

UNIVERSIDAD DE COSTA RICA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

MONITOREO Y ADMINISTRACIÓN DE LOS EQUIPOS DE
LA RED DEL BANCO NACIONAL DE COSTA RICA POR
MEDIO DE SNMP BASADO EN LA PLATAFORMA NNM DE
HP-OPEN VIEW

Por
CARLOS AGUILAR BOGANTES

Ciudad Universitaria Rodrigo Facio Costa Rica

Mayo, 2005

DEDICATORIA

A Dios y a mi familia,
por la fortaleza y el ánimo
que siempre me brindaron.

AGRADECIMIENTOS

Al Banco Nacional de Costa Rica, en especial al Ing. Danny Muñoz Ruiz y demás compañeros por el apoyo brindado.

Al profesor Víctor Hugo Chacón, por sus valiosos aportes para la conclusión de este proyecto de graduación.

CONSTANCIA DE APROBACIÓN

MONITOREO Y ADMINISTRACIÓN DE LOS EQUIPOS DE LA RED DEL BANCO
NACIONAL DE COSTA RICA POR MEDIO DE SNMP BASADO EN LA
PLATAFORMA HP-OPEN VIEW

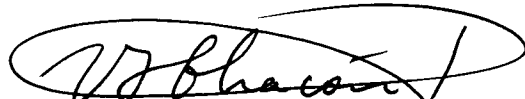
Por

Ing. Carlos Aguilar Bogantes.

Sometido a la Escuela de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de
Costa Rica, como requisito parcial para optar por el grado de:

LICENCIADO EN INGENIERÍA ELÉCTRICA.

Aprobado por el tribunal:



M.Sc. Víctor Hugo Chacón Pinedas.
Profesor guía



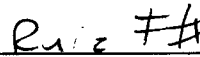
Dr. Jorge Romero Chacón
Miembro del Tribunal



Magister. Danny Muñoz Ruiz
Miembro del Tribunal



Ing. Francisco Siles Canales
Miembro del Tribunal



Ing. Federico Ruiz Ugalde
Miembro del Tribunal

Mayo, 2005

ÍNDICE GENERAL

DEDICATORIA	II
AGRADECIMIENTOS	III
CONSTANCIA DE APROBACIÓN	IV
ÍNDICE DE CUADROS	VIII
RESUMEN	IX
CAPÍTULO 1. Justificación de la elección del tema	11
1.1 OBJETIVOS GENERALES	14
1.2 OBJETIVOS ESPECÍFICOS	14
1.3. Antecedentes del tema	15
CAPÍTULO 2. Marco teórico	18
2.1 La gestión (administración de la red)	18
2.2 El protocolo simple de administración de red (SNMP)	21
2.3 El gestor Network Node Manager de HP OpenView	24
2.4 Lenguajes de programación para búsqueda de patrones y procesamiento de texto	28
2.5 Hipótesis	30
CAPÍTULO 3. Método de trabajo	31
CAPÍTULO 4. Desarrollo	35
4.1 Actividades Iniciales	35
4.2 Acciones automáticas	39
4.2.1 Generación de Respaldos	39
4.2.2 Generación de archivos para CLI	43
4.2.3 Nodo Caído	45
4.2.4 Enlace Caído	48
4.2.5 Servicios ATM Colgados	53
4.2.6 <i>Traps</i> de Salud o «Health»	55
4.2.7 <i>Traps</i> de Autenticación	61
4.2.8 Cambio de grupo	63
4.2.9 Vital Net	65
4.2.10 Reportes	67
4.2.11 Corolario de Acciones Automáticas	69
4.3 Acciones no automáticas.	72
4.3.1 Administración de Conmutadores Alcatel (Web-Vision)	73
4.3.1.1 Subrutinas	76
4.3.1.2 Lectura de las variables de estado y de las variables de entrada	77
4.3.1.3 Autenticación del usuario	78

4.3.1.4	Cambios en la configuración y registro de los cambios (Accounting)	80
4.3.1.5	Lectura de campos de MIB y procesamiento	81
4.3.1.6	Presentación en HTML	81
4.3.2	Administración de enrutadores Nortel (Web-Manager)	84
4.3.3	Corolario de Acciones No Automáticas	88
CAPÍTULO 5. CONCLUSIONES.		93
CAPÍTULO 6. RECOMENDACIONES.		95
CAPÍTULO 7. BIBLIOGRAFÍA.		97
CAPÍTULO 8. GLOSARIO.		98
ANEXO A. ACCIONES AUTOMÁTICAS		101
A.1	Archivo de <i>crontab</i>	101
A.2	Archivo de <i>generación y comprobación de respaldo para conmutadores Alcatel</i>	102
A.3	Archivo de <i>generación de archivos con formato CLI para conmutadores Alcatel</i>	104
A.4	Procesamiento del evento "Nodo Caído"	108
A.5	Procesamiento del evento "Enlace Caído"	110
A.6	Procesamiento del evento "ATM Inicial"	116
A.7	Procesamiento de los <i>traps</i> de salud	120
A.8	Procesamiento de los <i>traps</i> de autenticación	127
A.9	Procesamiento del <i>trap</i> de cambio de grupo	131
A.10	Generación de Reportes	132
ANEXO B. ACCIONES NO AUTOMÁTICAS.		136
B.1	Página inicial de Conmutadores Alcatel	136
B.2	Aplicación de Administración de Conmutadores Alcatel	138
B.3	Aplicación de Búsqueda de Direcciones IP en Conmutadores Alcatel	163
B.4	Página inicial de enrutadores Nortel	177
B.5	Marco Principal de la Aplicación de Administración de Enrutadores	178
B.6	Estado de los Circuitos IP	182
B.7	Interfaces Sincrónicas	192
B.8	Estadísticas de los Puertos Sincrónicos	202
B.9	Estadísticas de Frame Relay	211
B.10	Tabla de Rutas	220
B.11	Configuración del Protocolo RIP	228
B.12	Tabla ARP	240
B.13	Líneas de Frame Relay	249
B.14	Aplicación para Manejo de Archivos en los Enrutadores	258

ÍNDICE DE FIGURAS

<i>Figura 1. Red WAN del Banco Nacional de Costa Rica</i>	11
<i>Figura 2. Red LAN del Banco Nacional de Costa Rica en 3 capas (Acceso, Distribución y Núcleo)</i>	12
<i>Figura 3. Árbol de MIB</i>	21
<i>Figura 4. Trama SNMP</i>	22
<i>Figura 5. PDU de SNMP</i>	22
<i>Figura 6. Vista del NNM</i>	24
<i>Figura 7. Estructura del Trabajo</i>	33
<i>Figura 8. Comparación de Despliegue de mensajes</i>	37
<i>Figura 9. Proceso de Respaldo de configuraciones</i>	41
<i>Figura 10. Estructura de los respaldos en el servidor ftp</i>	43
<i>Figura 11. Diagrama de procesos para la acción Node Down</i>	46
<i>Figura 12. Consola de alarmas antes y después de utilizar el Node Down</i>	48
<i>Figura 13. Vista de Objeto no Manejado en NNM</i>	49
<i>Figura 14. Diagrama de Procesos ante el evento Link Down</i>	50
<i>Figura 15. Comparación del tratamiento de trap Link Down antes y después de procesar</i>	52
<i>Figura 16. Observación del estado Inicial de un SVC de ATM</i>	54
<i>Figura 17. Diagrama de procesos para el tratamiento del evento ATM Initial</i>	54
<i>Figura 18. Correo que recibe el usuario al tratar el evento de "ATM Initial"</i>	55
<i>Figura 19. Vista del Event Configurator del NNM al incluir la acción /usr/bin/HP_Ag/Health.pl</i>	57
<i>Figura 20. Diagrama de Procesos de /usr/bin/HP_Ag/Health.pl</i>	58
<i>Figura 21. Diagrama de procesos de /usr/bin/HP_Ag/Ejec_Health.pl</i>	59
<i>Figura 22. Mensajes de Salud en la consola de Alarmas</i>	60
<i>Figura 23. Diagrama de procesos para el evento de Falla de Autenticación</i>	62
<i>Figura 24. Correo enviado por el NNM indicando Evento de Falla de Autenticación</i>	63
<i>Figura 25. Diagrama de procesos para el evento de Cambio de Grupo</i>	64
<i>Figura 26. Vista de la consola de Alarmas cuando se cambia de grupo un puerto</i>	65
<i>Figura 27. Configuración del destino de Traps del Vital Net</i>	66
<i>Figura 28. Vista de la consola de Alarmas con evento del Vital Net</i>	67
<i>Figura 29. Diagrama de procesos para la generación de reportes de salud y caídas</i>	68
<i>Figura 30. Correo de Reporte semanal</i>	69
<i>Figura 31. Comparación de Alarmas procesadas y Alarmas Totales</i>	70
<i>Figura 32. Interacción de la página dinámica</i>	73
<i>Figura 33. Vista del Web-Vision</i>	75
<i>Figura 34. Página de autenticación del Web-Vision</i>	79
<i>Figura 35. Página de autenticación del Web-Vision</i>	83
<i>Figura 36. Vista del Web-Manager</i>	86
<i>Figura 37. Comparación entre Web-Manager y Site Manager para cambio de costo</i>	88
<i>Figura 38. Vista de Un Puerto por medio del Web Vision</i>	91
<i>Figura 39. Lista de Acceso de un conmutador</i>	92

ÍNDICE DE CUADROS

<i>Cuadro 1. Herramientas y equipos utilizados</i>	<u>34</u>
<i>Cuadro 2. Comparación del uso del módulo snmpwalk</i>	<u>36</u>
<i>Cuadro 3. Variables de estado y entrada de un formulario del Web-Vision</i>	<u>78</u>

RESUMEN

El objeto de este trabajo fue crear una plataforma de administración y monitoreo de equipos de redes adaptada a las necesidades del Banco Nacional de Costa Rica.

El Banco Nacional de Costa Rica cuenta con una de las redes de datos más grandes del país. Cualquier falla en esta red afecta el negocio y puede producir pérdidas millonarias, sin embargo, el enfoque de la alta gerencia está centrado más que nada generar altas ganancias con menor interés del aspecto tecnológico. Por esta razón la asignación de presupuesto para las compras de tecnología está limitado y debe ser muy bien justificado. Este marco lleva a que la administración de la red se haga de la forma más eficiente posible.

Para conseguir una gestión de red eficiente se propuso una plataforma no propietaria para administración y monitoreo, que pueda interactuar con programas escritos con lenguajes de búsqueda de patrones y procesamiento de texto. Una plataforma de red no propietaria se adapta perfectamente a las necesidades del banco ya que no importa la marca del equipo de red que se necesita administrar, basta con escribir programas que interactúen con la sección privada de la MIB (ver Glosario de Términos) de cada equipo.

Con el trabajo realizado se le presenta al usuario final un producto que le es más fácil de comprender, ya que se da el despliegue de las alarmas en español, el envío de correos electrónicos, una presentación gráfica de los aspectos configurables de los equipos, se elimina el despliegue de varias alarmas para indicar un mismo evento, se realizan respaldos de forma automática y se da un acceso sencillo pero seguro a estos respaldos. El usuario puede entonces resolver un problema más rápidamente y en algunos casos no esperar a que se presente una falla para actuar por lo que el negocio no se ve afectado o en caso de verse afectado llegar a la

corrección en poco tiempo. Antes de la realización de este trabajo no se tenía ninguna de estas ventajas.

Se llegó a la conclusión de que es completamente factible adaptar la plataforma de gestión de red a las necesidades del banco reduciendo el número de alarmas que no aportan información nueva lo cual hace que la consola de alarmas sea más fácil de leer y que se pueden crear aplicaciones Web para que el usuario pueda realizar cambios en la configuración de una forma sencilla y con menor posibilidad de cometer errores.

La plataforma no propietaria que se utilizó en este trabajo fue el Network Node Manager (NNM) de HP Open View, la cual fue adquirida por el Banco en el año 2000 para la administración y monitoreo de los equipos de red del edificio principal pero se estaba explotando a un nivel muy por debajo de sus capacidades reales. Además se crearon diversos programas para interactuar con esta plataforma escritos en PERL y AWK.

La selección del NNM se dio porque ya el Banco contaba con ella sin embargo se pudo haber utilizado otro software de administración como por ejemplo el OpenNMS que es un software gratis, de fuente abierta y de funciones similares al NNM.

CAPÍTULO 1. JUSTIFICACIÓN DE LA ELECCIÓN DEL TEMA

El Banco Nacional de Costa Rica (BNCR) cuenta con una de las redes de datos más grande del país cuya administración está en manos de la Unidad de Mantenimiento y Monitoreo de la Red (UMMR).

En total el BNCR cuenta con 237 equipos de comunicaciones distribuidos en el país como se muestra en las figuras 1 y 2.

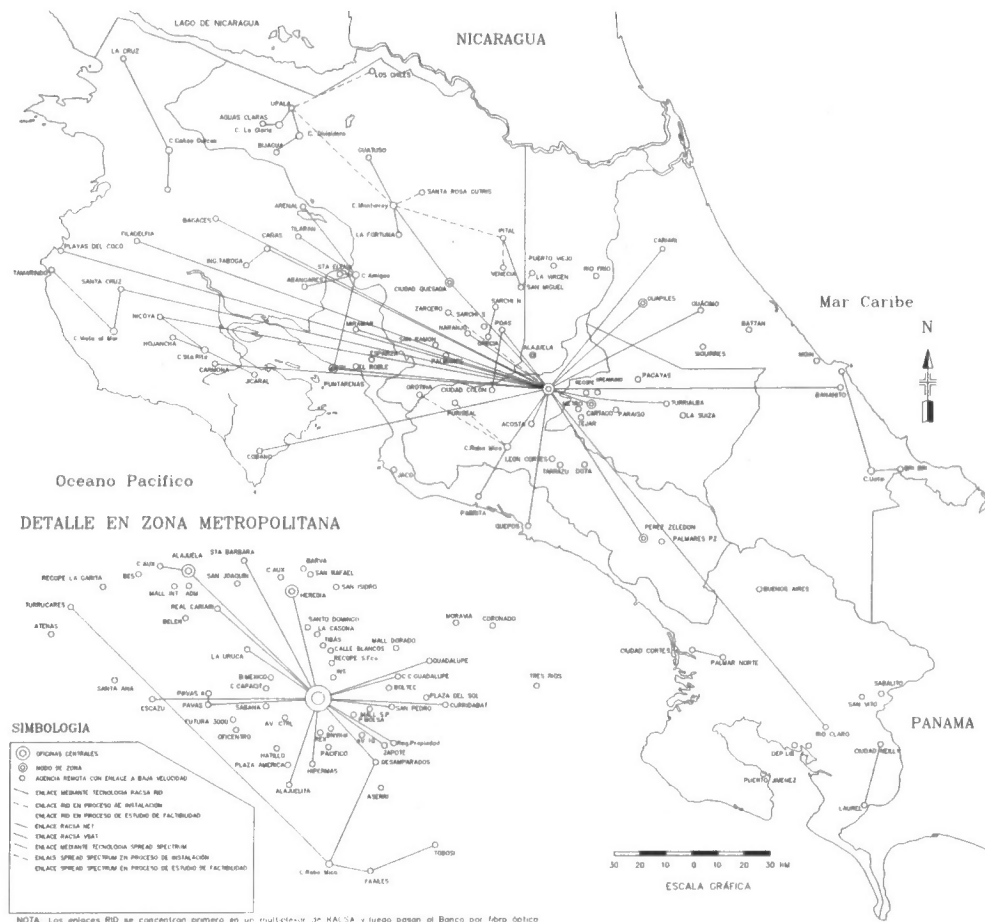


Figura 1. Red WAN del Banco Nacional de Costa Rica

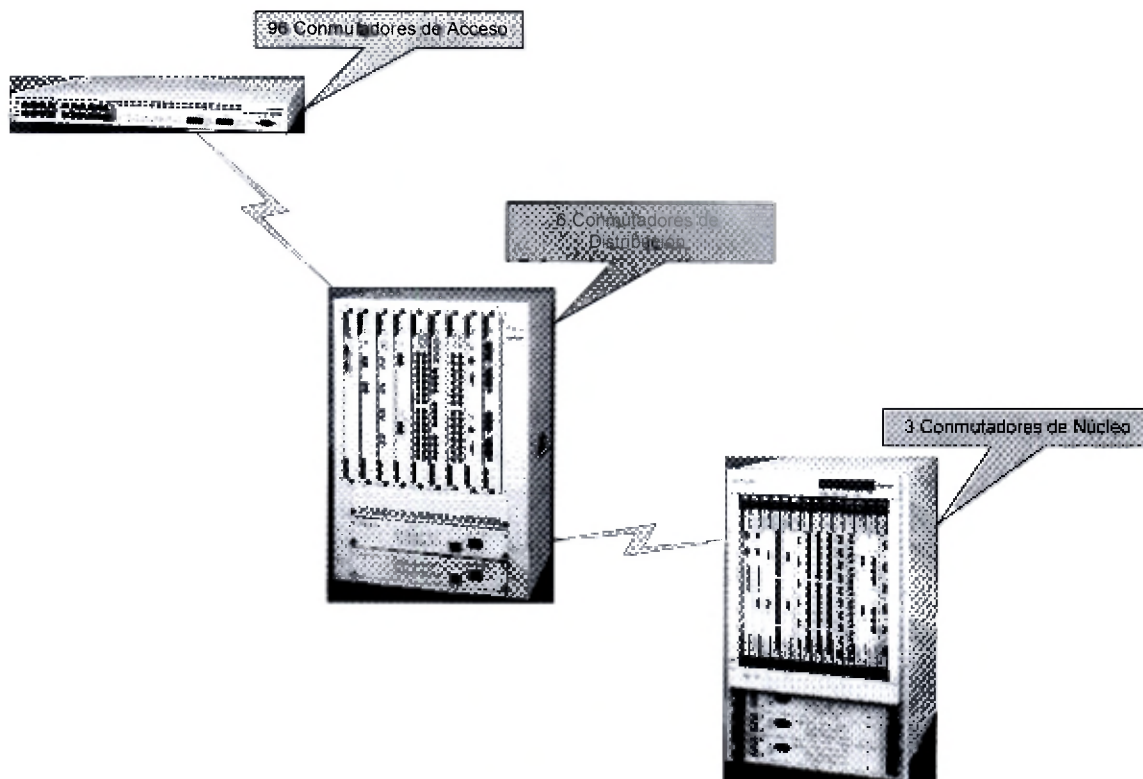


Figura 2. Red LAN del Banco Nacional de Costa Rica en 3 capas (Acceso, Distribución y Núcleo)

Los equipos de comunicación son de las marcas Nortel y Alcatel. En toda la red WAN se utilizan como medio de enrutamiento equipos Access Node (AN), Advanced Remote Node (ARN) y Advanced Snack Node (ASN) de Nortel mientras que los OmniStack 5024, Omni9 y OmniSwitch/Enrutador de Alcatel son utilizados en el Backbone de Oficinas Centrales y como casos especiales en las oficinas de La Uruca y San Pedro de Montes de Oca.

Cada una de estas marcas provee su software propietario de administración y monitoreo por medio del protocolo de administración estándar SNMP (por sus siglas en inglés Simple Network Management Protocol definido originalmente en el RFC 1157). El que el SNMP sea un estándar implica que no es necesaria una herramienta propietaria para la administración de los equipos

sino que se puede utilizar un software no propietario¹ que siga este estándar. Al utilizar este software no propietario se obtienen varias ventajas:

La primera de ellas es que basta con comprar solamente una licencia de este software no propietario y no varias de software propietarios independientes con lo que se obtiene un ahorro. Los softwares propietarios de administración como el X-Vision de Alcatel y el Site Manager de Nortel requieren una licencia por cada usuario que lo tenga instalado.

La segunda de ellas es que el software no propietario se puede configurar a gusto del usuario para mostrar solamente las alarmas que son relevantes en vez de todas las alarmas que el equipo envía. Además de permitir la reducción de “bytes” para la administración, porque la configuración de los equipos se ha hecho bajo cierto plan y cualquier configuración que se haga va a limitarse a dicho plan. Una herramienta propietaria por lo general lee la configuración completa del equipo de comunicación administrado, mientras que con una herramienta genérica completamente configurable por el usuario es posible leer solamente la información necesaria.

La tercera es que se pueden explotar otras ventajas como la generación de reportes basados en una plantilla o el envío de correo electrónico a los administradores de la red que sean necesarios. En otras palabras la herramienta de administración queda hecha a la medida del usuario al contrario de las herramientas propietarias donde el usuario debe capacitarse.

La reducción de costo en licencias de software y en capacitación es un beneficio considerable para cualquier institución. Esta se convierte en la principal justificación para desarrollar este trabajo. En un segundo plano está por supuesto el conocimiento a fondo del protocolo SNMP y de la MIB (Management Information Base) de los equipos de comunicaciones de la red del BNCR.

¹ Por software no propietario de administración se va a entender todo aquel que no esté relacionado con algún fabricante de los equipos de comunicaciones que utiliza el Banco Nacional de Costa Rica, esto es Nortel o Alcatel.

Los objetivos para este trabajo se detallan a continuación.

1.1 OBJETIVOS GENERALES

1. Crear un sistema de gestión de red eficiente y adaptado a las necesidades del Banco Nacional
2. Maximizar el aprovechamiento de la plataforma Solaris.

1.2 OBJETIVOS ESPECÍFICOS

1. Crear una administración sencilla de los equipos de red por medio de una interfaz Web.
2. Reducir el uso de documentos delicados en papel (claves, diagramas de red).
3. Realizar la integración de los equipos de comunicaciones y software de administración con el HP Open View.
4. Programar acciones tanto automáticas como manuales utilizando AWK y/o PERL.
5. Reducir tráfico de paquetes de administración innecesarios.

1.3. Antecedentes del tema

La importancia de la administración de red no es una tarea desarrollada desde el inicio de las redes mismas sino que comenzó como una necesidad posterior impulsada por su crecimiento. Se realizaron varias propuestas de arquitecturas para su manejo hasta que se definieron los estándares SNMP y CMIP. Esto se puede ver en la exposición de Chová²: «En los comienzos de Internet, se hacía hincapié en la definición e implementación de protocolos que alcanzaran la interoperación. A medida que crecía la red aparecieron situaciones en las que procedimientos desarrollados "ad hoc" para gestionar la red no eran capaces de crecer con ella. La configuración manual de tablas fue sustituida por algoritmos distribuidos automatizados y aparecieron nuevas herramientas para resolver problemas puntuales. En 1987 quedó claro que era necesario un protocolo que permitiera gestionar remota y uniformemente los elementos de una red, como los enrutadores. Se propusieron varios protocolos con este propósito, entre ellos el SNMP (Simple Network Management Protocol, protocolo simple de gestión de red) diseñado, como su propio nombre indica, buscando la simplicidad; HEMS, un diseño más complejo de la comunidad investigadora; y CMIP, desarrollado por la comunidad OSI. Una serie de reuniones llevaron a tomar la decisión de desestimar HEMS como candidato para la estandarización, dejando que tanto SNMP como CMIP siguieran adelante con la idea que el primero fuera una solución inmediata mientras que CMIP pasara a ser una aproximación a largo plazo: el mercado podría elegir el que resultara más apropiado. Hoy SNMP se usa casi universalmente para la gestión de red.»

La primera versión del protocolo SNMP quedó completamente definida en mayo 1990 cuando J. Case (SNMP Research), M. Fedor (Performance System Internacional), M. Schoffstall (Performance System Internacional) y J. Davin (MIT Laboratory for Computer Science) publicaron el *RFC-1157: A Simple Network Management Protocol (SNMP)* con el Internet Engineering Task Force (IETF).

² Ver referencia No.3

Se han realizado bastantes cambios en la versión original. De hecho existen al menos 31 documentos RFC relacionados con SNMP. Los más importantes de esta lista son el ya mencionado RFC-1157, el RFC-2274 publicado en Enero de 1998 donde se define la versión 3 de SNMP y el RFC-2578 publicado en Abril de 1999 que define la versión 2 de SMI (Structure of Management Information). En todos estos cambios se ha notado la participación de la industria de las telecomunicaciones con representantes como Cisco e IBM.

El papel de la industria en la definición de SNMP ha hecho que prácticamente todos los fabricantes de equipos de comunicación incluyan un agente de SNMP en los códigos fuente de los equipos.

En forma paralela ha existido el desarrollo de los gestores conocidos en inglés como *Network Management Software* (Software de Manejo de Redes) como contraparte de los agentes de SNMP. Cada fabricante de equipos comenzó la venta de su gestor propietario y poco después aparecieron los gestores genéricos como el Network Node Manager de Hewlett-Packard y Tivoli de IBM.

La utilización de los gestores en el Banco Nacional no ha seguido un crecimiento ordenado sino que se han formado varias islas. La política ha sido adquirir los gestores propietarios de cada equipo. Por esta razón se cuenta con Optivity y Site Manager para los equipos Nortel, X-Vision para plataforma Alcatel y Cisco Works, recientemente adquirido para los equipos Cisco que conforman el «Front End» de los servidores, además de Network Node Manager de HP Open View que se adquirió con la plataforma de los equipos Alcatel para funcionar en forma paralela a Optivity.

La plataforma propietaria X-Vision para los conmutadores (switches) Alcatel es la que presenta las desventajas más grandes ya que no es una plataforma multiusuario sino que la puede

usar solamente una persona a la vez. Esto hace que sea una plataforma que no es útil para el tipo de administración de red que utiliza el BNCR.

Las plataformas de gestión mencionadas anteriormente funcionan de forma independiente a pesar de que en el fondo todas utilizan SNMP por lo que se puede usar sólo una.

En el Banco Nacional de C.R se han realizado varios trabajos de graduación con el tema de la administración de red. Los más recientes son: Establecimiento de la Administración Proactiva en la Red de datos del Banco Nacional de Ronald Saborío Blanco. Este documento es un estudio administrativo que compara la situación de monitoreo actual que utiliza el Banco Nacional de C.R con los estándares OSI y UIT para dar recomendaciones tanto en personal técnico como en tecnología. Otra fue la propuesta de un centro de gestión de Monitoreo de la Red de Maricel Herrera Odio. Este trabajo es completamente comercial donde se buscan herramientas que están a la venta en el mercado nacional y que pueden servir para llevar a cabo una administración y monitoreo adecuados. La última fue la propuesta para la implementación de un sistema de gestión de servicios para el Banco Nacional de Costa Rica de Christiam Gómez que concluye con un cartel de compra cuyo objetivo fue la administración centralizada de los equipos de red.

Todos los trabajos mencionados están orientados a cambios administrativos o a compras de equipo. Con este trabajo se pretende no sólo trabajar con lo que el banco ya cuenta sino que también tiene un enfoque completamente técnico. Sin embargo el que se utiliza más como referencia es el Establecimiento de la Administración Proactiva en la Red de datos del Banco Nacional.

CAPÍTULO 2. MARCO TEÓRICO

2.1 La gestión (administración de la red)

Se entiende por gestión de redes y servicios de telecomunicaciones al conjunto de actividades destinadas a garantizar los servicios que prestan las redes.

Los recursos informáticos están interconectados mediante medios de transmisión y protocolos de comunicación organizados en lo que se conoce como sistema de comunicación. Estos sistemas de comunicación están implementados mediante infraestructura de equipos de comunicación (módem, enrutadores, multiplexores, etc.) y facilidades de transmisión. Estos sistemas son los que prestan los servicios finales, que los usuarios utilizan en la actividad diaria de las oficinas.

El tamaño y la complejidad de las redes han ido creciendo sin cesar debido en gran parte a la aparición de las redes públicas de datos y a la creciente oferta de servicios de comunicación de valor agregado. Actualmente los sistemas de comunicaciones prestan servicios a los usuarios utilizando redes privadas y redes públicas. La interconexión entre las mismas proporciona mejores posibilidades en la provisión de servicios pero complica el control de las redes. Habiendo conseguido la transferencia de información a través de esta complejidad de redes, surge la necesidad de gestionarlas, es decir, de controlar los recursos que las componen en términos de rendimiento, capacidad, utilización, reconfiguración, diagnósticos, planificación, entre otras.

El término gestión utilizado en muchos y diversos entornos de nuestra vida diaria, está relacionado con la planificación, el seguimiento, costos, control de recursos y actividades. Al aplicar este término al entorno de redes, la gestión de red comprende la administración de los

diferentes recursos que constituyen una red. La gestión de red toma la forma de seguimiento, coordinación y control de los recursos informáticos y de comunicación.

Los sistemas de comunicación pueden ser muy extensos y complejos en cuanto a que están constituidos por gran número de equipos y a su vez estos ofrecen gran diversidad, tal y como se ha venido mencionando, como por ejemplo: teléfonos, centrales telefónicas, módems, multiplexores, concentradores, enrutadores, conmutadores, terminales, etc.

Las fallas en los sistemas de comunicación son inevitables y el tiempo de no funcionamiento de los mismos es muy caro para las organizaciones, principalmente en entidades financieras.

Existe gran diversidad de sistemas de gestión de redes, que refleja en gran medida la diversidad de equipos y servicios de telecomunicaciones existentes. Una clasificación de forma muy general de los tipos de gestión es la siguiente:

- Gestión de averías (o mantenimiento).
- Gestión de la configuración.
- Gestión del desempeño.
- Gestión de la contabilidad.
- Gestión de la seguridad.

A la vez, los fabricantes de equipos de comunicación han desarrollado también sistemas de gestión de sus propios equipos por la necesidad de los usuarios de controlar los servicios proporcionados por dichos equipos.

La mayor parte del desarrollo de normas para la gestión se basan en las dos corrientes siguientes:

- Gestión ISO para las redes en protocolos OSI.
- Gestión SNMP para las redes con protocolos TCP/IP

Para el caso del Banco, que cuenta con un gran número de diversidad de equipos, la gestión de todos ellos se traduce en la necesidad de tener varios sistemas de gestión, esto hace que la operación y control sea una tarea muy compleja, por lo que se requiere un sistema de gestión centralizado para controlar todos los recursos de comunicaciones.

En el momento actual, la integración de diversos sistemas de gestión es un requisito fundamental en las organizaciones. Para llegar a ello se requiere un entendimiento en tres niveles:

- Definición de recursos gestionados. Los diferentes sistemas gestionan los mismos recursos pero no exactamente los mismos parámetros de cada uno, ni de la misma manera.
- Definición de las operaciones a realizar sobre ello. De la misma manera los diferentes sistemas de gestión realizan distintas operaciones sobre recursos de comunicación similares.
- Protocolos de comunicación utilizados. Para poder realizar transferencias de información entre sistemas de gestión diferentes es imprescindible utilizar protocolos comunes.

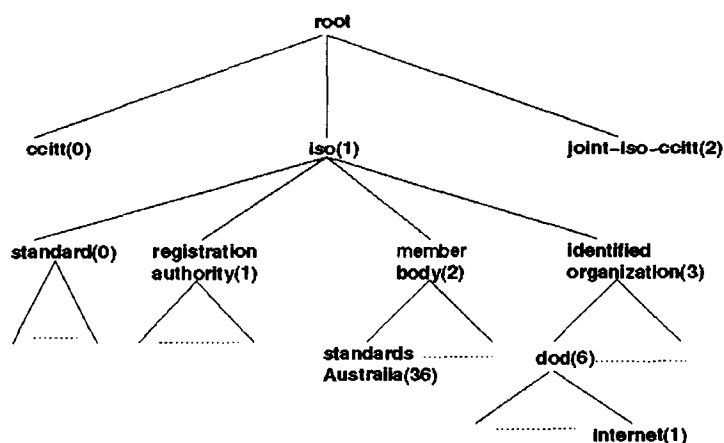
El protocolo de comunicación para la administración de red más utilizado es el SNMP, el cual se explica brevemente en el siguiente apartado.

2.2 El protocolo simple de administración de red (SNMP)

El SNMP se ha convertido en el protocolo preferido para la administración de las redes TCP/IP en el mundo. La primera definición formal de este protocolo se dio con el RFC 1157 en 1990 y surgió como una recomendación de la Internet Architecture Board (IAB) de que todas las implementaciones de IP y TCP fueran administrables desde otro punto de la misma red.

El objetivo principal de este protocolo es que sea un protocolo sencillo mediante el cual la información de administración de un elemento de la red pueda ser inspeccionada o alterada por usuario remoto. Otras metas en el diseño son buen tiempo de respuesta por lo que el protocolo está encapsulado en UDP en vez de TCP, robustez y habilidad para manejar una red de baja disponibilidad por lo que se hizo SNMP un protocolo donde no importa cuales hayan sido las requisiciones anteriores. Se buscaba también una fácil migración al CMIP por lo que se escogió el ASN.1 como la capa de presentación y el uso de una base de información común.

La información que se puede obtener del agente de SNMP es aquella que se guarda en la MIB (Management Information Base) que es una base de datos organizada en forma de árbol que guarda información estadística y también de la configuración del equipo donde corre el agente.



(Tomado de Referencia 9)

Figura 3. Árbol de MIB

Para representar un campo en la MIB se utiliza la notación de números separados por puntos. Por ejemplo 1.3.6.1 es el identificador de internet ya que al seguir el árbol el campo 1 es iso que tiene las ramas 0, 1, 2 y 3 donde la tercera corresponde a las organizaciones identificadas. La rama 6 de las organizaciones identificadas es el departamento de defensa de los Estados Unidos (DOD) y la rama 1 del departamento de defensa es internet.

Una trama de SNMP es bastante sencilla y se muestra en la figura 4.



Figura 4. Trama SNMP

Versión

El campo versión es de 3 bytes y especifica el tipo de versión que se está utilizando, el agente y el gestor deben utilizar la misma versión. Los paquetes con diferente número de versión son eliminados sin ningún otro procesamiento.

Comunidad

La comunidad es un campo de tamaño variable de tipo carácter y se utiliza como un medio simple para autenticación. Para poder comunicarse con el agente el gestor debe tener la misma comunidad y es sensible a mayúsculas y minúsculas.

PDU

La unidad de datos de protocolo (PDU) contiene la información mostrada en la figura 5.

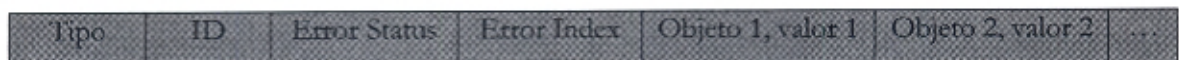


Figura 5. PDU de SNMP

Tipo

El tipo es un campo de 2 bytes que especifica la operación que se realiza.

- Tipo 0 *GetRequest*. Con este se obtiene el valor del Objeto especificado
- Tipo 1 *GetNextRequest*. Con este se obtiene el valor del objeto siguiente al especificado
- Tipo 2 *GetResponse*. Es la respuesta a cualquiera de los tipos de Request
- Tipo 3 *SetRequest*. Con una instrucción set se hace un cambio en el objeto de la MIB y es muy útil para hacer cambios en la configuración.
- Tipo 4 *Trap*. Los *traps* son alarmas. Estas son generadas por el agente de SNMP cuando ocurre algún evento que se desea anunciar al gestor. Es el único tipo que no tiene una solicitud previa.

ID

El ID es un campo de 3 bytes que se utiliza para correlacionar el *GetRequest* con el *GetResponse*. Como se mencionó SNMP está encapsulado en UDP por lo que no hay seguridad de que los paquetes lleguen o que lleguen en un orden distinto por ser no orientado a conexión. Con el ID en un *GetResponse* el gestor puede determinar cual paquete que envió generó esta respuesta.

Error Status y Error Index

Son campos de 3 bytes cada uno. El Error Status indica operación normal o uno de 5 tipos de errores posibles. El Error Index identifica la entrada entre la lista de variables (objetos) que causó el error.

- Status 0 *noError*. Ningún error. Operación apropiada.
- Status 1 *tooBig*. El tamaño de la respuesta excede el máximo tamaño del PDU
- Status 2 *noSuchName*. Se solicitó el valor de un objeto que no existe en el agente.

- Status 3 *badValue*. Ocurre cuando se da un SetRequest con un tipo inconsistente. Por ejemplo tratar de colocar una cadena de caracteres en un objeto tipo entero.
- Status 4 *readOnly*. Datos que son de solo lectura.
- Status 5 *genErr*. Ocurrió otro tipo de error no especificado.

2.3 El gestor Network Node Manager de HP OpenView

El Network Node Manager (NNM) de HP Open View es un software de gestión con orientación cliente-servidor que permite la interacción con los elementos de la red por medio del protocolo SNMP. Tiene además la ventaja de ser completamente gráfico y amigable con el usuario y tiene la posibilidad de poder interactuar con otros sistemas.

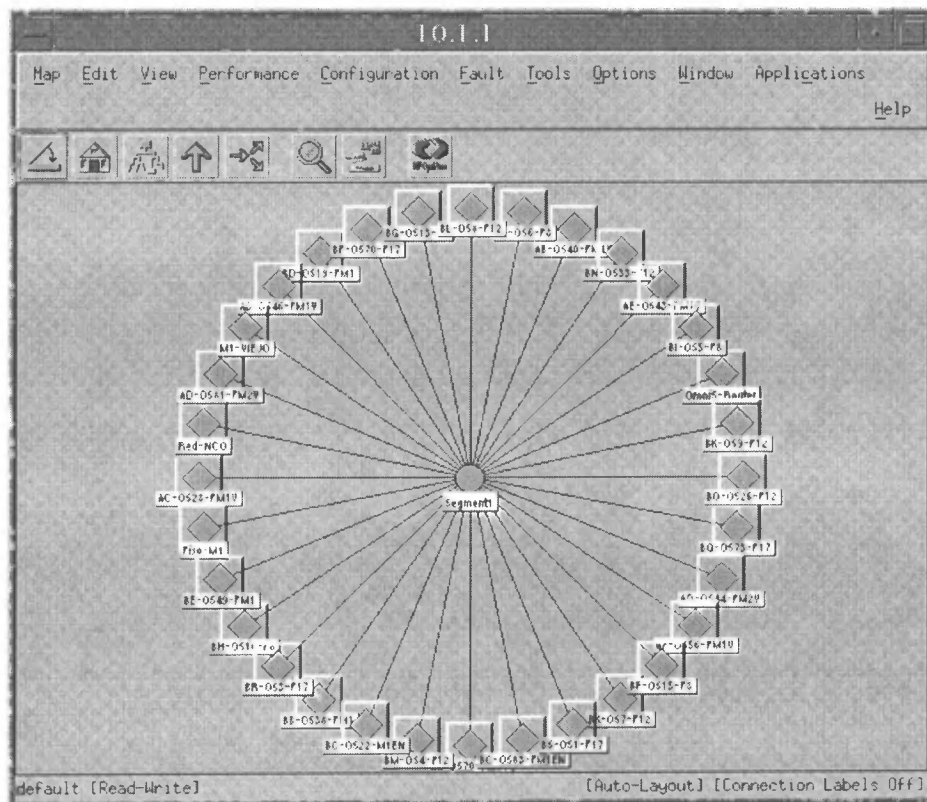


Figura 6. Vista del NNM

NNM tiene uso principalmente en 3 de las cinco áreas de la gestión de red: Gestión de Fallas, Gestión de Desempeño y Gestión de Configuración.

La arquitectura de NNM es bastante completa, se comunica con los agentes distribuidos en la red por medio del protocolo SNMP. Para realizar todas las funciones se cuenta con dos tipos de servicios:

- Servicios en el segundo plano (Background) que están corriendo continuamente independientemente de si está corriendo la aplicación ovw.
- Servicios en el primer plano (Foreground) los cuales corren solamente cuando ovw está corriendo.

A continuación se detallan los servicios más importantes que corren en primer plano que corren en primer plano. La lista total se encuentra en la referencia 7.

ovstart	Arranca los diferentes servicios que levanta el Network Node Manager. Solo en el inicio del NNM.
ovspmd	Lanza y administra todos los servicios. Ovspmd interactúa con los comandos ovstart, ovstop, ovstatus, ovpause y ovresume. Solamente trabaja en el inicio del NNM.
httpd	Maneja las solicitudes http. httpd es un servidor Web
netmon	Realiza la interrogación a los agentes SNMP para descubrir la red y luego detecta cambios en la configuración, topología y status en los elementos de la red. Como puede observarse netmon es uno de los servicios claves del NNM.
ovactiond	Recibe eventos del proceso pmd y ejecuta comandos
ovalarmsrv	Provee información de eventos al Alarm Browser.

ovsessionmanager.exe	Maneja las sesiones Web de los usuarios. ovsessionmanager.exe es iniciado por ovstart.
ovtopmd	Mantiene la base de datos de la topología de la red. La base de datos de la topología es un conjunto de archivos que conservan el resultado de los poleos que realiza netmon sobre el status y los objetos de la red incluida su interrelación.
ovtrapd	Recibe los <i>traps</i> de SNMP y los envía a pmd. También responde a las solicitudes de SNMP v2.
ovuispmd	Maneja los servicios de la interfase del usuario de NNM. Antes de correr ovw ovuispmd debe estar corriendo.
ovwdb	Controla los objetos de la base de datos. Los objetos de la base de datos almacenan información semántica sobre los objetos existentes en la red.
pmd	Recibe y envía eventos y logs de eventos a la base de datos de eventos. pmd también reenvía los eventos de la red a otras aplicaciones que están conectadas a pmd por medio del API de SNMP.

Los siguientes son los principales procesos que corren en primer plano, esto quiere decir que no corren si no se ha abierto la aplicación ovw. La lista completa de los procesos de primer plano se puede consultar en la referencia 7.

ipmap	Dibuja la topología en los mapas que representan la red.
ovw	Es el servicio que provee los mapas, la edición de los mapas y el manejo de menús. Cuando se inician los servicios al ejecutar el comando ovw, automáticamente se invocan ipmap, xnmevents y otras aplicaciones.

xnmbrowser	El ítem del menú <i>Tools: SNMP MIB Browser</i> que permite consultar y definir valores MIB para los objetos que cumplen el standard y objetos MIB específicos de algún fabricante particular.
xnmbuilder	El ítem del menú <i>Options: MIB Application Builder:SNMP</i> que permite construir pantallas echas por el usuario para manejar objetos MIB de diferentes fabricantes. La información que el usuario define utilizando el MIB Application Builder es guardada en los archivos de registro y los archivos de ayuda utilizando <i>mibform</i> , <i>mibtable</i> y <i>xnmgraph</i> .
xmnevents	Es el Alarm Browser que es invocado automáticamente con <i>ovw</i> para desplegar los eventos que son recibidos por <i>pmd</i> .
xnmloadmib	El ítem del menú <i>Options: Load/Unload MIBs:SNMP</i> usado para cambiar a nuevos estándares de MIB o MIBs, propios de otros fabricantes en la base de datos.
xnmpolling	El ítem del menú <i>Options:SNMP Configuration</i> que permite añadir, borrar y modificar parámetros de la configuración de SNMP y el status del intervalo de poleo del servicio <i>netmon</i> . La configuración de los parámetros SNMP incluye la comunidad de lectura, comunidad de escritura, número de reintentos e información del proxy.
xnmtrap	Es el evento invocado por el ítem del menú <i>Options:Event Configuration</i> que permite definir <i>traps</i> específicos de algún fabricante en especial.

2.4 Lenguajes de programación para búsqueda de patrones y procesamiento de texto

Los lenguajes para procesamiento de texto aparecieron desde la década de los 70 con el sistema operativo Unix. El objetivo de estos era realizar búsquedas de patrones y cambios de todo tipo en texto con un lenguaje sencillo para el programador.

Una integración adecuada entre los elementos del gestor NNM, el sistema operativo y los usuarios se puede realizar por medio de programas escritos con este tipo de lenguajes.

Los lenguajes usados en este trabajo no son compilados sino que son interpretados, esto quiere decir que no se genera un archivo objeto sino que el interpretador va ejecutando línea por línea del código fuente.

Para la realización de este trabajo se usaron el AWK y el PERL, dos lenguajes de procesamiento de texto descritos brevemente a continuación.

AWK

La primera versión de AWK fue escrita en 1977 en los Laboratorios Bell por Alfred Aho, Peter Weinberger y Brian Kernighan. El nombre del lenguaje proviene de las iniciales de los apellidos de los autores. Este lenguaje está basado en C, ed, egrep y snobol.

En 1985 una nueva versión hizo el lenguaje más poderoso introduciendo la facilidad de definición de funciones del usuario.

En 1986 Paul Rubin escribió la implementación de GNU de AWK conocida actualmente como gAWK. En 1997 Jürgen Kahrs le incluyó la posibilidad de conexión a red al gAWK que comenzó a utilizarse con la versión 3.1.

La función básica de este lenguaje es leer archivos por líneas (u otras unidades de texto) que contienen ciertos patrones. Cuando una línea contiene el patrón buscado, AWK ejecuta la acción especificada en esa línea. AWK sigue procesando las líneas de entrada de esta manera hasta que llega al final del archivo que sirve como entrada. Si no se especifica ningún archivo como entrada se toma la entrada de datos estandar del sistema operativo.

PERL

PERL (*Practical Extraction and Reporting Language*) es uno de los lenguajes más antiguos de aquellos utilizados comúnmente para el uso en la WEB. PERL está basado en C y posteriormente recibió la influencia de otros lenguajes como **BASIC**, **AWK**, **sed** y el **shell** de UNIX.

Sus orígenes se remontan a 1987 y, desde entonces, PERL ha ofrecido cuatro versiones más. El lenguaje fue desarrollado por Larry Wall y fue distribuido en forma gratuita desde su inicio. En 1988 y 1989 aparecieron las versiones 2.0 y 3.0 respectivamente. En 1991 se liberó la versión 4.0. En 1995 salió la versión 5.0 que fue un cambio radical comparado con las versiones anteriores.

En la versión 5 se realizó una reescritura completa del intérprete para aumentar la velocidad, eficacia y funcionalidad. Además se incluyó el soporte para módulos lo que permite que el programador pueda incluir bibliotecas y características orientadas a objetos, así como también más diagnósticos y herramientas para corrección de errores.

Actualmente existe la versión 6.0 de PERL pero su uso no se ha extendido debido a lo reciente que es.

El manejo de las expresiones regulares en PERL es mucho más sencillo que en AWK así como también los trabajos con archivos y la interacción con el sistema operativo.

2.5 Hipótesis

Es posible centralizar el monitoreo y la administración de todos los conmutadores y enrutadores de la red del Banco Nacional de Costa Rica en una sola máquina de Gestión de Red.

CAPÍTULO 3. MÉTODO DE TRABAJO

Como se mencionó Network Node Manager de HP Openview es un sistema abierto. Como sistema abierto puede interactuar con otros sistemas como el sistema operativo o sistemas hechos por un programador. Con este trabajo se busca que el NNM interactúe con otros programas para hacer la gestión en una forma ordenada y adaptada a las necesidades de los técnicos del Banco.

El trabajo se dividió en dos tareas: Programas automáticos y programas no automáticos.

Para poder elaborar estas acciones lo primero que se debe hacer es estudiar la MIB de los equipos que se van a administrar. Con esto se puede conocer cuáles objetos leer o modificar en caso de que se desee obtener alguna estadística o se quiera realizar un cambio en la configuración.

Los programas automáticos son aquellos que se ejecutan sin intervención del usuario, son disparados por tiempo o por algún evento de la red que indique que hay algún problema. Los programas no automáticos tienen que ser llamados por el usuario y utilizan algunos de los elementos del Network Node Manager para extraer información o para realizar un cambio en la configuración de alguno de los equipos gestionados en la red.

Crear una plataforma de administración completamente adaptada a los equipos de red del Banco, tomaría mucho tiempo, si una sola persona la realiza. Por esta razón solo se concentró en las acciones que más se necesitan. Basado en la experiencia del personal de averías de la Unidad de Mantenimiento y Monitoreo de la Red del Banco Nacional se determinaron cuáles eran las principales acciones (de ambos tipos) que se deberían ejecutar. El criterio principal fue realizar una programación que permita hacer los cambios en la configuración que se hacen con más

frecuencia y para el caso de las alarmas poner atención en aquellas que se presentan en mayor cantidad y las que son más críticas para la operación de la red.

Los puntos más importantes de toda la red se encuentran en el edificio de Oficina Principal donde se encuentran instalados los equipos Alcatel. Por esta razón se les presentó mayor atención a estos equipos.

Las acciones automáticas se realizaron utilizando el shell de Solaris (ksh), AWK y PERL.

Las acciones no automáticas se realizaron con una interfaz Web por medio de la CGI (*Common Gateway Interface*) del servidor httpd del NNM. La programación se hizo por medio de PERL.

La figura 7 muestra el procedimiento del trabajo realizado.

