UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Herramientas web para el monitoreo de servicios en Centros de Operación de Redes

Trabajo de graduación en modalidad de Tesis presentado por
Donald Antonio Velásquez Aguilar
para optar al grado académico de Licenciado
en Ingeniería en Ciencias de la Computación

Guatemala
2016

Herramientas web para el monitoreo de servicios en Centros de Operación de Redes

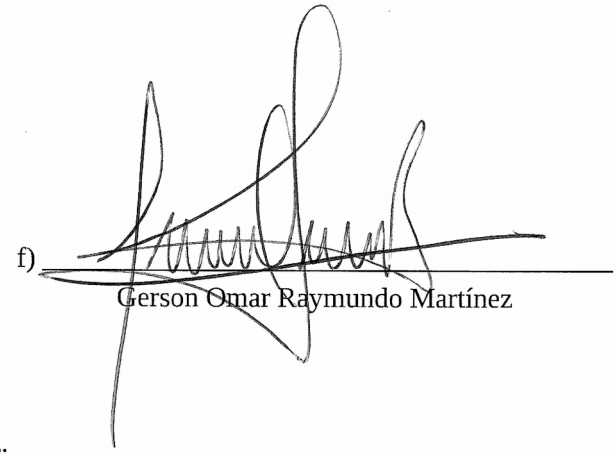
UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Herramientas web para el monitoreo de servicios en Centros de Operación de Redes

Trabajo de graduación en modalidad de Tesis presentado por
Donald Antonio Velásquez Aguilar
para optar al grado académico de Licenciado
en Ingeniería en Ciencias de la Computación

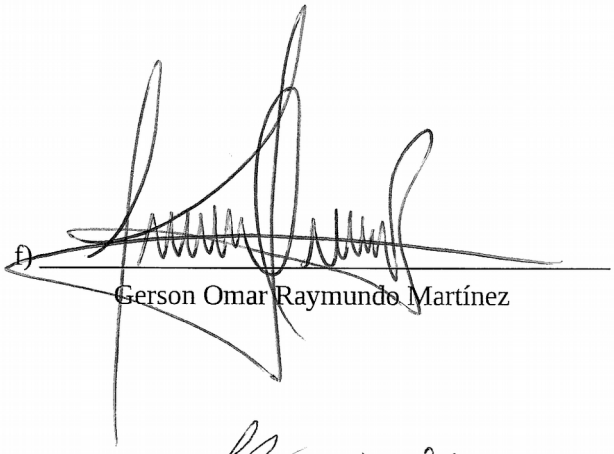
Guatemala
2016

Vo. Bo. :



f) _____
Gerson Omar Raymundo Martínez

Tribunal Evaluador:



f) _____
Gerson Omar Raymundo Martínez

f) _____
Douglas Leonel Barrios Gonzalez

f) _____
Etson Saúl Guerrero

Fecha de Aprobación: Lunes 5 de diciembre de 2016

ÍNDICE

LISTA DE TABLAS

LISTA DE FIGURAS

RESUMEN

En el presente trabajo de graduación, se presenta el resultado del desarrollo de una herramienta Web para el monitoreo de una red MPLS de equipos Juniper y los servicios VPN Capa 3 basados en BGP/MPLS. Además del monitoreo por medio de SNMP, se realiza una interpretación de la configuración de los elementos de los equipos, para lograr asociar los RPM's, VRF's e interfaces a una misma VPN. La configuración de monitoreo de un equipo se limita a activar los elementos que se desean monitorear de un servicio, como el rendimiento, el consumo de tráfico, tablas de rutas, entre otros.

Además de ls herramientas de monitoreo, se proveen herramientas para la ejecución de pruebas de Ping, Traceroute y Rutas BGP sobre la red de Internet, por medio de un Looking Glass, y pruebas de Ping sobre servicios VPN Capa 3 basados en BGP/MPLS. Estas se orientan a apoyar a los Ingenieros de NOC en las actividades de diagnóstico de servicios. Por último, se busca centralizar la información de las alertas generadas de los diferentes elementos en un tablero de alertas, donde se categorizan por criticidad de la alerta y tipo de cliente definido en el Sistema, orientado a la proactividad de los ingenieros de NOC, con la cual también es posible la generación de tickets de seguimiento.

I. INTRODUCCIÓN

Resulta difícil imaginarnos seguir realizando nuestras actividades diarias sin el uso de la tecnología. Tenemos una gran dependencia a la tecnología; ésta se emplea en diversos ámbitos o aspectos, ya que nos permite la ejecución de tareas de forma más efectiva y eficiente.

El éxito de muchas aplicaciones de la tecnología en ámbitos empresariales, educativos y de entretenimiento es la transmisión de información a través de una red ya sea personal, local, o incluso en internet. Esta tendencia se ha popularizado desde la introducción del computador personal, y se ha extendido aún más luego de la introducción de dispositivos inteligentes (teléfonos inteligentes, tabletas, televisores inteligentes, etc.), cuyas aplicaciones en muchos casos dependen de estar constantemente conectados a servidores en la nube o Internet para la actualización, procesamiento y enriquecimiento de datos, para brindar información de utilidad a los usuarios (mapas, noticias, redes sociales, juegos, etcétera).

Para los Proveedores de Servicios de Internet (ISP, por sus siglas en inglés *Internet Service Provider*), brindar un servicio estable, con bajos tiempo de latencia o retardo (RTT, *Round Trip Time*) es un factor diferenciador que fomenta la fidelidad de los clientes. Para asegurar mantener una red robusta y fiel, es importante contar con un buen control de la red principal o Red Core del ISP, por donde se transportan grandes cantidades de tráfico.

Un Centro de Operaciones de Redes o NOC (*Network Operation Center*), es el departamento dentro de un ISP que se encarga de la recepción de reportes de clientes, diagnosticar, dar seguimiento y, generalmente, resolver las fallas en los servicios proveídos a los clientes. Para evitar que muchos incidentes se conviertan en fallas, es importante también contar con un monitoreo constante de las diversas redes del ISP, y asegurar correcto funcionamiento de equipos, líneas de comunicación y servicios de clientes.

Existen mucha información que se puede recolectar de los equipos empleados en las redes de los ISP y parámetros diversos de acuerdo a la arquitectura de la red, incluso dependiendo del fabricante, que se pueden utilizar como indicadores para determinar el correcto funcionamiento de una red, riesgos e incidentes, que pueden ser corregidos antes de que se conviertan en una falla.

Actualmente existen muchas herramientas que proveen información del estado de los servicios, ya sea desarrolladas por terceros o por el mismo proveedor de equipos, capaces de proveer el estado de un equipo y el estado de sus interfaces de red. Aún así, es necesario agregar el monitoreo de servicios de clientes considerados como críticos, en donde proveer una alta disponibilidad es prioritario.

II. OBJETIVOS

A. GENERAL

1. Desarrollar una aplicación Web que facilite a ingenieros del NOC tener un control del estado de la Red del ISP y de de los servicios brindados por el ISP.

B. ESPECÍFICOS

1. Permitir a los ingenieros de NOC visualizar el estado de los servicios.
2. Permitir a los ingenieros de NOC realizar pruebas controladas en tiempo real sobre los servicios contratados dentro de la red del ISP.
3. Facilitar a los ingenieros del NOC realizar revisiones en los servicios reportados por los clientes del ISP.
4. Permitir a los ingenieros del NOC realizar acciones correctivas de forma proactiva en los servicios.
5. Diseñar y desarrollar una herramienta modular, que facilite el desarrollo de nuevos módulos.
6. Permitir la apertura de tickets en una herramienta externa al detectar un incidente en la red del ISP o en un servicio.

III. JUSTIFICACIÓN

Para las empresas, independientemente del giro del negocio, los medios de comunicación vienen a resolver muchas de sus necesidades. Uno de los medios con un amplio uso es el internet, el cual es utilizado para brindar una gran variedad de soluciones. Por la necesidad que representa el estar conectado a la *nube*, no sólo en el ámbito laboral, surgen diferentes empresas dedicadas a brindar servicios de internet, denominados Proveedores de Servicios de Internet (ISP, por sus siglas en inglés).

Dentro del ámbito empresarial, una solución que ha sido bastante aceptada, sobre todo para las empresas con presencia internacional, es el uso de Redes Privadas Virtuales (VPN, por sus siglas en inglés), el cual permite a equipos ubicados a grandes distancias geográficas, interactuar a través de internet como si se tratara de una Red de Area Local (LAN, por sus siglas en inglés).

La construcción de una VPN puede ser bastante compleja, sobre todo si se utilizan diversos puntos con grandes distancias, ya que se involucran diferentes tipos de equipos, diferentes protocolos de comunicación, diferentes ISP, etc. Lo anterior se ha facilitado luego de la introducción de MPLS (*Multi-Protocol Layer Switch*), el cual, como su nombre indica, provee un estándar para la creación de utilizando diferente tecnología y protocolos, la cual es fácilmente escalable.

Debido al rápido crecimiento de las redes, las empresas que proveen servicios de Internet han tenido la necesidad de contar con Centros de Operación de Redes (NOC, por sus siglas en inglés) y herramientas para el control o monitoreo de redes. Por la misma necesidad, actualmente se pueden encontrar diversas herramientas que realizan monitoreo de redes, muchas de las cuales se basan en SNMP (*Simple Network Monitoring Protocol*), para comunicarse con los diferentes equipos y recolectar información, por ejemplo: estado del equipo, estado de interfaces, estado de protocolos, consumo de recursos, consumo de ancho de banda, etc. Sin embargo, la mayoría de las herramientas que se pueden encontrar están orientadas para el uso de operadores de NOC.

Es importante tomar en cuenta que en muchos casos, los clientes de un ISP son empresas que también desempeñan esta misma función, y para estos últimos es necesario conocer detalles técnicos de los servicios que están contratando. Sin embargo, representa un alto riesgo para la operación del ISP, permitir que sus clientes tengan acceso para monitorear los equipos, pues los clientes podrían acceder a información de otros clientes o de la infraestructura que se tiene, lo que permitiría la existencia de ventanas de seguridad y facilitaría que se den ataques a la red del ISP o de los clientes. De lo anterior surge la necesidad de crear una herramienta que se adapte a las necesidades de brindar una visibilidad controlada a los clientes de los servicios contratados sin dejar en riesgo a la operación.

La herramienta a desarrollar se está estructurando para trabajar con equipos Juniper, ya que se están utilizando OID's (*Object Identifier*) propios de estos equipos para el monitoreo de RPM (*Real-time Performance Monitoring*). Sin embargo, como se indica en el Resumen, se está trabajando en el diseño de una herramienta que permita fácilmente agregar nuevos módulos, para continuar con el trabajo de desarrollo en esta herramienta que quedaría fuera del alcance que se está trazando para este trabajo de graduación.

IV. MARCO TEÓRICO

En la actualidad, tenemos a nuestra disposición cientos de dispositivos capaces de conectarse a una red, y cada vez se vuelve más importante que estos se encuentren conectados; ya sea desde una red personal hasta la red mundial Internet. Es importante resaltar que aunque estos dispositivos fueron desarrollados por diferentes fabricantes, utilizan diferentes medios para de conexión (*Ethernet*, *Wireless*, *Bluetooth*, etc.), e incluso utilizan diferentes protocolos de comunicación, la intercomunicación entre ellos es posible.

Durante el surgimiento de las redes de computadoras, existió bastante iniciativa de los diferentes fabricantes para lograr la interconexión entre sus equipos. Sin embargo, debido a que cada fabricante desarrolló su propia tecnología, la comunicación con los equipos de otros fabricantes no era posible. Debido a la necesidad de lograr comunicación entre equipos de diferentes fabricantes, fueron desarrollados los Modelos de Interconexión con la finalidad de romper la barrera creada por las diferencias en los diversos dispositivos (Lammle, 2013).

A. MODELO OSI

El Modelo de Interconexión de Sistemas Abiertos, conocido como Modelo OSI (siglas del nombre en inglés: *Open System Interconnection*), se basa en la propuesta desarrollada por la Organización Internacional para la Estandarización (*International Organization for Standardization*, ISO). Como su nombre indica, el Modelo OSI se refiere a los sistemas que están abiertos a la comunicación con otros sistemas (Tanenbaum, 2012).

Este modelo divide la comunicación entre dispositivos en siete capas, las cuales son: Capa Física, Capa de Enlace de Datos, Capa de Red, Capa de Transporte, Capa de Sesión, Capa de Presentación y Capa de Aplicación. Los principios aplicados al desarrollo de este modelo en capas se puede resumir de la siguiente forma:

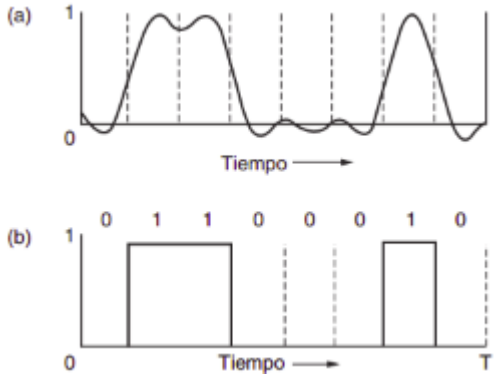
- Se debe crear una capa en donde se requiera un nivel diferente de abstracción.
- Cada capa debe realizar una función bien definida.
- La función de cada capa se debe elegir teniendo en cuenta la definición de protocolos estandarizados internacionalmente.
- Es necesario elegir los límites de las capas de modo que se minimice el flujo de información a través de las interfaces.
- La cantidad de capas debe ser suficiente para no tener que agrupar funciones distintas en la misma capa, y lo bastante pequeña para que la arquitectura no sea inmanejable (Tanenbaum, 2012).

El modelo OSI también puede ser visto como un modelo jerárquico y, debido a su abstracción en capas, este modelo ofrece diferentes ventajas:

- Se divide el complejo proceso de las redes de comunicación en procesos simples y pequeños, facilitando el desarrollo, diseño y solución de problemas.
- Permite a los diferentes fabricantes el desarrollo de componentes de redes a través de la estandarización.
- Orienta a la industria a la estandarización, definiendo claramente las funciones que deben hacer en cada capa del modelo.
- Permite a varios tipos de hardware y software comunicarse.
- Previene que los cambios en una capa afecten a las demás capas. (Lammle, 2013)

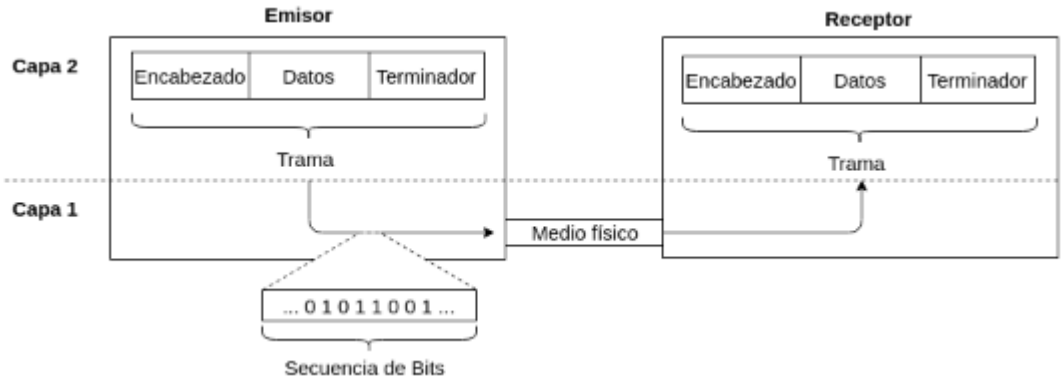
1. Capa física. La capa física, capa 1 o capa inferior del modelo OSI, es la encargada de la transmisión y recepción de la secuencia de bits (acrónimo del inglés *Binary Digit*) no estructurados a través de un medio físico. La transmisión se realiza utilizando un pulso eléctrico (electromagnético u óptico), de modo que un alto o bajo voltaje representa un 1 lógico o un 0 (cero) lógico respectivamente (Blank, 2004), como se representa en la Figura 1.

Figura 1. (a) Pulso eléctrico enviado por un medio físico. (b) Interpretación del pulso eléctrico en una secuencia de bits.



2. Capa de enlace de datos. Esta capa se encarga de la transmisión física de datos y del manejo de notificación de errores, topología de la red y control de flujo. En otras palabras, la capa de enlace de datos se asegura de que los mensajes sean entregados correctamente al equipo debido dentro de la red de Área Local o LAN (siglas del inglés *Local Area Network*) (Lammle, 2013). Los mensajes manejados por la capa de Enlace de Datos se realizan en unidades de información denominadas tramas, tramas de red o tramas de datos (en inglés *data frame*). En una trama, los datos a transmitir son encapsulados añadiendo un encabezado y un terminador de trama (Tanenbaum, 2012). En la Figura 2 se representa una estructura de trama y su relación con la Capa 1.

Figura 2. Representación del encapsulado de una trama y relación entre Capa 2 y Capa 1



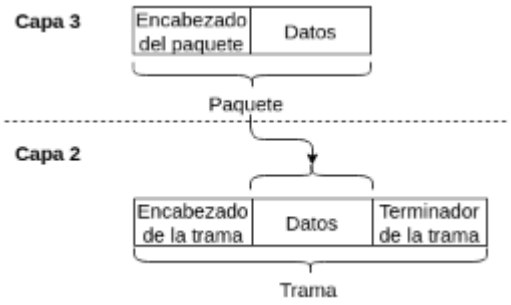
El encabezado de la trama inicia con un preámbulo, que es una secuencia de bits utilizados para reconocer el inicio de la misma. A continuación se incluye información manejada en la capa de enlace de Datos como la dirección MAC (siglas del inglés *Media Access Control*) destino, la dirección MAC origen, el tipo de protocolo utilizado (por ejemplo *Ethernet* o *ATM*) y la longitud de los datos contenidos en la trama. Por su parte, el terminador contiene una secuencia de bits para la verificación de la trama (Lammle, 2013). Luego de la construcción de una trama del lado del receptor, ésta es enviada a la capa física para ser transmitidos de forma secuencial, que luego espera a que el receptor envíe una confirmación de recepción o una notificación de error (Tanenbaum, 2012).

Una dirección MAC, dirección física o dirección de Hardware, es un identificador único para cada Tarjeta o Interfaz de Red, conformado por 48 bits, ó 6 octetos, y comúnmente representado por 6 pares de dígitos hexadecimales (por ejemplo: 00:23:df:a6:be:32) (Blank, 2004). El protocolo MAC define la forma en que los paquetes (entendiendo por paquetes como unidades de información de la capa superior) son colocados en el medio físico, es decir, regulan el uso del medio físico por los distintos elementos que lo comparten. Un manejo eficiente del medio físico ayuda a evadir problemas como la colisión de paquetes y la saturación de datos a un receptor lento por parte de un emisor rápido (Lammle, 2013).

3. Capa de red. La tercera capa del Modelo OSI, la Capa de red, se encarga de la administración de direcciones de dispositivos, de mapear las localizaciones de los dispositivos en la red y de determinar la mejor ruta para enviar los datos en estructuras llamadas paquetes (Lammle, 2013). Esta capa permite que se pueda enviar datos a dispositivos que no se encuentran conectados a nuestra LAN, pasando por diversos equipos o enrutadores intermedios, comúnmente conocidos por su nombre en inglés router (Tanenbaum, 2012). En esta capa existen diversos protocolos que trabajan de forma simultánea para lograr una eficiente comunicación a nivel de Capa 3, siendo uno de los más importantes el Protocolo de Internet (IP, por sus siglas en inglés: *Internet Protocol*), que se detalla en las siguientes secciones.

De forma similar a la Capa 2, en esta capa los *paquetes de red* manejados incluyen un encabezado, como se ilustra en la Figura 3. Este encabezado es utilizado por los enrutadores o *routers* intermedios para reconocer los paquetes y enrutar estos al destinatario o siguiente enrutador intermedio (Blank, 2004). Es importante notar que los paquetes enviados de la Capa 3 a la Capa 2 pueden ser fragmentados en diversas tramas (por lo general de algunos cientos o miles de bytes) por el emisor y transmitidos secuencialmente, para luego ser reconstruidos por el receptor cuando la información se entrega de la Capa 2 a la Capa 3 del receptor (Tanenbaum, 2012).

Figura 3. Relación entre un paquete y una trama.



Cuando un paquete tiene que viajar de una red a otra para llegar a su destino, pueden surgir muchos problemas. El direccionamiento utilizado por la(s) siguiente(s) red(es) puede ser distinto del que utilizan las anteriores. La Capa 3 se responsabiliza de solucionar los problemas y permitir la comunicación de redes heterogéneas (Tanenbaum, 2012).

4. Capa de transporte. La función de la capa de transporte es aceptar datos de la capa superior, dividirlos en unidades más pequeñas de ser necesario, pasar estos datos a la capa de red y asegurar que todas las piezas lleguen de forma exitosa al destinatario de forma eficiente. Esta capa se encarga de abstraer los diferentes tipos de tecnologías de hardware y los cambios que estas puedan tener; de esta forma, las capas superiores no deben preocuparse del manejo de comunicación con otros dispositivos (Tanenbaum, 2012).

Otra función de la Capa 4 es la de determinar el tipo de servicio que se debe proveer a la Capa de Sesión y a los usuarios de red. El tipo más popular de conexión de transporte es un canal punto a punto libre de errores (entendiendo por esto que la tasa de errores es muy baja para poder ignorarla) que entrega los mensajes o bytes en el orden en el que se enviaron, para el cual se utiliza el Protocolo de Control de Transmisión (TCP, de sus siglas en inglés *Transmission Control Protocol*) (Tanenbaum, 2012).

Existe otro tipo de servicio de transporte, el de mensajes aislados sin garantía sobre el orden de la entrega y la difusión de mensajes a múltiples destinos, que utiliza el Protocolo de Datagramas de Usuario

(UDP, de sus siglas en inglés *User Datagram Protocol*) (Tanenbaum, 2012). Sin embargo, este servicio tiene la ventaja de que es más rápido ya que no se espera una confirmación de recepción de los mensajes enviados (Blank, 2004).

Esta capa también cuenta con un control de tráfico de mensajes, la cual le indica al transmisor detener el *stream* de datos enviado en un canal específico cuando el receptor está saturado, es decir, que no hay espacio disponible el en búfer (o búferes) del receptor (Blank, 2004).

A cada mensaje enviado se le agrega un encabezado, donde se incluye el control de la información como el inicio y fin del mensaje, para el reconocimiento en la Capa de Transporte del receptor, y la información de la secuencia de los mensajes, para permitir que el receptor extremo obtenga las piezas en el orden correcto antes de entregar el mensaje recibido a la capa superior (Blank, 2004)

Un aspecto importante de esta capa es la introducción del concepto de puertos. Un puerto de red es una interfaz para comunicarse con un programa a través de una red y suele estar identificado por un valor numérico, que va de 0 (cero) a 65535 (un valor de 16 bits que va de 0000 a FFFF en nomenclatura hexadecimal). Cada canal de comunicación requiere de dos puertos: un puerto origen y uno destino asignados al emisor y receptor respectivamente; de esta forma, el emisor puede enviar información y esperar respuestas por parte del receptor. La asignación de puertos permite que una máquina tenga la capacidad de establecer simultáneamente diversas conexiones con máquinas distintas (Blank, 2004).

Aunque básicamente se puede utilizar cualquier puerto para establecer un canal de comunicación con otro dispositivo, existen regulaciones por parte de la organización *Internet Assigned Numbers Authority*, IANA por sus siglas, plasmados en el RFC 6335. La asignación creada por la IANA, divide los puertos en tres rangos (Internet Assigned Numbers Authority, 2016c):

- Puertos del sistema: abarca del puerto 0 al 1023 (del 0000 al 03FF en hexadecimal). Estos puertos son reservados para uso exclusivo de servicios del sistema operativo. La IANA tiene un listado de los servicios que corre en cada puerto (Internet Assigned Numbers Authority, 2016c), y aunque existe la posibilidad de cambiar utilizar uno de estos puertos para aplicaciones de usuarios, no es recomendable, ya que se debe desactivar el servicio que está utilizando el puerto deseado pues dos aplicaciones no pueden utilizar el mismo puerto.
- Puertos de usuarios: abarca del puerto 1024 al 49151 (del 0400 al BFFF en hexadecimal). Estos puertos pueden ser utilizados por cualquier aplicación de usuario, aunque existen varios puertos asignados para el uso de algunas aplicaciones o protocolos. Por ejemplo la Base de Datos de PostgreSQL, que por defecto utiliza el puerto 5432 (Internet Assigned Numbers Authority, 2016c), aunque en su configuración el puerto a utilizar puede ser cambiado.

- Puertos dinámicos o privados: abarca del 49152 al 65535 (del C000 al FFFF en hexadecimal). Estos puertos son utilizados por el sistema operativo, tanto del emisor como del receptor, cuando una aplicación tiene que conectarse a un servidor y por tanto necesita un puerto por donde salir (Internet Assigned Numbers Authority, 2016c). Por ejemplo, al establecer una sesión SSH (acrónimo del inglés *Secure Shell*), que utiliza el puerto 22, si se mantiene la sesión en este puerto del receptor, estará ocupado para nuevas sesiones SSH, por lo que el Sistema Operativo del receptor verifica entre sus puertos dinámicos, para asignar el siguiente puerto disponible y éste es el que se utilizará para el canal de comunicación. Por su parte el Emisor realiza una tarea similar, asignando un puerto dinámico para completar el canal de comunicación.

5. Capa de sesión. La capa de sesión tiene las responsabilidades de establecer, administrar y de terminar sesiones entre las capas de presentación de los dispositivos involucrados, además de mantener los datos de diferentes aplicaciones separados. El control de diálogo entre dispositivos también ocurre en esta capa (Lammle, 2013).

La comunicación entre aplicaciones de dispositivos en la capa de sesión, es coordinada y organizada por medio de tres diferentes modos: *simplex*, *half-duplex* y *full-duplex*. El modo simplex, es un tipo de comunicación de una vía, de modo que uno de los dispositivos únicamente puede enviar información y el segundo dispositivo sólo recibirá esta información. El modo half-duplex, funciona como un canal con dos vías de comunicación, pero sólo ocurrirá en una dirección a la vez, de manera análoga al funcionamiento de radios o un walkie-talkie. Por último, el modo full-duplex, funciona como un canal con dos vías de comunicación, en la que los datos pueden ser enviados y recibidos al mismo tiempo (Lammle, 2013).

6. Capa de presentación. A diferencia de las capas inferiores, que se enfocan principalmente en mover información de un lado a otro, la capa de presentación se enfoca en la sintaxis y la semántica de la información transmitida (Tanenbaum, 2012). Esta capa se puede ver como un traductor, pues se encarga de que los datos enviados, que llevan el formato utilizado por la capa de aplicación del dispositivo emisor, sea traducido a un formato común entre ambos dispositivos, emisor y receptor, para luego trasladar este formato común en un formato comprensible para la capa de aplicación del dispositivo receptor (Blank, 2004). En la capa de presentación se proporciona lo siguiente:

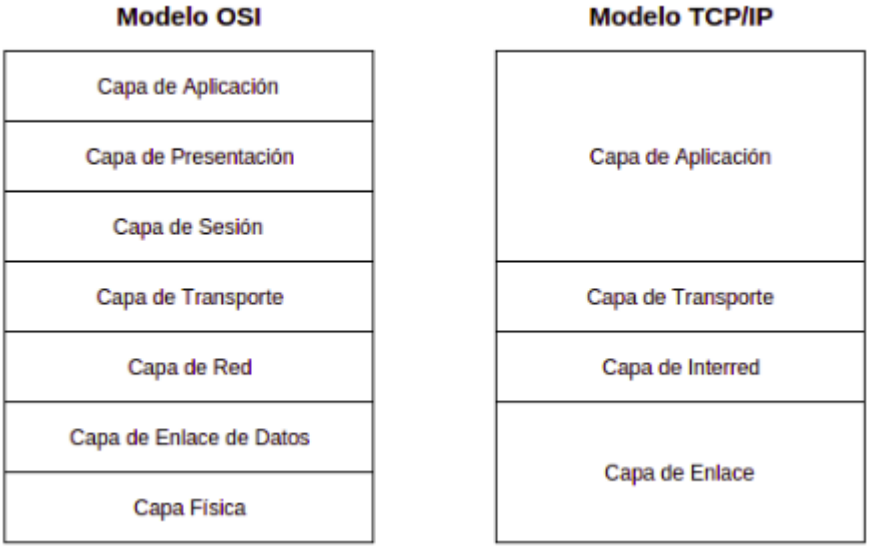
- Traducción de códigos de caracteres: por ejemplo de ASCII (siglas del nombre en inglés: *American Standard Code for Information Interchange*) a EBCDIC (siglas del nombre en inglés: *Extended Binary Coded Decimal Interchange*).
- Conversión de datos: Bits de orden, punto de salto, etc.
- Compresión de datos: reduce el número de bits que es necesario transmitir en la red.
- Cifrado de datos: en muchas aplicaciones es necesario cifrar los datos por motivos de seguridad. Por ejemplo, el cifrado de contraseñas (Blank, 2004).

7. Capa de aplicación. La Capa de Aplicación describe el funcionamiento de los programas de aplicación, por un lado su interacción con la Capa de Presentación y por el otro, la interacción con el usuario (Lammle, 2013) y se compone de diversos protocolos empleados por los usuarios con frecuencia. Posiblemente el más utilizado sea el Protocolo de Transferencia de Hipertexto o HTTP (acrónimo de su nombre en inglés: *Hypertext Transfer Protocol*), que es la base para la WWW (siglas del nombre en inglés de *World Wide Web*), usado ampliamente para la navegación Web (Tanenbaum, 2012).

B. MODELO TCP/IP

El modelo de referencia TCP/IP, es el modelo de referencia que se utiliza en redes ARPANET, la primera Red de Área Amplia (WAN, de sus siglas en inglés: *Wide Area Network*), y su sucesor Internet. Su nombre deriva de los dos protocolos primarios que utiliza para su funcionamiento. A diferencia del Modelo OSI, este modelo se compone únicamente de cuatro capas: Capa de enlace, Capa de interred, Capa de transporte y Capa de aplicación (Tanenbaum, 2012). En la Figura 4 se presenta una comparación de los modelos OSI y TCP/IP

Figura 4. Comparativa del Modelo de Referencia OSI y el Modelo de Referencia TCP/IP



1. Capa de enlace. La capa de enlace bajo el Modelo TCP/IP, describe qué enlaces (líneas seriales y Ethernet) se deben llevar a cabo para cumplir con las necesidades de la Capa de Interred. Su funcionalidad es equiparable a las que realizan las Capas 1 y 2 del Modelo OSI (Tanenbaum, 2012).

2. Capa de interred. Esta capa es el eje que mantiene unida a toda la arquitectura, se observa que tiene correspondencia con la Capa 3 del Modelo OSI, ya que sus funciones son bastante

similares. La Capa de interred define un formato de paquete y un protocolo oficial llamado IP, además de un protocolo complementario Protocolo de Mensajes de Control de Internet o ICMP (siglas del nombre en inglés *Internet Control Message Protocol*) que le ayuda en su funcionamiento. Esta capa se encarga de la entrega de paquetes a donde están destinados, aunque muchas veces no son entregados en el orden en que fueron enviados, caso en el que las Capas superiores se encargan de ordenarlos de ser necesario (Tanenbaum, 2012).

3. Capa de transporte. Esta capa está diseñada para que las entidades pares, en los nodos de origen y de destino creen canales de comunicación, al igual que en la Capa de transporte del modelo OSI. Al igual que en el modelo OSI, esta capa usa principalmente TCP (Protocolo de Control de Transmisión) que, como se mencionó anteriormente, permite el flujo de información sin errores, aunque también se puede utilizar UDP (Protocolo de Datagrama de Usuario) para establecer conexiones rápidas, útil en aplicaciones de transmisión de voz o video (Tanenbaum, 2012).

4. Capa de aplicación. En el modelo TCP/IP no se incluyen las Capas de sesión y Presentación, pues fueron consideradas innecesarias, incluso dentro del Modelo OSI estas capas son poco utilizadas. Las aplicaciones simplemente incluyen cualquier función de sesión y de presentación que requieran (Tanenbaum, 2012).

C. PROTOCOLO DE INTERNET

Se ha mencionado el Protocolo de Internet (IP, siglas de su nombre en inglés: I*nternet Protocol*) en la Capa 3 del Modelo OSI, la Capa de Red. IP es el principal protocolo para el funcionamiento de la Capa 3 del Modelo OSI, es decir, de la interconexión de redes y del mismo Internet, otros protocolos existentes le brindan soporte a IP (Lammle, 2013).

Existen dos versiones ampliamente utilizadas de IP: Internet Protocol Versión 4, referenciando comúnmente como IPv4, e Internet Protocol Versión 6, IPv6. En ambas versiones de IP se introduce el concepto de Dirección IP, que es una dirección lógica que cada interfaz de red tiene; esta dirección, tanto del emisor como del receptor, se coloca en el encabezado de los paquetes transmitidos, y se utiliza para el que los equipos enrutadores intermedios sepan cómo direccionarlos hasta su destino (Tanenbaum, 2012).

Piense que las direcciones IP funcionan como la dirección de una casa, cada casa tiene una dirección única y sigue un orden. Cuando se desea enviar una carta, primero se envía esta carta a la oficina postal, el personal revisa la dirección y si ésta se encuentra en el área cubierta por la oficina postal se procede a la entrega de la carta, de lo contrario se envía a la siguiente oficina postal para su procesamiento.

1. Direcciones IP. Las direcciones IP en IPv4 es un número compuesto por 32 bits, divididos en cuatro grupos de 8 bits (es decir, en octetos o bytes). Para facilidad de lectura humana, los valores se escriben en notación decimal separados por un punto, por ejemplo la dirección IP 192.168.10.10. La dirección IP también cuenta con una máscara de red, que también es un número de 32 bits, que tiene la finalidad de delimitar qué bits forman parte Dirección de Red y cuales indican el host, de ahora en adelante se refiere de esta forma a una interfaz de red con una IP, específico en la red. La máscara de red puede representarse de similar forma a la dirección IP o contando el número de bits con valor de 1, pues la máscara de red siempre será una cantidad n de bits con valor 1 seguido de 32-n bits con valor 0, donde n es menor a 32, así la máscara 11111111 11111111 11111111 00000000 puede representarse como 255.255.255.0 ó 24 (Lammle, 2013). Una dirección IP con su máscara de red se representa de la forma: 192.168.10.10/24.

Una Dirección de Red se obtiene aplicando la operación AND entre la dirección IP y su máscara de red. Haciendo uso del ejemplo anterior, con la dirección 192.168.10.10/32, la dirección de red sería 192.168.10.0. En la Figura 5 se obtiene de la siguiente forma:

Figura 5. Ejemplo de cálculo de una dirección de red a partir de la dirección IP y la máscara de red.

	Representación decimal	Representación binaria
Dirección IP	192.168. 10.10	11000000 10101000 00001010 00001010
Máscara de Red	255.255.255. 0	11111111 11111111 11111111 00000000
Dirección de Red	192.168. 10. 0	11000000 10101000 00001010 00000000

En el ejemplo de la Figura 5, se observa que para cualquier valor del último octeto se tendrá la misma dirección de red. Esto nos da la posibilidad de tener 256 valores, del 0 al 255, diferentes en el último octeto que pueden ser utilizados para direcciones de hosts, aunque no pueden ser utilizados pues existen algunas direcciones reservadas en una red: la propia dirección de red (192.168.10.0) y la dirección de broadcast (192.168.10.255). Como resultado, en una red con máscara 24, se tienen 254 direcciones para hosts: desde la 192.168.10.1 a la 192.168.10.254 (Davies, 2008). Como regla general, si tenemos una máscara de red n, el número de hosts disponibles será el resultado de operar 2^(32-n)-2. Las direcciones IP se dividen en 5 clases, enumeradas de la Clase A a la Clase E. Cada una de las clases contiene un rango de direcciones IP, las cuales se resumen en la Tabla 1.

Tabla 1. Rangos de direcciones IP por Clase de direcciones IP.

Clase de direcciones IP	Rango de direcciones IP
Clase A	1.0.0.0 - 126.0.0.0

Continuación Tabla 1.

Clase de direcciones IP	Rango de direcciones IP
Clase B	128.0.0.0 - 191.255.0.0
Clase C	192.0.0.0 - 223.255.255.0
Clase D	224.0.0.0 - 239.255.255.255
Clase E	240.0.0.0 - 255.255.255.255

Las redes Clase A utilizan máscaras de red 255.0.0.0, y son utilizados para redes con numerosos hosts, las redes Clase B emplean máscaras de red 255.255.0.0 y por último las redes Clase C utilizan máscaras de red 255.255.255.0. Las redes Clase D son utilizadas para multicast y las redes Clase E son para uso experimental, incluso algunos sistemas operativos Windows no soportan el uso de direcciones Clase E (Davies, 2008).

Es notorio que faltan algunos rangos de direcciones IP en la Tabla 1, y esto se debe a que existen algunos segmentos reservados, pero además dentro de las Clases A, B y C, existen segmentos de direcciones IP reservados por la IETF (por sus siglas en inglés, *Internet Engineering Task Force*) y la IANA (*Internet Assigned Numbers Authority*). En la Tabla 2 se presentan los segmentos reservados o segmentos IPv4 registrados con propósito especial y el uso de cada segmento de acuerdo (Internet Assigned Numbers Authority, 2016b).

Tabla 2. Segmentos de direcciones IPv4 reservados por la IETF y la IANA.

Segmento	Descripción
0.0.0.0/8	“This host on this network”
10.0.0.0/8	Redes de uso privado
100.64.0.0/10	Espacio de redes compartido
127.0.0.0/8	Loopback
169.254.0.0/16	Enlace local
172.16.0.0/12	Redes de uso privado
192.0.0.0/24	Asignación de Protocolo IETF
192.0.0.0/29	Prefijo de la continuidad del servicio IPv4
192.0.0.8/32	Direcciones IPv4 dummy
192.0.0.9/32	Protocolo de Control de Puerto Anycast
192.0.0.170/32, 192.0.0.171/32	Descubrimiento NAT64/DNS64
192.0.2.0/24	Documentación (TEST-NET-1)
192.31.196.0/24	AS112-v4
192.52.193.0/24	ATM
192.88.99.0/24	Obsoleto (6to4 Relay Anycast)
192.168.0.0/16	Redes de Uso Privado

Continuación Tabla 2.

Segmento	Descripción
192.175.48.0/24	Delegación directa al Servicio AS112
198.18.0.0/15	Benchmarking
198.51.100.0/24	Documentación (TEST-NET-2)
203.0.113.0/24	Documentación (TEST-NET-3)
240.0.0.0/4	Reservado
255.255.255.255/32	Limited Broadcast

2. Ruteo IP. El anterior ejemplo de la oficina postal se utilizó para representar el funcionamiento de las direcciones IP, pero existe otro concepto que brinda el Protocolo de Internet, que es el ruteo de paquetes (en inglés, el término usado es *routing*). La función del ruteo de paquetes es ejecutada por los routers, o enrutadores intermedios, y es análoga al trabajo que se realiza en las oficinas postales, donde se verifica la dirección IP del host destino y si este host está al alcance del router, procede a la entrega, de lo contrario verifica en su tabla de rutas y envía el paquete al siguiente router. Esta última acción es denominada un salto o *hop* (Tanenbaum, 2012).

El ruteo se apoya de otros protocolos, cuyo propósito es aprender las rutas disponibles que existen en la red, construir tablas de enrutamiento y tomar decisiones. Existen diversos protocolos de ruteo, con diferentes propósitos, pero éstos pueden agruparse en tres clases de acuerdo a su funcionamiento: protocolos de vector de distancia, protocolos de estado de enlace y protocolos híbridos (Lammle, 2013).

Los protocolos de vector de distancia son usados para encontrar la mejor ruta a una red remota juzgando la distancia. En el protocolo de ruteo RIP (*Routing Information Protocol*) la distancia se obtiene por el número de routers por los que debe pasar o saltar (de allí que se le conozca como el número de *hops*) un paquete para alcanzar la red destino, y se selecciona la ruta con el menor número de saltos. Además, RIP comparte periódicamente su tabla de rutas a los routers vecinos, es decir, a los routers conectados directamente (Lammle, 2013). Sin embargo, durante el proceso utilizan una gran cantidad de ancho de banda y son lentos para converger (Juniper Networks, 2012).

En un protocolo de estado de enlace, cada router crea tres tablas de rutas separadas. Una de las tablas mantiene la información de las redes directamente conectadas al router, la segunda tabla determina la topología de la red completa y la tercera, es usada como la tabla de rutas. Los routers que utilizan protocolos de estado de enlace conocen más información sobre la red que la que los routers que emplean protocolos de vector de distancia pueden llegar a obtener (Lammle, 2013). Los protocolos de estado de enlace son más eficientes, incluso en el uso de ancho de banda, pues para compartir sus tablas de rutas, únicamente anuncian actualizaciones del estado de sus rutas cuando se produce un cambio en las redes del

router, favoreciendo a una convergencia más rápida (Juniper Networks, 2012). Las actualizaciones de las tablas de rutas que se envían a los routers vecinos, en realidad son enviados a todos los routers vecinos del área, para que estos últimos también actualicen sus tablas de rutas. El protocolo de ruteo OSPF (*Open Shortest Path First*) es un ejemplo de un protocolo de ruteo completamente de estado de enlace (Lammle, 2013).

Por último, los protocolos híbridos utilizan aspectos de los protocolos de clase vector de distancia y de estado de enlace. Un ejemplo es el EIGRP (siglas de su nombre en inglés, *Enhanced Interior Gateway Routing Protocol*), el cual algunas veces es referenciado como un protocolo de ruteo de distancia de vector avanzado (Lammle, 2013).

Debido a la interconectividad de redes, como en el caso de Internet, se introduce el concepto de Sistema Autónomo o AS (*Autonomous System*), que es el conjunto de routers interconectados en una red, o en una colección de redes, bajo un dominio administrativo común (Juniper Networks, 2013a). Para que los routers tengan la capacidad de reconocer a los routers que se encuentran dentro de un mismo AS, estos están dotados de un número de AS, los cuales se encuentran regulados por la IANA (*Internet Assigned Numbers Authority*) por medio de los Registros de Internet Regional (RIR, siglas del nombre en inglés *Regional Internet Registries*), los cuales se encuentran en la Tabla 3 (Internet Assigned Numbers Authority).

Tabla 3. Registros de Internet Regional.

Registro	Area Cubierta
AFRINIC	Región de Africa
APNIC	Región de Asia del Pacífico
ARIN	Canadá, Estados Unidos y algunas islas del Caribe
LACNIC	América Latina y algunas islas del Caribe
RIPE NCC	Europa, el Medio Oriente y Asia Central

Los protocolos de ruteo tienen una segunda clasificación, de acuerdo al alcance que tienen durante el intercambio de tablas de rutas. El primer tipo está comprendido por los protocolos de puerta interna (IGP, por sus siglas en inglés *Interior Gateway Protocol*), que son aquellos protocolos utilizados para intercambiar información de las tablas de rutas entre los routers de un Sistema Autónomo, por ejemplo el OSPF (*Open Shortest Path First*). El segundo tipo, los protocolos de puerta externa (EGP, siglas de *Exterior Gateway Protocol*), son aquellos que son utilizados para establecer comunicación entre sistemas autónomos; un ejemplo de este tipo de protocolos es el BGP (*Border Gateway Protocol*), el cual por su funcionamiento se clasifica como un protocolo de vector de distancia (Lammle, 2013).

3. Open Shortest Path First. Anteriormente se explica que *Open Shortest Path First*, OSPF, es un protocolo de ruteo de estado usado dentro de un AS, es decir, que es de tipo IGP (*Internal Gateway Protocol*). Debido a la forma en que funcionan los algoritmos de estado de enlaces, permite rápidas convergencias, soporta redes extensas y son menos susceptibles a tener información incorrecta que los algoritmos de vector de distancia. Un router con OSPF envía información de sus redes y el estado de sus enlaces a los demás routers, dentro del mismo sistema autónomo. Esta información transmitida recibe el nombre de anuncios del estado de enlace (LSA, por sus siglas en inglés *Link-State Advertisements*). Los otros routers del mismo AS reciben esta información y la almacenan localmente. Con ello, un router puede conocer todos los enlaces posibles dentro de la red (Juniper Networs, 2012).

Adicional a la tarea de inundación de LSA's y descubrimiento de vecinos, existe una tercera tarea que ejecutan los protocolos de ruteo de estado de enlace: el establecimiento de una base de datos del estado de enlaces (LSDB, acrónimo de su nombre en inglés *Link-State Database*), también llamada base de datos topológica, que almacena todos los LSA's recibidos como una serie de registros y la información importante, como el costo asociado con un enlace o router, para el proceso de cálculo de rutas (Soricelli, 2013).

