

Gestión gráfica de dispositivos activos de red *multivendor*

Graphic Management of Active Multivendor Network Devices

James Rivera-Linares¹✉, Fredy Silva-Cubillos², Jairo Hernández-Gutiérrez³, Darin Mosquera-Palacios⁴

¹ Universidad Distrital Francisco José de Caldas, Bogotá, Colombia



² Universidad Distrital Francisco José de Caldas, Bogotá, Colombia



³ Universidad Distrital Francisco José de Caldas, Bogotá, Colombia



⁴ Universidad Distrital Francisco José de Caldas, Bogotá, Colombia



✉ Grupo de Investigación en Telemática Orión, Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, carrera 7 N.º 40B-53 Bogotá D.C. - República de Colombia. Correo electrónico: jhriveral@correo.udistrital.edu.co

Recibido: 5 de septiembre del 2017 **Aprobado:** 1 de diciembre del 2017 **Disponible en línea:** 1 de enero del 2018

How to cite this article: J. Rivera-Linares, F. Silva-Cubillos, J. Hernández-Gutiérrez, D. Mosquera-Palacios, "Gestión gráfica de dispositivos activos de red multivendor", *Revista Ingeniería Solidaria*, vol. 14, no. 24, pp. 11, enero 2018. doi: <https://doi.org/10.16925/in.v14i24.2179>

Resumen

Introducción: este artículo muestra el resultado del trabajo "Prototipo para la administración gráfica de dispositivos activos de red en ambientes *multivendor* por medio de una plataforma de gestión única", realizado en el grupo de investigación Orión de la Facultad Tecnológica de la Universidad Distrital Francisco José de Caldas, en el 2013. El trabajo mismo ha sido analizado y actualizado con los resultados de las últimas investigaciones sobre el tema hasta el 2017. Aborda el monitoreo y la gestión de red, actividades esenciales para mantener y controlar los entornos de TI que cada vez son más complejos y heterogéneos.

Metodología: se describe la construcción del prototipo de *software* denominado DeviceView, el cual permite a los administradores de red gestionar de forma gráfica dispositivos activos de red (tipo *switch*) que se encuentren en una LAN.

Resultados: el administrador de red no tiene que manipular cada una de las herramientas de gestión de cada fabricante. A fin de facilitar y centralizar la tarea al administrador de red el DeviceView está integrado al NMS de código abierto Nagios.

Conclusión: DeviceView se convierte en una herramienta unificada de gran utilidad para los administradores de red que trabajan en un ambiente *multivendor*, como los que hoy se tienen en las organizaciones.

Originalidad: hasta ahora no se habían realizado proyectos de gestión unificada basados en herramientas *open source* en un ambiente *multivendor*.

Limitaciones: el DeviceView en su primera versión incluye 36 identificadores de objeto, pero se puede ampliar para soportar otras marcas y familias de *switches*.

Palabras clave: código abierto, conmutador de red, dispositivo activo, gestión de red, multivendor.



Graphic Management of Active Multivendor Network Devices

Abstract

Introduction: This article shows the result of the research “Prototype for the graphic management of active network devices in multivendor environments by means of a single management platform” conducted by the Orión research group at the School of Technology, Universidad Distrital Francisco José de Caldas in 2013. The work itself has been analyzed and updated with the results of the latest research on the subject until 2017. It covers network monitoring and management, which are essential operations to maintain and control IT environments that are increasingly complex and heterogeneous.

Methods: Construction of the software prototype called DeviceView is described, which allows network administrators to graphically manage active network devices (switch type) in a LAN.

Results: The network administrator does not have to handle each manufacturer's management tool. In order to facilitate and centralize the task for the network administrator, DeviceView is integrated into the open source NMS Nagios.

Conclusion: DeviceView is a very useful unified tool for network administrators working in a multivendor environment, such as those that organizations have at present.

Originality: Until now, there have been no unified management projects based on open source tools in a multivendor environment.

Limitations: DeviceView in its first version includes 36 object identifiers, but it can be extended to support other switch brands and families.

Keywords: Open source, network switch, active device, network management, multivendor.

Gestão gráfica de dispositivos ativos de rede *multivendor*

Resumo

Introdução: este artigo mostra o resultado do trabalho “Protótipo para a administração gráfica de dispositivos ativos de rede em ambientes *multivendor* por meio de uma plataforma de gestão única”, realizado no grupo de pesquisa Orión da Faculdade Tecnológica da Universidad Distrital Francisco José de Caldas, em 2013. O trabalho em si tem sido analisado e atualizado com os resultados das últimas pesquisas sobre o tema até 2017. Aborda a monitoração e a gestão de rede, atividades essenciais para manter e controlar os ambientes de TI que cada vez são mais complexos e heterogêneos.

Metodologia: descreve-se a construção do protótipo de *software* denominado DeviceView, o qual permite aos administradores de rede gerir de forma gráfica dispositivos ativos de rede (tipo *switch*) que se encontrem numa LAN.