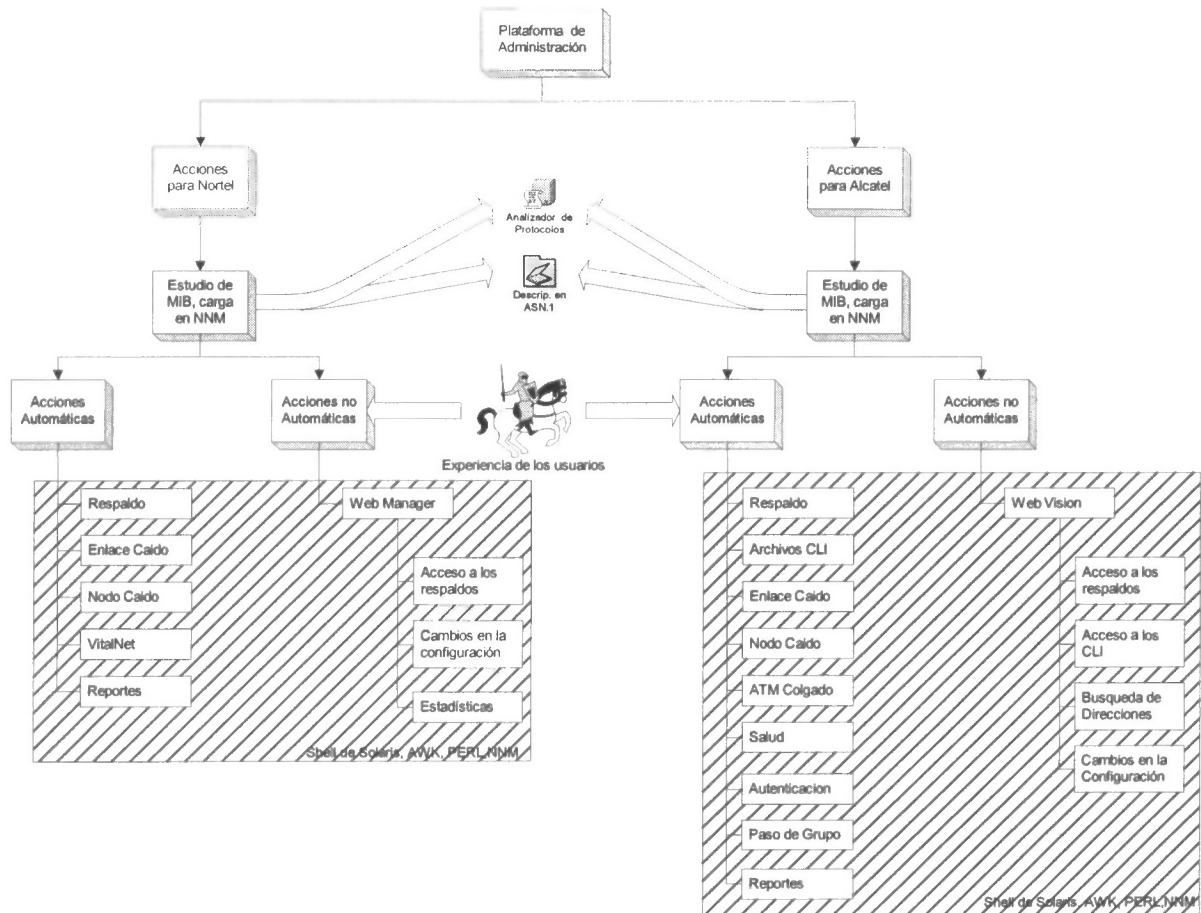


Figura 7. Estructura del Trabajo

Para este trabajo se utilizó solamente hardware y software que tuviera el banco por lo que el costo de realización no fue ninguna carga económica para la Institución. Los materiales utilizados se muestran en el Cuadro 1.

Cuadro 1. Herramientas y equipos utilizados

Material	Características
Estación SUN Ultra10 (HW)	Sistema Operativo: Solaris 8 Generic_108528-27 Procesador: sparcv9 440 MHz Memoria cache del procesador: 2 MB Memoria RAM: 512 MB
Network Node Manager (SW)	Versión: B.06.20 Tipo de Licencia: Enterprise (Sin límite de nodos)
PERL (SW)	Versión: 5.00503
AWK (SW)	Versión: 3.1
Analizador de Protocolos Portátil (SW)	Marca: Sniffer Versión:4.70.04
Macromedia Dreamweaver (SW)	Versión 6.0
VitalNet (SW)	Versión 9.1.0.24

Nota: (SW) indica herramientas de software. (HW) indica herramientas de hardware.

CAPÍTULO 4. DESARROLLO

A continuación se describe con detalle todo el proceso de adaptar el sistema de gestión abierto Network Node Manager a las necesidades de los técnicos de comunicaciones del Banco Nacional de C.R. y sus resultados.

4.1 Actividades Iniciales

Las actividades iniciales preparan el camino para el desarrollo de las actividades automáticas y las no automáticas.

Durante todo el proceso de adaptación se hace uso intensivo de varios módulos del NNM, principalmente *snmpwalk*, *snmpset* y *snmpset*. Estos módulos fueron desarrollados por la Universidad Carnegie-Mellon y la Compañía Hewlett-Packard para poder obtener valores de las instancias de los objetos en la MIB del agente por medio del protocolo SNMP. Los tres utilizan la base de datos *ovsnmp.conf_db* del NNM. Esta base contiene entre otras cosas la comunidad con la que se debe hacer la consulta al agente y la correlación entre la representación numérica de la MIB con un nombre, por ejemplo:

```
wfCSMACDEntry = 1.3.6.1.4.1.18.3.4.1.1  
wfCSMACDAutoNegEntry = 1.3.6.1.4.1.18.3.4.16.1.1  
wfFddiEntry = 1.3.6.1.4.1.18.3.4.4.1
```

El primer paso es entonces cargar la descripción de la MIB en el lenguaje ASN.1 en la base de datos *ovsnmp.conf_db*. Esto se realiza de una forma gráfica con *xnmloadmib*.

Una vez que se ha cargado la descripción de la MIB en la base de datos mencionada el resultado de ejecutar *snmpwalk* o *snmpget* es mucho más sencillo de procesar para uno de los lenguajes de búsqueda de patrones (PERL, AWK).

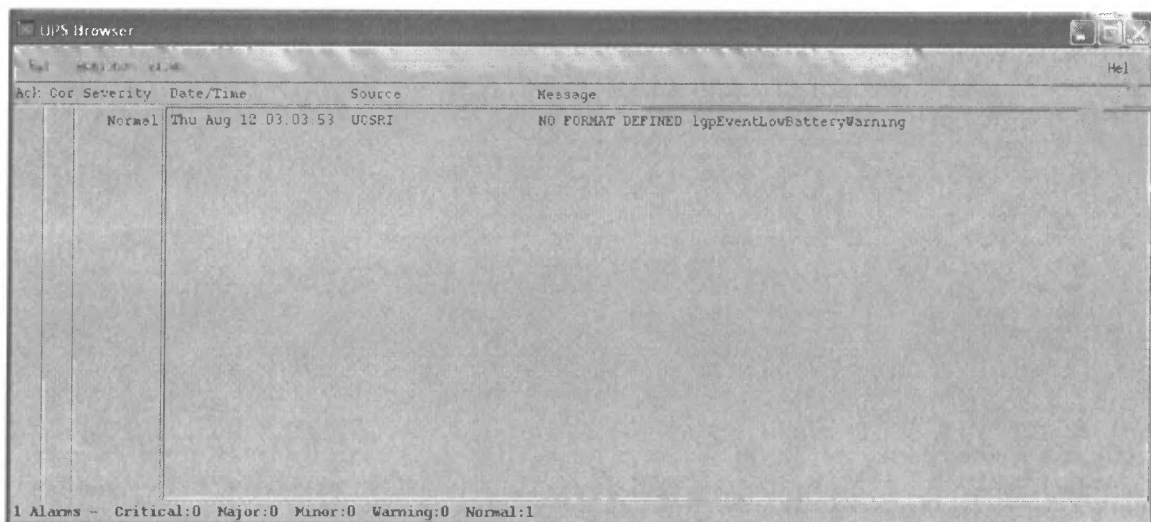
El Cuadro 2 muestra una comparación del antes y después de realizar la carga.

Cuadro 2. Comparación del uso del módulo snmpwalk

<p><u>Caso A:</u> Resultado de ejecutar un snmpwalk antes de la carga de la descripción de la MIB:</p> <pre> .1.3.6.1.4.800.2.3.1.1.1.14.2.1.4.1 : INTEGER 1 .1.3.6.1.4.800.2.3.1.1.1.14.2.2.4.1 : INTEGER 2 .1.3.6.1.4.800.2.3.1.1.1.14.3.1.4.1 : INTEGER 1 .1.3.6.1.4.800.2.3.1.1.1.14.3.2.4.1 : INTEGER 1 .1.3.6.1.4.800.2.3.1.1.1.14.3.3.4.1 : INTEGER 1 .1.3.6.1.4.800.2.3.1.1.1.14.3.4.4.1 : INTEGER 1 </pre>
<p><u>Caso B:</u> Resultado de ejecutar un snmpwalk después de la carga de la descripción MIB</p> <pre> xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportAdmStatus.2.1.4.1 : INTEGER: enable xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportAdmStatus.2.2.4.1 : INTEGER: disable xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportAdmStatus.3.1.4.1 : INTEGER: enable xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportAdmStatus.3.2.4.1 : INTEGER: enable xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportAdmStatus.3.3.4.1 : INTEGER: enable xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportAdmStatus.3.4.4.1 : INTEGER: enable </pre>

Puede observarse que después de cargar la descripción de la MIB en la base de datos del NNM se puede distinguir perfectamente a cuál objeto se está haciendo referencia, también se puede distinguir perfectamente el objeto y la instancia (2.1.4.1) lo cual es muy difícil en el caso A. También se nota que el valor de cada instancia queda traducido: en el caso B puede leerse *enable* o *disable* mientras que en el caso A se lee un número (1 ó 2) que el usuario debe interpretar. El hecho de ser capaz de distinguir todos estos elementos es crucial en el momento de definir una expresión regular con un lenguaje de búsqueda de patrones y procesamiento de texto.

Al cargar de la descripción de la MIB se configuran los servicios *ovtrapd*, *pmd* y *ovactiond*. En conjunto estos dos elementos reciben e interpretan los *traps* de los dispositivos y los muestran al usuario en la consola de alarmas. El formato en que deben ser mostrados queda guardado en el archivo *trapd.conf*. *xnmloadmib* también se encarga de incluir el formato adecuado en *trapd.conf* basado en la descripción en ASN.1 de los *traps*. La figura 8 muestra la comparación de la presentación de los *traps* al usuario.



(a)



(b)

Figura 8. Comparación de Despliegue de mensajes

En la figura 8 (a) se observa el mensaje “NOT FORMAT DEFINED” ya que el NNM no sabe como interpretar el mensaje recibido. En la figura 8 (b) los mensajes indican el tipo de problema y otra información contenida en el *trap*. Aunque el mensaje en la figura 8 (b) todavía necesita cierta interpretación (como por ejemplo en los dígitos hexadecimales) ya se tiene un mejor detalle que puede ser procesado por los lenguajes de búsqueda de patrones y procesamiento de texto.

Otra tarea mucho más sencilla pero de gran importancia es la resolución de nombres de los equipos de red. Para cualquier persona es más sencillo recordar un nombre que una dirección IP en la notación decimal puntuada. El hecho de habilitar esta resolución de nombres facilita la operación de los técnicos de comunicaciones que pueden hacer referencia por nombre a cualquier equipo sin necesidad de tener documentación de las direcciones IP de los equipos.

La habilitación de resolución de nombres se realiza en el sistema operativo. En Solaris se debe editar el archivo */etc/nsswitch.conf* y el campo de hosts asignarlo a files, además se debe llenar el archivo */etc/hosts* con la lista de las direcciones de los equipos y el nombre asociado.

En el archivo */etc/hosts* se incluyeron las direcciones IP y el nombre de los equipos. En el caso de los conmutadores de oficinas centrales se utilizó la convención de cableado estructurado para asignar el nombre. Para los enrutadores se definió un nemónico de 4 letras relacionado con el lugar donde se encuentra instalado el equipo.

Con estas tareas listas se procedió a realizar las tareas automáticas que se describen a continuación.

4.2 Acciones automáticas

Como se había mencionado las acciones automáticas son aquellas que no requieren ninguna intervención de los usuarios del sistema.

En la categoría de acciones automáticas se encuentran las tareas repetitivas pero de alta importancia como los respaldos de las configuraciones y la revisión de las alarmas que se han presentado en mayor cantidad durante cierto período. También se encuentran las que son disparadas por un *trap* como por ejemplo dar una interpretación de una alarma y determinar si el *trap* indica un síntoma de una situación superior o si es una situación aislada.

Este tipo de tareas se incluyen en la gestión de averías y gestión de la contabilidad.

El poder programar las tareas repetitivas permite que los técnicos se ocupen de tareas que requieren mayor capacidad de razonamiento. La interpretación de los *traps* permite que la consola de alarmas sea más fácil de leer. Para determinar a cuales *traps* se les va a dar tratamiento se observó cuales son los que se presentan con mayor frecuencia o son de alta importancia. Sin embargo con suficiente tiempo se puede realizar el mismo tratamiento para todos los *traps*.

4.2.1 Generación de Respaldos

Las configuraciones de los equipos de red quedan guardadas en algún medio de almacenamiento no volátil, generalmente memoria tipo “flash”. De esta forma si el equipo pierde la alimentación eléctrica no es necesario desplazarse y configurarlo desde sus valores iniciales de fábrica sino, solamente, esperar a que lea la configuración almacenada después de completar el ciclo de arranque.

Estos medios de almacenamiento no volátil pueden presentar fallos por lo que es importante mantener un respaldo en servidor ftp que esté a la disposición de los técnicos.

Un cambio en la configuración activa del equipo provoca que el respaldo sea obsoleto por lo que es importante que periódicamente se realicen los respaldos en las configuraciones en el servidor ftp.

Además de hacer el respaldo es muy importante realizar una comprobación de fidelidad del respaldo de los archivos.

El proceso del respaldo es un programa que se puede representar por el proceso de la figura 9.

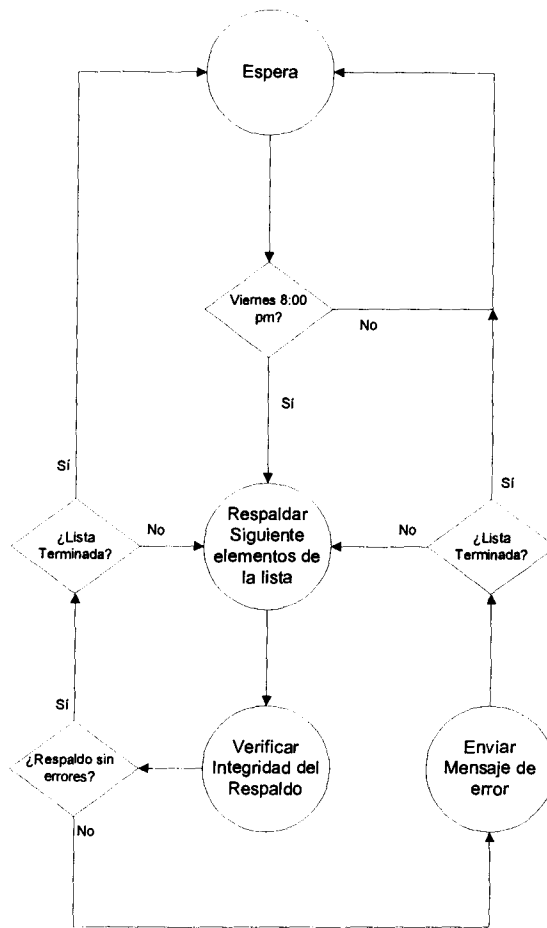


Figura 9. Proceso de Respaldo de configuraciones

Para activar el respaldo periódico se utiliza el *crontab* de Solaris. Basta con agregar la línea que describa el proceso en el *crontab*. Se desea ejecutar el respaldo todos los viernes a las 8 de la noche por lo que se debe agregar las líneas “0 20 * * 5 /usr/HP_Ag/respaldo_alcatel” y “0 20 * * 5 /usr/HP_Ag/respaldo_nortel”. Los primeros 5 caracteres indican el momento en que la aplicación *cron*d ejecuta el archivo */usr/HP_Ag/respaldo*, esto es a las 20:00 de todos los días viernes (el día de la semana 5 comenzando con Domingo=0). Mientras no se llegue a este momento el *cron*d del sistema operativo va a tener al programa de ejecución de los respaldos en modo de espera.

Una vez que se sale del estado de espera se recorre la totalidad de los equipos de red para realizar el respaldo.

El respaldo se hace de forma diferente en los equipos Nortel y en los equipos Alcatel. Los equipos Nortel pueden realizar transferencias de archivos por medio del protocolo TFTP que no necesita clave de acceso. Por no necesitar clave para el servicio tftp solamente se debe activar en el momento que se va a hacer el respaldo y desactivar una vez que se ha realizado el respaldo. La activación y desactivación del servicio de tftp se hace por medio de SNMP utilizando el *snmpset* del NNM. En Nortel basta con respaldar un archivo (el archivo “config”) que guarda la configuración ya que todas las versiones de sistema operativo instaladas en los equipos del banco entienden el archivo de configuración. En el caso de los equipos Alcatel es conveniente respaldar todos los archivos almacenados en la memoria “flash”, esto porque las configuraciones son dependientes del sistema operativo y porque la configuración no se guarda en un único archivo. Por tener que respaldar más información es conveniente que el respaldo de los equipos Alcatel se realice por medio del protocolo ftp.

Una vez que se ha hecho el respaldo se realiza una comprobación para asegurarse que la transferencia de archivos no tuvo ningún error. La comprobación se realiza por medio de una suma de comprobación (checksum). Si la suma de los bytes en la memoria “flash” del dispositivo de red no es igual a la suma de bytes en el disco duro del servidor se manda un mensaje de error al correo del operador encargado de revisar esta tarea para poder repetir el proceso.

La razón de que envíe el mensaje de error al correo y no insista en otros intentos de generar el respaldo nuevamente es porque se puede tratar de un nodo caído que puede generar un lazo que interrumpa el proceso de respaldo.

Una vez realizado el proceso los archivos quedan guardados en el servidor ftp de forma ordenada. Con la estructura mostrada en la figura 10.

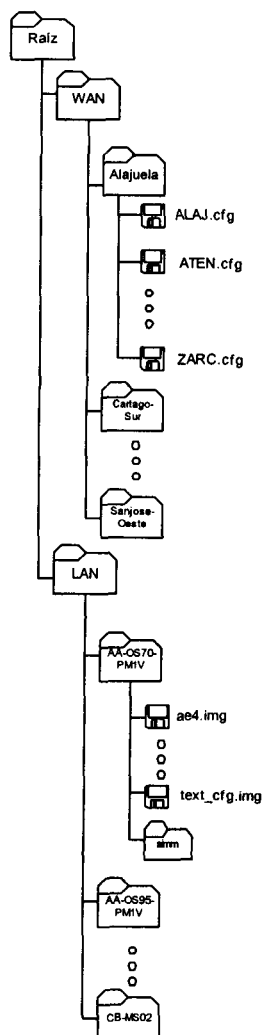


Figura 10. Estructura de los respaldos en el servidor ftp

El conjunto de scripts que realizan la tarea de respaldos completa se muestra en el apéndice A.

4.2.2 Generación de archivos para CLI

Los conmutadores Alcatel son configurables por medio de la línea de comandos conocida como CLI por las iniciales en inglés de “Command Line Interface”. Otra forma de respaldar la

configuración sería entonces generar archivos de texto con un formato válido para la interfaz de línea de comando.

La importancia de generar este tipo de archivo es que permite una configuración más rápida de un nuevo conmutador en caso de que existan problemas y también es útil cuando se quiere hacer una configuración muy similar a algún nuevo conmutador ya que bastaría con hacer pequeños cambios en el archivo de texto.

Para realizar el archivo CLI es necesario leer la configuración por medio de *snmpwalk* y procesarla. El procesamiento se hace con un script que utiliza AWK (*genera_cli*). Este script se puede ver en el apéndice A. Un ejemplo de archivo resultante se puede ver a continuación:

```
system description "Switch No.1"
system admin-contact "Administracion del Backbone UMMR"
system location "Piso 12 292908 90163010-A-01480051"
system name BM-OS4-P12
group 1 no router ip
group 2
group 2 enrutador ip 20.1.1.122 255.255.255.0
group 2 description "back2"
group 3
group 3 enrutador ip 20.1.128.1 255.255.255.0
group 3 description "Red Piso No.12"
group 2 interface 2/1
group 3 interface 2/2
group 3 interface 3/1
group 3 interface 4/3
group 3 interface 4/5
group 3 interface 5/2
group 3 interface 5/3
group 3 interface 5/4
interface ethernet 2/1 speed 100 duplex full
interface ethernet 2/2 speed 100 duplex full
interface ethernet 4/3 speed 100 duplex full
interface ethernet 5/8 speed 100 duplex full
bridge 2 priority 24576
bridge 3 priority 28672
snmp community read-only Dedalo
snmp community read-write Dedalo
snmp community trap Dedalo
snmp trap unicast
```

```
snmp station 20.21.24.151 ffffffff3:fffffe7f ffffffff:fffffff on 162 on
special
password admin switch A.291sdfj,qwiQ
password diag switch D,1kdlk.QWIq
```

4.2.3 Nodo Caído

El NNM, por medio del proceso *netmon* envía periódicamente una solicitud de eco de ICMP a todos los objetos IP dentro de la base de datos de objetos (Object database), si no hay respuesta a la solicitud de eco se considera que la interfase está caída. En caso de que ninguna de las interfaces IP correspondientes a un equipo de Red no responda el pmd genera una alerta de “Node Down” al *ovalarm*.

Esta alerta generada es perfectamente comprensible para el técnico de la red sin embargo presenta tres desventajas importantes. La primera de ella es que la alarma se genera con la pérdida de un paquete de ICMP. Esta pérdida no necesariamente se debe a una caída de la interfaz IP sino que puede deberse a congestión en la red. La segunda de ellas obedece a una razón administrativa: el Banco Nacional de C.R ha sido dividido por zonas regionales y cada técnico de monitoreo tiene asignada una zona regional y no es conveniente que se distraiga con alarmas de una zona que no es la propia. La tercera es que varios mensajes de alarma seguidos se pueden deber a una misma causa y por lo tanto no es necesario más que una indicación del evento.

Para corregir estas tres desventajas se realizó el script “*/usr/bin/HP_Ag/Node_Down*”. Este primero realiza 20 solicitudes de eco por medio de ICMP en vez de una y si se pierde más del 20 % de estas solicitudes entonces sí activa la alarma. Para conocer la zona regional a la que pertenece el dispositivo de red se consulta primero la base de datos de topología del NNM por medio del módulo *ovtopodump*. Esta relaciona el identificador de objeto con la IP. Una vez que se tiene el identificador de objeto que presenta la alarma se consulta la base de datos de objetos con *ovobjprint* para determinar a cual zona regional pertenece el elemento de red. Con este

conocimiento el mensaje se envía a la consola de alarmas en la categoría de la zona regional respectiva y con la severidad de “Critical” por medio del módulo *ovevent*. Para evitar que el mensaje se repita varias veces se utiliza un archivo auxiliar (*/tmp/caidos*). Si en la misma hora se tienen un evento de Node Down, se almacena la fuente del evento en el archivo mencionado y no se da ningún otro mensaje de alarma si el evento se vuelve a presentar en la misma hora.

El diagrama de procesos para el Node Down se observa en la figura 11.

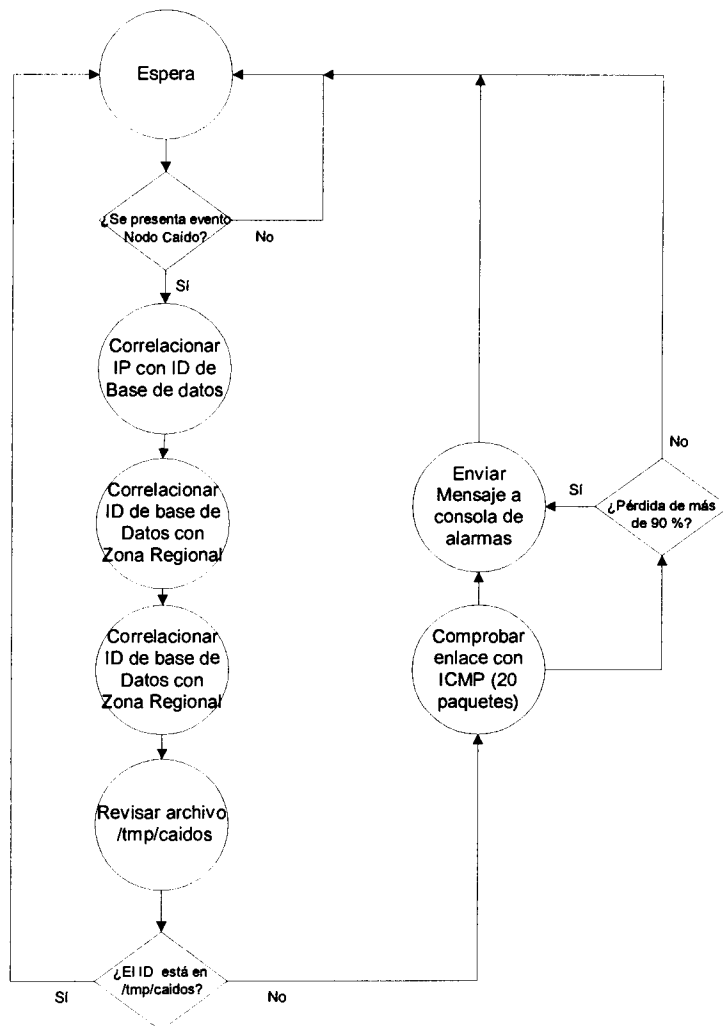
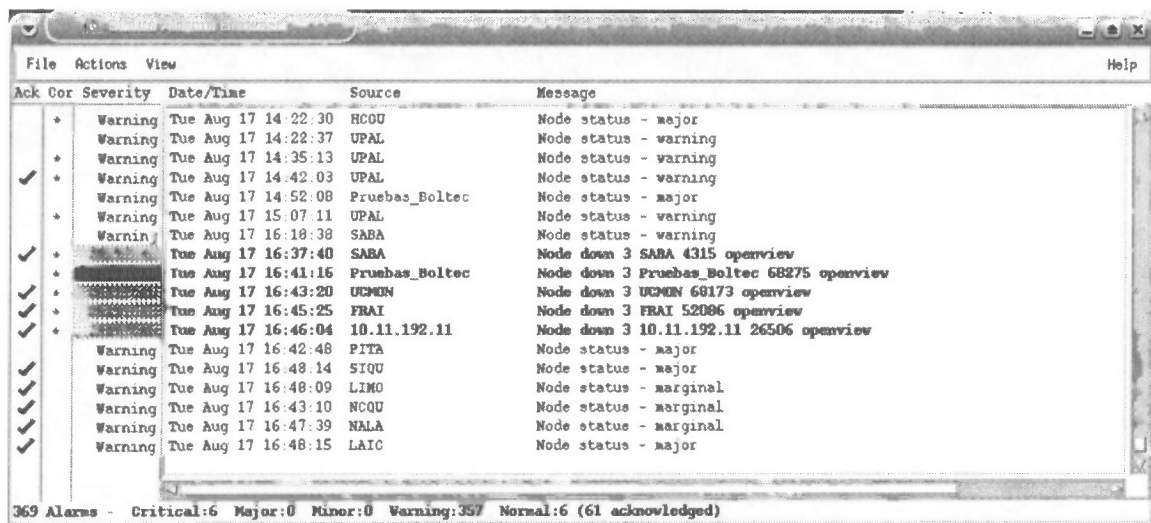


Figura 11. Diagrama de procesos para la acción Node Down

Para que la acción se ejecute cuando se da un evento Node Down se debe modificar en el archivo *trapd.conf* la acción y la categoría correspondiente al evento *OV_Node_Down*. La categoría se debe especificar como *LOGONLY* y se debe especificar que ejecute la acción *"/usr/bin/HP_Ag/Node_Down"* con el parámetro *\$2*. El parámetro *\$2* es el nombre del nodo que generó el evento. El formato que deben llevar en el archivo *trapd.conf* se muestra en el extracto siguiente:

```
EVENT OV_Node_Down .1.3.6.1.4.1.11.2.17.1.0.58916865 "LOGONLY" Critical
FORMAT Node down $1 $2 $3 $4 $5 $6 $7 $A
EXEC /usr/bin/HP_Ag/Node_Down $2
```

Las diferencias que se dan al utilizar este programa se pueden ver en la consola de alarmas de la figura 12. En la figura 12 (a) se observa que los mensajes se dan sin importar de cual zona se genera la alarma, además en algunos casos no se obtiene un nombre sino una dirección IP mientras que en la parte (b) se observa que para el grupo de Alajuela solamente se tienen Alarmas de equipos de la zona de Alajuela (CoopeVictoria, Pital, San Miguel de Sarapiquí, Caja Auxiliar de Alajuela, etc.), el mensaje está en español y el nombre del dispositivo que generó la Alarma está completamente claro.



Ack	Cor	Severity	Date/Time	Source	Message
		Warning	Tue Aug 17 14:22:30	HCGU	Node status - major
		Warning	Tue Aug 17 14:22:37	UPAL	Node status - warning
		Warning	Tue Aug 17 14:35:13	UPAL	Node status - warning
✓		Warning	Tue Aug 17 14:42:03	UPAL	Node status - warning
		Warning	Tue Aug 17 14:52:08	Pruebas_Boltec	Node status - major
		Warning	Tue Aug 17 15:07:11	UPAL	Node status - warning
		Warning	Tue Aug 17 16:18:38	SABA	Node status - warning
✓		Warning	Tue Aug 17 16:37:40	SABA	Node down 3 SABA 4315 openview
		Warning	Tue Aug 17 16:41:16	Pruebas_Boltec	Node down 3 Pruebas_Boltec 68275 openview
✓		Warning	Tue Aug 17 16:43:20	UCMEN	Node down 3 UCMEN 68173 openview
✓		Warning	Tue Aug 17 16:45:25	FRAI	Node down 3 FRAI 52086 openview
✓		Warning	Tue Aug 17 16:46:04	10.11.192.11	Node down 3 10.11.192.11 26506 openview
✓		Warning	Tue Aug 17 16:42:48	PITA	Node status - major
✓		Warning	Tue Aug 17 16:48:14	SIQU	Node status - major
✓		Warning	Tue Aug 17 16:48:09	LIMO	Node status - marginal
✓		Warning	Tue Aug 17 16:43:10	NCQU	Node status - marginal
✓		Warning	Tue Aug 17 16:47:39	NALA	Node status - marginal
✓		Warning	Tue Aug 17 16:48:15	LAIC	Node status - major

369 Alarms - Critical:6 Major:0 Minor:0 Warning:357 Normal:6 (61 acknowledged)

(a)

Ack	Cor	Severity	Date/Time	Source	Message
			Thu Aug 12 14:48:56	CVIC	Caído Totalmente el equipo en ATM CoopVictoria
		Warning	Thu Aug 12 15:53:51	CGMO	Caído enlace S12_MCOU_PITAL_PPPICE en Nodo Ciudad Quesada
		Warning	Sat Aug 14 11:35:48	CTAC	Caído enlace El_CTAC en ATM Coopetacares
		Warning	Sun Aug 15 13:16:00	CTAC	Caído enlace El_CTAC en ATM Coopetacares
		Warning	Sun Aug 15 13:22:58	CTAC	Caído enlace El_CTAC en ATM Coopetacares
		Warning	Sun Aug 15 14:20:11	CTAC	Caído enlace El_CTAC en ATM Coopetacares
			Sun Aug 15 15:52:34	PITA	Caído Totalmente el equipo en Pital
			Sun Aug 15 18:11:26	PITA	Caído Totalmente el equipo en Pital
			Mon Aug 16 07:30:55	SMSA	Caído Totalmente el equipo en San Miguel Sarapiquí
			Mon Aug 16 10:26:29	PITA2	Caído Totalmente el equipo en Pital 2
			Mon Aug 16 10:27:02	PITA	Caído Totalmente el equipo en Pital
		Warning	Mon Aug 16 10:55:07	PITAC	Caído enlace S12_PITA_SMSA_SSPPPP en Pital 2
			Mon Aug 16 11:04:23	PITA	Caído Totalmente el equipo en Pital
		Warning	Mon Aug 16 12:00:14	PITA2	Caído enlace S12_PITA_SMSA_SSPPPP en Pital 2
			Mon Aug 16 14:35:38	CAUX	Caído Totalmente el equipo en CAJA AUX. ALAJUELA CALABAZO.
		Warning	Mon Aug 16 16:00:02	PITA2	Caído enlace S12_PITA_SMSA_SSPPPP en Pital 2
			Mon Aug 16 18:50:58	CTAC	Caído Totalmente el equipo en ATM Coopetacares
		Warning	Mon Aug 16 20:00:24	PITA2	Caído enlace S12_PITA_SMSA_SSPPPP en Pital 2
		Warning	Tue Aug 17 04:00:05	PITA2	Caído enlace S12_PITA_SMSA_SSPPPP en Pital 2
		Warning	Tue Aug 17 08:00:23	PITA2	Caído enlace S12_PITA_SMSA_SSPPPP en Pital 2

20 Alarms - Critical:9 Major:0 Minor:0 Warning:11 Normal:0

(b)

Figura 12. Consola de alarmas antes y después de utilizar el Node Down

El script Node_Down se muestra completamente en el apéndice A.

4.2.4 Enlace Caído

El evento de enlace caído es disparado por los *traps* de los equipos de red cuando cualquiera de sus interfaces pasa al estado de no operación. El *outrapd* escucha el puerto 162 de UDP y cuando recibe un paquete con el indentificador de enlace caído (1.3.6.1.6.3.1.1.5) ejecuta la acción. De hecho, este evento es el que más se presenta en la red.

La lógica que se aplica con este evento es similar a la del Nodo Caído. Se discrimina de cuál zona regional proviene, el mensaje es en español, se da una correlación entre el nombre de la interfaz que presenta problemas y el *if-index* (índice de interfaz) especificado en el *trap* recibido.

Se presenta una particularidad específica del enlace caído. Cuando un enlace con el protocolo PPP sale de servicio los enrutadores intentan negociar con IPCP que es un protocolo

con el que el PPP conoce la dirección IP que se encuentra en el otro extremo. Se envían 3 paquetes de IPCP, el puerto pasa el estado de INIT, después al de WAIT y después otra vez al estado UP. Este ciclo se repite cada 30 segundos. Cada vez que el puerto pasa del estado UP a cualquier otro el agente de SNMP envía el *trap* de Link Down. Este comportamiento hace que a la consola de alarmas lleguen muchas alarmas repetidas que son síntomas de un solo evento. Lo correcto es que el programa de procesamiento se de cuenta si el *trap* que llegó determine si se encuentra ante esta situación y, si es así, que solamente indique una alarma.

El NNM tiene la facilidad de poner interfaces en modo «Unmanaged» o no Manejado. Este modo se utiliza cuando por ejemplo se está realizando algún trabajo de mantenimiento o se conoce de antemano que un puerto no va a estar disponible. Cuando alguno de los elementos de la base de datos se encuentra en este modo, el *netmon* no realiza la solicitud de eco de ICMP y en el nodo aparece un color claro como se muestra en el ícono con la etiqueta “S12_AREN_TILA_ICEPPP” en la figura 13. El *ovalarm* y *ovactiond* no tienen interacción directa con el *netmon* por lo que las alarmas que se generen relacionadas con este puerto siempre se mostrarían.

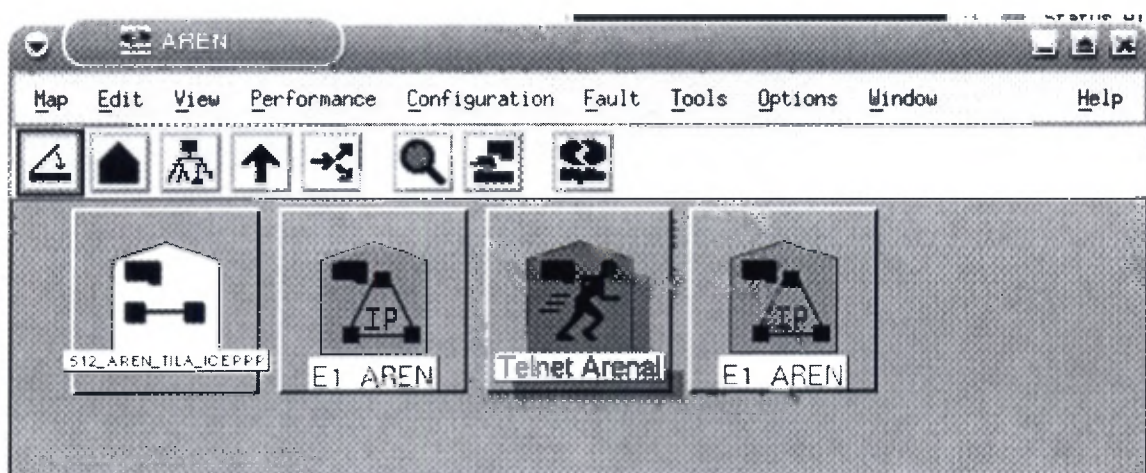


Figura 13. Vista de Objeto no Manejado en NNM

El archivo `/usr/bin/HP_Ag/LinkDown.pl` es el programa de procesamiento para los *traps* de Link Down. El proceso que realiza este programa se observa en el diagrama de la figura 14.

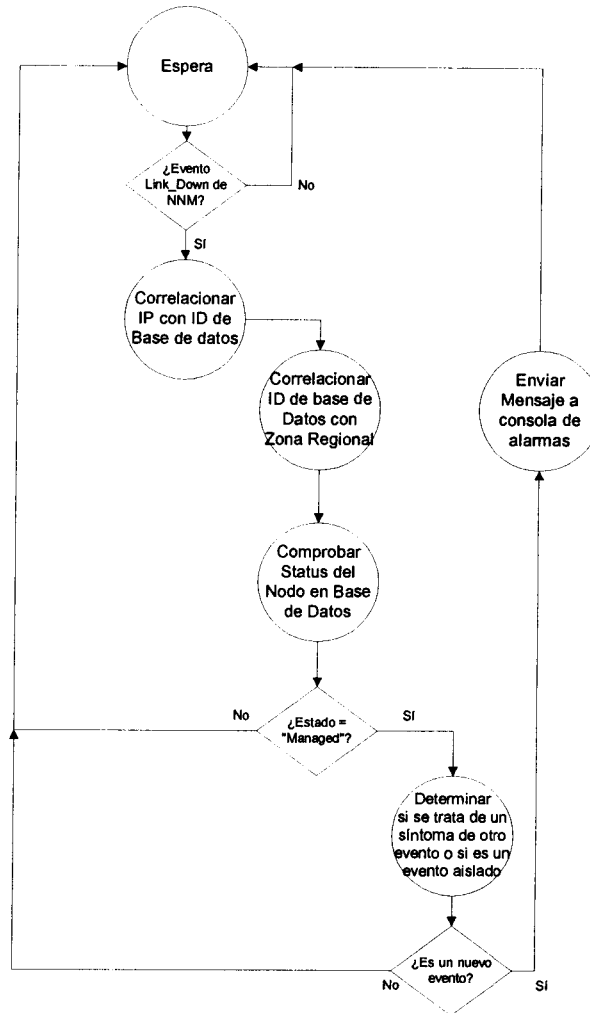


Figura 14. Diagrama de Procesos ante el evento Link Down

Para determinar si el *trap* generado es síntoma de un evento más grande se utilizan dos archivos auxiliares. El primero de ellos se utiliza para dejar registrada la hora en la que se presentó el evento. El nombre del primer archivo depende de la interfaz (if-index) que generó el evento y del identificador del objeto según la base de datos del NNM, el nombre se da de la

siguiente forma: `<if-index>_<Identificador del objeto>_LINK_D_hora_enable`. Cuando se presenta un nuevo evento, se compara la hora en la que llegó el último con la del inicial, si la diferencia de tiempos es alrededor de 30 segundos se asume que se trata de la negociación de IPCP, se guarda el identificador del objeto en el archivo `LINK_D_reporte_enable` y no se mandan más mensajes que provengan de la misma interfaz y del mismo elemento. Este archivo es eliminado cada 4 horas por el *crontab*.

Después de hacer el cambio en el archivo `trapd.conf` para que se ejecute el archivo `/usr/bin/HP_Ag/LinkDown.pl` se puede observar la diferencia:

Ack	Cor	Severity	Date/Time	Source	Message
		Major	Sun Sep 12 14:01:18	UPAL3	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:01:24	10.11.159.13	Agent Interface Down (linkDown Trap) on interface 3
		Major	Sun Sep 12 14:01:25	10.47.241.9	Agent Interface Down (linkDown Trap) on interface 7
		Major	Sun Sep 12 14:01:32	PVCC2	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:01:44	AZOT	Agent Interface Down (linkDown Trap) on interface 3
		Major	Sun Sep 12 14:01:50	UPAL3	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:01:55	10.11.159.13	Agent Interface Down (linkDown Trap) on interface 3
		Major	Sun Sep 12 14:01:57	10.47.241.9	Agent Interface Down (linkDown Trap) on interface 7
		Major	Sun Sep 12 14:02:03	PVCC2	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:02:16	AZOT	Agent Interface Down (linkDown Trap) on interface 3
		Major	Sun Sep 12 14:02:21	UPAL3	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:02:27	10.11.159.13	Agent Interface Down (linkDown Trap) on interface 3
		Major	Sun Sep 12 14:02:28	10.47.241.9	Agent Interface Down (linkDown Trap) on interface 7
		Major	Sun Sep 12 14:02:35	PVCC2	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:02:47	AZOT	Agent Interface Down (linkDown Trap) on interface 3
		Major	Sun Sep 12 14:02:53	UPAL3	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:02:58	10.11.159.13	Agent Interface Down (linkDown Trap) on interface 3
		Major	Sun Sep 12 14:03:00	10.47.241.9	Agent Interface Down (linkDown Trap) on interface 7
		Major	Sun Sep 12 14:03:07	PVCC2	Agent Interface Down (linkDown Trap) on interface 2
		Major	Sun Sep 12 14:03:19	AZOT	Agent Interface Down (linkDown Trap) on interface 3

66 Alarms - Critical:0 Major:66 Minor:0 Warning:0 Normal:0

(a)

Ack	Cor	Severity	Date/Time	Source	Message
		Warning	Sun Sep 12 14:30:51	AZOT	Caído enlace S13_AZOT_PVCC_S en Azotea

1 Alarms - Critical:0 Major:0 Minor:0 Warning:1 Normal:0

(b)

Figura 15. Comparación del tratamiento de trap Link Down antes y después de procesar

En la figura 15 (a) se puede observar primero que cada 30 segundos hay una alarma de varios equipos, para mayor claridad se muestra con una flecha roja el elemento AZOT que está instalado en la azotea del Banco en Oficina Principal. Este elemento presenta la alarma a las 14:02:16, el siguiente a las 14:02:47 y el último que se muestra a las 14:03:19. Lo que está sucediendo es la negociación del IPCP que genera una alarma aproximadamente cada 30 segundos más alarmas de las necesarias. En realidad basta una sola alarma, las demás no aportan ninguna información nueva. Se puede también notar que para indicar cual interfaz es la que presenta la caída, lo hace por medio del *if-index* señalado con una flecha negra; para determinar cual es la interfaz con problemas, el técnico de monitoreo debería conocer más detalles de la configuración para dar una interpretación más completa a la alarma.

En la figura 15 (b) se observa el resultado de la alarma después de realizar el procesamiento. El programa realizado en PERL toma la información del *if-index* y revisa a cual de los circuitos virtuales del equipo corresponde. El técnico de monitoreo puede entonces observar que la alarma corresponde al sincrónico 1.3 del enrutador de la azotea el cual hace la conexión con el

centro comercial de Pavas. Además el programa `/usr/bin/HP_Ag/LinkDown.pl` revisa si las alarmas se están presentando con un período aproximado de 30 segundos y, si este es el caso, solamente presenta una alarma con lo que la consola de alarmas queda con más claridad para su lectura.

4.2.5 Servicios ATM Colgados

Cada conmutador de distribución de la red principal tiene dos servicios ATM con los conmutadores del núcleo de la red del edificio principal. Algunas veces la negociación de los servicios ATM fallan y no pueden conectar con el conmutador remoto, si uno de los servicios falla el conmutador de distribución se queda sin respaldo y es cuestión de tiempo para que se presente una avería.

El síntoma que se presenta cuando no se logra conectar el circuito virtual de ATM, es la generación de un *trap* del tipo `portLinkUpEvent9` (.1.3.6.1.4.1.800.3.1.1.2.0.8) cada 10 segundos. Si se hace una sesión de *telnet* al conmutador y se consulta el estado de los servicios ATM se observa que el estado es “Initial”, o sea un estado de negociación como en la figura 16:

```
BR-CS-01/~/>vas
```

ATM Services

Slot	Port	Serv Num	Service Description	Service Type
3	1	2	back2	802.3 LEC
3	2	2	back2	802.3 LEC

ATM Services

Slot	Port	Serv Num	UC Typ	Oper Status	SEL Groups	Conn UPI/UCI	(Addr Index)
3	1	2	SUC	LANE Op.	02 2	0/33 0/34	0/35 0/36
3	2	2	SUC	Initial	02 2	0/37 0/39	0/40 0/654

```
FDDI Services do not exist!
```

```
Gigabit Ethernet 802.1Q Services do not exist.
Ethernet 802.1Q Services do not exist!
BR-CS-01/~/>
```

Figura 16. Observación del estado Inicial de un SVC de ATM

El procesamiento que se hace con este *trap* se hace con el archivo `/usr/bin/HP_Ag/vportEma.pl` y es similar al que se realiza con el de “Enlace Caído”. De hecho si se llegan dos *traps* de este tipo pero con mucho tiempo de diferencia se indica en la consola de alarmas como una caída en una interfaz. El diagrama de procesos para esta alerta se muestra en la figura 17.

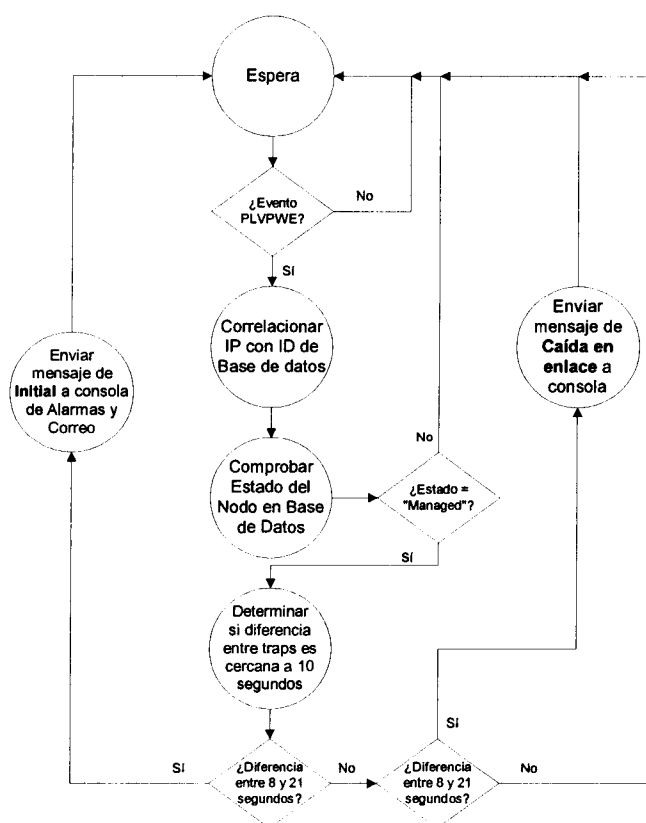


Figura 17. Diagrama de procesos para el tratamiento del evento ATM Inicial

Una variación importante es que este tipo de alarmas envía un correo electrónico para que el técnico de averías lo tenga aún más presente, ya que una situación de “ATM Inicial” puede ser grave en la red. El remitente del mensaje de correo es Gestion HP-OpenView. El mensaje de correo resultante se puede observar en la figura 18.