OSPF utiliza el algoritmo de Dijkstra, también conocido como *Shortest Path First* (SPF) para calcular la ruta más corta a todos los destinos. Dado que cada router OSPF tiene una copia de la LSDB y una tabla de ruta para cada área OSPF, cualquier cambio contenido en un LSA es detectado más rápidos que con los protocolos de vector de distancia y se determinan rutas alternas, ayudando a una rápida convergencia (Soricelli, 2013).

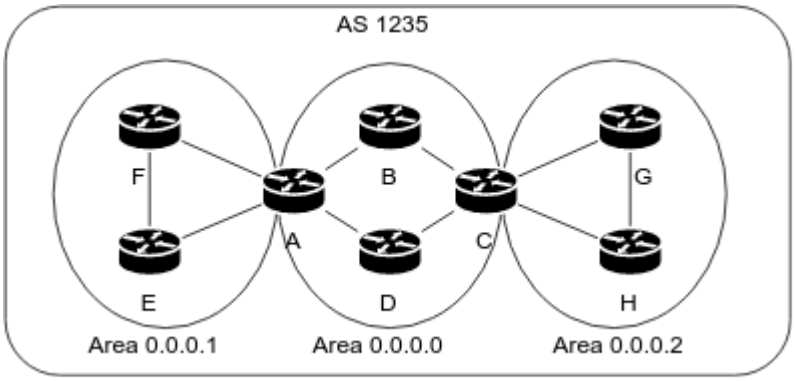
Cuando el número de routers en una red crece, también lo hace la cantidad de información LSDB. Adicionalmente, cada router requerirá más ancho de banda y recursos para lograr inundar la red con LSA's. Para evitar estos problemas, OSPF introduce el concepto de Áreas OSPF, con la cual se limita la inundación de LSA's y se controla el tamaño del LSDB al mantener la información únicamente dentro del área (Soricelli, 2013).

El área de OSPF es un identificador único de 32 bits, comúnmente representado de similar forma a una dirección IPv4. Para la asignación de áreas, debe existir un área principal o backbone, a la que deben de conectarse todas las demás áreas. El área backbone se encarga de interconectar todas las áreas y distribuir toda la información que no sea de backbone entre las áreas, como se muestra en la Figura 6 en la siguiente página (Soricelli, 2013).

Una red puede ser definida completamente como una sola área OSPF, sin embargo, cuando existen diferentes áreas OSPF se tienen también diferentes tipos de routers basados en su localización en la red:

- Router Interno: Es un router que mantiene todas sus interfaces dentro de una área OSPF. En la Figura 6, los routers de E, F, G y H.
- Router Backbone: Es un router que tiene al menos una interface en el área backbone, generalmente área 0 (0.0.0.0 en su representación cuadra-byte). En la Figura 6, los routers B y D.
- Router de Borde de Área: Es un router que conecta una o más áreas OSPF al área backbone. Esto significa que el router de borde de área tiene al menos una interfaz que pertenece al área backbone y una que pertenece a otra área. En la Figura 6, los routers A y C.

Figura 6. Ejemplo de una red OSPF con múltiples áreas.

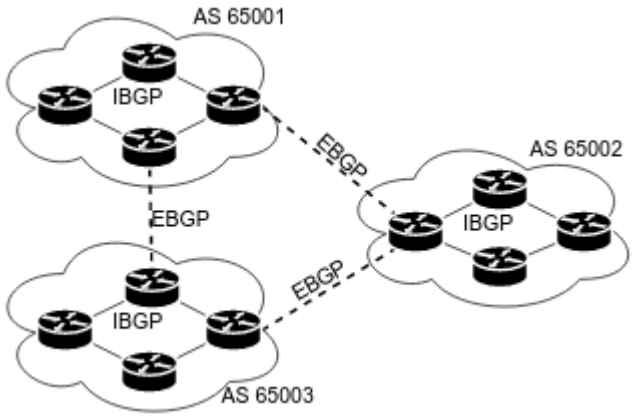


4. Border Gateway Protocol. Este protocolo de ruteo se utiliza para intercambiar información de rutas entre sistemas autónomos (AS), y se refiere a veces como un protocolo de enrutamiento ruta-vector, en inglés como *path-vector*, ya que utiliza un atributo llamado *AS path*, o ruta AS, usado como un vector para evitar bucles de ruteo entre dominios. El término ruta-vector significa que la información de enrutamiento BGP incluye una serie de números AS, que indican el camino a través de diversas redes para alcanzar al destinatario (Soricelli, 2013).

BGP es un protocolo bastante escalable y ofrece un mejor control que un protocolo IGP debido al uso de políticas, la cual es una de las mayores razones de uso en un ISP (*Internet Service Provider*). La información de ruteo de BGP incluye la ruta completa a cada destino. BGP utiliza la información de enrutamiento para mantener una base de información de la capa de información de accesibilidad de la red (NLRI, por las siglas del nombre en inglés *Network Layer Reachability Information*) que intercambia con otros sistemas BGP (Soricelli, 2013).

BGP establece sesiones BGP, que utilizan el protocolo TCP para la comunicación, para poder intercambiar información, la cual puede ser externa (EBGP, siglas del inglés *External BGP*) cuando el intercambio es entre routers de diferentes AS o interna (IBGP, siglas del inglés *Internal BGP*) cuando el intercambio es entre routers del mismo AS como se muestra en la Figura 7 (Soricelli, 2013).

Figura 7. Ejemplo de una red BGP.



5. Multi-Protocol Layer Switching. Multi-Protocol Layer Switching (MPLS) fue presentado inicialmente como una tecnología orientada a mejorar la velocidad que le toma a un router decidir a dónde debe enrutar los paquetes recibidos, actividad conocida como reenvío de paquetes o packet forwarding. Esta necesidad surge debido a que en la arquitectura tradicional de routers, se utiliza el CPU (*Central Processing Unit*) para tareas de mantenimiento de los protocolos de ruteo y el reenvío de paquetes, cuyas capacidades no eran suficientes al tener grandes anchos de banda y altas demandas en la red (Soricelli, 2013).

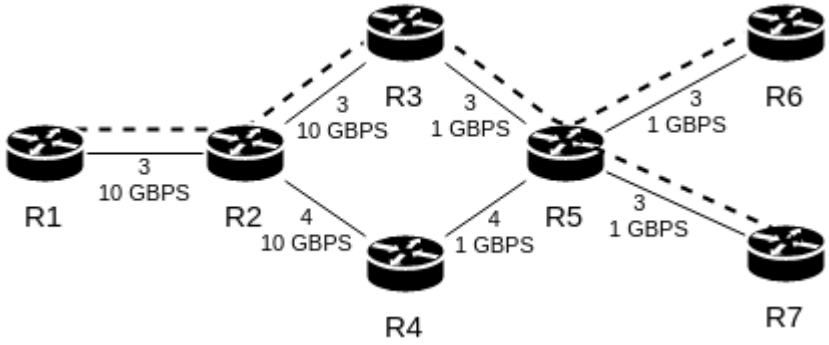
La esencia de MPLS es la generación de una etiqueta corta de longitud fija, que actúa como una representación abreviada de cabecera de un paquete IP. El funcionamiento de esta etiqueta es similar al de un código postal utilizado para identificar ciertas zonas o áreas de un país, facilitando el encaminamiento de un correo (Soricelli, 2013).

Con las nuevas tecnologías y arquitecturas de routers desarrolladas por los fabricantes, que permitían realizar las tareas de forwarding en hardware en lugar de software, se logró mejorar los tiempos de reenvío y MPLS perdió su sentido de existencia. Sin embargo, un nuevo inconveniente surgió derivado de las altas demandas de usuarios a sitios web populares: la existencia de cuellos de botella en la red (Soricelli, 2013).

En la red de ejemplo de la Figura 8, los dos valores agregados a cada enlace representan la preferencia (menor es mejor) y el ancho de banda de cada enlace. El tráfico de los routers R6 y R7 hacia el router R1, cuyas rutas preferidas se representan por la línea punteada, podrían provocar saturación en el enlace entre los routers R3 y R5 (R3-R5 para abreviar). Cambiar la preferencia de alguno de los enlaces R2-R3 o R3-R5 a 6 no es la solución pues el mismo escenario se tendría en el enlace R4-R5. De allí la necesidad de introducir la Ingeniería de Tráfico, refiriéndose a la habilidad de controlar las rutas por las que los paquetes se envían de un extremo a otro dentro de una red. Aplicado al ejemplo anterior, se podría forzar a que el

tráfico del router R6 al router R1 siempre utilice los enlaces R2-R3 y R3-R5, mientras que el del router R7 al R1 utilice los enlaces R2-R4 y R4-R5.

Figura 8. Ejemplo de red propensa a problemas de saturación.



Dentro de las tareas de ingeniería de tráfico, MPLS demostró superioridad sobre otras tecnologías IP disponibles, permitiendo dictar las rutas que determinado tráfico debe tomar a través de la red. Por lo anterior, MPLS es visto como uno de los desarrollos más importantes en redes de computadoras durante la década de 1990, época en la que Internet tuvo un crecimiento importante (Soricelli, 2013).

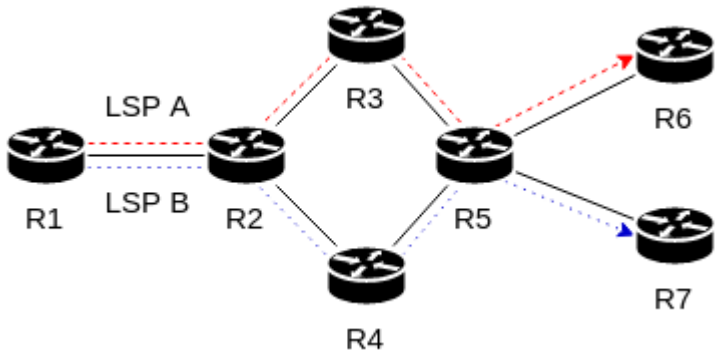
Aunque MPLS fue concebido como independiente de la capa 2, gran parte del éxito de este protocolo MPLS gira en torno a su promesa de proporcionar un medio más eficaz para la implementación de redes IP a través de redes troncales WAN (*Wide Area Network*) basadas en ATM (*Asynchronous Transfer Mode*). Este protocolo se encuentra en constante evolución y sigue siendo soportado por la mayoría de fabricantes de routers, además le permite a un ISP proveer servicios de diferentes tecnologías (como *ATM*, *Frame Relay*, *IPSec* y *Ethernet*) sobre la misma infraestructura MPLS (Soricelli, 2013).

Dentro de su definición, MPLS introduce el concepto de *Label Switched Path* (LSP) que es una ruta unidireccional que típicamente existe dentro de un AS. Estas rutas son los canales por medio de los cuales se enrutan los paquetes dentro de la red. Otro concepto introducido en la definición de MPLS es LSR o *Label Switched Routers*, que se refiere a los routers habilitados para trabajar con éste protocolo y que se encargan del forwarding de paquetes a través de los LSPs definidos en la red. Existen diferentes tipos de LSR basados en su ubicación y la función desempeñada en un LSP (Soricelli, 2013).

- Router de ingreso: es el único punto de entrada de tráfico a un LSP y es el que se encarga de encapsular los paquetes de IP dentro del protocolo MPLS, agregando una etiqueta de 5 bytes de longitud que representa la cabecera del paquete IP de forma abreviada. Esta acción se denomina label push operation. En el ejemplo de la Figura 9, el Router R1 es el router de ingreso para los LSPs A y B.

- Router de tránsito: son llamados así los routers intermedios que componen el LSP. Cada LSP puede contener de 0 a 253 routers de tránsito. Su función es bastante simple, se revisa la etiqueta MPLS de cada paquete recibido y se verifica en la tabla de forwarding de MPLS, al encontrar la etiqueta se realiza una operación de cambio de etiquetas (*label swap operation*) reemplazando la etiqueta de entrada por una de salida y decrementando el contador TTL de MPLS por 1; luego se reenvía el paquete al siguiente salto o *hop* del LSP. Notar que en ningún momento se utiliza la información del encabezado del paquete IP. En la Figura 9, los Routers R2 al R5.
- Penúltimo router: Es uno de los routers de tránsito del LSP, específicamente el penúltimo router en todo el LSP. Este router realiza una función especial denominada l*abel pop operation*, que desencapsula el paquete MPLS y obtiene el paquete IP encapsulado inicialmente en el router de ingreso. Luego de consultar en su tabla de switching de MPLS, se reenvía el paquete IP al siguiente y último salto del LSP después de decrementar el contador TTL de MPLS por una unidad. En la figura 9 el penúltimo router para los LSPs A y B es el Router R5.
- Router de salida: es el último router del LSP. Este router recibe el paquete IP del penúltimo router y ejecuta una operación de búsqueda de rutas en su tabla de rutas y ejecuta el reenvío del paquete al siguiente salto de la ruta (Soricelli, 2013). Corresponde a los Routers R6 y R7 para los LSPs A y B respectivamente.

Figura 9. Ejemplo de trayectorias LSP en una red MPLS



Cabe mencionar que por la naturaleza unidireccional del LSP, es necesario definir dos LSPs entre dos routers en direcciones contrarias para asegurar la comunicación entre los routers. Las rutas para los dos LSP's no necesariamente utilizan los mismos routers de tránsito.

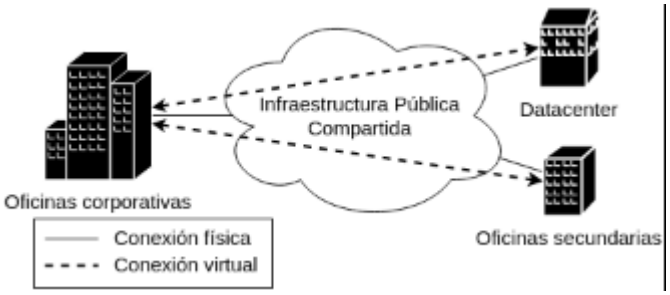
D. RED PRIVADA VIRTUAL

Una Red Privada Virtual (VPN, por sus siglas en inglés *Virtual Private Network*) es aquella Red Privada que utiliza Conexiones Virtuales sobre una red pública o infraestructura compartida para conectar

dos o más sitios remotos, como se ejemplifica en la Figura 10. La conexión es virtual cuando un medio físico se utiliza de forma compartida. Una conexión virtual puede entenderse como un canal o túnel de comunicación aislado en una red. Una VPN es una alternativa rentable a las líneas dedicadas entre redes que resultan en un gasto excesivo de recursos (Juniper Networks, 2010b).

Usualmente, una VPN se separa en dos áreas topológicas correspondientes a la red del proveedor y la red del cliente. La red del cliente comúnmente se encuentra en varios sitios físicos geográficamente distantes y suelen ser privada. El sitio de un cliente consiste en un grupo de routers y otros dispositivos situados en una sola ubicación física. La red del proveedor, que muchas veces comparte su infraestructura con la de Internet, consiste en un grupo de routers que proporcionan el servicio de VPN a la red del cliente, a través de la cual se comunican los diferentes sitios del cliente (Juniper Networks, 2010b).

Figura 10. Diagrama de ejemplo de una VPN.



Para asegurar que una VPN sea privada y se mantenga aislada de otras VPNs, y del servicio de Internet mismo, el ISP define políticas en su red que mantienen la información de las tablas de rutas de las diferentes VPN's separadas. De esta forma, un ISP puede brindar múltiples servicios de VPN sobre la misma infraestructura (Juniper Networks, 2010b).

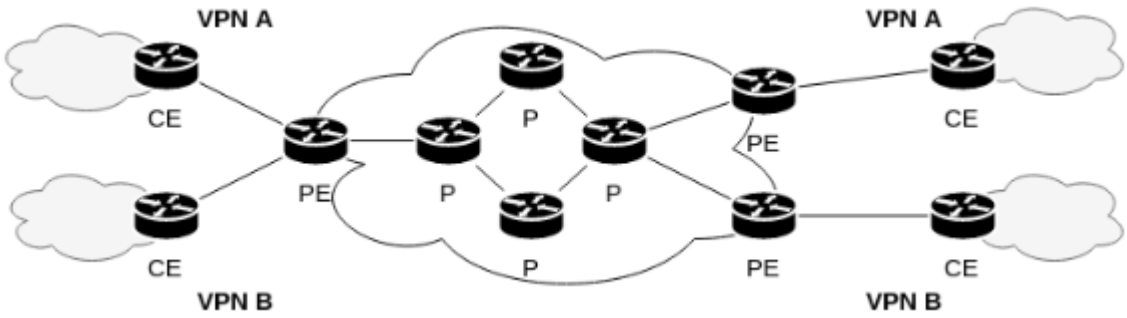
Existen distintos tipos de implementación para una VPN y de formas de clasificarlas basado en la Capa del modelo OSI en la que realiza el túnel de comunicación, los protocolos utilizados para la creación y control de los túneles de comunicación, la ubicación del punto donde termina el túnel de comunicación, entre otros. Aún así, existen dos grupos generales para clasificar una VPN: Sitio a Sitio (*Site-to-Site*) y de Acceso Remoto. Las conexiones Site-to-Site son aquellas en las que existe un dispositivo que se encarga de terminar el túnel de comunicación en cada extremo. Su nombre se deriva de que cada una de éstas conexiones conecta las redes de dos sitios remotos. Las conexiones de Acceso Remoto se componen de aquellas en donde existe un software en uno de los extremos de la red, y conectan un dispositivo a una red (Paquet, 2012).

Las VPN's más comercializadas por los ISP's son de tipo Site-to-Site. Las capas del modelo OSI preferidas para brindar el servicio de VPN son la Capa 2 y Capa 3, para ambos existen diversos protocolos usados para construir una VPN. A partir de ahora nos enfocaremos en las VPN's Capa 3 basada en BGP/MPLS.

1. VPN Capa 3 basada en BGP/MPLS. Este tipo de VPN's se encuentran definidas en el RFC 4364. La VPN se compone de un conjunto de sitios que están conectados a través de una red pública de un ISP. Los sitios comparten información de enrutamiento común y la conectividad de los sitios está controlada por un conjunto de políticas (Juniper Networks, 2013b).

En una VPN de este tipo, los routers del proveedor ejecutar las tareas de enrutamiento y reenvío de los paquetes recibidos de un sitio de cliente hacia otro sitio donde se encuentra el destinatario. Para ello, es necesario que la red del ISP aprenda las redes de los sitios de cliente conectados a la VPN, para luego construir sus tablas de rutas, que deben ser anunciadas y filtradas a toda la red proveedor. Para asegurar el aislamiento de la información de la red del cliente, toda esta información se empaqueta en una instancia de ruteo o routing instance, que es una colección de interfaces, tablas de ruteo y parámetros de protocolos de ruteo. Para las VPN's Capa 3, la instancia de ruteo es denominada *VPN Routing and Forwarding* (VRF). Para cada routing instance pueden existir múltiples tablas de ruteo (Juniper Networks, 2013b).

Figura 11. Esquema simplificado de una VPN.



Los routers situados en el borde la red del proveedor se les denomina routers PE (siglas del inglés *Provider Edge*). Estos se interconectan por un lado a los routers del lado del cliente, denominados routers CE (*Customer Edge*) y del otro lado, hacia los routers del núcleo del proveedor, denominados routers P. Los routers PE mantienen una VRF por cada sitio al que se interconectan, esto permite la utilización de redes IP privadas sin que existan colisiones con las redes privadas utilizadas por otros clientes. Los routers PE y CE trabajan en conjunto para el intercambio de información de rutas entre el sitio del cliente y el núcleo de la red del proveedor, para ello se puede emplear cualquier protocolo de ruteo. Las rutas aprendidas de los CE, se almacenan en las tablas de rutas de la VRF del router PE y son enviadas a los routers PE remotos

utilizando MP-BGP (*Multiprotocol Border Gateway Protocol*). Para el reenvío de tráfico de la VPN entre los sitios del cliente, los routers PE utilizan LSPs configurados en la red del proveedor (Juniper Networks, 2013b).

Típicamente la red del proveedor está configurada para manejar más de una VPN, por ello los routers PE pueden tener varias instancias VRF configuradas. Para reconocer las VFRs se utiliza un Distinguidor de Ruta o *Route Distinguisher*, que es un valor numérico único de 6 bytes. Cada VRF configurada en un router PE debe tener un route distinguisher, pudiendo ser aplicado en uno de los siguientes formatos (Juniper Networks, 2013b):

- *<Número de AS>:<identificador>*. Se recomienda utilizar el número de AS del ISP o el del Cliente. El identificador es un valor numérico de 2 o 4 bytes dependiendo de los bytes utilizados para el Número de AS.
- *<Dirección IP>:<identificador>*. La dirección IP corresponde a una dirección IPv4 dentro del rango de IP's asignado, aunque puede ser cualquier dirección unicast global única. El identificador es un valor numérico de 2 bytes.

Además del *route distinguisher*, cada VRF contiene en su configuración otro valor numérico llamado ruta de destino o *route target*, el cual define a qué VPN pertenece la VRF. El route target es único entre los diferentes servicios de VPN, siendo éste también un identificador para las distintas VPN's. Cada VPN cuenta con una política que define cómo se importan las rutas obtenidas de los routers CE. El router PE entonces exporta las rutas en sesiones IBGP a los otros routers del proveedor. La exportación de rutas se rige por cualquier política de ruteo aplicada a cada tabla de rutas de la VRF en particular. Durante la propagación de rutas dentro de la red del proveedor, el router PE convierte la tabla de rutas al formato de VPN-IPv4, agregando el route distinguisher de la VRF a cada ruta conocida (Juniper Networks, 2013b).

E. PROTOCOLO SIMPLE DE ADMINISTRACION DE RED

El Protocolo Simple de Administración de Red o SNMP (*Simple Network Management Protocol*) fue introducido en 1988, con la finalidad de permitir el manejo de diversos dispositivos en una red, además de asegurarse de que éstos se encuentren funcionando de forma óptima (Mauro, 2005).

El núcleo de SNMP es un conjunto de operaciones que permiten a los administradores la posibilidad de leer y cambiar el estado de diversos elementos de dispositivos basados en SNMP. Por ejemplo, se puede utilizar SNMP para apagar una interfaz de un router o verificar la velocidad a la cual una interfaz Ethernet se encuentra operando. El alcance de SNMP es bastante amplio, incluso permite el monitoreo de la temperatura de un equipo, el funcionamiento de ventiladores, etc (Mauro, 2005).

Es importante resaltar que SNMP no sólo se enfoca en el monitoreo de routers, pues este puede ser usado para administrar diferentes tipos de dispositivos, como sistemas sistemas UNIX, sistemas Windows, impresoras, fuentes de poder y cualquier dispositivo que pueda ejecutar software que permita la obtención de información SNMP (Mauro, 2005).

La definición de SNMP, como muchos otros protocolos, se encuentran bajo la responsabilidad de la *Internet Engineering Task Force* (IETF, por sus siglas), entidad que publica las especificaciones en documentos denominados RFC (*Requests for Comments*). Para SNMP existen tres versiones:

- SNMP versión 1 (SNMPv1) es la versión inicial de SNMP. La seguridad de SNMPv1 se basa en comunidades, que funcionan como un tipo de contraseña que permiten el acceso a la administración de la información de un dispositivo. Estas comunidades a su vez pueden ser de tres tipos: sólo lectura, lectura y escritura, y trap. Es importante mencionar que aunque esta es la primera versión de SNMP, muchos fabricantes la continúan implementando en sus dispositivos (Stallings, 1996).
- SNMP versión 2 (SNMPv2) es una evolución de SNMPv1 y en su definición se especifica el reemplazo para los traps. También incluye dos nuevas funciones: *GetBulk*, que permite obtener eficientemente grandes bloques de datos, e *Inform*, que permite a una Estación de Administración de Red o NMS (*Network Management System*) enviar traps de información a otra NMS y luego recibir respuesta (Stallings, 1996).
- SNMP versión 3 (SNMPv3) es la última versión de SNMP. El principal cambio en esta versión es la adición de seguridad criptográfica, que permite un acceso seguro a los routers mediante la autenticación y cifrado de paquetes (https://yordyalberto.wordpress.com/2013/12/04/protocolos-snmp-v1-v2-y-v3/). Aunque SNMPv3 luce bastante diferente de sus versiones anteriores, en realidad se debe a cambios en convenciones textuales, conceptos y terminología. Particularmente, los cambios en la terminología son tan radicales que es difícil creer que éstos describen el mismo software de las versiones anteriores (Mauro, 2005).

Los dispositivos con capacidad de ejecutar software SNMP son generalmente llamados agentes. De forma general, un NMS es el responsable de realizar las llamadas, consultas o polling a los agentes en la red, para conocer el estado de los elementos del Agente, también llamados objetos administrados o simplemente objetos. La información obtenida de la respuesta del agente es luego procesada para determinar si se tiene un evento que pueda afectar la operación normal de los servicios. En la Figura 12 se representa esta interacción entre NMS y Agentes (Stallings, 1996).

De forma alternativa, un agente puede enviar traps a un NMS cuando se detectan eventos importantes, por ejemplo que se cambia el estado lógico de una interfaz de red de up a down. Esta comunicación es asíncrona, es decir, que no ocurre en respuesta a una solicitud por parte del NMS, como se muestra en la

Figura 13. Cuando el NMS recibe un trap, debe de ejecutar una acción basado en la información contenida en éste. Los eventos que serán reportados por medio de un trap y el NMS al cual se informará son configurados en el agente; esto permite que se pueda reportar diferentes eventos a diferentes NMSs en la red (Mauro, 2005).

Figura 12. Eventos durante el polling de un NMS a un Agente

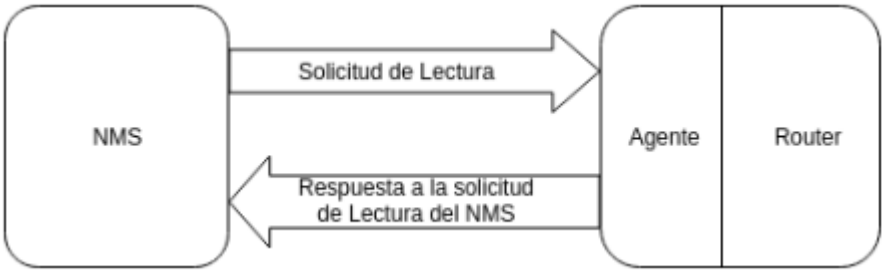
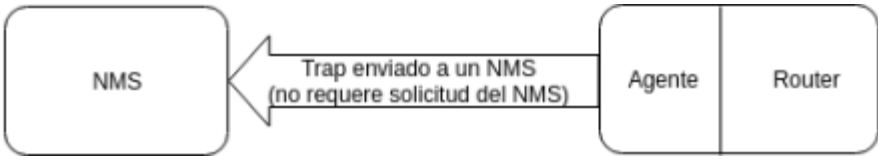


Figura 13. Envío asíncrono de un Trap de un Agente a un NMS



SNMP funciona sobre UDP, aunque también se pueden utilizar TCP, utilizando los puertos 161 para polling y 162 para traps. Para entender mejor el proceso de comunicación entre un NMS y un agente, es necesario introducir algunos conceptos referentes a la administración de la información (Mauro, 2005).

1. Estructura de la Información de Administración. La Estructura de la Información de Administración o SMI (*Structure of Management Information*) define la estructura de los objetos, es decir, la forma en que los objetos son nombrados y el tipo de dato asociado. De aquí deriva otro concepto muy importante: la Base de Información de Administración o MIB (*Management Information Base*), que es la definición de los objetos administrados por un agente haciendo uso de la sintaxis de SMI (Mauro, 2005).

Los objetos son nombrados con un *Object Identifier* (OID), el cual es único para cada objeto. Estos nombres suelen aparecer en dos formas: numérica y legible (para humanos); aunque en ambos casos suelen tenerse nombres bastante extensos. La forma de organizar los objetos administrados es dentro de una jerarquía de árbol, como la que se observa en la Figura 14. Esta estructura jerárquica es la base para nombrar a los objetos (Mauro, 2005).

Figura 14. Jerarquía de árbol de SMI

Una OID se forma de una serie de valores enteros, separados por puntos. Los valores se obtienen de los valores asignados a cada nodo al descender por el árbol jerárquico desde el nodo raíz o inicial. Por ejemplo, en el árbol de la Figura 14, para llegar al nodo de cisco, la OID sería “1.3.6.1.4.1.9” (el primer valor, “1”, corresponde a iso, “3” a org, y así sucesivamente). La forma legible se construye de forma similar, pero utilizando los nombres de los nodos en lugar de los valores enteros de cada nodo (Mauro, 2005).

De forma general, los objetos a monitorear se encuentran contenidos en la rama (o sub-árbol) de internet (OID 1.3.6.1). Para el presente trabajo, son de interés particular los nodos mgmt o management, el cual define un conjunto de elementos estándar de Internet, y el nodo private, definido unilateralmente, es decir, que está a cargo de entidades individuales u organizacionales, quienes se encargan de definir los objetos contenidos. Dentro de la rama privado existe un nodo asignado a cada entidad, los cuales son representados por un número. Por ejemplo, Cisco es representado con el valor 9 mientras que Juniper, con el 2636. Estos valores asignados a las entidades privadas son regulados por la IANA (Mauro, 2005).

De los otros nodos contenidos dentro de la rama de internet, directory no es usado, experimental se encuentra reservado para investigación, security se enfoca a objetos relacionados a seguridad y SNMPv2 contiene objetos agregados en la definición de SMIv2, detallado a continuación. El tipo de dato de un objeto es definido como un subconjunto de instrucciones de ASN.1 (*Abstract Syntax Notation One*). ASN.1 permite representar los datos transmitidos, dentro del contexto de SNMP, y la forma de codificación y decodificación entre un NMS y un agente independientemente de la máquina que se esté utilizando. De esta forma, es posible tener comunicación entre un equipo con un sistema operativo Linux (NMS) y un router

Cisco o Juniper (agente), sin preocuparse de problemas de compatibilidad (Stallings, 1996).

Existen diferentes tipos definidos dentro del SMIv1, versión 1 de SMI, que se utilizan para conocer el tipo de información que puede contener un objeto, listados en la Tabla 4. En la versión 2 de SMI, SMIv2, se agregan otros tipos de datos adicionales, también listados en la Tabla 4, y se amplía la rama de Internet dentro del árbol jerárquico al agregar el nodo snmpV2 (Mauro, 2005).

Tabla 4. Tipos de datos definidos en SMI

Tipo de dato	Descripción	Versión
INTEGER	Es un valor de 32 bits usado para especificar valores enumerados.	SMIv1
OCTET STRING	Es una cadena de octetos o bytes, usado para representar textos y direcciones físicas	SMIv1
Counter	Es un valor de 32 bits sin signo. Este tipo de valores se reinician a cero luego de alcanzar su valor máximo (2^32-1), y son usados generalmente para valores como la cantidad de octetos que pasan por una interface. Estos valores son usados únicamente de forma incremental.	SMIv1
OBJECT IDENTIFIER	Es una cadena de valores enteros separados por números para representar un objeto.	SMIv1
NULL	Valor nulo o vacío.	SMIv1
SEQUENCE	Es una lista que contiene cero o más tipos de datos ASN.1.	SMIv1
SEQUENCE OF	Son aquellos tipos de datos compuestos por un SEQUENCE de ASN.1.	SMIv1
IpAddress	Representa una dirección IPv4 (de 32 bits)	SMIv1
Network Address	Es similar a IpAddress, pero puede representar otro tipo de direcciones.	SMIv1
Gauge	Es un valor similar al contador, con la diferencia de que este puede decrementar y no es cíclico, es decir, que no puede exceder sus límites inferior (cero) y superior (2^32-1).	SMIv1
TimeTicks	Es un valor de 32 bits usado para medir tiempo en centésimas de segundo.	SMIv1
Opaque	Permite a cualquier otro valor ASN.1 ser tratado como un OCTET STRING	SMIv1
Integer32	Mismo que INTEGER	SMIv2
Counter32	Mismo que Counter	SMIv2
Gauge32	Mismo que Gauge	SMIv2
Unsined32	Representa valores de 0 a 2^32-1	SMIv2
Counter64	Es igual a Counter32, con la diferencia que es un valor de 64 bits. Es ideal para aquellos casos en los que Counter32 se reinicia en una corta cantidad de tiempo	SMIv2
BITS	Es una enumeración de bits nombrados no-negativos	SMIv2

2. Operaciones SNMP. Las operaciones entre un agente y un NMS se realizan utilizando mensajes en el formato definido por la Unidad de Datos de Protocolo o PDU (Protocol Data Unit). Las operaciones disponibles con SNMP son las siguientes:

- Get: es una acción iniciada por el NMS donde se envía una solicitud al agente, quien la recibe, procesa y envía una respuesta. En la solicitud de envío se incluye la comunidad SNMP del agente y se indica el OID del objeto que se quiere conocer.
- Getnext: esta operación permite obtener un grupo de valores de una MIB, recorriendo el árbol jerárquico en orden lexicográfico, es decir, como se haría en un diccionario. Gracias a que una OID es una secuencia de valores enteros, para el agente es fácil hacer un recorrido desde el nodo raíz en el árbol jerárquico hasta encontrar la OID deseada. Las solicitudes de getnext serán respondidas hasta recibir un error por parte del agente, indicando que se ha recorrido por completo la MIB definida.
- Getbulk: es una operación implementada en SNMPv2 que permite a un NMS obtener una sección extensa de una tabla en una sola solicitud. En la operación get, se intenta devolver varios valores solicitados en una misma solicitud, pero será limitado por el tamaño de los mensajes, devolviendo un mensaje de error cuando no puede devolver todos los valores solicitados. La operación bulk, le indica al agente enviar de vuelta tantos valores como le sea posible, pudiendo tener respuestas incompletas. En esta operación es importante agregar dos parámetros: nonrepeaters, que indica que los primeros n objetos pueden ser obtenidos con una operación getnext, y max-repetitions, que indica la cantidad máxima cantidad de operaciones getnext para obtener los objetos restantes.
- set: este comando es utilizado para cambiar el valor de un objeto o para crear nuevos valores en una tabla. Esto es aplicable para los objetos definidos como lectura y escritura dentro de la MIB.
- Getresponse: comprende es un conjunto de respuestas son utilizadas para determinar si alguna solicitud de get o set fue procesada de forma correcta por un agente. Los mensajes de error definidos en SNMPv1 no son muy robustos, por lo que en SNMPv2 se agregaron respuestas de error adicionales, para lo cual tanto el NMS como el agente deben soportar SNMPv2.
- Trap: como se menciona anteriormente, esta operación es iniciada por el agente. Se utiliza para notificarle a un NMS que se detectó un evento en el dispositivo asociado al agente como la caída de una interfaz de red o el fallo de un ventilador.
- Notification: esta operación fue introducida en SNMPv2 con la finalidad de estandarizar el formato de PDU para los traps de SNMPv1, en donde los mensajes son diferentes a los utilizados por las operaciones get y set. En el formato establecido de SNMPv2, los mensajes son idénticos a los de operaciones como get y set.
- Inform: SNMPv2 provee un mecanismo para informar, que le permite enviar mensajes de acknowledge o recibido a los mensajes de trap, acción que no se realiza en SNMPv1 y que permitía la pérdida de éstos mensajes.

- Report: Esta operación fue definida en SNMPv2 e implementada en SNMPv3. Su finalidad es permitir a los motores SNMP a comunicarse con otros, principalmente para reportar problemas de procesamiento de mensajes SNMP (Mauro, 2005).

3. SNMP y Python.

Python es un lenguaje de programación de alto nivel, el cual es interpretado. Soporta diferentes paradigmas de programación como la Programación Estructurada, la Programación Orientada a Objetos y la Programación Funcional. Python viene preinstalado en los sistemas operativos MacOS y Linux recientes, pero es recomendable verificar que se tenga una versión adecuada. En el caso de Windows será necesario bajar el instalador y seguir los pasos del asistente de instalación. De las principales fortalezas de Python, podemos citar la gran cantidad de librerías existentes, la simplicidad del código escrito en este lenguaje y la portabilidad del mismo, es decir, que puede ejecutarse en distintos Sistemas Operativos.

Python cuenta con dos versiones activamente utilizadas, Python2.7 y Python3 (su última versión es la 3.5). La versión que se encuentra actualmente en desarrollo activo es Python3, y aunque inicialmente no guardaba compatiblidad con sus versiones anteriores (2.x), existen esfuerzos en los últimos desarrollos para guardar compatibilidad con sus versiones anteriores. Para este trabajo de graduación se trabaja con Python versión 2.7.10.

Dentro de las librerías de Python se encuentra PySNMP, que como su nombre sugiere, es una librería de Python para trabajar con SNMP. Esta librería tiene la capacidad de trabajar con las tres versiones de SNMP tanto sobre redes IPv4 como IPv6. Esta librería funciona en Python2.7 y Python3. Su instalación resulta bastante simple al emplear la utilidad PIP, que es la forma de acceder al repositorio de Python llamado PyPI (acrónimo del nombre en inglés, *Python Package Index*) al cual almacena más de 80,000 librerías de Python, como se muestra en la Figura 15. PIP se instala automáticamente versiones de Python 2.7.9 o superior y Python 3.4 o superior (Etingof, 2016).

Como se observa en la Figura 15, la utilidad pip recibe únicamente los argumentos install y el nombre del paquete a instalar. Durante la ejecución, PIP verifica las dependencias de la librería a instalar, y de existir dependencias automáticamente verifica si se encuentran instaladas y si satisfacen la versión requerida, en caso contrario instala o actualiza estas versiones. Para PySNMP, sus cuatro dependencias son PyCrypto, PySMI, PyASN1 y PLY. En el ejemplo de instalación, el primer paquete (PyCrypto) ya se encuentra instalado en una versión que satisface la dependencia de PySNMP, los restantes tres no se encontraban instalados por lo que se procede a su instalación al igual que la del mismo pysnmp.

Figura 15. Instalación de una librería utilizando PIP

```
~$ sudo pip install pysnmp
Collecting pysnmp
  Downloading pysnmp-4.3.2-py2.py3-none-any.whl (254kB)
    100% |████████████████████████████████| 256kB 589kB/s
Requirement already satisfied (use --upgrade to upgrade): pycrypto>=2.4.1 in
/usr/lib/python2.7/dist-packages (from pysnmp)
Collecting pysmi (from pysnmp)
  Downloading pysmi-0.0.7-py2.py3-none-any.whl (62kB)
    100% |████████████████████████████████| 71kB 2.9MB/s
Collecting pyasn1>=0.1.8 (from pysnmp)
  Downloading pyasn1-0.1.9-py2.py3-none-any.whl
Collecting ply (from pysmi->pysnmp)
  Downloading ply-3.9.tar.gz (150kB)
    100% |████████████████████████████████| 153kB 351kB/s
Building wheels for collected packages: ply
  Running setup.py bdist_wheel for ply ... done
  Stored in directory:
/home/doanvelagui/.cache/pip/wheels/c1/0c/bd/306a63396decbe8353a4a056fcba97a092be0e03552
2bc567d
Successfully built ply
Installing collected packages: ply, pysmi, pyasn1, pysnmp
Successfully installed ply-3.9 pyasn1-0.1.9 pysmi-0.0.7 pysnmp-4.3.2
```

PySNMP permite realizar las operaciones de SNMP mencionadas anteriormente, permitiendo obtener los mismos resultados que se obtienen utilizando las utilidades SNMP que se incluyen en la mayoría de los Sistemas Operativos recientes. A continuación se encuentran los resultados de las funciones get (Figura 16) y getnext (Figura 17) utilizando PySNMP y las utilidades SNMP de un Sistema Operativo Linux en forma comparativa (Etingof, 2016).

Figura 16. Resultados de la operación get de SNMP con un script de Python y las utilidades de SNMP de Linux

```
~$ python
>>> from pysnmp.hlapi import *
>>> # ejecutar operación get de SNMP
>>> errorIndication, errorStatus, errorIndex, varBinds = next(
...     getCmd(
...         SnmpEngine(),
...         CommunityData('[redacted]', mpModel=0),
...         UdpTransportTarget(('[redacted]', 161)),
...         ContextData(),
...         ObjectType(ObjectIdentity('1.3.6.1.2.1.1.5.0'))
...     )
... )
...
>>> # presentación de resultados, verificando si se produjo algún error
>>> if errorIndication:
...     print(errorIndication)
... elif errorStatus:
... print('%s at %s' % (errorStatus.prettyPrint(),
...                     errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
... else:
...     for varBind in varBinds:
...         print(' = '.join([x.prettyPrint() for x in varBind]))
...
SNMPv2-MIB::sysName.0 = ON-NAP-01-MX960-RE0-01
>>> exit()
~$ snmpget -v 2c -c [redacted] [redacted] 1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: ON-NAP-01-MX960-RE0-01
```

La función getCmd de la librería PySNMP realiza la operación get de SNMP. Para su ejecución requiere de cinco parámetros:

- snmpEngine: es una clase que representa un motor SNMP.
- authData: es una instancia de las clases *CommunityData* o *UsmUserData*, que contiene las credenciales de acceso, en el ejemplo presentado, el nombre de la comunidad SNMP.
- transportTarget: puede ser una instancia de las clases *UdpTransportTarget* o *Udp6TransportTarget*, conteniendo la dirección IPv4 o IPv6 respectivamente y el puerto UDP a utilizar.
- contextData: es una clase utilizada para representar valores SNMP *ContextEngineID*, es un identificador único para identificar una instancia SNMP, y *ContextName*, nombre de una instancia de MIB que por defecto es una cadena de caracteres vacía.
- *varBinds: es una o más instancias de la clase *ObjectType* representando las variables MIB a enviar en la solicitud de SNMP (Etingof, 2016).

Figura 17. Resultados de la operación getnext de SNMP con un script de Python y las utilidades de SNMP de Linux

```
~$ python
>>> from pysnmp.hlapi import *
>>> # Ejecutar operación getnext
>>> results = nextCmd(SnmpEngine(),
...                   CommunityData('[redacted]', mpModel=0),
...                   UdpTransportTarget(('[redacted]', 161)),
...                   ContextData(),
...                   ObjectType(ObjectIdentity('1.3.6.1.2.1.1')) )
...
>>> # Presentación de resultados
>>> for (errorIndication, errorStatus, errorIndex, varBinds) in results:
...     if errorIndication:
...         print(errorIndication)
...         break
...     elif errorStatus:
...         print('%s at %s' % (errorStatus.prettyPrint(),
...                              errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
...         break
...     else:
...         for varBind in varBinds:
...             print(' = '.join([x.prettyPrint() for x in varBind]))
...
SNMPv2-MIB::sysDescr.0 = Monitoreo
SNMPv2-MIB::sysObjectID.0 = SNMPv2-SMI::enterprises.2636.1.1.1.2.21
SNMPv2-MIB::sysUpTime.0 = 340641565
SNMPv2-MIB::sysContact.0 = Tigo One Network
SNMPv2-MIB::sysName.0 = ON-NAP-01-MX960-RE0-01
SNMPv2-MIB::sysLocation.0 =
SNMPv2-MIB::sysServices.0 = 6
>>> exit()
~$ snmpwalk -v 2c -c [redacted] [redacted] 1.3.6.1.2.1.1
SNMPv2-MIB::sysDescr.0 = STRING: Monitoreo
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.2636.1.1.1.2.21
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (340497079) 39 days, 9:49:30.79
SNMPv2-MIB::sysContact.0 = STRING: Tigo One Network
SNMPv2-MIB::sysName.0 = STRING: ON-NAP-01-MX960-RE0-01
SNMPv2-MIB::sysLocation.0 = STRING:
SNMPv2-MIB::sysServices.0 = INTEGER: 6
```

La función *nextCmd* es el método utilizado para la obtención del resultado de la operación *getnext* de SNMP. Los parámetros que recibe esta función son los mismos que los recibidos por *getCmd*. Cabe mencionar que la operación *getnext* de SNMP también se conoce como *walk*, en referencia a la expresión *walking a MIB*, que se refiere al recorrido del árbol jerárquico de una MIB. En el ejemplo mostrado en la Figura 17, se recorre el árbol *system* con OID 1.3.6.1.2.1.1 (iso.org.dod.internet.mgmt.mib-2.system), cuyos nodos hijos se listan en la Tabla 5 (Etingof, 2016).

Tabla 5. Subnodos del árbol jerárquico system

Nombre del objeto	OID
system	1.3.6.1.2.1.1
sysDescr	1.3.6.1.2.1.1.1
sysObjectID	1.3.6.1.2.1.1.2
sysUpTime	1.3.6.1.2.1.1.3
sysContact	1.3.6.1.2.1.1.4
sysName	1.3.6.1.2.1.1.5
sysLocation	1.3.6.1.2.1.1.6
sysServices	1.3.6.1.2.1.1.7
sysORLastChange	1.3.6.1.2.1.1.8
sysORTable	1.3.6.1.2.1.1.9

La operación *getBulk* de SNMP se ejecuta con la función *bulkCmd* de PySNMP, como se muestra en la Figura 18. Esta función recibe parámetros adicionales a los de las funciones presentadas anteriormente.

