distribuidas en un formato que garantice futuras ediciones.
- Parte al uso de texto “invariable” que no puede ser modificado o eliminado y la bien intencionada pero exagerada prohibición en contra de sistemas DRM(Digital Righths Management, o gestión de derechos digitales), lo cual afecta también algunos usos válidos.
- Puede copiar y distribuir el Documento en cualquier medio, sea en forma comercial o no, siempre y cuando esta Licencia, las notas de derecho de autor, y la nota de licencia que indica que esta Licenciase aplica al Documento se reproduzca en todas las copias y que usted no adicione ninguna otra condición a las expuestas en ésta Licencia.
- Usted no puede usar medias técnicas para obstruir y controlar la lectura o copia posterior de las copias, que usted haga o distribuya.

- En caso de modificación usar en la Página de Título un título distinto al del Documento original, y de versiones anteriores a no ser que quien publicó originalmente otorgue el permiso.
- Listar en la Página de Título, como autores, una o mas personas o entidades responsables de las modificaciones en la Versión Modificada, junto con por lo menos cinco de los autores del Documento Original (Todos sus autores principales, si hay menos de cinco).
- Preservar todas las notas de derechos de autor del Documento.
- Incluir una copia sin modificaciones de ésta Licencia.
- Preservar los enlaces a webs, si hay, así como a copias anteriores.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially.

Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is

not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History"

section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you

may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may

include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have

been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.