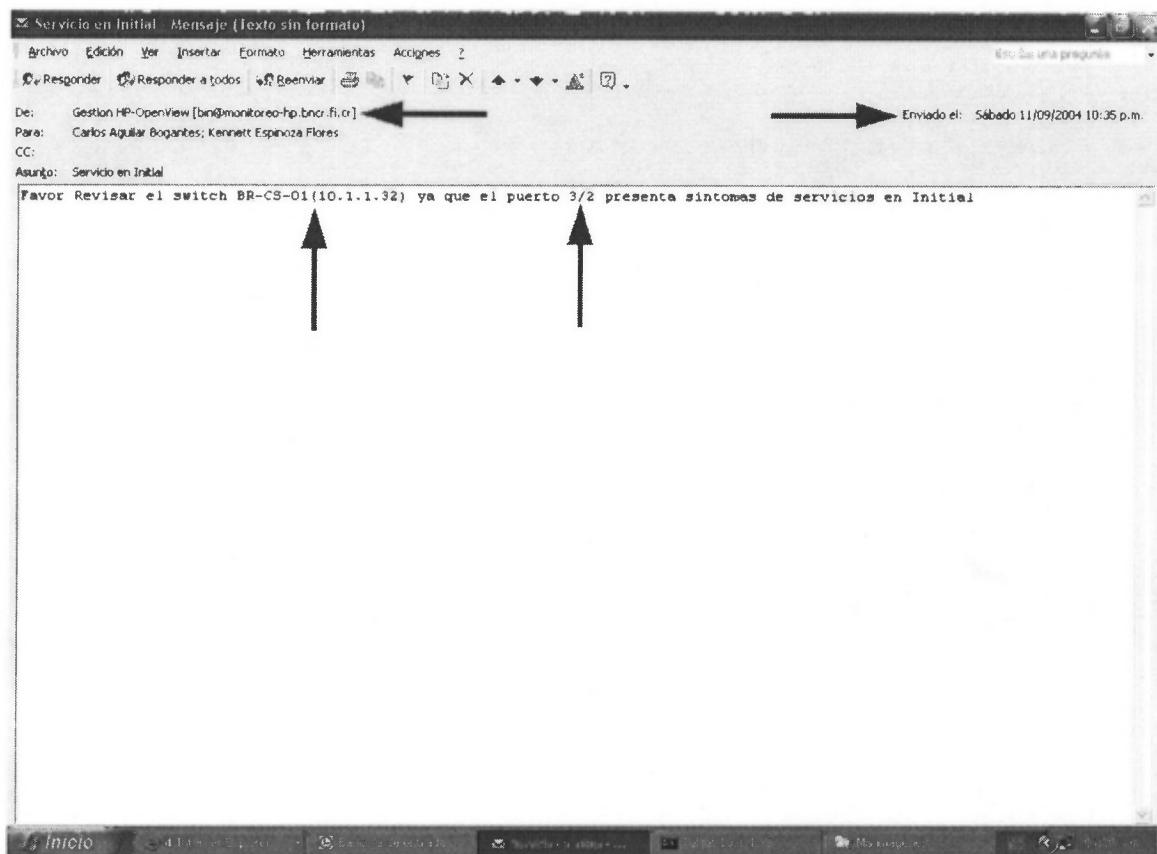


Figura 18. Correo que recibe el usuario al tratar el evento de “ATM Inicial”

El mensaje indica claramente la fecha en la que se dio el evento, el conmutador ATM en el que se presentó (Nombre e IP) y cual es la interfaz afectada. Se envía un correo al día hasta que esta situación sea corregida o se ponga como Unmanaged la interfaz afectada.

4.2.6 *Traps* de Salud o «Health»

Los conmutadores Alcatel utilizados en la red del Banco Nacional son capaces de enviar un mensaje de alerta si se excede el porcentaje de utilización del *backplane*, de la CAM

(Memoria de manejo de direcciones físicas), del CPU, de la Memoria y de la transmisión/recepción.

Es normal que se alcancen valores altos de utilización de estos elementos durante cierto tiempo, sin embargo, si estos valores permanecen altos por mucho tiempo el conmutador no va a trabajar adecuadamente y va a comenzar a descartar paquetes.

La idea con el procesamiento de este tipo de mensajes es que solamente se muestren, si durante un minuto, alguna de las utilizaciones críticas del conmutador se han mantenido por encima de un cierto valor.

El procesamiento de este tipo de mensajes se hace por medio del programa `/usr/bin/HP_Ag/Health.pl` que a su vez invoca al programa `/usr/bin/HP_Ag/Ejec_Health.pl`. Este tiene la particularidad importante que se puede introducir como un parámetro (es decir sin modificar el código) el porcentaje que se desea que sea tomado para determinar si se envía la alarma o no. Este se incluye desde la ventana del Event Configuration del NNM como se muestra en la figura 19. Para el caso de esta figura, si alguno de los elementos de la salud mencionados están por encima del 70 % en un minuto se da la alarma; si el valor que presenta el conmutador es menor que este, no se da ninguna alarma.

Event Name: healthThresholdDevice

Event Type: Enterprise Specific

Event Object Identifier: .1.3.6.1.4.1.800.3.1.1.0.39

Event Description:

Long Descr.:
 "A healthThresholdDevice trap occurs when at least one device-level threshold crossing is detected."
 Variables:
 1: healthThreshDevTrapData
 Syntax="Octet String (SIZE(0..21))"
 Descr="Contents of device-level rising/falling threshold trap.
 The format is as follows:
 Octet 1: length including this octet (L)
 Octet 2: number of threshold type/change pairs

Event Sources (all sources if list is empty)

Source: [Empty]

Category: Log only

Forward Event:

Severity: Warning

Event Log Message:
 healthThresholdDevice .1.3.6.1.4.1.800.3.1.1.0.39 \$1 \$A

Pop-up Notification (Optional): [Empty]

Command for Automatic Action (Optional):
 echo \$A 70 \$1 | /usr/bin/HP_Ag/Health.pl

Buttons: OK, Reset, Cancel, Help

Figura 19. Vista del Event Configurator del NNM al incluir la acción `/usr/bin/HP_Ag/Health.pl`

El programa `/usr/bin/HP_Ag/Health.pl` sólo realiza un procesamiento inicial, revisa el nombre de quien generó la alarma en la base de datos del NNM y sale si el elemento en la base de datos está registrado como "Unmanaged". Si el elemento no está como "Unmanaged" se invoca el archivo `/usr/bin/HP_Ag/Ejec_Health.pl` para que corra como un proceso independiente. Este archivo invocado espera un minuto antes de comenzar a trabajar para dar tiempo a que el conmutador se establezca en caso de que se trate de una operación normal.

Después de esperar este minuto revisa en el conmutador por medio de *snmpwalk* los parámetros que le envió el archivo “padre” y, si el conmutador se mantuvo por encima del valor introducido como parámetro, da la alarma a la consola de alarma del NNM en la categoría Backbone.

Este programa también llena un archivo informando acerca de todos los equipos que presentaron la superación de un umbral. El objetivo de guardar este archivo es que sirva de insumo para la generación de reportes y poder llevar un control de la reincidencia de eventos. Si, por ejemplo, un conmutador presenta muchas alarmas de superación del umbral, puede ser una indicación de que no tiene las características necesarias para la labor que está cumpliendo y, si la red continúa creciendo, sería recomendable cambiarlo por uno con más memoria o un CPU superior.

Las figuras 20 y 21 muestran los procesos en Health.pl y Ejec_Health.pl.

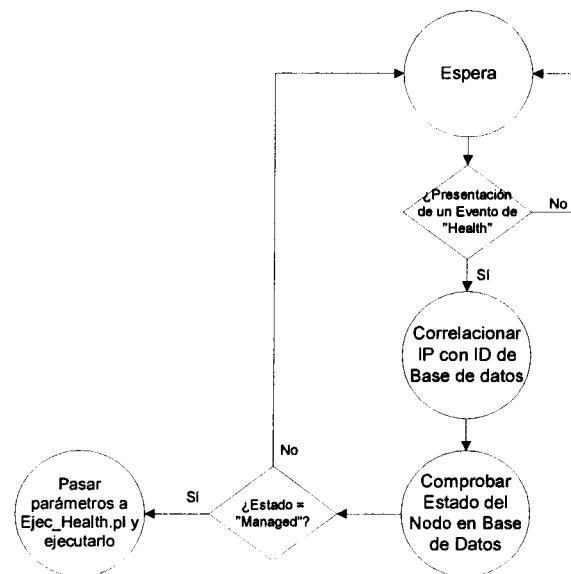


Figura 20. Diagrama de Procesos de /usr/bin/HP_Ag/Health.pl

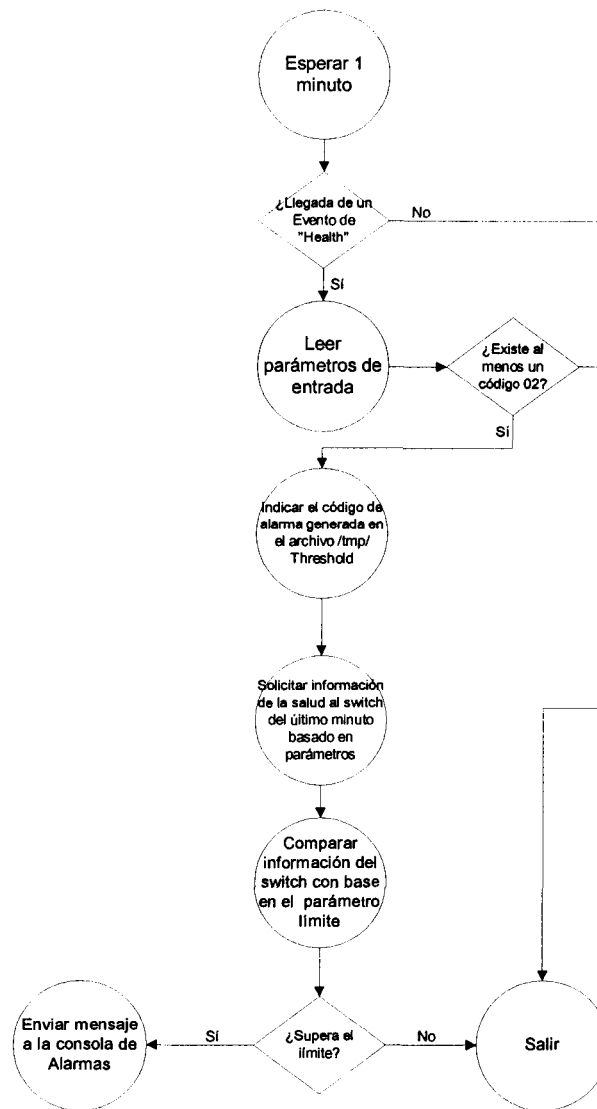


Figura 21. Diagrama de procesos de /usr/bin/HP_Ag/Ejec_Health.pl

La presentación en la consola de alarmas de este evento es un cambio realmente importante porque el mensaje que se genera por defecto es un código hexadecimal que el usuario de la plataforma de monitoreo tiene que interpretar como se puede apreciar en la figura 22 (a). Aquí el mensaje con el código 0x04 01 08 02 se debe interpretar como una superación del umbral del CPU 1 y el código 0x04 01 08 02 significa que se volvió a pasar por debajo del umbral.

File Actions View Help

Ack	Cor	Severity	Date/Time	Source	Message
		Warning	Wed Sep 15 11:47:53	BR-CS-01	Device-level threshold crossing is detected (Data 0x04 01 08 02)
		Warning	Wed Sep 15 11:47:58	BR-CS-01	Device-level threshold crossing is detected (Data 0x04 01 08 01)

2 of 26 Alarms - Critical:0 Major:0 Minor:0 Warning:2 Normal:0 Filters: Source Time

(a)

File Actions View Help

Ack	Cor	Severity	Date/Time	Source	Message
		Warning	Wed Sep 15 11:41:03	BR-CS-01	Excedido Threshold de la Memoria. Ultimo Minuto=66
		Warning	Wed Sep 15 11:47:23	BR-CS-01	Excedido Threshold del CPU 1 Ultimo Minuto=25

2 of 26 Alarms - Critical:0 Major:0 Minor:0 Warning:2 Normal:0 Filters: Source Time

(b)

Figura 22. Mensajes de Salud en la consola de Alarmas

En la figura 22 (b) se observa que el mensaje aparece en español con la excepción de la palabra «Threshold» (Umbral), que se mantiene en inglés por ser más clara así para los técnicos, e indica quién se afectó (Memoria, CPU 1) y en cuanto ha estado el porcentaje de utilización el

último minuto. Para el caso de la figura 22 (b) el límite de comparación que se incluye por parámetro se colocó en 30%.

4.2.7 *Traps de Autenticación*

El *trap* de Autenticación o «Authentication» se genera cuando un gestor de SNMP intenta acceder un agente que no tiene permitido. Este mensaje puede ser normal o puede deberse a un intento de ataque de un «cracker».

El manejo de este tipo de evento es sencillo. Un sólo paquete no representa ningún peligro pero cuando se dan varios cercanos sí es necesario ponerle atención.

Para procesar este tipo de *traps* se utiliza el archivo `/usr/bin/AuthentFail.pl`. Este revisa si se presenta más de un evento del *trap* con el evento de “Authentication Failure” y si se dan 2 en un cierto tiempo envía un mensaje a la consola de alarmas y envía un correo electrónico. El proceso que realiza este evento se muestra en la figura 23.

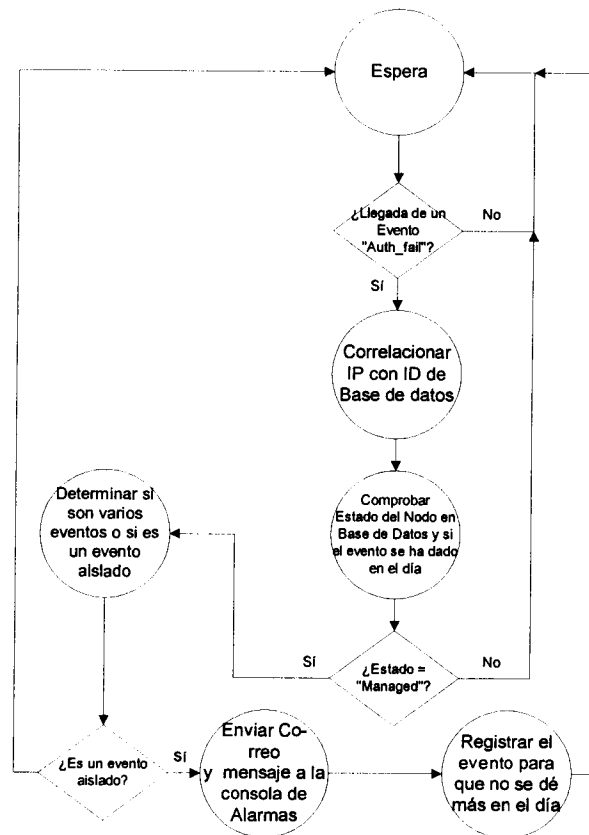


Figura 23. Diagrama de procesos para el evento de Falla de Autenticación

En la figura 24 se observa el correo que se envía al usuario de monitoreo. Con esta alerta ya se puede investigar por medio de un analizador de protocolos si es por alguien no autorizado tratando de tener acceso al equipo o si es un error en la configuración de la comunidad de alguno de los equipos de gestión autorizados.

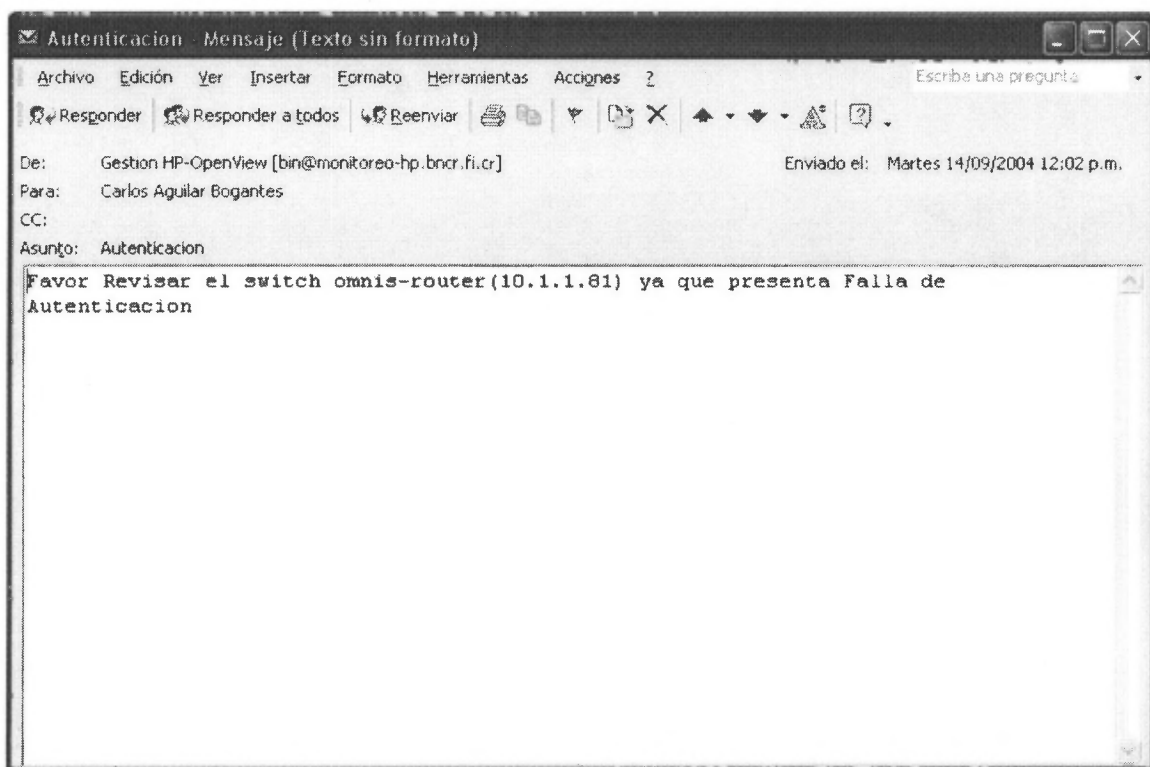


Figura 24. Correo enviado por el NNM indicando Evento de Falla de Autenticación

4.2.8 Cambio de grupo

Un cambio de grupo en uno de los puertos de un conmutador puede hacer que uno de los servidores principales salga de operación en la red si se da un error en el momento de cambiar el grupo. Por esta razón es importante notar cuando esto sucede. También es conveniente para un adecuado control de cambios.

El evento que dispara este mensaje es el *portMove* (1.3.6.1.4.1.800.3.1.1.0.16) y es procesado por el archivo */usr/bin/HP_Ag/Eliminacion_Delete*.

Cuando el conmutador hace el cambio de grupo tiene que borrarlo en el grupo anterior y crearlo en el grupo actual por lo que por cada evento se generan dos mensajes por lo que

Eliminacion_Delete elimina uno de estos mensajes y pone el mensaje en español. El diagrama de procesos se puede ver en la figura 25.

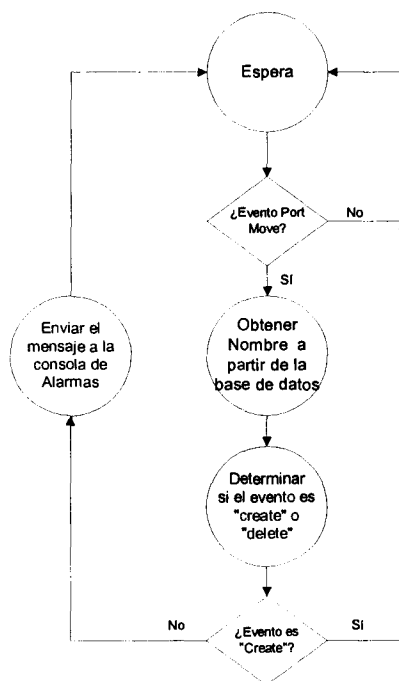


Figura 25. Diagrama de procesos para el evento de Cambio de Grupo

En la consola de alarmas se puede observar el resultado al cambiar un puerto de grupo (figura 26). Se puede ver claramente el conmutador que genera la alarma y cuál fue el puerto que cambió.

File	Actions	View			Help
Ack	Cor	Severity	Date/Time	Source	Message
		Normal	Wed Sep 15 16:55:36	CB-MS02	Puerto 4/3 cambio de grupo
		Normal	Wed Sep 15 16:56:53	BW-054-P12	Puerto 3/5 cambio de grupo

2 Alarms - Critical:0 Major:0 Minor:0 Warning:0 Normal:2

Figura 26. Vista de la consola de Alarmas cuando se cambia de grupo un puerto

4.2.9 Vital Net

El Vital Net es una herramienta de Lucent Technologies para la colección, almacenamiento y presentación de datos de equipos de red. La interrogación al equipo de red lo realiza cada minuto y obtiene datos de bytes recibidos y transmitidos, errores, uso de memoria y cpu así como también el tiempo de respuesta promedio de los circuitos IP.

El Vital Net puede ser configurado para que cuando alguno de los parámetros que colecciona se exceda de un cierto valor, se envíe una notificación al NNM con el mismo formato de un *trap* de SNMP, por lo que el NNM lo puede procesar de la misma forma que lo hace con un *trap* proveniente de un equipo de comunicaciones que tiene un agente de SNMP corriendo.

Habilitar esta interacción es sencillo en ambos extremos. En el Vital Net se debe llenar una línea con la información de la IP del receptor de *traps* y la comunidad SNMP como se muestra en la figura 27, para luego marcar el botón *Submit*.

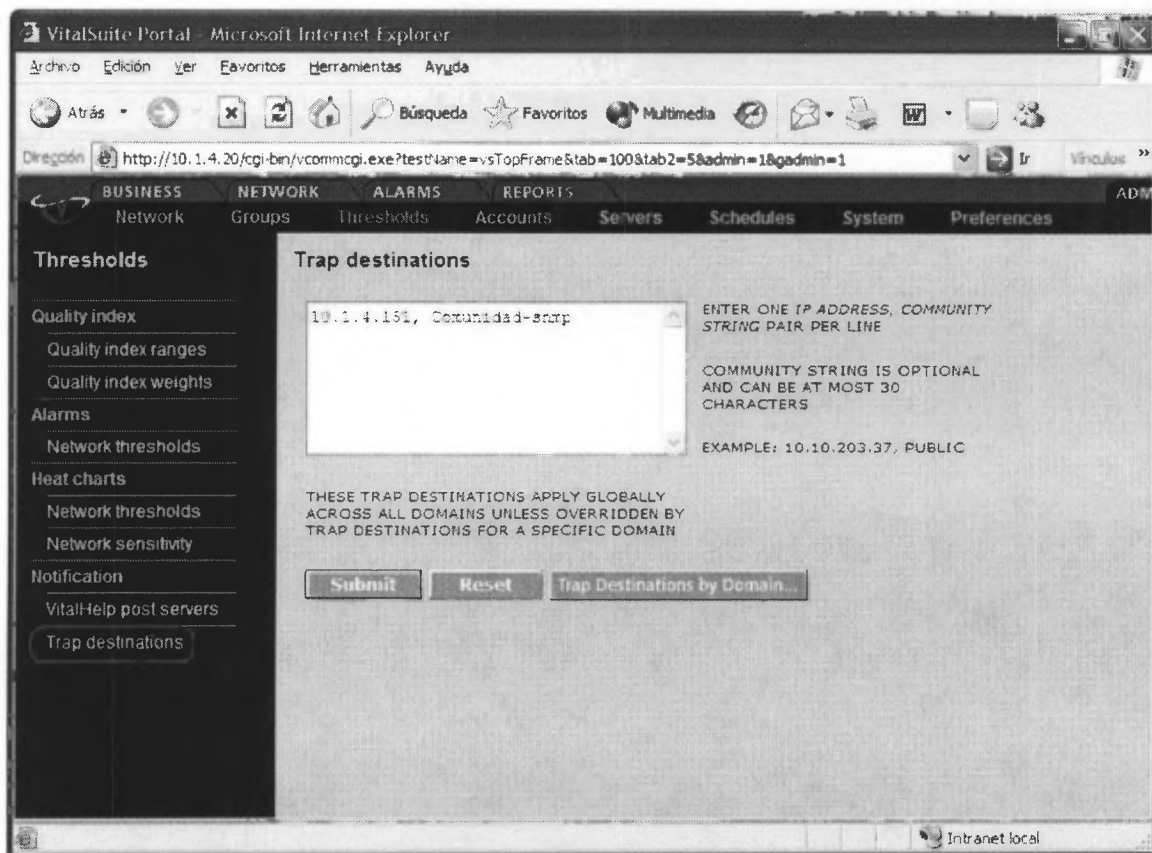


Figura 27. Configuración del destino de Traps del Vital Net

El identificador del *trap* enviado por el Vital Net se puede observar con el Sniffer que es el mismo que produce el colector del NNM o sea `OV_DataCollectThresh.1.3.6.1.4.1.11.2.17.1.0.58720263`. Con esto basta definir el formato del mensaje en el archivo `trapd.conf` o por medio del “Event Configurator”. Se muestra como se ve en el archivo `trapd.conf` (en negritas el formato del mensaje).

```
EVENT OV_DataCollectThresh .1.3.6.1.4.1.11.2.17.1.0.58720263 "Vital Net"
Major
FORMAT $5.$6 Threshold Excedido ($13 $7 * $14): $8. Valor pico $9 a las
$10 Valor Mínino $11 a las $12
SDESC
```

This event is generated by HP OpenView data collector when a sampled value exceeds a preconfigured level. (See also specific event #58720264 for rearm events). The user may also configure value exceeded events with specific trap numbers in the odd range of 1 to 9999.

EDESC

El resultado de este evento se puede ver en la figura 28. Con esta integración, el técnico de monitoreo no necesita estar ingresando continuamente al Vital Net y hacer un análisis de todos los elementos, sino que puede concentrarse en los que se observan en la consola de alarmas del NNM.

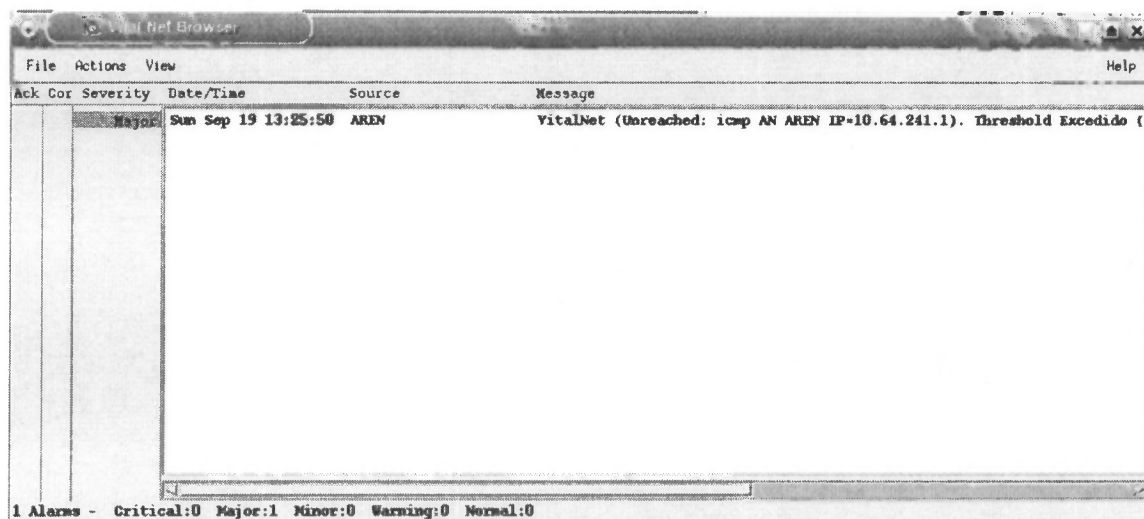


Figura 28. Vista de la consola de Alarmas con evento del Vital Net

4.2.10 Reportes

El manejo de eventos reincidentes es de gran importancia en la administración de la red. El hecho que un dispositivo de la red presente alarmas continuamente puede indicar que está cerca del punto de colapso aunque no haya llegado ahí. También continuas fallas en un enlace puede indicar algún problema físico al que debe prestársele mayor atención.

Debido a la importancia del manejo de eventos incidentes se decidió aprovechar el *crontab* y generar un programa para revisar quien ha generado más eventos de salud y quien ha presentado más caídas.

La idea es que cada viernes a las 12:00 mediodía se presente en un correo electrónico indicando quién tuvo más caídas y quién presentó más problemas de salud. El código para realizar esta tarea es bastante sencillo y se puede ver en el apéndice A.

La figura 29 muestra el diagrama de procesos para esta tarea.

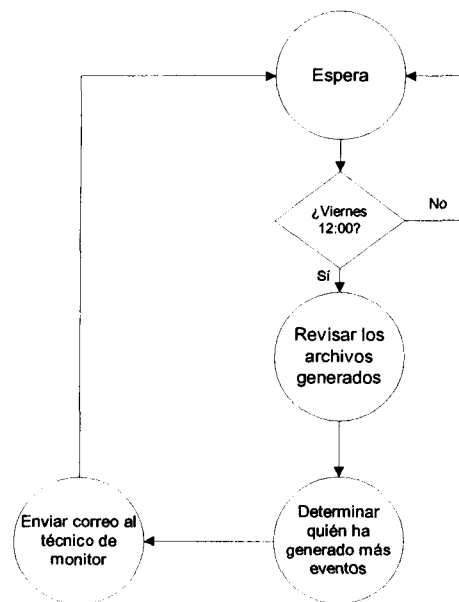


Figura 29. Diagrama de procesos para la generación de reportes de salud y caídas

El correo resultante de correr un reporte se puede observar en la figura 30. Puede notarse que hubo tres tipos de alarmas de salud durante la semana: Rx-Tx, CAM del HRE y CPU e indica quienes presentaron los mayores problemas y cuantas veces se superó el umbral. También se puede notar que quien tuvo más caídas fue Parrita con 4 durante la semana.

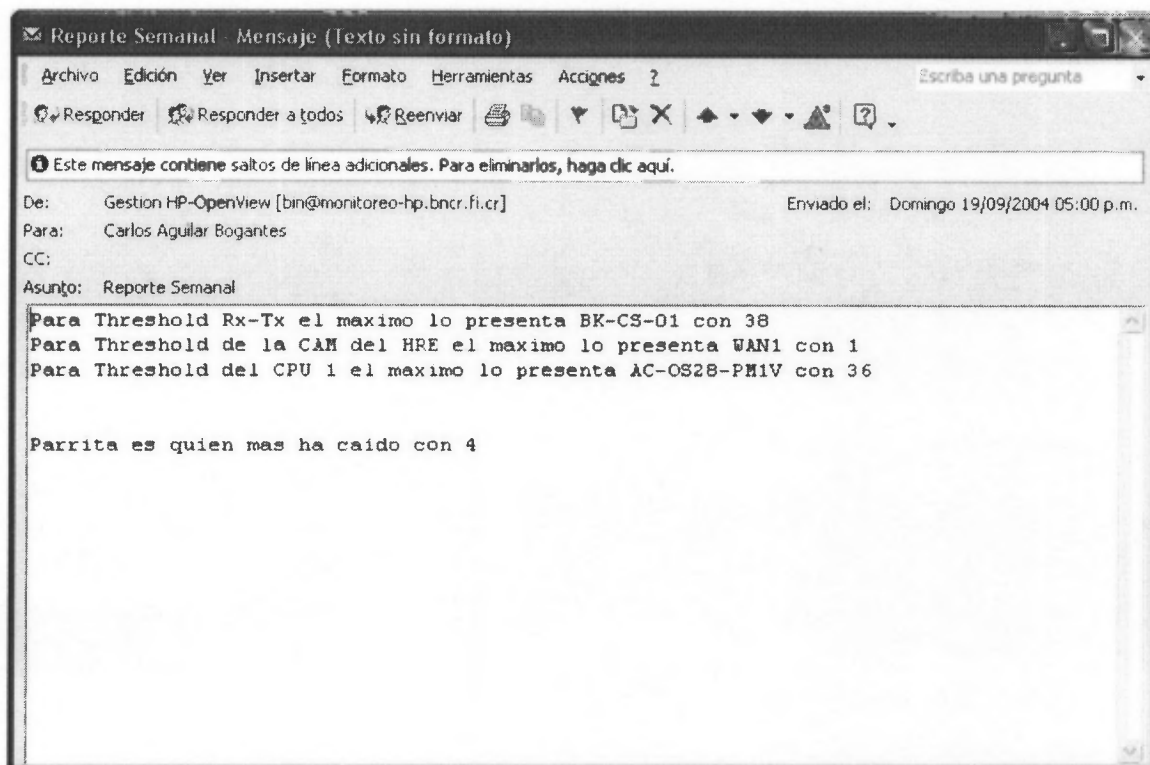


Figura 30. Correo de Reporte semanal

4.2.11 Corolario de Acciones Automáticas

La programación de acciones automáticas que se realizó elimina en una gran medida la cantidad de alarmas redundantes.

Se puede observar en la figura 31 (a) que la cantidad de alarmas mostradas en la consola de alarmas, esto es, las alarmas que el usuario observa es de 3426 para un momento dado. Mientras que como lo muestra la figura 31 (b) las alarmas totales que recibió el *ovtrapd* del NNM es de 79632. El dato de las alarmas totales que procesó el *ovtrapd* se obtuvo por medio del comando *ovdumpevents* el cual recorre la base de datos de eventos y la presenta en la terminal una alarma por línea y se cuentan las líneas por medio del utilitario de UNIX *wc*.

Ack	Cor	Severity	Date/Time	Source	Message
*		Warning	Sat Sep 25 09:12:02	Pruebas Boltec	Node status - major
		Warning	Sat Sep 25 09:20:46	BC-BS-01	The status of an XMAP-tracked virtual port has changed (Port - 30, L
		Warning	Sat Sep 25 09:20:46	AC-BS-01	The status of an XMAP-tracked virtual port has changed (Port - 31, L
		Normal	Sat Sep 25 09:20:58	BC-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 394 admin status
		Normal	Sat Sep 25 09:20:58	BI-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 679 admin status
		Normal	Sat Sep 25 09:20:58	BC-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 395 admin status
		Normal	Sat Sep 25 09:20:59	BI-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 678 admin status
		Warning	Sat Sep 25 09:20:59	AC-BS-01	The status of an XMAP-tracked virtual port has changed (Port - 31, L
		Warning	Sat Sep 25 09:20:59	BC-BS-01	The status of an XMAP-tracked virtual port has changed (Port - 30, L
*		Warning	Sat Sep 25 09:17:33	NGUA	Node etatus - warning
		Minor	Sat Sep 25 09:32:22	AIEV	Inconsistent subnet mask 255.255.255.192 on interface lan0, should b
		Major	Sat Sep 25 09:35:40	WAN2	VitalNet (Errors: ipt WAN-2 IP=10.1.0.16). Valor pico=53. Fecha: Sa
		Normal	Sat Sep 25 09:38:39	BR-CS-01	ATM VCC created or deleted (slot 3 port 1 VPI 0 VCI 157 admin status
		Normal	Sat Sep 25 09:38:39	BC-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 396 admin status
		Normal	Sat Sep 25 09:40:16	BI-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 679 admin status
		Normal	Sat Sep 25 09:40:16	BC-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 395 admin status
		Normal	Sat Sep 25 09:44:16	BI-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 678 admin status
		Normal	Sat Sep 25 09:44:16	BC-BS-01	ATM VCC created or deleted (slot 3 port 2 VPI 0 VCI 394 admin status

Critical:262 Major:1160 Minor:188 Warning:909 Normal:907 (129 acknowledged)

Cantidad total de alarmas mostradas en la consola de alarmas

(a)

```

Trying 10.2.191.75...
telnet: Unable to connect to remote host: Connection refused
[monitoreo-hp] oper2 >/export/home/oper2> ping 10.2.191.75
10.2.191.75 is alive
[monitoreo-hp] oper2 >/export/home/oper2> ovdumpevents | wc
 79632 1461886 11005792
[monitoreo-hp] oper2 >/export/home/oper2>

```

(b)

Figura 31. Comparación de Alarmas procesadas y Alarmas Totales

Con el procesamiento de las alarmas más importantes realizadas durante este trabajo, se pudo eliminar más de un 95% de las alarmas que no aportan ninguna información nueva, sino que más bien dificultan la lectura de la consola de alarmas, lo cual hace que el técnico de monitoreo pierda más tiempo investigando las causas de alguna falla. Con esto se beneficia la gestión del desempeño y la gestión de averías.

El proceso se debe continuar hasta llegar a un punto óptimo donde no se muestre ninguna alarma redundante siguiendo el procedimiento que se utilizó para eliminar las alarmas de mayor importancia, esto es, clasificar las alarmas como síntomas y diagnósticos y realizar el procesamiento requerido.

4.3 Acciones no automáticas.

En la administración de la red no sólo es necesario tener un manejo adecuado de las alarmas sino que también se requiere que el técnico pueda realizar cambios en la configuración de los equipos de una forma sencilla.

La programación de las tareas de configuración de los equipos se consideró como tareas no automáticas ya que el usuario necesita acceder la herramienta y ejecutar los cambios que requiera.

Con esta programación se buscó abarcar las acciones que se realizan más comúnmente en los conmutadores y los enrutadores, además no se necesita que el usuario conozca la clave de acceso de los equipos a administrar ya que no necesita conectarse por medio de telnet.

Para realizar los cambios en la configuración de los equipos el usuario se conecta utilizando un navegador de páginas Web en su máquina por medio del protocolo de transferencia de hipertexto (HTTP) y abre una página dinámica que se corre en el CGI del servidor Web que corre en el NNM (httpd). A diferencia de una página estática, la página dinámica presenta diferentes resultados o ejecuta diferentes acciones dependiendo de la información que recibe del usuario por medio de formularios. El programa de la página dinámica lee las variables del formulario y se comunica por medio del protocolo SNMP para interactuar con los equipos de la red utilizando los utilitarios *snmpwalk* y *snmpset*. Se puede considerar que el NNM actúa como un *proxy* al procesar las solicitudes en HTTP y traducirlas a los requerimientos de SNMP necesarios para realizar los cambios que desea el usuario final.

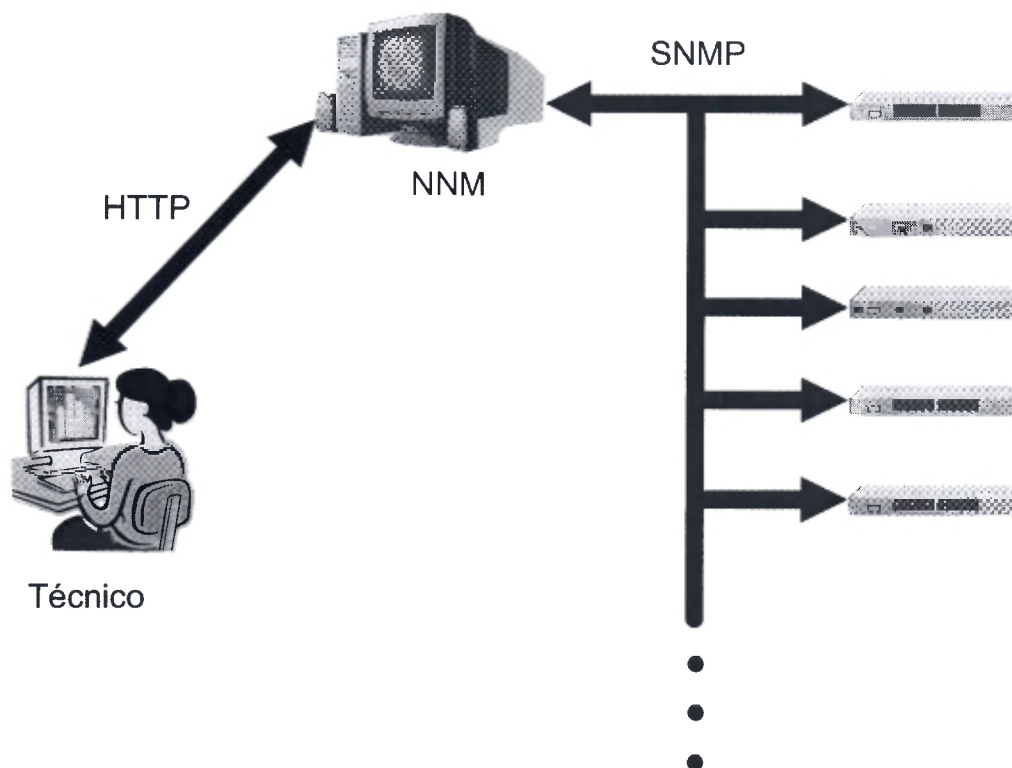


Figura 32. Interacción de la página dinámica

Debido a que los enrutadores y los conmutadores cumplen acciones diferentes en la red y que el usuario final está acostumbrado a obtener información diferente de cada uno de ellos, se hicieron dos interfaces gráficas diferentes.

A continuación se muestran los detalles y resultados en la elaboración de las dos interfaces gráficas de administración.

4.3.1 Administración de Conmutadores Alcatel (Web-Vision)

Hasta el momento el banco contaba con un programa de administración de conmutadores propietario denominado X-Vision. Sin embargo el banco cuenta únicamente con

una licencia por lo que no se cumple con lo que el Banco necesita, esto es acceso a la configuración de los equipos desde diferentes puntos. Por esta razón se le dio especial importancia a la construcción de una interfaz de administración que sea accesible para los usuarios que la requieran.

El programa para hacer los cambios en las configuraciones de los conmutadores se denominó “Web-Vision” haciendo referencia a la plataforma propietaria X-Vision y se encuentra en el archivo /opt/OV/www/cgi-bin/aut3.pl.

Los cambios que se pueden hacer en la configuración de los conmutadores por medio del Web-Vision son:

- Cambio de grupo de un puerto.
- Cambio de la velocidad de un puerto (10, 100, autonegociación).
- Cambio del modo del puerto (half, full, autonegociación).
- Habilitar o deshabilitar un puerto.
- Realizar un «port mirroring» (Ver glosario).

Se puede también observar sin posibilidad de modificar los siguientes elementos:

- El nombre del conmutador.
- Los grupos (VLAN) y direcciones IP del conmutador
- Si el puerto tiene algún elemento conectado.
- Si el puerto ha permanecido sin actividad.
- La localización del conmutador.
- El número de placa del conmutador (Número de activo del Banco Nacional).
- El número de serie del conmutador.

Los tres últimos son muy útiles en la gestión de la contabilidad.

Presenta accesos directos a los respaldos de la «flash card», a los respaldos de las configuraciones *CLI* (ver sección 4.2.2), un acceso *ftp* al conmutador y un utilitario de búsqueda de direcciones IP.

El utilitario de búsqueda de direcciones IP sirve para identificar en cuál conmutador y cuál puerto se encuentra una dirección IP determinada. También permite el proceso inverso, o sea dado un conmutador y un puerto, establecer cuál es la dirección IP conectada en él. Este utilitario realiza una búsqueda iterativa por medio de SNMP y el programa que lo hace se puede ver en el apéndice B.

La figura 33 muestra la interfaz mostrada al usuario.

AD-OS44 PM1V
Visite: www.bncr.fi.cr

Localización:
Piso M2 Edificio Viejo 307833 901630 10-A-02282217

VLAN ID	Descripción del Grupo	Dirección IP	Mascara
1	Default GROUP (#1)		
2	back2	10.1.1.63	255.255.255.0
3	Red No.2 Piso M2 Edifc. Viej.	10.0.112.1	255.255.255.0

Otros Links:

Nuevo Switch:

Puerto	Grupo	C. Grupo	Veloc.	C. Veloc.	Duplex	C. Dup.	Status	C. Status	Port Mirroring	Ptos Disponib.	Refrescar
2/1	2	2	100	100	full	full	enable	ena	<input type="checkbox"/>	-	2/1
2/2	3	3	100	100	full	full	enable	ena	<input type="checkbox"/>	-	2/2
3/1	3	3	100	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/1
3/2	3	3	auto	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/2
3/3	3	3	100	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/3
3/4	3	3	100	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/4
3/6	3	3	100	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/6
3/6	3	3	auto	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/6
3/7	3	3	100	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/7
3/8	3	3	100	auto	half	half	enable	ena	<input type="checkbox"/>	-	3/8
4/1	3	3	100	auto	half	half	enable	ena	<input type="checkbox"/>	-	4/1
4/2	3	3	auto	auto	half	half	enable	ena	<input type="checkbox"/>	-	4/2

Figura 33. Vista del Web-Vision

El programa principal del Web-Vision se puede analizar en bloques funcionales.

4.3.1.1 Subrutinas

El objetivo de las subrutinas es procesar la información obtenida por medio del *snmpwalk* del NNM.

Los parámetros son pasados por valor y la salida es un arreglo de datos asociativo, conocido como *hash*, cuyas llaves son las instancias de la MIB y cuyos valores son el contenido de la MIB asociado con la instancia.

En otras palabras se pasa de una variable con información ordenada en esta forma:

```
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.1.4.1: INTEGER: portUp
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.2.4.1: INTEGER: portDown
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.3.4.1: INTEGER: portDown
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.4.4.1: INTEGER: portDown
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.5.4.1 : INTEGER: portUp
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.6.4.1 : INTEGER: portUp
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.7.4.1 : INTEGER: portUp
xylan.xylanArch.xylanVportArch.virtualPort.vportTable.vportEntry.vportOperStatus.5.8.4.1 : INTEGER: portUp
```

A un arreglo de variables ordenado así:

```
$HPortStatus {5/1} =portUp
$HPortStatus {5/2} =portDown
$HPortStatus {5/3} =portDown
$HPortStatus {5/4} =portDown
$HPortStatus {5/5} =portUp
$HPortStatus {5/6} =portUp
$HPortStatus {5/7} =portUp
```

```
$HPortStatus{5/8}=portUp
```

El arreglo resultante es más sencillo de manejar que la variable resultante de recorrer la MIB ya que la llave del arreglo es el puerto del switch que interesa.

Debido a que el contenido en la MIB del conmutador varía, fue necesario construir diferentes tipos de subrutinas para dar un tratamiento adecuado a cada una de las variables que interesan. Los diferentes tipos de subrutina se pueden observar en el apéndice B.

4.3.1.2 *Lectura de las variables de estado y de las variables de entrada*

El Web-Vision y prácticamente cualquier otra página dinámica es una máquina de estados donde las variables son pasadas por medio de formularios de HTML. Con base en estas variables se toma la decisión de qué se va a hacer.

Las variables están incluidas en los formularios de cualquier página HTML. Aquellas tipo oculto (*hidden*) representan las variables de estado. Cualquier otro tipo de variable representa una variable de entrada. En el cuadro 3 se muestran las variables en uno de los formularios del Web-Vision y su clasificación como variable de entrada o variable de estado.

El estado actual está completamente determinado por las variables de estado. Si se nota que por ejemplo la variable `gru_2/1` que es una variable de estado es igual a 2 pero que la variable de entrada `c_gru_2/1` es diferente de 2 significa que el usuario desea que el puerto 2/1 del conmutador AD-OS44-PM1V (contenido en la variable de estado `Nom_switch`) cambie de grupo por lo que el programa utiliza el *snmpset* para realizar el cambio deseado.

Cuadro 3. Variables de estado y entrada de un formulario del Web-Vision

Nombre de la Variable	Clasificación	Valor de ejemplo
IP_switch	Variable de estado	120.10.112.1
Puerto	Variable de estado	2/1
Formul	Variable de estado	Ver
Name	Variable de estado	oper2
Nom_switch	Variable de estado	AD-OS44-PM1V
gru_2/1	Variable de estado	2
c_gru_2/1	Variable de entrada	1
veloc_2/1	Variable de estado	100
c_veloc_2/1	Variable de entrada	100
dup_2/1	Variable de estado	full
c_dup_2/1	Variable de entrada	full
status_2/1	Variable de estado	enable
c_status_2/1	Variable de entrada	enable
checkbox_pm_2/1	Variable de entrada	Habilitar_PM
pm_o_2/1	Variable de estado	_checked

Las variables de entrada sirven también para la autenticación del usuario y para el registro de los cambios que le hizo a la configuración (*Accounting*). Las tareas de autenticación y “*Accounting*” la realizan los dos siguientes bloques del programa.

4.3.1.3 Autenticación del usuario

La autenticación consiste en determinar si uno de los usuarios del sistema es realmente quién dice ser.

Microsoft decidió incluir en forma gratuita su Browser para Internet en el sistema operativo Windows por lo que es importante restringir al acceso a las aplicaciones Web que puedan afectar la operación de algún sistema como es el caso del Web-Vision.

Para poder ingresar al Web-Vision solamente se puede hacer por medio de la página de solicitud de autenticación. Si se trata de ingresar por otra ruta aparece el mensaje de que no se cuenta con autorización para ver la página. En la página de solicitud de autenticación se solicita el nombre del usuario y la clave.