- nonRepeaters: es un valor de tipo entero utilizado para representar el valor del parámetros non-repeaters requerido por la operación getBulk.
- maxRepetitions: es un valor de tipo entero utilizado para representar el valor del parámetros max-repetitions requerido por la operación getBulk (Etingof, 2016).

Figura 18. Ejecución de la operación getbulk de SNMP con PySNMP.

```
$~ python
>>> from pysnmp.hlapi import *
>>> # Ejecutar operación getnext
>>> results = bulkCmd(SnmpEngine(),
...            CommunityData('[redacted]'),
...            UdpTransportTarget(('[redacted]', 161)),
...            ContextData(),
...            0, 25, #nonRepeaters y maxRepetitions
...            ObjectType(ObjectIdentity('SNMPv2-MIB', 'sysDescr')))
>>> # Presentación de resultados
>>> for (errorIndication, errorStatus, errorIndex, varBinds) in results:
...     if errorIndication:
...         print(errorIndication)
...         break
...     elif errorStatus:
...         print('%s at %s' % (errorStatus.prettyPrint(),
...                              errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
...         break
...     else:
...         for varBind in varBinds:
...             print(' = '.join([x.prettyPrint() for x in varBind]))
...
SNMPv2-MIB::sysDescr.0 = Monitoreo
SNMPv2-MIB::sysObjectID.0 = SNMPv2-SMI::enterprises.2636.1.1.1.2.21
SNMPv2-MIB::sysUpTime.0 = 341544825
SNMPv2-MIB::sysContact.0 = Tigo One Network
SNMPv2-MIB::sysName.0 = ON-NAP-01-MX960-RE0-01
SNMPv2-MIB::sysLocation.0 =
SNMPv2-MIB::sysServices.0 = 6
```

F. PROTOCOLO DE MENSAJES DE CONTROL DE INTERNET

El Protocolo de Mensajes de Control de Internet (ICMP, por sus siglas en inglés *Internet Control Message Protocol*) trabaja en la Capa de Red, o capa 3, del modelo OSI y es utilizada por el protocolo de Internet (IP) para diferentes servicios. Los paquetes ICMP tienen la capacidad de brindar información al equipo acerca de problemas con la red y pueden ser encapsulados en paquetes IP para su transporte (Lammle, 2013).

Algunos de los eventos y mensajes más comunes relacionados a ICMP son los siguientes:

- Destination unreachable: Si un router es incapaz de enviar un paquete IP a su destinatario, o no encuentra un intermediario al que pueda ser enviado, éste envía un mensaje al emisor informando sobre la situación (Lammle, 2013).
- Buffer full/source quench: si la memoria de un router se encuentra saturada y no puede recibir más paquetes, se enviará un mensaje de alerta hasta que la congestión disminuya (Lammle, 2013).
- Hops/Time exceeded: cada paquete IP tiene asignado un número de routers, llamados saltos, hops o TTL (Time to Live), a través de los cuales puede pasar. Si éste número de saltos es alcanzado antes de llegar al destinatario, el último router elimina el paquete IP y envía un mensaje ICMP indicándole al emisor la acción tomada con el paquete (Lammle, 2013).

- Echo Request y Echo Reply: son mensajes utilizados para comprobar que un host destino sea alcanzable y se encue en correcto funcionamiento. Su funcionamiento es en pares, pues el emisor envía un mensaje echo request al host destino y por su parte, el host destino al recibir el echo request le responde al emisor con un mensaje echo reply (en algunos casos llamado únicamente reply) (Tanenbaum, 2012).

Para el diagnóstico del estado de un host, pueden ser utilizadas las utilidades *Ping* y *Traceroute*, que normalmente vienen incluidas como parte de los sistemas operativos actuales y son ejecutables desde una consola o terminal. Estas utilidades utilizan paquetes ICMP para luego presentarle al usuario la información resultante del estado de la red.

Ping (*Packet Internet Groper*) es comúnmente llamado así por su relación etimológica al sonido producido por un sonar. Ping utiliza mensajes ICMP de solicitud de respuesta (*echo request*) y de respuesta (*reply o echo reply*) para verificar la conectividad física o lógica de dos equipos en una red (Tanenbaum, 2012). En la Figura 19 se observa un ejemplo de la utilidad ping.

Figura 19. Ejemplo de pruebas de ping

```
~$ ping -c 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_req=1 ttl=55 time=27.0 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=55 time=26.8 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=55 time=27.1 ms
64 bytes from 8.8.8.8: icmp_req=4 ttl=55 time=26.8 ms
64 bytes from 8.8.8.8: icmp_req=5 ttl=55 time=28.7 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 26.802/27.333/28.711/0.723 ms
```

Es importante resaltar que al final de la prueba de ping se presentan las estadísticas de la misma. Estas pruebas contienen información relevante como el porcentaje de paquetes perdidos (packet loss) que indica cuántos mensajes echo request no alcanzaron al destinatario y, consecuentemente, no se recibió el respectivo mensaje reply. Otra estadística importante es el Tiempo de Viaje o rtt (siglas del inglés Round Trip Time), que indican el promedio del tiempo total desde que el mensaje echo request fue enviado hasta que se recibió el mensaje reply.

La utilidad de traceroute o traza es utilizada para descubrir el listado de enrutadores intermedios desde un emisor para alcanzar a un destinatario a través de una red. Para identificar a los enrutadores intermedios, traceroute hace un uso inteligente de los mensajes *Hops/Time Exceeded* al enviar paquetes IP con valores TTL desde 1 hasta el valor especificado en la prueba (normalmente la utilidad tiene un valor por defecto de 30) o alcanzar al destinatario. Con lo anterior, se asegura que los contadores llegarán a un valor de cero (cada enrutador intermedio decrementa el valor en una unidad) en los enrutadores intermedios sucesivos,

obteniendo sus direcciones IP de los mensajes *Hops/Time Exceeded* recibidos (Tanenbaum, 2012). En la Figura 20 se presenta un ejemplo de esta utilidad.

Figura 20. Ejemplo de pruebas de traza.

```
~$ traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.43.1 (192.168.43.1)  3.909 ms  4.155 ms  4.237 ms
 2  * * *
 3  10.189.130.6 (10.189.130.6)  133.853 ms  133.928 ms  134.018 ms
 4  10.35.32.18 (10.35.32.18)  133.698 ms  134.431 ms  134.533 ms
 5  200.49.165.65 (200.49.165.65)  134.122 ms  134.264 ms  134.563 ms
 6  200.49.165.65 (200.49.161.101)  133.441 ms  108.364 ms  108.648 ms
 7  190.106.193.85 (190.106.193.85)  109.320 ms  24.757 ms  24.684 ms
 8  190.106.192.36 (190.106.192.36)  60.853 ms  65.952 ms  74.714 ms
 9  72.14.217.40 (72.14.217.40)  60.737 ms  60.721 ms  65.944 ms
10  216.239.50.57 (216.239.50.57)  74.517 ms  74.414 ms  74.311 ms
11  216.239.51.141 (216.239.51.141)  74.685 ms  82.014 ms  50.591 ms
12  google-public-dns-a.google.com (8.8.8.8)  72.394 ms  49.099 ms  49.187 ms
```

Existen casos en que un enrutador intermedio (u otro dispositivo como un *firewall*) se encuentra configurado para no responder mensajes ICMP, por lo que se presenta al usuario tres asteriscos (* * *) o el texto “no reply”, al no recibir el mensaje Hops/Time Exceeded esperado de este enrutador intermedio.

1. Monitoreo del Rendimiento en Tiempo Real en Dispositivos Juniper. El Monitoreo del Rendimiento en Tiempo Real o RPM (*Real-time Performance Monitoring*) forma parte de las herramientas de monitoreo disponibles en el Sistema Operativo de los Dispositivos Juniper, llamado Junos OS (*Operating System*). Con esta herramienta es posible monitorear el rendimiento de la red, además de evaluar y analizar la eficiencia de la red. El rendimiento es evaluado en tiempo real basado en métricas como RTT, Jitter y pérdida de paquetes obtenidos en los resultados de una prueba o *test* de RPM. Para obtener estos resultados, RPM realiza un intercambio de pruebas con otros dispositivos IP para fines de seguimiento y vigilancia de la red (Juniper Networks, 2010a).

Las métricas obtenidas de las pruebas intercambiadas brindan una aproximación del retardo y del jitter experimentado por la red en tiempo real. Estas pruebas también se pueden utilizar para verificar el estado de los equipos a los cuales las pruebas son enviadas. Las pruebas o *test* de RPM consisten en realizar un número determinado de pruebas de ping, las cuales utiliza para obtener estadísticas. RPM permite obtener los resultados del último test completado, el que está en curso y el global de todas las pruebas realizadas, como se muestra en la Figura 21. Un grupo homogéneo de tests se denomina probe. Las pruebas de un test utilizada paquetes ICMP por defecto, similar a los utilizados por un ping, aunque se puede configurar el uso de otros tipos de paquetes como HTTP GET, UDP echo, TCP connection, entre otros (Juniper Networks, 2010a).

Figura 21. Salida del resultado del test de un probe.

```
user@host> show services rpm probe-results owner probe-name test test-name
    Owner: probe-name, Test: probe-test
    Target address: 192.168.1.8, Source address: 192.168.1.245,
    Probe type: icmp-ping
    Routing Instance Name: vrf-name
    Test size: 10 probes
    Probe results:
      Response received, Mon Sep 12 22:11:16 2016, No hardware timestamps
      Rtt: 33330 usec, Round trip jitter: -59 usec,
      Round trip interarrival jitter: 823 usec
    Results over current test:
      Probes sent: 4, Probes received: 4, Loss percentage: 0
      Measurement: Round trip time
        Samples: 4, Minimum: 33330 usec, Maximum: 33510 usec,
        Average: 33393 usec, Peak to peak: 180 usec, Stddev: 71 usec,
        Sum: 133570 usec
    Results over last test:
      Probes sent: 10, Probes received: 10, Loss percentage: 0
      Test completed on Mon Sep 12 22:10:43 2016
      Measurement: Round trip time
        Samples: 10, Minimum: 33261 usec, Maximum: 37725 usec,
        Average: 34032 usec, Peak to peak: 4464 usec, Stddev: 1331 usec,
        Sum: 340323 usec
    Results over all tests:
      Probes sent: 1011714, Probes received: 1011606, Loss percentage: 0
      Measurement: Round trip time
        Samples: 1011606, Minimum: 26060 usec, Maximum: 267821 usec,
        Average: 34844 usec, Peak to peak: 241761 usec, Stddev: 8063 usec,
        Sum: 35248678397 usec
```

En el ejemplo anterior, se presenta que el test actual aún no se ha completado, pues solamente se han enviado 4 pruebas, mientras que el tamaño del test es de 10 pruebas. Junos OS almacena las estadísticas de los resultados de RPM en MIBs que pueden ser recuperados por medio de SNMP, la tabla de MIBs utilizada se encuentra en la Tabla 6. Dentro de la configuración de cada test se puede especificar la dirección destino, el tipo de prueba, la cantidad de paquetes, el intervalo entre envío de paquetes y pruebas, entre otros; como se muestra en la Figura 22.

Tabla 6. Organización de MIBs usadas para resúmenes de Ping y RPM

Nombre del objeto	OID	Descripción
pingMIB	1.3.6.1.2.1.80	
pingObjects	1.3.6.1.2.1.80.1	
pingCtlTable	1.3.6.1.2.1.80.1.2	Tabla utilizada para crear un probe básico de un RPM.
pingResultsTable	1.3.6.1.2.1.80.1.3	Tabla utilizada para almacenar los resultados de las pruebas de ping del último test.
pingProbeHistoryTable	1.3.6.1.2.1.80.1.4	Tabla utilizada para almacenar los resultados previos de una operación de ping.

Continuación Tabla 6.

Nombre del objeto	OID	Descripción
jnxMibs	1.3.6.1.4.1.2636.3	
jnxPingMIB	1.3.6.1.4.1.2636.3.7	
jnxPingObjects	1.3.6.1.4.1.2636.3.7.1	
jnxPingCtlTable	1.3.6.1.4.1.2636.3.7.1.2	Tabla utilizada para almacenar pruebas adicionales como HTTP, VRF instance timestamps, y thresholds para probes de RPM
jnxPingResultsTable	1.3.6.1.4.1.2636.3.7.1.3	Aumenta la información de la tabla pingResultsTable con resultados adicionales
jnxpingProbeHistoryTable	1.3.6.1.4.1.2636.3.7.1.4	Aumenta la información de la tabla pingProbeHistoryTable con resultados adicionales
jnxPingLastTestResultTable	1.3.6.1.4.1.2636.3.7.1.5	Contiene los resultados del último test completado.
jnxRpmMibRoot	1.3.6.1.4.1.2636.3.50	
jnxRpmMib	1.3.6.1.4.1.2636.3.50.1	
jnxRpmResultsSampleTable	1.3.6.1.4.1.2636.3.50.1.1	Contiene las métricas de la última prueba ejecutada de cada test.
jnxRpmResultsSummaryTable	1.3.6.1.4.1.2636.3.50.1.2	Contiene el resumen del último test completado.
jnxRpmResultsCalculatedTable	1.3.6.1.4.1.2636.3.50.1.3	Contiene las métricas calculadas del último test completado.
jnxRpmHistorySampleTable	1.3.6.1.4.1.2636.3.50.1.4	Contiene las métricas de cada muestra almacenada de historial de las tablas de probe de RPM.
jnxRpmHistorySummaryTable	1.3.6.1.4.1.2636.3.50.1.5	Similar a jnxRpmResultsSummaryTable, esta tabla contiene un resumen de cada prueba completada.
jnxRpmHistoryCalculatedTable	1.3.6.1.4.1.2636.3.50.1.6	Esta tabla contiene las métricas calculadas para cada entrada de RPM.

Figura 22. Parámetros de configuración de un test de RPM.

```
configuration {
    services {
        rpm {
            probe probe-name {
                test test-name {
                    data-fill data;
                    data-size size;
                    destination-interface interface-name;
                    destination-port port;
                    dscp-code-point dscp-bits;
                    hardware-timestamp;
                    history-size size;
                    moving-average-size number;
                    one-way-hardware-timestamp;
                    probe-count count;
                    probe-interval seconds;
                    probe-type type;
                    routing-instance instance-name;
                    source-address address;
                    target (url url | address address);
                    test-interval interval;
                    thresholds thresholds;
                    traps traps;
                }
            }
        }
    }
}
```

Los objetos descritos en los árboles jerárquicos de MIBs presentados en la Tabla 6, contienen diferentes entradas, que a su vez cuentan con un valor para cada probe y test configurado en un dispositivo Juniper. La construcción de las OID's para los elementos dentro de *jnxRpmMib* resulta diferente al que se suele utilizar dentro de MIB-2, pues la OID está formada por el nombre del probe seguido del nombre del test, utilizando el valor numérico de cada carácter ASCII para representar una entrada numérica de la OID. Por ejemplo en la Figura 23, se muestra el resultado obtenido al hacer un walk al valor de *Packet Loss* desde la Interfaz de Línea de Comandos o CLI (*Command Line Interface*) de Juniper.

Figura 23. Obtención de paquetes enviados y paquetes perdidos por medio de OID desde la CLI de Juniper

```
user@host> show snmp walk get 1.3.6.1.4.1.2636.3.50.1.2.1.2
jnxRpmResSumSent.10.112.114.111.98.101.45.110.97.109.101.9.116.101.115.116.45.110.97.109
.101.1 = 5
jnxRpmResSumSent.10.112.114.111.98.101.45.110.97.109.101.9.116.101.115.116.45.110.97.109
.101.2 = 10
jnxRpmResSumSent.10.112.114.111.98.101.45.110.97.109.101.9.116.101.115.116.45.110.97.109
.101.4 = 16425

user@host> show snmp walk get 1.3.6.1.4.1.2636.3.50.1.2.1.4
jnxRpmResSumPercentLost.10.112.114.111.98.101.45.110.97.109.101.9.116.101.115.116.45.110
.97.109.101.1 = 0
jnxRpmResSumPercentLost.10.112.114.111.98.101.45.110.97.109.101.9.116.101.115.116.45.110
.97.109.101.2 = 0
jnxRpmResSumPercentLost.10.112.114.111.98.101.45.110.97.109.101.9.116.101.115.116.45.110
.97.109.101.4 = 0
```

Las OID's 1.3.6.1.4.1.2636.3.50.1.2.1.4 y 1.3.6.1.4.1.2636.3.50.1.2.1.4 equivalen a *jnxRpmResSumPercentLost* y *jnxRpmResSumPercentLost* en la salidas desplegadas. En ambas salidas se tiene como primer valor numérico de la OID el valor 10, que indica la longitud del nombre del probe formado por los valores *112.114.111.98.101.45.110.97.109.101*, que en este ejemplo equivale a la cadena de caracteres *probe-name*. La siguiente entrada numérica 9 indica la longitud del nombre del test formado por las entradas *116.101.115.116.45.110.97.109.101* que equivalen a *test-name*. Por último, los valores 1, 2 y 4 de la última entrada de la OID corresponden al resultado de la prueba actual, la última prueba completada y el resultado total de todas las pruebas realizadas.

G. PROTOCOLO DE CONFIGURACION DE RED

El protocolo de configuración de Red o NETCONF (acrónimo del nombre en inglés Network Configuration), define un mecanismo simple a través del cual se puede administrar o manipular la configuración de los dispositivos en la red, permitiendo extraer la configuración de un dispositivo o bien enviarle nuevos bloques de configuración (Internet Engineering Task Force, 2011).

NETCONF utiliza el paradigma RPC (*Remote Procedure Call*), entendiendo por RPC como un protocolo que permite a un dispositivo cliente la llamada de servicios, rutinas o funciones contenidos en programas alojados en otro dispositivo, llamado también servidor, sin necesidad de que el cliente conozca el detalle de este servicio. Los mensajes RPC intercambiados entre el cliente y el servidor son una extensión de XDR (acrónimo de *External Data Representation*). Cabe mencionar que el tipo de comunicación de RPC es síncrona (Internet Engineering Task Force, 2011).

Durante una llamada con NETCONF, los mensajes del cliente y del servidor son codificados en XML (acrónimo de *Extended Markup Language*) y se utiliza una conexión segura para el intercambio de mensajes. Uno de los aspectos importantes de NETCONF es que permite que las funcionalidades de administración sean muy similares a las funcionalidades nativas del dispositivo (Internet Engineering Task Force, 2011).

1. NETCONF en Dispositivos Juniper. Al igual que otros Sistemas Operativos de routers, Junos OS tiene capacidad para trabajar con NETCONF, definiendo operaciones básicas equivalentes a los comandos disponibles en la Interfaz de Línea de Comandos (CLI, por sus siglas en inglés *Command Line Interface*). Las aplicaciones que utilizan el protocolo NETCONF tienen la capacidad de desplegar, editar y guardar cambios en las sentencias de configuración equivalente a como lo realizaría un administrador en la línea de comandos (Juniper Networks, 2014a).

Junos provee una API (siglas del nombre en inglés, *Application Program Interface*) que utiliza XML para representar las sentencias de configuración de Junos OS. Dentro de la definición de la API, se incluyen diferentes etiquetas que son utilizadas al codificar la información en formato XML, facilitando el trabajo de parseo de la configuración al solicitarla, como se muestra en la Figura 24. Las llamadas al router también se realizan con formatos XML, los cuales se pueden obtener desde la CLI del router (Juniper Networks, 2014a).

Figura 24. Comparación de los formatos ASCII, XML y RPC de una interface.

```
user@host> show interface fxp0
Physical interface: fxp0, Enabled, Physical link is Up
  Interface index: 4, SNMP ifIndex: 3
  Type: Ethernet, Link-level type: Ethernet, MTU: 1514, Speed: 1000mbps

user@host> show interface fxp0 | display xml
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/13.3R6/junos">
    <interface-information xmlns="http://xml.juniper.net/junos/13.3R6/junos-interface"
junos:style="normal">
        <physical-interface>
            <name>fxp0</name>
            <admin-status junos:format="Enabled">up</admin-status>
            <oper-status>up</oper-status>
            <local-index>4</local-index>
            <snmp-index>3</snmp-index>
            <if-type>Ethernet</if-type>
            <link-level-type>Ethernet</link-level-type>
            <mtu>1514</mtu>
            <speed>1000mbps</speed>
        </physical-interface>
    </interface-information>
</rpc-reply>

user@host> show interface fxp0 | display xml rpc
<rpc-reply xmlns:junos="http://xml.juniper.net/junos/13.3R6/junos">
    <rpc>
        <get-interface-information>
                <interface-name>fxp0</interface-name>
        </get-interface-information>
    </rpc>
</rpc-reply>
```

2. Junos PyEZ. Es un microframework para python que permite administrar y automatizar dispositivos con un sistema operativo Junos (Junos OS) de forma remota. Junos OS provee la capacidad de ejecutar los comandos que se ejecutan en la interface línea de comandos (CLI, siglas del nombre en inglés *Command-Line Interface*) en un ambiente construido para la automatización de tareas y también la ejecución de comandos utilizando la API de Junos basada en XML (Juniper Networks, 2014b).

Junos PyEZ se basa en NETCONF para la extracción de información de un dispositivo y la ejecución de las funciones disponibles a través de la API de Junos XML. Las utilidades que incluye PyEZ se dividen en los siguientes grupos:

- Utilidades de Software: permiten la instalación de software, reinicio y apagado de los dispositivos administrados (Juniper Networks, 2014b).

- Utilidades de administración de configuración: permiten la comparación de archivos de configuración, la creación de archivos de respaldo de la configuración, modificar la configuración y deshacer cambios de configuración de los dispositivos administrados. Para la modificación de la configuración, acepta diversos formatos estándar, incluyendo texto ASCII, elementos en formato XML, comandos set de configuración de Junos OS e incluso plantillas de Jinja 2 para proveer mayor flexibilidad y personalización (Juniper Networks, 2014b).
- Utilidades del sistema: permiten ejecutar tareas administrativas de uso común, como la copia de archivos, cálculos de sumas de comprobación, entre otros (Juniper Networks, 2014b).

En la Figura 25 se muestra el uso de la librería Junos PyEZ para obtener la información de la interfaz *fxp0*, mostrada en el ejemplo de la Figura 24. Se observa la secuencia para establecer una sesión NETCONF sobre un canal seguro. NETCONF facilita la obtención de resultados, evitando tener que construir el XML que requieren las llamadas RPC de forma manual, por medio de métodos asociados a las etiquetas utilizadas comúnmente.

Figura 25. Obtención de la información de la interfaz fxp0 por medio de PyEZ

```
~$ python
Python 2.7.10 (default, Jul 30 2015, 11:06:27)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from getpass import getpass
>>> from jnpr.junos import Device
>>> from jnpr.junos.op.xcvr import XcvrTable
>>> from lxml import etree
>>> user = getpass('username: ')
username:
>>> passwd = getpass('password: ')
password:
>>> device = getpass('device: ')
device:
>>> dev = Device(device, user=user, password=passwd)
>>> dev.open()
Device([redacted])
>>> if_info = dev.rpc.get_interface_information(interface_name='fxp0')
>>> print(etree.tostring(if_info))
<interface-information style="normal">
<physical-interface>
<name>fxp0</name>
<admin-status format="Enabled">up</admin-status>
<oper-status>up</oper-status>
<local-index>4</local-index>
<snmp-index>3</snmp-index>
<if-type>Ethernet</if-type>
<link-level-type>Ethernet</link-level-type>
<mtu>1514</mtu>
<speed>1000mbps</speed>
</physical-interface>
</interface-information>

>>> dev.close()
>>> exit()
```

H. PROTOCOLO DE ACCESO A OBJETOS SIMPLES

El Protocolo de Acceso a Objetos Simples o SOAP (*Simple Object Access Protocol*) es un protocolo utilizado en la Implementación de Servicios Web o *Web Services*. Un web service es aquel servicio que, generalmente, es accesible por medio de HTTP (*Hyper Text Transfer Protocol*) y permite la comunicación entre computadoras (Chase, 2006).

SOAP se basa en el intercambio de mensajes en formato XML, que puede ser usado para la representación de objetos, expandiendo las posibilidades de comunicación entre diferentes aplicaciones. Los mensajes de SOAP se empaquetan en un sobre, como el que se muestra en la Figura 26 (Chase, 2006).

Figura 26. Ejemplo de un mensaje SOAP.

```
<SOAPenv:Envelope
          xmlns:SOAPenv="http://schemas.xmlSOAP.org/SOAP/envelope/"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

    <SOAPenv:Header>
        <wsa:From>
            <wsa:Address>
                http://localhost:8080/axis2/services/MyService
            </wsa:Address>
        </wsa:From>
    </SOAPenv:Header>

    <SOAPenv:Body>
        <req:getNumberOfArticles xmlns:req="http://daily-moon.com/CMS/">
            <req:category>
                Classifieds
            </req:category>
        </req:getNumberOfArticles>
    </SOAPenv:Body>

</SOAPenv:Envelope>
```

La estructura de un mensaje, como el mostrado anteriormente, se componen de tres elementos básicos:

- Sobre: es la unidad básica de un mensaje del web service. Este contiene la información necesaria para procesar el mensaje.
- Encabezado: provee información acerca del mensaje mismo, por ejemplo el remitente, la dirección a la que se debe responder, etc.
- Cuerpo: esta es la parte denominada carga útil o payload, pues contiene información para solicitar al destinatario la ejecución una acción o para compartir información o resultados de una acción ejecutada (Chase, 2006).

El intercambio de mensajes dentro de SOAP puede realizarse siguiendo alguno de los patrones Request/Response o One-way messaging. El primero, Request/Response, funciona mediante el envío de un mensaje desde un emisor a un receptor, y se espera una respuesta por parte del receptor. Esta comunicación puede ser tanto síncrona como asíncrona. Por su parte, el patrón One-way messaging, también conocido como enviar y olvidar, únicamente involucra el envío de un mensaje a un destinatario, sin esperar una respuesta del destinatario. Este patrón es útil para las situaciones en las que únicamente se desea compartir información o no importa la respuesta del destinatario (Chase, 2006).

V. PLANTEAMIENTO DEL PROBLEMA

Como su nombre indica, un NOC (*Network Operation Center*) tiene la principal responsabilidad de la administración y monitoreo de los diversos equipos que componen las redes a su cargo y, consecuentemente, del buen funcionamiento de los servicios prestados por un ISP (*Internet Service Provider*). Como parte de la tarea de Monitoreo, dentro del NOC se utilizan diversas plataformas para la administración y monitoreo de los equipos. Algunas de las plataformas utilizadas son proporcionadas por el fabricante de los equipos, aunque en otros casos se utilizan herramientas de terceros para el monitoreo.

Además de la administración y monitoreo de redes, el NOC tiene a su cargo otra tarea muy importante: la atención de clientes. La importancia de esta tarea se debe a la competitividad entre proveedores, quienes buscan factores diferenciadores ante los clientes

Debido a la gran cantidad de plataformas y herramientas que utiliza el NOC para la administración y monitoreo de redes, y para la atención de clientes, surge la necesidad de una herramienta que centralice la información de monitoreo de la red y permita agilizar el diagnóstico de servicios. Incluso, permitir a los ingenieros actuar de forma proactiva ante los eventos e incidentes detectados en la red.

A. PROACTIVIDAD

La proactividad es una actitud en la que un sujeto u organización asume una conducta de activa, lo que implica tomar la iniciativa en el desarrollo de acciones creativas y audaces para generar mejoras. Además, se asume la responsabilidad de ejecución, es decir, hacer que las cosas sucedan, decidir en cada momento lo que se debe hacer y cómo hacerlo. El concepto opuesto es el de reactividad o tomar una actitud pasiva y ser sujeto de las circunstancias y los problemas. A nivel empresarial tiene mucha importancia adelantarse y ofrecer no solo lo que el cliente solicita, sino ir más allá, utilizando los recursos y experiencias profesionales para brindarle al cliente un valor añadido (Jiménez, 2009).

En el ámbito de un ISP, si el servicio brindado a un cliente está funcionando incorrectamente, no se debe esperar a que el cliente nos haga un reporte. En una actitud proactiva se debe tomar la iniciativa, detectando de forma inmediata el problema y notificando al cliente. Acto seguido, se debe iniciar un proceso de diagnóstico y solución del problema. Es cierto, que ser proactivos tanto a nivel personal como profesional no es fácil. Es algo que se ha de ir cultivando constantemente hasta conseguir que ésta sea parte de la actitud de un individuo u organización. La proactividad a nivel profesional, se considera más difícil que en la vida personal. Pero el secreto de conseguir la proactividad a nivel profesional, está en siempre querer dar lo mejor al cliente e incluirlo como parte de la cultura organizacional (Jiménez, 2009).

B. IMPORTANCIA DEL MONITOREO DE UNA RED

El monitoreo de la red es la función de recoger información administrativa de la red. El propósito del monitoreo es la recolección de información útil, a partir de varias partes de la red, para generar conocimiento del estado de la misma, tomando en cuenta que la mayoría de los dispositivos están ubicados en lugares remotos. Estos dispositivos por lo general no se conectan directamente a terminales de manera que la aplicación de gestión de red pueda controlar sus estados fácilmente. Por lo tanto, se han desarrollado técnicas de monitoreo de la red para permitir que las aplicaciones de gestión de red puedan comprobar los estados de sus dispositivos de red (Stallings, 1996).

A medida que más personas se comunican mediante redes, las redes se han convertido más grandes y complejas, impulsados principalmente por el aumento en el uso de Internet. Es importante que las aplicaciones de monitoreo utilicen medios eficaces para verificar el estado de sus redes, de modo que sean soluciones económicas y de alta calidad. Es muy importante saber cuáles son los objetivos a alcanzar en el monitoreo de una red, para elegir las técnicas de monitoreo adecuadas (Stallings, 1996).

En general, existen tres rubros básicos para el buen monitoreo de una red: el monitoreo del rendimiento, el monitoreo de fallas y el monitoreo de cuentas. Estos objetivos son tres de las cinco áreas funcionales de administración de redes propuestos por la OSI (*Open Systems Interconnect*). Las otras dos áreas funcionales son la gestión de la configuración y la gestión de la seguridad (Stallings, 1996).

El monitoreo del rendimiento, como su nombre indica, trata de medir el rendimiento de la red, basado en tres aspectos importantes: definir qué indicadores y medidas son importantes, definir el marco de tiempo para obtención de resultados y, definir un modelo de comportamiento para planificar la expansión de la red y problemas actuales de uso. Algunos de los indicadores comúnmente utilizados son descritos en la Tabla 7 (Stallings, 1996).

Tabla 7. Descripción de indicadores de red.

Indicador de red	Descripción
Disponibilidad del circuito	El tiempo verdadero que un usuario puede utilizar una red y la conexión de la red está disponible para el usuario.
Disponibilidad del nodo	El tiempo verdadero que un usuario puede usar nodos de la red, multiplexores y routers sin tener errores.
Factor de bloqueo	El número de usuarios que no puede accesar la red por estar ocupada.
Tiempo de respuesta	El tiempo para transmitir una señal y recibir una respuesta.

El monitoreo de fallas se ocupa de la medición de los problemas en la red. Hay dos cuestiones importantes en la supervisión de fallas: determinar en qué capas del modelo OSI se están dando los problemas y establecer una serie de características consideradas normales en una red. Siempre hay errores en la red, pero esto no significa que la red está teniendo problemas, de hecho, algunos de estos errores se espera que se produzcan. Un ejemplo de errores esperados, son los errores de transmisión producidos por el ruido en un enlace de red. Se dice que la red tiene un problema cuando el número de errores ha incrementado por encima de su comportamiento normal (Stallings, 1996).

El monitoreo de cuentas ocupa cómo los usuarios utilizan la red. La red mantiene un registro de los dispositivos utilizados por los usuarios y la frecuencia de uso. Este tipo de información puede ser utilizada para la facturación por el uso de la red y para predecir el uso futuro de la red (Stallings, 1996).

A continuación se indican brevemente algunos puntos que demuestran la importancia del monitoreo de una red (Downing, 2013).

- Fiabilidad: el monitoreo de la red permite el seguimiento de los incidentes críticos y notifica a los administradores de la red antes de que el impacto sea muy alto. Por ejemplo, el monitoreo de red le permite saber si un servidor falla, si el servicio deja de responder o si usted está en peligro de quedarse sin espacio en disco. Esto garantiza un enfoque proactivo para hacer frente a los problemas, en lugar de esperar a que los usuarios finales se encuentren con un problema.
- Conocimiento: el sistema de monitoreo le informa a los administradores de la red de los problemas de rendimiento y eventos de fallo mediante el envío de una serie de alertas a las computadoras, localizadores o dispositivos móviles. Esto permite que el administrador esté consciente de los problemas, independientemente de dónde se encuentre.
- Capacidad: tener un conocimiento profundo de cómo se están utilizando sus dispositivos le permite identificar de manera proactiva las áreas que requieren un mayor ancho de banda, mayor capacidad de procesamiento o más espacio en disco y desplegar capacidad adicional de una manera controlada.
- Solución de problemas: con el monitoreo de la red se puede identificar rápidamente el dispositivo que está causando el problema, lo que limita el tiempo de inactividad y la pérdida de tiempo tratar de diagnosticar el problema. En lugar de esperar a que un usuario final informe de un problema y la solución del mismo, el control de red permite al equipo de apoyo la detección, diagnóstico y solución del problema antes de que los usuarios sean conscientes de ello.
- Seguimiento de las tendencias: los problemas que se producen intermitentemente o en determinadas horas punta pueden ser complicados de encontrar, pero los informes de monitoreo de red en curso permiten comprender las principales tendencias en el rendimiento y la salud general de la red.

- Plan de mejoras y cambios: si un dispositivo está funcionando constantemente cerca de su límite, este nos puede indicar que es el momento de hacer un cambio. Las aplicaciones de monitoreo de red permiten realizar un seguimiento de este tipo de datos y planificar con anticipación para hacer los cambios necesarios con facilidad.
- Mostrar lo que está sucediendo: informes y estadísticas sobre la salud y la actividad de la red son una gran herramienta para probar la adhesión a un acuerdo de nivel de servicio o demostrar el por qué a las necesidades específicas del dispositivo de fijación o sustitución.
- Saber cuándo aplicar las soluciones de recuperación de desastres: sin monitoreo de la red, los principales problemas pueden pasar desapercibidos. Con la notificación adicional proporcionada por un control adecuado, se pueden implementar protocolos de recuperación de desastres a tiempo para evitar el tiempo de inactividad y/o fallos del sistema. Ejemplos de esto incluyen baterías de alimentación ininterrumpida durante un corte de energía, el estado de realización de copia de seguridad y la capacidad de predecir el fallo del disco duro.
- Asegurar que los sistemas de seguridad funcionan correctamente: sin monitoreo de la red, no hay forma de garantizar que los sistemas de seguridad de alta prioridad están desempeñando sus funciones. Un ejemplo es el firewall, que protege los datos y permite una conectividad segura a Internet.
- Ahorrar dinero: reducir el tiempo de inactividad y el tiempo de investigación, ya que menos horas de trabajo significa menos dinero gastado cuando se producen problemas y una mayor productividad en toda la organización.
- Aumento los beneficios: evitar las pérdidas financieras causadas por fallas en el sistema sin ser detectado es el resultado final de ser capaces de señalar de manera proactiva y hacer frente a problemas de la red. Todos sus servicios de misión crítica se ejecutarán sin problemas con un servicio de vigilancia, lo que permite más tiempo para hacer funcionar a la empresa.

C. ANTECEDENTES

En un ISP generalmente se proveen servicios que van desde Capa 1 hasta Capa 3 del modelo OSI. Los ingenieros de NOC trabajan con herramientas diariamente para el manejo, detección, diagnóstico y solución de los eventos en la red y las solicitudes de clientes. Las solicitudes del cliente, suelen agruparse en tres categorías:

- Reportes de problemas con el servicio: son los reportes en los que el servicio del cliente está completamente fuera de servicio.
- Reportes de inestabilidad en el servicio: son los reportes en los que el servicio del cliente está funcionando con irregularidades, como una alta pérdida de paquetes, problemas de Calidad del Servicio o QoS (*Quality of Service*), entre otros.

- Solicitudes del cliente: son aquellos reportes en los que el cliente solicita un cambio de configuración en el servicio, información de la configuración del servicio, entre otro tipo de solicitudes que no están relacionadas al funcionamiento del mismo.

Los primeros dos suelen estar relacionados con incidentes dentro de la red, e incluso un solo evento puede ser la razón de problemas o inestabilidad de varios servicios. Cada solicitud de cliente, requiere la apertura de un ticket para el seguimiento correspondiente. Las herramientas de tickets, generalmente permiten la asociación de estos tickets de seguimiento a otros denominados tickets maestros, para facilitar el seguimiento colectivo.

Para el diagnóstico de los servicios, a partir de ahora servicio se refere a los servicios VPN Capa 3 basados en BGP/MPLS, luego de un reporte de problemas o inestabilidad, se requiere de la revisión desde los dos extremos de la red del proveedor, a manera de segmentar el diagnóstico y determinar si se debe a problemas en la red del proveedor, o en la última milla del servicio, es decir, la conexión entre la red del proveedor y la del cliente.

Cuando la red del proveedor es demasiado grande, es conveniente separar ésta en pequeñas unidades. En algunos casos se define un ente regional que se encarga de la parte internacional y unidades pequeñas asignadas a cada país en donde el proveedor tiene presencia. En este modelo, las operaciones locales ven al operador regional como un proveedor, cuando existe la necesidad de implementar servicios internacionales en donde se requiere la intervención del mismo.

También existen casos en los que se requiere establecer un servicio en países o áreas en las que el ISP contratado por el cliente no tiene presencia, en esos casos se suele contratar servicios de ISP’s terceros que actúan como proveedores y a su vez estos ISP’s terceros catalogan a su cliente como cliente wholesale. En algunos casos se puede involucrar a más de un proveedor.

En casos donde se involucran diferentes redes, ya sea del mismo ISP o las de otros proveedores, la revisión de un servicio puede volverse una tarea compleja al tener que involucrar diferentes operaciones. Sin embargo, teniendo las IP’s del router CE, es posible determinar desde qué red se tiene problemas mediante pruebas desde ambos extremos de la red en la que se está trabajando. En este sentido, el servicio puede representarse de la forma mostrada en la Figura 27. Al tener una noción del origen del problema, se puede atacar si se encuentra en la red administrada o escalar a las operaciones correspondientes.

Figura 27. Representación general de un servicio de cliente

En un NOC regional, usualmente no se conocen las direcciones IP de los routers CE, pues son proveídas por las redes más cercanas al sitio del cliente, generalmente utilizando segmentos diferentes que los utilizados en las interconexiones entre redes para evitar colisiones. Observar las redes conocidas para una VPN, como se muestra en la Figura 28, no solventa el inconveniente, pues existen casos en que las Tablas de Rutas construidas son bastante extensas.

Figura 28. Extracción de la Tabla de Redes para un Servicio VPN Capa 3.

```
user@host> show route logical-system [redacted] table [redacted]

[redacted].inet.0: 26 destinations, 174 routes (26 active, 0 holddown,
147 hidden)
+ = Active Route, - = Last Active, * = Both

169.254.1.0/30     *[BGP/170] 6w5d 11:15:15, localpref 100, from 10.20.128.4
                      AS path: [redacted] [redacted] ?, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 50198, Push 557504(top)
169.254.2.0/30     *[BGP/170] 6w5d 11:15:46, localpref 100, from 10.20.128.4
                      AS path: [redacted] ?, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 45742, Push 557504(top)
169.254.5.0/30     *[BGP/170] 19w6d 21:15:41, localpref 100, from 10.20.128.4
                      AS path: [redacted] [redacted] I, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 41148, Push 557504(top)
169.254.100.0/30   *[Direct/0] 37w2d 17:07:04
                    > via ge-1/2/1.2006
                    [BGP/170] 1w4d 17:18:32, localpref 100, from 10.20.137.76
                      AS path: [redacted] [redacted] I, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 304288, Push 771856(top)
169.254.100.1/32   *[Local/0] 37w2d 17:07:15
                      Local via ge-1/2/1.2006
172.18.51.6/32     *[BGP/170] 6w5d 11:15:15, localpref 100, from 10.20.128.4
                      AS path: [redacted] [redacted] ?, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 49931, Push 557504(top)
172.31.231.104/30  *[BGP/170] 6w5d 11:15:15, localpref 100, from 10.20.128.4
                      AS path: [redacted] [redacted] ?, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 38155, Push 557504(top)
192.168.50.0/24    *[BGP/170] 6w5d 11:15:15, localpref 100, from 10.20.128.4
                      AS path: [redacted] [redacted] ?, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 50513, Push 557504(top)
192.168.60.0/24    *[BGP/170] 6w5d 11:15:15, localpref 100, from 10.20.128.4
                      AS path: [redacted] [redacted] ?, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 54551, Push 557504(top)
192.168.70.0/24    *[BGP/170] 6w5d 11:15:15, localpref 100, from 10.20.128.4
                      AS path: [redacted] [redacted] ?, validation-state: unverified
                    > to 10.20.11.101 via ae0.200, Push 51820, Push 557504(top)
---(more)---
```

Una revisión alternativa cuando no se conocen las redes de los routers es realizar pruebas en la red administrada y revisar el estado de las conexiones de los extremos de la red a las siguientes redes. Sin embargo, en este caso no se logra hacer una revisión completa o End-to-End del servicio VPN. En la Figura 29 se ejemplifica la revisión del estado de las conexiones a otras redes desde los routers PE.

Figura 29. Revisión del estado de las conexiones EGP desde los routers PE.

```
user@host> show bgp summary logical-system [redacted] table [redacted]
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
[redacted].inet.0
                     290         24          0          0          0          0
[redacted].mdt.0
                       0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State |
#Active/Received/Accepted/Damped...
172.18.206.106          [redacted]    2279089    2440725       0       3    25w3d14h Establ
  [redacted].inet.0: 1/1/1/0
192.168.254.6           [redacted]     653030     699728       0      11       7w2d4h Establ
  [redacted].inet.0: 1/1/1/0
```

En la revisión anterior, se observa el tiempo que llevan establecidas las sesiones BGP con las direcciones IP 172.18.206.106 y 192.168.254.6. Los estados que puede tener una sesión BGP son los siguientes:

- Estado libre: no se ha iniciado el proceso para establecer una sesión BGP.
- En conexión: indica que uno de los extremos intenta una conexión TCP.
- Activo: cuando uno de los extremos no puede establecer una conexión y lo reintanta periódicamente.
- OpenSent: un extremo envía un mensaje de identificación.
- OpenConfirm: se recibe una respuesta al mensaje de identificación enviado.
- Establecida: se aceptan las identificaciones. Desde este punto, la sesión se considera completamente activa (Soricelli, 2013).

El sistema utilizado para el control de servicios brindados a clientes, puede ser de utilidad, pues en muchos casos contiene la información de las IP's asignadas a los routers CE para los servicios de un cliente. Sin embargo, estas no se encuentran almacenadas en un registro determinado, algunas veces se colocan en los diseños de alto nivel (HLD, por sus siglas en inglés *High Level Design*) o en los comentarios del servicio cuando un operador de NOC obtiene estas direcciones y las agrega para futuras referencias.

D. IMPACTO DEL PROBLEMA

En la sección de Antecedentes se hace notar algunos de los inconvenientes que se tienen dentro de un NOC. Siendo más objetivos, además de la falta de conocimiento en algunos casos para poder realizar un diagnóstico *End-to-End*, se tiene una falta de proactividad en los procesos de atención al cliente del ISP.

La rutina de pruebas para determinar el origen del problema o inestabilidad de un servicio, puede demorar varios minutos. En momentos, donde existe una alta cantidad de incidentes, puede provocar la saturación de los ingenieros de turno, afectando también los tiempos de resolución o TTR (*Time to Resolve*) definidos en los contratos de servicio realizados entre el ISP y sus clientes. De lo anterior podemos observar que existen diversos afectados al no tener un monitoreo adecuado de la red del ISP y de los servicios prestados.

VI. DISEÑO DE LA SOLUCIÓN

Una de las principales tendencias de desarrollo en la actualidad son las aplicaciones web y móviles, es por ello que se eligió un desarrollo web para una herramienta que centralice la información del estado de la red y de los servicios brindados a los clientes.

A. METODOLOGÍA

La metodología empleada para el desarrollo y seguimiento del proyecto es Programación Extrema o XP (acrónimo del nombre en inglés *Extreme Programming*). XP es una metodología ágil que se enfoca en mejorar la calidad del software desarrollado y la sensibilidad al cambio de los requerimientos por parte del cliente, refiriéndose a los interesados en el proyecto. También busca que el código desarrollado sea ejecutable y mantenible (Wells, 2013).

El éxito de esta metodología se debe a que se orienta a la satisfacción del cliente por medio de ciclos cortos de entrega durante el desarrollo del proyecto, en lugar de la ejecución de un único ciclo de vida del proyecto, tratando de abarcar el desarrollo de todos los requerimientos. Esto le permite a los programadores a responder de forma confiable ante los cambios de los requerimientos del cliente (Wells, 2013).