Resultados: o administrador de rede não tem que manipular todas as ferramentas de gestão de cada fabricante. A fim de facilitar e centralizar a tarefa ao administrador de rede, o DeviceView está integrado ao NMS de código aberto Nagios.

Conclusão: o DeviceView torna-se uma ferramenta unificada de grande utilidade para os administradores de rede que trabalham num ambiente *multivendor*, como os que hoje se têm nas organizações.

Originalidade: até agora, não têm sido realizados projetos de gestão unificada baseados em ferramentas open source num ambiente *multivendor*.

Limitações: o DeviceView, em sua primeira versão, inclui 36 identificadores de objeto, mas pode ser ampliado para suportar outras marcas e famílias de *switches*.

Palavras-chave: código aberto, comutador de rede, dispositivo ativo, gestão de rede, *multivendor*.

1. Introducción

La administración de redes, según Clemm [1], involucra tareas como el mantenimiento de la infraestructura, el flujo de información constante y los servicios de red operativos en todo momento, lo cual implica la utilización de técnicas administrativas y de informática enfocadas al monitoreo continuo de todos los dispositivos activos que la componen. Para Groves y Arnold [2], al realizar esta labor el administrador de red debe realizar varias tareas tales como la detección y corrección de fallos, la monitorización del rendimiento y la detección de cuellos de botella, entre otras, lo cual le permita asegurar aspectos determinantes para la estabilidad y continuidad de una red como, por ejemplo, su control de incidentes, la actualización, la planificación, la seguridad, la integridad y la confiabilidad de los datos que circulan por ella. El ingeniero de soporte debe tener a mano la información exacta que le ayude a resolver cualquier problema que se presente en el menor tiempo posible, lo que garantiza el crecimiento del sistema de una manera controlada. Cuando las redes de gran magnitud cuentan con dispositivos activos de red de varios fabricantes (ambientes *multivendor*) [3], la gestión se aísla y la administración para la infraestructura de datos se torna compleja al no contar con un sistema de gestión de red único provisto por los fabricantes de los dispositivos. Una administración eficiente es entonces una necesidad creciente para las redes empresariales; no contar con un sistema de gestión de red implica desmejoras en la calidad del servicio que presta una red corporativa y afecta el concepto de alta disponibilidad que debe tener, al igual que el seguimiento y monitoreo constante del cual siempre es objeto una red corporativa.

Como lo expresa Stallings [4], para la ejecución de tareas administrativas es indispensable el uso del protocolo SNMP (protocolo de gestión simple de redes) [5]-[9], el cual permite el acceso a la MIB (base de datos de la información gestionada) que contiene los objetos OID (identificador de objeto) susceptibles de ser gestionados dentro de los equipos activos de la red y el protocolo RMON (monitoreo remoto) versión 1 (esta es básicamente una extensión de la MIB del protocolo SNMP).

1.1 Antecedentes de investigación

Con el ánimo de establecer el estado del arte, antes de empezar este trabajo se investigó sobre la existencia de proyectos o investigaciones relacionadas con este tema concreto en otras universidades o centros de investigación sin obtener resultados positivos en las bases de datos consultadas, lo cual se convirtió en un aspecto motivante para su desarrollo. Sin embargo, es importante resaltar que en el mercado de la industria de las tecnologías de la Información existen herramientas de *software* que permiten gestionar y monitorear las redes de datos en ambientes *multivendor*, tales como Tivoli [10] de IBM; ProactiveNet Performance Manager [11] de BMC; Unified Infrastructure Management [12] de Computer Associates (CA); OP Manager [13] de ManageEngine como división de IT de ZOH Corporation; y Magellan NMS Network Management System [14] de Imagine Communications. Sin embargo, estas soluciones son costosas y se deben licenciar para usarse en ambientes empresariales o académicos. También se debe considerar que algunas herramientas de gestión y monitoreo de algunos fabricantes como Cisco, Hewlett Packard o Dell permiten administrar dispositivos de otros fabricantes, siempre y cuando se cuente con los OID en la MIB para cada caso en particular, lo cual se torna engorroso y dispendioso de implementar.

El trabajo de grado se llevó a cabo como parte de los proyectos desarrollados por el Grupo de Investigación Orión de la Universidad Distrital Francisco José de Caldas en el 2013, y en este documento se han actualizado las referencias y fuentes bibliográficas con el fin de analizar e incluir los resultados de las últimas investigaciones sobre el tema hasta el 2017; si bien no existen muchos trabajos publicados, se citan algunos a continuación.

Se revisó un proyecto de un prototipo realizado para la aplicación de agentes de *software* en gestión de redes, en el cual se presenta el diseño y el desarrollo de un conjunto de componentes que facilitan las funciones de monitoreo de dispositivos de red para garantizar al usuario una adecuada experiencia de navegación desde clientes inalámbricos [15]. Sin embargo, aunque el trabajo es sobre gestión de redes, su enfoque es sobre la gestión de dispositivos inalámbricos en un ambiente web.