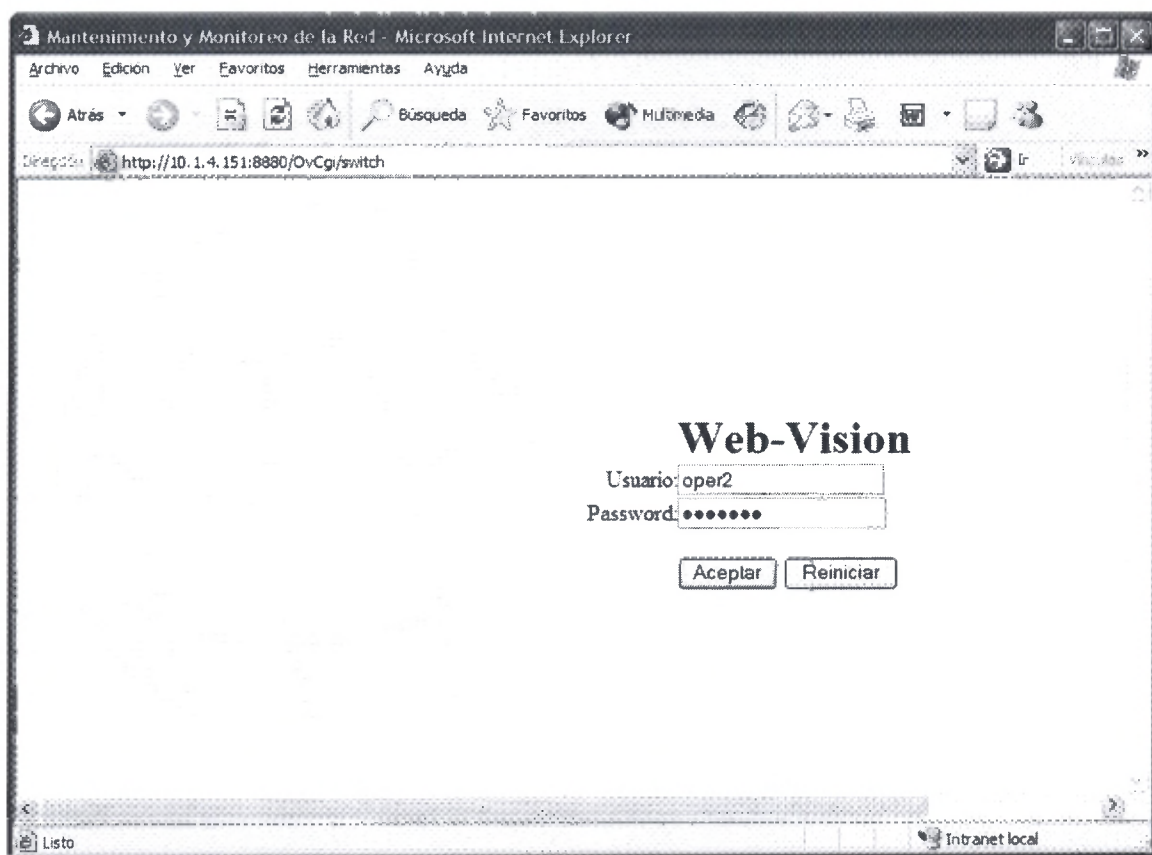


Figura 34. Página de autenticación del Web-Vision

La información recogida por la página de autenticación es enviada a la página del Web-Vision que revisa si la dirección IP que está haciendo la solicitud de ingreso está dentro de la lista

de las direcciones IP autorizadas para el ingreso. La lista de direcciones se encuentra en el archivo de texto */etc/IP.wv* por lo que el Web-Vision debe abrir el archivo y realizar la comprobación. Este es el primer paso de la autenticación. El segundo paso es comprobar que la clave que se digitó corresponde al asignado al usuario correspondiente el cual se guarda en el archivo */etc/usuarios.wv*. Si se cumplen estas dos condiciones se concluye que el usuario es quien dice ser y se permite el acceso. El nombre del usuario pasa a ocupar el valor de la variable Name que va a ser necesaria para el proceso de registro de los cambios. En caso de no se pueda autenticar aparece un mensaje de que no está autorizado para ver la página.

4.3.1.4 Cambios en la configuración y registro de los cambios (*Accounting*)

El bloque de cambios y registro de los cambios compara las variables de entrada y estado asociadas con valores de configuración. La función de próximo estado en este caso es igual a la entrada.

Después de aplicar la función de próximo estado aplica los cambios en el conmutador por medio del *snmpset*. Además se llena el archivo con el registro de cambios el cual solamente puede ser observado y cambiado por el administrador del NNM. Este archivo de cambios indica la fecha y hora en que se realizó el cambio, quién lo hizo, qué hizo y en cuál conmutador lo hizo. En otras palabras el archivo de control de cambios muestra cuándo, quién, qué y dónde.

Una muestra de este archivo se muestra a continuación:

```
Mon Sep 20 10:59:21 CST 2004 : El Usuario oper18 habilito el puerto 5/8 en el switch BI-OS52-P8 (10.1.18.11)
```

```
Mon Sep 20 11:00:18 CST 2004 : El Usuario oper18 deshabilito el port mirroring en el puerto 3/5 del switch BI-OS52-P8 (10.1.18.11)
```

```
Mon Sep 20 11:32:14 CST 2004 : El Usuario oper18 cambio el grupo del puerto 5/7 del 1 al 3 en el switch BS-OS93-P17 (10.2.64.12)
```


Wed Sep 22 08:27:50 CST 2004 : El Usuario oper2 cambio el grupo del puerto 4/1 del 1 al 3 en el switch BM-OS96-P12 (10.1.128.14)

Wed Sep 22 08:47:22 CST 2004 : El Usuario oper4 habilito el port mirroring en el puerto 3/1 para verlo en el 3/4 del switch AB-OS39-PM1V (10.0.48.13)

Con esto se lleva un control de cambios completo. En caso de que se presente una falla lo primero que puede hacer el técnico es preguntar cuándo se dio la misma y revisar los cambios que se hayan hecho en los conmutadores a esa hora, en caso de notar un cambio en la configuración equivocado, puede invertir el cambio rápidamente ya que el registro muestra también la configuración anterior..

4.3.1.5 Lectura de campos de MIB y procesamiento

El siguiente bloque es el que realiza la lectura de los campos de interés de la MIB. Los campos de interés se colocan en el vector de lectura llamado “@uma_guma” en el programa. Luego se extrae la información del agente SNMP del conmutador y se deja en una variable (\$MISC) que luego es dividida en bloques para ser procesada por las subrutinas de procesamiento.

Las funciones de generación del arreglo asociativo dejan las variables listas para la presentación en HTML que procesa el browser de Internet.

En caso de que en el futuro se desee que el Web-Vision desee mostrar otra parte de la configuración del conmutador esta se debe colocar en el vector de lectura y modificar el bloque de separación, lo cual es una tarea sencilla.

4.3.1.6 Presentación en HTML

Una vez que se ha realizado todo el procesamiento, el último paso es preparar la presentación al usuario.

Trabajar con el HTML es bastante difícil si no se cuenta con la experiencia necesaria principalmente cuando se trata de las proporciones de la presentación por lo que se utilizó una herramienta para el diseño de la presentación: el Dreamweaver de Macromedia. Mediante este software se puede trabajar en forma simultánea con el texto HTML y con la presentación.

La figura 35 muestra el Dreamweaver. Si se realiza un cambio en la ventana inferior el Dreamweaver refresca inmediatamente el código HTML asociado con el cambio realizado mostrado en la ventana superior de la figura 35 y viceversa.

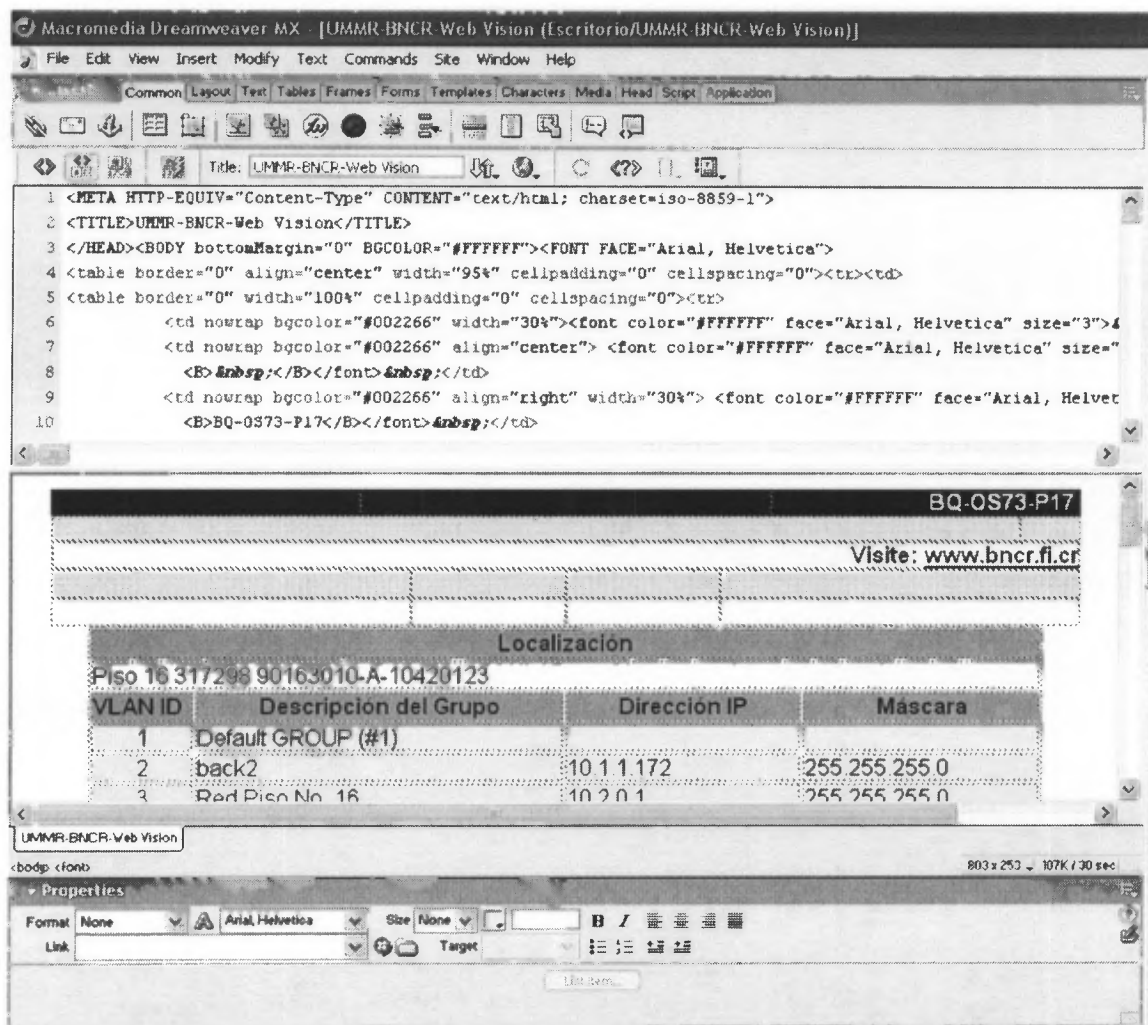


Figura 35. Página de autenticación del Web-Vision

El texto generado por el Dreamweaver se incluye en el programa de PERL incluyendo las variables generadas en el bloque de cambios y procesamiento.

El programa completo se muestra en el apéndice B.

4.3.2 Administración de enrutadores Nortel (Web-Manager)

El software comercial de administración para los enrutadores Nortel es el Site Manager. Este software permite hacer cualquier cambio en la configuración por medio de SNMP pero es solamente para configuración y no muestra si alguno de los circuitos IP está caído, no muestra la tabla de rutas, no muestra estadísticas de los puertos ni el status de los puertos; para poder ver estos apartados, lo único que se puede hacer es un ingreso por medio de telnet.

Se decidió entonces crear una herramienta que permitiera hacer los cambios más frecuentes de la configuración en forma sencilla y además que permitiera ver información del estado del enrutador que no se puede ver por «*Site Manager*». A esta herramienta se le denominó “Web-Manager” para hacer referencia al software propietario «*Site Manager*» y se programó en varios archivos.

Utilizar varios archivos para programar el «Web-Manager» permite que el ancho de banda necesario para cada una de las acciones se reduzca ya que cada módulo se concentra en una parte pequeña de la MIB. Debe tomarse en cuenta que los enrutadores se encuentran en la red WAN donde los anchos de banda son limitados. El ancho de banda en la red de oficina principal no es un factor de peso limitante por el momento.

El archivo que da el marco principal, la primera autenticación y conecta con los demás archivos es el `/opt/OV/www/cgi-bin/aut4.pl`

Los cambios que se pueden hacer en la configuración de los enrutadores por medio del Web-Manager son:

- Cambio de costos en los circuitos IP.
- Habilitación y Deshabilitación de los circuitos IP.
- Habilitación y Deshabilitación de los puertos sincrónicos.

- Modificación de la configuración del RIP.
- Eliminación de entradas de la tabla ARP del enrutador.
- Cambios en el sistema de archivos de la flash card (Copiar archivos, borrar archivo, salvar la configuración, compactar la flash card).

Se puede también observar los siguientes elementos:

- Estado, nombre, direcciones y máscaras de los circuitos IP.
- Estado, nombre, Dirección MAC, número de línea, tramas transmitidas y recibidas, bytes transmitidos y recibidos, errores y protocolo de los puertos sincrónicos.
- Bytes, Tramas, *FECNs*, *BECNs* y *DLCI* de los circuitos con Frame Relay.
- La localización del enrutador, el nombre y la dirección regional a la que pertenece.
- El estado del manejo del Frame Relay.

Presenta también un acceso directo a los respaldos de los archivos de configuración de los enrutadores en el servidor ftp.

La figura 36 muestra la interfaz gráfica del Web Manager que ve el usuario. En el marco superior se encuentra la información general del enrutador. En el marco izquierdo se muestran los accesos a los diferentes módulos y el marco principal es el que muestra el resultado del módulo que se está corriendo.

Administración de Routers Nortel -- Banco Nacional

Nombre	Lugar	Dirección Regional	Dirección IP
ARN MSPE	Mall San Pedro 303642 NEP0152207	Dir. Reg. San Jose Este	10.10.16.1

Nuevo Router:

Circuitos IP

Nombre	No. de Circuito	Estado	Dirección IP	Mascara	Costo
E1_MSPE	1	up <input type="checkbox"/> Dis	10.10.16.1	255.255.255.0	1 1 ▾
G703_MSPE_WAN1_ICEFRE	5	up <input type="checkbox"/> Dis	10.1.5.17	255.255.255.0	1 1 ▾
S11_MSPE_BCNI_TELPPP	2	up <input type="checkbox"/> Dis	10.10.31.5	255.255.255.252	4 4 ▾

Figura 36. Vista del Web-Manager

Los programas o módulos que componen el Web-Manager son los siguientes:

```

/opt/OV/www/cgi-bin/enrutador
/opt/OV/www/cgi-bin/aut4.pl
/opt/OV/www/cgi-bin/sic.pl
/opt/OV/www/cgi-bin/ssb.pl
/opt/OV/www/cgi-bin/sss.pl
/opt/OV/www/cgi-bin/sfs.pl
/opt/OV/www/cgi-bin/sir.pl
/opt/OV/www/cgi-bin/rip.pl
/opt/OV/www/cgi-bin/arp.pl

```

```
/opt/OV/www/cgi-bin/sfl.pl  
/opt/OV/www/cgi-bin/arc.pl
```

Para programar cada uno de estos módulos se utilizaron los mismos bloques y se reutilizó mucho código del Web-Vision por lo que cada subprograma tiene una sección de subrutinas para el procesamiento, una sección para autenticación, otra para cambios en la configuración y *accounting* (para los módulos que así lo requieren ya que hay módulos que son sólo de observación y no se puede realizar ningún cambio), otra parte para leer la parte de la MIB de interés y el procesamiento de esta y por último la presentación en HTML.

El código completo de todos los programas se muestra en el apéndice B.

Como se había mencionado, la principal ventaja de realizar cada una de las acciones por separado es un uso más reducido del ancho de banda. Esto se puede apreciar en la figura 37 que muestra la comparación de realizar un cambio en el costo de un circuito IP por medio del «Site-Manager» y por medio del Web-Manager. La columna de «*Cumulative Bytes*» indica la suma de todos los bytes que se necesitaron para llevar a cabo el proceso. Puede observarse que mediante Site Manager se necesitaron 83768 bytes para realizar el cambio de costo mientras que con el Web-manager se necesitaron solamente 10324. En otras palabras se economiza el 87% de los bytes necesarios.

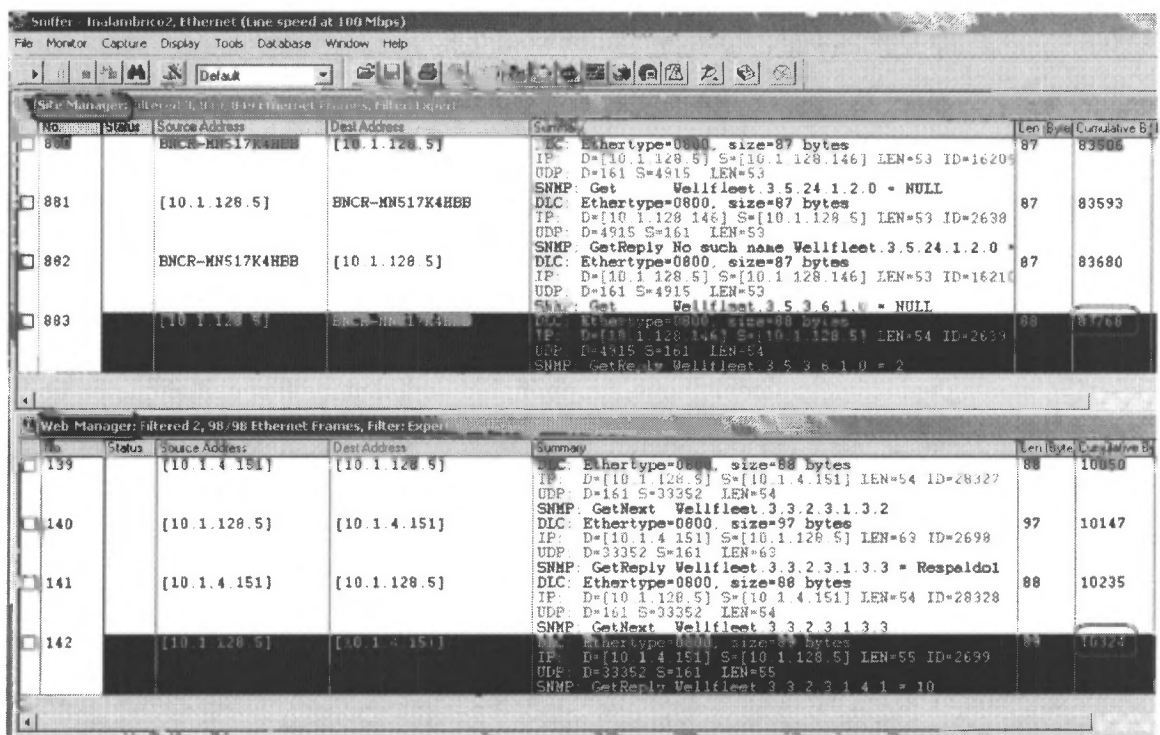


Figura 37. Comparación entre Web-Manager y Site Manager para cambio de costo

4.3.3 Corolario de Acciones No Automáticas

Las aplicaciones Web, al ser completamente gráficas permiten que el usuario las domine en menos tiempo lo cual reduce el tiempo de la capacitación de empleados nuevos.

Son bastante claros los beneficios de las acciones automáticas. En primer lugar se obtiene una interfaz completamente gráfica y fácil de usar para el usuario. Por otro lado no es necesario el uso de una lista de claves ya que solamente se requiere una autenticación inicial para entrar a la herramienta y de ahí en adelante se puede tener acceso a cualquier elemento de la red del Banco Nacional.

Al contar con una herramienta gráfica, al acceder un switch la visualización de todos los detalles de un puerto es mucho más rápida. Si se quisiera solamente conocer la información

mostrada en la figura 38 de la forma clásica (telnet) Se requerirían una serie de pasos: el primero es el *login* donde por supuesto es necesario que el técnico conozca la clave de acceso.

```
Welcome to the Alcatel OmniStack! Version 4.3.3.142.B PATCH
login   : admin
password:
```

```
*****
```

```
Alcatel OmniStack
Copyright (c), 1994-2001 Alcatel Internetworking, Incorporated. All
rights reserved.
OmniStack is a trademark of Alcatel Internetworking, Incorporated,
registered in the United States Patent and Trademark Office.
System Name:      BM-OS66-P12
System Location:  Piso 12 296416 90163010-A-01581921 (1)
```

Posteriormente debería aplicarse el commando *gp* para ver la información de las *VLAN*.

```
BM-OS66-P12/->gp
Group ID          Group Description          Network Address Proto/
(:VLAN ID)                                     (IP Subnet Mask) Encaps
or (IPX Node Addr)
=====
  1 Default GROUP (#1)
  3 Red Piso No.12          10.1.128.12      IP / No Routing
                               (ff.ff.ff.00 )  ETH2 (2)
```

A continuación se debería dar el comando *vi* para ver que el puerto está activo, habilitado y a cuál *VLAN* está asociado.

```
BM-OS66-P12/->vi 4/7
Virtual Interface Summary Information- For All Interfaces

Slot/ Type/                                     Status
Group Intf  Inst/Srvc  MAC Address  Prt Encp Admin Oper Spn Tr
=====
  3 4/7 Brg/  1/ na  00d095:26fc01 Tns DFLT Enabld Active Disabl (3)
```

Para ver la configuración de velocidad y duplex se necesitaría ejecutar el comando *10/100vc*.

BM-OS66-P12/Interface/ETH10+100->**10/100vc**

10/100 Configured Values for all slots

Slot/ Intf	Auto- negotiate	DETECTED		SET		
		Line Speed	Duplex Mode	Line Speed	Duplex Mode	
4/ 7	disabled	100	FULL-D	100	full-d	(4)

Por último para ver si tiene algún *port-mirroring* configurado debería usarse el comando *modvp*.

BM-OS66-P12/Interface/ETH10+100->**modvp 3 4/7**

Modify local port 4/7 (Virtual port (#17)) ? (y) :

Modify Ether 10/100 Vport 4/7 Configuration

```

1) Vport/Group/Instance/Type : 17/3/1/Brg
2) Description                : Virtual port (#17)
3) Bridge Mode                : Auto-Switched
  31) Switch Timer            : 60
4) Flood Limit                : 192000
5) Output Format Type          : Default(IP-Eth II, IPX-802.3)
6) Ethernet 802.2 Pass Through : Yes
7) Admin, Operational Status  : Enabled, active
8) Mirrored Port Status      : Disabled, available      (5)
9) MAC address                 : 00D095:26FC01

```

Command {Item=Value/?/Help/Quit/Redraw/Next/Previous/Save} (Redraw) : q
 BM-OS66-P12/Interface/ETH10+100->

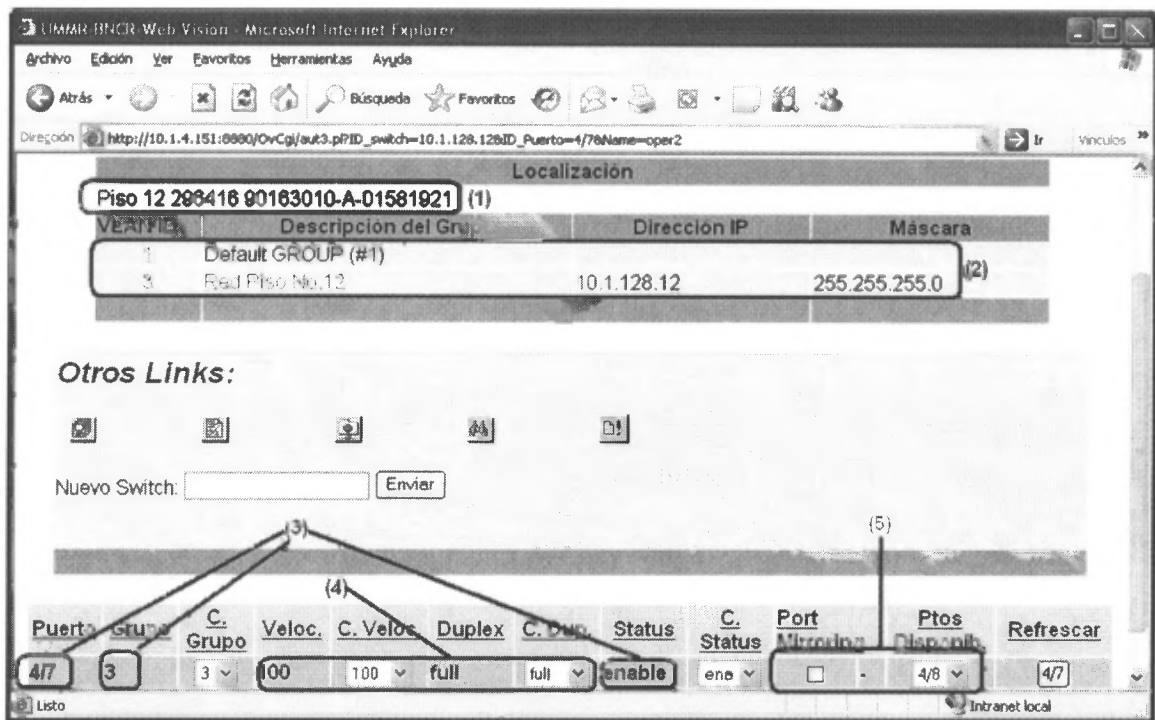


Figura 38. Vista de Un Puerto por medio del Web Vision

Los puntos del 1 al 5 incluyen los pasos necesarios para la visualización por lo que si se desea realizar algún cambio en la configuración serían necesarios más pasos o más comandos que el técnico debería conocer y ejecutar sin error. Este ejemplo se puede extender a otros pasos que debería conocer para buscar direcciones IP o para realizar cambios en los enrutadores.

Se tiene además el beneficio de la reducción de bytes que se trasiegan en la red lo cual implica un menor consumo del ancho de banda disponible para el caso del Web-Manager.

Otro beneficio es que se concentra la cantidad de estaciones NNS a solamente 1 que es el NNM mismo. Esto hace que las listas de acceso de los equipos se puedan disminuir y dificulta la labor de los “hackers”. La figura 39 muestra la lista de acceso de uno de los conmutadores. Como puede apreciarse basta con definir dos estaciones a pesar de contar con 15 técnicos.

```
Modify (m) :  
Find (f) :  
Help (h) :  
Quit (q) :  
  
Enter selection : 1  
  
Filter Name          Source IP      Source Mac     Slot  Port  
Address              Address        Address        #     #  
-----  
VitalNet             10.1.4.20     ANY            ANY   ANY  
monitoreo-hp         10.1.4.151    ANY            ANY   ANY  
  
Secure Access Filter Database  
  
List (l) :  
Create (c) :  
Delete (d) :  
Modify (m) :  
Find (f) :  
Help (h) :  
Quit (q) :  
  
Enter selection :
```

Figura 39. Lista de Acceso de un conmutador

CAPÍTULO 5. CONCLUSIONES.

1. En este trabajo se adaptó un software de administración no propietaria a los equipos de red del Banco Nacional de Costa Rica. Esto prueba la hipótesis planteada. Es claro que la integración del sistema NNM de Hewlett-Packard no es directa (“plug and play”) sino que requiere la realización de ciertos programas que interactúen con el sistema. El PERL es un lenguaje que cumple con las características necesarias para realizar esta programación ya que realiza un excelente procesamiento de texto y puede interrelacionarse con el sistema operativo.
2. Con el estudio realizado sobre los eventos que se presentan con mayor frecuencia fue posible determinar un patrón y dar un diagnóstico con el que se puede presentar adecuadamente las alarmas de los equipos. La presentación adecuada de las alarmas incluye la eliminación de los mensajes redundantes, esto es, varios mensajes que alertan sobre un mismo evento; también incluye la presentación en español, sin código hexadecimal, haciendo una referencia al nombre del equipo y separada por zona regional del Banco. Como se puede ver en la sección 4.2.11 la cantidad de alarmas mostradas, en comparación con las alarmas desplegadas al técnico, se reduce en un 95% lo cual hace que la lectura de la consola de alarmas sea más sencilla. El estudio se puede también extender a los eventos que se presentan con menor frecuencia hasta que la totalidad de los mensajes en la consola de alarmas sean presentados adecuadamente.
3. Los resultados de este trabajo muestran que es posible construir una aplicación Web que permita la administración sencilla y eficiente de los equipos de red. Administración sencilla se refiere a que se realiza con una interfaz gráfica completamente amigable al usuario, lo cual hace que el usuario no tenga que aprenderse los comandos del equipo de red ni la clave del

equipo ni sus direcciones IP lo cual reduce el uso de diagramas y de claves así como también mejora el tiempo de capacitación en el uso de los equipos ya que las aplicaciones gráficas realizadas en este trabajo (Web Vision y Web Manager) fueron presentadas en una reunión de pocas horas para que los técnicos empiecen a usarlas. Con administración eficiente se quiere decir que se reducen los paquetes de SNMP que pasan por la red como se puede ver en la sección 4.3.2. En la programación de la aplicación Web el PERL también es pieza clave principalmente con la librería CGI.

4. El principal problema para la realización de este trabajo fue el hecho de que la red del Banco está conformada por equipos diferentes lo cual hace que el tiempo del estudio de las estructuras de las MIBs de equipos de diferente fabricante aumente. Este estudio representó bastante trabajo para una sola persona.

CAPÍTULO 6. RECOMENDACIONES.

1. Unificar los equipos de comunicaciones para no tener que hacer programación para equipos de distinto fabricante. Con esto bastaría concentrarse en el estudio de una sola estructura de MIB para generar los programas que interaccionan con el NNM. Además es conveniente que sean dos personas las que trabajen con el NNM y su adaptación con los equipos de red del banco.
2. Se mencionó que se trabajó con los eventos principales para determinar un diagnóstico sin embargo no se realizó ningún trabajo con las alarmas que se presentan con menor frecuencia, las cuales sí tienen que ser interpretadas por el técnico que lee la consola de alarmas. Es recomendable trabajar con estas alarmas menos frecuentes y hacerles el mismo tratamiento de las más frecuentes para su presentación adecuada, hasta que el 100% de las alarmas de la consola sea el óptimo.
3. Las aplicaciones Web también fueron desarrolladas para cumplir con las tareas más frecuentes sin embargo es conveniente darle seguimiento y obtener retroalimentación de los técnicos para detectar las necesidades de incluir nuevas acciones y satisfacer estas necesidades, es decir, programar las acciones que los técnicos consideran necesarias.
4. Para la comunicación por SNMP con los equipo se utilizaron los utilitarios *snmpwalk* y *snmpset* que son parte del NNM. El *snmpwalk* realiza una lectura de un objeto de la MIB a la vez lo que hace que sea susceptible a mejora. Por medio de la versión de GNU del AWK se puede componer una cadena de bits siguiendo el protocolo de SNMP definido en el RFC-1157 y enviarlo al agente de SNMP que corre en el equipo de red. La versión de GNU de AWK también contienen un operador bidireccional de coproceso (`|&`) lo cual permite que haya

comunicación de dos vías con el proceso, esto es enviar la comunicación a la red y recibir la respuesta. Así:

```
BEGIN {
    "/inet/tcp/0/direccion del agente/161" |& getline
    print $VAR
    close("/inet/tcp/0/direccion del agente/161")
}
```

El *getline* lee el resultado por estar después del operado bidireccional de coproceso. En el \$VAR se debe poner la cadena de bits respetando el protocolo SNMP y debe contener varios objetos en lugar de 1 como lo hace el *snmpwalk*. Con esto se tendría una menor cantidad de bytes trasegados en la red por la economía en bytes de encabezados.

5. La seguridad se puede mejorar aunque existe un nivel de seguridad en las aplicaciones Web, ya que es necesaria la autenticación del usuario por medio de una clave y existe solo un cierto número de direcciones IP que pueden accederlas la seguridad se puede mejorar. Las claves se pasan al servidor Web por medio de una POST de http. La clave pasa en texto plano por la red por lo que con un analizador de protocolos sería posible conocerlo. Este se puede mejorar de forma sencilla por medio de un "javascript" que aplique un algoritmo de encriptación de la clave antes de pasarlo por el POST. Este "javascript" ya existe y puede descargarse de Internet en <http://pajhome.org.uk/crypt/md5/md5src.html>. La descripción del algoritmo de hash se puede ver en el RFC-1321 (The MD5 Messge Digest Algorithm). Bastaría con incluir una subrutina que haga el mismo hash que el que hace el java script y hacer modificación en el módulo de autenticación del usuario para que no compare con la clave del archivo */etc/usuarios.wv*, sino que lo compare con el hash de la clave que aparece en el archivo mencionado. Si los dos hashes son iguales se garantiza la autenticación y la clave no pasa por la red.

CAPÍTULO 7. BIBLIOGRAFÍA.

1. Alcatel. *Manual del usuario de los equipos de comunicación de datos Alcatel*.
2. Case, J. et al. *RFC-1157 A Simple Network Management Protocol*. Internet Engineering Task Force. 1990. <http://www.ietf.org/rfc/rfc1157.txt?number=1157> .
3. Chova, Gabriel. *Una Breve Historia de Internet*. [http://www.articulos.astalaWeb.com/Internet%20-%20Historia/Una%20breve%20historia%20de%20Internet%20\(II\).asp](http://www.articulos.astalaWeb.com/Internet%20-%20Historia/Una%20breve%20historia%20de%20Internet%20(II).asp) .
4. Free Software Foundation INC. *The GNU AWK User's Guide*. 2004. <http://www.gnu.org/software/gAWK/manual/gAWK.html>.
5. Free Software Foundation INC. *TCP/IP Internetworking with gAWK Guide*. 2004. <http://www.gnu.org/software/gAWK/manual/gAWK.html>.
6. Gómez, Christiam. *Propuesta para la implementación de un sistema de gestión de Servicios para el Banco Nacional de Costa Rica*. San José, Costa Rica. Universidad Latina, 2003.
7. Hewlett-Packard Company. *Managing your Network with HP OpenView Network Node Manager*. Mayo 2002.
8. How Stuff Works. *How CGI Scripting Works*. 2004. <http://computer.howstuffworks.com/cgi.htm>.
9. RAD. *SNMP – Simple Network Management Protocol*. <http://www2.rad.com/networks/1995/snmp/snmp.html>
10. Saborío, Ronald, *Establecimiento de la administración preactiva en la red de datos del Banco Nacional*. San José, Costa Rica. Universidad de Costa Rica, 2004.
11. Sniffer University. *TCP/IP Network Analysis and Troubleshooting*. Network Associates Inc. 2002
12. Tanenbaum, A. *Redes de computadoras*. 3º Edición. Perason. México, D.F. 2001.
13. Wyke, Allan y Thomas, Donald. *Programación en PERL*. I Edición. Osborne-McGraw Hill Colombia. Colombia, 2002.
14. Nortel. *Manual del usuario de los equipos de comunicación de datos Nortel*.

CAPÍTULO 8. GLOSARIO.

AD HOC. Dispuesto para un fin, adecuado, apropiado.

AGENTE. Un programa que realiza una tarea específica y muy bien definida en “background”.

ARP. Protocolo de resolución de direcciones.

ASN.1. Abstract Syntax Notation 1 es un lenguaje que define la forma en que los datos son transmitidos entre sistemas de comunicación diferentes.

ATM. Tecnología de redes basada en la transmisión de datos en celdas de longitud fija, pequeña comparada con otras tecnologías.

BACKBONE. Red de conexión Principal.

BACKPLANE. Tarjeta que sirve para unir varios circuitos electrónicos.

BECN. Notificación de congestión anterior en conmutadores Frame Relay.

BROWSER. Aplicación de Software utilizada para localizar y desplegar páginas Web.

CAM. Espacio de memoria para almacenamiento de direcciones físicas.

CMIP. Protocolo estandar de OSI para administración de redes.

CONMUTADOR. Ver Switch.

CPU. Unidad Central de Procesamiento de un sistema computador.

DAEMON. Un proceso o servicio de una computadora que corre en “background”.

DIRECCIÓN IP. Dirección de 32 bits asignada a cada anfitrión que participa en una red de redes TCP/IP. Una dirección IP es una abstracción del hardware físico. Para hacer el ruteo eficiente, cada dirección IP se divide en red y parte de anfitrión.

DUPLEX. Direcciones simultaneas en que se comunican dos dispositivos. Full duplex significa que dos dispositivos pueden transmitir datos en forma simultánea. Half duplex significa que solo puede transmitir uno a la vez.

ENRUTADOR. Dispositivo que utiliza la dirección destino en un paquete IP, u otro protocolo con capacidad de definir un esquema de direcciones, para determinar ha cuál de sus interfaces enviarlo.

FECN. Notificación de congestión posterior en conmutadores Frame Relay.

FLASH CARD. Memoria de solo lectura borrable eléctricamente que puede grabarse a alta velocidad y que se utiliza como medio de almacenamiento en varios tipos de dispositivos computacionales.

FRAME RELAY. Protocolo de transmisión de datos en redes WAN.

GESTIÓN DE RED: Término genérico utilizado para describir sistemas o tareas que contribuyen a mantener, caracterizar y controlar efectivamente una red.

HACKER. En un sentido peyorativo este nombre se da a individuos que han ganado acceso no autorizado a sistemas de computación con el propósito de robar o corromper datos.

HARDWARE. Partes materiales de una computadora.

HASH. Denominación de un arreglo de datos asociativo en PERL.

HEXADECIMAL. Sistema de numeración en base 16, o sea contiene 16 símbolos diferentes para representar todos los números.

HRE. Módulo de enrutamiento por Hardware.

HTML. Lenguaje de Hipertexto. HTML define la estructura de un documento Web usando una variedad de etiquetas y atributos

HTTP. Protocolo de transferencia de hipertexto.

IP. Protocolo Internet, encargado del manejo del esquema de direccionamiento en redes. Desarrollado por el Departamento de Defensa de los Estados Unidos.

IPCP. Protocolo de Control de IP sobre PPP. Mediante este protocolo se establece y controla una conexión con IP con el protocolo de Punto a Punto.

ISO. Organización Internacional para la Estandarización.

LAN. Cualquier tecnología de red física diseñada para cubrir distancias cortas (del orden de unos cuantos cientos de metros). Un ejemplo son las redes *ethernet*.

MIB. Una base de datos de objetos que puede ser accesada por medio del protocolo SNMP. Esta base de datos está en todos los equipos que usen SNMP y guardan estadísticas e información de la configuración del equipo.

NNM. Iniciales de “Network Node Manager”, el cuál es un producto de la empresa Hewlett Packard para el monitoreo y la Administración de Redes IP.

OSI: Interconexión de sistemas abiertos. Es el programa de estandarización internacional creado por la ISO y la UTU-I para desarrollar estándares para las redes de datos que faciliten la interoperabilidad de equipos fabricados por diferentes proveedores.

PDU. Unidad de datos que no forman parte del encabezado sino que va dirigido a una capa superior.

PING. Nombre de un programa utilizado con las redes TCP/IP, que se usa para probar la accesibilidad de un destino, enviando una solicitud de eco ICMP y esperando una respuesta. El término es utilizado ahora como verbo, por ejemplo, “haz ping a la estación A para ver si esta activo”.

PORT MIRRORING. Estado de un puerto en el cual copia la información que está pasando por otro puerto con fines de monitoreo.

PPP. Protocolo de punto a punto para transmisión de datos en redes WAN.

PROXY. Un servidor que actúa en nombre de otro servidor.

PROCOLO. Acuerdo de formato para la transmisión de datos entre dos dispositivos.

RFC. Serie de notas acerca de la Internet iniciada en 1969 y sometidas a estudio del Internet Engineering Task Force, una comunidad internacional de diseñadores de red, operadores, vendedores e investigadores.

RIP. Protocolo de información de Rutas.

Rx. Recepción.

SNMP. Protocolo Simple de Administración de Redes.

SOFTWARE. Instrucciones o datos de computadora. Cualquier cosa que pueda ser almacenada electrónicamente.

SOLARIS. Sistema operativo tipo Unix de la empresa SUN Microsystems.

SWITCH. Dispositivo que filtra y conmuta paquetes en una red LAN. En algunos casos también realiza labores de enrutamiento.

TABLA ARP. Espacio reservado en memoria que contiene los pares de direcciones físicas y direcciones IP asociadas.

TCP. Protocolo de transferencia de datos. Protocolo que sirve para establecer una conexión entre dos máquinas.

TCP/IP. Es el nombre común que se utiliza para nombrar al conjunto de protocolos desarrollados por el Departamento de Defensa de Estados Unidos.

TELNET. Un programa de simulación de terminal para redes TCP/IP.

TRAP. Un evento de alarma de un dispositivo enviado con el protocolo SNMP.

Tx. Transmisión.

UDP. Protocolo de Datagramas de Usuario. Con UDP se realiza envío de datos sin establecer una conexión previa lo cual lo hace más rápido que TCP pero menos seguro.

VLAN O GRUPO. Conjunto de puertos en un conmutador que forman un mismo dominio de Broadcast.

WAN. Cualquier tecnología de red de computadoras que abarca distancias geográficas extensas. También llamadas redes de gran alcance, la WAN actualmente operan a bajas velocidades y tienen retardos significativos mayores que las redes que operan sobre distancias cortas.

ANEXO A. ACCIONES AUTOMÁTICAS

Cualquier copia modificación o distribución de todos los programas del Anexo A deben hacerse bajo los términos y condiciones de la licencia GPL descritos en <http://www.gnu.org/licenses/gpl.html>.