1. Valores de la metodología.

Esta metodología se basa en los valores de comunicación, simplicidad, retroalimentación, respeto y coraje. La forma de trabajo es mediante la comunicación constante de todos los involucrados en el proyecto, incluyendo desarrolladores, administradores y clientes, en la que todos tienen una participación equitativa y colaborativa (Wells, 2013).

- Comunicación: Todos los involucrados forman parte del equipo de trabajo y buscan mantener una comunicación presencial diaria. El equipo mantiene comunicación a lo largo del ciclo de vida del proyecto.
- Simplicidad: Se busca realizar lo que el cliente solicita y no ir más allá, buscando que el valor de lo realizado sea el mayor posible. Se busca la ejecución de pasos simples para llegar al objetivo final, lo cual mitiga el impacto de las fallas ocurridas. La codificación también debe ser limpia, de modo que el mantenimiento sea sencillo.
- Retroalimentación: Cada iteración del ciclo de vida del proyecto debe ser tomado con un compromiso serio. Se debe comunicar efectivamente y escuchar cuidadosamente para poder aplicar cualquier cambio necesario.
- Respeto: Todos los miembros del equipo deben brindar y recibir el respeto merecido como miembro. Se busca que la participación de todos sea tomada como valiosa por más simple que parezca, valorando la experiencia tanto del cliente como de los desarrolladores.

- Coraje: Se debe decir la verdad acerca del progreso y las estimaciones. Se debe buscar siempre la solución a problemas encontrados en equipo, para evitar alto impacto en las fallas que puedan ocurrir. Este valor también se enfoca en la adaptabilidad a los cambios, buscando que los clientes indiquen cuando algo no cumple con lo requerido (Wells, 2013).

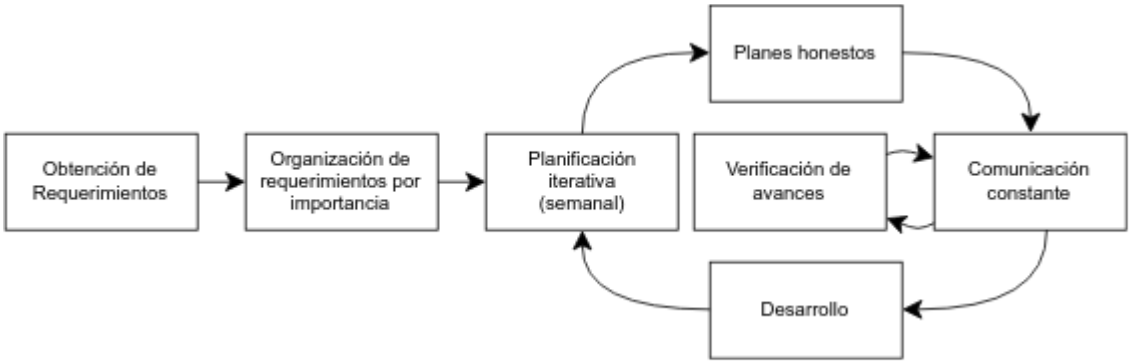
2. Buenas prácticas. En la metodología XP se reconoce que el objetivo principal de la administración de proyectos es brindar un producto de calidad, tanto en funcionalidad como en mantenibilidad. Para asegurar que el producto sea de calidad, XP hace uso de las siguientes buenas prácticas del desarrollo de software (Sambasivam):

- Planificación del juego: se refiere a la cooperación de las partes de negocio y desarrollo para concebir un producto con el máximo valor y el menor tiempo posible.
- Pequeñas entregas: las pequeñas entregas o iteraciones serán siempre con la entrega de código, pruebas y entrega al cliente con su respectiva valoración.
- Diseño simple: la implementación debe mantenerse siempre simple, pues los requerimientos pueden cambiar.
- Metáfora: se refiere a una visión común de los desarrolladores de cómo debe funcionar el programa. Esta metáfora debe mantenerse simple.
- Pruebas continuas: los equipos deben enfocarse en mantener la mentalidad de validar todo el código desarrollado.
- Refactorización: el equipo debe evitar tener código duplicado. Esto ayuda a simplificar el código y los casos de prueba.
- Programación en parejas: todo el código debe ser escrito por dos programadores utilizando una única máquina para asegurar que el código sea validado durante la escritura del mismo.
- Propiedad colectiva de código: ninguna persona debe hacerse dueña de un módulo específico, y todos los desarrolladores deben sentirse familiarizados con el trabajo de cualquier módulo del proyecto.
- Integración continua: Todos los cambios deben ser integrados en un repositorio luego de comprobar su funcionalidad.
- Trabajo semanal de 40 horas: No se debe exceder los tiempos de desarrollo semanales, pues un exceso en la exigencia de trabajo es un indicador de que algo está mal con el proceso o con la calendarización.
- Cliente en sitio: el equipo de desarrollo tiene constante interacción con el cliente, quien se encuentra actualmente utilizando el sistema.
- Estándares de codificación: todos deben codificar bajo el mismo estándar. No importa el estándar utilizado, pero sí importa que el código le parezca familiar a cualquier desarrollador, lo cual facilita también el soporte aplicativo (Sambasivam).

3. Aplicación. La metodología XP no es una plantilla completa que deba aplicarse a cualquier organización. Más bien, es un compendio de buenas prácticas y valores que ayudan a mejorar la interacción entre el equipo de desarrollo y el cliente. En el sentido de proceso, éste debe brindarle al equipo de involucrados la capacidad de crecimiento, cambio y adaptación ante las dificultades surgidas y nuevas necesidades del negocio.

Durante el desarrollo de este proyecto, se cuenta con algunas dificultades, siendo las principales el tiempo limitado para desarrollo de código (entre 10 a 15 horas semanales) y que cuenta con un único desarrollador. Debido a lo anterior, la calendarización del mismo también se ve extendida, teniendo reuniones semanales en lugar de reuniones diarias como sugiere XP. El ciclo de vida, queda definido como se muestra en la Figura 30, donde se incluyen los tiempos. La planificación de desarrollo del proyecto se resume en la Figura 31, aunque muchos aspectos de la planificación fueron modificados debido a cambios en los requerimientos y sus prioridades.

Figura 30. Ciclo del proyecto.



Debido al limitado número de desarrolladores, algunas de las buenas prácticas no pueden ser aplicadas. Sin embargo, esto no indica que la metodología no pueda ser exitosa, pues permite que exista una interacción constante y que los recursos limitados de desarrollo puedan orientarse y aprovecharse de la mejor forma posible.

Figura 31. Calendarización del proyecto.

	Name	Duration	Start	Finish	Predecessors	Resources
1	**Monitoreo**	**162d?**	**04/30/2015**	**12/11/2015**		
2	**Equipos para configuración de Probes**	**77d?**	**04/30/2015**	**08/14/2015**		
3	Selección de Servicios a monitorear	14d	04/30/2015	05/19/2015		
4	Cotización de Equipos	3d	04/30/2015	05/04/2015		
5	Ingreso de Orden de Compra	1d	05/05/2015	05/05/2015	4	
6	Aprobación de Orden de Compra	35d?	05/06/2015	06/23/2015	5	
7	Compra y entrega de Equipos	3d	06/24/2015	06/26/2015	6	
8	Distribución e Instalación de Equipos	21d?	06/29/2015	07/27/2015	7	
9	Pruebas de configuración de Equipos	7d	06/29/2015	07/07/2015	7	
10	Configuración general de Equipos	7d	07/28/2015	08/05/2015	8,9	
11	Configuración de Pruebas de Monitoreo	7d	08/06/2015	08/14/2015	3,10	
12	**Portal de Monitoreo**	**162d?**	**04/30/2015**	**12/11/2015**		
13	**Desarrollo**	**162d**	**04/30/2015**	**12/11/2015**		
14	**Re-estructuración Modular Portal NOC**	**19d**	**04/30/2015**	**05/26/2015**		
15	Actualización Pulling de RCP a SNMP	3d	04/30/2015	05/04/2015		
16	Detección Automática Logical Systems	3d	05/05/2015	05/07/2015	15	
17	**Estructura de Centralización de Datos**	**13d**	**05/08/2015**	**05/26/2015**	**16**	
18	Data Pulling Lecturas Modulares	3d	05/08/2015	05/12/2015		
19	Manejo de Usuarios y accesos Modulares	5d	05/13/2015	05/19/2015	18	
20	Relación de Datos por Módulo	5d	05/20/2015	05/26/2015	19	
21	**Soporte Capa 2 VPLS**	**26d**	**05/27/2015**	**07/01/2015**	**14**	
28	**Límite de pruebas por dispositivo**	**7d**	**07/02/2015**	**07/10/2015**	**21**	
31	**Actualización Pruebas RPM**	**7d**	**07/13/2015**	**07/21/2015**	**28**	
34	**Mapa de Rendimiento (RPM)**	**18d**	**07/22/2015**	**08/14/2015**	**31**	
35	Equipos por ubicación	3d	07/22/2015	07/24/2015		
36	Mapas por ubicación	3d	07/27/2015	07/29/2015	35	
37	Despliegue de equipos por ubicación	3d	07/30/2015	08/03/2015	36	
38	Filtrado de información de equipos y pruebas por e	4d	08/04/2015	08/07/2015	37	
39	Despliegue de datos por prueba en equipo	5d	08/10/2015	08/14/2015	38	
40	**Matriz de Rendimiento**	**7d**	**08/17/2015**	**08/25/2015**	**34**	
41	Parametrización de Orígenes	3d	08/17/2015	08/19/2015		
42	Parametrización de Destinos	4d	08/20/2015	08/25/2015	41	
43	**Tendencia de tráfico de Internet**	**9d**	**08/26/2015**	**09/07/2015**	**40**	
46	**Despliegue resultados históricos RPM**	**6d**	**09/08/2015**	**09/15/2015**	**43**	
49	**API con Sistema de Clientes**	**13d**	**09/16/2015**	**10/02/2015**	**46**	
53	Pruebas a servicios en base a instalación	5d	10/05/2015	10/09/2015	49	
54	**VPN existentes**	**6d**	**10/12/2015**	**10/19/2015**	**53**	
57	**Procesos Integrados**	**16d**	**10/20/2015**	**11/10/2015**	**54**	
58	Integrar proceso de Bajas	5d	10/20/2015	10/26/2015		
59	Integrar proceso de Cambios	6d	10/27/2015	11/03/2015	58	
60	Integrar proceso de Upgrades	5d	11/04/2015	11/10/2015	59	
61	**Integrar proceso de Delivery**	**15d**	**11/11/2015**	**12/01/2015**	**57**	
62	Definición de proceso	4d	11/11/2015	11/16/2015		
63	Implementación de automatización	5d	11/17/2015	11/23/2015	62	
64	Validación de proceso	6d	11/24/2015	12/01/2015	63	
65	Dashboard de Notificaciones	8d	12/02/2015	12/11/2015	61	
66	**Ambiente de Ejecución y Deploy**	**36d?**	**04/30/2015**	**06/18/2015**		
72	**Pruebas**	**21d**	**06/19/2015**	**07/17/2015**	**66**	

B. REQUERIMIENTOS

Durante la concepción del proyecto, se obtuvieron diversos requerimientos, varios de los cuales quedan fuera del alcance de este trabajo. A continuación se listan los requerimientos considerados, incluyendo también el detalle de los cambios concebidos durante la vida del proyecto.

El sistema debe proveer la facilidad de acceso a información centralizada y a un conjunto de herramientas que ayuden en el proceso de diagnóstico de inconvenientes con los servicios brindados por un ISP (*Internet Service Provider*). Se desea que la herramienta cuente con una interfaz Web, a través de la cual se pueda acceder de forma controlada a los diferentes módulos de la misma, los cuales se agrupan en las siguientes categorías:

1. Herramientas de diagnóstico de la red y servicios
    a. Looking Glass
    b. Monitoreo de los equipos de la red MPLS
        1) Consumo de memoria RAM
        2) Consumo de procesador
        3) Temperatura del dispositivo
    c. Monitoreo de la red MPLS
        1) Rendimiento de la red basado en RPM
        2) Consumo de tráfico por LSP
        3) Consumo de tráfico por interfaz de red
2. Herramientas de diagnóstico de un servicio VPN Capa 3
    a. Monitoreo de servicios VPN Capa 3
        1) Rendimiento del servicio basado en RPM
        2) Consumo de tráfico en interfaces de red de los routers PE
        3) Configuración del servicio
        4) Tablas de rutas del servicio
    b. Pruebas en tiempo real sobre servicios VPN Capa 3
3. Herramientas de monitoreo
    a. Tablero de alertas
    b. Apertura automática de tickets

Dentro de las herramientas de diagnóstico se concibe inicialmente la existencia de usuarios para clientes del ISP, de modo que ellos tengan la capacidad de visualizar el estado de sus propios servicios y de realizar pruebas controladas. Este módulo estaba pensado exclusivamente para clientes Wholesale, quienes generalmente se tratan de otros NOC's. Sin embargo, esta idea fue descartada por cambios en los requerimientos por parte del cliente del proyecto, quedando únicamente para el uso del NOC del ISP, y se dio prioridad a la apertura automática de Tickets por medio del consumo de los servicios Web disponibles en la herramienta *Tívoli Service Request Manager* (TSRM) de IBM.

Dentro de los requerimientos recopilados, podemos también hacer notar la necesidad de los siguientes requerimientos no funcionales:

1. Conexión a los dispositivos Juniper de la red MPLS regional
    a. Conexión por NETCONF
    b. Conexión por SNMP
2. Descubrimiento de Sistema Lógicos
3. Descubrimiento de servicios VPN capa 3 en la red
    a. Descubrimiento de routing-instances de tipo VRF
    b. Descubrimiento de interfaces asociadas a una routing instance
    c. Asociación de routing-instances pertenecientes a la misma VPN
4. Descubrimiento de RPM's en equipos de monitoreo
    a. Clasificación de RPM's para verificación del estado de la red
    b. Clasificación de RPM's para verificación del estado de servicios VPN Capa 3
5. Descubrimiento de LSP's
6. Descubrimiento de Interfaces de Red y asociación a routing instances

C. LIMITACIONES

Es importante mencionar las limitaciones consideradas en este trabajo de graduación, las cuales son listadas a continuación:

- Compatibilidad con dispositivos de red: anteriormente se ha mencionado que se estará trabajando en el desarrollo de una herramienta que se conecte a equipos Juniper. Los equipos Juniper con los que se trabajó son de las series MX y M.
- Servicios monitoreados: únicamente se consideran servicios de VPN Capa 3 basados en BGP y MPLS.
- Soporte del sistema: el soporte del sistema se divide en dos secciones, la primera es el soporte de hardware que será responsabilidad por el ISP, quien brinda el hosting para el sistema en un ambiente virtualizado. El soporte de Software es mínimo, pues se está utilizando herramientas de software distribuidas bajo licencias Open Source.

D. CASOS DE USO DEL SISTEMA

Los casos de uso ayudan a especificar el comportamiento de un sistema, describiendo las interacciones que tiene con usuarios, dispositivos y otros sistemas, conocidos como Actores o Roles.

1. Actores. Los actores identificados se presentan a continuación:

    a. Usuario: o Usuario no autenticado al sistema. Las acciones que puede realizar un usuario no autenticado requieren que se encuentre previamente registrado en el Sistema.

b. Administrador: o administrador del Sistema. Es un usuario autenticado a cargo de la administración del sistema. Su perfil incluye una serie de permisos asociados a tareas administrativas.

c. Ingeniero de NOC: usuario autenticado que cuenta con acceso a los diferentes módulos, cada uno de los cuales se encuentra asociado a perfiles o roles de usario que pueden ser asignados por un administrador.

d. Dispositivo de Red: representa todos aquellos dispositivos a los cuales el sistema estará conectándose para obtener información administrativa y del estado de los mismos con fines de monitoreo de la red a la que pertenecen. En el alcance se indica que se está trabajando con Routers Juniper de las series MX y M.

e. Herramienta de Tickets: representa el Sistema de Tickets utilizado dentro de un NOC. El sistema utilizado es la solución IBM TSRM, a la cual se conecta a este servicio por medio de Webservices de tipo SOAP (siglas del inglés *Simple Object Access Protocol*).

2. Resumen de casos de uso. En la Tabla 8 se lista los principales Casos de Uso identificados y actores involucrados.

Tabla 8. Resumen de casos de uso

Código	Caso de uso	Actores
CU-01	Autenticación al sistema	Usuario no autenticado
CU-02	Recuperación de constraseña	Usuario no autenticado
CU-03	Administración de usuarios	Administrador
CU-04	Administración de perfiles de usuario	Administrador
CU-05	Administración de dispositivos de red	Administrador
CU-06	Administración de tareas en CRON	Administrador
CU-07	Descubrimiento de sistemas lógicos	Dispositivo de red
CU-08	Descubrimiento de RPM’s	Dispositivo de red
CU-09	Descubrimiento de VRF’s	Dispositivo de red
CU-10	Descubrimiento de interfaces de red	Dispositivo de red
CU-11	Descubrimiento de LSP’s	Dispositivo de red
CU-12	Monitoreo de dispositivos	Dispositivo de red
CU-13	Monitoreo de RPM’s	Dispositivo de red

Continuación Tabla 8.

Código	Caso de uso	Actores
CU-14	Monitoreo de tablas de rutas	Dispositivo de red
CU-15	Monitoreo de tablas de direcciones MAC	Dispositivo de red
CU-16	Monitoreo de interfaces de Red	Dispositivo de red
CU-17	Monitoreo de LSP’s	Dispositivo de red
CU-18	Configuración de VRF’s	Ingeniero de NOC
CU-19	Configuración de RPM’s	Ingeniero de NOC
CU-20	Configuración de interfaces de red	Ingeniero de NOC
CU-21	Configuración de LSP’s	Ingeniero de NOC
CU-22	Configuración de umbrales de alerta	Ingeniero de NOC
CU-23	Pruebas en Looking Glass	Ingeniero de NOC, Dispositivo de Red
CU-24	Ver detalle y estado de instalaciones	Ingeniero de NOC
CU-25	Pruebas en tiempo real sobre VPN’s	Ingeniero de NOC, Dispositivo de Red
CU-26	Ver alertas activas, inactivas e históricas	Ingeniero de NOC
CU-27	Marcar alerta activa como inactiva	Ingeniero de NOC
CU-28	Enviar alerta activa a historial	Ingeniero de NOC
CU-29	Apertura de tickets	Ingeniero de NOC, IBM TSRM
CU-30	Busqueda de tickets	Ingeniero de NOC, IBM TSRM
CU-31	Añadir comentarios a ticket	Ingeniero de NOC, IBM TSRM

El diagrama de casos de uso general se muestra en las Figuras 32, 33 y 34. En este diagrama se incluyen los casos de uso listados anteriormente, así como otros casos utilizados para explicar actividades específicas y actividades generales de algunos casos de uso.

Figura 32. Diagrama de casos de uso - parte 1.

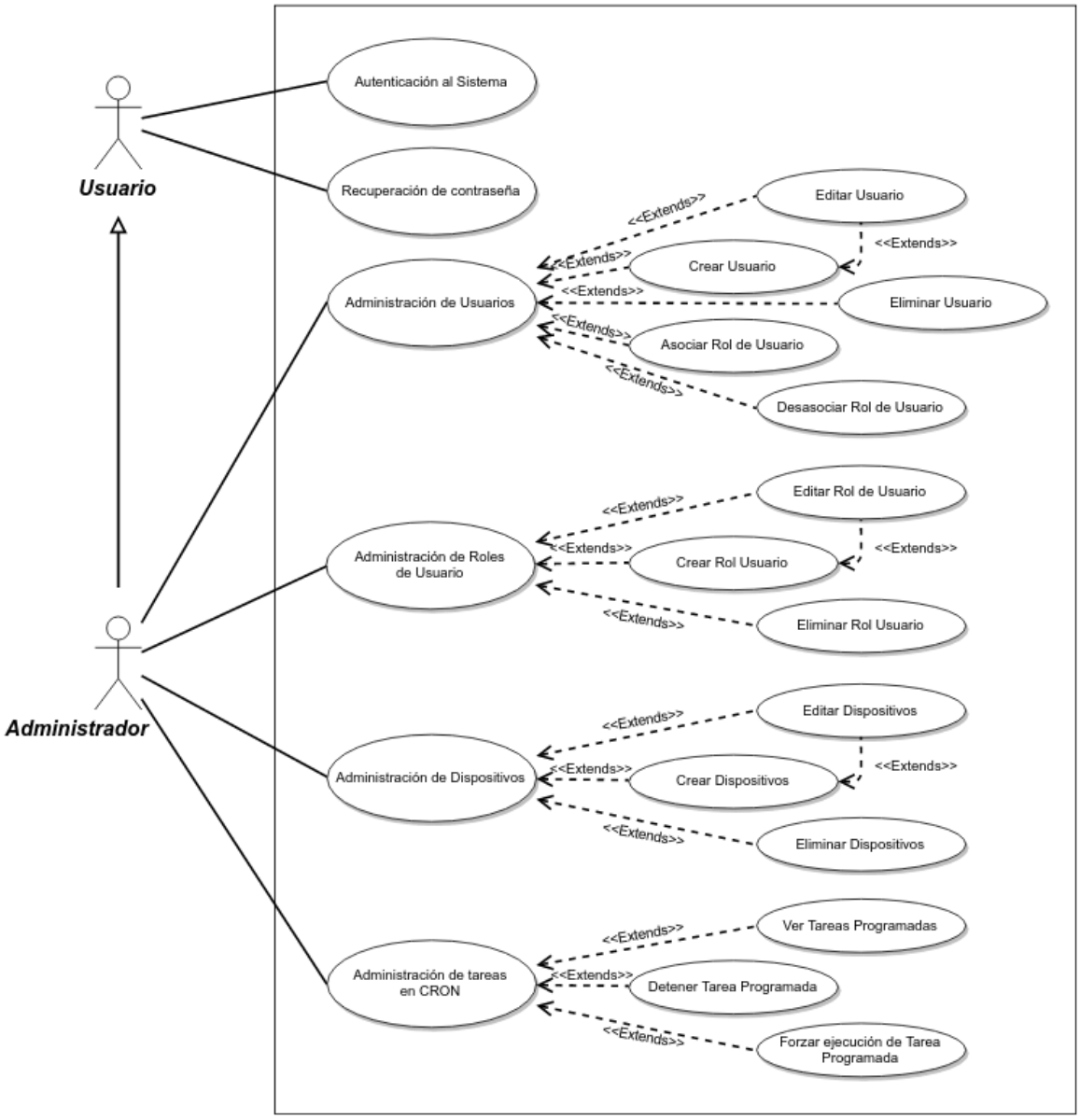


Figura 33. Diagrama de casos de uso - parte 2.

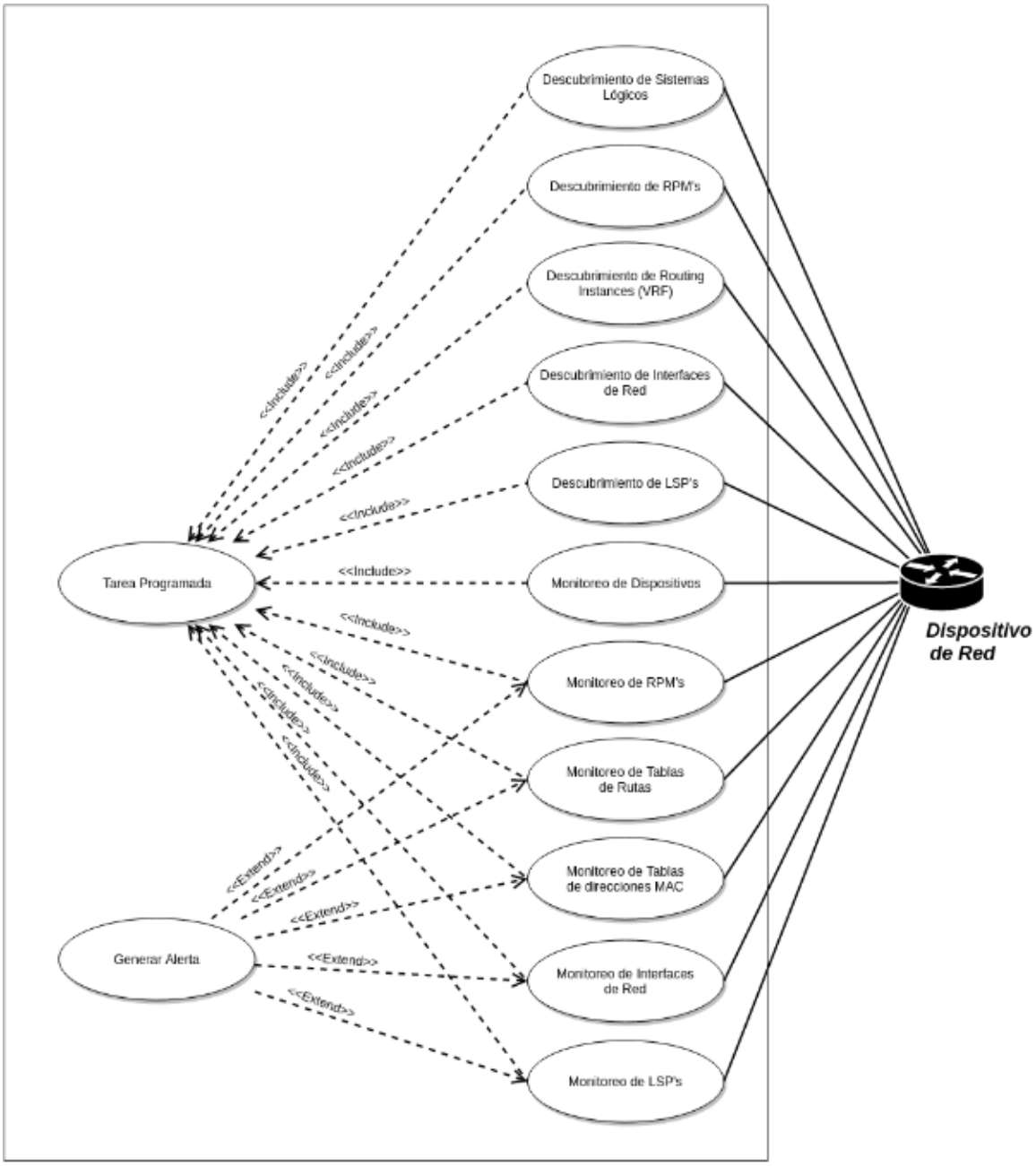
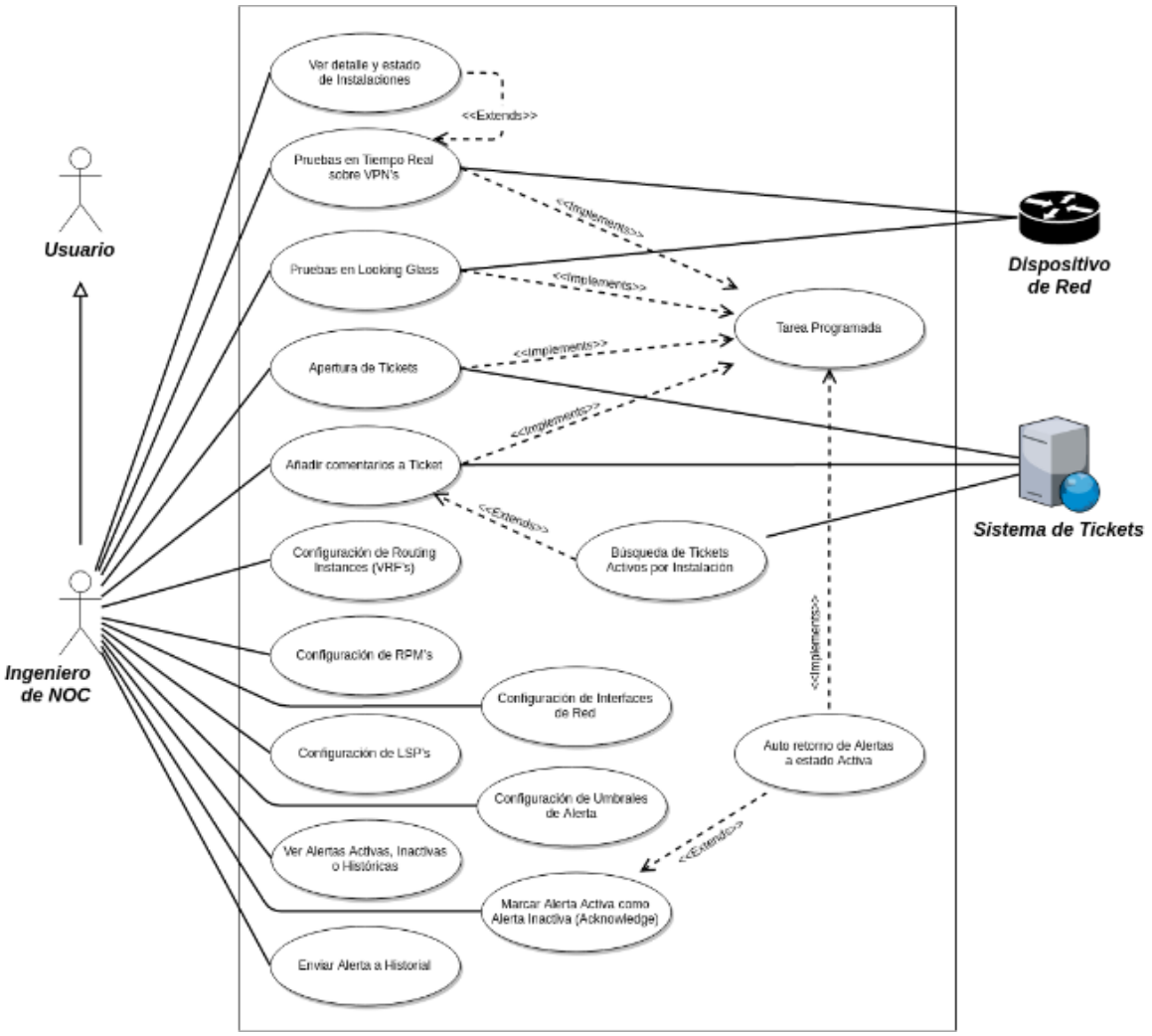


Figura 34. Diagrama de casos de uso - parte 3.



3. Descripción de los casos de uso. Los casos de uso son descritos desde las Tablas 33 hasta la Tabla 52.

Tabla 9. Descripción de caso de uso “Autenticación al sistema”.

Nombre	Autenticación al sistema.	Código	CU-01
Actores	Usuario no autenticado.		
Descripción	Permite a un usuario no autenticado, previamente fue registrado en el sistema, autenticarse al sistema.		
Requerimientos	Correo electrónico del usuario registrado. Contraseña del usuario registrado.		

Continuación Tabla 9.

Precondiciones	Usuario previamente registrado en el sistema.
Flujo normal 1. El Usuario accede a la vista de autenticación del sistema. 2. El Usuario ingresa su correo electrónico y contraseña. 3. El Usuario hace click en el botón “ingresar”. 4. El Usuario es autenticado correctamente.	
Flujo alternativo FA4. El usuario no es autenticado correctamente debido al ingreso incorrecto de su correo electrónico o contraseña. Regresa al paso 2 del Flujo Normal.	
Poscondiciones	El usuario se autentica correctamente.

Tabla 10. Descripción de caso de uso “Recuperación de contraseña”.

Nombre	Recuperación de contraseña.	Código	CU-02
Actores	Usuario no autenticado.		
Descripción	Permite a un usuario, previamente registrado al sistema, definir una nueva contraseña de acceso.		
Requerimientos	Correo electrónico de usuario registrado.		
Precondiciones	Usuario previamente registrado en el sistema.		
Flujo normal 1. El usuario accede a la vista de recuperación de contraseña. 2. El usuario ingresa su correo electrónico. 3. El usuario hace click en el botón “Reiniciar contraseña”. 4. El usuario recibe un correo electrónico, que contiene un enlace temporal para el ingreso de una nueva contraseña. 5. El usuario sigue el enlace recibido y define una nueva contraseña. 6. El sistema realiza el cambio de contraseña y notifica al usuario.			
Flujo alternativo FA4. El correo electrónico ingresado no corresponde a un usuario registrado en el sistema. Regresar al paso 2 del Flujo normal.			
Poscondiciones	El Usuario define una nueva contraseña.		

Tabla 11. Descripción de caso de uso “Edición de usuario”.

Nombre	Edición de usuario.	Código	CU-03-1
Actores	Administrador.		
Descripción	Permite la edición de un usuario.		
Requerimientos	Nuevos datos del usuario a editar.		
Precondiciones	El Usuario a editar existe en el sistema. Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de usuarios. 2. El administrador selecciona el usuario a editar y hace click en el botón “Editar” correspondiente. 3. El administrador modifica alguno de los campos del formulario de usuario con los datos deseados. 4. El administrador hace click en el botón “Enviar”. 5. El sistema almacena los datos del usuario.			
Flujo alternativo FA5. Existen valores inválidos en alguno de los campos del formulario de edición de usuario. Regresa al paso 3 del Flujo normal.			
Poscondiciones	El usuario es editado.		

Tabla 12. Descripción de caso de uso “Crear usuario”.

Nombre	Crear usuarios.	Código	CU-03-2
Actores	Administrador.		
Descripción	Permite la creación de un nuevo usuario.		
Requerimientos	Datos del usuario a crear.		
Precondiciones	Usuario administrador autenticado al Sistema.		
Flujo normal 1. El administrador ingresa a la vista de usuarios. 2. El administrador hace click en el botón “Nuevo usuario”. 3. El administrador llena los campos del formulario de usuario. 4. El administrador hace click en el botón “Enviar”. 5. El nuevo usuario es creado.			

Continuación Tabla 12.

Flujo alternativo FA5. Existen valores inválidos en alguno de los campos del formulario de usuario. Regresa al paso 3 del Flujo normal.

Poscondiciones	El Usuario es creado.

Tabla 13. Descripción de caso de uso “Eliminar usuario”.

Nombre	Eliminar usuario.	Código	CU-03-3
Actores	Administrador.		
Descripción	Permite elminar un usuario registrado en el sistema.		
Requerimientos	Nombre del usuario a eliminar.		
Precondiciones	El usuario a eliminar debe estar registrado en el sistema Usuario administrador autenticado al sistema.		
Flujo normal 1. El Administrador ingresa a la vista de Usuarios. 2. El Administrador selecciona el usuario a eliminar y hace click en el botón “Eliminar” correspondiente. 3. El administrador confirma la eliminación del usuario. 4. El usuario es eliminado.			
Flujo alternativo FA3. El administrador cancela la eliminación del usuario. Regresa al paso 1 del flujo normal			
Poscondiciones	El usuario es eliminado		

Tabla 14. Descripción de caso de uso “Asociar perfil de usuario”

Nombre	Asociar perfil de usuario.	Código	CU-03-4
Actores	Administrador.		
Descripción	Permite asociar un perfil de usuario a un usuario registrado en el sistema.		
Requerimientos	Identificador del usuario Identificador del perfil de usuario		
Precondiciones	El usuario está registrado en el sistema. El perfil de usuario existe en el sistema (ver CU-04-2 en la Tabla 17). Usuario administrador autenticado al sistema.		

Continuación Tabla 14.

Flujo normal 1. El administrador ingresa a la vista de usuarios. 2. El administrador selecciona el usuario a editar y hace click en el botón “Perfiles” correspondiente. 3. El administrador accede a la vista de perfil de usuario. 4. El administrador hace click en el botón “agregar perfil de usuario”. 5. El administrador accede a la vista de asociación de perfil a un usuario. 6. El administrador selecciona el perfil de usuario a asociar. 7. El administrador hace click en el botón “Enviar”. 8. El perfil de usuario es asociado al usuario.
Flujo alternativo FA8. El perfil de usuario ya se encuentra asociado. Se continúa en el paso 6 del Flujo normal.

Poscondiciones	Se asigna un nuevo perfil de usuario a un usuario.

Tabla 15. Descripción de caso de uso “Desasociación de perfil usuario”

Nombre	Desasociación de perfil de usuario.	Código	CU-03-5
Actores	Administrador.		
Descripción	Permite desasociar un perfil de usuario de un usuario registrado en el sistema.		
Requerimientos	Identificador del usuario Identificador del perfil de usuario		
Precondiciones	El usuario está registrado en el sistema. El perfil de usuario existe en el sistema (ver CU-04-2 en la Tabla 17). Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de usuarios. 2. El administrador selecciona el usuario a editar y hace click en el botón “Perfiles” correspondiente. 3. El administrador accede a la vista de perfiles de usuario, donde se listan los roles de usuario asociados al usuario 4. El administrador hace click en el botón “Eliminar perfil de usuario” correspondiente. 5. El administrador confirma la eliminación. 6. El perfil de usuario es desasociado del usuario.			

Continuación Tabla 15.

Flujo alternativo FA5. El usuario cancela la eliminación del perfil de usuario. Continúa en el paso 3 del Flujo normal.	
Poscondiciones	El perfil de usuario es desasociado de un usuario.

Tabla 16. Descripción de caso de uso “Edición de perfil de usuario”.

Nombre	Edición de perfil de usuario.	Código	CU-04-1
Actores	Administrador.		
Descripción	Permite la edición de un perfil de usuario.		
Requerimientos	El perfil de usuario a editar existe en el sistema.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de perfiles de usuarios. 2. El administrador selecciona el perfil de usuario a editar y hace click en el botón “Editar” correspondiente. 3. El administrador modifica alguno de los campos del formulario de perfil de usuario. 4. El administrador hace click en el botón “Enviar”. 5. El perfil de usuario es editado correctamente.			
Flujo alternativo FA5. Existen valores inválidos en alguno de los campos del formulario de perfil de usuario. Regresa al paso 3 del Flujo normal.			
Poscondiciones	El perfil de usuario es editado.		

Tabla 17. Descripción de caso de uso “Crear perfil de usuario”.

Nombre	Crear perfil de usuario.	Código	CU-04-2
Actores	Administrador.		
Descripción	Permite la creación de un nuevo perfil de usuario.		
Requerimientos	Datos del perfil de usuario a crear.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de perfiles de usuarios. 2. El administrador hace click en el botón “Nuevo perfil de usuario”.			

Continuación Tabla 17.

Flujo normal 3. El administrador llena los campos del formulario de perfil de usuario. 4. El administrador hace click en el botón “Enviar”. 5. El nuevo perfil de usuario es creado.
Flujo alternativo FA5. Existen valores inválidos en alguno de los campos del formulario de perfil de usuario. Regresa al paso 3 del Flujo normal.

Poscondiciones	El perfil de usuario es creado.

Tabla 18. Descripción de caso de uso “Eliminar perfil de usuario”

Nombre	Eliminar perfil de usuario.	Código	CU-03-3
Actores	Administrador.		
Descripción	Permite eliminar un perfil de usuario del sistema.		
Requerimientos	El perfil de usuario a eliminar debe existir en el sistema.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de perfil de usuarios. 2. El administrador selecciona el perfil de usuario a eliminar y hace click en el botón “Eliminar” correspondiente. 3. El administrador confirma la eliminación del perfil de usuario. 4. El perfil de usuario es eliminado.			
Flujo alternativo FA3. El administrador cancela la eliminación del perfil de usuario; regresa al paso 1 del Flujo normal.			
Poscondiciones	El perfil de usuario es eliminado.		

Tabla 19. Descripción de caso de uso “Edición de dispositivos de red”.

Nombre	Edición de dispositivos de red.	Código	CU-05-1
Actores	Administrador.		
Descripción	Permite la edición de un dispositivos de red.		
Requerimientos	El dispositivos de red a editar existe en el sistema.		

Continuación Tabla 19.

Precondiciones	Usuario administrador autenticado al sistema.
Flujo normal 1. El administrador ingresa a la vista de dispositivos de red. 2. El administrador selecciona el dispositivos de red a editar y hace click en el botón “Editar” correspondiente. 3. El administrador modifica alguno de los campos del formulario de dispositivos de red. 4. El administrador hace click en el botón “Enviar”. 5. El dispositivos de red es editado correctamente.	
Flujo alternativo FA5. Existen valores inválidos en alguno de los campos del formulario de dispositivos de red. Regresa al paso 3 del Flujo formal.	
Poscondiciones	El dispositivos de red es editado.

Tabla 20. Descripción de caso de uso “Crear dispositivos de red”.

Nombre	Crear dispositivos de red.	Código	CU-05-2
Actores	Administrador.		
Descripción	Permite la creación de un nuevo dispositivos de red.		
Requerimientos	Datos del dispositivos de red a crear.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de dispositivos de red. 2. El administrador hace click en el botón “Nuevo dispositivos de red”. 3. El administrador llena los campos del formulario de dispositivos de red. 4. El administrador hace click en el botón “Enviar”. 5. El nuevo dispositivos de red es creado.			
Flujo alternativo FA5. Existen valores inválidos en alguno de los campos del formulario de dispositivos de red. Regresa al paso 3 del Flujo formal.			
Poscondiciones	El dispositivos de red es creado.		

Tabla 21. descripción de caso de uso “Eliminar dispositivos de red”

Nombre	Eliminar dispositivos de red.	Código	CU-05-3
Actores	Administrador.		
Descripción	Permite eliminar un dispositivos de red del sistema.		
Requerimientos	El dispositivos de red a eliminar debe existir en el sistema.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de usuarios. 2. El administrador selecciona el usuario a eliminar y hace click en el botón “Eliminar” correspondiente. 3. El administrador confirma la eliminación del usuario. 4. El usuario es eliminado.			
Flujo alternativo FA3. El administrador cancela la eliminación del usuario. Regresa al paso 1 del Flujo normal			
Poscondiciones	El usuario es eliminado		

Tabla 22. Descripción de caso de uso “Ver tarea programada”

Nombre	Ver tarea programada.	Código	CU-06-1
Actores	Administrador.		
Descripción	Permite visualizar los detalles de una tarea programada.		
Requerimientos	La tarea programada debe existir en el sistema.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de tareas programadas. 2. El administrador selecciona la tarea programada a visualizar. 3. Se despliega el detalle de la vista programada.			
Flujo alternativo Ninguno			
Poscondiciones	Ninguna		

Tabla 23. Descripción de caso de uso “Detener tarea programada”

Nombre	Detener tarea programada.	Código	CU-06-2
Actores	Administrador.		
Descripción	Permite detener la ejecución de una tarea programada.		
Requerimientos	La tarea programada debe existir en el sistema.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de tareas programadas. 2. El administrador selecciona la tarea programada a visualizar. 3. El administrador accede al detalle de tarea programada. 4. El usuario administrador hace click en el botón “Detener tarea”. 5. La tarea programada es detenida, quedando en estado detenido.			
Flujo alternativo FA4. El administrador cancela la acción de detener la tarea programada. Continúa en el paso 3 del Flujo normal.			
Poscondiciones	La tarea programada es detenida.		

Tabla 24. Descripción de caso de uso “Forzar ejecución de tarea programada”

Nombre	Forzar ejecución de tarea programada.	Código	CU-06-3
Actores	Administrador.		
Descripción	Permite forzar la ejecución de una tarea programada.		
Requerimientos	La tarea programada debe existir en el sistema.		
Precondiciones	Usuario administrador autenticado al sistema.		
Flujo normal 1. El administrador ingresa a la vista de tareas programadas. 2. El administrador selecciona la tarea programada a visualizar. 3. El administrador accede al detalle de tarea programada. 4. El administrador hace click en el botón “Ejecutar ahora”. 5. La tarea programada es modificada para ejecutarse, quedando en estado en cola.			
Flujo alternativo Ninguno			
Poscondiciones	La tarea programada es modificada para su ejecución inmediata.		

Tabla 25. Descripción de caso de uso adicional “Ejecución de una tarea programada”.

Nombre	Ejecución de una tarea programada.	Código	CUA-01
Actores	Ninguno		
Descripción	Es una tarea que se ejecuta sin intervención de un actor. Estas tareas son iniciadas según una calendarización con alguna periodicidad.		
Requerimientos	Tarea programada dentro de la calendarización de ejecución.		
Precondiciones	La tarea programada existe en el sistema y se encuentra en estado en cola.		
Flujo normal 1. El Scheduler verifica la calendarización de las tareas en cola. 2. Se alcanza el tiempo de ejecución definido en la calendarización de las tareas programada. 3. Se llama un método o función asociado a la tarea programada para ejecutarse como tarea secundaria. 4. Se finaliza la ejecución, quedando la tarea en estado finalizado.			
Flujo alternativo FA2. No se alcanza el tiempo de ejecución. Se espera hasta el siguiente ciclo de ejecución del Scheduler de Web2py. Se regresa al paso 1 del Flujo normal. FA4. Ocurre un error durante la ejecución de la Tarea. Se termina la tarea quedando en estado de error y se almacena el traceback generado por el error. FA4. La ejecución de la tarea no termina en el tiempo estipulado para su ejecución. La tarea se termina quedando en estado timeout.			
Poscondiciones	Se ejecuta una tarea programada.		

Tabla 26. Descripción de caso de uso adicional “Generar alerta”.