A.1 Archivo de *crontab*

```
#ident "@(#)root      1.19    98/07/06 SMI"    /* SVr4.0 1.1.3.1      */
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
# Lucas Lucas Lucas
#
10 3 * * 0,4 /etc/cron.d/logchecker
10 3 * * 0   /usr/lib/newsyslog
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
0-59 * * * * /usr/bin/perl /usr/local/bin/bloquear.pl
0 * * * * /usr/bin/echo > /tmp/caidos
0 0 * * 2 /usr/local/bin/correo_bloqueados.pl
0 12 * * * rm /tmp/reporte_enable /tmp/reporte_authF
0 4,8,12,16,20 * * * rm /tmp/LINK_D_reporte_enable
0 20 * * 5 /usr/HP_Ag/respaldo_alcatel
0 20 * * 5 /usr/HP_Ag/respaldo_nortel
0 12 * * 6 su - bin -c "/bin/HP_Ag/Reporte.pl"
```

A.2 Archivo de *generación y comprobación de respaldo para conmutadores Alcatel*

```

#!/bin/ksh
nombre=`snmpget $1 system.sysName.0 | AWK '{print $6}'`
if [ "$nombre" == "" ]
then
nombre=`grep $1 /etc/hosts | nAWK '{print $2}'`
fi
echo
echo
'*****
*'
echo
echo $nombre
echo
(echo user admin $2
sleep 1
echo lcd /export/home/oper2/ba/respaldo/$nombre
echo bin
sleep 1
echo prompt
sleep 1
echo hash
sleep 1
echo mget .
sleep 1
echo lcd simm
snmpwalk $1 .1.3.6.1.4.1.800.2.1.3.2.1.3.1 | grep simm > /dev/null
if [ $? -ne 0 ]
then
echo lcd ..
else
echo cd /simm
sleep 1
echo mget .
fi
sleep 2
echo by) | ftp -n $1
sleep 10

nombre=`snmpget $1 system.sysName.0 | AWK '{print $6}'`
if [ "$nombre" == "" ]
then
nombre=`nAWK '$1 == "'$1"' {print $2}' /etc/hosts`
fi
Operacion=`snmpget $1 chasControlState.1 | nAWK '{print $NF}'`
if [ "$Operacion" == "primary" ]; then op=1; else op=2; fi
flash=`snmpwalk $1 chasControlFFSFileSize.$op | nAWK '{x+=$NF} END {print
x}'`
echo

```

```
echo
'*****
*'
echo
echo $nombre
echo
cd /export/home/oper2/ba/respaldo/$nombre
dif=`ls -lRF | grep -v '/' | nAWK '{x+=5} END {print x}'`
if [ "$dif" != "$flash" ]
then
echo ERROR FLASH: $1 Switch=$flash Disco=$dif | mailx -s "$nombre" oper2
oper9 caguilarb@bncr.fi.cr
echo ERROR FLASH: $1 Switch=$flash Disco=$dif
echo
fi
echo $flash
echo $dif
```

A.3 Archivo de generación de archivos con formato CLI para conmutadores Alcatel

```

nombre_archivo=`snmpwalk $1 sysName | nawk '{print $NF}'`
printf "Informacion del Sistema . . . "
echo system description \``snmpwalk $1 chasDescription | nAWK '{for
(i=6;i<NF+1;i++) a=(a " " $i)} END {sub(/ /,"",a); print a}'`" >
$nombre_archivo
echo system admin-contact \`"Administracion del Backbone UMMR\`" >>
$nombre_archivo
echo system location \``snmpwalk $1 sysLocation | nAWK '{for
(i=6;i<NF+1;i++) a=(a " " $i)} END {sub(/ /,"",a); print a}'`" >>
$nombre_archivo
echo system name $nombre_archivo >> $nombre_archivo
echo Listo
printf "Configuracion grupos e IP . . . "
echo group 1 no enrutador ip >> $nombre_archivo
# Aqui se observa si tiene una direccion de la 10.1.1.0
echo $1 | /export/home/oper2/ba/rec_ip_xy.AWK | nAWK '$4=="10.1.1.0" |
grep 10 > /dev/null
if [ $? -ne 0 ]
then
k="routing-off"
else
k="Licas"
fi
echo $1 $nombre_archivo $k | nAWK '
{
sd="snmpwalk " $1 " vIPEnrutadorNetAddress"
sm="snmpwalk " $1 " vIPEnrutadoresubNetMask"
sn="snmpwalk " $1 " vLanDescription"
sp="snmpwalk " $1 " vportVlanNumber"
sa="snmpwalk " $1 " vportInUcastPkts"
ss="snmpwalk " $1 " mesmPortCfgSpeed"
sf="snmpwalk " $1 " mesmPortCfgDuplexMode"
rout=""
if ($3=="routing-off") rout="routing-off"
while ((sn | getline eso) > 0) {
split(eso,a,"(vLanDescription\.[ \t]+)")
x=0
while ((x+7) in a)
VLAN[a[2]]=(VLAN[a[2]] " " a[7+x++])
sub(/ /,"",VLAN[a[2]])
}
close(sn)
printf ". "
indice=3
while ((sd | getline eso) > 0) {
split(eso,a,"(vIPEnrutadorNetAddress\.[ \t]+)")
DIREC[a[2]]=a[5]}

```



```

close(sd)
printf ". "
while ((sm | getline eso) > 0) {
    split(eso,a,"(vIPEnrutadoresubNetMask\\.|[ \t]+)")
    MASK[a[2]]=a[5]}
close(sm)
printf ". "
while ((sp | getline puertos)>0) {
    split(puertos,gp,"[\\. ]")
    pto=(gp[8] "/" gp[9])
    PUERTO[pto]=gp[14]
}
close(sp)
while ((sa | getline actividad)>0) {
    split(actividad,ap,"[\\. ]")
    act=(ap[8] "/" ap[9])
    ACTI[act]=ap[14]
}
close(sa)
while (ss | getline speed) {
    split(speed,spp,"[\\. ]")
    spl=(spp[8] "/" spp[9])
    SPEED[spl]=spp[12]
}
close(ss)
while (sf | getline duplex) {
    split(duplex,hfd,"[\\. ]")
    dup=(hfd[8] "/" hfd[9])
    DUPLEX[dup]=hfd[12]
}
close(sf)
for (m in SPEED) if (SPEED[m]=="speed-auto") delete SPEED[m]
for (m in DIREC) {
    split(MASK[m],mascara,"\\.")
    split(DIREC[m],d,"\\.")
    ult_octet=(d[4]-d[4]%(256-mascara[4]))
    subr=(d[1] "." d[2] "." d[3] "." ult_octet)
    if (subr=="10.1.1.0") {
        pvlan[2]=m
        ndire[2]=DIREC[m]
        nmask[2]=MASK[m]
        nDesc[2]="back2"
        for (mm in PUERTO)
            if (PUERTO[mm]==m) PUERTO_AUXILIAR[mm]=2
    }
    else {
        ndire[indice]=DIREC[m]
        nmask[indice]=MASK[m]
        nDesc[indice]=VLAN[m]
        for (mm in PUERTO)
            if (PUERTO[mm]==m) PUERTO_AUXILIAR[mm]=indice
        pvlan[indice++]=m
    }
}

```

```

    }
  }
  for (m in VLAN)
    if (m in DIREC) delete VLAN[m]
  for (m in VLAN) {
    if (m==1) continue
    nDesc[indice]=VLAN[m]
    for (mm in PUERTO)
      if (PUERTO[mm]==m) PUERTO_AUXILIAR[mm]=indice
    pvlan[indice++]=m
  }
  for (m in ndire) {
    print "group " m >> $2
    print "group " m " enrutador ip", ndire[m], nmask[m], rout >> $2
    print "group " m " description \"\" nDesc[m] \"\" >> $2
    printf ". "
  }
  for (m in VLAN) {
    if (m==1) continue
    print "group " m >> $2
    print "group " m " no enrutador ip" >> $2
    print "group " m " description \"\" nDesc[m] \"\" >> $2
  }
  print "Listo"
  printf "Asociacion de Puertos a Grupos . . . "
  for (m in PUERTO_AUXILIAR) {
    if (ACTI[m]==0) delete PUERTO_AUXILIAR[m]
  }
  for (m in PUERTO_AUXILIAR) {
    print "group " PUERTO_AUXILIAR[m], "interface " m >> $2
    printf ". "
  }
  print "Listo"
  printf "Configuracion Autonegociacion . . . "
  for (m in SPEED) {
    duplex="half"
    speed=10
    if (SPEED[m]=="speed-100") speed=100
    if (DUPLEX[m]=="full-duplex") duplex="full"
    printf ". "
    print "interface ethernet " m " speed " speed " duplex " duplex >>
$2
  }
  print "Listo"
  if (rout=="routing-off") {
    for (m in DIREC) {
      split(MASK[m],mascara,"\.")
      split(DIREC[m],d,"\.")
      ult_octet=(d[4]-d[4]%(256-mascara[4]))+1
      subr=(d[1] "." d[2] "." d[3] "." ult_octet)
      print "ip route 0.0.0.0 0.0.0.0 " subr >> $2
    }
  }

```

```

}
printf "Configuracion Spanning Tree . . . "
if (rout!="routing-off") print "bridge 2 priority      24576" >> $2
print "bridge 3 priority      28672" >> $2
if (rout=="routing-off") print "bridge 3 priority      29952" >> $2
}'
print "Listo"
printf "Configuracion SNMP . . . "
echo "snmp community read-only Dedalo" >> $nombre_archivo
echo "snmp community read-write Dedalo" >> $nombre_archivo
echo "snmp community trap Dedalo" >> $nombre_archivo
echo "snmp trap unicast" >> $nombre_archivo
echo "snmp station 10.1.4.151 ffffffff3:fffffe7f ffffffff:fffffff on 162
on special" >> $nombre_archivo
print "Listo"
pdiag=`nAWK '$1=="$1"' {print $3}' /export/home/oper2/ba/pass2.txt2`
padmi=`nAWK '$1=="$1"' {print $2}' /export/home/oper2/ba/pass2.txt2`
echo "password admin switch" $padmi >> $nombre_archivo
echo "password diag switch" $pdiag >> $nombre_archivo
$NF}'`
sleep 3

```

A.4 Procesamiento del evento “Nodo Caído”

```

#!/usr/bin/ksh
aoa=aortiz@bncr.fi.cr
ccc=ccruz@bncr.fi.cr
dmr=dmunoz@bncr.fi.cr
gmc=gmorac@bncr.fi.cr
jcg=jchacong@bncr.fi.cr
fgk=jgonzalezk@bncr.fi.cr
mvv=jvargasv@bncr.fi.cr
mho=mherrerao@bncr.fi.cr
vaa=vadanis@bncr.fi.cr
jsw=jsancho@bncr.fi.cr
#kef=kespinoza@bncr.fi.cr
cab=caguilarb@bncr.fi.cr
#rsb=rsaborio@bncr.fi.cr
averias="$jcg $fgk $mvv $cab $vaa $jsw $gmc $ccc"
pr=/opt/OV/bin/ovobjprint
event=/opt/OV/bin/ovevent
mp=/opt/OV/bin/ovtopodump
Objeto=`$mp $1 | nAWK '/'$1'/ {A=$3} END {print A}'`
Fecha=`date | nAWK 'BEGIN {FS=":"} {print $1}'`
if [ "$Objeto" != "" ]
then
Regional=`$pr -s $Objeto | nAWK 'BEGIN {FS="\\"} /sysContact/ {print $(NF-1)}'`
OID=`$mp $1 | nAWK 'FNR==2 {print $1}'`
Nombre=`$pr -s $Objeto | nAWK 'BEGIN {FS="(\\"| )"} /sysLocation/ {for (i=9;i<NF-2;i++) printf $i " "}'`
fi
Alarm=""
case $Regional in
"Dir. Reg. Guanacaste-Puntarenas") Alarm="Guanacaste - Puntarenas" ;;
"Dir. Reg. Alajuela") Alarm=Alajuela ;;
"Dir. Reg. Cartago-Sur") Alarm="Cartago - Sur" ;;
"Dir. Reg. Heredia-Limon") Alarm="Heredia - Limon" ;;
"Dir. Reg. San Jose Este") Alarm="San Jose Este" ;;
"Dir. Reg. San Jose Oeste") Alarm="San Jose Oeste" ;;
"Administracion del Backbone UMMR") Alarm="Backbone";;
UPS) Alarm="UPS";;
UMMR) Alarm=Enrutador ;;
esac
if [ "$Alarm" != "" ]
then
Perdida=`/usr/sbin/ping -s $Objeto 10 20 | nAWK 'BEGIN {FS="[,%]}"} %/
{print $3}'`
if [ $Perdida -gt 90 ]
then
a=.1.3.6.1.4.1.11.2.17.1.0.58916872
b=.1.3.6.1.4.1.11.2.17.2.1.0
c=.1.3.6.1.4.1.11.2.17.2.2.0

```

```
d=.1.3.6.1.4.1.11.2.17.2.4.0
$event -s Critical -c "$Alarm" "" $a $b Integer 14 $c OctetString
"$Objeto" $d OctetString "Caido Totalmente el equipo en $Nombre"
if ! `grep $OID /tmp/caidos > /dev/null`
then
#echo "Caido Totalmente el equipo en $Nombre Hora: " `date` | mailx -s
"Problemas con $Nombre" $averias
echo $OID >> /tmp/caidos
echo $OID >> /tmp/caidos2
fi
fi
fi
```

A.5 Procesamiento del evento “Enlace Caído”

```
#!/usr/bin/perl

#
# Este programa esta hecho para filtrar la alerta de
# Caído el enlace tal
# .1.3.6.1.6.3.1.1.5.3.1.3.6.1.4.1.18.3
# .1.3.6.1.6.3.1.1.5.3
#

sub conv_hor_sec($) {
    my $todo=$_[0];
    my @VEC_TODO=split(":",$todo);
    my $Resultado=3600*$VEC_TODO[0]+60*$VEC_TODO[1]+$VEC_TODO[2];
    return $Resultado;
}

sub determinar_if_index($$) {

    sub determinar_syscontact($) {
        my @CONTACTO = split(" ", $_[0]);
        @CONTACTO[0 .. 2]=("","","");
        my $RESULTADO = join(" ",@CONTACTO);
        $RESULTADO =~ s/ [ ]+//g;
        $RESULTADO =~ s/ \w+ \w+$/ /g if ($_[0] =~ "Location");
        return $RESULTADO;
    }

    my @VEC_INST=split("\n", $_[0]);
    my $accion = '/opt/OV/bin/ovobjprint -o ';
    foreach my $b (@VEC_INST) {
        $b =~ s/[ \s]+/ /g;
        my @VEC_INST_IND=split(" ", $b);
        next if ($VEC_INST_IND[0] !~ "/" );
        my @VEC_OID=split("/", $VEC_INST_IND[0]);
        my $c=`$accion $VEC_OID[1]`;
        my $e=`$accion $VEC_OID[0]`;
        my @VECTOR_INSTANC=split("\n", $e . $c);
        foreach my $d (@VECTOR_INSTANC) {
            $d =~ s/[ \s]+/ /g;
            $d =~ s/"/ /g;
            my @VALORES=split(" ", $d);
            $determinar_if_index_CONTACT =
determinar_syscontact($d) if ($d =~ "Contact");
            $determinar_if_index_LUGAR =
determinar_syscontact($d) if ($d =~ "Location");
            $determinar_if_index_STATUS = $VALORES[3] if ($d
=~ "Status");
```

```

        $determinar_if_index_DESCRIPCION = $VALORES[3] if
($d =~ "SNMP ifDescr");
        $determinar_if_index_INDEX = $VALORES[3] if ($d =~
"SNMP ifIndex");
    }
    last if ($determinar_if_index_INDEX eq $_[1]);
}
my @RESULTADO = ($determinar_if_index_STATUS,
$determinar_if_index_DESCRIPCION,
$determinar_if_index_CONTACT, $determinar_if_index_LUGAR);
return @RESULTADO;
}

#
# Los parametros se pasan por valor
#

$DATOS=<STDIN>;
chop($DATOS);

# Se hace la separacion de los datos

@VEC_DATOS=split(" ", $DATOS);
$IP=$VEC_DATOS[0];
$IF=$VEC_DATOS[1];
#$Puerto=$VEC_DATOS[2];
#$Tipo=$VEC_DATOS[3];

#
#
# Datos para generar la alarma en HP OpenView
#

$event='/opt/OV/bin/ovevent';

$a='.1.3.6.1.4.1.11.2.17.1.0.58916872';
$b='.1.3.6.1.4.1.11.2.17.2.1.0';
$c='.1.3.6.1.4.1.11.2.17.2.2.0';
$d='.1.3.6.1.4.1.11.2.17.2.4.0';

#
# Usuarios de correo
#

$aoa="aortiz@bncr.fi.cr";
$ccc="ccruz@bncr.fi.cr";
$dmr="dmunoz@bncr.fi.cr";
$gmc="gmorac@bncr.fi.cr";
$jcg="jchacong@bncr.fi.cr";
$fgk="jgonzalezk@bncr.fi.cr";

```

```

$mvv="jvargasv\@bncr.fi.cr";
#$mho="mherrerao\@bncr.fi.cr";
$vaa="vadanis\@bncr.fi.cr";
$jsw="jsancho\@bncr.fi.cr";
$kef="kespinoza\@bncr.fi.cr";
$cab="caguilarb\@bncr.fi.cr";
#$rsb="rsaborio\@bncr.fi.cr";
@averias=($jcg, $fgk, $mvv, $cab, $vaa, $jsw, $gmc, $ccc);

#
#Logica del negocio
#

# Sacar nombre y OID a partir de la direccion IP

$NOMBRE_OID=`/opt/OV/bin/ovtopodump $IP`;
chop($NOMBRE_OID);

@VEC_NOMBRE_OID_STATUS=split("\n",$NOMBRE_OID);
$VEC_NOMBRE_OID_STATUS[1] =~ s/[\s]+/ /g;
@VEC_NOMBRE=split(" ",$VEC_NOMBRE_OID_STATUS[1]);

@VEC_OID=split("/", $VEC_NOMBRE[0]);

$oid=$VEC_OID[0];
$Nombre=$VEC_NOMBRE[2];
$Status=$VEC_NOMBRE[3];
$IP=$VEC_NOMBRE[4];
#print $Status;
exit if ($Status eq "Unmanaged");

$Nombre = $IP if ($Nombre eq "");

$NOMBRE_OID=`/opt/OV/bin/ovtopodump -o $oid`;
chop($NOMBRE_OID);

#
# Aqui va la funcion para ver si determinar la instancia del Ifindex y
# si es unmanaged
#
#

($IF_Status, $IF_Descr, $Dir_Reg, $Lugar)=
determinar_if_index($NOMBRE_OID,$IF);

exit if ($IF_Status =~ "Unmanag");

#
# Determinacion de la categoria en el HP
#

```



```

$Alarm{"Dir. Reg. Guanacaste-Puntarenas"} = "Guanacaste - Puntarenas";
$Alarm{"Dir. Reg. Alajuela"} = "Alajuela";
$Alarm{"Dir. Reg. Cartago-Sur"} = "Cartago - Sur";
$Alarm{"Dir. Reg. Heredia-Limon"} = "Heredia - Limon";
$Alarm{"Dir. Reg. San Jose Este"} = "San Jose Este";
$Alarm{"Dir. Reg. San Jose Oeste"} = "San Jose Oeste";
$Alarm{"Administracion del Backbone UMMR"} = "Backbone";
$Alarm{"UMMR"} = "Enrutador";

```

```

$TOT="/tmp/" . $IF . "_" . $oid . "_LINK_D_hora_enable";
$archivo_horas=$TOT;
$archivo_reporte='/tmp/LINK_D_reporte_enable';

```

```

open(REPORTE,$archivo_reporte);

```

```

$cont_reporte=<REPORTE>;
chop($cont_reporte);

```

```

$OID = $oid . "_" . $IF;

```

```

while ($cont_reporte) {
    if ($OID eq $cont_reporte) {
        $k=`ls $archivo_horas`;
        chop($k);
        system("rm $archivo_horas") if ($k);
        exit;
    }
    $cont_reporte=<REPORTE>;
    chop($cont_reporte);
}

```

```

$fecha=`date`;
chop($fecha);
close(REPORTE);
@VEC_HORA=split(" ", $fecha);
$hora_actual=$VEC_HORA[3];

```

```

open(HORAS, $archivo_horas);

```

```

$cont_horas=<HORAS>;
chop($cont_horas);

```

```

$no_encontrado=1;

```

```

while ($cont_horas) {
    @VEC_CONT_HORAS=split(" ", $cont_horas);
    if (!($VEC_CONT_HORAS[0] eq $oid)) {

```

```

        $cont_horas=<HORAS>;
        chop($cont_horas);
        next;
    }
    else {
        $hora_anterior=$VEC_CONT_HORAS[1];
        $no_encontrado=0;
        last;
    }
}

close(HORAS);

if ($no_encontrado) {
    open(HORAS,">>$archivo_horas");
    print HORAS "$oid $hora_actual\n";
    $hora_anterior=$hora_actual;
    close(HORAS);

    system("$event -s Warning -c \"\$Alarm{$Dir_Reg}\" \"\" $a $b
Integer 14 $c OctetString \"\$Nombre\" $d OctetString \"Caído enlace
$IF_Descr en $Lugar\"");

    $OID="Caído enlace " . $IF_Descr . " en " . $Lugar;
#    system("echo $OID | mailx -s \"Servicio en Initial\" $cab");

#    system("sleep 80; rm $archivo_horas");
}

$diferencia=conv_hor_sec($hora_actual)-conv_hor_sec($hora_anterior);

if ($diferencia>25 && $diferencia<45) {

#    Se cumplen todas las condiciones para que de la alarma y
#    guarde en el archivo para que no la vuelva a dar en todo el día
#

    system("echo $OID >> $archivo_reporte");

#    system("$event -s Warning -c $Alarm{$Dir_Reg} \"\" $a $b Integer
14 $c OctetString \"\$Nombre\" $d OctetString \"Favor Revisar el switch
$Nombre ($IP) ya que el puerto $Slot/$Puerto presenta sintomas de
servicios en Initial\"");

#    $OID="Favor Revisar el switch " . $Nombre . "\\(" . $IP . "\\) ya
que el puerto " . $Slot . "/" . $Puerto . " presenta sintomas de
servicios en Initial";
#    system("echo $OID | mailx -s \"Servicio en Initial\" $cab");
}

```

```
}  
  
if ($diferencia > 180) {  
    system("$event -s Warning -c \"\$Alarm{$Dir_Reg}\" \"\" $a $b  
Integer 14 $c OctetString \"\$Nombre\" $d OctetString \"Caído enlace  
$IF_Descr en $Lugar\"");  
    system("rm $archivo_horas");  
}
```

A.6 Procesamiento del evento "ATM Initial"

```

#!/usr/bin/perl

# Nombre Archivo: Health.pl

#
# Este programa esta hecho para filtrar la alerta de
# Physical Logical or virtual port was enable generada por el trap
# .1.3.6.1.4.1.800.3.1.1.2.0.8
#

sub conv_hor_sec($) {
    my $todo=$_[0];
    my @VEC_TODO=split(":",$todo);
    my $Resultado=3600*$VEC_TODO[0]+60*$VEC_TODO[1]+$VEC_TODO[2];
    return $Resultado;
}

#
#Los parametros se pasan por valor
#

$DATOS=<STDIN>;
chop($DATOS);

#Se hace la separacion de los datos

@VEC_DATOS=split(" ",$DATOS);
$IP=$VEC_DATOS[0];
$Slot=$VEC_DATOS[1];
$Puerto=$VEC_DATOS[2];
$Tipo=$VEC_DATOS[3];

#
#
# Datos para generar la alarma en HP OpenView
#

$event='/opt/OV/bin/ovevent';

$a='.1.3.6.1.4.1.11.2.17.1.0.58916872';
$b='.1.3.6.1.4.1.11.2.17.2.1.0';
$c='.1.3.6.1.4.1.11.2.17.2.2.0';
$d='.1.3.6.1.4.1.11.2.17.2.4.0';

#
# Usuarios de correo
#

```

```

$aoa="aortiz\@bncr.fi.cr";
$ccc="ccruz\@bncr.fi.cr";
#$dmr="dmunoz\@bncr.fi.cr";
$gmc="gmorac\@bncr.fi.cr";
$jcg="jchacong\@bncr.fi.cr";
$fgk="jgonzalezk\@bncr.fi.cr";
$mvv="jvargasv\@bncr.fi.cr";
#$mho="mherrerao\@bncr.fi.cr";
$vaa="vadanis\@bncr.fi.cr";
$jsw="jsancho\@bncr.fi.cr";
$kef="kespinoza\@bncr.fi.cr";
$cab="caguilarb\@bncr.fi.cr";
#$rsb="rsaborio\@bncr.fi.cr";
@averias=( $jcg, $fgk, $mvv, $cab, $vaa, $jsw, $gmc, $ccc);

```

```

#
#Logica del negocio
#

# Sacar nombre y OID a partir de la direccion IP

$NOMBRE_OID=`/opt/OV/bin/ovtopodump $IP`;
chop($NOMBRE_OID);

@VEC_NOMBRE_OID_STATUS=split("\n",$NOMBRE_OID);
$VEC_NOMBRE_OID_STATUS[1] =~ s/[\s]+/ /g;
@VEC_NOMBRE=split(" ", $VEC_NOMBRE_OID_STATUS[1]);

@VEC_OID=split("/", $VEC_NOMBRE[0]);

$oid=$VEC_OID[0];
$Nombre=$VEC_NOMBRE[2];
$Status=$VEC_NOMBRE[3];
$IP=$VEC_NOMBRE[4];
#print $Status;
exit if ($Status eq "Unmanaged");

$TOT="/tmp/" . $Slot . "_" . $Puerto . "_" . $oid . "_hora_enable";
$arquivo_horas="$TOT";
$arquivo_reporte='/tmp/reportes_enable';

open(REPORTE,$arquivo_reporte);

$cont_reporte=<REPORTE>;
chop($cont_reporte);
$OID = $oid . "_" . $Slot . "_" . $Puerto;
while ($cont_reporte) {
    if ($OID eq $cont_reporte) {
        $k=`ls $arquivo_horas`;
    }
}

```

```

        chop($k);
        system("rm $archivo_horas") if ($k);
        exit;
    }
    $cont_reporte=<REPORTE>;
    chop($cont_reporte);
}

$fecha=`date`;
chop($fecha);
close(REPORTE);
@VEC_HORA=split(" ", $fecha);
$hora_actual=$VEC_HORA[3];

open(HORAS, $archivo_horas);

$cont_horas=<HORAS>;
chop($cont_horas);

$no_encontrado=1;

while ($cont_horas) {
    @VEC_CONT_HORAS=split(" ", $cont_horas);
    if (!(($VEC_CONT_HORAS[0] eq $oid)) {
        $cont_horas=<HORAS>;
        chop($cont_horas);
        next;
    }
    else {
        $hora_anterior=$VEC_CONT_HORAS[1];
        $no_encontrado=0;
        last;
    }
}

close(HORAS);

if ($no_encontrado) {
    open(HORAS, ">>$archivo_horas");
    print HORAS "$oid $hora_actual\n";
    $hora_anterior=$hora_actual;
    close(HORAS);
}

#$hora_anterior=$hora_actual if (!(($hora_anterior));

$diferencia=conv_hor_sec($hora_actual)-conv_hor_sec($hora_anterior);

if ($diferencia>8 && $diferencia<21) {

#       Se cumplen todas las condiciones para que de la alarma y

```

```

#         guarde en el archivo para que no la vuelva a dar en todo el dia
#
        system("echo $OID >> $archivo_reporte");
        system("$event -s Warning -c Backbone \"\" $a $b Integer 14 $c
OctetString \"Nombre\" $d OctetString \"Favor Revisar el switch $Nombre
($IP) ya que el puerto $Slot/$Puerto presenta sintomas de servicios en
Initial\");

        $OID="Favor Revisar el switch " . $Nombre . "\"(\" . $IP ."\" ya
que el puerto " . $Slot . "/" . $Puerto . " presenta sintomas de
servicios en Initial";
        system("echo $OID | mailx -s \"Servicio en Initial\" $cab $kef");

}

if ($diferencia > 180) {
#         system("$event -s Warning -c \"$Alarm{$Dir_Reg}\" \"\" $a $b
Integer 14 $c OctetString \"Nombre\" $d OctetString \"Caído enlace
$IF_Descr en $Lugar\");
        system("rm $archivo_horas");
}

```

A.7 Procesamiento de los *traps* de salud

```
#!/usr/bin/perl

#
# Este programa esta hecho para filtrar la alerta de
# HealthThreshold Device
# .1.3.6.1.4.1.800.3.1.1.0.39
#
#

#
#Los parametros se pasan por valor
#

$DATOS=<STDIN>;
chomp($DATOS);

#Se hace la separacion de los datos

@VEC_DATOS=split(" ", $DATOS);

$IP=$VEC_DATOS[0];
$Threshold=$VEC_DATOS[1];
$Tipo_Dev=join(" ", @VEC_DATOS[4 .. $#VEC_DATOS]);
$Tipo_Dev=$Threshold . " " . $Tipo_Dev;

#
#Logica del negocio
#

# Sacar nombre y OID a partir de la direccion IP

$NOMBRE_OID=`/opt/OV/bin/ovtopodump $IP`;
chop($NOMBRE_OID);

@VEC_NOMBRE_OID_STATUS=split("\n", $NOMBRE_OID);
$VEC_NOMBRE_OID_STATUS[1] =~ s/[ \s]+/ /g;
@VEC_NOMBRE=split(" ", $VEC_NOMBRE_OID_STATUS[1]);

@VEC_OID=split("/", $VEC_NOMBRE[0]);

$oid=$VEC_OID[0];
$Nombre=$VEC_NOMBRE[2];
$Status=$VEC_NOMBRE[3];
#$IP=$VEC_NOMBRE[4];
#print $Status;
```



```

exit if ($Status eq "Unmanaged");

$Nombre = $IP if (!(($VEC_NOMBRE_OID_STATUS[1]));

#print "$Nombre $Status - $#VEC_NOMBRE";

system("\(echo $Nombre; echo $Tipo_Dev\) | /usr/bin/HP_Ag/Ejec_Health.pl
&");

```

```

#!/usr/bin/perl

# Nombre del archivo : Ejec_health.pl
#
# Subrutinas
#
#

sub Invertir {

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
}

```

```

#
#
# Funcion para generar hash Dobles
#
#
#

```

```

sub Genera_Hash_Doble($$$$) {

```

```

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Repeticiones2) = $_[2];
my ($Tipo) = $_[3];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {
    delete $DESCVLan_2{$re};
}

my @APORTStatus=split("\n",$DDesc);

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    my $inst_raw=shift(@B);

    my $Valor=pop(@B);

    my @C=split("[\.]",$inst_raw);

    for (my $i=1; $i<$Repeticiones1; $i++) {
        shift(@C);
    }

    for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--
) {
        pop(@C);
    }

    my $inst=join($Tipo,@C);

    $DESCVLan_2{$inst}=$Valor;
}

return %DESCVLan_2;
}

#
#

sub Valor_M($) {

    return $Rx{0}          if ($_[0] == 1);
    return $RxTx{0}       if ($_[0] == 2);
    return $Backplane{0}  if ($_[0] == 3);
    return $Vcc{0}        if ($_[0] == 4);
    return $Hre_Cam{0}    if ($_[0] == 5);
}

```

```

        return $Mpm_Cam{0}      if ($_[0] == 6);
        return $Memory{0}      if ($_[0] == 7);
        return $CPU{1}         if ($_[0] == 8);
        return $CPU{2}         if ($_[0] == 9);
        return $CPU{3}         if ($_[0] eq "0A");
        return $CPU{4}         if ($_[0] eq "0B");
        return $Hre_Coll{0}    if ($_[0] eq "0C");

    }

# Aqui terminan las subrutinas

#
#
# Inicio del programa principal
#
#

system("sleep 61");

$IP=<STDIN>;

chomp($IP);

$Tipo_Al=<STDIN>;

chomp($Tipo_Al);

$Tipo_Al =~ tr/[a-z]/[A-Z]/;

#
#
# Tipo de Alerta que esta entregando
#

$Mensaje{"01"} = "Excedido Threshold Rx";
$Mensaje{"02"} = "Excedido Threshold Rx-Tx";
$Mensaje{"03"} = "Excedido Threshold del Backplane";
$Mensaje{"04"} = "Excedido Threshold del VCC";
$Mensaje{"05"} = "Excedido Threshold de la CAM del HRE";
$Mensaje{"06"} = "Excedido Threshold de la CAM de la MPM";
$Mensaje{"07"} = "Excedido Threshold de la Memoria";
$Mensaje{"08"} = "Excedido Threshold del CPU 1";
$Mensaje{"09"} = "Excedido Threshold del CPU 2";
$Mensaje{"0A"} = "Excedido Threshold del CPU 3";
$Mensaje{"0B"} = "Excedido Threshold del CPU 4";

```

```
$Mensaje{"0C"} = "Excedido Threshold de Colisiones del HRE";
```

```
$Query{"01"} = "health2DeviceRx1MinAvg";
$Query{"02"} = "health2DeviceRxTx1MinAvg";
$Query{"03"} = "health2DeviceBackplane1MinAvg";
$Query{"04"} = "health2DeviceVcc1MinAvg";
$Query{"05"} = "health2DeviceHreCam1MinAvg";
$Query{"06"} = "health2DeviceMpmCam1MinAvg";
$Query{"07"} = "health2DeviceMemory1MinAvg";
$Query{"08"} = "health2DeviceCpulMinAvg";
$Query{"09"} = "health2DeviceCpulMinAvg";
$Query{"0A"} = "health2DeviceCpulMinAvg";
$Query{"0B"} = "health2DeviceCpulMinAvg";
$Query{"0C"} = "health2DeviceHreCollision1MinAvg";
```

```
#
# Datos para generar la alarma en HP OpenView
#
```

```
$event="/opt/OV/bin/ovevent";
```

```
$a='.1.3.6.1.4.1.11.2.17.1.0.58916872';
$b='.1.3.6.1.4.1.11.2.17.2.1.0';
$c='.1.3.6.1.4.1.11.2.17.2.2.0';
$d='.1.3.6.1.4.1.11.2.17.2.4.0';
```

```
#
# Usuarios de correo
#
```

```
$aoa="aortiz@bncr.fi.cr";
$ccc="ccruz@bncr.fi.cr";
#$cgb="cgomez@bncr.fi.cr";
#$dmr="dmunoz@bncr.fi.cr";
$gmc="gmorac@bncr.fi.cr";
$jcg="jchacong@bncr.fi.cr";
$fgk="jgonzalezk@bncr.fi.cr";
$mvv="jvargasv@bncr.fi.cr";
#$mho="mherrerao@bncr.fi.cr";
$vaa="vadanis@bncr.fi.cr";
$jsw="jsancho@bncr.fi.cr";
$kef="kespinoza@bncr.fi.cr";
$cab="caguilarb@bncr.fi.cr";
#$rsb="rsaborio@bncr.fi.cr";
@averias=($jcg, $fgk, $mvv, $cab, $vaa, $jsw, $gmc, $ccc);
```

```

@Alarmas = split(" ", $Tipo_Al);

for ($i=1; $i< $#Alarmas; $i+=2) {

    if ($Alarmas[$i+1] == 2) {
        push(@uma_guma, $Query{$Alarmas[$i]});
        open(REP_LOG, '>>/tmp/Threshold');
        print REP_LOG "$IP $Alarmas[$i]\n";
        close(REP_LOG);
        $Alarma{$Alarmas[$i]}=1;
    }

}

#
#
# Carga de la informacion del equipo
#
#

exit if (!$suma_guma[0]);

$MISC=`/opt/OV/bin/snmpwalk $IP @uma_guma`;
@APM=split("\n", $MISC);

$encontrado0=0;

foreach $b (@APM) {
    if ($b =~ "health2DeviceRx1MinAvg" && $b =~ "INTEGER") {
        $health2DeviceRx1MinAvg="$health2DeviceRx1MinAvg" . "\n"
    }
    if ($b =~ "health2DeviceRxTx1MinAvg" && $b =~ "INTEGER") {
        $health2DeviceRxTx1MinAvg="$health2DeviceRxTx1MinAvg" .
    }
    if ($b =~ "health2DeviceBackplane1MinAvg" && $b =~ "INTEGER") {
        $health2DeviceBackplane1MinAvg="$health2DeviceBackplane1MinAvg" . "\n" .
    }
    if ($b =~ "health2DeviceVcc1MinAvg" && $b =~ "INTEGER") {
        $health2DeviceVcc1MinAvg="$health2DeviceVcc1MinAvg" .
    }
    if ($b =~ "health2DeviceMpmCam1MinAvg" && $b =~ "INTEGER") {
        $health2DeviceMpmCam1MinAvg="$health2DeviceMpmCam1MinAvg"
    }
    if ($b =~ "health2DeviceHreCam1MinAvg" && $b =~ "INTEGER") {

```

```

        $health2DeviceHreCamlMinAvg="$health2DeviceHreCamlMinAvg"
. "\n" . "$b";
    }
    if ($b =~ "health2DeviceMemorylMinAvg" && $b =~ "INTEGER") {
        $health2DeviceMemorylMinAvg="$health2DeviceMemorylMinAvg"
. "\n" . "$b";
    }
    if ($b =~ "health2DeviceCpulMinAvg" && $b =~ "INTEGER") {
        $health2DeviceCpulMinAvg=$health2DeviceCpulMinAvg. "\n"
. "$b";
    }
    if ($b =~ "health2DeviceHreCollisionlMinAvg" && $b =~ "INTEGER")
{
    $health2DeviceHreCollisionlMinAvg=$health2DeviceHreCollisionlMinA
vg. "\n" . "$b";
    }
}

%Rx =          Genera_Hash_Doble($health2DeviceRxlMinAvg,6,6, ".") if
($Alarma{"01"});
%RxTx =        Genera_Hash_Doble($health2DeviceRxTxlMinAvg,6,6, ".") if
($Alarma{"02"});
%Backplane =   Genera_Hash_Doble($health2DeviceBackplanelMinAvg,6,6, ".")
if ($Alarma{"03"});
%Vcc =         Genera_Hash_Doble($health2DeviceVcclMinAvg,6,6, ".") if
($Alarma{"04"});
%Hre_Cam =     Genera_Hash_Doble($health2DeviceHreCamlMinAvg,6,6, ".") if
($Alarma{"05"});
%Mpm_Cam =     Genera_Hash_Doble($health2DeviceMpmCamlMinAvg,6,6, ".") if
($Alarma{"06"});
%Memory =      Genera_Hash_Doble($health2DeviceMemorylMinAvg,6,6, ".") if
($Alarma{"07"});
%CPU =         Genera_Hash_Doble($health2DeviceCpulMinAvg,8,8, ".") if
($Alarma{"08"} || $Alarma{"09"} || $Alarma{"0A"} || $Alarma{"0B"});
%Hre_Coll =    Genera_Hash_Doble($health2DeviceHreCollisionlMinAvg,6,6, ".") if
($Alarma{"0C"});

$destino = "Backbone";

$destino = "WAN1" if ($IP =~ "WAN");

@AAAA=keys(%Alarma);

foreach $e (@AAAA) {
    if (Valor_M($e)>$Alarmas[0]) {
        $Valor_T=Valor_M($e);
    }
}

```

```

                system("$event -s Warning -c \"\$destino\" \"\$\" \$a \$b
Integer 14 $c OctetString \"\$IP\" $d OctetString \"\$Mensaje{$e}. Ultimo
Minuto=$Valor_T%\");
    }
}

```

A.8 Procesamiento de los *traps* de autenticación

```

#! /usr/bin/perl

# Nombre del archivo: AuthenFail.pl

#
# Este programa esta hecho para filtrar la alerta de
# Authentication Failure generada por el trap
# .1.3.6.1.6.1.1.5.5
#

sub conv_hor_sec($) {
    my $todo=$_[0];
    my @VEC_TODO=split(":",$todo);
    my $Resultado=3600*$VEC_TODO[0]+60*$VEC_TODO[1]+$VEC_TODO[2];
    return $Resultado;
}

#
#Los parametros se pasan por valor
#

$DATOS=<STDIN>;
chop($DATOS);

#Se hace la separacion de los datos

@VEC_DATOS=split(" ",$DATOS);
$IP=$VEC_DATOS[0];
$Slot=$VEC_DATOS[1];
$Puerto=$VEC_DATOS[2];
$Tipo=$VEC_DATOS[3];

#
#
# Datos para generar la alarma en HP OpenView
#

$event='/opt/OV/bin/ovevent';

```

```

$a='.1.3.6.1.4.1.11.2.17.1.0.58916872';
$b='.1.3.6.1.4.1.11.2.17.2.1.0';
$c='.1.3.6.1.4.1.11.2.17.2.2.0';
$d='.1.3.6.1.4.1.11.2.17.2.4.0';

#
# Usuarios de correo
#

$aoa="aortiz\@bncr.fi.cr";
$ccc="ccruz\@bncr.fi.cr";
#$cgb="cgomez\@bncr.fi.cr";
#$dmr="dmunoz\@bncr.fi.cr";
$gmc="gmorac\@bncr.fi.cr";
$jcg="jchacong\@bncr.fi.cr";
$fgk="jgonzalezk\@bncr.fi.cr";
$mvv="jvargasv\@bncr.fi.cr";
#$mho="mherrerao\@bncr.fi.cr";
$vaa="vadanis\@bncr.fi.cr";
$jsw="jsancho\@bncr.fi.cr";
$kef="kespinoza\@bncr.fi.cr";
$cab="caguilarb\@bncr.fi.cr";
#$rsb="rsaborio\@bncr.fi.cr";
@averias=( $jcg, $fgk, $mvv, $cab, $vaa, $jsw, $gmc, $ccc);

#
#Logica del negocio
#

# Sacar nombre y OID a partir de la direccion IP

$NOMBRE_OID=`/opt/OV/bin/ovtopodump $IP`;
chop($NOMBRE_OID);

@VEC_NOMBRE_OID_STATUS=split("\n",$NOMBRE_OID);
$VEC_NOMBRE_OID_STATUS[1] =~ s/[\s]+/ /g;
@VEC_NOMBRE=split(" ", $VEC_NOMBRE_OID_STATUS[1]);

@VEC_OID=split("/", $VEC_NOMBRE[0]);

$oid=$VEC_OID[0];
$Nombre=$VEC_NOMBRE[2];
$Status=$VEC_NOMBRE[3];
$IP=$VEC_NOMBRE[4];
exit if ($Status eq "Unmanaged");

$TOT="/tmp/" . $oid . "_hora_authF";
$archivo_horas="$TOT";
$archivo_reporte='/tmp/report_e_authF';

```



```

open(REPORTE,$archivo_reporte);

$cont_reporte=<REPORTE>;
chop($cont_reporte);
$OID = $oid;
while ($cont_reporte) {
    if ($OID eq $cont_reporte) {
        $k=`ls $archivo_horas`;
        chop($k);
        system("rm $archivo_horas") if ($k);
        exit;
    }
    $cont_reporte=<REPORTE>;
    chop($cont_reporte);
}

$fecha=`date`;
chop($fecha);
close(REPORTE);
@VEC_HORA=split(" ",$fecha);
$hora_actual=$VEC_HORA[3];

open(HORAS, $archivo_horas);

$cont_horas=<HORAS>;
chop($cont_horas);

$no_encontrado=1;

while ($cont_horas) {
    @VEC_CONT_HORAS=split(" ",$cont_horas);
    if (!(($VEC_CONT_HORAS[0] eq $oid)) {
        $cont_horas=<HORAS>;
        chop($cont_horas);
        next;
    }
    else {
        $hora_anterior=$VEC_CONT_HORAS[1];
        $no_encontrado=0;
        last;
    }
}

close(HORAS);

if ($no_encontrado) {
    open(HORAS,">>$archivo_horas");
    print HORAS "$oid $hora_actual\n";
    $hora_anterior=$hora_actual;
    close(HORAS);
}

```

```

#$hora_anterior=$hora_actual if (!( $hora_anterior));

$diferencia=conv_hor_sec($hora_actual)-conv_hor_sec($hora_anterior);

if ($diferencia<9) (

#       Se cumplen todas las condiciones para que de la alarma y
#       guarde en el archivo para que no la vuelva a dar en todo el dia
#
#       print $oid;
#       system("echo $OID >> $archivo_reporte");
#       system("$event -s Warning -c Backbone \"\" $a $b Integer 14 $c
OctetString \"$Nombre\" $d OctetString \"Favor Revisar el switch $Nombre
($IP) ya que Presenta Falla de Autenticacion\");

#       $OID="Favor Revisar el switch " . $Nombre . "\"(\" . $IP ."\") ya
que presenta Falla de Autenticacion";
#       system("echo $OID | mailx -s \"Autenticacion\" $cab");

}

if ($diferencia > 360) {

#       system("rm $archivo_horas");

}

```

A.9 Procesamiento del *trap* de cambio de grupo

```
#!/usr/bin/nAWK -f
{
# Nombre Archivo: Eliminación_Delete

# $1 es la IP fuente del trap
# $2 es el tipo (create o delete)
if ($2=="delete") {
B_OBJETO="/opt/OV/bin/ovtopodump " $1
B_OBJETO | getline OBJ_DB
B_OBJETO | getline OBJ_DB
split(OBJ_DB,NOM)
close(B_OBJETO)
Nombre=NOM[3]
if (Nombre=="OBJECT") Alarm="Enrutador"
if (Nombre=="OBJECT") Nombre=$1
Alarm="Backbone"
a=".1.3.6.1.4.1.11.2.17.1.0.58916872"
b=".1.3.6.1.4.1.11.2.17.2.1.0"
c=".1.3.6.1.4.1.11.2.17.2.2.0"
d=".1.3.6.1.4.1.11.2.17.2.4.0"
print "/opt/OV/bin/ovevent -s Normal -c \"\" Alarm \"\ " \", a, b,
\"Integer 14\", c, \"Octetstring \"\" Nombre \"\", d, \"Octetstring \"Puerto\",
$3 \"/" $4 \" cambio de grupo\"\" | \"/bin/ksh"}
}
```

A.10 Generación de Reportes

```
#!/usr/bin/perl

# Nombre: Reporte.pl

#
# Subrutinas
#
#

sub Invertir {

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVlan_2{$Lucas{$b}}=$b;
    }
    return %DESCVlan_2;
}

sub determinar_if_index($$) {

    sub determinar_syscontact($) {
        my @CONTACTO = split(" ",$_[0]);
        @CONTACTO[0 .. 2]=("","","");
        my $RESULTADO = join(" ",@CONTACTO);
        $RESULTADO =~ s/ [ ]+//g;
        $RESULTADO =~ s/ \w+ \w+$//g if ($_[0] =~ "Location");
        return $RESULTADO;
    }

    my @VEC_INST=split("\n",$_[0]);
    my $accion = '/opt/OV/bin/ovobjprint -o ';
    foreach my $b (@VEC_INST) {
        $b =~ s/[\s]+//g;
        my @VEC_INST_IND=split(" ",$b);
        next if ($VEC_INST_IND[0] !~ "/");
        my @VEC_OID=split("/", $VEC_INST_IND[0]);
        my $c=`$accion $VEC_OID[1]`;
        my $e=`$accion $VEC_OID[0]`;
    }
}
```

```

        my @VECTOR_INSTANC=split("\n",$e . $c);
        foreach my $d (@VECTOR_INSTANC) {
            $d =~ s/[\\s]+//g;
            $d =~ s/\\/\\/g;
            my @VALORES=split(" ",$d);
            $determinar_if_index_CONTACT =
determinar_syscontact($d) if ($d =~ "Contact");
            $determinar_if_index_LUGAR =
determinar_syscontact($d) if ($d =~ "Location");
            $determinar_if_index_STATUS = $VALORES[3] if ($d
=~ "Status");
            $determinar_if_index_DESCRIPCION = $VALORES[3] if
($d =~ "SNMP ifDescr");
            $determinar_if_index_INDEX = $VALORES[3] if ($d =~
"SNMP ifIndex");
        }
        last if ($determinar_if_index_INDEX eq $_[1]);
    }
    my @RESULTADO = ($determinar_if_index_STATUS,
$determinar_if_index_DESCRIPCION,
$determinar_if_index_CONTACT, $determinar_if_index_LUGAR);
    return @RESULTADO;
}

# Aqui terminan las subrutinas

#
#
# Inicio del programa principal
#
#
#
# Tipo de Alerta que esta entregando
#
#
$Mensaje{"01"} = "Threshold Rx";
$Mensaje{"02"} = "Threshold Rx-Tx";
$Mensaje{"03"} = "Threshold del Backplane";
$Mensaje{"04"} = "Threshold del VCC";
$Mensaje{"05"} = "Threshold de la CAM del HRE";
$Mensaje{"06"} = "Threshold de la CAM de la MPM";
$Mensaje{"07"} = "Threshold de la Memoria";
$Mensaje{"08"} = "Threshold del CPU 1";
$Mensaje{"09"} = "Threshold del CPU 2";
$Mensaje{"0A"} = "Threshold del CPU 3";
$Mensaje{"0B"} = "Threshold del CPU 4";
$Mensaje{"0C"} = "Threshold de Colisiones del HRE";

```

```

$Max{"01"} = 0;
$Max{"02"} = 0;
$Max{"03"} = 0;
$Max{"04"} = 0;
$Max{"05"} = 0;
$Max{"06"} = 0;
$Max{"07"} = 0;
$Max{"08"} = 0;
$Max{"09"} = 0;
$Max{"0A"} = 0;
$Max{"0B"} = 0;
$Max{"0C"} = 0;

#
# Usuarios de correo
#

$aoa="aortiz@bncr.fi.cr";
$ccc="ccruz@bncr.fi.cr";
$gmc="gmorac@bncr.fi.cr";
$jcg="jchacong@bncr.fi.cr";
$fgk="jgonzalezk@bncr.fi.cr";
$mvv="jvargasv@bncr.fi.cr";
$vaa="vadanis@bncr.fi.cr";
$jsw="jsancho@bncr.fi.cr";
$kef="kespinoza@bncr.fi.cr";
$cab="caguilarb@bncr.fi.cr";
@averias=($jcg, $fgk, $mvv, $cab, $vaa, $jsw, $gmc, $ccc);

open(SALUD, '/tmp/Threshold');

$Salud=<SALUD>;
chomp($Salud);

while ($Salud) {
    @VEC_SA=split(" ", $Salud);
    $var_SA{"$VEC_SA[0]-$VEC_SA[1]"}++;
    $DIF=$var_SA{"$VEC_SA[0]-$VEC_SA[1]"}-$Max{"$VEC_SA[1]"};
    $Max{"$VEC_SA[1]"}=$var_SA{"$VEC_SA[0]-$VEC_SA[1]"} if ($DIF>0);
    $Max_el{"$VEC_SA[1]"}=$VEC_SA[0] if ($DIF>0);
    $Salud=<SALUD>;
    chomp($Salud);
}

close(SALUD);

foreach $b (keys(%Max)) {
    next if ($Max{$b}==0);

```

```

    $MENSAJE1 = $MENSAJE1 . "Para $Mensaje{$b} el maximo lo presenta
    $Max_el{$b} con $Max{$b}\n";
}

```

```

open(CAIDOS, '/tmp/caidos2');

```

```

$Caidos=<CAIDOS>;
chomp($Caidos);
$Max_c=0;

```

```

while ($Caidos) {
    $DIF=+$Caido{$Caidos}-$Max_c;
    $Max_c=$Caido{$Caidos} if ($DIF>0);
    $Max_ce=$Caidos if ($DIF>0);
    $Caidos=<CAIDOS>;
    chomp($Caidos);
}

```

```

close($Caidos);

```

```

$NOMBRE_OID=`/opt/OV/bin/ovtopodump -o $Max_ce`;

```

```

#
# Aqui va la funcion para ver si determinar la instancia del Ifindex y
# si es unmanaged
#
#

```

```

($IF_Status, $IF_Descr, $Dir_Reg, $Lugar)=
determinar_if_index($NOMBRE_OID,1);

```

```

$MENSAJE2 = "\n\n$Lugar es quien mas ha caido con $Max_c\n";

```

```

$MENSAJE = $MENSAJE1 . $MENSAJE2;

```

```

open(REPO, '>/tmp/Reporte_');

```

```

print REPO $MENSAJE;

```

```

close(REPO);

```

```

system("mailx -s \"Reporte Semanal\" $cab < /tmp/Reporte_");

```

```

system("rm /tmp/caidos2");
system("rm /tmp/Threshold");

```

ANEXO B. ACCIONES NO AUTOMÁTICAS.

Cualquier copia modificación o distribución de todos los programas del Anexo B deben hacerse bajo los términos y condiciones de la licencia GPL descritos en <http://www.gnu.org/licenses/gpl.html>.