Nombre	Generar alerta.	Código	CUA-02
Actores	Ninguno.		
Descripción	Permite la generación de una nueva alerta. Esta rutina es invocada por las tareas de monitoreo.		
Requerimientos	Parámetros para la generación de la nueva alerta.		
Precondiciones	Ninguna.		
Flujo normal 1. Se verifican los parámetros de la nueva alerta a generar. 2. Se genera un nuevo objeto alerta.			

Continuación Tabla 26.

Flujo normal 3. Se verifica que no existan alertas asociadas en estado activo o Acknowledge. 4. Se almacena la nueva alerta creada.
Flujo alternativo FA3. Se encuentran alertas asociadas en estado activo o acknowledge. Se aumenta el contador de eventos de la alerta y se actualiza la información de la misma.

Poscondiciones	La genera una nueva alerta.

Tabla 27. Descripción de caso de uso “Descubrimiento de sistemas lógicos”.

Nombre	Descubrimiento de sistemas lógicos.	Código	CU-07
Actores	Dispositivo de red.		
Descripción	Permite la obtención de los sistemas lógicos configurados en un dispositivo de Red. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Ninguna.		
Flujo normal 1. Se abre una conexión por NETCONF al dispositivo. 2. Se solicita la información de los sistemas lógicos configurados. 3. Se cierra la conexión establecida en el paso 1. 4. Se almacena la información en la base de datos.			
Flujo alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.			
Poscondiciones	Se almacena la información de los sistemas lógicos configurados en un dispositivo de red.		

Tabla 28. Descripción de caso de iso “Descubrimiento de RPM’s”.

Nombre	Descubrimiento de RPM’s.	Código	CU-08
Actores	Dispositivo de red.		
Descripción	Permite la obtención de los RPM’s configurados en un dispositivo de red. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Ninguna.		
Flujo normal 1. Se abre una conexión por NETCONF al dispositivo. 2. Se solicita la información de los RPM’s configurados. 3. Se cierra la conexión establecida en el paso 1. 4. Se almacena la información en la base de datos.			
Flujo alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.			
Poscondiciones	Se almacena la información de los RPM’s configurados en un dispositivo de red.		

Tabla 29. Descripción de caso de uso “Descubrimiento de Routing Instances”.

Nombre	Descubrimiento de Routing Instances.	Código	CU-09
Actores	Dispositivo de red.		
Descripción	Permite la obtención de las Routing Instances configuradas en los sistemas lógicos de un dispositivo de red. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Se ha realizado el proceso de descubrimiento de sistemas lógicos (referirse al CU-07, en la Tabla 27).		
Flujo normal 1. Se abre una conexión por NETCONF al dispositivo. 2. Se solicita la información de las Routing Instances configuradas en cada sistema lógico. 3. Se cierra la conexión establecida en el paso 1. 4. Se almacena la información en la base de datos.			

Continuación Tabla 33.

Flujo alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.	
Poscondiciones	Se almacena la información de las Routing Instances configuradas en los sistemas lógicos de un dispositivo de red.

Tabla 30. Descripción de caso de uso “Descubrimiento de interfaces de red”.

Nombre	Descubrimiento de interfaces de red.	Código	CU-10
Actores	Dispositivo de red.		
Descripción	Permite la obtención de las interfaces de red configuradas en los sistemas lógicos de un dispositivo de red. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Se ha realizado el proceso de descubrimiento de sistemas lógicos (referirse al CU-07, en la Tabla 27).		
Flujo normal 1. Se abre una conexión por NETCONF al dispositivo. 2. Se solicita la información de las interfaces de red configuradas en cada sistema lógico. 3. Se cierra la conexión establecida en el paso 1. 4. Se almacena la información en la base de datos.			
Flujo alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.			
Poscondiciones	Se almacena la información de las interfaces de red configuradas en los sistemas lógicos de un dispositivo de red.		

Tabla 31. Descripción de caso de uso “Descubrimiento de LSP’s”.

Nombre	Descubrimiento de LSP’s.	Código	CU-11
Actores	Dispositivo de red.		
Descripción	Permite la obtención de los LSP’s configurados en los sistemas lógicos de un dispositivo de red. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		

Continuación Tabla 31.

Requerimientos	El dispositivo debe existir dentro del sistema.
Precondiciones	Se ha realizado el proceso de descubrimiento de sistemas lógicos (CU-07, en la Tabla 27).
Flujo normal 1. Se abre una conexión por NETCONF al dispositivo. 2. Se solicita la información de los LSP's configurados en cada sistema lógico. 3. Se cierra la conexión establecida en el paso 1. 4. Se almacena la información en la base de datos.	
Flujo alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.	
Poscondiciones	Se almacena la información de los LSP's configurados en los sistemas lógicos de un dispositivo de red.

Tabla 32. Descripción de caso de uso "Monitoreo de dispositivos".

Nombre	Monitoreo de dispositivos.	Código	CU-12
Actores	Dispositivo de red.		
Descripción	Permite la obtención del estado del hardware de un dispositivo. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Ninguna.		
Flujo normal 1. Se solicita la información del estado de los componentes físicos de un dispositivo por medio de SNMP. 2. Se recibe la información solicitada en el paso 1. 3. Se verifica los valores obtenidos y si amerita, se genera una alerta (referirse al CUA-2 en la Tabla 26). 4. Se almacena la información en la base de datos.			
Flujo alternativo FA2. Se recibe un error por parte del dispositivo de red al solicitar información por medio de SNMP. Se finaliza la tarea programada, reportando el error.			
Poscondiciones	Se realiza el proceso de monitoreo de un dispositivo de red.		

Tabla 33. Descripción de caso de uso “Monitoreo de RPM’s”.

Nombre	Monitoreo de RPM’s.	Código	CU-13
Actores	Dispositivo de red.		
Descripción	Permite la obtención del estado de los RPM’s descubiertos en los sistemas lógicos de un dispositivo. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Se ha realizado el proceso de descubrimiento de sistemas lógicos (referirse al CU-08, en la Tabla 28).		
Flujo normal 1. Se solicita la información del estado de los RPM’s descubiertos en los sistemas lógicos de un dispositivo por medio de SNMP. 2. Se recibe la información solicitada en el paso 1. 3. Se verifica los valores obtenidos y si amerita, se genera una alerta (referirse al CUA-2 en la Tabla 26). 4. Se almacena la información en la base de datos.			
Flujo Alternativo FA2. Se recibe un error por parte del dispositivo de red al solicitar información por medio de SNMP. Se finaliza la tarea programada, reportando el error.			
Poscondiciones	Se realiza el proceso de monitoreo de los RPM’s.		

Tabla 34. Descripción de caso de uso “Monitoreo de tablas de rutas”.

Nombre	Monitoreo de tablas de rutas.	Código	CU-14
Actores	Dispositivo de red.		
Descripción	Permite la obtención de las tablas de rutas de las Routing Instances descubiertas en los sistemas lógicos de un dispositivo. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Se ha realizado el proceso de descubrimiento de routing instances (referirse al CU-09, en la Tabla 29).		
Flujo normal 1. Se abre una conexión por NETCONF al dispositivo.			

Continuación Tabla 34.

Flujo normal 2. Se solicita la tabla de rutas de cada Routing Instance de tipo VRF descubierto en cada sistema lógico. 3. Se cierra la conexión establecida en el paso 1. 4. Se almacena la información en la base de datos.	
Flujo alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.	
Poscondiciones	Se realiza el proceso de obtención de las tablas de rutas de las Routing Instances de tipo VRF.

Tabla 35. Descripción de caso de uso “Monitoreo de tablas de direcciones MAC”.

Nombre	Monitoreo de tablas de direcciones MAC.	Código	CU-15
Actores	Dispositivo de red.		
Descripción	Permite la obtención de las tablas de direcciones MAC asociadas a las interfaces de red en los sistemas lógicos de un dispositivo. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Se ha realizado el proceso de descubrimiento de interfaces de red (referirse al CU-10, en la Tabla 30).		
Flujo normal 1. Se solicita la tabla de direcciones MAC de las interfaces de red por medio de SNMP descubiertas en cada sistema lógico. 2. Se recibe la información solicitada en el paso 1. 3. Se almacena la información en la base de datos.			
Flujo alternativo FA2. Se recibe un error por parte del dispositivo de red al solicitar información por medio de SNMP. Se finaliza la tarea programada, reportando el error.			
Poscondiciones	Se obtiene la tabla de direcciones MAC de las interfaces de red de cada sistema lógico del dispositivo de red.		

Tabla 36. Descripción de caso de uso “Monitoreo de interfaces de red”.

Nombre	Monitoreo de interfaces de red.	Código	CU-16
Actores	Dispositivo de red.		
Descripción	Permite la obtención del estado de las interfaces de red en los sistemas lógicos de un dispositivo, así como el ancho de banda utilizado. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		
Requerimientos	El dispositivo debe existir dentro del sistema.		
Precondiciones	Se ha realizado el proceso de descubrimiento de interfaces de red (referirse al CU-10, en la Tabla 30).		
Flujo normal 1. Se solicita la información del estado de las interfaces de red descubiertas en los sistemas lógicos de un dispositivo por medio de SNMP. 2. Se recibe la información solicitada en el paso 1. 3. Se verifica los valores obtenidos y si amerita, se genera una alerta (referirse al CUA-2 en la Tabla 26). 4. Se almacena la información en la base de datos. 5. Se solicita la información del ancho de banda utilizado en las interfaces de red descubiertas en los sistemas lógicos de un dispositivo por medio de SNMP. 6. Se recibe la información solicitada en el paso 5. 7. Se almacena la información en la base de datos.			
Flujo alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error. FA5. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.			
Poscondiciones	Se realiza el monitoreo de las interfaces de red.		

Tabla 37. Descripción de caso de uso “Monitoreo de LSP’s”.

Nombre	Monitoreo de LSP’s.	Código	CU-17
Actores	Dispositivo de Red.		
Descripción	Permite la obtención del estado de los LSP’s en los sistemas lógicos de un dispositivo, así como el ancho de banda utilizado. Esta rutina se ejecuta como parte de una tarea programada (referirse al CUA-01 en la Tabla 25).		

Continuación Tabla 37.

Requerimientos	El dispositivo debe existir dentro del sistema.
Precondiciones	Se ha realizado el proceso de descubrimiento de LSP's (referirse al CU-11, en la Tabla 31).
Flujo normal 1. Se solicita la información del estado de los LSP's descubiertos en los sistemas lógicos de un dispositivo por medio de SNMP. 2. Se recibe la información solicitada en el paso 1. 3. Se verifica los valores obtenidos y si amerita, se genera una alerta (referirse al CUA-2 en la Tabla 26). 4. Se almacena la información en la base de datos. 5. Se solicita la información del ancho de banda utilizado en los LSP's descubiertos en los sistemas lógicos de un dispositivo por medio de SNMP. 6. Se recibe la información solicitada en el paso 5. 7. Se almacena la información en la base de datos.	
Flujo Alternativo FA1. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error. FA5. No se logra establecer una conexión por NETCONF al dispositivo. Se finaliza la tarea programada reportando el error.	
Poscondiciones	Se realiza el monitoreo de los LSP's.

Tabla 38. Descripción de caso de uso "Configuración de Routing Instances".

Nombre	Configuración de VRF's.	Código	CU-18
Actores	Ingeniero de NOC.		
Descripción	Caso de uso para representar la configuración de una Routing Instance. La configuración de una Routing Instance de tipo VRF, incluye si se verificará su tabla de rutas, la prioridad de la instalación asociada, si se encuentra activo el monitoreo.		
Requerimientos	Parámetros para la configuración de la Routing Instance de tipo VRF.		
Precondiciones	La secuencia de descubrimiento de VRF's se ha realizado (referise al CU-09 en la Tabla 29). Usuario de tipo ingeniero de NOC autenticado al sistema. El usuario de tipo ingeniero de NOC tiene permisos para configurar Routing Instances.		

Continuación Tabla 38.

Flujo normal 1. El ingeniero de NOC accede a la vista de configuración de VRF's. 2. El ingeniero de NOC busca en la lista de VRF's la que desea configurar y hace click en el botón "Configuración". 3. En el formulario de configuración, el ingeniero coloca los valores de configuración deseados. 4. El ingeniero de NOC hace click en el botón "Guardar". 5. Los cambios a la configuración de la VRF son guardados de forma exitosa.
Flujo alternativo FA4. El ingeniero de NOC hace click en el botón "Cancelar". Los cambios realizados a la configuración de la VRF no son guardados. Continúa en el paso 1 del Flujo normal. FA5. Ocurre un problema durante el proceso de almacenamiento de los cambios de la configuración de la VRF. Se notifica el inconveniente. Continúa en el paso 4 del Flujo normal.

Poscondiciones	El ingeniero de NOC realiza cambios en la configuración de una VRF.

Tabla 39. Descripción de caso de uso "Configuración de RPM's".

Nombre	Configuración de RPM's.	Código	CU-19
Actores	Ingeniero de NOC.		
Descripción	Caso de uso para representar la configuración de un RPM. La configuración incluye el tipo de RPM y si se encuentra activo el monitoreo del RPM.		
Requerimientos	Parámetros para la configuración del RPM.		
Precondiciones	La secuencia de descubrimiento de RPM's se ha realizado (referise al CU-8 en la Tabla 28). Usuario de tipo ingeniero de NOC autenticado al sistema. El usuario de tipo ingeniero de NOC tiene permisos para configurar RPM's.		

Flujo normal 1. El ingeniero de NOC accede a la vista de configuración de RPM's. 2. El ingeniero de NOC busca en la lista de RPM's el que desea configurar y hace click en el botón "Configuración". 3. En el formulario de configuración, el ingeniero de NOC coloca los valores de configuración deseados. 4. El ingeniero de NOC hace click en el botón "Guardar". 5. Los cambios a la configuración del RPM son guardados de forma exitosa.

Continuación Tabla 39.

Flujo alternativo FA4. El ingeniero de NOC hace click en el botón “Cancelar”. Los cambios realizados a la configuración del RPM son descartados. Continúa en el paso 1 del Flujo normal. FA5. Ocurre un problema durante el proceso de almacenamiento de los cambios de la configuración del RPM. Se notifica el inconveniente. Continúa en el paso 4 del Flujo normal.

Poscondiciones	El ingeniero de NOC realiza cambios en la configuración de un RPM.

Tabla 40. Descripción de caso de uso “Configuración de interfaces de red”.

Nombre	Configuración de interfaces de red.	Código	CU-20
Actores	Ingeniero de NOC.		
Descripción	Caso de uso para representar la configuración de una interfaz de red. La configuración permite definir si se encuentra activo el monitoreo del tráfico de la interfaz de red y de las direcciones MAC.		
Requerimientos	Parámetros para la configuración de la interfaz de red.		
Precondiciones	La secuencia de descubrimiento de interfaces de red se ha realizado (referise al CU-10 en la Tabla 30). Usuario de tipo ingeniero de NOC autenticado al sistema. El usuario de tipo ingeniero de NOC tiene permisos para configurar interfaces de red.		

Flujo normal 1. El ingeniero de NOC accede a la vista de configuración de interfaces de red. 2. El ingeniero de NOC busca en la lista de interfaces de red la que desea configurar y hace click en el botón “Configuración”. 3. En el formulario de configuración, el ingeniero de NOCcoloca los valores de configuración deseados. 4. El ingeniero de NOC hace click en el botón “Guardar”. 5. Los cambios a la configuración de la interfaz de red son guardados de forma exitosa.
Flujo alternativo FA4. El ingeniero de NOC hace click en el botón “Cancelar”. Los cambios realizados a la configuración de la interfaz de red son descartados. Continúa en el paso 1 del Flujo normal. FA5. Ocurre un problema durante el proceso de almacenamiento de los cambios de la configuración de la interfaz de red. Se notifica el inconveniente. Continúa en el paso 4 del Flujo normal.

Poscondiciones	El ingeniero de NOC realiza cambios en la configuración de una interfaz de red.

Tabla 41. Descripción de caso de uso “Configuración de LSP’s”.

Nombre	Configuración de LSP’s.	Código	CU-21
Actores	Ingeniero de NOC.		
Descripción	Caso de uso para representar la configuración de un LSP. La configuración permite definir si se encuentra activo el monitoreo del tráfico del LSP.		
Requerimientos	Parámetros para la configuración del LSP.		
Precondiciones	La secuencia de descubrimiento de LSP’s se ha realizado (referise al CU-11 en la Tabla 31). Usuario de tipo ingeniero de NOC autenticado al sistema. El usuario de tipo ingeniero de NOC tiene permisos para configurar LSP’s.		
Flujo normal 1. El ingeniero de NOC accede a la vista de configuración de LSP’s. 2. El ingeniero de NOC busca en la lista de LSP’s el que desea configurar y hace click en el botón “Configuración”. 3. En el formulario de configuración, el ingeniero coloca los valores de configuración deseados. 4. El ingeniero de NOC hace click en el botón “Guardar”. 5. Los cambios a la configuración del LSP son guardados de forma exitosa.			
Flujo alternativo FA4. El Ingeniero de NOC hace click en el botón “Cancelar”. Los cambios realizados a la configuración del LSP son descartados. Continúa en el paso 1 del Flujo normal. FA5. Ocurre un problema durante el proceso de almacenamiento de los cambios de la configuración del LSP. Se notifica el inconveniente. Continúa en el paso 4 del Flujo normal.			
Poscondiciones	El ingeniero de NOC realiza cambios en la configuración de un LSP.		

Tabla 42. Descripción de caso de uso “Configuración de umbrales de alerta”.

Nombre	Configuración de umbrales de alerta.	Código	CU-22
Actores	Ingeniero de NOC.		
Descripción	Caso de uso para representar la configuración de umbrales de alerta. La configuración permite definir en qué circunstancias se debe generar una alerta.		
Requerimientos	Valores de los umbrales de alerta.		
Precondiciones	Usuario de tipo ingeniero de NOC autenticado al sistema. El usuario de tipo ingeniero de NOC tiene permisos para configurar umbrales de alerta.		

Continuación Tabla 42.

Flujo normal 1. El ingeniero de NOC accede a la vista de configuración de umbrales de alerta. 2. El ingeniero de NOC define en el formulario de configuración, las circunstancias que deben generar una alerta. 3. El ingeniero de NOC hace click en el botón “Guardar”. 4. Los cambios a la configuración de umbrales de alerta son guardados de forma exitosa.
Flujo alternativo FA3. El ingeniero de NOC hace click en el botón “Cancelar”. Los cambios realizados a la configuración de umbrales de alerta son descartados. Continúa en el paso 1 del Flujo normal. FA4. Ocurre un problema durante el proceso de almacenamiento de los cambios de la configuración de umbrales de alerta. Se notifica el inconveniente. Continúa en el paso 3 del Flujo normal.

Poscondiciones	El ingeniero de NOC realiza cambios en la configuración de umbrales de alerta.

Tabla 43. Descripción de caso de uso “Pruebas en Looking Glass”.

Nombre	Pruebas en Looking Glass	Código	CU-23
Actores	Ingeniero de NOC, dispositivo de red		
Descripción	Permite la ejecución de pruebas de Ping, Traceroute y BGP de forma controlada desde una interfaz de red o sistema lógico de un dispositivo de red.		
Requerimientos	Tipo de prueba a realizar. Origen desde donde se realizará la prueba. Dirección IP a la que se realizarán las pruebas.		
Precondiciones	Existen interfaces de red y sistemas lógicos asociados al Looking Glass.		

Flujo normal 1. El ingeniero de NOC accede a la vista de Looking Glass. 2. El ingeniero de NOC ingresa la dirección IP a la que desea realizar pruebas, el tipo de prueba y el origen desde dónde se realizará la prueba. 3. El ingeniero de NOC hace click en el botón “Realizar prueba”. 4. Se programa una tarea, en estado en cola, en el Scheduler de Web2py para ejecutarse de forma inmediata (referirse al CUA-01, en la Tabla 25). a. Se establece una conexión por medio de NETCONF al dispositivo de red que contiene el origen seleccionado en el que se debe ejecutar la prueba. b. Se solicita la ejecución de la prueba deseada. c. Se obtiene el resultado de la prueba solicitada en el paso 4.b.

Continuación Tabla 43.

Flujo normal d. Se cierra la conexión establecida en el paso 4.c. e. Se retorna el resultado y se finaliza la tarea. 5. Se espera a que la tarea programada finalice para desplegar el resultado.
Flujo alternativo FA4.a No se logra establecer comunicación con el dispositivo de red. Se indica el error y se finaliza la tarea. FA5 La tarea programada del paso 4 finaliza con error. Se presenta un mensaje indicando que no fue posible completar la prueba en tiempo real.

Poscondiciones	Se realiza pruebas en tiempo real en el Looking Glass.

Tabla 44. Descripción de caso de uso “Ver detalle y estado de instalaciones”.

Nombre	Ver detalle y estado de instalaciones.	Código	CU-24
Actores	Ingeniero de NOC.		
Descripción	Permite la visualización del detalle de una instalación y del estado de la misma.		
Requerimientos	Ninguno.		
Precondiciones	La instalación se encuentra activa.		
Flujo normal 1. El ingeniero de NOC accede a la vista de instalaciones. 2. El ingeniero de NOC busca en la lista de instalaciones la que desea visualizar y hace click en el botón “Ver instalación”. 3. El ingeniero de NOC accede a la vista de detalle de instalación, donde puede visualizar el detalle y estado de la instalación.			
Flujo alternativo FA2. El ingeniero de NOC puede realizar búsquedas modificando los valores de búsqueda en los filtros disponibles. Continúa en el paso 3 del Flujo normal.			
Poscondiciones	El ingeniero de NOC visualiza el detalle y estado de una instalación.		

Tabla 45. Descripción de caso de uso “Pruebas en tiempo real sobre VPN’s”.

Nombre	Pruebas en tiempo teal sobre VPN’s.	Código	CU-25
Actores	Ingeniero de NOC, dispositivo de red		

Continuación Tabla 45.

Descripción	Permite la ejecución de pruebas controladas de Ping sobre una VPN, desde la vista de detalle de instalación, cuyo acceso se detalla en el CU-18 de la Tabla 38.
Requerimientos	Número de identificación de la instalación. Dirección IP a la que se realizarán las pruebas.
Precondiciones	La instalación se encuentra activa.
Flujo normal 1. El ingeniero de NOC hace click en el botón “Pruebas en tiempo real”. 2. El ingeniero de NOC ingresa la dirección IP a la que desea realizar pruebas en tiempo real. 3. El Ingeniero de NOC hace click en el botón “Realizar prueba”. 4. Se programa una tarea, en estado en cola, en el Scheduler de Web2py para ejecutarse de forma inmediata (referirse al CUA-01, en la Tabla 25). a. Se establece una conexión por medio de NETCONF al dispositivo de red desde el que se debe ejecutar la prueba. b. Se solicita la ejecución de pruebas de Ping desde la Routing Instance correspondiente a la dirección IP deseada. c. Se obtiene el resultado de la prueba solicitada en el paso 4.b. d. Se cierra la conexión establecida en el paso 4.c. e. Se retorna el resultado y se finaliza la tarea. 5. Se espera a que la tarea programada finalice para desplegar el resultado.	
Flujo alternativo FA4.a No se logra establecer comunicación con el dispositivo de red. Se indica el error y se finaliza la tarea. FA5 La tarea programada del paso 4 finaliza con error. Se presenta un mensaje indicando que no fue posible completar la prueba en tiempo real.	
Poscondiciones	Se realiza pruebas en tiempo real de forma controlada sobre una VPN.

Tabla 46. Descripción de caso de uso “Ver alertas activas, inactivas e históricas”.

Nombre	Ver alertas activas, inactivas e históricas.	Código	CU-26
Actores	Ingeniero de NOC.		
Descripción	Permite la visualización y búsqueda de alertas activas, inactivas e históricas en el sistema.		
Requerimientos	Valores de búsqueda de alertas.		

Continuación Tabla 46.

Precondiciones	Usuario de tipo ingeniero de NOC autenticado al sistema.
Flujo normal 1. El ingeniero de NOC accede a la vista de alertas activas, inactivas e históricas. 2. El ingeniero de NOC ingresa los valores de búsqueda de alertas. 3. El sistema filtra las alertas basado en los valores de búsqueda recibidos y presenta el resultado. 4. El ingeniero de NOC hace click en el botón “Detalles” para ver el detalle de las alertas activas, inactivas e históricas.	
Flujo alternativo FA3 El sistema no encuentra ninguna alerta con los valores de búsqueda recibidos. Se retorna una vista vacía de alertas. Se continúa en el paso 2, con la alteración de los valores de búsqueda.	
Poscondiciones	El ingeniero de NOC visualiza el listado de alertas activas, inactivas e históricas y sus detalles.

Tabla 47. Descripción de caso de uso “Auto retorno de alertas inactivas a estado activo”.

Nombre	Auto retorno de alertas inactivas a estado activo	Código	CU-27-01
Actores	Ingeniero de NOC.		
Descripción	Es una tarea programada que realiza el cambia automático del estado de una alerta inactiva a activa en el sistema. Esta tarea programada se asocia al CUA-01 en la Tabla 25.		
Requerimientos	Ninguno.		
Precondiciones	Ninguno.		
Flujo normal 1. El sistema verifica el tiempo de retorno de las alertas en estado inactivo. 2. Si el tiempo de retorno es menor a la hora y fecha actual, se realiza el cambio de estado inactivo a activo. 3. Se finaliza la tarea.			
Flujo alternativo Ninguno.			
Poscondiciones	El sistema realiza el cambio automático del estado de una alerta inactiva a activa.		

Tabla 48. Descripción de caso de uso “Marcar alerta activa como inactiva”.

Nombre	Marcar alerta activa como inactiva	Código	CU-27
Actores	Ingeniero de NOC.		
Descripción	Permite la cambiar el estado de una alerta activa a inactiva en el sistema.		
Requerimientos	Tiempo en el que la tarea retornará a estado activo.		
Precondiciones	Usuario de tipo ingeniero de NOC autenticado al sistema.		
Flujo normal 1. El ingeniero de NOC accede al detalle de una alerta activa (referirse al CU-26 en la Tabla 46). 2. El ingeniero de NOC hace click en el botón “Acknowledge”. 3. El sistema solicita al ingeniero de NOC que indique el tiempo en que la alerta regresará a estado activa (ver CU-26-01, en la Tabla 47). 4. Se notifica el cambio exitoso del estado de la alerta.			
Flujo alternativo FA3 El ingeniero de NOC hace click en el botón “Cancelar”. Los cambios son descartados y se continúa en el paso 1 del Flujo normal.			
Poscondiciones	El ingeniero de NOC cambia el estado de una alerta activa a inactiva.		

Tabla 49. Descripción de caso de uso “Mover alerta activa al historial de alertas”.

Nombre	Mover alerta activa al historial de alertas	Código	CU-28
Actores	Ingeniero de NOC.		
Descripción	Permite la cambiar el estado de una alerta activa a histórica en el sistema.		
Requerimientos	Ninguno.		
Precondiciones	Usuario de tipo ingeniero de NOC autenticado al sistema.		
Flujo normal 1. El ingeniero de NOC accede al detalle de una alerta activa (referirse al CU-26 en la Tabla 46). 2. El ingeniero de NOC hace click en el botón “Enviar a histórico”. 3. El sistema solicita al ingeniero de NOC que confirme la acción. 4. Se notifica el cambio exitoso del estado de la alerta.			
Flujo alternativo FA3 El ingeniero de NOC hace click en el botón “Cancelar”. Los cambios son descartados y se continúa en el paso 1 del Flujo normal.			
Poscondiciones	El ingeniero de NOC cambia el estado de una alerta activa a histórica.		

Tabla 50. Descripción de caso de uso “Apertura de ticket”.

Nombre	Apertura de ticket.	Código	CU-29
Actores	Ingeniero de NOC, sistema de tickets.		
Descripción	Permite la apertura de un ticket en un sistema externo, por medio de Webservices de tipo SOAP.		
Requerimientos	Valores del formulario de apertura de ticket.		
Precondiciones	Usuario de tipo ingeniero de NOC autenticado al sistema.		
Flujo normal 1. El ingeniero de NOC accede a la vista de alertas activas. 2. El ingeniero de NOC hace click en el botón “Abrir ticket” junto a la alerta asociada. 3. El ingeniero de NOC es dirigido a la vista de apertura de ticket. 4. El ingeniero de NOC ingresa los datos necesarios para la apertura de un ticket. 5. El ingeniero de NOC hace click en el botón “Enviar” 6. Se genera una tarea programada en estado en cola, y se programa en el Scheduler de Web2py para ejecutarse de forma inmediata (referirse al CUA-01, en la Tabla 25). a. Se construye el mensaje a enviar a la herramienta de tickets, en formato XML. b. Se envía el mensaje por medio de HTTPS a la herramienta de tickets. c. Se obtiene respuesta del mensaje enviado en el paso 6.b. d. Se retorna el resultado y se finaliza la tarea. 7. Se espera a que la tarea programada finalice y se notifica la creación del ticket.			
Flujo Alternativo FA5 El ingeniero de NOC hace click en el botón “Cancelar”. Los valores ingresados en el formulario de apertura de ticket son descartados. Continúa en el paso 1 del Flujo normal. FA6.c No se obtiene respuesta por parte de la herramienta de tickets y se alcanza el tiempo duración máximo de la tarea. Se finaliza la tarea programada en estado Timeout y se notifica el error de la tarea. FA6.d El resultado obtenido indica que hubo un error durante el proceso de apertura. Se finaliza la tarea programada y se notifica del error recibido.			
Poscondiciones	El ingeniero de NOC realiza la apertura de un ticket.		

Tabla 51. Descripción de caso de uso “Búsqueda de tickets”.

Nombre	Búsqueda de tickets	Código	CU-30
Actores	Ingeniero de NOC, sistema de tickets.		
Descripción	Permite la búsqueda de tickets activos en un sistema externo, por medio de Webservices SOAP, asociados a un número de instalación.		
Requerimientos	Número de instalación para la búsqueda de tickets activos.		
Precondiciones	Usuario de tipo ingeniero de NOC autenticado al sistema.		
Flujo normal 1. El ingeniero de NOC accede a la vista de alertas activas. 2. El ingeniero de NOC hace click en el botón “Buscar ticket asociado”. 3. El ingeniero de NOC es dirigido a la vista de búsqueda de tickets. 4. El ingeniero de NOC ingresa número de instalación para la búsqueda de tickets. 5. El ingeniero de NOC hace click en el botón “Enviar” 6. Se genera una tarea programada en estado en cola, y se programa en el Scheduler de Web2py para ejecutarse de forma inmediata (referirse al CUA-01, en la Tabla 25). a. Se construye el mensaje a enviar a la herramienta de tickets, en formato XML. b. Se envía el mensaje por medio de HTTPS a la herramienta de tickets. c. Se obtiene respuesta del mensaje enviado en el paso 6.b. d. Se retorna el resultado y se finaliza la tarea. 7. Se espera a que la tarea programada finalice y se retorna el listado de tickets obtenidos como resultado de la búsqueda.			
Flujo alternativo FA5 El ingeniero de NOC hace click en el botón “Cancelar”. Los valores ingresados en el formulario de búsqueda de ticket son descartados. Continúa en el paso 1 del flujo normal. FA6.c No se obtiene respuesta por parte de la herramienta de tickets y se alcanza el tiempo duración máximo de la tarea. Se finaliza la tarea programada en estado Timeout y se notifica el error de la tarea. FA6.d El resultado obtenido indica que hubo un error durante el proceso de búsqueda. Se finaliza la tarea programada y se notifica del error recibido.			
Poscondiciones	El ingeniero de NOC realiza la búsqueda de tickets activos asociados a un número de instalación.		

Tabla 52. Descripción de caso de uso “Añadir comentarios a ticket”.

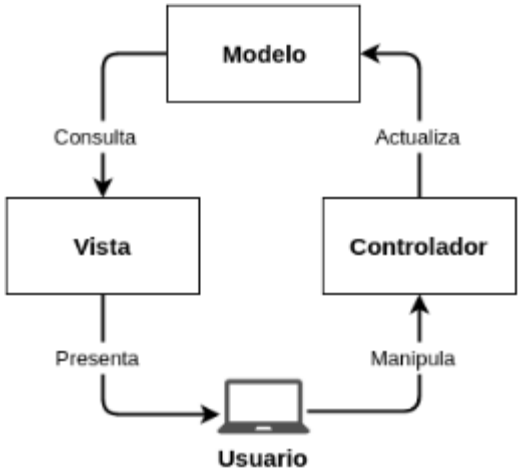
Nombre	Añadir comentarios a ticket	Código	CU-31
Actores	Ingeniero de NOC, sistema de tickets.		
Descripción	Permite la adición de comentarios a un ticket activos en un sistema externo, por medio de Webservices SOAP.		
Requerimientos	Número de ticket al cual se añadirá el comentario. Valores del formulario de comentario.		
Precondiciones	Usuario de tipo ingeniero de NOC autenticado al sistema.		
Flujo normal 1. El ingeniero de NOC accede a la vista de alertas activas. 2. El ingeniero de NOC hace click en el botón “Agregar comentario”. 3. El ingeniero de NOC es dirigido a la vista de agregar comentario a ticket. 4. El ingeniero de NOC llena los campos del formulario para agregar un comentario a un ticket. 5. El ingeniero de NOC hace click en el botón “Enviar” 6. Se genera una tarea programada en estado en cola, y se programa en el Scheduler de Web2py para ejecutarse de forma inmediata (referirse al CUA-01, en la Tabla 25). a. Se construye el mensaje a enviar a la herramienta de tickets, en formato XML. b. Se envía el mensaje por medio de HTTPS a la herramienta de tickets. c. Se obtiene respuesta del mensaje enviado en el paso 6.b. d. Se retorna el resultado y se finaliza la tarea. 7. Se espera a que la tarea programada finalice y se notifica la adición del comentario al ticket activo.			
Flujo alternativo FA5 El ingeniero de NOC hace click en el botón “Cancelar”. Los valores ingresados en el formulario para agregar comentario a ticket son descartados. Continúa en el paso 1 del Flujo normal. FA6.c No se obtiene respuesta por parte de la herramienta de tickets y se alcanza el tiempo duración máximo de la tarea. Se finaliza la tarea programada en estado Timeout y se notifica el error de la tarea. FA6.d El resultado obtenido indica que hubo un error durante el proceso de búsqueda. Se finaliza la tarea programada y se notifica del error recibido.			
Poscondiciones	El ingeniero de NOC agrega un comentario a un ticket activo.		

E. ARQUITECTURA DEL SISTEMA

La presente arquitectura tiene como objetivo mostrar a alto nivel los componentes arquitectónicos involucrados en el desarrollo de la plataforma digital, así como definir la interacción entre ellos, de manera que se pueda dar a conocer de forma específica el funcionamiento interno del sistema.

Como se menciona en las secciones anteriores, se busca que la solución sea un sistema Web. El paradigma de arquitectura utilizado es el de Modelo Vista Controlador, o MVC, que divide el software en tres partes interconectadas como se muestra en la Figura 35.

Figura 35. Colaboración típica entre las partes de la arquitectura MVC.



- Modelo: Es la parte o componente que se encarga del manejo de los datos del sistema. Este se encarga de definir de la lógica y reglas de la aplicación, mediante la encapsulación de los datos de forma independiente de la vista y del controlador.
- Vista: Este componente se encarga de la presentación de los datos o información al usuario. Puede consultar los datos del modelo, pero no aplicar cambios.
- Controlador: Este componente es el encargado de indicarle al modelo los cambios de los datos, de acuerdo a las acciones realizadas por el usuario, luego de una ejecución adecuada.

Dentro de un sistema Web con conexiones a una base de datos, el modelo es el encargado de realizar las conexiones a la base de datos y la manipulación de los datos incluidos en la misma y abstraer esta información a los otros componentes del paradigma MVC. El controlador puede verse como el encargado de reaccionar ante una solicitud HTTP, siendo disparado un método o función del controlador asociado a una dirección URL (*Uniform Resource Location*). La vista, es quien tiene la responsabilidad de construir los documentos HTML, o de otro tipo, que se retornarán como respuesta a la solicitud HTTP recibida.

1. Objetivos de la arquitectura. Los objetivos identificados de la arquitectura se presentan a continuación:

- Usabilidad: se busca una interfaz de usuario simple y entendible, que permita a los usuarios familiarizarse rápidamente con el sistema. Se busca que los módulos desarrollados brinden la máxima funcionalidad posible, de manera que pueda formar parte de las herramientas utilizadas en las labores diarias dentro de un NOC y facilite el trabajo del NOC así como en la mejora de la proactividad.
- Portabilidad: al desarrollar una herramienta Web, el usuario podrá utilizarla en diferentes ambientes de ejecución, es decir, diferentes navegadores Web ejecutados en diversos sistemas operativos.
- Funcionalidad: la funcionalidad del sistema no debe verse afectada o alterada por el cambio del ambiente de ejecución del lado del usuario.
- Centralización de información: la herramienta busca centralizar la información del estado de la red MPLS del ISP y del estado de los servicios VPN Capa 3 brindados por el ISP. Esta centralización busca ampliarse mediante el futuro desarrollo de módulos que abarquen otros dispositivos de red y otro tipo de servicios brindados por el ISP.
- Modularidad: se consigue mediante la definición de funciones independientes del sistema agrupadas de módulos, que cumplan con los requerimientos identificados. Esta definición modular facilita el trabajo de mantenimiento, nuevos desarrollos y cambios de funcionalidades o requerimientos.
- Reutilización: logrado a través de la reutilización de código, por medio de la definición de métodos,funciones y clases con objetivos bien definidos, que puedan ser utilizados en otros módulos o desarrollos.

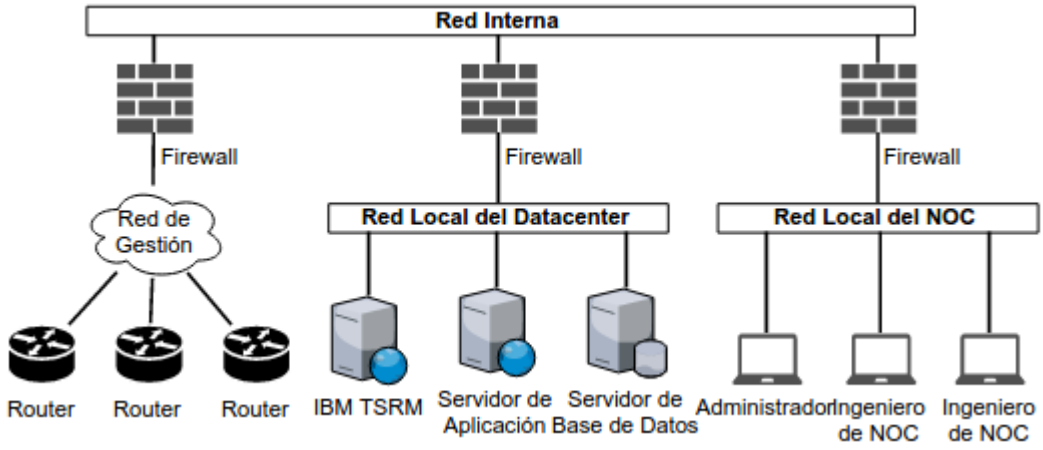
2. Despliegue del sistema. El despliegue del sistema tiene como objetivo mostrar los elementos físicos que componen el sistema, así como las interacciones que existen entre estos. En la Figura 36 se muestra el Diagrama de despliegue del sistema.

Es necesario describir algunos de los componentes de la Figura 36, para una mejor comprensión del Diagrama:

- Servidor de aplicación: Servidor Web Virtual que se encarga de atender y procesar las solicitudes HTTP. El servidor HTTP es Apache y se encuentra asociado a Python por medio de WSGI (*Web Server Gateway Interface*). También cuenta con un número de procesos denominados Workers, que son empleados para la ejecución de tareas programadas.
- Servidor de base de datos: Es un servidor Virtual que provee el almacenamiento de datos, utilizando PostgreSQL como motor de base de datos.

- IBM TSRM: Servidor aplicativo que provee una API de tipo SOAP, accesible por el puerto 443, permitiendo realizar funciones de búsqueda, creación y modificación de Tickets.
- Datacenter: es una instalación especializada que alberga equipos de red y servidores físicos o virtuales, utilizados con propósitos empresariales.
- Red de gestión: es una red fuera de banda, o secundaria, que se utiliza para asegurar la conectividad a los dispositivos de una red, aún cuando se tienen problemas dentro de la red del ISP.
- Administrador/ingeniero de NOC: representa a los usuarios y el punto de acceso que tendrán desde la red asignada a los operadores de NOC.
- Red interna: o red empresarial, representa la red administrativa del ISP, puede tratarse de una o varias VPN utilizadas con fines de manejo del negocio.

Figura 36. Diagrama de despliegue del sistema.



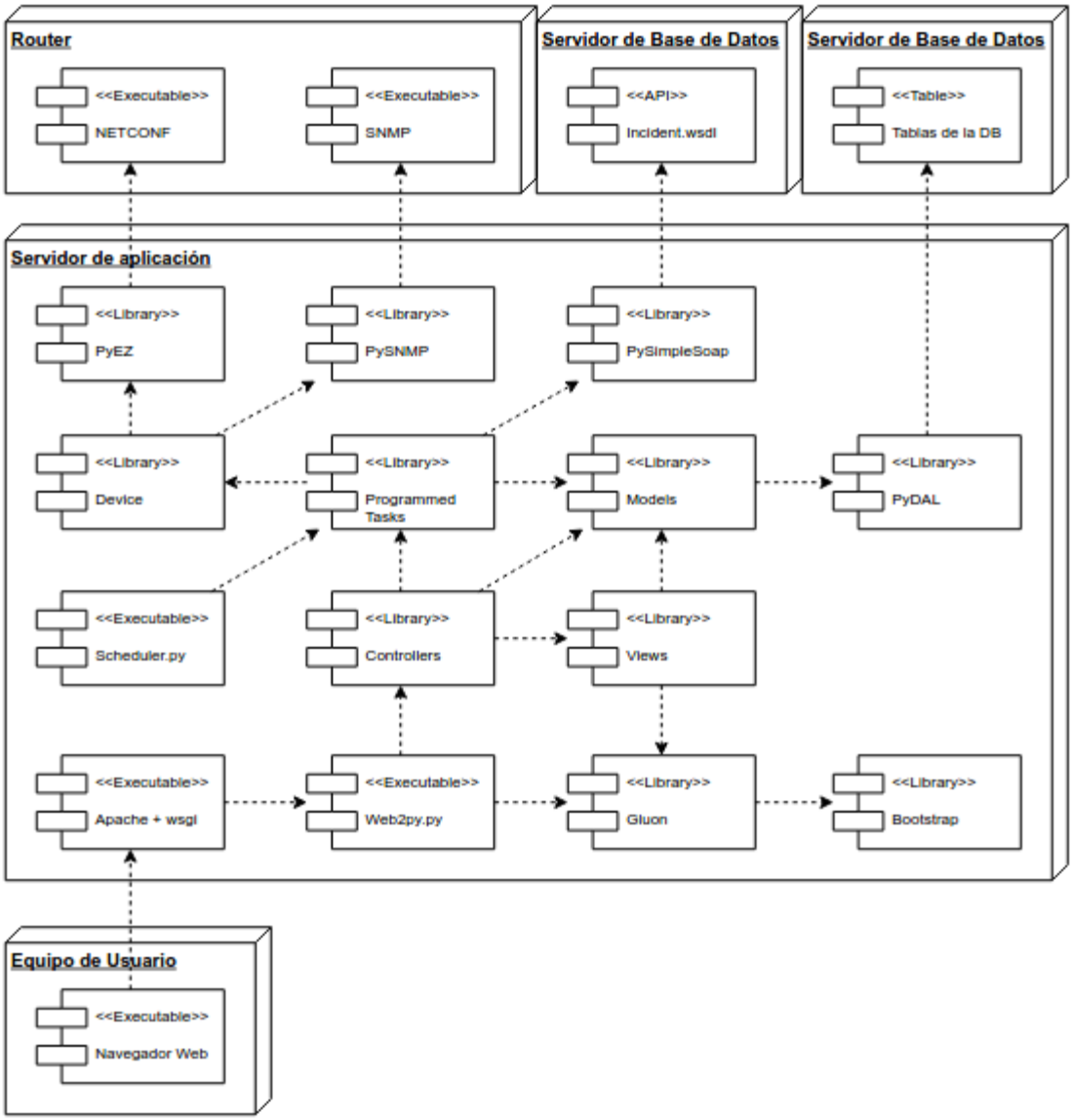
3. Componentes del sistema. En un diagrama de componentes se busca representar la organización de los componentes de software, identificando las dependencias y agrupaciones. Es importante mencionar que en el diagrama de la Figura 37 se presentan algunos de los componentes más importantes y otros agrupados para su simplificación. Los componentes del contenidos en el servidor de aplicación del diagrama de la Figura 37 son descritos a continuación:

- PySNMP: Librería que contiene rutinas para la ejecución de operaciones de SNMP. La información de esta librería se amplía en la sección de Protocolo Simple de Administración de Red del Marco Teórico.
- PyEZ: es un microframework de Juniper, que contiene métodos para la comunicación por medio de NETCONF con routers Juniper, empleados para la administración de configuración y ejecución de rutinas o funciones del lado del router. Este es ampliado en la sección de Protocolo de Configuración de Red del Marco Teórico.