B.1 Página inicial de Conmutadores Alcatel

Switch:

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<html>\n";
print "<head>\n";
print "<script type=\"text/javascript\">\n";
print "function openwindow()\n";
print "{\n";
print "}\n";
print "</script>\n";
print "</head>\n";
print "<body background=\"/OvDocs/images/Globe.gif\">\n";
print "<TITLE>Mantenimiento y Monitoreo de la Red</TITLE>\n";
print "  <FORM METHOD=POST
ACTION=\"http://10.1.4.151:8880/OvCgi/aut3.pl\" TARGET=\"_blank\" >\n";
print "<table width=\"900\" border=\"0\" cellspacing=\"0\"
cellpadding=\"0\">";
print "<tr>\n ";
print "<td width=\"170\" height=\"50\">&nbsp;</td>\n";
print "<td width=\"170\" height=\"50\">&nbsp;</td>\n";
print "</tr><!td>\n";
print "<tr>\n ";
print "<td \nwidth=\"150\" height=\"50\">&nbsp;</td>";
print "<td width=\"150\" height=\"50\">&nbsp;</td>\n";
print "</tr><!td>\n";
print "<tr>\n";
print "<td width=\"150\" height=\"50\">&nbsp;</td>\n";
print "<td width=\"150\" height=\"50\">&nbsp;</td></tr>\n";
print "<tr><td>&nbsp;</td><th align=\"left\">\n";
print "<font size=+3>Web-Vision</font></th></tr><tr><td
align=\"right\">\n";
print "  Usuario:</td><td>\n";
print "<input name=\"Name\" size=20 maxlength=50></td></tr>\n";
print "  <tr><td align=\"right\">\n";
print "  Password:</td><td>\n";
print "  <input type=\"password\" name=\"Password\" size=20
maxlength=50>\n";
print "<input name=\"ID_switch\" type=\"hidden\" id=\"ID_switch\"
value=\"10.2.0.1\">\n";
print "  </td></tr><tr><td width=\"150\"
height=\"10\">&nbsp;</td></tr><tr><td>&nbsp;</td><td align=\"left\">\n";
```



```
print " <INPUT TYPE=submit value=\"Aceptar\"
onclick=\"openwindow();javascript>window.close();\">\n";
print "<!/td><!/td>\n";
print " <INPUT TYPE=reset value=\"Reiniciar\"></td></tr>\n";
print " </FORM>\n";
print "</table>\n";
print "</body>\n";
print "</html>\n";
```

B.2 Aplicación de Administración de Conmutadores Alcatel

aut3.pl

```

#! /usr/perl5/5.00503/bin/perl
use CGI;
$query = new CGI;
#
# Definicion de Variables
#
# $nombre          El nombre del switch como en system
# $localizacion    La localizacion del switch como en system
# %DESCVlan        Hash de las descripciones de los grupos
# %IPVlan          Hash que tiene las direcciones IP
# %MASKVlan        Hash que contiene las mascararas
# %HPortStatus     Hash que contiene el Status de los Puertos
(Ena/Dis)
# %HPortVelocConf Hash que contiene la velocidad configurada en el Pto
# %HPortDuplexConf Hash que contiene el Duplex configurado para el
Pto
# %HPortLink       Hash que contiene el Link (Up/Down) del puerto
# %HPortGroup      Hash que contiene el numero de grupo del pto.
# %HPortAutoNeg    Hash que contiene la Autoneg (Ena/Dis/NA)
# %HPortAutoSpeed Hash que contiene la velocidad del pto.
# %HPortAutoDuplexMode Hash que contiene el Duplex del puerto
# %HPortMirrorStatus Hash que contiene el Status del Port Mirroring
# %HPortMSlot      Hash que contiene el Slot del PM
# %HPortMIF        Hash que contiene el Puerto del PM
#
#
#
# Funcion de creacion de hash
#
#
sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLANSplit=split("\n",$Desc);
    foreach $b (@AVLANSplit) {
        my @B=split(" ",$b);
    }
}

```

```

        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]", $inst_raw);
        $DESCVLan_2{$C[7]}=join(" ", @B);
    }

    delete $DESCVLan_2{""};
    return %DESCVLan_2;
}

#
#
#

sub Genera_Hash_Simple2($$$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];
    my ($Instan) = $_[2];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @AVLANDesc=split("\n", $Desc);
    foreach $b (@AVLANDesc) {
        my @B=split(" ", $b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]", $inst_raw);
        $C[$Instan] =~ s/00/\\/;
        $DESCVLan_2{$C[$Instan]}=join(" ", @B);
    }

    delete $DESCVLan_2{""};
    return %DESCVLan_2;
}

#
#
# Funcion para generar hash Dobles
#
#

sub Genera_Hash_Doble($$$) {

```

```

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ", $b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.]", $inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        my $inst1=shift(@C);

        for ($i=1; $i<$Repeticiones2-$Repeticiones1; $i++) {
            shift(@C);
        }
        my $inst2=shift(@C);

        my $inst="$inst1" . "/" . "$inst2";

        $DESCVLan_2{$inst}=$Valor;
    }

    delete $DESCVLan_2{"/"};

    return %DESCVLan_2;
}

#
#

$IP_switch = $query->param('ID_switch');
$usuarioW = $query->param('Name');
$nombre = $query->param('Nom_switch');
$ID_Puerto = $query->param('ID_Puerto');

#@ve_Desc=sort(keys(%HPortMSlot));
#foreach $b (@ve_Desc) {

```

```

#       print "$b\t\t\t$HPortMSlot{$b}\n";
#       }
# $usuarioW = $query->param('Name');
# $usuarioW = "oper2";
# $passwordW = $query->param('Password');
# $passwordW = "Lucas";
#
# Lectura del archivo para verificacion
#
open (USERS, '/etc/usuarios.wv') || die();
$i = 1;
$text=<USERS>;
while ($text) {
    chop($text);
    @usu = split(" ", $text);
    $usuario[$i++] = $usu[0];
    $password{$usu[0]} = $usu[1];
    $text=<USERS>;
}
close(USERS);
#
#
# Lectura de la IP que solicita
#
#
$IPs = $ENV{'REMOTE_ADDR'};
#$IPs = "10.1.128i146";
#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
#print "$usu[1]\n";
#print "$usuario[1], $usuario[2]\n";
$autentUS = 0;
$autentIP = 0;
foreach $b (@usuario) {
    if (! $b) {next;}
    if (! $usuarioW) {last;}
    if (! $passwordW) {last;}
#       print "$b\n";
    if ($b eq $usuarioW && $password{$b} eq $passwordW) {
        $autentUS = 1;
    }
}

```

```

        last;}
    }
#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
#
    print "$b\n";
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
    }
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/aut3.pl" ||
$ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Buscar.pl") {
    $autentUS=$autentIP=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#
# Self_call
#
#

if ($query->param('Formul')) {

$Slot=$Puerto=$query->param('Puerto');

#
#
# Asignacion de las variables
#
#

$IP_switch = $query->param('IP_switch');
$Grupo_org = $query->param("gru_$Puerto");
$Grupo_nue = $query->param("c_gru_$Puerto");
$Veloc_org = $query->param("veloc_$Puerto");
$Veloc_nue = $query->param("c_veloc_$Puerto");
$Duple_org = $query->param("dup_$Puerto");
$Duple_nue = $query->param("c_dup_$Puerto");
$Stats_org = $query->param("status_$Puerto");
$Stats_nue = $query->param("c_status_$Puerto");
$PMirr_nue = $query->param("checkbox_pm_$Puerto");
$PMirr_org = $query->param("pm_o_$Puerto");
$PMDis_nue = $query->param("pm_l_$Puerto");

```

```

$Slot =~ s/\\d+//g;
$Puerto =~ s/^\d+//g;

#print "$Slot      ;   $Puerto";

#
#
#Archivo de control de cambios
#
#

open(LOG_CAMBIOS, '>>/opt/OV/www/cgi-bin/Control.txt');

$FECHA=`date`;
chop($FECHA);

if (!( $Grupo_org eq $Grupo_nue )) {

    system("/opt/OV/bin/snmpset $IP_switch
vportVlanNumber.$Slot.$Puerto.4.1 integer $Grupo_nue");
    if ( $Stats_nue eq "dis" ) {
        system("/opt/OV/bin/snmpset $IP_switch
vportAdmStatus.$Slot.$Puerto.4.1 integer 1");
    }

print LOG_CAMBIOS "$FECHA : El Usuario $usuarioW cambio el grupo del
puerto $Slot/$Puerto del $Grupo_org al $Grupo_nue en el switch $nombre
($IP_switch)\n\n";
}

if (!( $Veloc_org eq $Veloc_nue )) {

    if ( $Veloc_nue eq "auto" ) {
        system("/opt/OV/bin/snmpset $IP_switch
mesmPortAutoNegotiate.$Slot.$Puerto integer 2");
    }
    else {
        system("/opt/OV/bin/snmpset $IP_switch
mesmPortAutoNegotiate.$Slot.$Puerto integer 1");
    }

    if ( $Veloc_nue eq "100" ) {

        system("/opt/OV/bin/snmpset $IP_switch
mesmPortCfgSpeed.$Slot.$Puerto integer 1");

    }
}

```

```

    if ($Veloc_nue eq "10") {
        system("/opt/OV/bin/snmpset $IP_switch
mesmPortCfgSpeed.$Slot.$Puerto integer 2");
    }

print LOG_CAMBIOS "$FECHA : El Usuario $usuarioW cambio la velocidad del
puerto $Slot/$Puerto de $Veloc_org a $Veloc_nue en el switch $nombre
($IP_switch)\n\n";
}

if (!(($Duple_org eq $Duple_nue)) {
    if ($Duple_nue eq "auto") {

        system("/opt/OV/bin/snmpset $IP_switch
mesmPortCfgDuplexMode.$Slot.$Puerto integer 3");

    }

    if ($Duple_nue eq "half") {

        system("/opt/OV/bin/snmpset $IP_switch
mesmPortCfgDuplexMode.$Slot.$Puerto integer 2");

    }

    if ($Duple_nue eq "full") {

        system("/opt/OV/bin/snmpset $IP_switch
mesmPortCfgDuplexMode.$Slot.$Puerto integer 1");

    }

print LOG_CAMBIOS "$FECHA : El Usuario $usuarioW cambio el modo de Duplex
en el puerto $Slot/$Puerto de $Duple_org a $Duple_nue en el switch
$nombre ($IP_switch)\n\n";
}

if (!(($Stats_org =~ $Stats_nue)) {

    if ($Stats_nue eq "ena") {
        system("/opt/OV/bin/snmpset $IP_switch
vportAdmStatus.$Slot.$Puerto.4.1 integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $usuarioW habilito el
puerto $Slot/$Puerto en el switch $nombre ($IP_switch)\n\n";
    }

    if ($Stats_nue eq "dis") {

```



```

        system("/opt/OV/bin/snmpset $IP_switch
vportAdmStatus.$Slot.$Puerto.4.1 integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $usuarioW deshabilito
el puerto $Slot/$Puerto en el switch $nombre ($IP_switch)\n\n";
    }

}

if ($PMirr_nue) {
    if ($PMirr_org eq "_checked") {

        $PMDis_Slo=$PMDis_Pto=$PMDis_nue;
        $PMDis_Slo =~ s/\/\d+//g;
        $PMDis_Pto =~ s/^\d\///g;

        system("/opt/OV/bin/snmpset $IP_switch
mirrorMirroringSlot.$Slot.$Puerto.4.1 integer $PMDis_Slo");
        system("/opt/OV/bin/snmpset $IP_switch
mirrorMirroringIF.$Slot.$Puerto.4.1 integer $PMDis_Pto");
        system("/opt/OV/bin/snmpset $IP_switch
mirrorStatus.$Slot.$Puerto.4.1 integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $usuarioW habilito el
port mirroring en el puerto $Slot/$Puerto para verlo en el
$PMDis_Slo/$PMDis_Pto del switch $nombre ($IP_switch)\n\n";

    }
}

else {

    if ($PMirr_org eq "checked") {
        system("/opt/OV/bin/snmpset $IP_switch
mirrorStatus.$Slot.$Puerto.4.1 integer 1");
        system("/opt/OV/bin/snmpset $IP_switch
mirrorMirroringSlot.$Slot.$Puerto.4.1 integer 26");
        system("/opt/OV/bin/snmpset $IP_switch
mirrorMirroringIF.$Slot.$Puerto.4.1 integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $usuarioW deshabilito
el port mirroring en el puerto $Slot/$Puerto del switch $nombre
($IP_switch)\n\n";

    }

}

}

```

```

close(LOG_CAMBIOS);

#
#
#
# Generacion de MIB Miscelaneos
#
#
#

@uma_guma=("vLanDescription", "vIPRouterNetAddress",
"vIPRouterSubNetMask",
        "vportAdmStatus", "vportOperStatus", "vportVlanNumber",
        "mesmConfEntry", "mirrorStatus", "mirrorMirroringSlot",
        "mirrorMirroringIF", "system", "vportInUcastPkts",
"ifLastChange");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);
foreach $b (@APM) {
    if ($b =~ "vLanDescription" && $b =~ "STRING") {
        $VLANDesc="$VLANDesc" . "\n" . "$b";
    }
    if ($b =~ "vIPRouterNetAddress" && $b =~ "IpAddress") {
        $VLANIP="$VLANIP" . "\n" . "$b";
    }
    if ($b =~ "vIPRouterSubNetMask" && $b =~ "IpAddress") {
        $VLANMASK="$VLANMASK" . "\n" . "$b";
    }
    if ($b =~ "vportAdmStatus" && $b =~ "INTEGER") {
        $PortStatus="$PortStatus" . "\n" . "$b";
    }
    if ($b =~ "vportOperStatus" && $b =~ "INTEGER") {
        $PortLink="$PortLink" . "\n" . "$b";
    }
    if ($b =~ "vportVlanNumber" && $b =~ "INTEGER") {
        $PortGroup="$PortGroup" . "\n" . "$b";
    }
    if ($b =~ "mesmPortCfgSpeed" && $b =~ "INTEGER") {
        $PortVelocConf="$PortVelocConf" . "\n" . "$b";
    }
    if ($b =~ "mesmPortCfgDuplexMode" && $b =~ "INTEGER") {
        $PortDuplexConf="$PortDuplexConf" . "\n" . "$b";
    }
    if ($b =~ "mesmPortAutoNegotiate" && $b =~ "INTEGER") {
        $PortAutoNeg="$PortAutoNeg" . "\n" . "$b";
    }
    if ($b =~ "mesmPortAutoSpeed" && $b =~ "INTEGER") {
        $PortAutoSpeed="$PortAutoSpeed" . "\n" . "$b";
    }
}

```

```

    }
    if ($b =~ "mesmPortAutoDuplexMode" && $b =~ "INTEGER") {
        $PortAutoDuplexMode="$PortAutoDuplexMode" . "\n" . "$b";
    }
    if ($b =~ "mirrorStatus" && $b =~ "INTEGER") {
        $PortMirrorStatus="$PortMirrorStatus" . "\n" . "$b";
    }
    if ($b =~ "mirrorMirroringSlot" && $b =~ "INTEGER") {
        $MSlot="$MSlot" . "\n" . "$b";
    }
    if ($b =~ "mirrorMirroringIF" && $b =~ "INTEGER") {
        $MIF="$MIF" . "\n" . "$b";
    }
    if ($b =~ "system" && $b =~ "STRING") {
        $sw="$sw" . "\n" . "$b";
    }
    if ($b =~ "sysUpTime" && $b =~ "Timeticks") {
        $UpTime=$b;
    }
    if ($b =~ "vportInUcastPkts" && $b =~ "Counter") {
        $vportInUcastPkts="$vportInUcastPkts" . "\n" . "$b";
    }
    if ($b =~ "ifLastChange" && $b =~ "Timeticks") {
        $ifLastChange="$ifLastChange" . "\n" . "$b";
    }
}

@VUpTime=split("[\\(\\)]", $UpTime);
@syst=split("\n", $sw);
#print "$syst[4]";
@NOMBRE=split(" ", $syst[3]);
#print "@NOMBRE\n";
@LOCALIZACION=split(" ", $syst[4]);
for ($i=1; $i<6; $i++) {
    shift(@NOMBRE);
    shift(@LOCALIZACION);
}
$nombre=join(" ", @NOMBRE);
#$nombre="KKKK";
$Loco="NOMBRE";
$localizacion=join(" ", @LOCALIZACION);
#print "$nombre\n$localizacion\n";

#
#
#
#* Descripcion de la VLAN
#
#
#

#$VLANDesc=`snmpwalk $IP_switch `;

```

```

%DESCVlan=Genera_Hash_Simple($VLANDesc,5);

@Llave_Desc=keys(%DESCVlan);
#foreach $b (@Llave_Desc) {
#    print "$b\t\t$DESCVlan{$b}\n";
#    }

#
#
# IP
#
#

#$VLANIP=`snmpwalk $IP_switch vIPRouterNetAddress`;

%IPVlan=Genera_Hash_Simple($VLANIP,3);

@Llave_Desc=keys(%IPVlan);

#
#
# Mascara
#
#
#

#$VLANMASK=`snmpwalk $IP_switch vIPRouterSubNetMask`;

%MASKVlan=Genera_Hash_Simple($VLANMASK,3);

@Llave_Desc=keys(%MASKVlan);
#foreach $b (@Llave_Desc) {
#    print "$b\t\t$MASKVlan{$b}\n";
#    }

#####
#
#
#
# Inicio del recorrido de los Hash dobles
#
#

```

```
#####
```

```
#  
#  
#  
#
```

```
#$PortStatus=`snmpwalk $IP_switch vportAdmStatus`;
```

```
%HPortStatus=Genera_Hash_Doble($PortStatus,8,9);
```

```
#  
#  
#  
#  
#
```

```
%HUnicast=Genera_Hash_Doble($vportInUcastPkts,8,9);
```

```
#  
#  
#  
#  
#  
#
```

```
#$PortLink=`snmpwalk $IP_switch vportOperStatus`;
```

```
%HPortLink=Genera_Hash_Doble($PortLink,8,9);
```

```
#  
#  
#  
#  
#  
#  
#
```

```
#$PortGroup=`snmpwalk $IP_switch vportVlanNumber`;
```

```
%HPortGroup=Genera_Hash_Doble($PortGroup,8,9);
```

```
#
#
# Configuracion de la velocidad del puerto
#
#
# $PortVelocConf=`snmpwalk $IP_switch `;
# HPortVelocConf=Genera_Hash_Doble($PortVelocConf,8,9);

#
#
# Configuracion del Duplex del puerto
#
#
# $PortDuplexConf=`snmpwalk $IP_switch `;
# HPortDuplexConf=Genera_Hash_Doble($PortDuplexConf,8,9);

#
#
# Autonegociacion del puerto (Enable/Disable)
#
#
# $PortAutoNeg=`snmpwalk $IP_switch `;
# HPortAutoNeg=Genera_Hash_Doble($PortAutoNeg,8,9);

#
#
# Velocidad del Puerto
#
#
#
```

```
#$PortAutoSpeed=`snmpwalk $IP_switch `;  
%HPortAutoSpeed=Genera_Hash_Doble($PortAutoSpeed,8,9);  
  
#  
#  
#  
# Modo del Duplex  
#  
#  
  
#$PortAutoDuplexMode=`snmpwalk $IP_switch `;  
%HPortAutoDuplexMode=Genera_Hash_Doble($PortAutoDuplexMode,8,9);  
  
#  
#  
#  
# Mirroring Status  
#  
#  
  
#$PortMirrorStatus=`snmpwalk $IP_switch mirrorStatus`;  
%HPortMirrorStatus=Genera_Hash_Doble($PortMirrorStatus,8,9);  
  
#  
#  
#  
# Slot del PM y puerto del PM  
#  
#  
  
%HPortMSlot=Genera_Hash_Doble($MSlot,8,9);  
%HPortMIF=Genera_Hash_Doble($MIF,8,9);  
  
#  
#  
# Tiempo de Actividad del switch  
#  
#  
  
%HPortChange=Genera_Hash_Simple2($ifLastChange,2,4);
```

```

foreach $b (keys(%HPortChange)) {
    @VChange=split("[\\(\\)",$HPortChange{$b});
    $HPortChange{$b} = $VChange[1];
}

$autentUS = 0;
print $query->header;

#
#
#
#
# Determinacion de la clave
#
#
#
#

$TRES="3";

if ($nombre eq "La-Uruca" || $nombre eq "San-Pedro" || $nombre eq
"Recursos-Humanos") {$TRES="4";}
if ($nombre eq "Red-Institucional") {$TRES="14";}
if ($nombre =~ "-BS-" || $nombre =~ "-CS-" || $nombre eq "NC-14-SW-05")
{$TRES="2";}

@Vec_IP_aux = split("[\\.]",$IPVLan{$TRES});
@Vec_tercer = split("", $Vec_IP_aux[2]);
@Vec_cuarto = split("", $Vec_IP_aux[3]);

@l_IP=("p", "q", "w", "e", "r", "t", "y", "u", "i", "o");
#@L_IP=("P", "Q", "W", "E", "R", "T", "Y", "U", "I", "O");

#$aux_suma=$Vec_IP_aux[0] + $Vec_IP_aux[1] + $Vec_IP_aux[2] +
$Vec_IP_aux[3];

#$AUX_PASS= "A." . $aux_suma . ",";

#foreach $b (@Vec_tercer) {$AUX_PASS .= $l_IP[$b];}
#foreach $b (@Vec_cuarto) {$AUX_PASS .= $L_IP[$b];}

$AUX_PASS= "A/";

foreach $b (@Vec_tercer) {$AUX_PASS .= $b;}
foreach $b (@Vec_cuarto) {$AUX_PASS .= $b;}

$AUX_PASS= $AUX_PASS . "*";

```



```
foreach $b (@Vec_tercer) {$AUX_PASS .= $l_IP[$b];}
foreach $b (@Vec_cuarto) {$AUX_PASS .= $l_IP[$b];}
```

```
$cuenta="Si Ud. Considera util este programa por favor deposite \$l a la
cuenta electronica 200-02-000-9999999-9";
```

```
print "<META HTTP-EQUIV=\"Content-Type\" CONTENT=\"text/html;
charset=iso-8859-1\">\n";
print "<TITLE>UMMR-BNCR-Web Vision</TITLE>\n";
print "</HEAD><BODY bottomMargin=\"0\" BGCOLOR=\"#FFFFFF\"><FONT
FACE=\"Arial, Helvetica\"> \n";
print "<table border=\"0\" align=\"center\" width=\"95%\"
cellpadding=\"0\" cellspacing=\"0\"><tr><td>\n";
print "<table border=\"0\" width=\"100%\" cellpadding=\"0\"
cellspacing=\"0\"><tr>\n";
print "      <td nowrap bgcolor=\"#002266\" width=\"30%\"><font
color=\"#FFFFFF\" face=\"Arial, Helvetica\" size=\"3\">&nbsp;</font>\n";
print "      <td nowrap bgcolor=\"#002266\" align=\"center\"> <font
color=\"#FFFFFF\" face=\"Arial, Helvetica\" size=\"3\"> \n";
print "      <B>&nbsp;</B></font>&nbsp;</td>\n";
print "      <td nowrap bgcolor=\"#002266\" align=\"right\"
width=\"30%\"> <font color=\"#FFFFFF\" face=\"Arial, Helvetica\"
size=\"3\"> \n";
print "      <B>$nombre";
print "</B></font>&nbsp;</td>\n";
print "\n";
print "\n";
print "</table></td></tr><tr><td bgcolor=\"#DCDCDC\">\n";
print "\n";
print "<table border=\"0\" width=\"100%\" cellpadding=\"0\"
cellspacing=\"0\"><tr>\n";
print "\n";
print "      <td nowrap bgcolor=\"#DCDCDC\"> &nbsp;&nbsp;&nbsp;</td>\n";
print "      <td nowrap bgcolor=\"#DCDCDC\" align=\"right\">&nbsp;&nbsp;&
</td>\n";
print "</table>\n";
print "\n";
print "</td></tr></table>\n";
print "\n";
print "<table width=95% border=0 cellspacing=1 cellpadding=0
align=center>\n";
print "<tr>\n";
print "      <td align=\"right\" nowrap><strong>&nbsp;<font
class=\"smalltext\">";
#print $query->h3("$nombre");
print "Visite: <a
Title=\"$cuenta\"href=\"http://www.bnrcr.fi.cr\">www.bnrcr.fi.cr</a></font>
</strong> \n";
print "      </td>\n";
print " </tr></table>\n";
```

```

print "\n";
print "\n";
print "<table border=\"0\" align=\"center\" width=\"95%\"
cellpadding=\"1\" cellspacing=\"0\">\n";
print " <tr> \n";
print "     <td nowrap bgcolor=\"#DCDCDC\">&nbsp;</td>\n";
print "     <td align=\"center\" bgcolor=\"#DCDCDC\"
nowrap>&nbsp;</td>\n";
print "     <td align=\"center\" bgcolor=\"#DCDCDC\"
nowrap>&nbsp;</td>\n";
print "     <td align=\"right\" nowrap bgcolor=\"#DCDCDC\">&nbsp;</td>\n";
print " </tr>\n";
print " <tr> \n";
print "     <td width=\"35%\" nowrap>&nbsp;</td>\n";
print "     <td align=\"center\" nowrap>&nbsp;</td>\n";
print "     <td align=\"center\" nowrap>&nbsp;</td>\n";
print "     <td width=\"35%\" align=\"right\" nowrap>&nbsp;</td>\n";
print " </tr>\n";
print "</table>\n";
print "<div align=\"center\">\n";
print " <table width=\"88%\" border=\"0\">\n";
print "     <tr bgcolor=\"9999FF\"> \n";
print "         <td colspan=\"4\"> <div
align=\"center\"><strong>Localizaci&oacute;n</strong> \n";
print "             </div>\n";
print "             <div align=\"center\"></div></td>\n";
print "     </tr>\n";
print "     <tr> \n";
print "         <td colspan=\"4\">$localizacion</td>\n";
print "     </tr>\n";
print "     <tr bgcolor=\"9999FF\"> \n";
print "         <td width=\"11%\"><strong>VLAN ID</strong></td>\n";
print "         <td width=\"39%\"><div
align=\"center\"><strong>Descripci&oacute;n del
Grupo</strong></div></td>\n";
print "             <td width=\"25%\"><div
align=\"center\"><strong>Direcci&oacute;n IP</strong></div></td>\n";
print "             <td width=\"25%\"><div
align=\"center\"><strong>M&aacute;scara</strong></div></td>\n";
print "         </tr>\n";

```

```
@Llaves = sort(keys(%DESCVlan));
```

```
foreach $b (@Llaves) {
```

```

    print "     <tr bgcolor=\"EEEEEE\"> \n";
    print "         <td><div align=\"center\">$b</div></td>\n";
    print "         <td><p>$DESCVlan{$b}</p></td>\n";
    if ($IPVlan{$b}) {$IPPP=$IPVlan{$b};} else {$IPPP="&nbsp;";}
    print "         <td><p>$IPPP</p></td>\n";
    if ($MASKVlan{$b}) {$IPPP=$MASKVlan{$b};} else {$IPPP="&nbsp;";}

```

```

print "      <td><p>$IPPP</p></td>\n";
print "      </tr>\n";
}

print "      <tr bgcolor=\"9999FF\"> \n";
print "      <td>&nbsp;</td>\n";
print "      <td>&nbsp;</td>\n";
print "      <td>&nbsp;</td>\n";
print "      <td>&nbsp;</td>\n";
print "      </tr>\n";
print "      <tr> \n";
print "      <td> \n";
print "      <div align=\"center\"></div></td>\n";
print "      <td>&nbsp;</td>\n";
print "      <td>&nbsp;</td>\n";
print "      <td>&nbsp;</td>\n";
print "      </tr>\n";
print " </table>\n";
print " \n";
print "</div>\n";
print "<table border=\"0\" align=\"center\" width=\"95%\"
cellpadding=\"1\" cellspacing=\"1\" bgcolor=\"#FFFFFF\">\n";
print " <tr> \n";
print "      <td width=\"33%\" bgcolor=\"#D3DCE3\">\n";
print "      </td>\n";
print " </tr>\n";
print " <tr> \n";
print "      <td bgcolor=\"#EEEEEE\"><p><font size=\"+2\" face=\"Arial,
Helvetica, sans-serif\"><strong><em>Otros
Links:</em></strong></font></p>\n";
print "      <p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a Title=\"Respaldos\"
href=\"ftp://oper2:openview@10.1.4.151/ba/respaldo\"
target=\"_blank\"><img src=\"/OvDocs/images/Respaldos.gif\"
border=\"0\"></a>\n";
print "      &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;\n";
print "      &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;\n";
print "      <a Title=\"Configuraciones CLI\"
href=\"ftp://oper2:openview@10.1.4.151/ba/CLI\" target=\"_blank\"><img
src=\"/OvDocs/images/CLI.gif\" border=\"0\"></a>";
print "      &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;\n";
print "      &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;\n";
print "      <a Title=\"Servidor ftp (Switch)\"
href=\"ftp://admin:$AUX_PASS@$IPVlan($TRES)\" target=\"_blank\"><img
src=\"/OvDocs/images/FTP.gif\" border=\"0\"></a>";
print "      &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;\n";
print "      &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;\n";
print "<a Title=\"Buscar Direcciones IP y MAC\"
href=\"Buscar.pl?Name=$usuarioW\" target=\"_blank\"><img
src=\"/OvDocs/images/Buscar.gif\" border=\"0\"></a>";

```

```

if ($usuarioW eq "oper2" || $usuarioW eq "oper4" || $usuarioW eq "oper8")
{
    print "    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; \n";
    print "    &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; \n";
    print "<a Title=\"Consulta al Log de Cambios\"
href=\"Log.pl?Name=$usuarioW\" target=\"_blank\"><img
src=\"/OvDocs/images/VerLog.gif\" border=\"0\"></a>\n";
}

```

```

print "<form name=\"form1\" method=\"post\" action=\"aut3.pl\">\n";
print "    Nuevo Switch: \n";
print "        <input name=\"ID_switch\" type=\"text\">\n";
print "        <input type=\"submit\" name=\"Submit2\"
value=\"Enviar\">\n";
print "        <input name=\"Name\" type=\"hidden\" id=\"Name\"
value=\"$usuarioW\">\n";
print "        <input name=\"Nom_switch\" type=\"hidden\"
id=\"Nom_switch\" value=\"$nombre\">\n";
print "    </form>\n";

```

```

print "</p>\n";
print "    <p>&nbsp;</p></td>\n";
print " </tr>\n";
print " <tr>\n";
print "     <td bgcolor=\"#9999FF\">&nbsp;</td>\n";
print " </tr>\n";
print " <tr> \n";
print "     <td bgcolor=\"#FFFFFF\">&nbsp;</td>\n";
print " </tr>\n";
print "</table>\n";
print "\n";
print "<div align=\"center\">\n";
print " <table width=\"100%\" border=\"0\">\n";
print " <tr> \n";
print "     <td width=\"7%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>Puerto</u> \n";
print "         </strong></div></td>\n";
print "     <td width=\"7%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>Grupo</u> \n";
print "         </strong></div></td>\n";
print "     <td width=\"7%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>C. \n";
print "         Grupo</u> </strong></div></td>\n";
print "     <td width=\"7%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>Veloc.</u> \n";
print "         </strong></div></td>\n";
print "     <td width=\"9%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>C. \n";
print "         Veloc.</u> </strong></div></td>\n";
print "     <td width=\"8%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>Duplex</u> \n";

```

```

print "          </strong></div></td>\n";
print "          <td width=\"8%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>C. \n";
print "          Dup.</u> </strong></div></td>\n";
print "          <td width=\"8%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>Status</u> \n";
print "          </strong></div></td>\n";
print "          <td width=\"8%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>C. \n";
print "          Status</u> </strong></div></td>\n";
print "          <td width=\"10\" colspan=\"2\" align=\"center\"
bgcolor=\"#D3DCE3\"><div align=\"center\"><strong><u>Port \n";
print "          Mirroring</u></strong></div></td>\n";
print "          <td width=\"10%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>Ptos \n";
print "          Disponib.</u> </strong></div></td>\n";
print "          <td width=\"11%\" bgcolor=\"#D3DCE3\" align=\"center\"><div
align=\"center\"><strong><u>Refrescar</u> \n";
print "          </strong></div></td>\n";
print "          </tr>\n";

```

```

@VELOCIDADES=("auto", "10", "100");
@DUPLEX=("auto", "half", "full");
@STATUS=("ena", "dis");

```

```

@Llaves=sort (keys(%HPortLink));

```

```

$Contador=0;
foreach $b (@Llaves) {
    $DES_PM{$b}="checkbox";
    if ($HPortMirrorStatus{$b} eq "enable")
    {$PUERTO_NDIS{$Contador}=$HPortMSlot{$b} . "/" . $HPortMIF{$b};}
    if ($HPortGroup{$b}==1) {$PUERTO_DIS{$Contador}=$b;}
    $Contador++;
}

foreach $a (keys(%PUERTO_DIS)) {
    foreach $b (keys(%PUERTO_NDIS)) {
        $DES_PM{$PUERTO_NDIS{$b}}="hidden";
        if ($PUERTO_DIS{$a} eq $PUERTO_NDIS{$b}) {delete
$PUERTO_DIS{$a};}
    }
    if ($PUERTO_DIS{$a} eq "2/1" || $PUERTO_DIS{$a} eq "2/2") {delete
$PUERTO_DIS{$a};}
}

```

```

if ($ID_Puerto) {
    foreach $b (@Llaves) {

```

```

        delete $HPortLink{$b} if (!(($b eq $ID_Puerto));
    }
    @Llaves=sort(keys(%HPortLink));
}

$Contador=0;
foreach $b (@Llaves) {
    $nombre_form="\form_" . "$b\";
    if ($Contador++ % 2 == 1) {$BGCOLOR="\#EEEEEE\";} else
    {$BGCOLOR="\#CCCCCC\";}
    if ($HPortLink{$b} eq "portUp") {$strong="<strong>";
    $nstrong="</strong>";}
    else {$strong=""; $nstrong="";}
    $group="\gru_" . "$b\";
    $c_group="\c_gru_" . "$b\";
    $veloc="\veloc_" . "$b\";
    $c_veloc="\c_veloc_" . "$b\";
    $dup="\dup_" . "$b\";
    $c_dup="\c_dup_" . "$b\";
    $status="\status_" . "$b\";
    $c_status="\c_status_" . "$b\";
    $pm="\$DES_PM{$b}_pm_" . "$b\";
    $pm_o="\pm_o_" . "$b\";
    $pm_l="\pm_l_" . "$b\";
#}

#$b="2/1";

# Aqui se inserto marca para indicar que el puerto no ha tenido actividad
#$Camb_Grupo=($HUnicast{$b} == 0 && $HPortGroup{$b} != 1) ? "*" : "";
#$Camb_GrupoT=($HUnicast{$b} == 0 && $HPortGroup{$b} != 1 &&
$VUpTime[1]>345600000) ? "*" : "";

$Delta_T=$VUpTime[1]-$HPortChange{$b};

$Camb_GrupoT="";

$Camb_Grupo=($Delta_T>345600000 && $HPortLink{$b} eq "portDown" &&
$HPortGroup{$b} != 1) ? "* *" : "";

print "    <tr bgcolor=$BGCOLOR
onMouseOver=\'this.style.backgroundColor=\#D0DCE3\';\'
onMouseOut=\'this.style.backgroundColor = $BGCOLOR;\'
id=\'$Contador\''>\n";

```

```

#print "      <tr bgcolor=$BGCOLOR
onMouseOver=\'this.style.backgroundColor=\'"#0FCCCC\';\'
onMouseOut=\'this.style.backgroundColor =
document.getElementById(\'"$Contador"\').checked?
\'"#DDDDDD\':\'"#EEEEEE\';\' id=\'"$Contador"\">>\n";

#print "      <tr bgcolor=$BGCOLOR>\n";
print "      <form name=$nombre_form method=\'"post\'
action=\'"aut3.pl\'">\n";
print "      <td nowrap><div align=\'"left\'">$strong $b $Camb_Grupa
$Camb_GrupaT $nstrong\n";
print "      <input name=\'"IP_switch\'" type=\'"hidden\'"
id=\'"IP_switch\'" value=\'"$IPVLan{$TRES}\'">\n";
print "      <input name=\'"Puerto\'" type=\'"hidden\'" id=\'"Puerto\'"
value=\'"$b\'">\n";
print "      <input name=\'"Formul\'" type=\'"hidden\'" id=\'"Formul\'"
value=\'"Ver\'">\n";
print "      <input name=\'"ID_Puerto\'" type=\'"hidden\'"
id=\'"ID_Puerto\'" value=\'"$ID_Puerto\'">\n" if ($ID_Puerto);
print "      <input name=\'"Name\'" type=\'"hidden\'" id=\'"Name\'"
value=\'"$usuarioW\'">\n";
print "      <input name=\'"Nom_switch\'" type=\'"hidden\'"
id=\'"Nom_switch\'" value=\'"$nombre\'">\n";
print "      <input type=\'"checkbox\'" />\n";
print "      <td nowrap><div align=\'"left\'">$strong $HPortGroup{$b}
$nstrong\n";
print "      <input name=$group type=\'"hidden\'" id=$group
value=\'"$HPortGroup{$b}\'">\n";
print "      </div></td>\n";
print "      <td nowrap><div align=\'"center\'"> \n";
print "      <select name=$c_group id=$c_group >\n";

foreach $a (sort(keys(%DESCVLan))) {

    print "      <option value=\'"$a\'" ";
    if ($a==$HPortGroup{$b}) {print "selected >";} else {print ">";}
    print "$a</option>\n"; }

print "      </select>\n";
print "      </div></td>\n";

$HPortAutoSpeed{$b} =~ s/speed-//;
$HPortAutoSpeed{$b} =~ s/unknown//;

#%HPortVelocConf
$HPortVelocConf{$b} =~ s/speed-//;
$HPortVelocConf{$b} =~ s/unknown//;

if (!$HPortAutoSpeed{$b}) {$HPortAutoSpeed{$b}=$HPortVelocConf{$b};}

```

```

print "          <td nowrap><div align=\"left\">$strong $HPortAutoSpeed{$b}
$strong\n";
print "          <input name=$veloc type=\"hidden\" id=$veloc
value=\"$HPortVelocConf{$b}\">\n";
print "          </div></td>\n";
print "          <td nowrap> <div align=\"center\"> \n";

print "          <select name=$c_veloc width=\"1\">\n";

foreach $a (@VELOCIDADES) {

    print "          <option value=\"$a\"";
    if ($a eq $HPortVelocConf{$b}) {print " selected>";} else {print
">";}
    print "$a</option>\n"; }

print "          </select>\n";
print "          </div></td>\n";

$HPortAutoDuplexMode{$b} =~ s/-duplex//;
$HPortAutoDuplexMode{$b} =~ s/unknown//;

$HPortDuplexConf{$b} =~ s/-duplex//;
$HPortDuplexConf{$b} =~ s/unknown//;

if (!$HPortAutoDuplexMode{$b})
{$HPortAutoDuplexMode{$b}=$HPortDuplexConf{$b};}

print "          <td nowrap><div align=\"left\">$strong
$HPortAutoDuplexMode{$b} $strong\n";
print "          <input name=$dup type=\"hidden\" id=$dup
value=$HPortDuplexConf{$b}>\n";
print "          </div></td>\n";
print "          <td nowrap><div align=\"center\"> \n";
print "          <select name=$c_dup >\n";

foreach $a (@DUPLEX) {

    print "          <option value=\"$a\"";
    if ($HPortDuplexConf{$b} eq $a) {print " selected>";} else {print
">";}
    print "$a</option>\n"; }

print "          </select>\n";
print "          </div></td>\n";

```



```

print "          <td nowrap><div align=\"left\">$strong $HPortStatus{$b}
$strong\n";
print "          <input name=$status type=\"hidden\" id=$status
value=$HPortStatus{$b}>\n";
print "          </div></td>\n";
print "          <td nowrap> <div align=\"center\"> \n";
print "          <select name=$c_status id=$c_status >\n";

foreach $a (@STATUS) {

    print "          <option value=\"$a\"";
    if ($HPortStatus{$b} =~ $a) {print " selected >";} else {print
">";}
    print "$a</option>\n"; }

print "          </select>\n";
print "          </div></td>\n";
print "          <td nowrap> <div align=\"center\"> \n";
print "          <input name=$pm type=\"$DES_PM{$b}\" id=$pm
value=\"habilitar_PM\"";

if ($HPortMirrorStatus{$b} eq "enable") (

    print " checked>\n";
    print "          <input name=$pm_o type=\"hidden\" id=$pm_o
value=\"checked\">\n";
    print "          </div></td>\n";
    print "          <td nowrap>$strong $HPortMSlot{$b}/$HPortMIF{$b}
$strong</td>\n";)

else {
    print ">\n";
    print "          <input name=$pm_o type=\"hidden\" id=$pm_o
value=\"_checked\">\n";
    print "          </div></td>\n";
    print "          <td nowrap>$strong - $strong</td>\n";}

print "          <td nowrap><div align=\"center\"> \n";
print "          <select name=$pm_l id=$pm_l >\n";

foreach $a (keys(%PUERTO_DIS)) {

    if (!( $PUERTO_DIS{$a} eq $b)) {
        print "          <option
value=\"$PUERTO_DIS{$a}\">$PUERTO_DIS{$a}</option>\n";}}

```

```

print "          </select>\n";
print "        </div></td>\n";
print "      <td nowrap><div align=\"center\"> \n";
print "        <input type=\"submit\" name=\"Submit\" value=\"$b >\n";
print "      </div></td>\n";
print "    </form>\n";
print "  </tr>\n";

```

```

#Aqui se cierra el lazo
}

```

```

print " </table>\n";
print " <p>&nbsp;</p>\n";
print "</div>\n";
print "</font></BODY></HTML>\n";

```

```

}
else
{
print $query->header;
print $query->start_html(-title=>'No Autorizado');
print $query->h1('No esta autorizado para ver esta pagina');
# foreach $envvar (sort keys %ENV) {
#   print "$envvar : $ENV{$envvar}<br>\n";
# }
print $query->end_html;
}

```

B.3 Aplicación de Búsqueda de Direcciones IP en Conmutadores Alcatel

```

#! /usr/perl5/5.00503/bin/perl
use CGI;
$query = new CGI;
#
# Definicion de Variables
#
# $nombre          El nombre del switch como en system
# $localizacion    La localizacion del switch como en system
# %DESCVLan        Hash de las descripciones de los grupos
# %IPVLan          Hash que tiene las direcciones IP
# %MASKVLan        Hash que contiene las mascararas
# %HPortStatus     Hash que contiene el Status de los Puertos
(Ena/Dis)
# %HPortVelocConf Hash que contiene la velocidad configurada en el Pto
# %HPortDuplexConf Hash que contiene el Duplex configurado para el
Pto
# %HPortLink       Hash que contiene el Link (Up/Down) del puerto
# %HPortGroup      Hash que contiene el numero de grupo del pto.
# %HPortAutoNeg    Hash que contiene la Autoneg (Ena/Dis/NA)
# %HPortAutoSpeed Hash que contiene la velocidad del pto.
# %HPortAutoDuplexMode Hash que contiene el Duplex del puerto
# %HPortMirrorStatus Hash que contiene el Status del Port Mirroring
# %HPortMSlot      Hash que contiene el Slot del PM
# %HPortMIF        Hash que contiene el Puerto del PM
#

sub Invertir {

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
}

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];

```

```

my ($Repeticiones) = $_[1];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {
    delete $DESCVLan_2{$re};
}

my @AVLANDesc=split("\n",$Desc);
foreach $b (@AVLANDesc) {
    my @B=split(" ",$b);
    my $inst_raw=shift(@B);
    for (my $i=1; $i<$Repeticiones; $i++) {
        shift(@B);
    }
    my @C=split("[\.]",$inst_raw);
    $DESCVLan_2{$C[7]}=join(" ",@B);
}

delete $DESCVLan_2{""};
return %DESCVLan_2;

}

#
#
#

#
#
# Funcion para generar hash Dobles
#
#
#

sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

```

```

my $Valor=pop(@B);

my @C=split("[\.]", $inst_raw);

for (my $i=1; $i<$Repeticiones1; $i++) {
    shift(@C);
}

for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--) {
    pop(@C);
}

my $inst=join($Tipo,@C);

$DESCVLan_2{$inst}=$Valor;
}

return %DESCVLan_2;
}

#
#

sub Genera_Hash_HEX($$$$$) {

#
# Vectores de conversion a Decimal
#

for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;
}

```

```

}

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Separador) = $_[2];
    my ($Tipo) = $_[4];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

my @APORTStatus=split("\n",$DDesc);

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    if ($b =~ "OCTET STRING-") {
        my $inst_raw=shift(@B);

        my @C=split("[\.]",$inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        $inst1=join($Separador,@C);
        $DESCVlan_2{$inst1}="";
        next;
    }
    else {
        $B[1] = $HEX{$B[1]} if ($Tipo);
        $Valor=$B[1] . " ";
        for ($i=2; $i<17; $i++) {
            $B[$i] = $HEX{$B[$i]} if ($Tipo);
            $Valor=$Valor . $B[$i] . " ";
        }
        $Valor =~ s/ +//g;
        $Valor =~ s/ -- //g;
        $Valor =~ s/--//g;
        $Valor =~ s/ $//g;
        $Valor =~ s/ /$_[3]/g;
    }
    $DESCVlan_2{$inst1}=$DESCVlan_2{$inst1} . $Valor;
}

delete $DESCVlan_2{""};

return %DESCVlan_2;
}

```

```

# Aqui terminan las subrutinas

#
#
# Asignacion de las variables
#
#

$IP_switch = $query->param('ID_switch');
$Tipo_Busq = $query->param('Tipo_Busqueda');
$Tipo_Busq = 0 if (!$Tipo_Busq);
$usuarioW = $query->param('Name');
$PUERTO = $query->param('ID_Puerto');

$DIRECC[0]="Dirección Buscada";
$DIRECC[1]="Dirección Buscada";
$DIRECC[2]="Dirección del switch";

$SELECC[2]="selected";

#
# Lectura de la IP que solicita
#
$IPs = $ENV{'REMOTE_ADDR'};
#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

```

```

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
#    print "$b\n";
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
    }
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Buscar.pl" ||
$ENV{'HTTP_REFERER'} eq "http://10.1.4.151:8880/OvCgi/aut3.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
#$autentUS = $autentIP =1;
if ($autentUS && $autentIP) {

#if ($query->param('Formul')) {

#
#
#
#
# Generacion de MIB Miscelaneos
#
#
#

if ($IP_switch) {
    $IP=$IP_switch;
    chomp($IP);
    @Cons_Rap=split("[\.]", $IP);
    $SWI1=$Cons_Rap[0] . "." . $Cons_Rap[1] . "." . $Cons_Rap[2] .
".1";

    $SWI1="10.1.1.81" if ($SWI1 eq "10.1.0.1" || $SWI1 eq "10.1.1.1");

    $Cons=`/opt/OV/bin/snmpwalk $SWI1 atPhysAddress`;
    $IP_AUX=$IP . " : OCTET";
    $Cons = $IP_AUX if ($Tipo_Busq == 2);

    if ($Cons !~ $IP_AUX) {

        $PING=`ping $IP`;

```



```

        $no_pinea=($PING !~ "alive");
        @TRACEROUTE=`traceroute $IP 2>/dev/null` if (!(($no_pinea));
        @SWI=split(" ",$TRACEROUTE[ $#TRACEROUTE-1]);
        $SWITCH=$SWI[1];
    }
else {
    $no_pinea=0;
    $SWITCH=$SWI1;
}

if (!(($no_pinea)) {

    @uma_guma = ("vBrdgTpFdbRcvPortSlot", "vBrdgTpFdbRcvPortIF",
    "vBrdgTpFdbDupStatus", "vportNumber", "atPhysAddress",
    "xylanXIPXMAPHostMac", "xylanXIPXMAPAdjPort");

    @MIBS = ` /opt/OV/bin/snmpwalk $SWITCH @uma_guma`;

    $encontrado0=0;
    $encontrado1=0;

    foreach $b (@MIBS) {
        if ($b =~ "atPhysAddress" && $b =~ "OCTET STRING" ||
$encontrado0) {
            $atPhysAddress=$atPhysAddress . "\n" . "$b";
            $encontrado0=1;
            $encontrado0=0 if ($b =~ "-- -- --");
        }
        if ($b =~ "xylanXIPXMAPHostMac" && $b =~ "OCTET STRING" ||
$encontrado1) {
            $xylanXIPXMAPHostMac=$xylanXIPXMAPHostMac . "\n"
.$b";
            $encontrado1=1;
            $encontrado1=0 if ($b =~ "-- -- --");
        }
        if ($b =~ "vBrdgTpFdbRcvPortSlot" && $b =~ "INTEGER") {
            $vBrdgTpFdbRcvPortSlot=$vBrdgTpFdbRcvPortSlot .
"\n" . "$b";
        }
        if ($b =~ "vBrdgTpFdbRcvPortIF" && $b =~ "INTEGER") {
            $vBrdgTpFdbRcvPortIF=$vBrdgTpFdbRcvPortIF . "\n" .
"$b";
        }
        if ($b =~ "vBrdgTpFdbDupStatus" && $b =~ "INTEGER") {
            $vBrdgTpFdbDupStatus=$vBrdgTpFdbDupStatus . "\n" .
"$b";
        }
        if ($b =~ "vportNumber" && $b =~ "INTEGER") {
            $vportNumber=$vportNumber . "\n" . "$b";
        }
    }
}

```

```

        if ($b =~ "xylanXIPXMAPAdjPort" && $b =~ "INTEGER") {
            $xylanXIPXMAPAdjPort=$xylanXIPXMAPAdjPort . "\n" .
"$b";
        }
    }

%Tabla_ARP=Genera_Hash_HEX($atPhysAddress,6,".",",","D");
%Tabla_ARP2=Genera_Hash_HEX($atPhysAddress,6,".",",","-","");
%VIA=Genera_Hash_Doble($vportNumber,8,9,"/");
%Slot=Genera_Hash_Doble($vBrdgTpFdbRcvPortSlot,9,14,".");
%Puerto=Genera_Hash_Doble($vBrdgTpFdbRcvPortIF,9,14,".");
%XMAPPto=Genera_Hash_Doble($xylanXIPXMAPAdjPort,9,14,".");
%DupStatus=Genera_Hash_Doble($vBrdgTpFdbDupStatus,9,14,".");
%XMAPIP=Genera_Hash_HEX($xylanXIPXMAPHostMac,14,".",",","G");

#
# Determinacion de si el puerto es cross
#
#
$cross=0;
$Slo=$Slot{$Tabla_ARP{$IP}};
$Pto=$Puerto{$Tabla_ARP{$IP}};
foreach $rec (keys(%Slot)) {
    next if ($rec eq $Tabla_ARP{$IP});
    if ($Slot{$rec} . "/" . $Puerto{$rec} eq
$Slot{$Tabla_ARP{$IP}} . "/" . $Puerto{$Tabla_ARP{$IP}}) {
        $cross=1;
        last;
    }
}

$cross = 0 if ($XMAPIP{$IP});