- PySimpleSoap: Es una librería simple y liviana de SOAP para Python, empleada para la creación interfaces de servicios web cliente y servidor. Se amplía ésta en la sección de Herramientas de Desarrollo, encontrada más adelante.
- Device: es una librería que comprende un conjunto de rutinas para comunicarse a un router con el objetivo de sustraer información del estado del router y sus componentes, información de configuración y resultado de tareas. Esta librería contiene una interfaz que deben heredar las clases definidas para conexión con dispositivos de diversos fabricantes.
- Programmed Tasks: agrupa diferentes funciones definidas para la realización de tareas de monitoreo, sustracción de configuraciones, manejo de tickets, entre otros. La finalidad de esta librería es abstraer funciones comunes, para que puedan ser llamadas por el Scheduler de forma síncrona, variando únicamente en la programación del Scheduler los parámetros recibidos.
- Scheduler: es uno de los componentes de Web2py, cuyo funcionamiento se asemeja al de un CRON, que es un calendarizador de tareas de los sistemas operativos Unix y Unix-like, el cual se ejecuta periódicamente en intervalos determinados. La utilidad de *Scheduler* de web2py, provee una API con diferentes métodos que facilitan la administración de las tareas. Web2py se amplía en la sección de Herramientas de Desarrollo, encontrada más adelante.
- PyDAL: es una librería que se incluye como parte del framework Web2py, que funciona como una Capa de Abstracción de la Base de Datos o DAL (*Database Abstraction Layer*). Se amplía el funcionamiento de PyDAL en la sección de Herramientas de Desarrollo, encontrada más adelante.
- Models: corresponde a la parte modelo dentro del paradigma MVC. En esta librería se incluyen todas las definiciones de los objetos necesarios dentro del sistema. El esquema de la base de datos se presenta en la siguiente sección, Diseño de la Base de Datos.
- Bootstrap: es un framework de código abierto orientado al desarrollo de la capa de presentación en aplicaciones y sitios Web, comúnmente conocidos como *frameworks frontend*. Este framework se incluye como parte de Web2py. Bootstrap define estilos para todos los componentes HTML y componentes adicionales de Javascript en forma de Plugins de JQuery.
- Gluon: contiene diversas librerías de Web2py necesarias para el funcionamiento de Web2py. Estas librerías incluyen herramientas para el manejo de caché, plantillas HTML, serializadores y un manejador de errores.
- Views: corresponde a la parte de vista dentro del paradigma MVC. Esta librería contiene diversas plantillas de documentos HTML, JSON y XML, que son definidas con un lenguaje de plantillas propio de Web2py, las cuales son renderizadas en tiempo de ejecución y luego retornadas como parte del *HTTP Response*.
- Controllers: corresponde a la parte de controlador del paradigma MVC. En esta librería se incluyen los métodos o funciones que son ejecutados como respuesta a una solicitud HTTP request por parte del navegador Web de un cliente. Usualmente existe un método que corresponde a una dirección URL que es parte del *HTTP request*.

- Web2py.py: es el archivo principal de Web2py, este se encarga del manejo de la comunicación con el servidor HTTP y los demás elementos definidos dentro de Web2py.
- Apache y WSGI: Apache es un servidor HTTP, el proyecto es desarrollado como código abierto. Apache cuenta con diferentes módulos desarrollados, dentro de los que podemos mencionar a mod_wsgi, que implementa el estándar WSGI (*Web Server Gateway Interface*), en cuya especificación define la forma en que un servidor Web se comunica con aplicaciones Web.

Figura 37. Diagrama de componentes del sistema.



4. Diseño de la base de datos. A continuación se muestra el Diagrama de base de datos, contenido en las Figuras 38 y 39. Este ayuda a una mejor comprensión de la forma en que se organiza la información manejada por el sistema. El diseño utilizado para la base de datos es de tipo relacional. El diseño es bastante simple, pues se busca que pueda ser implementado por medio del DAL de Python sin necesidad de escribir código SQL.

Figura 38. Diagrama de diseño de la base de datos relacional - parte 1.

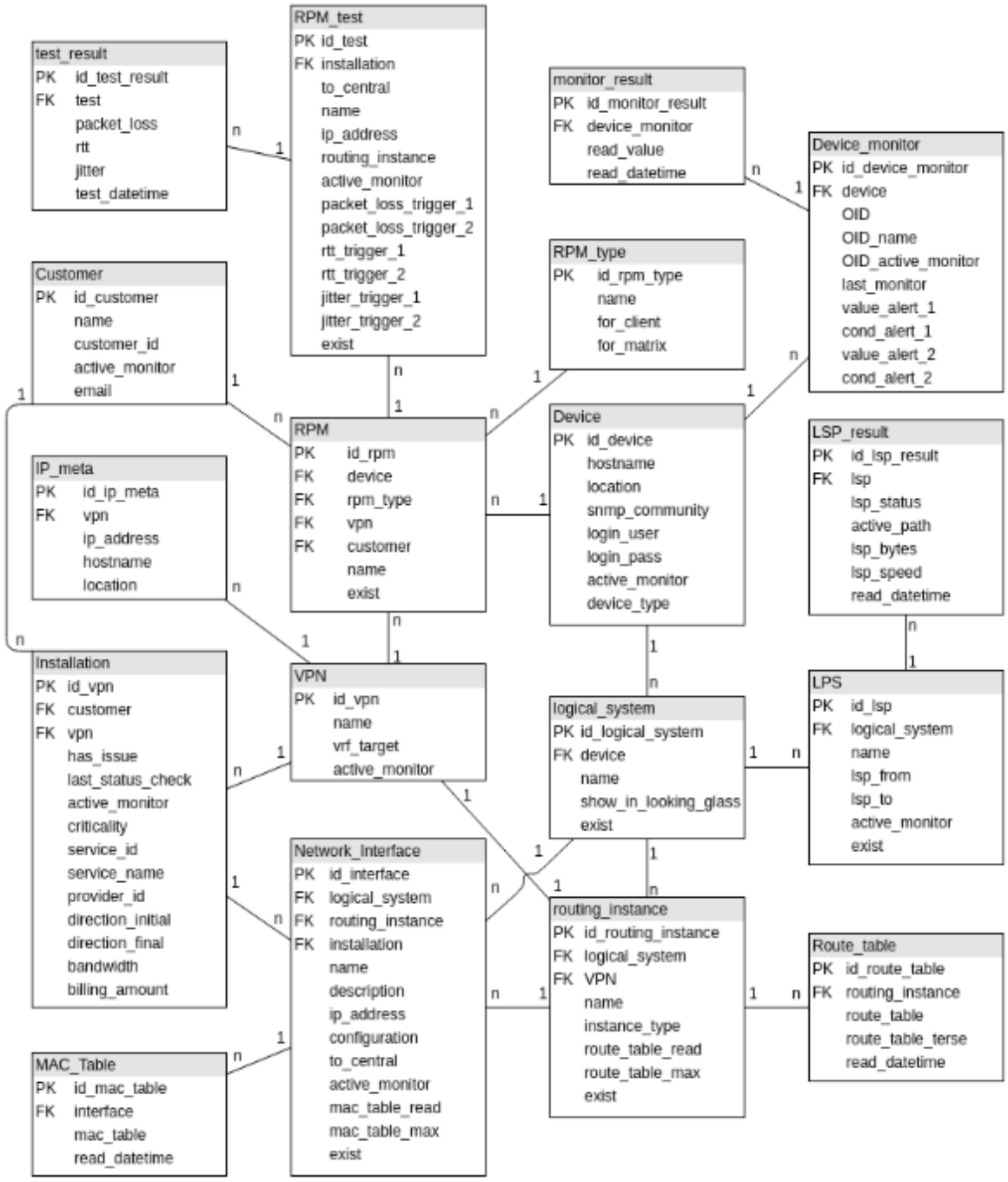
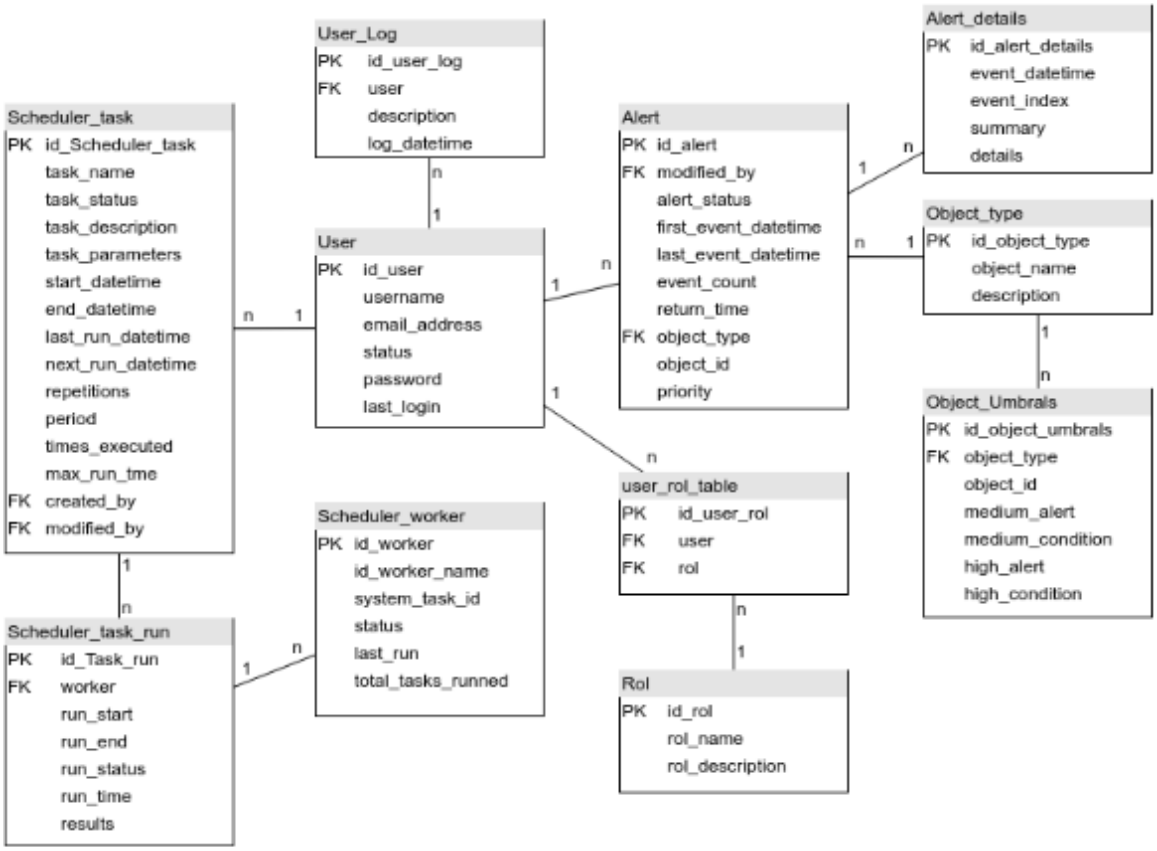


Figura 39. Diagrama de diseño de la base de datos relacional - parte 2.



F. HERRAMIENTAS DE DESARROLLO

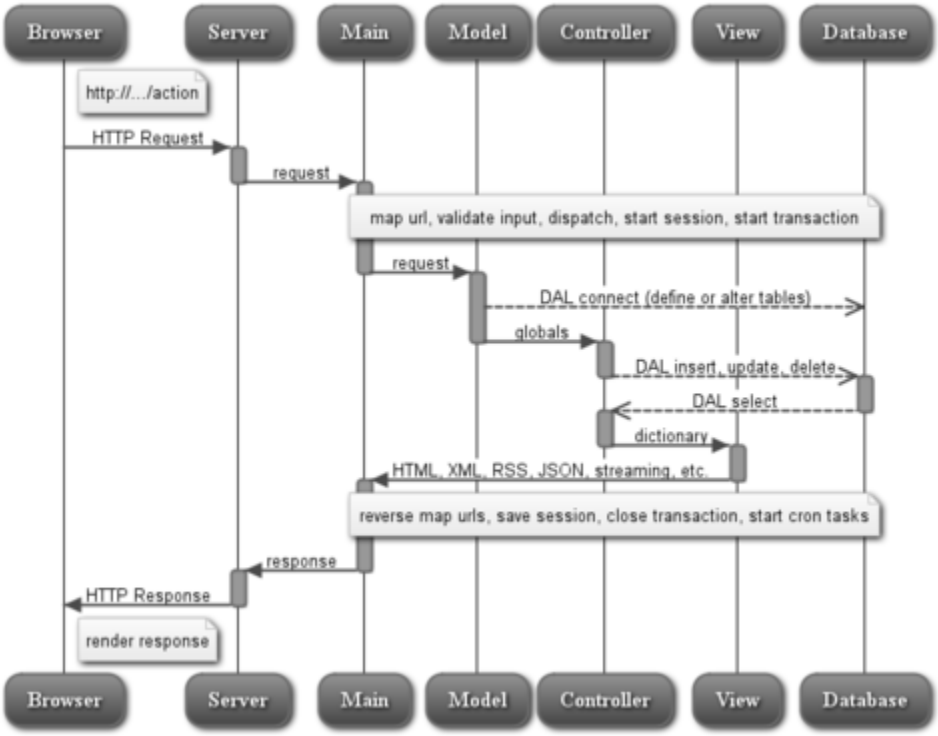
Las principales herramientas de desarrollo empleadas para el proyecto son listadas a continuación. Algunas de estas fueron presentadas anteriormente en el Capítulo IV. Marco Teórico.

1. Web2py. Es un framework para desarrollo de aplicaciones web seguras con conexiones a base de datos; este framework es de código abierto desarrollado en Python y cumple con ser un framework full-stack, es decir, que contiene los componentes necesarios para el desarrollo de aplicaciones web completamente funcionales.

El diseño de Web2py está orientado para guiar al desarrollador a seguir buenas prácticas de ingeniería de software, como el uso del patrón Modelo-Vista-Controlador (MVC) que separa la representación de los datos (modelo) de la presentación de los mismos (vista) y también separa la lógica y el flujo de trabajo (controlador). También las buenas prácticas se aplican a aspectos de seguridad, facilitando que la introducción de vulnerabilidades al desarrollar aplicaciones sea evadida durante el desarrollo.

Uno de los aspectos sobresalientes de Web2py frente a otros frameworks, es que cuenta con un DAL (Data Abstraction Layer) en lugar de un ORM (*Object-Relational Mapping*). Un DAL se encarga de generar código SQL (*Structured Query Language*) de forma dinámica, de modo que el desarrollador no debe hacerlo. Sin embargo, el DAL permite la ejecución de código SQL directamente escrito por el desarrollador y retornar los resultados en una forma estructurada, sin necesidad de contar con un mapeo de las tablas de la base de datos a objetos como normalmente funciona un ORM. En la siguiente Figura, se describe el flujo de trabajo de una solicitud realizada a una dirección URL de una aplicación de Web2py.

Figura 40. Flujo de trabajo de una solicitud en Web2py



Los elementos o componentes involucrados en el flujo de trabajo del anterior diagrama se describen a continuación:

- El servidor puede ser incorporado (aunque sólo es recomendado durante el desarrollo) o un tercero, como nginx o Apache.
- Main es la aplicación principal WSGI, que se encarga de las tareas comunes y envuelve la aplicación del usuario. Se encarga del manejo de cookies, sesiones, transacciones, enrutamiento de URL y enrutamiento inverso, y la administración de direcciones o dispatching. Puede administrar y realizar transmisiones de archivos estáticos si el servidor web no se ha configurado para hacerlo.
- Los elementos modelo, controlador y vista conforman la aplicación de usuario. Web2py permite alojar diferentes aplicaciones en una misma instancia.

- Las flechas con líneas punteadas representan la comunicación con el(los) motor(es) de base de datos.
- Es posible registrar tareas recurrentes, por medio de cron, para que se ejecuten según un programa en un momento determinado y/o al finalizar ciertas acciones. De esta forma es posible correr tareas largas y que requieren un uso intensivo del hardware en segundo plano sin que la navegación se torne más lenta.

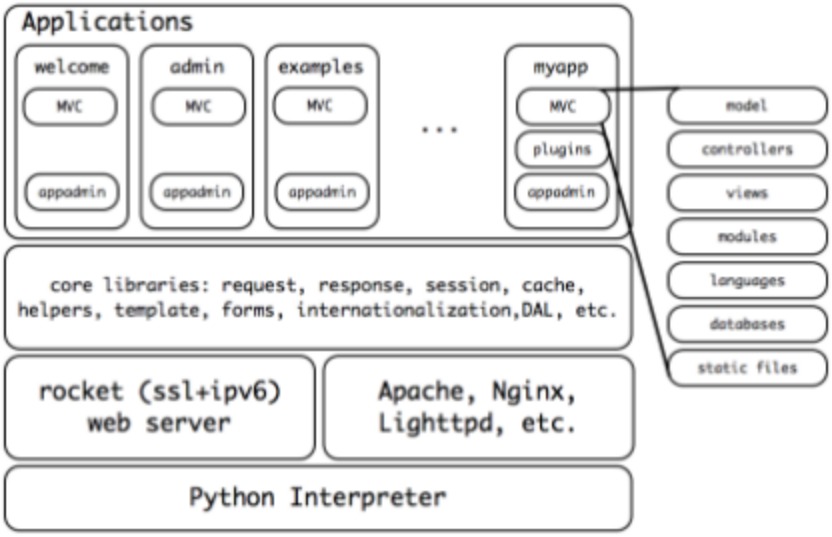
La estructura de Web2py está formada por los componentes descritos a continuación:

- Librerías: proveen las funcionalidades del núcleo de Web2py y se puede acceder a ellas en forma programática.
- Servidor web: el servidor web WSGI Rocket.
- La aplicación admin: se usa para crear, diseñar y administrar las demás aplicaciones. Esta provee de un completo Entorno de Integrado de Desarrollo (IDE, *Integrated Development Environment*) para la creación de aplicaciones. Además incluye otras funcionalidades, como el entorno de pruebas para interfaz web y la consola shell para navegador.
- La aplicación examples: contiene documentación y ejemplos interactivos. examples es un clon del sitio web oficial, e incluye la documentación epydoc.
- La aplicación welcome: la plantilla de andamiaje para cualquier otra aplicación. Por defecto, incluye un menú desplegable escalonado CSS y un sistema de autenticación.

En la Figura 41, en la siguiente página, se ilustra la estructura aproximada de Web2py. En la parte inferior se puede encontrar el intérprete. Si nos desplazamos hacia arriba encontramos el servidor web (rocket), las librerías, y las aplicaciones. Cada aplicación cuenta con su propio diseño MVC (modelos, controladores, vistas, idiomas, bases de datos, y archivos estáticos). Cada aplicación incluye su propia interfaz de administración de la base de datos (*appadmin*). Cada instancia de web2py incluye por defecto tres aplicaciones: welcome, admin, y examples, descritos anteriormente.

Web2py sirve cada solicitud de HTTP en un Thread asignado, los cuales son reciclados para eficiencia. Sin embargo, por seguridad, el servidor Web define un tiempo máximo en el que debe atenderse una solicitud de HTTP, por lo que las acciones no pueden tardar demasiado tiempo, de lo contrario se retorna un error de timeout al emisor de la solicitud recibida. La forma correcta de ejecutar tareas que demoran mucho tiempo es mediante el uso de tareas en segundo plano, o background, por medio del uso de CRON, colas de tareas y el Scheduler de Web2py. Es importante mencionar que Web2py implementa su propio CRON, no se refiere a la funcionalidad de los sistemas operativos Unix y Unix-like.

Figura 41. Estructura de Web2py



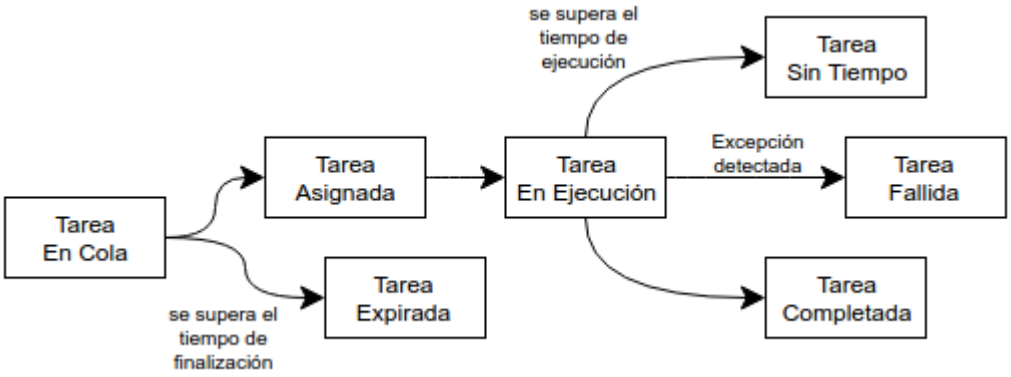
El CRON de Web2py, provee a las aplicaciones web la habilidad de ejecutar tareas en determinados momentos de forma independiente. Para cada aplicación, se define un archivo crontab, que utiliza la misma sintaxis del CRON de Unix, facilitando el manejo independiente de cada aplicación web. Aunque el CRON de Web2py puede resultar útil para la ejecución de tareas en segundo plano, no siempre es la mejor solución. Web2py permite la ejecución de cualquier método o función desde una consola o terminal, siempre que se encuentre dentro de un controlador. De esta forma, se puede implementar una cola que invoque la ejecución de determinadas funciones.

Finalmente, la solución principal, y más elegante, que ofrece Web2py para la ejecución de tareas en segundo plano, es la implementación de un Scheduler. El scheduler de Web2py es un componente que se encuentra por defecto como parte de Web2py. El Scheduler de Web2py provee una API, que provee funciones para la administración de las tareas programadas. Las ventajas que brinda son las siguientes:

- Estandarización para la creación, calendarización y monitoreo de las tareas.
- No es un sólo proceso que se puede configurar para la ejecución en segundo plano, más bien se trata de un conjunto de procesos, denominados workers.
- Los workers pueden ser monitoreados, además el estado de estos y sus tareas es almacenado en la base de datos.

Las tareas del Scheduler de Web2py pueden ser programadas desde la consola, un método del controlador, e incluso desde la aplicación admin. El ciclo de vida de una tarea programada se muestra en la Figura 42. Es importante notar que de acuerdo a los parámetros pasados durante la creación de una tarea, al completarse la ejecución puede volverse a encolar para una nueva ejecución.

Figura 42. Ciclo de vida de una tarea programada en el Scheduler de Web2py.



Dentro de Web2py se incluyen diferentes librerías adicionales, que ayudan a acelerar el desarrollo de aplicaciones web. Dentro de sus librerías, se incluye Bootstrap, un framework para desarrollo de frontend, es decir, la presentación en aplicaciones y sitios Web. Bootstrap se compone de un conjunto de elementos de hojas de estilo o CSS (*Cascade Style Sheet*) y componentes de Javascript. Bootstrap incluye como parte de sus componentes de Javascript componentes de jQuery, una librería de Javascript ampliamente utilizada, y manejo de eventos AJAX (acrónimo de *Asynchronous Javascript XML*).

2. PySNMP. Esta es una librería de Python que contiene rutinas para la ejecución de operaciones de SNMP. La información de esta librería se amplía en la sección de Protocolo Simple de Administración de Red del Marco Teórico.

3. Junos PyEZ. Junos PyEZ, o simplemente PyEZ, es un microframework desarrollado por Juniper. Este contiene librerías que ofrecen la capacidad de comunicación con dispositivos Juniper, por medio de NETCONF, principalmente para la administración de configuración de los equipos. Con PyEZ también es posible la ejecución de varios de los comandos disponibles en la CLI de los dispositivos Juniper. La información referente a PyEZ es ampliada en la sección de Protocolo de Configuración de Red, en el Marco Teórico.

4. PySimpleSoap. Es una librería simple y liviana de SOAP para Python, empleada para la creación interfaces de servicios web cliente y servidor. Esta librería es desarrollada como un proyecto open source, y sostiene la filosofía de mantener código simple, de fácil uso y que soporte la mayoría de las funcionalidades comunes. Dentro de los objetivos perseguidos por los desarrolladores sobresalen los siguientes:

- Simplicidad: la librería contiene una poca cantidad de líneas de código y se enfoca en funcionalidades concretas para una fácil implementación y mejoras. Los parámetros y valores retornados son estructuras básicas de Python, como listas y diccionarios.

- Flexibilidad: en su base, soporta diferentes dialectos de SOAP utilizado por diversos servidores como *Java Axis, .NET, WCF, JBoss*, entre otros. Adicionalmente, permite la fácil implementación de nuevos dialectos SOAP.
- Compatibilidad: provee una API estable, que mantiene compatibilidad con versiones anteriores, permitiendo que una actualización no afecte el desarrollo realizado.

5. NVD3. NVD3.js es un proyecto que ofrece facilidad para la creación de diversos tipos de gráficos interactivos, manejados del lado del navegador Web. Esta librería, escrita en Javascript, se basa en el proyecto D3.js (su nombre proviene del inglés, *Data Driven Documents*), que es una librería para la visualización de datos utilizando estándares de Web.

6. PostgreSQL. PostgreSQL es un sistema para la administración de bases de datos relacionales. El proyecto es de tipo open source, y es mantenido por el departamento de Ciencias de la Computación de la Universidad de Berkeley. Este soporta una gran cantidad de estándares de SQL, algunas características disponibles sólo en algunos sistemas de bases de datos comerciales y otras características adicionales propias de PostgreSQL.

Debido a la licencia con que es distribuido PostgreSQL, se puede utilizar, realizar modificaciones al código y distribuir para diferentes propósitos. Esto también permite que sea soportado por diversos lenguajes de programación, para los que existe una gran cantidad de drivers o controladores que facilitan la comunicación a la base de batos. Esta base de datos, por su robustez y flexibilidad, así como el alto rendimiento en soluciones de alta demanda, suele ser referente por diferentes frameworks desarrollados también bajo la filosofía de open source. De hecho, frameworks populares escritos en Python como Django y Web2py, recomiendan el uso de este motor de base de datos, en conjunto con el driver *psycopg2*, distribuido gratuitamente bajo la Licencia Pública General de GNU.

7. Bitbucket. Bitbucket es un servidor web para hosting y control de versiones de proyectos. Este permite utilizar git como sistema de control de versiones. Este proyecto fue desarrollado utilizando el framework de Python llamado Django. Bitbucket es uno de los servidores para hosting y control de versiones más utilizado junto a GitHub. Tanto Bitbucket como GitHub ofrecen planes gratuitos, pagados y soluciones empresariales.

VII. RESULTADOS

Los resultados presentados son vistas del funcionamiento del Sistema desarrollado. Cabe mencionar que las direcciones URL siguen el formato */<app_name>/<controller_name>/<function_name>/ <parameters>*, para facilitar la organización y mantenimiento del proyecto.

A. PÁGINA INICIAL Y AUTENTICACIÓN DE USUARIO

La página inicial se muestra en la Figura 43. Cuando un usuario no autenticado accede a cualquier otra URL, automáticamente será redirigido a la vista de autenticación de la Figura 44, a la que también puede acceder en el menú “Log in” en la esquina superior derecha de la pantalla inicial.

Figura 43. Página inicial del sistema.

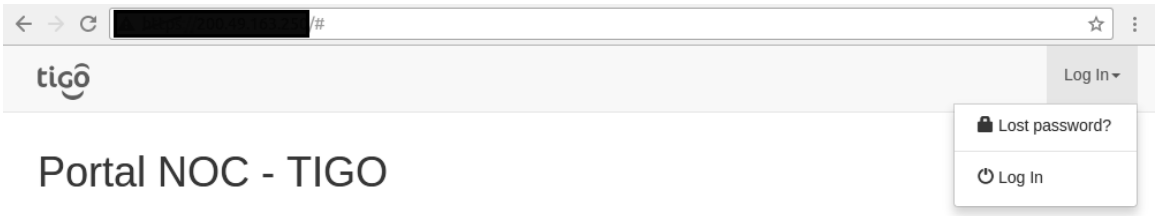
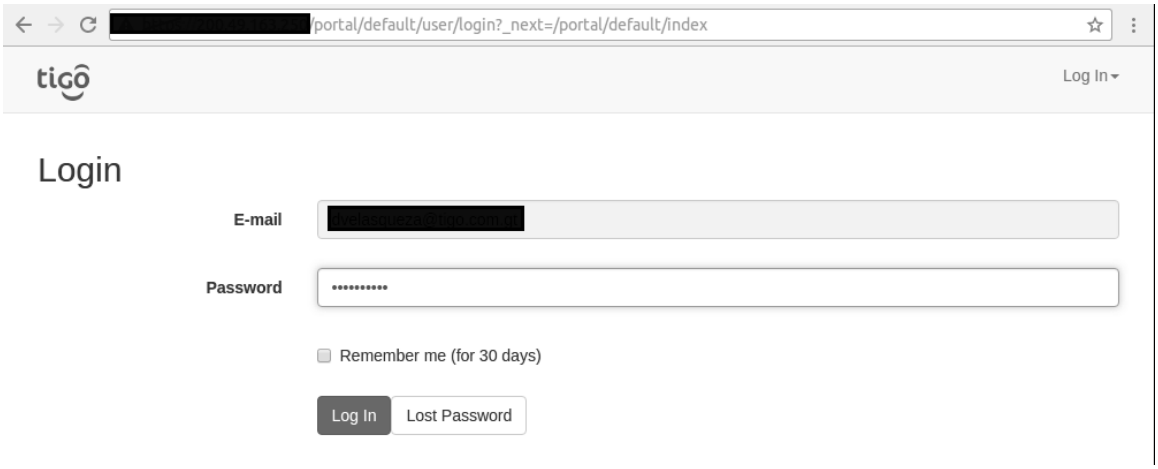
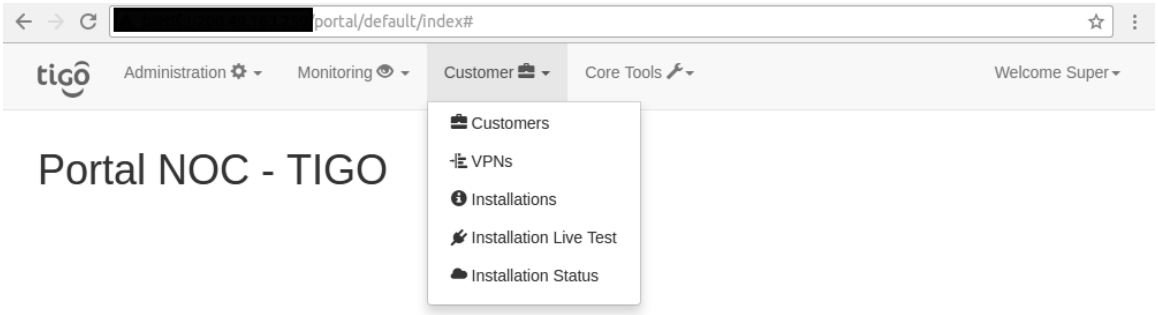


Figura 44. Vista de autenticación al sistema



Luego de la autenticación, la vista inicial muestra en la barra de menú las opciones asociadas a los roles que tiene el usuario tiene asociados. En el ejemplo de la Figura 45, el usuario autenticado tiene todos los permisos asignados.

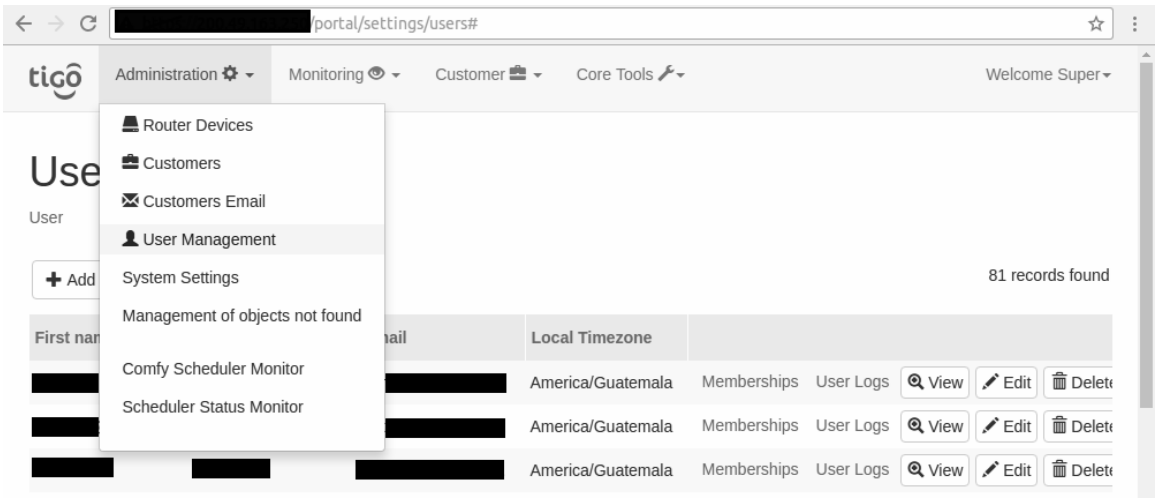
Figura 45. Página inicial del sistema para un usuario autenticado.



B. ADMINISTRACIÓN DE USUARIO Y ROLES DE USUARIO

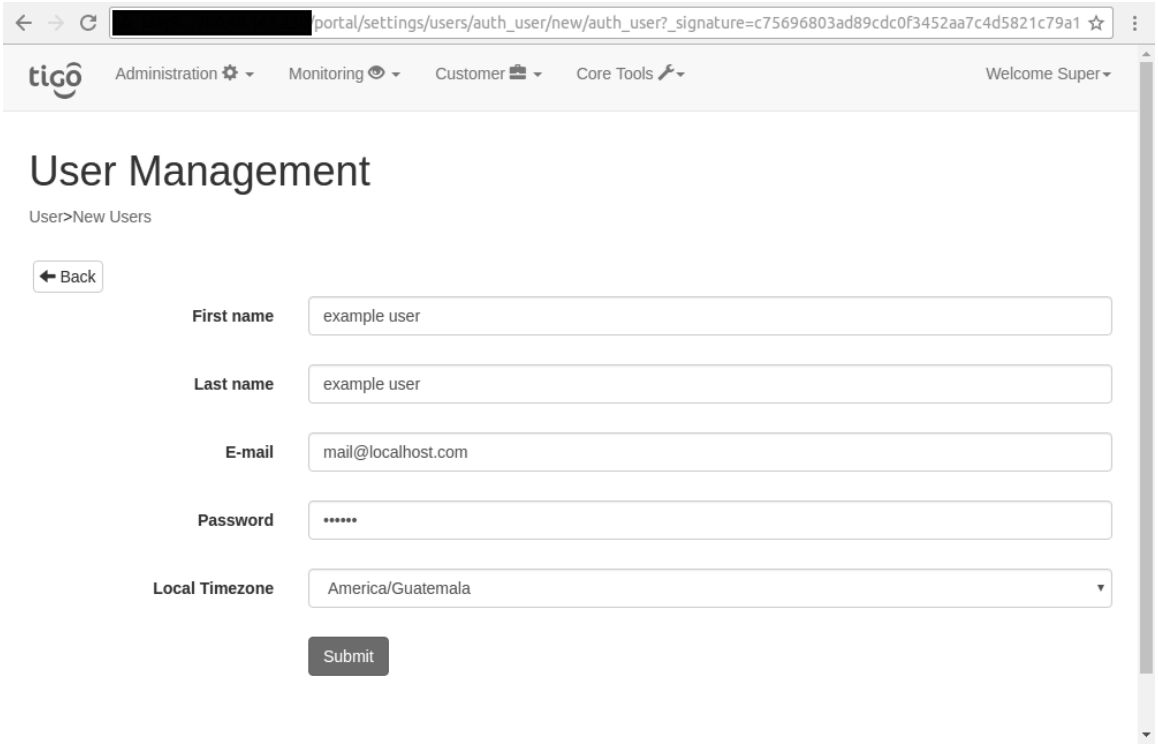
La administración de usuarios y roles, se encuentran relacionados. En el sistema no se permite el registro de usuarios por medio de una vista, únicamente puede realizarlo un administrador del sistema. En la Figura siguiente se despliega el menú de autenticación, desde el cual podemos acceder a la administración de usuarios.

Figura 46. Vista de administración de usuarios y despliegue del menú de administración.



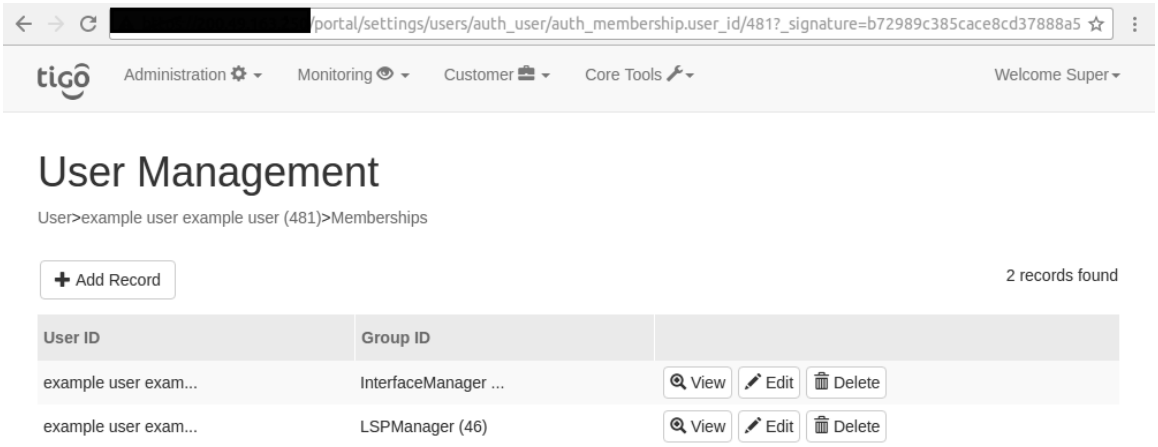
Casa usuario registrado tiene enlaces para la administración de membresías, visualizar logs, ver detalles de usuario, editar un usuario y eliminar un usuario. Se observa parcialmente cubierto por el menú de administración también el botón “Add User”, utilizado para registrar un nuevo usuario, que redirige a la vista de adición de usuarios mostrado en la Figura 46. Las vistas para adición y edición de usuarios son muy similares.

Figura 47. Vista de adición y edición de usuarios.



El botón para la administración de roles o membresías, redirige a una vista, mostrada en la Figura 48, en la que se puede observar los roles asociados a un usuario. En esta vista, se puede asociar un rol existente a un usuario, así como editar y eliminar los roles ya asignados.

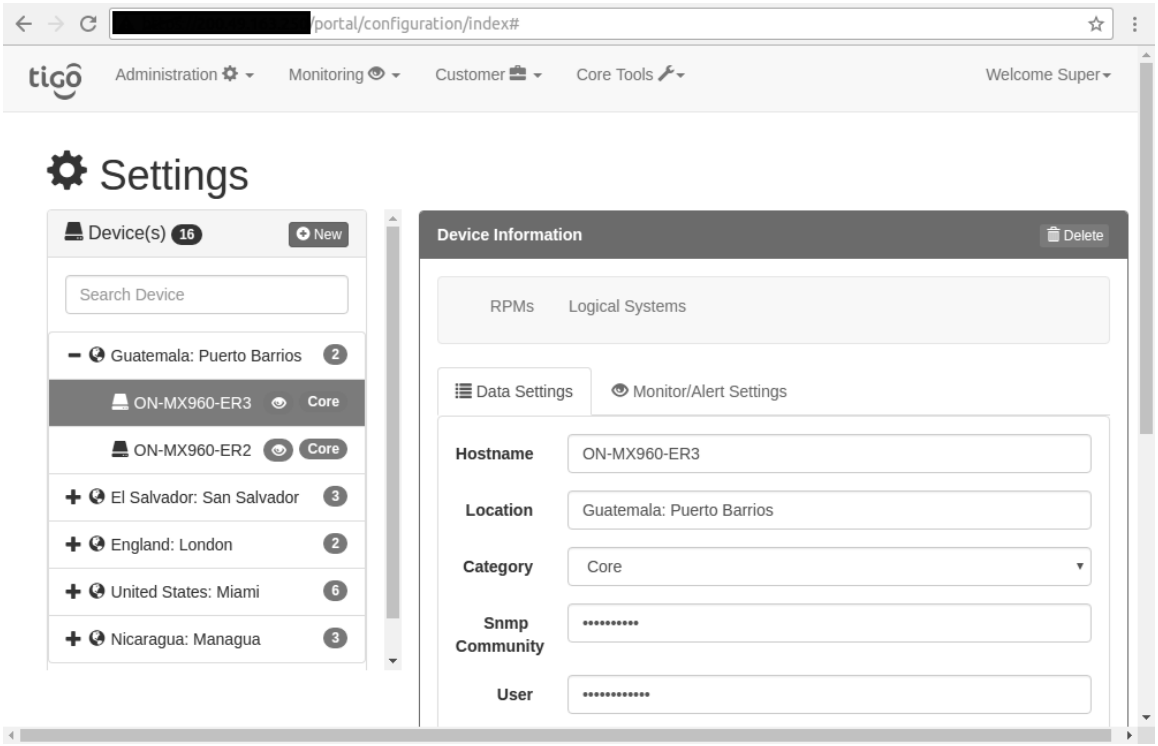
Figura 48. Administración de roles de usuario.



C. ADMINISTRACIÓN DE DISPOSITIVOS DE RED

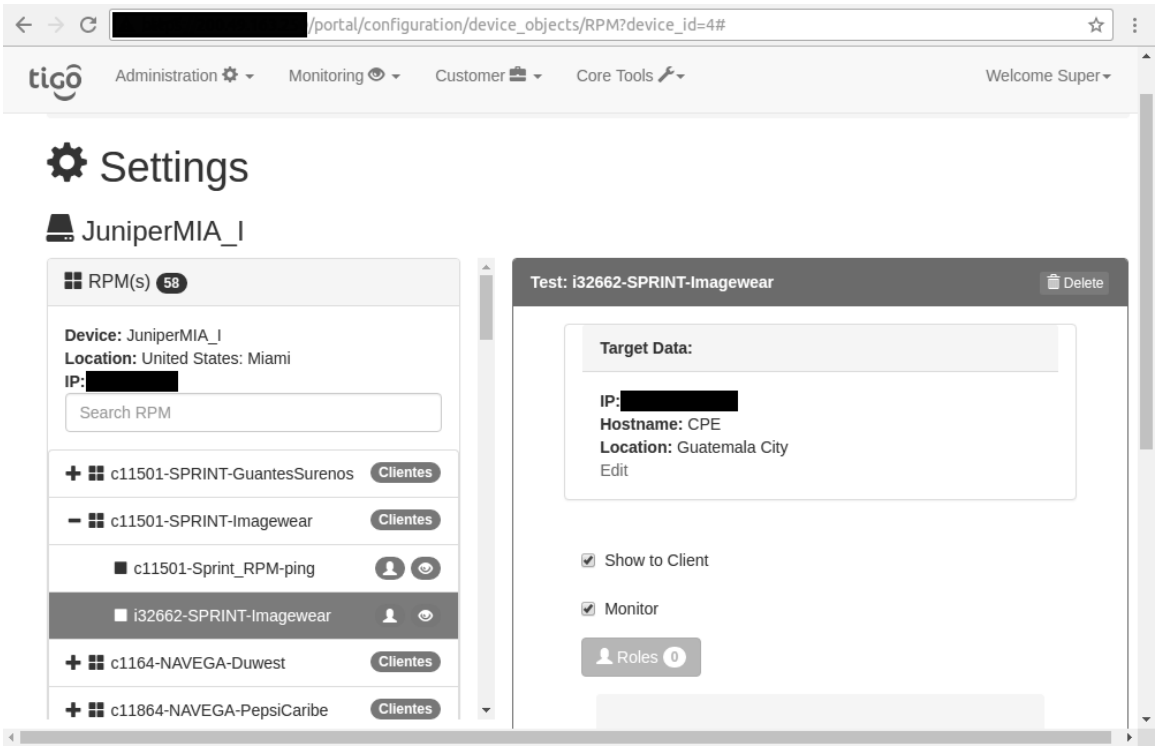
La administración de dispositivos de red es una de las actividades principales para el funcionamiento de los demás componentes del sistema, cuyos objetivos giran entorno a los dispositivos de red. El acceso es desde el menú de administración, a la opción Router Devices, cuya vista es mostrada en la Figura 49.

Figura 49. Vista de administración de dispositivos de red.