$cross = 1 if ($Tipo_Busq == 2 && !($IP eq $SWITCH));
#
#Continue si es un cross
#

if ($cross) {
    %IXMAPPto=Invertir(%XMAPPto);
    %IXMAPIP=Invertir(%XMAPIP);
    $Aux_PS=$Slot($Tabla_ARP{$IP}) . "/" .
$Puerto{$Tabla_ARP{$IP}};
    $SWITCH=($Tipo_Busq != 2) ?
$IXMAPIP{$IXMAPPto{$VIA{$Aux_PS}}} : $IP;
    @relic = ("vBrdgTpFdbRcvPortSlot", "vBrdgTpFdbRcvPortIF");
    @MIBS = `/opt/OV/bin/snmpwalk $SWITCH @relic`;
    $vBrdgTpFdbRcvPortSlot="";
    $vBrdgTpFdbRcvPortIF="";
    $vBrdgTpFdbDupStatus="";
    foreach $b (@MIBS) {
        if ($b =~ "vBrdgTpFdbRcvPortSlot" && $b =~ "INTEGER") {

```

```

                                $vBrdgTpFdbRcvPortSlot=$vBrdgTpFdbRcvPortSlot .
"\n" . "$b";
                                }
                                if ($b =~ "vBrdgTpFdbRcvPortIF" && $b =~ "INTEGER") {
                                    $vBrdgTpFdbRcvPortIF=$vBrdgTpFdbRcvPortIF . "\n" .
"$b";
                                }
                                if ($b =~ "vBrdgTpFdbDupStatus" && $b =~ "INTEGER") {
                                    $vBrdgTpFdbDupStatus=$vBrdgTpFdbDupStatus . "\n" .
"$b";
                                }
                            }

                            %Slot=Genera_Hash_Doble($vBrdgTpFdbRcvPortSlot,9,14,".");
                            %Puerto=Genera_Hash_Doble($vBrdgTpFdbRcvPortIF,9,14,".");
                            $Slo=$Slot{$Tabla_ARP{$IP}};
                            $Pto=$Puerto{$Tabla_ARP{$IP}};
                        }
                    }
                    $cuenta="Si Ud. Considera util este programa por favor deposite \$1 a la
                    cuenta electronica 200-02-000-701965-9";
                    print $query->header;
                    print<<"_HTML_";
                    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
                    <!-- saved from
                    url=(0051)http://10.1.4.151:8880/OvCgi/aut3.pl?ID_switch=os66 -->
                    <HTML><HEAD><TITLE>UMMR-BNCR-Busqueda de Direcciones IP</TITLE>
                    <META http-equiv=Content-Type content="text/html; charset=iso-8859-1">
                    <META content="MSHTML 6.00.2800.1400" name=GENERATOR></HEAD>
                    <script language="JavaScript" type="text/JavaScript">
                    <!--
                    function nuevaAccion() {
                        FormIP.action = "http://10.1.4.151:8880/OvCgi/Buscar.pl";
                        FormIP.submit();
                    }

                    function FormIP_onsubmit() {
                        window.alert(FormIP.method);
                        FormIP_onsubmit = true;
                    }
                    //-->
                    </script>

                    <BODY bgColor=#ffffff><FONT face="Arial, Helvetica">
                    <TABLE cellSpacing=0 cellPadding=0 width="95%" align=center border=0>
                    <TBODY>
                    <TR>
                    <TD>
                    <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
                    <TBODY>

```

```

        <TR>
            <TD width="43%" height="19" noWrap bgColor=#002266><FONT
face="Arial, Helvetica"
            color=#ffffff size=3>&nbsp;</FONT> <font color="ffffff"
size="3" face="Arial, Helvetica"><strong>B&Uacute;SQUEDA
            DE DIRECCIONES</strong></font>
            <TD width="16%" align=middle noWrap bgColor=#002266><FONT
face="Arial, Helvetica" color=#ffffff
size=3><B>&nbsp;</B></FONT>&nbsp;</TD>
            <TD noWrap align=right width="41%" bgColor=#002266><FONT
face="Arial, Helvetica" color=#ffffff
size=3><B>UMMR</B></FONT>&nbsp;</TD>
        </TR></TBODY></TABLE></TD></TR>
<TR>
    <TD bgColor=#dcdcdc>
        <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
            <TBODY>
                <TR>
                    <TD noWrap bgColor=#dcdcdc>&nbsp;&nbsp;&nbsp;</TD>
                    <TD noWrap align=right bgColor=#dcdcdc>&nbsp;&nbsp;&
                </TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=1 cellPadding=0 width="95%" align=center border=0>
    <TBODY>
        <TR>
            <TD noWrap align=right><STRONG>&nbsp;<FONT class=smalltext>Visite:
<A Title="$cuenta"
            href="http://www.bncr.fi.cr/">www.bncr.fi.cr</A></FONT></STRONG>
        </TD>
    </TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=1 width="95%" align=center border=0>
    <TBODY>
        <TR>
            <TD noWrap bgColor=#dcdcdc>&nbsp;</TD>
            <TD noWrap align=middle bgColor=#dcdcdc>&nbsp;</TD>
            <TD noWrap align=middle bgColor=#dcdcdc>&nbsp;</TD>
            <TD noWrap align=right bgColor=#dcdcdc>&nbsp;</TD></TR>
        <TR>
            <TD noWrap width="35%">&nbsp;</TD>
            <TD noWrap align=middle>&nbsp;</TD>
            <TD noWrap align=middle>&nbsp;</TD>
            <TD noWrap align=right width="35%">&nbsp;</TD></TR></TBODY></TABLE>
<DIV align=center>
    <TABLE width="88%" border=0>
        <TBODY>
            <TR>
                <TD width="11%"> <DIV align=center></DIV></TD>
                <TD width="39%">&nbsp;</TD>
                <TD width="25%">&nbsp;</TD>
                <TD width="25%">&nbsp;</TD>
            </TR>
        </TBODY>
    </TABLE>
</DIV>

```

```

</DIV>
<TABLE cellSpacing=1 cellPadding=1 width="95%" align=center
bgColor=#ffffff
border=0>
  <TBODY>
    <TR>
      <TD width="33%" bgColor=#d3dce3></TD></TR>
    <TR>
      <TD bgColor=#eeeeee>
        <P><FONT face="Arial, Helvetica, sans-serif"
size=+2><STRONG><EM>B&uacute;squeda:</EM></STRONG></FONT></P>
<form name="FormIP" method="post" action="Buscar.pl">
  <select name="Tipo_Busqueda" onChange="return nuevaAccion()">
    <option value="1">Busqueda de IP</option>
    <option value="2" $SELECC[$Tipo_Busq]>Busqueda en Puerto de
Switch</option>
  </select>
<BR>&nbsp;<BR>

          <font face="Arial, Helvetica">$DIRECC[$Tipo_Busq]</font>:
          <INPUT
            name=ID_switch>
          <input name="Name" type="hidden" id="Name" value="$usuarioW">

_HTML_

if ($Tipo_Busq == 2) {
  print<<"_HTML_";
  Puerto:
  <input name="ID_Puerto" type="text" id="ID_Puerto">
_HTML_
}

print<<"_HTML_";
<INPUT type=submit value=Aceptar name=Submit2></form></P>
  <P>&nbsp;</P></TD></TR>
  <TR>
    <TD bgColor=#9999ff>&nbsp;</TD></TR>
  <TR>
    <TD bgColor=#ffffff>&nbsp;</TD></TR></TBODY></TABLE>
<DIV align=center>
_HTML_

$Pto="sss" if ($Tipo_Busq == 2);

if ($IP && $Pto && !($no_pinea)) {
  print<<"_HTML_";
  <TABLE width="56%" border=0>
    <TBODY>
      <TR>
        <TD align=middle width="23%" bgColor=#d3dce3> <DIV
align=center><STRONG><U><font face="Arial, Helvetica">IP</font></U>

```

```

        </STRONG></DIV></TD>
        <TD width="33%" align="middle" bgColor=#d3dce3> <DIV
align="center"><STRONG><U><font face="Arial,
Helvetica">MAC</font></U></STRONG></DIV></TD>
        <TD width="21%" align="middle" bgColor=#d3dce3><div
align="center"><strong><U>Switch</U></strong></div></TD>
        <TD align="middle" width="23%" bgColor=#d3dce3> <DIV
align="center"><STRONG><U><font face="Arial,
Helvetica">Puerto</font></U></STRONG></DIV></TD>
    </TR>
<TR>

```

```

_HTML_

```

```

if ($Tipo_Busq != 2) {
    print<<"_HTML_";

        <TD noWrap bgColor=#cccccc> <DIV align="left"><font face="Arial,
Helvetica">$IP</font>
        </DIV></TD>
        <TD noWrap bgColor=#cccccc> <DIV align="left"><font face="Arial,
Helvetica">$Tabla_ARP2{$IP}</font>
        </DIV></TD>
        <TD noWrap bgColor=#cccccc><a
href="http://10.1.4.151:8880/OvCgi/aut3.pl?ID_switch=$SWITCH&Name=$usuari
oW">$SWITCH</a>
        </TD>
        <TD noWrap bgColor=#cccccc><a
href="http://10.1.4.151:8880/OvCgi/aut3.pl?ID_switch=$SWITCH&ID_Puerto=$S
lo/$Pto&Name=$usuarioW"><div align="center">$Slo/$Pto</a></TD>
_HTML_
}

```

```

else {

    $COLOR[1]="#cccccc";
    $COLOR[0]="#eeeeee";
    $i=1;

    %IP_en_Puerto=Invertir(%Tabla_ARP);
    %H=Invertir(%HEX);

    foreach $b (keys(%Slot)) {
        @VEC_AUX=split("[\.]", $b);
        $MAC_AUX{$b}=$H{$VEC_AUX[0]} . "-" . $H{$VEC_AUX[1]} . "-" .
        $H{$VEC_AUX[2]} . "-" . $H{$VEC_AUX[3]} . "-" . $H{$VEC_AUX[4]} . "-" .
        $H{$VEC_AUX[5]};
        $Pto_actual=$Slot{$b} . "/" . $Puerto{$b};
        if ($PUERTO eq $Pto_actual || $PUERTO eq "t") {
            print<<"_HTML_";

```

```

        <TD noWrap bgColor=$COLOR[$i%2]> <DIV align=left><font
face="Arial, Helvetica">$IP_en_Puerto{$b}</font>
        </DIV></TD>
        <TD noWrap bgColor=$COLOR[$i%2]> <DIV align=left><font
face="Arial, Helvetica">$MAC_AUX{$b}</font>
        </DIV></TD>
        <TD noWrap bgColor=$COLOR[$i%2]><a
href="http://10.1.4.151:8880/OvCgi/aut3.pl?ID_switch=$SWITCH&Name=$usuari
oW">$SWITCH</a>
        </TD>
        <TD noWrap bgColor=$COLOR[$i%2]><a
href="http://10.1.4.151:8880/OvCgi/aut3.pl?ID_switch=$SWITCH&ID_Puerto=$P
to_actual&Name=$usuarioW"><div align="center">$Pto_actual</a></TD></TR>
    <TR>
    _HTML_
        $i++;
    }
}

    print<<"_HTML_";
    </TR>
    <TR>
    </TBODY>
    </TABLE>

_HTML_
}

if ($IP && $no_pinea) {
print<<"_HTML_";
    <TABLE width="56%" border=0>
        <TBODY>
            <TR>
                <TD align=middle bgColor=#d3dce3> <DIV align=center><strong><font
face="Arial, Helvetica">La
                    direcci&oacute;n buscada no est&aacute;
activa</font></strong>
                    <hr>
                    <font color="#FF0000" face="Arial, Helvetica">Favor intente
de nuevo
                    con otra IP</font> </DIV>
                </TD>
            </TR>
        </TBODY>
    </TABLE>

_HTML_

```

```
}  
  
print<<"_HTML_";  
  <p>&nbsp;</p>  
  <P> </P></DIV></FONT></BODY></HTML>  
_HTML_  
  
  }  
else  
  {  
  print $query->header;  
  print $query->start_html(-title=>'No Autorizado');  
  print $query->h1('No esta autorizado para ver esta pagina');  
  print $query->end_html;  
  }
```


B.4 Página inicial de enrutadores Nortel

```

#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<html>\n";
print "<head>\n";
print "<script type=\"text/javascript\">\n";
print "function openwindow()\n";
print "(\n";
print ")\n";
print "</script>\n";
print "</head>\n";
print "<body background=\"/OvDocs/images/Globe.gif\">\n";
print "<TITLE>Mantenimiento y Monitoreo de la Red</TITLE>\n";
print "  <FORM METHOD=POST
ACTION=\"http://10.1.4.151:8880/OvCgi/aut4.pl\" TARGET=\"_blank\" >\n";
print "<table width=\"900\" border=\"0\" cellspacing=\"0\"
cellpadding=\"0\">";
print "<tr>\n ";
print "<td width=\"170\" height=\"50\">&nbsp;</td>\n";
print "<td width=\"170\" height=\"50\">&nbsp;</td>\n";
print "</tr><!td>\n";
print "<tr>\n ";
print "<td \nwidth=\"150\" height=\"50\">&nbsp;</td>";
print "<td width=\"150\" height=\"50\">&nbsp;</td>\n";
print "</tr><!td>\n";
print "<tr>\n";
print "<td width=\"150\" height=\"50\">&nbsp;</td>\n";
print "<td width=\"150\" height=\"50\">&nbsp;</td></tr>\n";
print "<tr><td>&nbsp;</td><th align=\"left\">\n";
print "<font size=+3>Web-Manager</font></th></tr><tr><td
align=\"right\">\n";
print "  Usuario:</td><td>\n";
print "<input name=\"Name\" size=20 maxlength=50></td></tr>\n";
print "  <tr><td align=\"right\">\n";
print "  Password:</td><td>\n";
print "  <input type=\"password\" name=\"Password\" size=20
maxlength=50>\n";
print "<input name=\"ID_switch\" type=\"hidden\" id=\"ID_switch\"
value=\"10.2.0.1\">\n";
print "  </td></tr><tr><td width=\"150\"
height=\"10\">&nbsp;</td></tr><tr><td>&nbsp;</td><td align=\"left\">\n";
print "  <INPUT TYPE=submit value=\"Aceptar\"
onclick=\"openwindow();javascript:window.close();\">\n";
print "<!/td><!td>\n";
print "  <INPUT TYPE=reset value=\"Reiniciar\"></td></tr>\n";
print "  </FORM>\n";
print "</table>\n";
print "</body>\n";
print "</html>\n";

```

B.5 Marco Principal de la Aplicación de Administración de Enrutadores

```

#!/usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLANDesc=split("\n",$Desc);
    foreach $b (@AVLANDesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]",$inst_raw);
        $DESCVlan_2{$C[2]}=join(" ",@B);
    }

    delete $DESCVlan_2{""};
    return %DESCVlan_2;
}

#

$IP_switch = $query->param('IP_switch');
$IP_switch = "10.2.131.2" if (!$IP_switch);
$Name = $query->param('Name');
$Name = "oper2" if (!$Name);

$passwordW = $query->param('Password');

open (USERS, '/etc/usuarios.wv') || die();

```

```
$i = 1;
$text=<USERS>;
while ($text) {
    chop($text);
    @usu = split(" ", $text);
    $usuario[$i++] = $usu[0];
    $password{$usu[0]} = $usu[1];
    $text=<USERS>;
}
close(USERS);

#
#
# Lectura de la IP que solicita
#

$IPs = $ENV{'REMOTE_ADDR'};

#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#

open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);

$i = 1;

$autentUS = 0;
$autentIP = 0;

foreach $b (@usuario) {
    if (! $b) {next;}
    if (! $usuarioW) {last;}
    if (! $passwordW) {last;}
```

```

        if ($b eq $usuarioW && $password{$b} eq $passwordW) {
            $autentUS = 1;
            last;}
    }

#
# Revision que la IP este en la lista de acceso
#

foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
    }
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/aut3.pl" ||
$ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Buscar.pl") {
    $autentUS=$autentIP=1;
}
if ($IPs eq "10.1.4.151" || $IPs eq "10.1.128.146" || $IPs eq
"10.2.128.74") {$autentUS = $autentIP =1;}

$MISC=`/opt/OV/bin/snmpwalk $IP_switch system`;
@APM=split("\n", $MISC);
foreach $b (@APM) {
    if ($b =~ "sysContact" && $b =~ "STRING") {
        $sysContact="$sysContact" . "\n" ."$b";
    }
    if ($b =~ "sysName" && $b =~ "STRING") {
        $sysName=$b;
    }
    if ($b =~ "sysLocation" && $b =~ "STRING") {
        $sysLocation="$sysLocation" . "\n" . "$b";
    }
}

%HsysName=Genera_Hash_Simple($sysName,5);
%HDir_Reg=Genera_Hash_Simple($sysContact,5);
%HLugar=Genera_Hash_Simple($sysLocation,5);

$sysName=$HsysName{"0"};
$Dir_Reg=$HDir_Reg{"0"};
$Lugar=$HLugar{"0"};

$IP_switch =~ tr/[a-z]/[A-Z]/;

```

```

$IP_NOM=`grep \" $IP_switch\" /etc/hosts`;

@NOM_IP=split(" ", $IP_NOM);
$IP_NOM=$NOM_IP[0];

$IP_switch = (!($IP_NOM)) ? $IP_switch : $IP_NOM;

if ($autentUS && $autentIP) {

    print $query->header;

print<<"_HTML_";
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>Administracion de Enrutadores</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<frameset rows="160,*" cols="*" frameborder="NO" border="0"
framespacing="0">
  <frame
src="Presentacion_y_Busqueda.pl?IP_switch=$IP_switch&Name=$Name&sysName=$
sysName&Dir_Reg=$Dir_Reg&Lugar=$Lugar" name="topFrame" scrolling="NO"
noresize >
  <frameset rows="*" cols="150,*" framespacing="0" frameborder="YES"
border="0">
    <frame
src="Accesos_Dir.pl?IP_switch=$IP_switch&Name=$Name&sysName=$sysName&Dir_
Reg=$Dir_Reg&Lugar=$Lugar" name="leftFrame" scrolling="NO" noresize>
      <frame src="Mai.pl" name="mainFrame">
    </frameset>
  </frameset>
</frameset><body>

</body></noframes>
</html>

_HTML_
}
else
{
  print $query->header;
  print $query->start_html(-title=>'No Autorizado');
  print $query->h1('No esta autorizado para ver esta pagina');
  print $query->end_html;
}

```

B.6 Estado de los Circuitos IP

```

#!/usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLAndesc=split("\n",$Desc);
    foreach $b (@AVLAndesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[.]",$inst_raw);
        $DESCVlan_2{$C[7]}=join(" ",@B);
    }

    delete $DESCVlan_2{" "};
    return %DESCVlan_2;
}

#
#
#

sub Invertir (

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }
}

```

```

    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
}

#
#
# Funcion para generar hash Dobles
#
#
sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.\]",$inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--)
    ) {
        pop(@C);
    }

    my $inst=join($Tipo,@C);

```

```

        $DESCVLan_2{$inst}=$Valor;
    }

    return %DESCVLan_2;
}

#
#

sub Genera_Hash_HEX($$$$$) (

#
# Vectores de conversion a Decimal
#

for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Separador) = $_[2];
my ($Tipo) = $_[4];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {
    delete $DESCVLan_2{$re};
}

my @APORTStatus=split("\n", $DDesc);

```



```

foreach $b (@APORTStatus) {
  my @B=split(" ",$b);
  if ($b =~ "OCTET STRING-") {
    my $inst_raw=shift(@B);

    my @C=split("[\.",$inst_raw);

    for ($i=1; $i<$Repeticiones1; $i++) {
      shift(@C);
    }

    $inst1=join($Separador,@C);
    $DESCVLan_2{$inst1}="";
    next;
  }
  else {
    $B[1] = $HEX{$B[1]} if ($Tipo);
    $Valor=$B[1] . " ";
    for ($i=2; $i<17; $i++) {
      $B[$i] = $HEX{$B[$i]} if ($Tipo);
      $Valor=$Valor . $B[$i] . " ";
    }
    $Valor =~ s/ +//g;
    $Valor =~ s/ -- //g;
    $Valor =~ s/--//g;
    $Valor =~ s/ $//g;
    $Valor =~ s/ /$_[3]/g;
  }
  $DESCVLan_2{$inst1}=$DESCVLan_2{$inst1} . $Valor;
}

delete $DESCVLan_2{" "};

return %DESCVLan_2;
}

```

```
# Aqui terminan las subrutinas
```

```

#
#
#
# Autenticacion
#

```

```

#

#
# Lectura de la IP que solicita
#
$IPs = $ENV{'REMOTE_ADDR'};
#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/sic.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

@IP_Instancias = split(" ", $query->param('Lista_Instancias'));

$IP_switch = $query->param('IP_switch');

```

```

$Lugar = $query->param('sysName');
$Name = $query->param('Name');

foreach $b (@IP_Instancias) {

    $Disable{$b} = $query->param("Disable_{$b}");
    $Disable_org{$b} = $query->param("_Disable_{$b}");
    $IP_Costo_org{$b} = $query->param("_IP_Costo_{$b}");
    $IP_Costo{$b} = $query->param("IP_Costo_{$b}");
    $Instanc{$b} = $query->param("Instanc_{$b}");
}

$checked{"enabled"} = "";
$checked{"disabled"} = "checked";

#
#
# Archivo de control de cambios
#
#

open(LOG_CAMBIOS, '>>/tmp/Control_R.txt');

$FECHA=`date`;
chop($FECHA);

#
#
# Logica de cambios en la configuracion
#
#
foreach $b (@IP_Instancias) {

    if ($Disable_org{$b} eq "enabled" && $Disable{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch
wfIpIntfCfgEntry.2.{$b} integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name deshabilito
la interface IP Instanc{$b} en el enrutador $IP_switch ($Lugar)\n\n";
    }

    if ($Disable_org{$b} eq "disabled" && !($Disable{$b})) {
        system("/opt/OV/bin/snmpset $IP_switch
wfIpIntfCfgEntry.2.{$b} integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name habilito la
interface IP $Instanc{$b} en el enrutador $IP_switch ($Lugar)\n\n";
    }
}

```

```

        if (!(($IP_Costo_org{$b} eq $IP_Costo{$b})) {
            system("/opt/OV/bin/snmpset $IP_switch wfIpIntfCfgCost.$b
integer $IP_Costo{$b}");
            print LOG_CAMBIOS "$FECHA : El Usuario $Name paso el costo
de $IP_Costo_org{$b} a $IP_Costo{$b} en la instancia $Instanc{$b} del
enrutador $IP_switch ($Lugar)\n\n";

        }
    }

close(LOG_CAMBIOS);

#
#
# Carga de la informacion del equipo
#
#
@uma_guma=("wfIpIntfCfgEnable", "wfIpIntfCfgState",
          "wfIpIntfCfgMask", "wfIpIntfCfgCost", "wfCircuitName");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);
foreach $b (@APM) {
    if ($b =~ "wfIpIntfCfgEnable" && $b =~ "INTEGER") {
        $wfIpIntfCfgEnable="$wfIpIntfCfgEnable" . "\n" . "$b";
    }
    if ($b =~ "wfIpIntfCfgState" && $b =~ "INTEGER") {
        $wfIpIntfCfgState="$wfIpIntfCfgState" . "\n" . "$b";
    }
    if ($b =~ "wfIpIntfCfgMask" && $b =~ "IpAddress") {
        $wfIpIntfCfgMask="$wfIpIntfCfgMask" . "\n" . "$b";
    }
    if ($b =~ "wfIpIntfCfgCost" && $b =~ "INTEGER") {
        $wfIpIntfCfgCost="$wfIpIntfCfgCost" . "\n" . "$b";
    }
    if ($b =~ "wfCircuitName" && $b =~ "STRING") {
        $wfCircuitName="$wfCircuitName" . "\n" . "$b";
    }
}

%Enabl=Genera_Hash_Doble($wfIpIntfCfgEnable,10,14,".");
%State=Genera_Hash_Doble($wfIpIntfCfgState,10,14,".");
%Masca=Genera_Hash_Doble($wfIpIntfCfgMask,10,14,".");
%Costo=Genera_Hash_Doble($wfIpIntfCfgCost,10,14,".");
%CName=Genera_Hash_Simple($wfCircuitName,5);

```

```

delete $Enabl{""};

@List_Inst = keys(%Enabl);

foreach $b (@List_Inst) {
    @IP_Circ=split("[\.]", $b);
    $Num_Circ{$b}=pop(IP_Circ);
    $IP_Circ{$b}=join(".", @IP_Circ);
}

print $query->header;

print<<"_HTML_";

<HTML>
<HEAD>
<TITLE>Bay Networks Enrutador - System Information</TITLE>

</HEAD>

<body bgcolor=#C0C0C0>
<center>
    <h3>Circuitos IP</h3>
</center>
<center>
<table border=3>
    <tr>
        <form name="Formul_SIC" method="post" action="sic.pl">
            <input type="hidden" name="IP_switch" id="IP_Switch"
value="$IP_switch">
            <input type="hidden" name="Lista_Instancias"
id="Lista_Instancias" value="@List_Inst">
            <input type="hidden" name="sysName" id="sysName" value="$Lugar">
            <input type="hidden" name="Name" id="Name" value="$Name">
            <th width="239" align=left>Nombre</th>
            <th width="62" align=left>No. de Circuito</th>
            <th width="135" align=left>Estado</th>
            <th width="109" align=left>Direcci&oacute;n IP</th>
            <th width="117" align=left>M&aacute;scara</th>
            <th width="64" align=left>Costo</th>
        <tr>

_HTML_

foreach $b (@List_Inst) {

print<<"_HTML_";

        <th align=left>$CName{$Num_Circ{$b}}</th>
        <th align=left>$Num_Circ{$b}</th>

```



```
}  
  
else  
  
    {  
    print $query->header;  
    print $query->start_html(-title=>'No Autorizado');  
    print $query->h1('No esta autorizado para ver esta pagina');  
    print $query->end_html;  
    }
```

B.7 Interfaces Sincrónicas

```

#! /usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLAnDesc=split("\n",$Desc);
    foreach $b (@AVLAnDesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.",$inst_raw);
        $DESCVlan_2{$C[7]}=join(" ",@B);
    }

    delete $DESCVlan_2{""};
    return %DESCVlan_2;
}

#
#
#

sub Invertir {

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }
}

```



```

    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
}

#
#
# Funcion para generar hash Dobles
#
#

sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.]", $inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--)
    ) {
        pop(@C);
    }

    my $inst=join($Tipo,@C);

    $DESCVLan_2{$inst}=$Valor;
}

```

```

    }

    return %DESCVLan_2;
}

#
#

sub Genera_Hash_HEX($$$$) {

#
# Vectores de conversion a Decimal
#

for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Separador) = $_[2];
my ($Tipo) = $_[4];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {
    delete $DESCVLan_2{$re};
}

my @APORTStatus=split("\n", $DDesc);

foreach $b (@APORTStatus) {

```

```

my @B=split(" ",$b);
if ($b =~ "OCTET STRING-") {
    my $inst_raw=shift(@B);

    my @C=split("[\.]", $inst_raw);

    for ($i=1; $i<$Repeticiones1; $i++) {
        shift(@C);
    }

    $inst1=join($Separador,@C);
    $DESCVLan_2{$inst1}="";
    next;
}
else {
    $B[1] = $HEX{$B[1]} if ($Tipo);
    $Valor=$B[1] . " ";
    for ($i=2; $i<17; $i++) {
        $B[$i] = $HEX{$B[$i]} if ($Tipo);
        $Valor=$Valor . $B[$i] . " ";
    }
    $Valor =~ s/ +//g;
    $Valor =~ s/ -- //g;
    $Valor =~ s/--//g;
    $Valor =~ s/ $//g;
    $Valor =~ s/ /$_[3]/g;
}
$DESCVLan_2{$inst1}=$DESCVLan_2{$inst1} . $Valor;
}

delete $DESCVLan_2{" "};

return %DESCVLan_2;
}

```

Aqui terminan las subrutinas

```

#
#
#
# Autenticacion
#
#

```

```

#
# Lectura de la IP que solicita
#

$IIPs = $ENV{'REMOTE_ADDR'};

#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#

open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IIPs) {last;}
    if ($b eq $IIPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/ssb.pl") {
    $autentUS=1;
}
if ($IIPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

@IP_Instancias = split(" ", $query->param('Lista_Instancias'));

```

```

$IP_switch = $query->param('IP_switch');
$Lugar = $query->param('sysName');
$Name = $query->param('Name');

foreach $b (@IP_Instancias) {

    $Disable{$b} = $query->param("Disable_{$b}");
    $Disable_org{$b} = $query->param("_Disable_{$b}");
    $Instanc{$b} = $query->param("Instanc_{$b}");

    #${A}{$b}= $b . $Disable{$b} . $Instanc{$b} . $Disable_org{$b};

}

#
#
# Archivo de control de cambios
#
#

open(LOG_CAMBIOS, '>>/tmp/Control_R.txt');

$FECHA=`date`;
chop($FECHA);

#
#
# Logica de cambios en la configuracion
#
#
foreach $b (@IP_Instancias) {

    if ($Disable_org{$b} eq "enabled" && $Disable{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch wfSyncEntry.2.{$b
integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name deshabilito
la interface $b en el enrutador $IP_switch ($Lugar)\n\n";
    }

    if ($Disable_org{$b} eq "disabled" && !($Disable{$b})) {
        system("/opt/OV/bin/snmpset $IP_switch wfSyncEntry.2.{$b
integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name habilito la
interface $Instanc{$b} en el enrutador $IP_switch ($Lugar)\n\n";
    }

}
}

```

```

close(LOG_CAMBIOS);

$checked{"enabled"} = "";
$checked{"disabled"} = "checked";

#
#
# Carga de la informacion del equipo
#

@uma_guma=("wfSyncDisable", "wfSyncState",
           "wfSyncCct", "wfSyncMtu", "wfSyncMadr", "wfSyncWanProtocol",
           "wfSyncLineNumber", "wfCircuitName");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);

$encontrado0=0;

foreach $b (@APM) {
    if ($b =~ "wfSyncDisable" && $b =~ "INTEGER") {
        $wfSyncDisable="$wfSyncDisable" . "\n" . "$b";
    }
    if ($b =~ "wfSyncState" && $b =~ "INTEGER") {
        $wfSyncState="$wfSyncState" . "\n" . "$b";
    }
    if ($b =~ "wfSyncCct" && $b =~ "INTEGER") {
        $wfSyncCct="$wfSyncCct" . "\n" . "$b";
    }
    if ($b =~ "wfSyncMtu" && $b =~ "INTEGER") {
        $wfSyncMtu="$wfSyncMtu" . "\n" . "$b";
    }
    if ($b =~ "wfSyncWanProtocol" && $b =~ "INTEGER") {
        $wfSyncWanProtocol="$wfSyncWanProtocol" . "\n" . "$b";
    }
    if ($b =~ "wfSyncLineNumber" && $b =~ "INTEGER") {
        $wfSyncLineNumber="$wfSyncLineNumber" . "\n" . "$b";
    }
    if ($b =~ "wfCircuitName" && $b =~ "STRING") {
        $wfCircuitName="$wfCircuitName" . "\n" . "$b";
    }
    if ($b =~ "wfSyncMadr" && $b =~ "OCTET STRING" || $encontrado0) {
        $wfSyncMadr="$wfSyncMadr" . "\n" . "$b";
        $encontrado0=1;
        $encontrado0=0 if ($b =~ "--- -- --");
    }
}

```

```

    }
}

%Enabl = Genera_Hash_Doble($wfSyncDisable,7,8,".");
%State = Genera_Hash_Doble($wfSyncState,7,8,".");
%Circu = Genera_Hash_Doble($wfSyncCct,7,8,".");
%MaxTU = Genera_Hash_Doble($wfSyncMtu,7,8,".");
%WAN_P = Genera_Hash_Doble($wfSyncWanProtocol,7,8,".");
%LineN = Genera_Hash_Doble($wfSyncLineNumber,7,8,".");
%MAC_A = Genera_Hash_HEX($wfSyncMadr,7,".", "-", "");
%CName = Genera_Hash_Simple($wfCircuitName,5);

delete $Enabl{""};

@List_Inst = keys(%Enabl);

foreach $b (@List_Inst) {
    @Slot_Conn=split("[\\.]", $b);
    $Slot{$b}=$Slot_Conn[0];
    $Conn{$b}=$Slot_Conn[1];
    $MAC_A{$b} =~ tr/[a-z]/[A-Z]/;
}

print $query->header;

print<<"_HTML_";

<HTML>
<HEAD>
<TITLE>Bay Networks Enrutador - System Information</TITLE>

</HEAD>

<body bgcolor=#C0C0C0>
<center>
    <h3>Puertos Sincr&oacute;nicos</h3>
</center>
<center>
<table border=3>
    <tr>
        <th width="26" align=left><div align="center">Slot</div></th>
        <th width="35" align=left><div align="center">Conn</div></th>
        <form name="Formul_SSB" method="post" action="ssb.pl">

```



```
</BODY>  
</HTML>
```

```
_HTML_
```

```
}
```

```
else
```

```
{  
  print $query->header;  
  print $query->start_html(-title=>'No Autorizado');  
  print $query->h1('No esta autorizado para ver esta pagina');  
  print $query->end_html;  
}
```

B.8 Estadísticas de los Puertos Sincrónicos

```

#! /usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLAnDesc=split("\n",$Desc);
    foreach $b (@AVLAnDesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]",$inst_raw);
        $DESCVlan_2{$C[7]}=join(" ",@B);
    }

    delete $DESCVlan_2{" "};
    return %DESCVlan_2;
}

#
#
#

sub Invertir {

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {

```

```

        delete $DESCVLan_2{$re};
    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
}

#
#
# Funcion para generar hash Dobles
#
#
#
sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.]",$inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--
) {
            pop(@C);
        }

        my $inst=join($Tipo,@C);

```

```

        $DESCVLan_2{$inst}=$Valor;
    }

    return %DESCVLan_2;
}

#
#

sub Genera_Hash_HEX($$$$$) {

#
# Vectores de conversion a Decimal
#

for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Separador) = $_[2];
my ($Tipo) = $_[4];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {
    delete $DESCVLan_2{$re};
}

my @APORTStatus=split("\n", $DDesc);

```

```

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    if ($b =~ "OCTET STRING-") {
        my $inst_raw=shift(@B);

        my @C=split("[\\.]", $inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        $inst1=join($Separador,@C);
        $DESCVLan_2{$inst1}="";
        next;
    }
    else (
        $B[1] = $HEX{$B[1]} if ($Tipo);
        $Valor=$B[1] . " ";
        for ($i=2; $i<17; $i++) {
            $B[$i] = $HEX{$B[$i]} if ($Tipo);
            $Valor=$Valor . $B[$i] . " ";
        }
        $Valor =~ s/ +//g;
        $Valor =~ s/ -- //g;
        $Valor =~ s/--//g;
        $Valor =~ s/ $//g;
        $Valor =~ s/ /$_[3]/g;
    }
    $DESCVLan_2{$inst1}=$DESCVLan_2{$inst1} . $Valor;
}

delete $DESCVLan_2{" "};

return %DESCVLan_2;
}

```

```
# Aqui terminan las subrutinas
```

```

#
#
#
# Autenticacion
#

```

```

#

#
# Lectura de la IP que solicita
#

$IPs = $ENV{'REMOTE_ADDR'};

#
# Lectura del archivo de IP para ver si estan las adecuadas
#

open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/sss.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

$IP_switch = $query->param('IP_switch');

```

```

$Lugar = $query->param('sysName');
$Name = $query->param('Name');

@IP_Instancias = split(" ", $query->param('Lista_Instancias'));

foreach $b (@IP_Instancias) {

    $TxOct_{$b} = $query->param("TxOct_{$b}");
    $TxFra_{$b} = $query->param("TxFra_{$b}");
    $RxOct_{$b} = $query->param("RxOct_{$b}");
    $RxFra_{$b} = $query->param("RxFra_{$b}");
    $Error{$b} = $query->param("Error_{$b}");
}

#
#
# Carga de la informacion del equipo
#
#

@uma_guma=("wfSyncRxOctets", "wfSyncRxFrames", "wfSyncTxOctets",
           "wfSyncTxFrames", "wfSyncRxErrors", , "wfSyncTxErrors",
           "wfCircuitName", "wfSyncCct");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);

$encontrado0=0;

foreach $b (@APM) {
    if ($b =~ "wfSyncRxOctets" && $b =~ "Counter") {
        $wfSyncRxOctets="$wfSyncRxOctets" . "\n" . "$b";
    }
    if ($b =~ "wfSyncRxFrames" && $b =~ "Counter") {
        $wfSyncRxFrames="$wfSyncRxFrames" . "\n" . "$b";
    }
    if ($b =~ "wfSyncTxOctets" && $b =~ "Counter") {
        $wfSyncTxOctets="$wfSyncTxOctets" . "\n" . "$b";
    }
    if ($b =~ "wfSyncTxFrames" && $b =~ "Counter") {
        $wfSyncTxFrames="$wfSyncTxFrames" . "\n" . "$b";
    }
    if ($b =~ "wfSyncRxErrors" && $b =~ "Counter") {
        $wfSyncRxErrors="$wfSyncRxErrors" . "\n" . "$b";
    }
    if ($b =~ "wfSyncTxErrors" && $b =~ "Counter") {
        $wfSyncTxErrors="$wfSyncTxErrors" . "\n" . "$b";
    }
}

```

```

    }
    if ($b =~ "wfSyncCct" && $b =~ "INTEGER") {
        $wfSyncCct="$wfSyncCct" . "\n" . "$b";
    }
    if ($b =~ "wfCircuitName" && $b =~ "STRING") {
        $wfCircuitName="$wfCircuitName" . "\n" . "$b";
    }
}

%RxOct = Genera_Hash_Doble($wfSyncRxOctets,7,8,".");
%RxFra = Genera_Hash_Doble($wfSyncRxFrames,7,8,".");
%TxOct = Genera_Hash_Doble($wfSyncTxOctets,7,8,".");
%TxFra = Genera_Hash_Doble($wfSyncTxFrames,7,8,".");
%RxErr = Genera_Hash_Doble($wfSyncRxErrors,7,8,".");
%TxErr = Genera_Hash_Doble($wfSyncTxErrors,7,8,".");
%Circu = Genera_Hash_Doble($wfSyncCct,7,8,".");
%CName = Genera_Hash_Simple($wfCircuitName,5);

delete $RxOct("");

@List_Inst = keys(%RxOct);

foreach $b (@List_Inst) {
    @Slot_Conn=split("[\.]", $b);
    $Slot{$b}=$Slot_Conn[0];
    $Conn{$b}=$Slot_Conn[1];
    $Errores{$b}=$TxErr{$b}+$RxErr{$b};

    $dTxDct{$b} = $TxOct{$b} - $TxOct_{$b} if ($TxOct_{$b});
    $dRxOct{$b} = $RxOct{$b} - $RxOct_{$b} if ($RxOct_{$b});
    $dTxFra{$b} = $TxFra{$b} - $TxFra_{$b} if ($TxFra_{$b});
    $dRxFra{$b} = $RxFra{$b} - $RxFra_{$b} if ($RxFra_{$b});
    $dError{$b} = $Errores{$b} - $Error{$b} if ($Error{$b});
}

print $query->header;

print<<"_HTML_";

<HTML>
<HEAD>
<TITLE>Bay Networks Enrutador - System Information</TITLE>

</HEAD>

```



```

<body bgcolor=#C0C0C0>
<center>
  <h3>Puertos Sincronizados</h3>
</center>
<center>
<table border=3>
  <tr>
    <th width="26" align=left><div align="center">Slot</div></th>
    <th width="35" align=left><div align="center">Conn</div></th>
    <form name="Formul_SSB" method="post" action="sss.pl">
      <input type="hidden" name="IP_switch" id="IP_Switch"
value="$IP_switch">
      <input type="hidden" name="Lista_Instancias" id="Lista_Instancias"
value="@List_Inst">

    <th width="255" align=left><div align="center">Nombre</div></th>
    <th width="128" align=left>Bytes Recibidos</th>
    <th width="128" align=left><div align="center">Tramas
Recibidas</div></th>
    <th width="128" align=left><div align="center">Bytes
Transmitidos</div></th>
    <th width="128" align=left><div align="center">Tramas
Transmitidas</div></th>
    <th width="128" align=left><div align="center">Errores</div></th>

  _HTML_

  foreach $b (@List_Inst) {

  print<<"_HTML_";

  <tr>
    <th align=left>${Slot}{$b}</th>
    <th align=left>${Conn}{$b}</th>
    <th align=left>${CName}{${Circu}{$b}}</th>
    <th align=left>${RxOct}{$b}<br>Dif = ${dRxOct}{$b}</th>
    <th align=left>${RxFra}{$b}<br>Dif = ${dRxFra}{$b}</th>
    <th align=left>${TxOct}{$b}<br>Dif = ${dTxOct}{$b}</th>
    <th align=left>${TxFra}{$b}<br>Dif = ${dTxFra}{$b}</th>
    <th align=left>${Errores}{$b}<br>Dif = ${dError}{$b}</th>
    <input type="hidden" name="RxOct_${b}" id="RxOct_${b}"
value="${RxOct}{$b}">
    <input type="hidden" name="TxOct_${b}" id="TxOct_${b}"
value="${TxOct}{$b}">
    <input type="hidden" name="RxFra_${b}" id="RxFra_${b}"
value="${RxFra}{$b}">
    <input type="hidden" name="TxFra_${b}" id="TxFra_${b}"
value="${TxFra}{$b}">
    <input type="hidden" name="Error_${b}" id="Error_${b}"
value="${Errores}{$b}">

```

```
    _HTML_
}

print<<"_HTML_";

</table>
<p align="center">&nbsp;
  <input type="submit" value="Actualizar"></form>

</BODY>
</HTML>

    _HTML_

}

else

    {
    print $query->header;
    print $query->start_html(-title=>'No Autorizado');
    print $query->hl('No esta autorizado para ver esta pagina');
    print $query->end_html;
    }
```

B.9 Estadísticas de Frame Relay

```

#! /usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLAnDesc=split("\n",$Desc);
    foreach $b (@AVLAnDesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ", $b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]", $inst_raw);
        $DESCVlan_2{$C[7]}=join(" ", @B);
    }

    delete $DESCVlan_2{""};
    return %DESCVlan_2;
}

#
#
#

sub Invertir {

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }
}

```

```

    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
)

#
#
# Funcion para generar hash Dobles
#

sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.",$inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--)
    ) {
        pop(@C);
    }

    my $inst=join($Tipo,@C);

    $DESCVLan_2{$inst}=$Valor;
}

```

```

        return %DESCVLan_2;
    }

#
#

sub Genera_Hash_HEX($$$$) {

#
# Vectores de conversion a Decimal
#

for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Separador) = $_[2];
my ($Tipo) = $_[4];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {
    delete $DESCVLan_2{$re};
}

my @APORTStatus=split("\n", $DDesc);

foreach $b (@APORTStatus) {
    my @B=split(" ", $b);

```

```

if ($b =~ "OCTET STRING-") {
    my $inst_raw=shift(@B);

    my @C=split("[\\.]", $inst_raw);

    for ($i=1; $i<$Repeticiones1; $i++) {
        shift(@C);
    }

    $inst1=join($Separador, @C);
    $DESCVlan_2{$inst1}="";
    next;
}
else {
    $B[1] = $HEX{$B[1]} if ($Tipo);
    $Valor=$B[1] . " ";
    for ($i=2; $i<17; $i++) {
        $B[$i] = $HEX{$B[$i]} if ($Tipo);
        $Valor=$Valor . $B[$i] . " ";
    }
    $Valor =~ s/ +//g;
    $Valor =~ s/ -- //g;
    $Valor =~ s/--//g;
    $Valor =~ s/ $//g;
    $Valor =~ s/ /$_[3]/g;
}
$DESCVlan_2{$inst1}=$DESCVlan_2{$inst1} . $Valor;
}

delete $DESCVlan_2{" "};

return %DESCVlan_2;
}

```

```
# Aqui terminan las subrutinas
```

```

#
#
#
# Autenticacion
#
#

```

```

#
# Lectura de la IP que solicita
#

$IPs = $ENV{'REMOTE_ADDR'};

#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#

open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/sfs.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

$IP_switch = $query->param('IP_switch');
$Lugar = $query->param('sysName');
$Name = $query->param('Name');

```

```

@IP_Instancias = split(" ", $query->param('Lista_Instancias'));

foreach $b (@IP_Instancias) {

    $TxOct_{$b} = $query->param("TxOct_{$b}");
    $TxFra_{$b} = $query->param("TxFra_{$b}");
    $RxOct_{$b} = $query->param("RxOct_{$b}");
    $RxFra_{$b} = $query->param("RxFra_{$b}");
    $BECNs_{$b} = $query->param("BECNs_{$b}");
    $FECNs_{$b} = $query->param("FECNs_{$b}");
}

#
#
# Carga de la informacion del equipo
#
#

@uma_guma=("wfFrCircuitDlci", "wfFrCircuitNumber",
"wfFrCircuitSentFrames",
    "wfFrCircuitSentOctets", "wfFrCircuitReceivedFrames",
    "wfCircuitName", "wfFrCircuitReceivedOctets",
    "wfFrCircuitReceivedFECNs", "wfFrCircuitReceivedBECNs");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);

$encontrado0=0;

foreach $b (@APM) {
    if ($b =~ "wfFrCircuitSentFrames" && $b =~ "Counter") {
        $wfFrCircuitSentFrames="$wfFrCircuitSentFrames" . "\n"
        ."$b";
    }
    if ($b =~ "wfFrCircuitDlci" && $b =~ "INTEGER") {
        $wfFrCircuitDlci="$wfFrCircuitDlci" . "\n" . "$b";
    }
    if ($b =~ "wfFrCircuitNumber" && $b =~ "INTEGER") {
        $wfFrCircuitNumber="$wfFrCircuitNumber" . "\n" . "$b";
    }
    if ($b =~ "wfFrCircuitSentOctets" && $b =~ "Counter") {
        $wfFrCircuitSentOctets="$wfFrCircuitSentOctets" . "\n" .
"$b";
    }
}

```



```

    }
    if ($b =~ "wfFrCircuitReceivedFrames" && $b =~ "Counter") {
        $wfFrCircuitReceivedFrames="$wfFrCircuitReceivedFrames" .
"\n" . "$b";
    }
    if ($b =~ "wfSyncTxErrors" && $b =~ "Counter") {
        $wfSyncTxErrors="$wfSyncTxErrors" . "\n" . "$b";
    }
    if ($b =~ "wfFrCircuitReceivedOctets" && $b =~ "Counter") {
        $wfFrCircuitReceivedOctets="$wfFrCircuitReceivedOctets" .
"\n" . "$b";
    }
    if ($b =~ "wfCircuitName" && $b =~ "STRING") {
        $wfCircuitName="$wfCircuitName" . "\n" . "$b";
    }
    if ($b =~ "wfFrCircuitReceivedFECNs" && $b =~ "Counter") {
        $wfFrCircuitReceivedFECNs="$wfFrCircuitReceivedFECNs" .
"\n" . "$b";
    }
    if ($b =~ "wfFrCircuitReceivedBECNs" && $b =~ "Counter") {
        $wfFrCircuitReceivedBECNs="$wfFrCircuitReceivedBECNs" .
"\n" . "$b";
    }
}

```

```

%RxOct = Genera_Hash_Doble($wfFrCircuitReceivedOctets,9,11,".");
%RxFra = Genera_Hash_Doble($wfFrCircuitReceivedFrames,9,11,".");
%TxOct = Genera_Hash_Doble($wfFrCircuitSentOctets,9,11,".");
%TxFra = Genera_Hash_Doble($wfFrCircuitSentFrames,9,11,".");
%FECNs = Genera_Hash_Doble($wfFrCircuitReceivedFECNs,9,11,".");
%BECNs = Genera_Hash_Doble($wfFrCircuitReceivedBECNs,9,11,".");
%Circu = Genera_Hash_Doble($wfFrCircuitNumber,9,11,".");
%CName = Genera_Hash_Simple($wfCircuitName,5);

```

```
delete $RxOct{""};
```

```
@List_Inst = keys(%RxOct);
```

```

foreach $b (@List_Inst) {
    @Slot_Conn=split("[\.]", $b);
    $DLCI{$b}=$Slot_Conn[2];

    $dTxOct{$b} = $TxOct{$b} - $TxOct_{ $b} if ($TxOct_{ $b});
    $dRxOct{$b} = $RxOct{$b} - $RxOct_{ $b} if ($RxOct_{ $b});
    $dTxFra{$b} = $TxFra{$b} - $TxFra_{ $b} if ($TxFra_{ $b});
    $dRxFra{$b} = $RxFra{$b} - $RxFra_{ $b} if ($RxFra_{ $b});
}

```


B.10 Tabla de Rutas

```

#!/usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLANDesc=split("\n",$Desc);
    foreach $b (@AVLANDesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ", $b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.", $inst_raw);
        $DESCVlan_2{$C[7]}=join(" ", @B);
    }

    delete $DESCVlan_2{""};
    return %DESCVlan_2;
}

#
#
#

sub Invertir {

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {

```

```

        delete $DESCVLan_2{$re};
    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
}

#
#
#  Funcion para generar hash Dobles
#
#

sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.",$inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--)
    ) {
        pop(@C);
    }

    my $inst=join($Tipo,@C);

```

```

        $DESCVLan_2{$inst}=$Valor;
    }

    return %DESCVLan_2;
}

#
#

sub Genera_Hash_HEX($$$$$) {

#
# Vectores de conversion a Decimal
#

for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Separador) = $_[2];
my ($Tipo) = $_[4];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {
    delete $DESCVLan_2{$re};
}

my @APORTStatus=split("\n", $DDesc);

```

```

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    if ($b =~ "OCTET STRING-") {
        my $inst_raw=shift(@B);

        my @C=split("[\\.]", $inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        $inst1=join($Separador,@C);
        $DESCVLan_2{$inst1}="";
        next;
    }
    else {
        $B[1] = $HEX{$B[1]} if ($Tipo);
        $Valor=$B[1] . " ";
        for ($i=2; $i<17; $i++) {
            $B[$i] = $HEX{$B[$i]} if ($Tipo);
            $Valor=$Valor . $B[$i] . " ";
        }
        $Valor =~ s/ +//g;
        $Valor =~ s/ -- //g;
        $Valor =~ s/--//g;
        $Valor =~ s/ $//g;
        $Valor =~ s/ /$_[3]/g;
    }
    $DESCVLan_2{$inst1}=$DESCVLan_2{$inst1} . $Valor;
}

delete $DESCVLan_2{" "};

return %DESCVLan_2;
}

```

```
# Aqui terminan las subrutinas
```

```

#
#
#
# Autenticacion

```

```

#
#

#
# Lectura de la IP que solicita
#

$IPs = $ENV{'REMOTE_ADDR'};
#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/sir.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

```



```

$IP_switch = $query->param('IP_switch');
$Lugar = $query->param('sysName');
$Name = $query->param('Name');

#
#
# Carga de la informacion del equipo
#
#

@uma_guma=("ipRouteMetric1", "ipRouteNextHop", "ipRouteProto",
           "ipRouteAge", "ipRouteMask");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);

$encontrado0=0;

foreach $b (@APM) {
    if ($b =~ "ipRouteMetric1" && $b =~ "INTEGER") {
        $ipRouteMetric1="$ipRouteMetric1" . "\n" . "$b";
    }
    if ($b =~ "ipRouteNextHop" && $b =~ "IpAddress") {
        $ipRouteNextHop="$ipRouteNextHop" . "\n" . "$b";
    }
    if ($b =~ "ipRouteProto" && $b =~ "INTEGER") {
        $ipRouteProto="$ipRouteProto" . "\n" . "$b";
    }
    if ($b =~ "ipRouteAge" && $b =~ "INTEGER") {
        $ipRouteAge="$ipRouteAge" . "\n" . "$b";
    }
    if ($b =~ "ipRouteMask" && $b =~ "IpAddress") {
        $ipRouteMask="$ipRouteMask" . "\n" . "$b";
    }
}

}

%RMetr = Genera_Hash_Doble($ipRouteMetric1,5,8,".");
%RNxtH = Genera_Hash_Doble($ipRouteNextHop,5,8,".");
%RProt = Genera_Hash_Doble($ipRouteProto,5,8,".");
%RoAge = Genera_Hash_Doble($ipRouteAge,5,8,".");
%RMask = Genera_Hash_Doble($ipRouteMask,5,8,".");

```

```

delete $RMetr("");

@List_Inst = sort(keys(%RMetr));

$KNombre=`snmpget $RNxtH{"0.0.0.0"} sysName.0`;
$KNombre=~ s/\s/ /g;
@VNombre=split(" ", $KNombre);
@VNombre[0..4]="";
$KNombre=join(" ",@VNombre);
$KNombre=~ s/^ +//g;
$Nom_NxtH{$b}=$KNombre;

print $query->header;

print<<"_HTML_";

<HTML>
<HEAD>
</HEAD>
<body bgcolor=#C0C0C0>
<center>
  <h3>Tabla de Rutas del Enrutador $Lugar</h3>
</center>
<center>
<table border=3>
  <tr>
    <th width="150" align=left> <div align="center">Destino</div></th>
    <th width="150" align=left><div
align="center">M&aacute;scara</div></th>
    <form name="Formul_SIR" method="post" action="sir.pl">
      <input type="hidden" name="IP_switch" id="IP_Switch"
value="$IP_switch">
      <input type="hidden" name="Lista_Instancias" id="Lista_Instancias"
value="@List_Inst">
      <input type="hidden" name="sysName" id="sysName" value="$Lugar">
      <input type="hidden" name="Name" id="Name" value="$Name">
    <th width="150" align=left><div align="center">Protocolo</div></th>
    <th width="150" align=left><div align="center">Tiempo</div></th>
    <th width="100" align=left><div align="center">Costo</div></th>
    <th width="200" align=left><div align="center">NextHop</div></th>

_HTML_

foreach $b (@List_Inst) {

```


B.11 Configuración del Protocolo RIP

```

#!/usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLANDesc=split("\n",$Desc);
    foreach $b (@AVLANDesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]",$inst_raw);
        $DESCVlan_2{$C[7]}=join(" ",@B);
    }

    delete $DESCVlan_2{""};
    return %DESCVlan_2;
}

#
#
#

sub Invertir {

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }
}

```

```

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVLan_2{$Lucas{$b}}=$b;
    }
    return %DESCVLan_2;
}

#
#
# Funcion para generar hash Dobles
#
#
#
sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.]", $inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--) {
            pop(@C);
        }

        my $inst=join($Tipo,@C);

        $DESCVLan_2{$inst}=$Valor;
    }
}

```

```

    }

    return %DESCVlan_2;
}

#
#

sub Genera_Hash_HEX($$$$$) {

#
# Vectores de conversion a Decimal
#

for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Separador) = $_[2];
my ($Tipo) = $_[4];

my @Llaves = keys(%DESCVlan_2);
for my $re (@Llaves) {
    delete $DESCVlan_2{$re};
}

my @APORTStatus=split("\n", $DDesc);

```

```

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    if ($b =~ "OCTET STRING-") {
        my $inst_raw=shift(@B);

        my @C=split("[\\.]",$inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        $inst1=join($Separador,@C);
        $DESCVlan_2{$inst1}="";
        next;
    }
    else {
        $B[1] = $HEX{$B[1]} if ($Tipo);
        $Valor=$B[1] . " ";
        for ($i=2; $i<17; $i++) {
            $B[$i] = $HEX{$B[$i]} if ($Tipo);
            $Valor=$Valor . $B[$i] . " ";
        }
        $Valor =~ s/ +//g;
        $Valor =~ s/ -- //g;
        $Valor =~ s/--//g;
        $Valor =~ s/ $//g;
        $Valor =~ s/ /$_[3]/g;
    }
    $DESCVlan_2{$inst1}=$DESCVlan_2{$inst1} . $Valor;
}

delete $DESCVlan_2{"");

return %DESCVlan_2;
}

# Aqui terminan las subrutinas

#
#
#
# Autenticacion
#
#

#
# Lectura de la IP que solicita
#
$IPs = $ENV{'REMOTE_ADDR'};

```

```

#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/rip.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

@IP_Instancias = split(" ", $query->param('Lista_Instancias'));

$IP_switch = $query->param('IP_switch');
$Lugar = $query->param('sysName');
$Name = $query->param('Name');

foreach $b (@IP_Instancias) {

    $Disable{$b} = $query->param("Disable_{$b}");
}
}

```



```

$Disable_org{$b} = $query->param("_Disable_$b");
$Instanc{$b} = $query->param("Instanc_$b");
$Supply{$b} = $query->param("Supply_$b");
$Supply_org{$b} = $query->param("_Supply_$b");
$Listen{$b} = $query->param("Listen_$b");
$Listen_org{$b} = $query->param("_Listen_$b");
$DSupply{$b} = $query->param("DSupply_$b");
$DSupply_org{$b} = $query->param("_DSupply_$b");
$DListen{$b} = $query->param("DListen_$b");
$DListen_org{$b} = $query->param("_DListen_$b");
}

$checked{"enabled"} = "";
$checked{"disabled"} = "checked";

#
#
# Archivo de control de cambios
#
#

open(LOG_CAMBIOS, '>>/tmp/Control_R.txt');

$FECHA=`date`;
chop($FECHA);

#
#
# Logica de cambios en la configuracion
#
#
foreach $b (@IP_Instancias) {
    if ($Disable_org{$b} eq "enabled" && $Disable{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch wfRipIntfEnable.$b
integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name deshabilito
la interface RIP Instanc{$b} en el enrutador $IP_switch ($Lugar)\n\n";
    }
    if ($Disable_org{$b} eq "disabled" && !($Disable{$b})) {
        system("/opt/OV/bin/snmpset $IP_switch wfRipIntfEnable.$b
integer 1");
    }
}

```

```

        print LOG_CAMBIOS "$FECHA : El Usuario $Name habilito la
interface RIP $Instanc{$b} en el enrutador $IP_switch ($Lugar)\n\n";
    }

    if ($Supply_org{$b} eq "enabled" && $Supply{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch wfRipIntfSupply.$b
integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name desahabilito
el RIP Supply de la Instancia $Instanc{$b} del enrutador $IP_switch
($Lugar)\n\n";
    }
    if ($Supply_org{$b} eq "disabled" && !($Supply{$b})) {
        system("/opt/OV/bin/snmpset $IP_switch wfRipIntfSupply.$b
integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name habilito el
RIP Supply de la Instancia $Instanc{$b} del enrutador $IP_switch
($Lugar)\n\n";
    }
    if ($Listen_org{$b} eq "enabled" && $Listen{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch wfRipIntfListen.$b
integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name desahabilito
el RIP Listen de la Instancia $Instanc{$b} del enrutador $IP_switch
($Lugar)\n\n";
    }
    if ($Listen_org{$b} eq "disabled" && !($Listen{$b})) {
        system("/opt/OV/bin/snmpset $IP_switch wfRipIntfListen.$b
integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name habilito el
RIP Listen de la Instancia $Instanc{$b} del enrutador $IP_switch
($Lugar)\n\n";
    }
    if ($DListen_org{$b} eq "enabled" && $DListen{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch
wfRipIntfDefaultRouteListen.$b integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name desahabilito
el Default Route Listen de la Instancia $Instanc{$b} del enrutador
$IP_switch ($Lugar)\n\n";
    }
    if ($DListen_org{$b} eq "disabled" && !($DListen{$b})) {
        system("/opt/OV/bin/snmpset $IP_switch
wfRipIntfDefaultRouteListen.$b integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name habilito el
Default Route Listen de la Instancia $Instanc{$b} del enrutador
$IP_switch ($Lugar)\n\n";
    }
    if ($DSupply_org{$b} eq "enabled" && $DSupply{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch
wfRipIntfDefaultRouteSupply.$b integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name desahabilito
el Default Route Supply de la Instancia $Instanc{$b} del enrutador
$IP_switch ($Lugar)\n\n";
    }

```

```

    }
    if ($DSupply_org{$b} eq "disabled" && !($DSupply{$b})) {
        system("/opt/OV/bin/snmpset $IP_switch
wfRipIntfDefaultRouteSupply.$b integer 1");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name desahabilito
el Default Route Supply de la Instancia $Instanc{$b} del enrutador
$IP_switch ($Lugar)\n\n";
    }
}

close(LOG_CAMBIOS);

#
#
# Carga de la informacion del equipo
#
#

@uma_guma=("wfRipIntfEnable", "wfRipIntfState", "wfRipIntfSupply",
"wfRipIntfListen", "wfRipIntfDefaultRouteSupply",
"wfCircuitName", "wfRipIntfDefaultRouteListen",);

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);
foreach $b (@APM) {
    if ($b =~ "wfRipIntfEnable" && $b =~ "INTEGER") {
        $wfRipIntfEnable="$wfRipIntfEnable" . "\n" . "$b";
    }
    if ($b =~ "wfRipIntfState" && $b =~ "INTEGER") {
        $wfRipIntfState="$wfRipIntfState" . "\n" . "$b";
    }
    if ($b =~ "wfRipIntfSupply" && $b =~ "INTEGER") {
        $wfRipIntfSupply="$wfRipIntfSupply" . "\n" . "$b";
    }
    if ($b =~ "wfRipIntfListen" && $b =~ "INTEGER") {
        $wfRipIntfListen="$wfRipIntfListen" . "\n" . "$b";
    }
    if ($b =~ "wfRipIntfDefaultRouteSupply" && $b =~ "INTEGER") {
$wfRipIntfDefaultRouteSupply="$wfRipIntfDefaultRouteSupply" . "\n" .
"$b";
    }
    if ($b =~ "wfCircuitName" && $b =~ "STRING") {
        $wfCircuitName="$wfCircuitName" . "\n" . "$b";
    }
    if ($b =~ "wfRipIntfDefaultRouteListen" && $b =~ "INTEGER") {

```

```

        $wfRipIntfDefaultRouteListen="$wfRipIntfDefaultRouteListen" .
"\n" . "$b";
    }
}

```

```

%Enabl = Genera_Hash_Doble($wfRipIntfEnable,10,14,".");
%State = Genera_Hash_Doble($wfRipIntfState,10,14,".");
%Liste = Genera_Hash_Doble($wfRipIntfListen,10,14,".");
%Suppl = Genera_Hash_Doble($wfRipIntfSupply,10,14,".");
%DSupp = Genera_Hash_Doble($wfRipIntfDefaultRouteSupply,10,14,".");
%DList = Genera_Hash_Doble($wfRipIntfDefaultRouteListen,10,14,".");
%CName = Genera_Hash_Simple($wfCircuitName,5);

```

```
delete $Enabl{""};
```

```
@List_Inst = keys(%Enabl);
```

```
foreach $b (@List_Inst) {
    @IP_Circ=split("[\.]", $b);
    $Num_Circ{$b}=pop(IP_Circ);
    $IP_Circ{$b}=join(".", @IP_Circ);
}

```

```
print $query->header;
```

```
print<<"_HTML_";
```

```

<HTML>
<HEAD>
</HEAD>
<body bgcolor=#C0C0C0>
<center>
    <h3>Configuraci&oacute;n del RIP</h3>
</center>
<center>
<table border=3>
    <tr>
        <th width="208" align=left> <div align="center">Nombre</div></th>
        <th width="112" align=left><div align="center">IP</div></th>
        <form name="Formul_RIP" method="post" action="rip.pl">

```

```



```

```
_HTML_
```

```
foreach $b (@List_Inst) {
```

```
print<<"_HTML_";
```

```

<tr>
<th align=left>$CName{$Num_Circ{$b}}</th>
<th align=left>$IP_Circ{$b}</th>
<th align=left><table width="101%" border="0">
<tr>
<td width="47%"><strong>$State{$b}</strong></td>
<td width="53%"><p>


```

```

                <input name="Listen_{$b}" type="checkbox" value="2"
$checked{$Liste{$b}}>
                <input name="_Listen_{$b}" type="hidden" value="$Liste{$b}">
                &nbsp;&nbsp;&nbsp;Dis</p></td>
            </tr>
        </table></th>
        <th align=left><table width="101%" border="0">
            <tr>
                <td width="47%"><strong>${DSuppx{$b}}</strong></td>
                <td width="53%"><p>
                    <input name="DSupply_{$b}" type="checkbox" value="2"
$checked{$DSupp{$b}}>
                    <input name="_DSupply_{$b}" type="hidden" value="$DSupp{$b}">
                    &nbsp;&nbsp;&nbsp;Dis</p></td>
            </tr>
        </table></th>
        <th align=left><table width="101%" border="0">
            <tr>
                <td width="47%"><strong>${DListx{$b}}</strong></td>
                <td width="53%"><p>
                    <input name="DListen_{$b}" type="checkbox" value="2"
$checked{$DList{$b}}>
                    <input name="_DListen_{$b}" type="hidden" value="$DList{$b}">
                    &nbsp;&nbsp;&nbsp;Dis</p></td>
            </tr>
        </table></th>

    _HTML_
}

print<<"_HTML_";

</table>
<p align="center">&nbsp;&nbsp;&nbsp;
    <input type="submit" value="Actualizar"></form>&nbsp;&nbsp;&nbsp;
    <input type="reset" value="Deshacer">
</BODY>
</HTML>

    _HTML_
}

else

    {
    print $query->header;
    print $query->start_html(-title=>'No Autorizado');

```

```
print $query->h1('No esta autorizado para ver esta pagina');  
print $query->end_html;  
}
```

B.12 Tabla ARP

```

#!/usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLAndesc=split("\n",$Desc);
    foreach $b (@AVLAndesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]",$inst_raw);
        $DESCVlan_2{$C[7]}=join(" ",@B);
    }

    delete $DESCVlan_2("");
    return %DESCVlan_2;
}

#
#
#

#
#
#

sub Invertir {

```



```

my @Llaves = keys(%DESCVlan_2);
for my $re (@Llaves) {
    delete $DESCVlan_2{$re};
}

my %Lucas = @_;
my @Llav = keys(%Lucas);
foreach my $b (@Llav) {
    $DESCVlan_2{$Lucas{$b}}=$b;
}
return %DESCVlan_2;
}

#
#
# Funcion para generar hash Dobles
#
#
sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.]", $inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--
    ) {
        pop(@C);
    }
}

```

```

        my $inst=join($Tipo,@C);

        $DESCVLan_2{$inst}=$Valor;
    }

    return %DESCVLan_2;
}

#
#

sub Genera_Hash_HEX($$$$$) {

#
# Vectores de conversion a Decimal
#

    for ($m=0; $m<16; $m++) {

        $HEXAD=sprintf("0%x", $m);
        $ASCCI=sprintf("%c", $m);

        $HEX{$HEXAD} = $m;
        $ASC{$ASCCI} = $m;

    }

    for ($m=16; $m<256; $m++) {

        $HEXAD=sprintf("%x", $m);
        $ASCCI=sprintf("%c", $m);

        $HEX{$HEXAD} = $m;
        $ASC{$ASCCI} = $m;

    }

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Separador) = $_[2];
    my ($Tipo) = $_[4];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }
}

```

```

    }

my @APORTStatus=split("\n",$DDesc);

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    if ($b =~ "OCTET STRING-") {
        my $inst_raw=shift(@B);

        my @C=split("[\.]",$inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        $inst1=join($Separador,@C);
        $DESCVLan_2{$inst1}="";
        next;
    }
    else {
        $B[1] = $HEX{$B[1]} if ($Tipo);
        $Valor=$B[1] . " ";
        for ($i=2; $i<17; $i++) {
            $B[$i] = $HEX{$B[$i]} if ($Tipo);
            $Valor=$Valor . $B[$i] . " ";
        }
        $Valor =~ s/ +//g;
        $Valor =~ s/ -- //g;
        $Valor =~ s/--//g;
        $Valor =~ s/ $//g;
        $Valor =~ s/ /$_{3}/g;
    }
    $DESCVLan_2{$inst1}=$DESCVLan_2{$inst1} . $Valor;
}

delete $DESCVLan_2{" "};

return %DESCVLan_2;
}

```

```
# Aqui terminan las subrutinas
```

```

#
#
#
# Autenticacion
#

```

```

#

#
# Lectura de la IP que solicita
#
$IPs = $ENV{'REMOTE_ADDR'};
#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
#
    print "$b\n";
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/arp.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
#$autentUS = $autentIP =1;
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

```

```

@IP_Instanceias = split(" ",$query->param('Lista_Instanceias'));

$IP_switch = $query->param('IP_switch');
$Lugar = $query->param('sysName');
$Name = $query->param('Name');

foreach $b (@IP_Instanceias) {

    $Borrar{$b} = $query->param("Borrar_{$b}");
    $Borrar_org{$b} = $query->param("_Borrar_{$b}");
    $Instanc{$b} = $query->param("Instanc_{$b}");

}

#
#
# Archivo de control de cambios
#
#

open(LOG_CAMBIOS, '>>/tmp/Control_R.txt');

$FECHA=`date`;
chop($FECHA);

#
#
# Logica de cambios en la configuracion
#
#
foreach $b (@IP_Instanceias) {

    if ($Borrar{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch
wfIpNetToMediaType.{$b} integer 2");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name borro la IP
$Instanc{$b} en el enrutador $IP_switch ($Lugar)\n\n";
    }

}

close(LOG_CAMBIOS);

```

```

#
#
# Carga de la informacion del equipo
#
#

@uma_guma=("wfIpNetToMediaPhysAddress",
           "wfIpNetToMediaType");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);

$encontrado0=0;

foreach $b (@APM) {
    if ($b =~ "wfIpNetToMediaType" && $b =~ "INTEGER") {
        $wfIpNetToMediaType="$wfIpNetToMediaType" . "\n" . "$b";
    }
    if ($b =~ "wfIpNetToMediaPhysAddress" && $b =~ "OCTET STRING" ||
$encontrado0) {
        $wfIpNetToMediaPhysAddress=$wfIpNetToMediaPhysAddress.
"\n" . "$b";
        $encontrado0=1;
        $encontrado0=0 if ($b =~ "-- -- --");
    }
}

%Med_T = Genera_Hash_Doble($wfIpNetToMediaType,10,14,".");
%MAC_A = Genera_Hash_HEX($wfIpNetToMediaPhysAddress,10,".", "-", "");

delete $Med_T{"");

@List_Inst = keys(%Med_T);

foreach $b (@List_Inst) {
    @Slot_Conn=split("[\.]", $b);
    $IPs{$b}="$Slot_Conn[1]";
    for ($cont=2; $cont <5; $cont++) {
        $IPs{$b}=$IPs{$b} . "." . $Slot_Conn[$cont];
    }
}

```

```

    }
    $MAC_A{$b} =~ tr/[a-z]/[A-Z]/;
}

print $query->header;

print<<"_HTML_";

<HTML>
<HEAD>
</HEAD>

<body bgcolor=#C0C0C0>
<center>
  <h3>Tabla ARP</h3>
</center>
<center>
<table border=3 align="center">
  <tr>
    <th width="208" align=left> <div align="center">Direcci&oacute;n
IP</div></th>
    <th width="128" align=left><div align="center">Tipo</div></th>
    <form name="Formul_ARP" method="post" action="arp.pl">
    <input type="hidden" name="IP_switch" id="IP_Switch"
value="10.10.16.1">
    <input type="hidden" name="Lista_Instancias" id="Lista_Instancias"
value="@List_Inst">
    <input type="hidden" name="sysName" id="sysName" value="$Lugar">
    <input type="hidden" name="Name" id="Name" value="$Name">
    <th width="147" align=left><div align="center">MAC</div></th>
    <th width="128" align=left><div align="center">Borrar</div></th>

_HTML_

foreach $b (@List_Inst) {

print<<"_HTML_";

  <tr>
    <th align=left>$IPs{$b}</th>
    <th align=left>$Med_T{$b}</th>
    <th align=left>$MAC_A{$b}</th>
    <th align=left><div align="center">
      <input type="checkbox" name="Borrar_$b" value="2">
      <input name="Instanc_$b" type="hidden" id="Instanc_$b"
value="$IPs{$b}">
    </div></th>

_HTML_

```


B.13 Líneas de Frame Relay

```

#!/usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#
#

sub Genera_Hash_Simple($$) {
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLAndesc=split("\n",$Desc);
    foreach $b (@AVLAndesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.]",$inst_raw);
        $DESCVlan_2{$C[7]}=join(" ",@B);
    }

    delete $DESCVlan_2{""};
    return %DESCVlan_2;

}

#
#
#

sub Invertir {

```

```

my @Llaves = keys(%DESCVlan_2);
for my $re (@Llaves) {
    delete $DESCVlan_2{$re};
}

my %Lucas = @_;
my @Llav = keys(%Lucas);
foreach my $b (@Llav) {
    $DESCVlan_2{$Lucas{$b}}=$b;
}
return %DESCVlan_2;
}

#
#
# Funcion para generar hash Dobles
#
#
#
sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\\.]",$inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--)
    ) {
        pop(@C);
    }
}

```

```

        my $inst=join($Tipo,@C);

        $DESCVLan_2{$inst}=$Valor;
    }

    return %DESCVLan_2;
}

#
#

sub Genera_Hash_HEX($$$$$) {

#
# Vectores de conversion a Decimal
#

    for ($m=0; $m<16; $m++) {

        $HEXAD=sprintf("0%x",$m);
        $ASCCI=sprintf("%c",$m);

        $HEX{$HEXAD} = $m;
        $ASC{$ASCCI} = $m;

    }

    for ($m=16; $m<256; $m++) {

        $HEXAD=sprintf("%x",$m);
        $ASCCI=sprintf("%c",$m);

        $HEX{$HEXAD} = $m;
        $ASC{$ASCCI} = $m;

    }

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Separador) = $_[2];
    my ($Tipo) = $_[4];

    my @Llaves = keys(%DESCVLan_2);
    for my $re (@Llaves) {
        delete $DESCVLan_2{$re};
    }
}

```

```

    }

my @APORTStatus=split("\n",$DDesc);

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    if ($b =~ "OCTET STRING-") {
        my $inst_raw=shift(@B);

        my @C=split("[\.]",$inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        $inst1=join($Separador,@C);
        $DESCVLan_2{$inst1}="";
        next;
    }
    else {
        $B[1] = $HEX{$B[1]} if ($Tipo);
        $Valor=$B[1] . " ";
        for ($i=2; $i<17; $i++) {
            $B[$i] = $HEX{$B[$i]} if ($Tipo);
            $Valor=$Valor . $B[$i] . " ";
        }
        $Valor =~ s/ +//g;
        $Valor =~ s/ -- //g;
        $Valor =~ s/--//g;
        $Valor =~ s/ $//g;
        $Valor =~ s/ /$_[3]/g;
    }
    $DESCVLan_2{$inst1}=$DESCVLan_2{$inst1} . $Valor;
}

delete $DESCVLan_2{""};

return %DESCVLan_2;
}

```

```
# Aqui terminan las subrutinas
```

```
#
```

```

#
#
# Autenticacion
#
#

#
# Lectura de la IP que solicita
#
$IPs = $ENV{'REMOTE_ADDR'};
#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/sfl.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
if ($autentUS && $autentIP) {

#
#

```

```

# Lectura de las variables de estado
#
#

$IP_switch = $query->param('IP_switch');
$Lugar = $query->param('sysName');
$Name = $query->param('Name');

@IP_Instancias = split(" ", $query->param('Lista_Instancias'));

foreach $b (@IP_Instancias) {

    $TxOct_{$b} = $query->param("TxOct_{$b}");
    $TxFra_{$b} = $query->param("TxFra_{$b}");
    $RxOct_{$b} = $query->param("RxOct_{$b}");
    $RxFra_{$b} = $query->param("RxFra_{$b}");
    $BECNs_{$b} = $query->param("BECNs_{$b}");
    $FECNs_{$b} = $query->param("FECNs_{$b}");
}

#
#
# Carga de la informacion del equipo
#
#

@uma_guma=("wfFrDlcmiInterfaceType",
           "wfFrDlcmiManagementType", "wfFrDlcmiStatus",
           "wfCircuitName", "wfFrErrFaults", "wfFrDlcmiCircuit",
           "wfFrErrDiscards", "wfFrErrDrops");

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);

$encontrado0=0;

foreach $b (@APM) {
    if ($b =~ "wfFrDlcmiManagementType" && $b =~ "INTEGER") {

```

```

        $wfFrDlcmiManagementType="$wfFrDlcmiManagementType" .
"\n" . "$b";
    }
    if ($b =~ "wfFrDlcmiInterfaceType" && $b =~ "INTEGER") {
        $wfFrDlcmiInterfaceType="$wfFrDlcmiInterfaceType" . "\n" .
"$b";
    }
    if ($b =~ "wfFrDlcmiStatus" && $b =~ "INTEGER") {
        $wfFrDlcmiStatus="$wfFrDlcmiStatus" . "\n" . "$b";
    }
    if ($b =~ "wfFrDlcmiCircuit" && $b =~ "INTEGER") {
        $wfFrDlcmiCircuit="$wfFrDlcmiCircuit" . "\n" . "$b";
    }
    if ($b =~ "wfFrErrFaults" && $b =~ "Counter") {
        $wfFrErrFaults="$wfFrErrFaults" . "\n" . "$b";
    }
    if ($b =~ "wfFrErrDiscards" && $b =~ "Counter") {
        $wfFrErrDiscards="$wfFrErrDiscards" . "\n" . "$b";
    }
    if ($b =~ "wfFrErrDrops" && $b =~ "Counter") {
        $wfFrErrDrops="$wfFrErrDrops" . "\n" . "$b";
    }
    if ($b =~ "wfCircuitName" && $b =~ "STRING") {
        $wfCircuitName="$wfCircuitName" . "\n" . "$b";
    }
}

```

```

%RxOct = Genera_Hash_Doble($wfFrDlcmiManagementType,9,10,".");
%RxFra = Genera_Hash_Doble($wfFrDlcmiInterfaceType,9,10,".");
%TxOct = Genera_Hash_Doble($wfFrDlcmiStatus,9,10,".");
%TxFra = Genera_Hash_Doble($wfFrErrFaults,9,10,".");
%FECNs = Genera_Hash_Doble($wfFrErrDiscards,9,10,".");
%BECNs = Genera_Hash_Doble($wfFrErrDrops,9,10,".");
%Circu = Genera_Hash_Doble($wfFrDlcmiCircuit,9,10,".");
%CName = Genera_Hash_Simple($wfCircuitName,5);

```

```
delete $RxOct{""};
```

```
@List_Inst = keys(%RxOct);
```

```
foreach $b (@List_Inst) {
```

```

    $dRxFra{$b} = $RxFra{$b} - $RxFra_{$b} if ($RxFra_{$b});
    $dFECNs{$b} = $FECNs{$b} - $FECNs_{$b} if ($FECNs_{$b});

```

```

        $dBECNs{$b} = $BECNs{$b} - $BECNs_{$b} if ($BECNs_{$b});
    }

print $query->header;

print<<"_HTML_";

<HTML>
<HEAD>
</HEAD>

<body bgcolor=#C0C0C0>
<center>
    <h3>Frame Relay</h3>
</center>
<center>
<table border=3>
    <tr>
        <th align=left> <div align="center">Nombre</div></th>
        <th width="128" align=left><div
align="center">Line.LLindex</div></th>
        <form name="Formul_SFL" method="post" action="sfl.pl">
        <input type="hidden" name="IP_switch" id="IP_Switch"
value="$IP_switch">
        <input type="hidden" name="Lista_Instancias" id="Lista_Instancias"
value="@List_Inst">
        <input type="hidden" name="sysName" id="sysName" value="$Lugar">
        <input type="hidden" name="Name" id="Name" value="$Name">
        <th width="128" align=left><div align="center">Mgmt Type</div></th>
        <th width="128" align=left><div align="center">Intf Type</div></th>
        <th width="128" align=left><div align="center">Status</div></th>
        <th width="128" align=left><div align="center">Faults</div></th>
        <th width="128" align=left><div align="center">Discards</div></th>
        <th width="128" align=left><div align="center">Drops</div></th>

    _HTML_

foreach $b (@List_Inst) {

print<<"_HTML_";

    <tr>
        <th align=left>$CName{$Circu{$b}}</th>
        <th align=left>$b</th>
        <th align=left>$RxOct{$b}</th>
        <th align=left>$RxFra{$b}</th>
        <th align=left>$TxOct{$b}</th>
        <th align=left>$TxFra{$b}<br>
        Dif=$dTxFra{$b}</th>
        <th align=left>$FECNs{$b}<br>

```


B.14 Aplicación para Manejo de Archivos en los Enrutadores

```

#! /usr/perl5/5.00503/bin/perl

use CGI;
$query = new CGI;

#
#
# Funcion de creacion de hash
#
#
#

sub Genera_Hash_Simple($$) (
    my ($Desc) = $_[0];
    my ($Repeticiones) = $_[1];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @AVLANDesc=split("\n",$Desc);
    foreach $b (@AVLANDesc) {
        $b=~ s/\s/ /g;
        my @B=split(" ", $b);
        my $inst_raw=shift(@B);
        for (my $i=1; $i<$Repeticiones; $i++) {
            shift(@B);
        }
        my @C=split("[\.", $inst_raw);
        $DESCVlan_2{$C[7]}=join(" ", @B);
    }

    delete $DESCVlan_2{""};
    return %DESCVlan_2;

}

#
#
#
#
#
#
#

```

```

sub Invertir {

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my %Lucas = @_;
    my @Llav = keys(%Lucas);
    foreach my $b (@Llav) {
        $DESCVlan_2{$Lucas{$b}}=$b;
    }
    return %DESCVlan_2;
}

#
#
#  Funcion para generar hash Dobles
#
#
#

sub Genera_Hash_Doble($$$$) {

    my ($DDesc) = $_[0];
    my ($Repeticiones1) = $_[1];
    my ($Repeticiones2) = $_[2];
    my ($Tipo) = $_[3];

    my @Llaves = keys(%DESCVlan_2);
    for my $re (@Llaves) {
        delete $DESCVlan_2{$re};
    }

    my @APORTStatus=split("\n",$DDesc);

    foreach $b (@APORTStatus) {
        my @B=split(" ",$b);
        my $inst_raw=shift(@B);

        my $Valor=pop(@B);

        my @C=split("[\.]",$inst_raw);

        for (my $i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }
    }
}

```

```

    for (my $i=$#C+1; $i>$Repeticiones2-$Repeticiones1+1; $i--
) {
        pop(@C);
    }

    my $inst=join($Tipo,@C);

    $DESCVLan_2{$inst}=$Valor;
}

return %DESCVLan_2;
}

sub Genera_Hash_HEX($$$$$) {
#
# Vectores de conversion a Decimal
#
for ($m=0; $m<16; $m++) {

    $HEXAD=sprintf("0%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

for ($m=16; $m<256; $m++) {

    $HEXAD=sprintf("%x", $m);
    $ASCCI=sprintf("%c", $m);

    $HEX{$HEXAD} = $m;
    $ASC{$ASCCI} = $m;

}

my ($DDesc) = $_[0];
my ($Repeticiones1) = $_[1];
my ($Separador) = $_[2];
my ($Tipo) = $_[4];

my @Llaves = keys(%DESCVLan_2);
for my $re (@Llaves) {

```

```

        delete $DESCVlan_2{$re};
    }

my @APORTStatus=split("\n",$DDesc);

foreach $b (@APORTStatus) {
    my @B=split(" ",$b);
    if ($b =~ "OCTET STRING-") {
        my $inst_raw=shift(@B);

        my @C=split("[\.]",$inst_raw);

        for ($i=1; $i<$Repeticiones1; $i++) {
            shift(@C);
        }

        $inst1=join($Separador,@C);
        $DESCVlan_2{$inst1}="";
        next;
    }
    else {
        $B[1] = $HEX{$B[1]} if ($Tipo);
        $Valor=$B[1] . " ";
        for ($i=2; $i<17; $i++) {
            $B[$i] = $HEX{$B[$i]} if ($Tipo);
            $Valor=$Valor . $B[$i] . " ";
        }
        $Valor =~ s/ +//g;
        $Valor =~ s/ -- //g;
        $Valor =~ s/--//g;
        $Valor =~ s/ $//g;
        $Valor =~ s/ /$_[3]/g;
    }
    $DESCVlan_2{$inst1}=$DESCVlan_2{$inst1} . $Valor;
}

delete $DESCVlan_2{""};

return %DESCVlan_2;
}

```

```
# Aqui terminan las subrutinas
```

```

#
#
#
# Autenticacion
#

```

```

#

# Lectura de la IP que solicita
#
$IPs = $ENV{'REMOTE_ADDR'};
#
#
# Lectura del archivo de IP para ver si estan las adecuadas
#
open (IP, '/etc/IP.wv') || die();
$i = 1;
$text=<IP>;
while ($text) {
    chop($text);
    $IPa[$i++] = $text;
    $text=<IP>;
}
close(IP);
#
$i = 1;
$autentIP = 0;

#
# Revision que la IP este en la lista de acceso
#
foreach $b (@IPa) {
    if (! $b) {next;}
    if (! $IPs) {last;}
#
    print "$b\n";
    if ($b eq $IPs) {
        $autentIP = 1;
        last;}
}
if ($ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/Accesos_Dir.pl"
|| $ENV{'HTTP_REFERER'} =~ "http://10.1.4.151:8880/OvCgi/arc.pl") {
    $autentUS=1;
}
if ($IPs eq "10.1.4.151") {$autentUS = $autentIP =1;}
#$autentUS = $autentIP =1;
if ($autentUS && $autentIP) {

#
#
# Lectura de las variables de estado
#
#

$Copia_completa = 0;

@IP_Instancias = split(" ", $query->param('Lista_Instancias'));

```

```

$IP_switch = $query->param('IP_switch');
$Lugar = $query->param('sysName');
$Name = $query->param('Name');
$Borrar = $query->param('Borrar');
$Copiar = $query->param('Copiar');
$Salvar_Conf = $query->param('Salvar_Conf');
$Arc_Dest_Salvar = $query->param('Arc_Dest_Salvar');
$Compactar = $query->param('Compactar');

foreach $b (@IP_Instancias) {

    $Borrar_arc{$b} = $query->param("Nombre_{$b}");
    $Disable_org{$b} = $query->param("_Disable_{$b}");
    $IP_Costo_org{$b} = $query->param("_IP_Costo_{$b}");
    $IP_Costo{$b} = $query->param("IP_Costo_{$b}");
    $Instanc{$b} = $query->param("Instanc_{$b}");
    $Arc_Dest{$b} = $query->param("Arc_Dest_{$b}");
}

#
#
# Archivo de control de cambios
#
#

open(LOG_CAMBIOS, '>>/tmp/Control_R.txt');

$FECHA=`date`;
chop($FECHA);

$Copia_completa0 = 0;

    if ($Salvar_Conf && $Arc_Dest_Salvar) {
        system("/opt/OV/bin/snmpset $IP_switch wfTiRui.2.0
octetstring \"save config 1:$Arc_Dest_Salvar\");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name salvo la
configuracion del enrutador con el nombre $Arc_Dest_Salvar en $IP_switch
($Lugar)\n\n";
    }

    if ($Compactar) {
        system("/opt/OV/bin/snmpset $IP_switch wfTiRui.2.0
octetstring \"compact 1:\");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name compacto la
flash en el enrutador $IP_switch ($Lugar)\n\n";
    }

```

```

#
#
# Logica de cambios en la configuracion
#
#
foreach $b (@IP_Instancias) {

    if ($Borrar_arc{$b} && $Borrar) {
        system("/opt/OV/bin/snmpset $IP_switch wftiRui.2.0
octetstring \"delete 1:$Borrar_arc{$b}\");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name borro el
archivo $Borrar_arc{$b} del enrutador $IP_switch ($Lugar)\n\n";
    }
    if ($Copiar && $Arc_Dest{$b}) {
        system("/opt/OV/bin/snmpset $IP_switch wftiRui.2.0
octetstring \"copy 1:$b 1:$Arc_Dest{$b}\");
        print LOG_CAMBIOS "$FECHA : El Usuario $Name copio el
archivo $b a $Arc_Dest{$b} en el enrutador $IP_switch ($Lugar)\n\n";
        $Copia_completa0=1;
    }

}

$Copia_completa = 1 if ($Copia_completa0);

close(LOG_CAMBIOS);

if ($Compactar) {

    print $query->header;
    print $query->start_html(-bgcolor=>'#COCOCO'),"\n",
        $query->start_form(-method=>'post', -action=>'Mai.pl'),
        "Se va a compactar la flash card por lo que el FS no va a
estar disponible por cierto tiempo",
        $query->p,"\n",
        $query->hidden(-name=>'IP_switch',-
value=>"$IP_switch"),"\n",
        $query->hidden(-name=>'Name',-value=>"$Name"),"\n",
        $query->hidden(-name=>'sysName',-value=>"$Lugar"),"\n",
        $query->submit(-name=>'Compactar',-value=>'Aceptar'),"\n",
        $query->end_form;
    print $query->end_html;
    exit;

}

```



```

if ($Salvar_Conf && !($Arc_Dest_Salvar)) {

    print $query->header;
    print $query->start_html(-bgcolor=>'#C0C0C0'),"\n",
        $query->start_form(-method=>'post', -action=>'arc.pl'),
        "Nombre del archivo a salvar: ", $query-
>textfield('Arc_Dest_Salvar'),
        $query->p,"\n",
        $query->hidden(-name=>'IP_switch',-
value=>"$IP_switch"),"\n",
        $query->hidden(-name=>'Name',-value=>"$Name"),"\n",
        $query->hidden(-name=>'sysName',-value=>"$Lugar"),"\n",
        $query->submit(-name=>'Salvar_Conf',-
value=>'Aceptar'),"\n",
        $query->end_form;

    print $query->end_html;
    exit;

}

#
#
# Carga de la informacion del equipo
#
#

system("/opt/OV/bin/snmpset $IP_switch wfFsAction.100 integer 1");

@uma_guma=("wfFsSize", "wfFsFreeSize", "wfFsContigFree",
          "wfFsDirFileName", "wfFsDirCreated", "wfFsDirFileSize");

$encontrado0=0;

$MISC=`/opt/OV/bin/snmpwalk $IP_switch @uma_guma`;
@APM=split("\n", $MISC);
foreach $b (@APM) {

    if ($b =~ "wfFsSize" && $b =~ "Gauge32") {
        $wfFsSize="$wfFsSize" . "\n" ."$b";
    }
    if ($b =~ "wfFsFreeSize" && $b =~ "Gauge32") {
        $wfFsFreeSize="$wfFsFreeSize" . "\n" . "$b";
    }
    if ($b =~ "wfFsContigFree" && $b =~ "Gauge32") {

```

```

        $wfFsContigFree="$wfFsContigFree" . "\n" . "$b";
    }
    if ($b =~ "wfFsDirFileName" && $b =~ "STRING") {
        $wfFsDirFileName="$wfFsDirFileName" . "\n" . "$b";
    }
    if ($b =~ "wfFsDirCreated" && $b =~ "STRING" || $encontrado0) {
        $wfFsDirCreated="$wfFsDirCreated" . "\n" . "$b";
        $encontrado0=1;
        $encontrado0=0 if ($b =~ "-- -- --");
    }
    if ($b =~ "wfFsDirFileSize" && $b =~ "Gauge32") {
        $wfFsDirFileSize="$wfFsDirFileSize" . "\n" . "$b";
    }
}

```

```

%FsSize = Genera_Hash_Doble($wfFsSize,9,9,".");
%FsFSiz = Genera_Hash_Doble($wfFsFreeSize,9,9,".");
%FsCFre = Genera_Hash_Doble($wfFsContigFree,9,9,".");
%DFName = Genera_Hash_Doble($wfFsDirFileName,9,23,".");
%DCreat = Genera_Hash_HEX($wfFsDirCreated,9,".",",","");
%FiSize = Genera_Hash_Doble($wfFsDirFileSize,9,23,".");

```

```
delete $DFName{""};
```

```
@List_Inst = sort(keys(%DFName));
```

```
foreach $b (@List_Inst) {
```

```

    @FECHA_AUX = split(" ",$DCreat{$b});
    $Year = $HEX{$FECHA_AUX[0]}*256+$HEX{$FECHA_AUX[1]};
    $Mes = $HEX{$FECHA_AUX[2]};
    $Dia = $HEX{$FECHA_AUX[3]};
    $Hora = $HEX{$FECHA_AUX[4]};
    $Min = ($HEX{$FECHA_AUX[5]}<10) ? "0" . $HEX{$FECHA_AUX[5]} :
$HEX{$FECHA_AUX[5]};
    $Seg = ($HEX{$FECHA_AUX[6]}<10) ? "0" . $HEX{$FECHA_AUX[6]} :
$HEX{$FECHA_AUX[6]};

```

```

    $Fecha{$b} = $Dia . "/" . $Mes . "/" . $Year;
    $Tiempo{$b} = $Hora . ":" . $Min . ":" . $Seg;

```

```

    @ORDEN_AUX = split("[\\.]", $b);
    $Valor{$b} = pop(@ORDEN_AUX);
    $Max_Indice = ($Valor{$b} > $Max_Indice) ? $Valor{$b} :
$Max_Indice;

```

```

}

#
# Ordenamiento de los archivos segun tiempo de creacion
#

%ORDEN=Invertir(%Valor);

for ($i=1; $i<=$Max_Indice; $i++) {

    push(@List_AUX,$ORDEN{$i});
}

@List_Inst=@List_AUX;

print $query->header;

@List_Noms=values(%DFName);

print<<"_HTML_";

<HTML>
<HEAD>
</HEAD>
<body bgcolor=#C0C0C0>
<center>
    <h3>Sistema de Archivos</h3>
</center>
<center>
<table width="740" border=3 align="center">
    <tr>
        <th width="267" align=left> <div align="center">Nombre del
Archivo</div></th>
        <th width="101" align=left><div
align="center">Tama&ntilde;o</div></th>
        <form name="Formul_ARC" method="post" action="arc.pl">
            <input type="hidden" name="IP_switch" id="IP_Switch"
value="$IP_switch">
            <input type="hidden" name="Lista_Instancias" id="Lista_Instancias"
value="@List_Noms">
            <input type="hidden" name="sysName" id="sysName" value="$Lugar">
            <input type="hidden" name="Name" id="Name" value="$Name">
            <th width="88" align=left><div align="center">Fecha</div></th>
            <th width="86" align=left><div align="center">Hora</div></th>

```

HTML

```

print "      <th width=\"142\" align=left><div align=\"center\">Archivo
destino</div></th>\n" if ($Copiar && !($Copia_completa));

foreach $b (@List_Inst) {

print<<"_HTML_";

    <tr>
        <th align=left> <div align="left">

_HTML_

print "          <input type=\"checkbox\" name=\"Nombre_$(DFName{$b})\"
value=\"$(DFName{$b})\"";
print " checked" if ($Borrar_arc$(DFName{$b}) && $Copiar
&& !($Copia_completa));

print<<"_HTML_";
    >
        $(DFName{$b})</div></th>
        <th align=left><div align="right">$(FiSize{$b})</div></th>
        <th align=left><div align="right">$(Fecha{$b})</div></th>
        <th align=left><div align="left">$(Tiempo{$b})</div></th>

_HTML_

if ($Copiar && !($Copia_completa)) {

print "      <th align=left> <div align=\"left\">\n";
print "          <input name=\"Arc_Dest_$(DFName{$b})\" type=\"text\">\n" if
$(Borrar_arc$(DFName{$b}));
print "      </div></th>\n";

}

}

print<<"_HTML_";

</table>
<table width="740" border="0" align="center">
    <tr>
        <td width="26%">Capacidad Total:</td>
        <td width="23%">$(FsSize{100}) bytes</td>
        <td width="1%">&nbsp;</td>
        <td width="47%">&nbsp;</td>
    </tr>
    <tr>
        <td>Espacio Libre:</td>

```

```

        <td>$FsFSiz{100} bytes</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>

_HTML_

print "      <input name=\"Compactar\" type=\"submit\"
value=\"Compactar\">&nbsp;<n\" if ($Copia_completa || !($Copiar));
print "      <input name=\"Borrar\" type=\"submit\" value=\"  Borrar
\">&nbsp;<n\" if ($Copia_completa || !($Copiar));
print "      <input name=\"Copiar\" type=\"submit\" value=\"  Copiar
\">&nbsp;<n\" if ($Copia_completa || !($Copiar));
print "      <input name=\"Salvar_Conf\" type=\"submit\" value=\"Salvar
Conf\">&nbsp;<n\" if ($Copia_completa || !($Copiar));
print "      <input name=\"Copiar\" type=\"submit\" value=\"      Aceptar
\">&nbsp;<n\" if (!(($Copia_completa) && $Copiar);

print<<"_HTML_";
    </td>
  </tr>
  <tr>
    <td>Espacio Libre Contiguo:</td>
    <td>$FsCFre{100} bytes</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
<p>&nbsp;</p>
</BODY>
</HTML>

_HTML_

}

else

{
print $query->header;
print $query->start_html(-title=>'No Autorizado');
print $query->hl('No esta autorizado para ver esta pagina');
print $query->end_html;
}

```