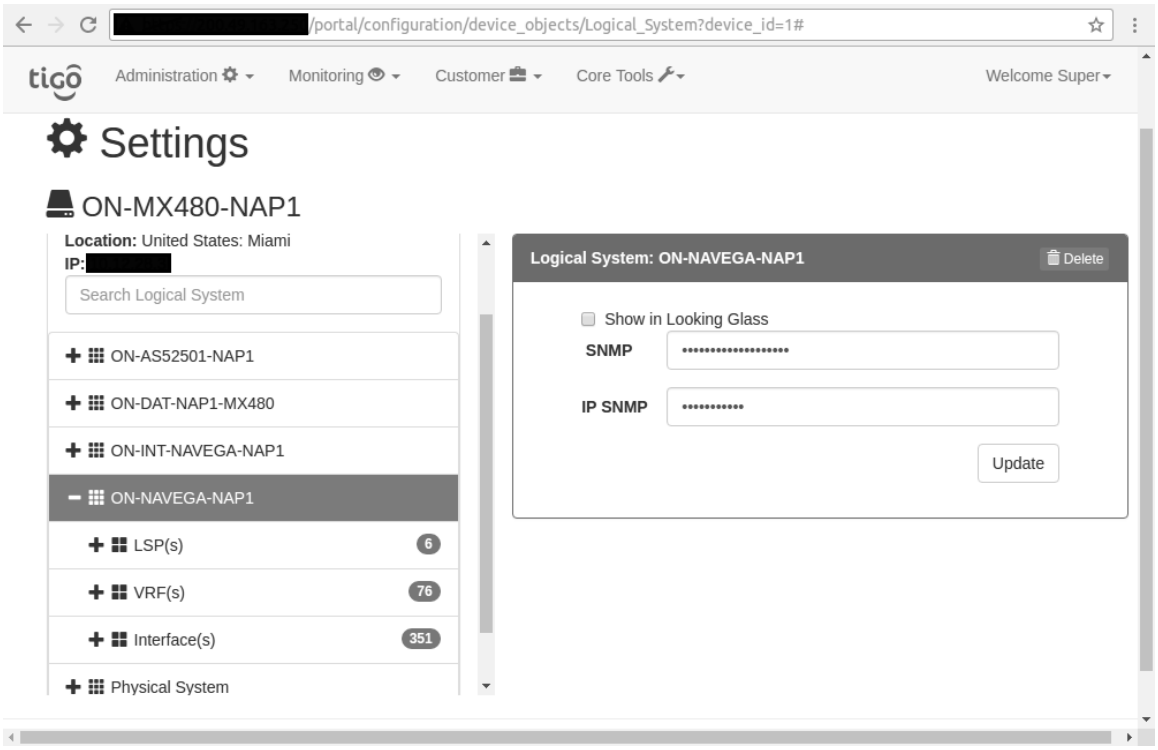
En la parte izquierda se tiene un árbol, que agrupa los dispositivos de acuerdo a su localización geográfica. En la parte derecha, la vista de información del dispositivo es un formulario que permite editar la información del dispositivo. Aquí también se puede eliminar un dispositivo, que generará una eliminación en cascada, eliminando todos los elementos descubiertos del dispositivo, así como las configuraciones y resultados de monitoreo respectivos. Desde aquí también se puede acceder a la vista de RPM's, mostrada en la Figura 50, y sistemas lógicos, en la Figura 51, de un dispositivo de red.

Figura 50. Vista administrativa de RPM's.



La vista administrativa de RPM's también se separa en dos partes, la parte izquierda es una vista de árbol, cuyas raíces corresponden al nombre del Probe y las hojas lo hacen con el nombre de un Test. Cada probe se puede asociar a un tipo de RPM, que puede ser para monitoreo de clientes o de la red del ISP. La parte derecha, mostrará un formulario con las opciones disponibles para la configuración de un Probe o RPM.

Figura 51. Vista administrativa de sistemas lógicos

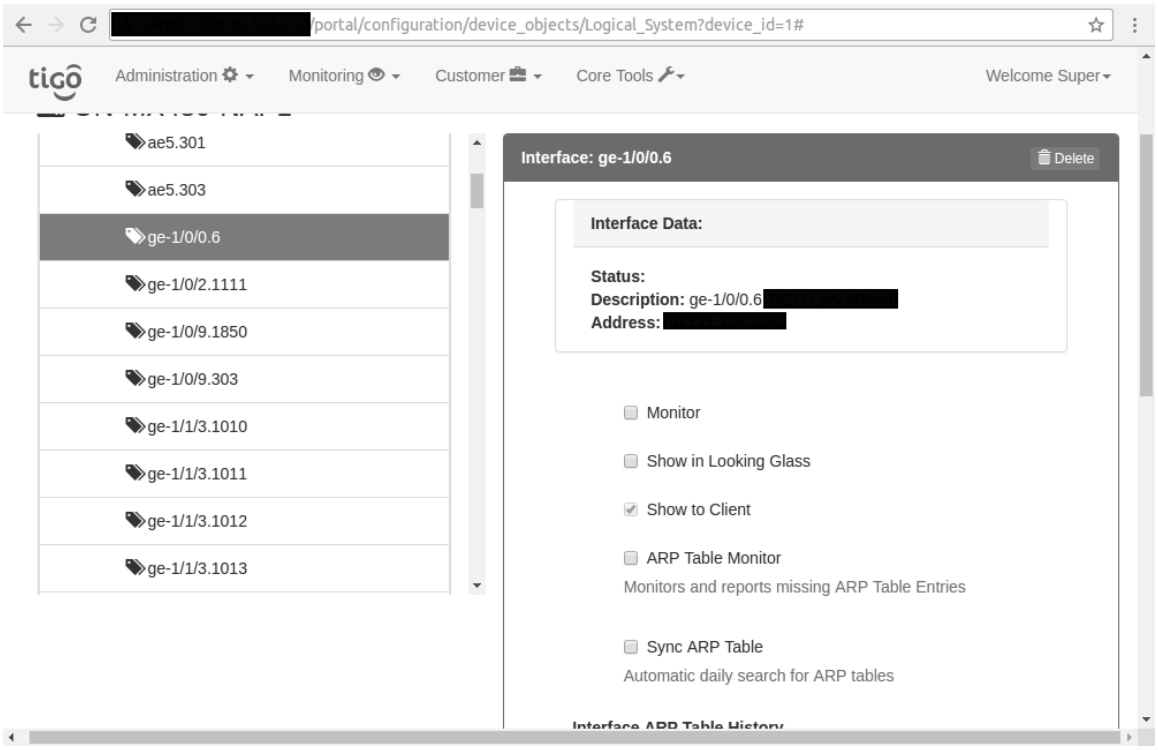


Por su parte la vista administrativa de sistemas lógicos, despliega la información de similar forma a la vista administrativa, conteniendo en la parte izquierda una vista de árbol con una rama para cada sistema lógico, el que a su vez contiene ramas para los LSP's, VRF's e interfaces de red descubiertos en el sistema lógico. Notar que se asume la existencia de un sistema lógico llamado Physical System usado para el descubrimiento en el Router.

En la parte derecha, de acuerdo al elemento seleccionado en el listado izquierdo, se despliega la configuración que se puede aplicar a cada elemento. En el caso de un sistema lógico, si la comunidad SNMP será la misma que la definida para el Sistema Físico, la IP de acceso y si se puede mostrar en el Looking Glass como un dispositivo desde el cual realizar pruebas hacia Internet (ampliado más adelante).

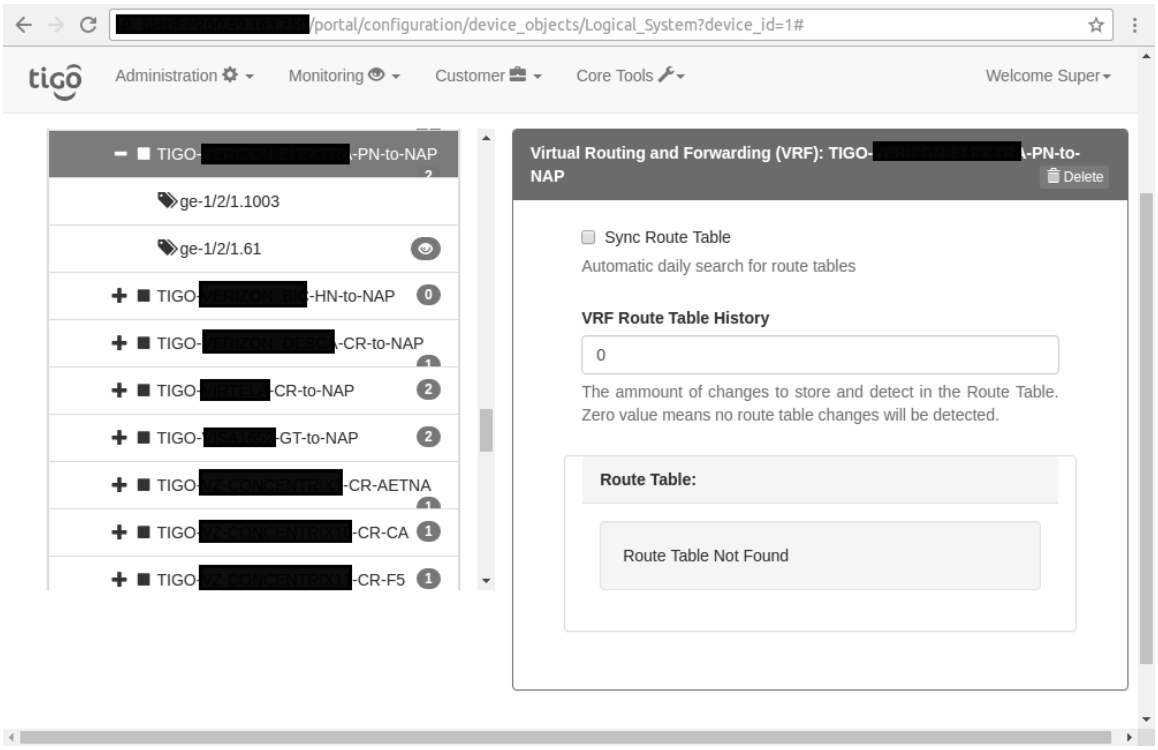
En la Figura 52 se muestra parte de las opciones de configuración de una Interfaz de red. Dentro de la configuración existe la posibilidad de seleccionar qué interfaces pueden ser utilizadas en el Looking Glass, opciones de monitoreo y si se encuentran asociadas a una Routing Instance o instalación.

Figura 52. Configuración de una interfaz de red.



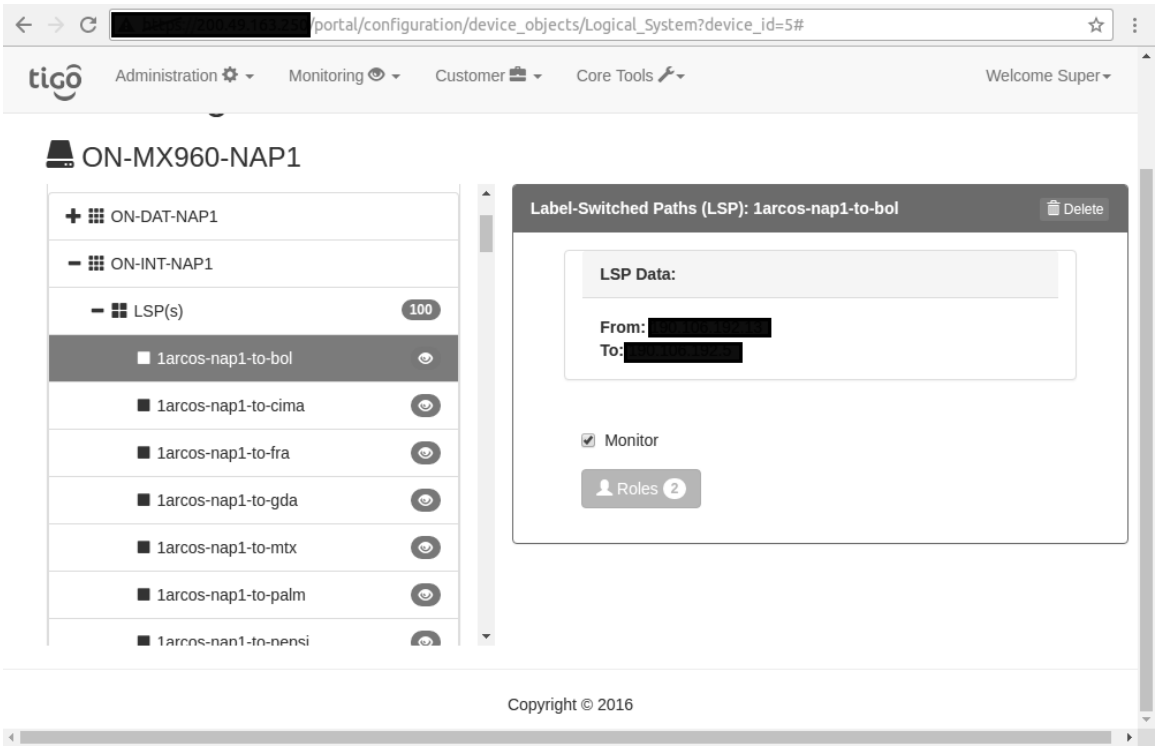
La vista de configuración de una VRF se muestra en la Figura 53. En esta podemos ver también qué interfaces se encuentran asociadas a una VRF, y si se desea monitorear la tabla de rutas, así como definir la cantidad de tablas que se desea almacenar.

Figura 53. Configuración de una Routing Instance de tipo VRF.



La vista administrativa de LSP's es más simple que las anteriores, pues únicamente se indica si un se encuentra activo el monitoreo del LSP. En la Figura 54 se muestra la vista administrativa de un LSP.

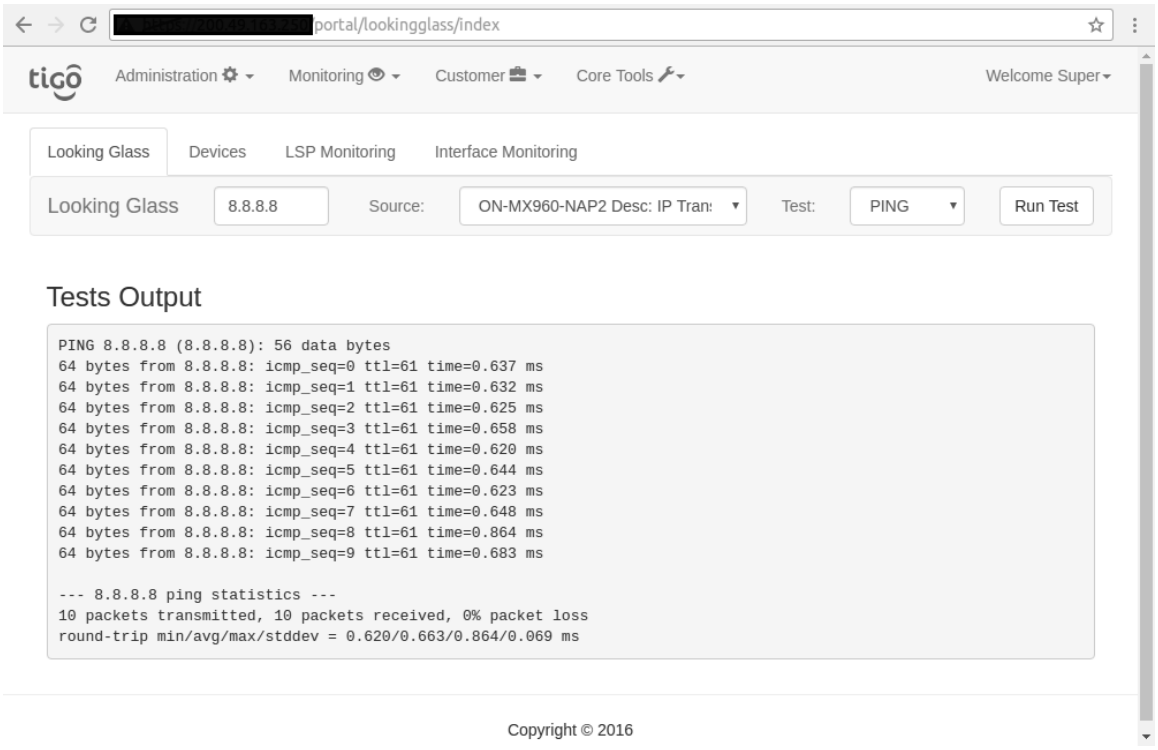
Figura 54. Configuración de un LSP.



D. HERRAMIENTAS DE DIAGNÓSTICO DE RED

1. Looking Glass. Dentro de las herramientas de diagnóstico, se presenta como primer punto el Looking Glass. Anteriormente se muestra la configuración de sistema lógicos e interfaces de red, que pueden ser asociados al Looking Glass. Los sistemas lógicos se utilizan principalmente para la ejecución de pruebas BGP, mientras que las interfaces de red se utilizan para pruebas de Ping y Traza. En la Figura 55 se muestra un ejemplo de prueba de Ping.

Figura 55. Ejemplo de prueba de Ping desde Looking Glass.



2. Monitoreo de dispositivos. Esta es la segunda herramienta de monitoreo requerida por el cliente. Esta herramienta por defecto monitorea el porcentaje de uso del CPU, el porcentaje de uso de memoria y la temperatura de funcionamiento. En la Figura 56 se observa la presentación del monitoreo de un dispositivo en forma de gráficas. Cuando se tiene el permiso de configuración de dispositivos, se puede acceder a la vista de correspondiente, como se muestra en la Figura 57, en la que se puede definir si se monitorean los elementos antes mencionados.

Figura 56. Monitoreo de un dispositivo de red.

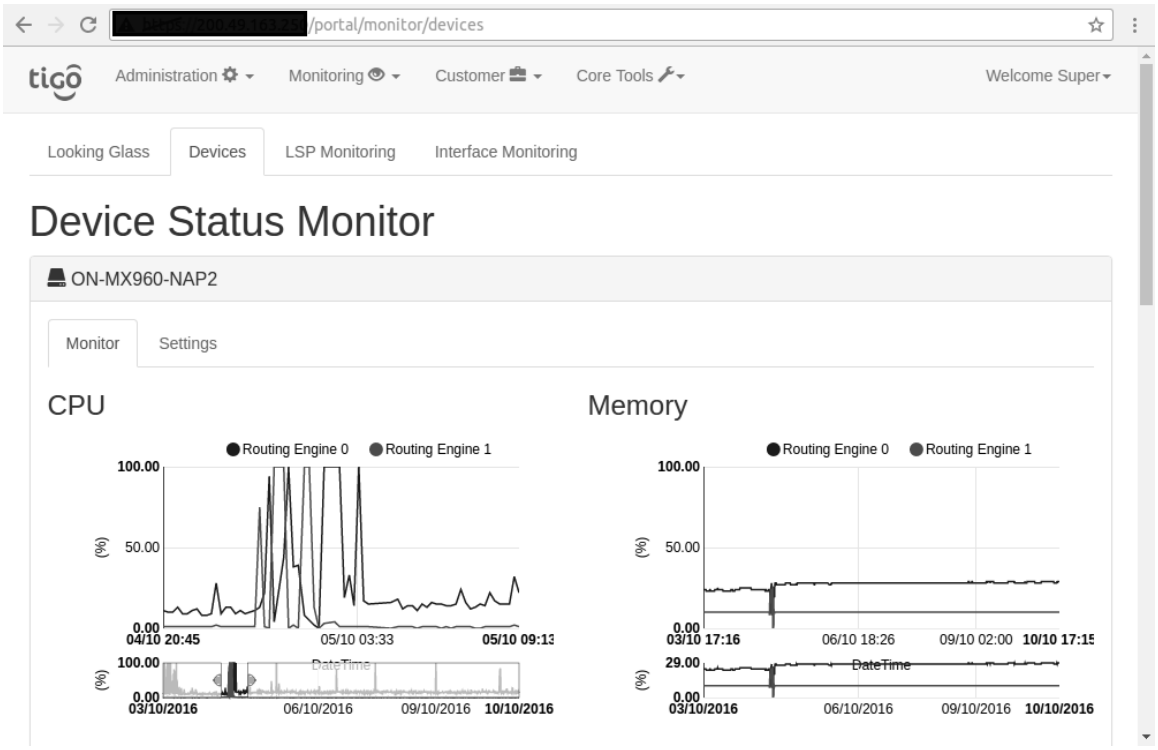
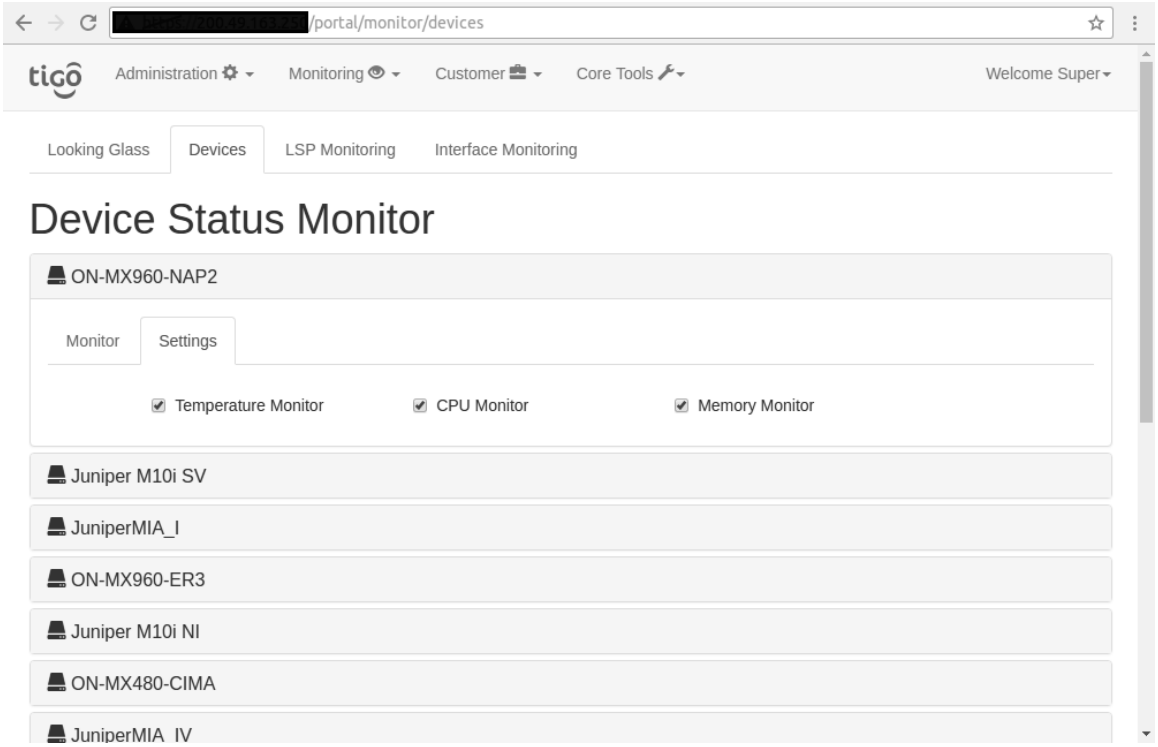


Figura 57. Activación o desactivación del monitoreo.



3. Monitoreo del rendimiento de la red. Este se compone de dos elementos: consumo de tráfico de interfaces y LSP's y pruebas de rendimiento de la red. El consumo de tráfico en LSP's cuenta con su propia vista, en la que se listan los LSP's identificados, es requerimiento que se configuren antes de su uso desde las vistas antes presentadas. En la vista, se puede filtrar los LSP's, y visualizar el estado y el path activo de un LSP, como se muestra en la Figura 58. Las gráficas de tráfico también se pueden visualizar al hacer click en el botón correspondiente, como se muestra en la Figura 59.

Figura 58. Vista de LSP's monitoreados.

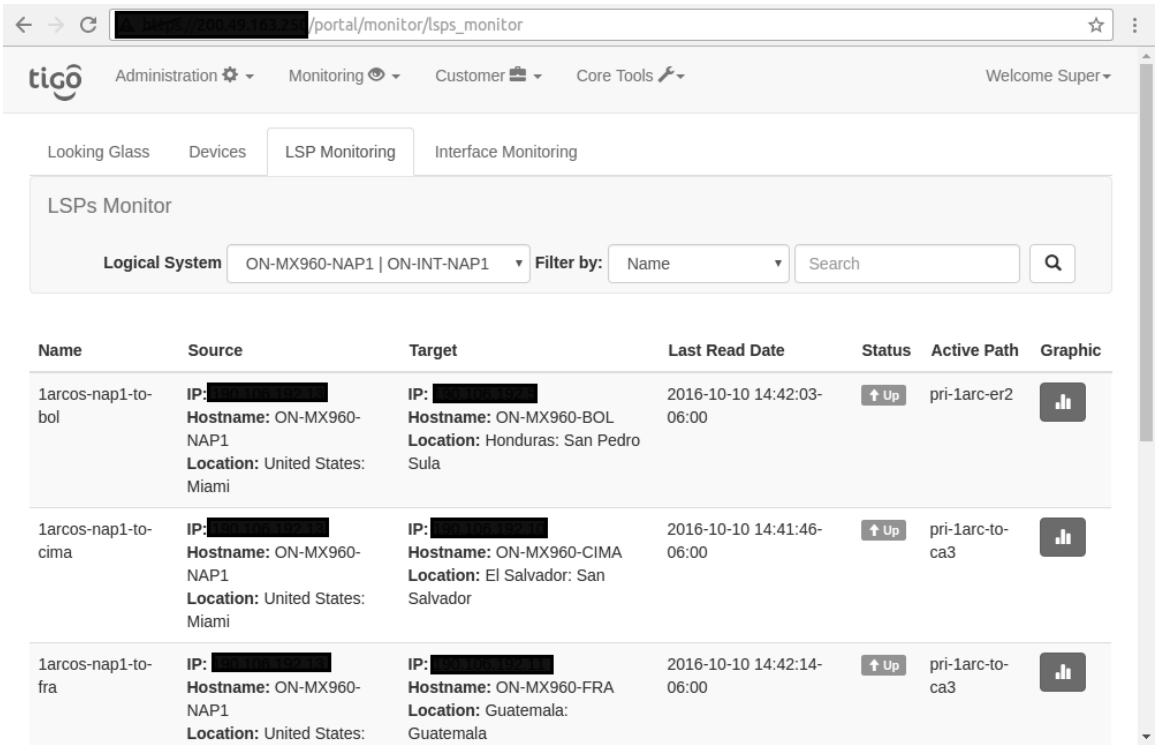
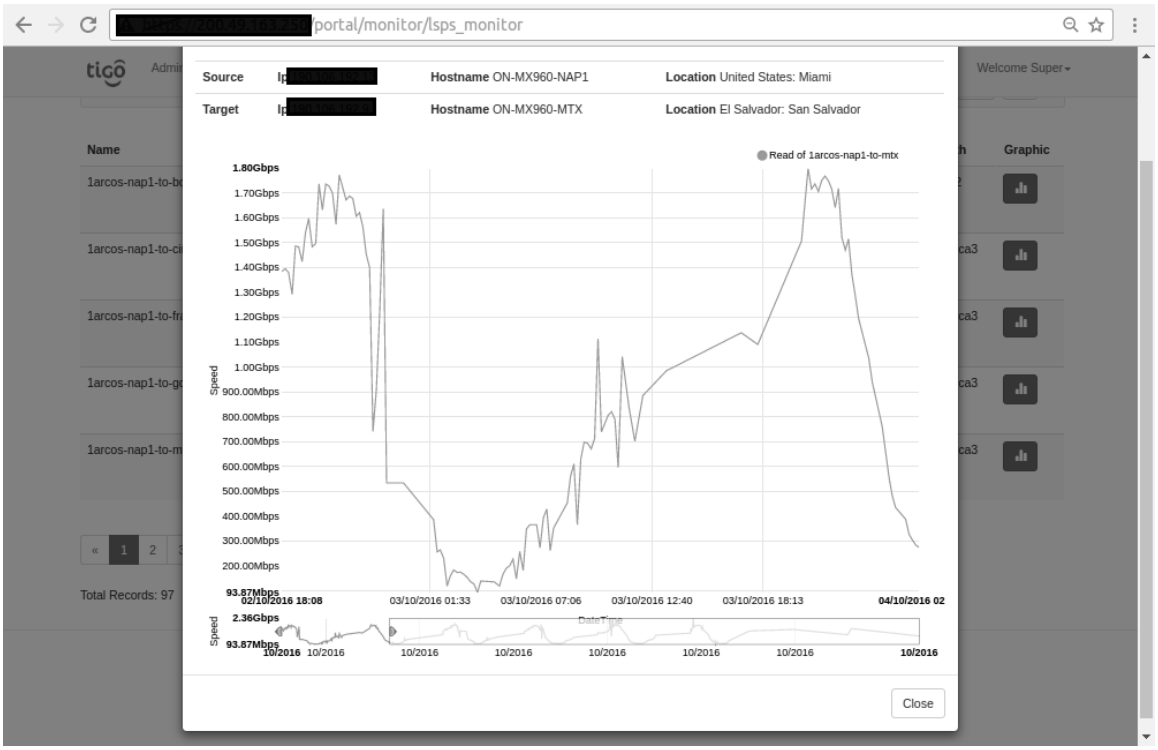


Figura 59. Gráfica del consumo de tráfico de un LSP



Las vista de interfaces es similar a la de LSP's. En esta también se visualiza listado de interfaces de red, su respectivo estado y la gráfica de tráfico. En las Figuras 60 y 61 se muestra la vista de interfaces y una gráfica de consumo.

Figura 60. Vista de interfaces de red monitoreadas.

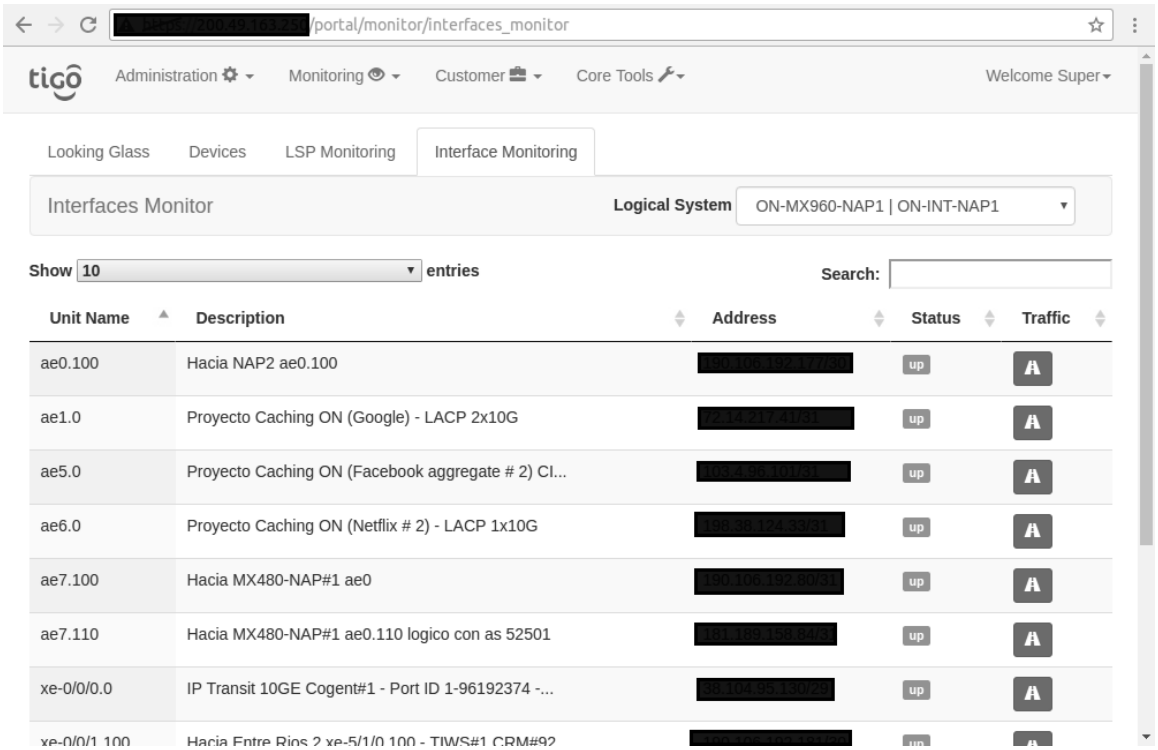
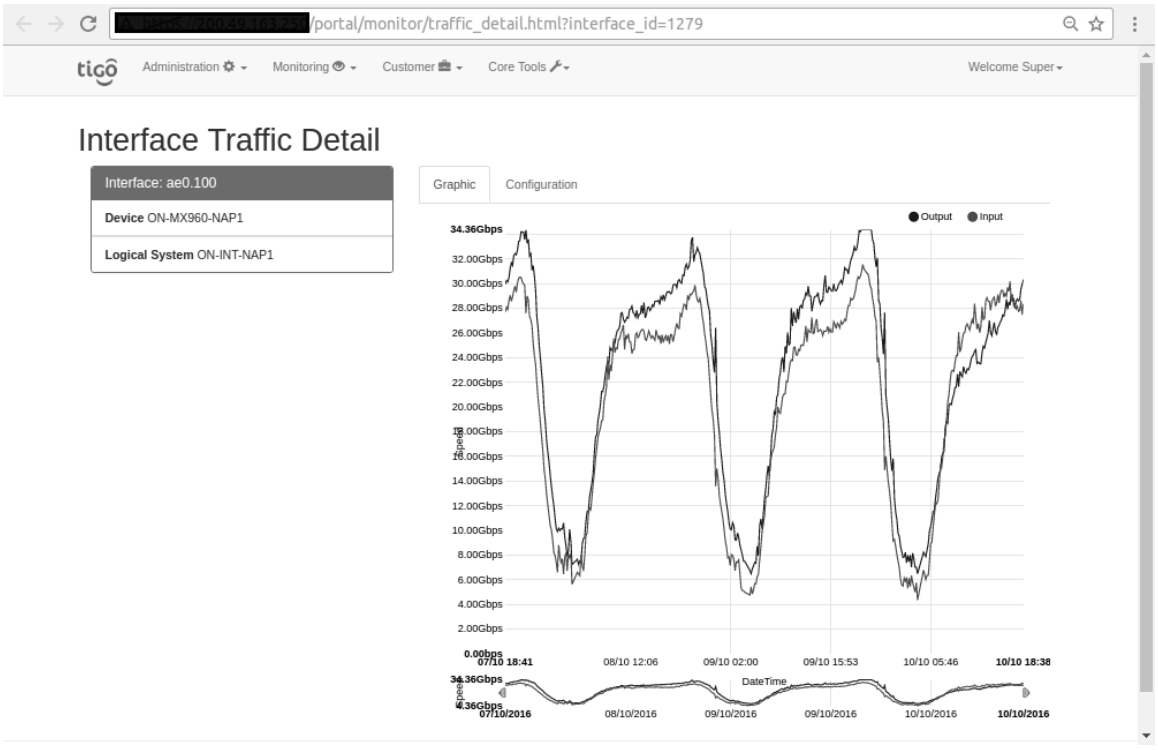


Figura 61. Gráfica del consumo de tráfico de una interfaz de red.



Para el monitoreo se optó por presentar los valores en una forma matricial. En la Figura 62 se muestra la vista de rendimiento. En el panel izquierdo se puede elegir diferentes opciones de visualización, como la sobre la que se desea probar, la vista (por dispositivos o por router) y el parámetro (RTT, Jitter o Packet Loss). Cada valor presentado es un valor promedio, y también funciona como un enlace que permite visualizar la gráfica de rendimiento, como se muestra en la Figura 63.

Figura 62. Vista matricial del Rendimiento de una Red.

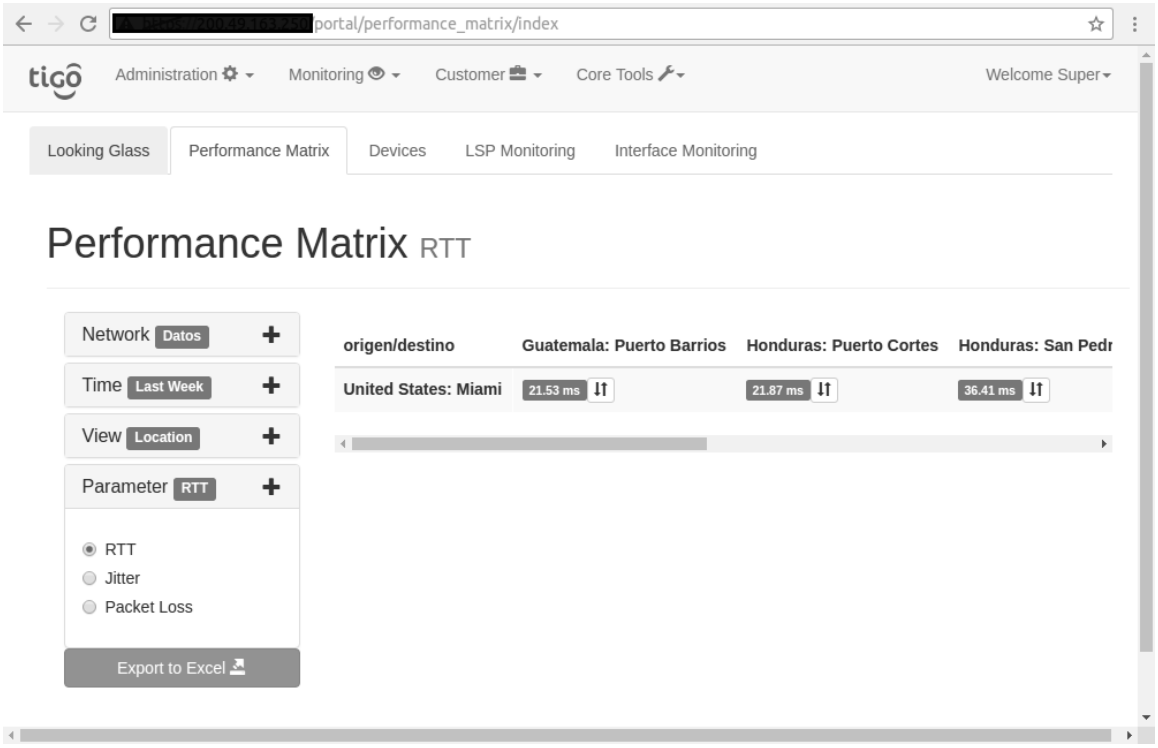
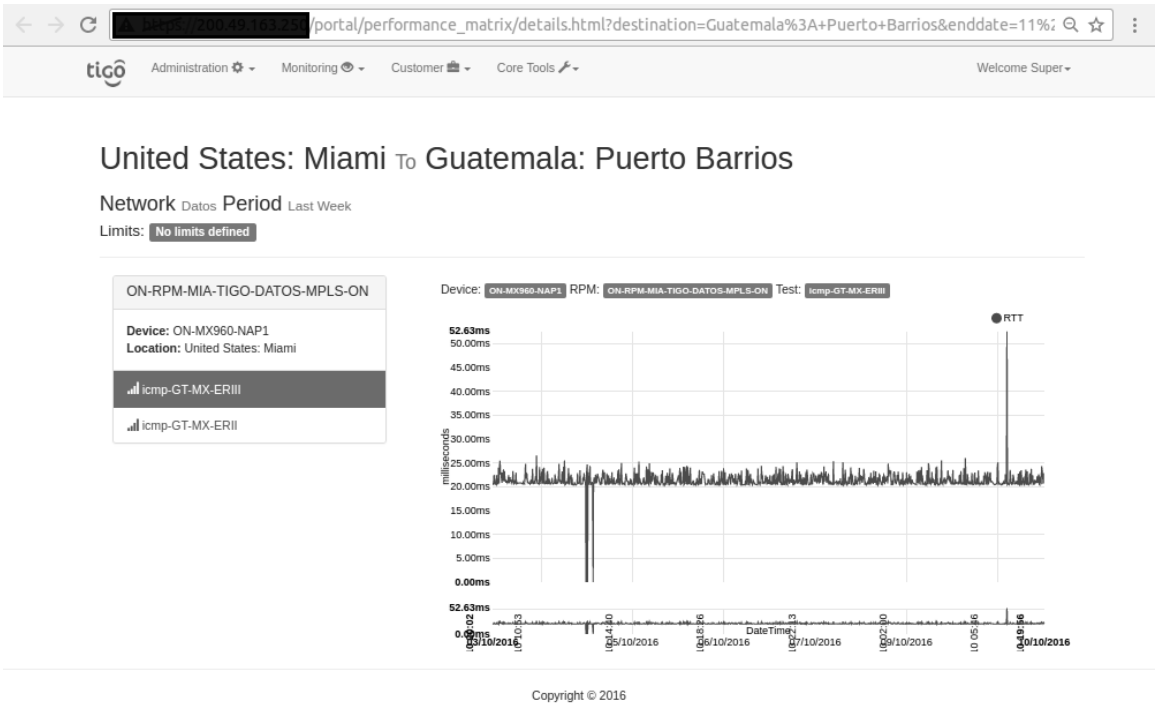


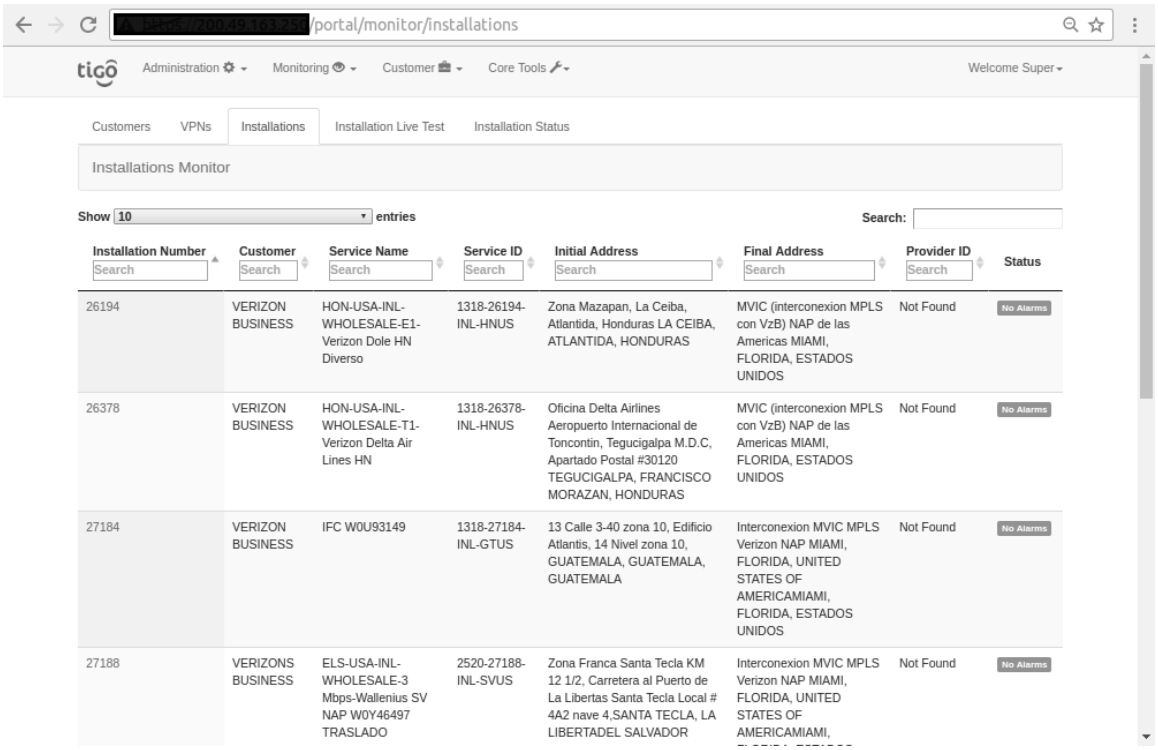
Figura 63. Gráfica histórica del RTT de un RPM usado para rendimiento.



E. HERRAMIENTAS DE DIAGNÓSTICO DE UN SERVICIO VPN CAPA 3 BASADO EN BGP/MPLS

Los servicios VPN Capa 3 basados en BGP/MPLS, tienen diferentes vistas para su visualización. Estos pueden ser buscados por el nombre de la VPN o el número de instalación. En la Figura 64 se muestra la vista de instalaciones.

Figura 64. Vista de instalaciones.



El número de instalación funciona como un enlace para acceder a la vista de cada instalación. Dentro de la vista de instalación se puede ver el detalle de la instalación, configuración de las interfaces de red del PE, gráficas de consumo de tráfico de las interfaces de red, tablas de direcciones MAC, gráficas de rendimiento, tablas de rutas y ejecutar pruebas en tiempo real. Si se tiene el permiso, también se puede acceder a la vista de configuración de la instalación. Estos detalles se muestran en las Figuras 65 a 72.

Figura 65. Detalles de una instalación.

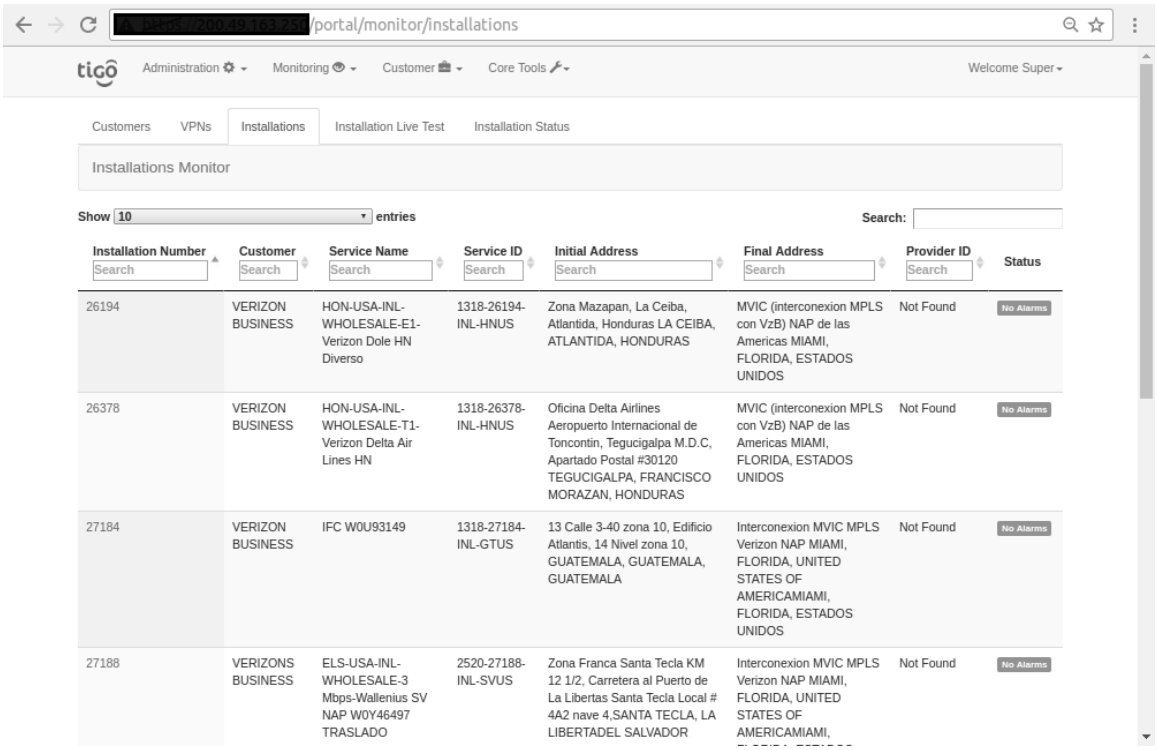


Figura 66. Configuración de interfaces de red.

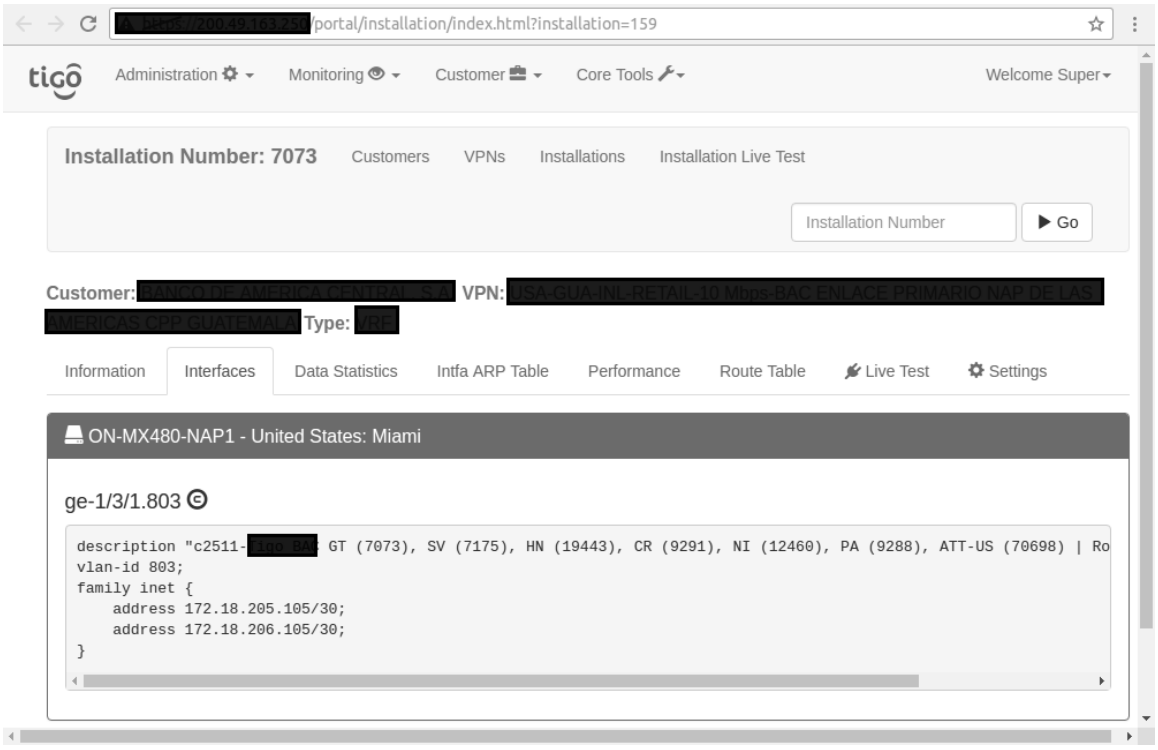


Figura 67. Gráfica de consumo de tráfico.

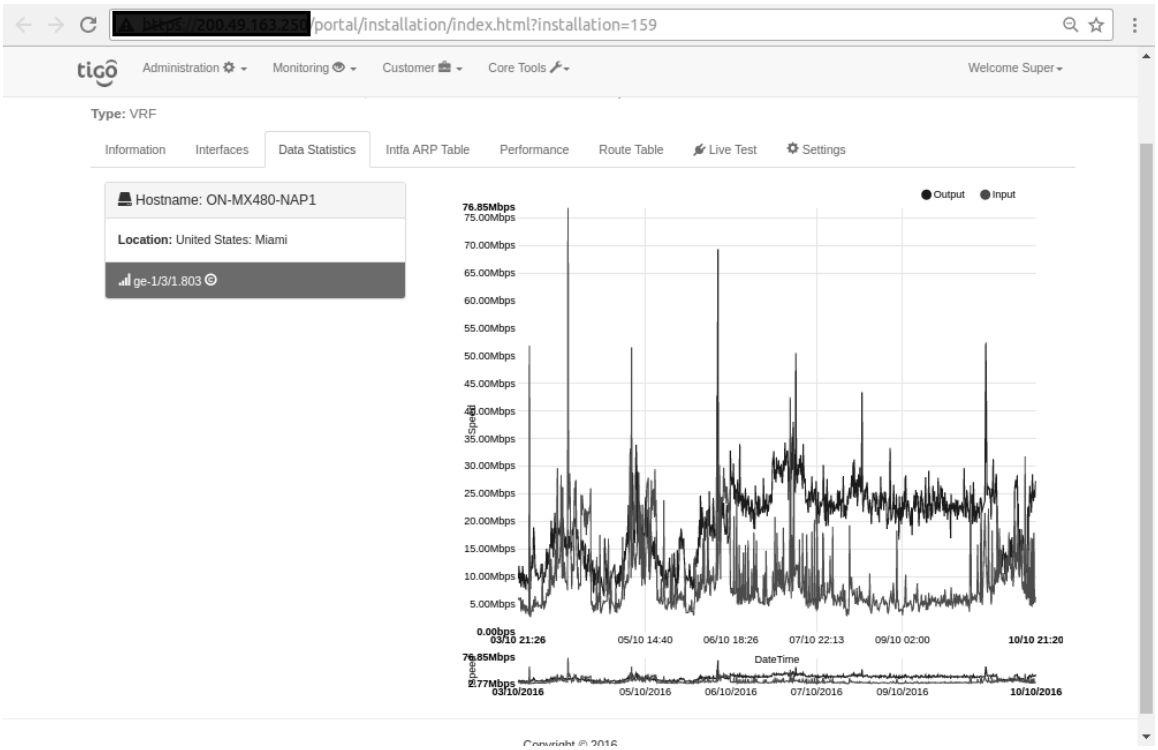


Figura 68. Gráfica de rendimiento del monitoreo de la instalación.

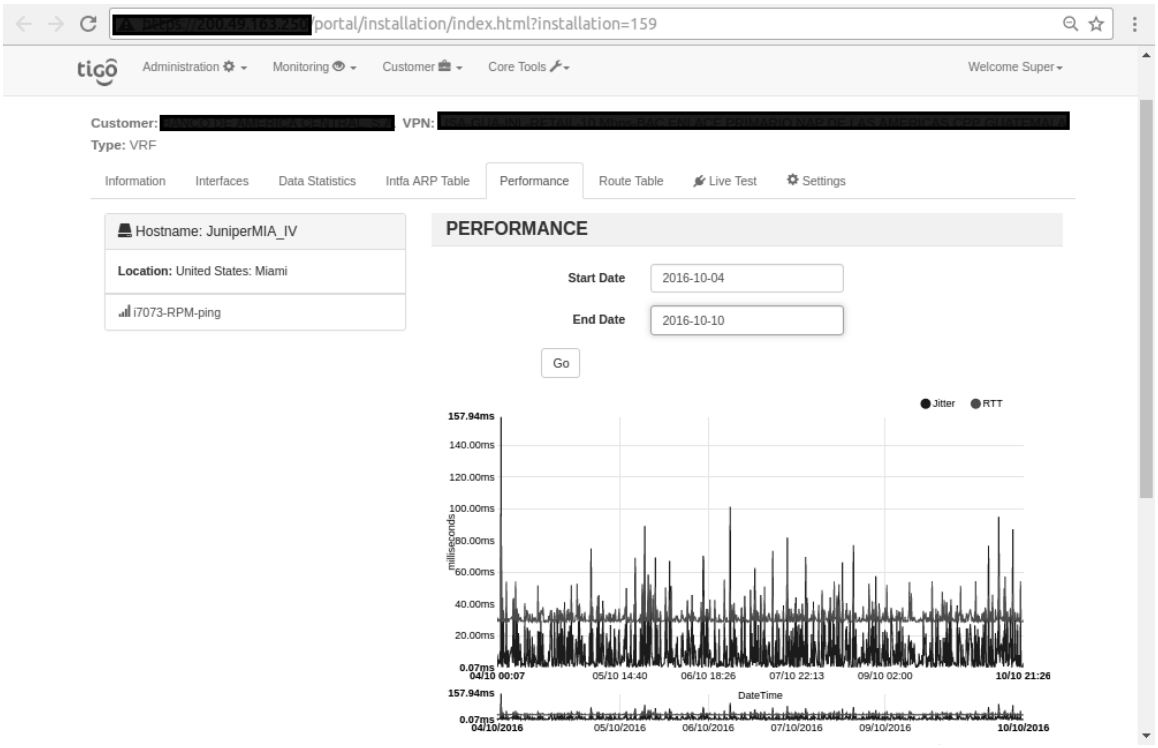


Figura 69. Tabla de direcciones MAC de una interfaz de red.

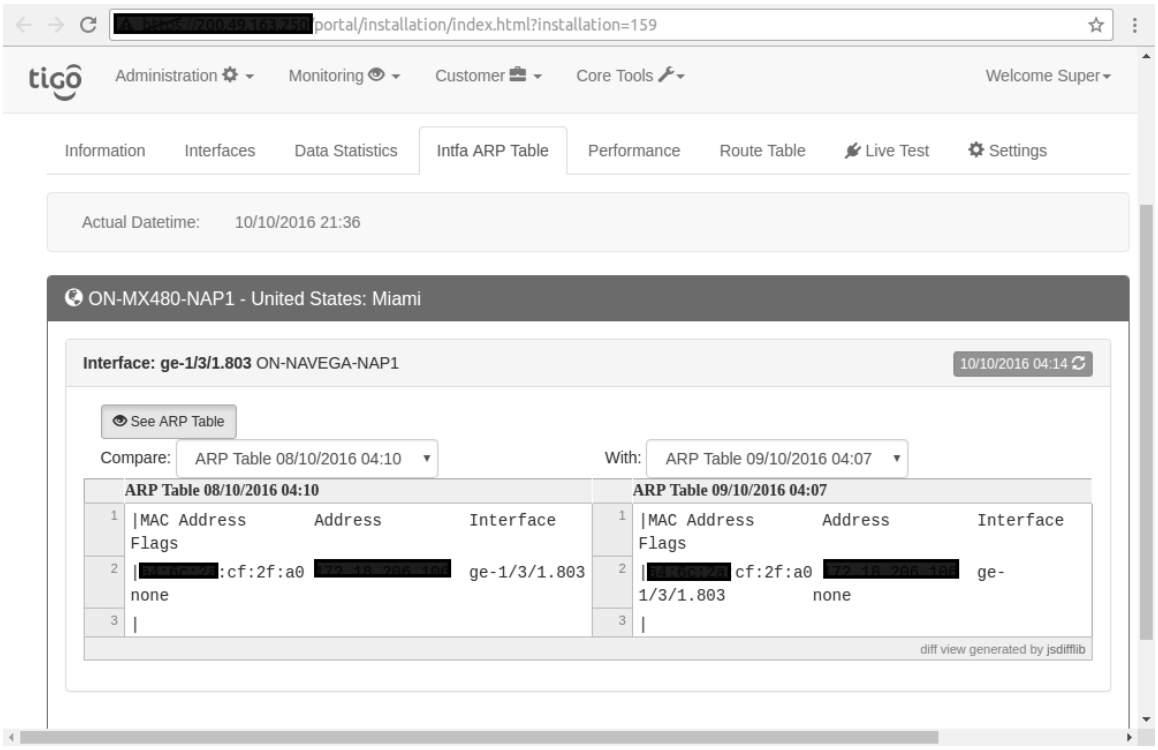


Figura 70. Tabla de rutas de una Routing Instance.

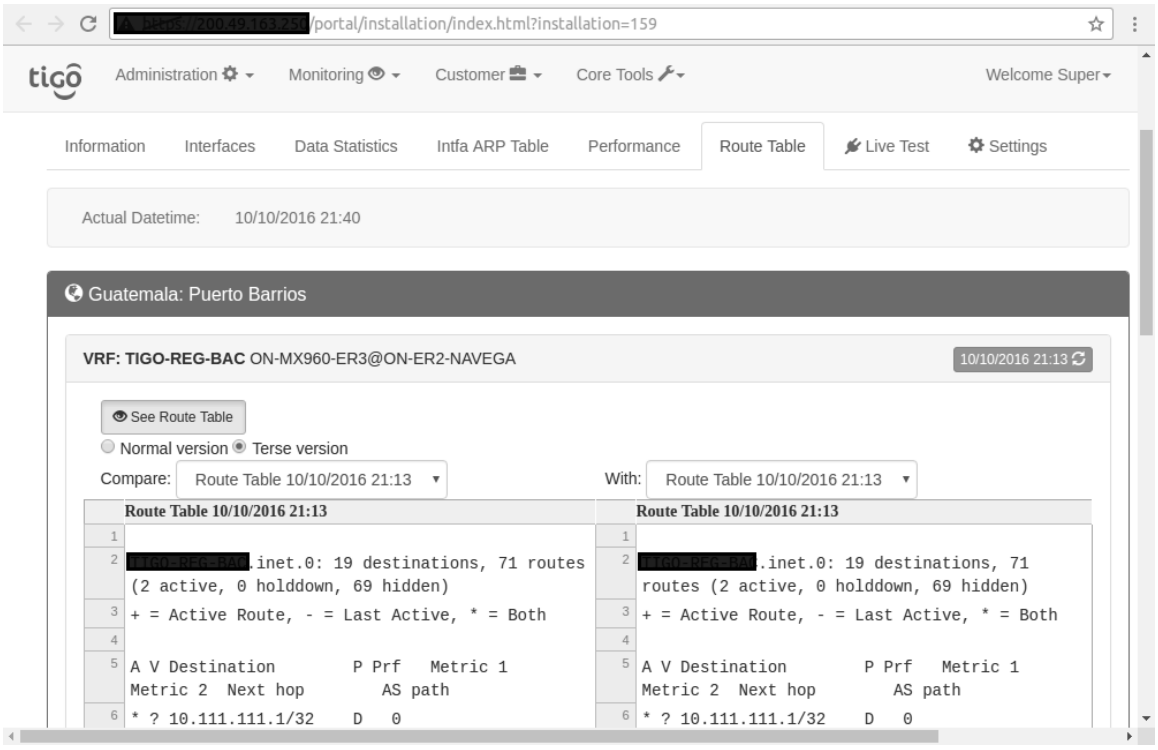


Figura 71. Pruebas en tiempo real sobre una VPN.

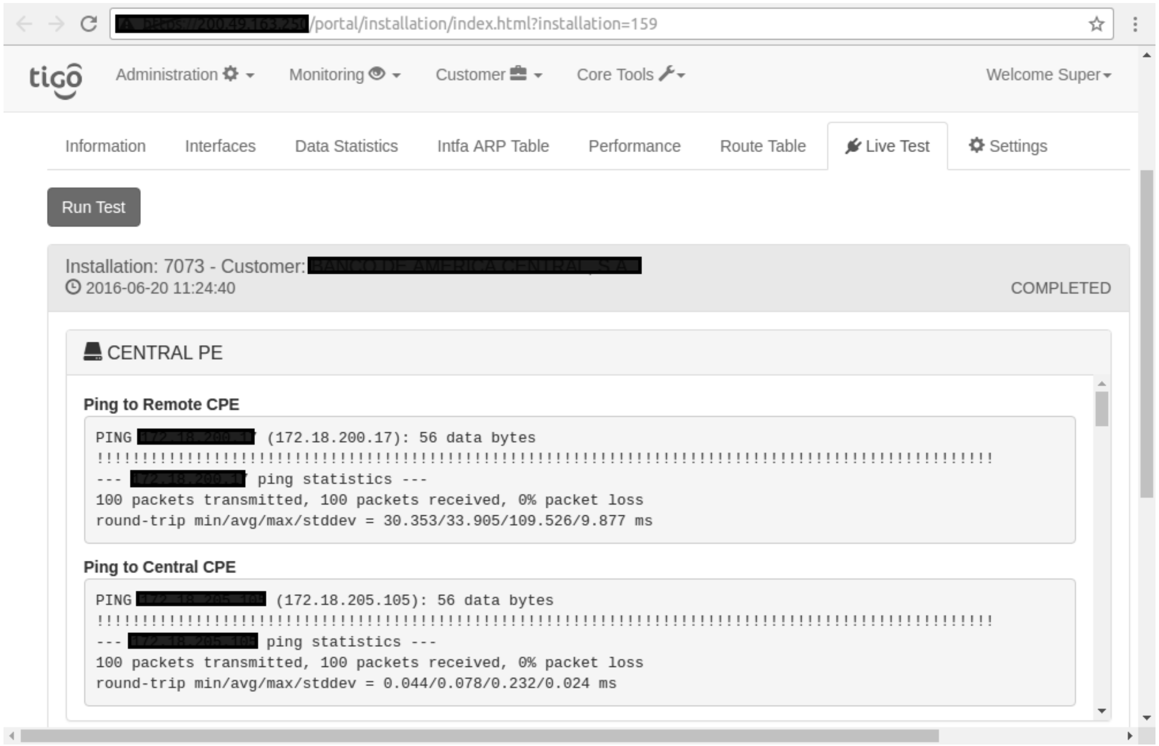
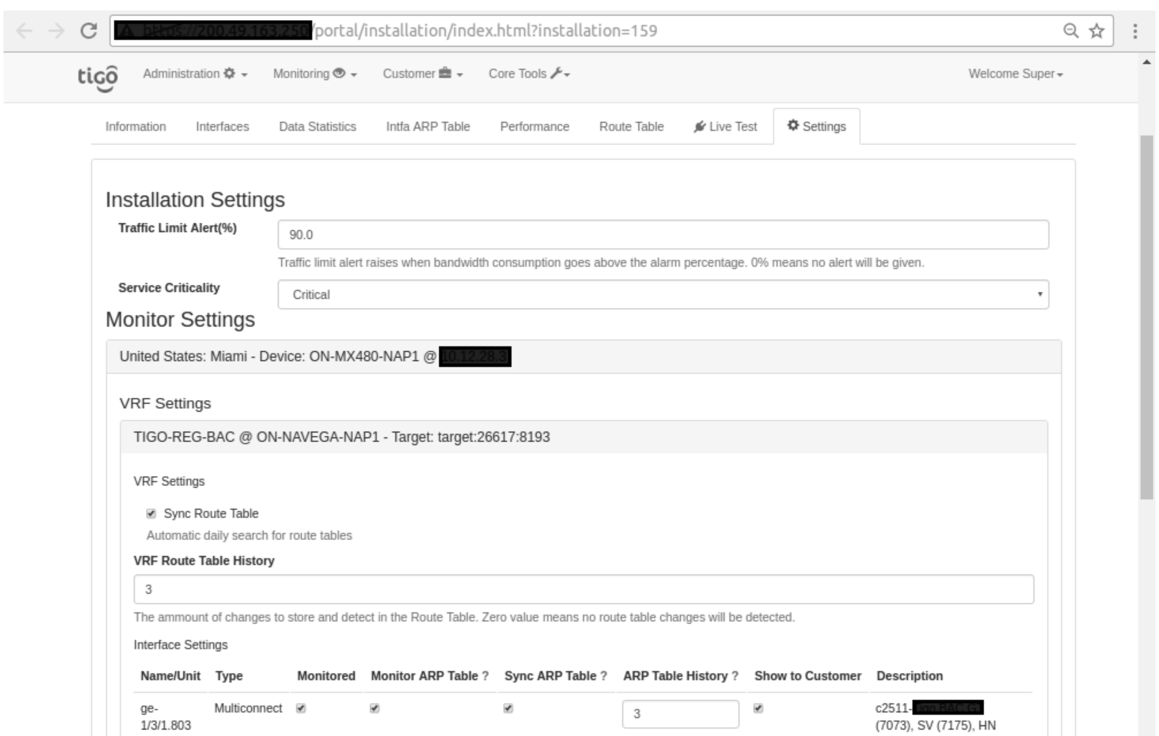


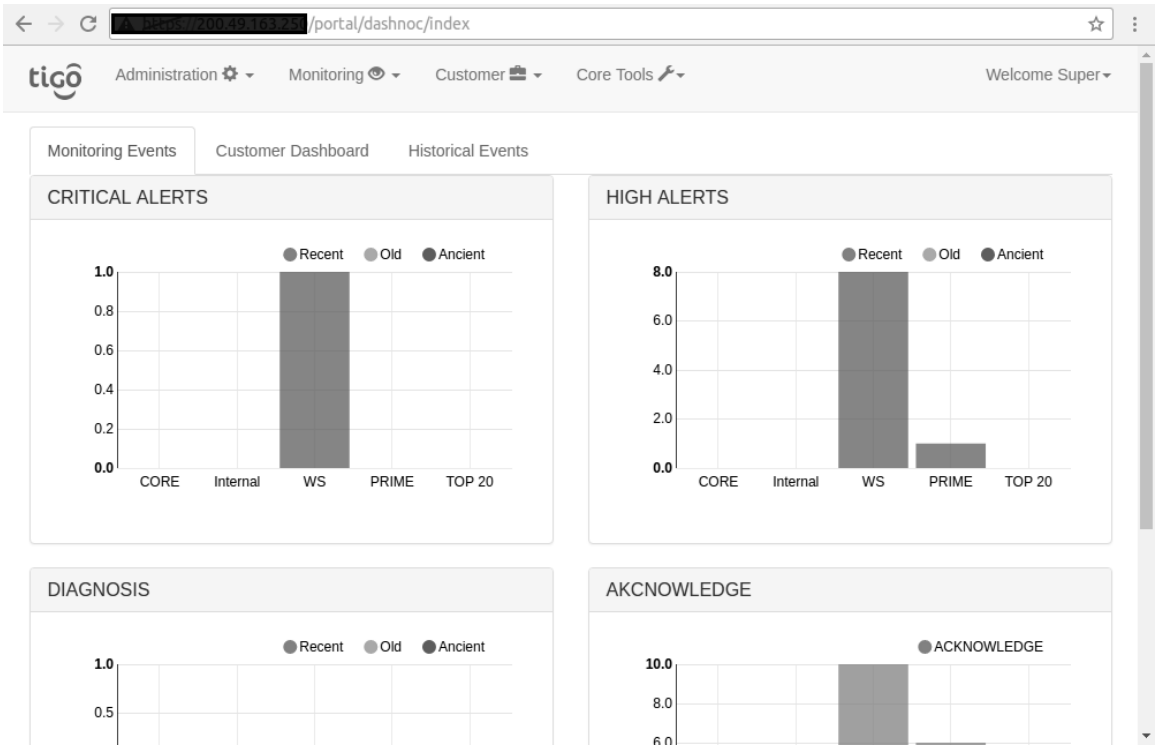
Figura 72. Configuración de una VRF.



F. HERRAMIENTAS DE MONITOREO

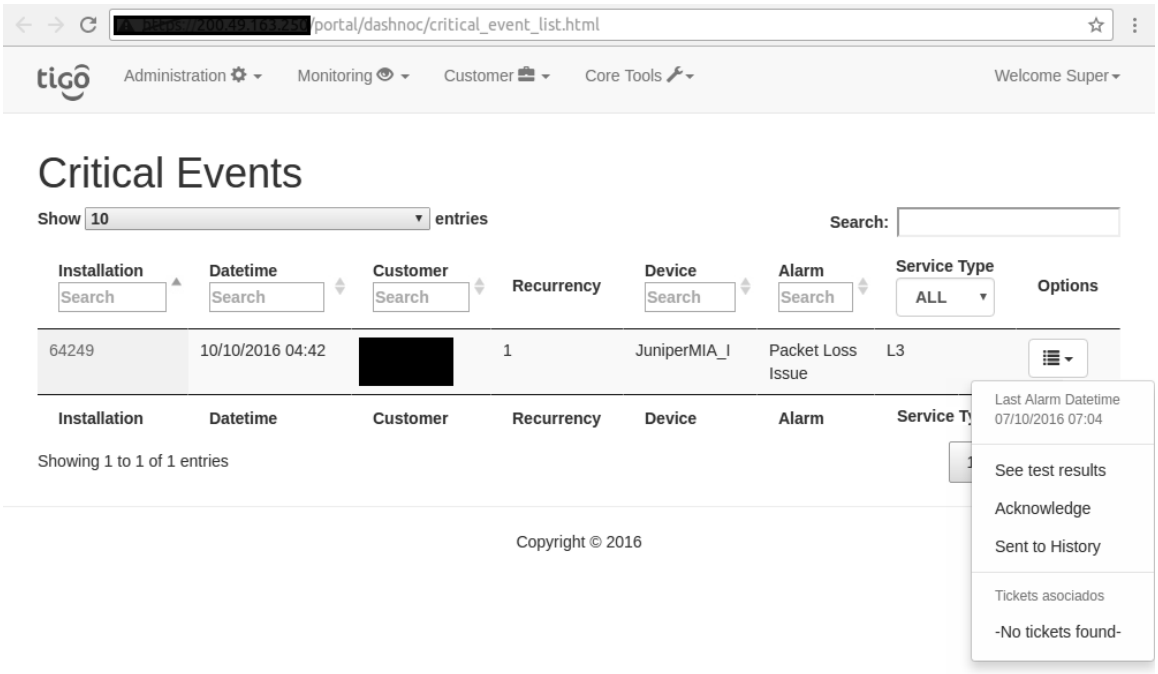
Las herramientas de monitoreo buscan ser la principal herramienta de uso en un NOC. Estas se encargan del manejo de las alertas generadas por los diferentes elementos del sistema. El tablero de alertas muestra en tiempo real una gráfica de la cantidad de alertas identificadas, como se observa en la Figura 73. Las alertas se agrupan por tipo clientes y por criticidad de las mismas y de los servicios.

Figura 73. Tablero de alertas.



Al hacer click en una de las barras, se accede al listado de alertas, donde se puede observar el detalle de las alertas. En la opciones de cada alerta, están las opciones para enviar a Acknowledge e histórico. Cuando se encuentra un ticket activo, se puede añadir un comentario al mismo, de igual forma, si no se encuentra un ticket activo se puede crear uno nuevo.

Figura 74. Listado de alertas.



Los resultados de la alerta son accedidos en el enlace “See test results”, que son desplegados como se muestra en la Figura 75. Al pie de las alertas, que pueden ser colapsadas, existe un menú con las opciones para la creación de tickets y, de existir un ticket asociado, agregar un comentario en el ticket. Este menú se muestra en la Figura 76.

Figura 75. Visualización de resultados del test que genera una alerta.

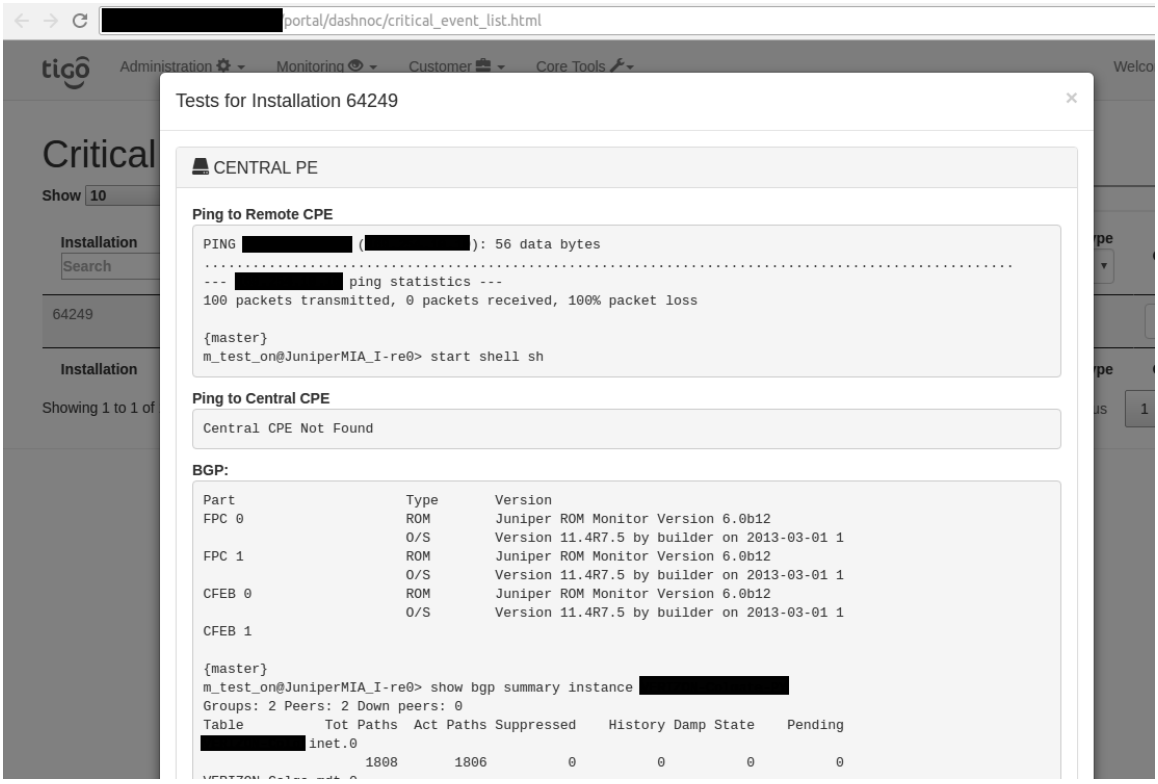
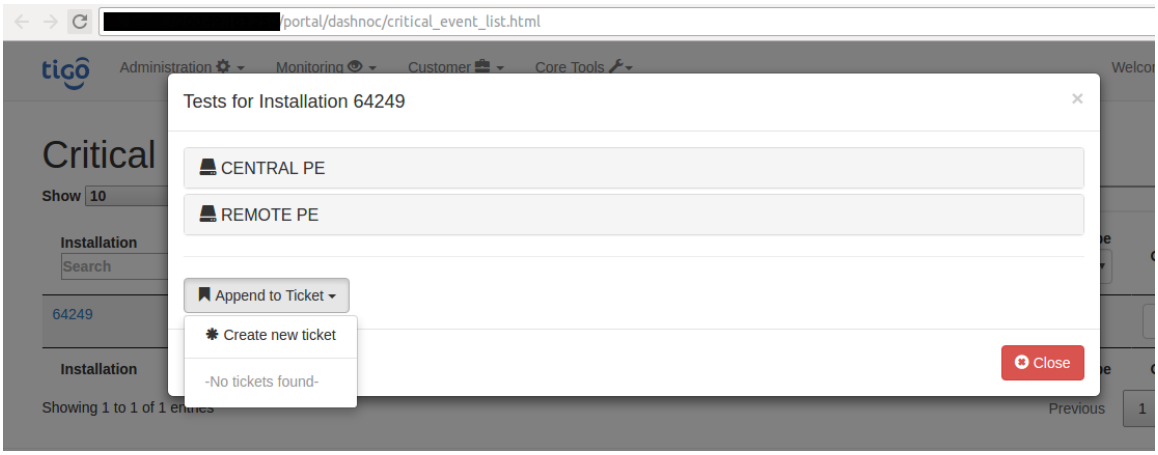


Figura 76. Opciones para el manejo de tickets.



VIII. DISCUSIÓN

El uso de herramientas para mejorar los procesos de atención de incidentes dentro de un NOC es una práctica bastante común. De hecho, existen diversas herramientas disponibles, generalmente de forma comercial, que ofrecen beneficios como el monitoreo de dispositivos y sus componentes principalmente por SNMP. Algunas de las herramientas más utilizadas son Cacti y Solarwinds.

Cacti es una solución bastante completa y de código abierto. Se enfoca principalmente en la generación de gráficas de los elementos de una red que se desea mantener en monitoreo. También es posible definir condiciones en las que se debe generar una alerta, aunque no cuentan con un dashboard en el cual se pueda clasificar la criticidad de los eventos detectados o agruparlos en una categoría particular. Una característica de la que goza Cacti es la opción para el envío de correos electrónicos, a manera de notificación de un problema en la red. Sin embargo, esta puede llegar a ser contraproducente, ya que cada vez que se realiza la rutina de monitoreo se estará generando un nuevo correo electrónico, principalmente en aquellas situaciones en las que el incidente es bastante tardado o cuando se tiene una actividad planificada.

Solarwinds es una herramienta más compleja de tipo comercial. Al igual que Cacti, permite el monitoreo de diversos elementos de una red y la generación de gráficas asociadas a los resultados de monitoreo. Una característica que es de gran ayuda en Solarwinds es la generación de mapas de monitoreos, que muestran diferentes lecturas (como rendimiento, consumo de tráfico, estado de interfaces) de los elementos de la red y el estado de los mismos. Este también ofrece una API desde la cual se puede acceder a la información de las alertas generadas y utilizarlas para la generación de tableros con otras herramientas, así como el envío de correos electrónicos. Sin embargo, la configuración de todos los elementos es manual, es decir, que se debe configurar cada uno de los elementos de forma individual y no es capaz de detectar cambios en la configuración del equipo.

Para el manejo de la configuración de los equipos también se tiene diversas herramientas, como RANCID y Junos Space. Estas se enfocan más que todo en la administración de la configuración de los equipos, detección de cambios y almacenamiento de configuraciones de respaldo. Estas herramientas, aunque son de naturaleza simple, son excelentes realizando las tareas para las que fueron destinadas. Junos Space, por su parte, también es capaz de realizar cambios programados, actualizaciones de software, entre otros. Pero el alcance no está destinado como una herramienta de monitoreo.

Este proyecto busca aprovechar las características de las herramientas de administración de configuración y, aplicarlas a una herramienta de monitoreo más inteligente, que además de facilitar la tarea de descubrimiento de configuraciones en un equipo, sea capaz de interpretar estas configuraciones y asociar los elementos de diferentes dispositivos que se encuentren asociados como parte de un mismo servicio.

Esta herramienta también se alimenta de la información disponible en la plataforma que maneja la información de clientes (aunque no se menciona como parte de los requerimientos, fue implementado más adelante) para mejorar la presentación de los datos y que sean fácilmente comprendidos por un ingeniero de NOC.

Es importante mencionar que el alcance real de proyecto es mayor a lo mostrado en este trabajo de graduación, pues existen más características de otras herramientas de monitoreo e ideas surgidas por parte de los involucrados en el proyecto, se pueden ser implementadas. Es importante resaltar que las mejoras no se limitan únicamente a la adición de más herramientas y características, sino que se debe evaluar aquellas mejoras que realmente ayudarán a mejorar el proceso de atención a clientes de un ISP.

También es importante no perder de vista que se busca la centralización de la información del estado de la red y de los servicios brindados, y puede ser de gran ayuda el aprovechamiento de la información que puede ser obtenida de herramientas de monitoreo terceras. Tal es el caso del deseo de implementar monitoreo sobre otras redes, como la red SDH y DWDM, que cuentan con su propia plataforma de gestión, a las cuales es posible solicitar información del estado o programar una comunicación asíncrona de las plataformas de gestión al sistema en desarrollo.

Aunque no se ha realizado un trabajo de evaluación para validar que el proyecto sea de gran ayuda en el proceso de solución de incidentes, se observa un importante aumento en la cantidad de casos proactivos abiertos en la herramienta de tickets. Estos casos se traducen en un diagnóstico más inmediato, mejores tiempos de disponibilidad de servicios y una diferenciación del servicio brindado por el ISP a sus clientes.

IX. CONCLUSIONES

1. Como principal logro se tiene la implementación de una aplicación Web orientada al monitoreo del estado de la red de un ISP y de los servicios VPN Capa 3 basados en BGP/MPLS brindados a los clientes del ISP.

2. Se ofrece la capacidad de visualización del estado de los servicios VPN Capa 3 basados en BGP/MPLS desde el listado de servicios, así como el detalle de estos y la generación de alertas al detectar un incidente con alta criticidad.

3. El sistema desarrollado permite la ejecución de pruebas de ping en tiempo real sobre los servicios VPN Capa 3 basados en BGP/MPLS.

4. Se facilita la revisión y diagnóstico de un servicio VPN Capa 3 para los ingenieros de NOC, al asociar los elementos de diferentes dispositivos de red como parte de un servicio y permitir la visualización centralizada de la información recopilada de estos elementos durante el monitoreo.

5. El tablero de alertas, con actualización automática, permite que los ingenieros de NOC puedan actuar de forma proactiva ante los incidentes detectados.

6. Se desarrolla una herramienta que facilita la implementación de conectividad a dispositivos de otros fabricantes, así como la adición de nuevas funcionalidades al sistema al trabajar con una arquitectura modular siguiendo el paradigma Modelo - Vista - Controlador o MVC.

7. Se permite la apertura de tickets desde el sistema desarrollado mediante el consumo de los servicios Web brindados por la herramienta de manejo de tickets.

X. RECOMENDACIONES

1. Resulta muy favorable acompañar el desarrollo del proyecto con un análisis de tipo cuantitativo que permita la validar la mejora de los procesos de atención a los clientes. Dentro de los valores recomendados están la disponibilidad de la red, disponibilidad de los servicios, tiempos de resolución de incidentes detectados y reportes de clientes y porcentaje de yickets generados de forma proactiva.

2. Los resultados de una evaluación cuantitativa, favorecen a que dentro de la organización exista una mejor aceptación en la implementación de este tipo de proyectos, permitiendo que se destine más recursos para su desarrollo.

3. Es importante considerar la implementación de monitoreo de otro tipo de servicios. Aunque los servicios VPN Capa 3 basados en BGP/MPLS conforman una de las soluciones más populares de un ISP para sus clientes, no representan el total de los servicios brindados.

4. Considerar la adición de pruebas adicionales a las prubas de Ping, para el diagnóstico de un Servicio VPN Capa 3 basado en BGP/MPLS, como la revisión del estado de las sesiones BGP, verificación de las rutas conocidas, verificación del consumo tráfico y tendencias en el consumo de tráfico.

5. Es importante validar la utilidad que tendría la implementación de una API en el sistema desarrollado que permita a herramientas externas acceder a la información del estado de la red y los servicios brindados.

XI. BIBLIOGRAFÍA

Blank, Andrew G. (2004). *TCP/IP Foundations*. Alameda, California: Sybex Inc. 283 págs.

Chase, Nicholas. (2006). *SOA and Web Services.* http://www.ibm.com/developerworks/webservices/tutorials/ws-understand-web-services1/ws-understand-web-services1.html [consultado el 25 de Septiembre de 2016].

Davies, Joseph, *et al*. (2008). *Windows Server 2008 Networking and Network Access Protection (NAP)*. Redmond, Washington: Microsoft Press. 816 págs.

Downing, Michael. (2013). *The Importance of Network Monitoring*. http://www.animate.com/the-importance-of-network-monitoring/ [consultado el 24 de Septiembre de 2016].

Etingof, Ilya. (2016). SNMP library for Python. http://pysnmp.sourceforge.net/ [consultado el 27 de Agosto de 2016].

Internet Assigned Numbers Authority. (2016a). *Autonomous Systems (AS) Numbers*. http://www.iana.org/numbers [consultado el 24 de Julio de 2016].

Internet Assigned Numbers Authority. (2016b). *IANA IPv4 Special-Purpose Address Registry*. http://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml [consultado el 9 de Julio de 2016].

Internet Assigned Numbers Authority. *Numbers Resources*. http://www.iana.org/numbers [consultado el 24 de Julio de 2016].

Internet Assigned Numbers Authority. (2016c). *Service Name and Transport Protocol Port Number Registry*. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml [consultado el 9 de Julio de 2016].

Internet Assigned Numbers Authority. (2015). *Special-Purpose Autonomous System (AS) Numbers*. http://www.iana.org/assignments/iana-as-numbers-special-registry/iana-as-numbers-special-registry.xhtml [consultado el 24 de Julio de 2016].

Internet Engineering Task Force. (2011). *Network Configuration Protocol (NETCONF)*. https://tools.ietf.org/pdf/rfc6241.pdf [30 de Agosto de 2016].

Jiménez Z., Ana. I. y Joan Torrent S. (2009). *Orientación proactiva hacia el cliente, cooperación y uso de las TIC: un análisis empírico sobre sus interrelaciones y efectos como potenciadores de la innovación en producto*. REVISTA INNOVAR. _19_ (33): 55-76.

Juniper Networks. (2010a). *Real-time Performance Monitoring on Juniper Network Devices.* California: Juniper Networks, Inc. 33 págs.

Juniper Networks. (2010b). *Junos MPLS and VPNs*. California: Juniper Networks, Inc.

Juniper Networks. (2012). *JNCIA - Junos Study Guide - Part 2*. California: Juniper Networks, Inc. 59 págs.

Juniper Networks. (2013a). *Autonomous-System*. http://www.juniper.net/documentation/en_US/junos13.1/topics/reference/configuration-statement/autonomous-system-edit-routing-options.html [consultado el 24 de Julio 2016].

Juniper Networks. (2013b). *JNCIS-SP Study Guide - Part 3*. California: Juniper Networks, Inc. 428 págs.

Juniper Networks. (2014a). *NETCONF XML Management Protocol and Junos XML API Overview*. https://www.juniper.net/techpubs/en_US/junos14.2/topics/concept/netconf-xml-protocol-and-junos-api-overview.html [consultado 17 de Septiembre de 2016].

Juniper Networks. (2014b). *Understanding Junos PyEZ.* https://www.juniper.net/techpubs/en_US/junos-pyez1.0/topics/concept/junos-pyez-overview.html [consultado 17 de Septiembre de 2016].

Lammle, Todd. (2013). *CCNA: Routing and Switching Study Guide*. Indianapolis: Sybex Inc. 1101 págs.

Mauro, Douglas y Kevin Schmidt. (2005). *Essential SNMP* (2nd. edition). California: O'Reilly Media, Inc. 442 págs.

Paquet, Catherine. (2012). *Implementing Cisco IOS Network Security (IINS 640-554) Foundation Learning Guide* (2nd. edition). Indianapolis: Cisco Systems Inc. 742 págs.

Sambasivam, Ganesh. *Extreme Programming (XP).* https://www.unf.edu/~broggio/cen6016/ExtremeProgramming(XP)Article.pdf [consultado el 27 de Septiembre de 2016]

Soricelli, Joseph M., *et al*. (2013). *Juniper Networks Certified Internet Associate.* California: Juniper Networks Inc. 594 págs.

Stallings, William. (1996). *SNMP, SNMPv2, and RMON: practical network management* (2nd edition). Michigan: Addison-Wesley Pub. Co. 478 págs.

Tanenbaum, Andrew S., David. J. Wetherall. (2012). *Redes de Computadoras* (5ta. edición). Traducción de Alfonso Romero. Revisión de Cyntia Enríquez. Mexico D.F.: Pearson Education. 791 págs.

Wells, Don. (2013). *Extreme Programming: A gentle introduction*. http://www.extremeprogramming.org/ [consultado el 27 de Septiembre de 2016].