Martínez, Yannuzzi y López [16] examinan en detalle los desafíos de interoperabilidad de la gestión de redes multicapa y proveedores *multivendor*, así como revisan las tendencias actuales y los estándares recientes en el área con un fuerte enfoque en los avances industriales. También discuten algunas de las razones por las cuales las propuestas de gestión de nivel de operador relevantes no han podido cumplir con los requisitos de los proveedores de servicios de Internet (ISP), e identifican un conjunto de características que pueden ayudar a allanar el camino para comercializar nuevos productos de administración. Es un enfoque de los desafíos de la gestión desde el punto de vista de redes WAN en un entorno de múltiples fabricantes, el cual no ha sido satisfecho con las soluciones actuales del mercado.

Para Solarwinds [17], sin importar si se está administrando una red pequeña o grande, lograr la eficiencia y reducir el costo de mantenimiento debe ser un objetivo del administrador. La infraestructura de red heterogénea puede presentar desafíos cuando las operaciones dinámicas se realizan a través de los sistemas, pero el uso de una herramienta NMS que admita múltiples proveedores definitivamente puede ser útil para los ingenieros durante la creación de la red y el proceso de expansión. Esto significa que contar con una consola central única para monitorear todos los dispositivos es lo ideal, concepto que refuerza el propósito definido para DeviceView.

DeviceView está integrado al sistema de gestión de red de código abierto Nagios [18], al extender las funcionalidades de este para proporcionar la tarea de administrar de manera gráfica dispositivos activos de red de varios fabricantes en una plataforma de gestión única del DeviceView.

En la sección de metodología se describen los requerimientos funcionales y no funcionales del software DeviceView, las herramientas y la tecnología que hicieron parte de la construcción de la solución, el número de OID seleccionadas que limitan el número de elementos gestionables y la elección del paquete de SNMP escritos en el lenguaje Java. Asimismo, se muestra un fragmento de código fuente que define una de las funcionalidades de la solución y en la parte final de la sección se indican unas consideraciones de diseño. La sección de resultados describe el módulo DeviceView y la estructura funcional de la solución al estar vinculada a Nagios para proporcionar facilidad en la gestión de la red a sus administradores. Finalmente, se

muestran las características y ventajas de la utilización del DeviceView.

2. Metodología

Utilizando el enfoque de la metodología RUP de Rational Software [19] se estableció que DeviceView debería permitir la centralización de la administración de los dispositivos activos de red, de forma escalable, segura, disponible, confiable, con soporte *multivendor* integrado a una herramienta de monitoreo adecuada y que el *software* en su totalidad fuera de carácter *open source* para garantizar su uso y su estudio, así como sus actualizaciones y mejoras. Las características propias que se tuvieron en cuenta dentro de los requerimientos no funcionales para el desarrollo del DeviceView fueron: la selección del sistema gestor de red, la selección de las OID a utilizar, la obtención del paquete de comandos SNMP, la construcción del DeviceView, el diseño de la interfaz gráfica y la integración con Nagios.

Se analizaron y estudiaron las herramientas NMS que actualmente operan en redes LAN; Nagios fue seleccionado como el sistema gestor de red que integró en su estructura visual las funcionalidades del DeviceView debido a varias características que viabilizan su utilización, entre las cuales las más importantes son: su licencia GNU de código abierto (incluyendo toda la plataforma tecnológica en la que funciona), arquitectura cliente-servidor ampliamente utilizado por los administradores de red, permite visualizar un mapa de red completo, monitorear *hardware* y *software* (servicios de red) en una LAN, generar informes y gráficos estadísticos, facilidad para el diseño de *plugins*, complementos u otros módulos en diferentes lenguajes de programación.

En su primera versión, el DeviceView soporta dos perfiles de usuario: uno es el administrador de red, quien lleva la responsabilidad de realizar tareas de monitoreo y gestión de los dispositivos de red, y el otro perfil es el superadministrador, quien tiene a su disposición las mismas funcionalidades del administrador de red, y adicionalmente puede administrar otros usuarios, es decir, crear, actualizar y eliminar usuarios. El superadministrador solo puede ser uno (quien accede por primera vez al sistema), mientras que el número de los administradores de red pueden ser ilimitado.

Los requisitos funcionales del sistema se representan a través los casos de uso. La figura 1 muestra 14 de los 26 casos de uso para el perfil Administrador de red.

Se determinó el uso de tres familias específicas de dispositivos activos de las marcas Cisco, HP Networking y Allied Telesis, tomando en consideración que, para efectuar tareas de consulta, gestión y monitoreo de los dispositivos a través de los protocolos SNMP y RMON, se debe conocer la dirección única dentro del árbol MIB que identifica cada uno de los objetos OID susceptibles de ser accedidos para identificar el dispositivo, conocer su estado y efectuar las tareas antes mencionadas.

La figura 2 muestra las 36 OID que satisfacen el alcance del proyecto; estos identificadores de objeto están divididos en cinco grupos, las OID de uso general, tareas RMON y un grupo por cada una de las marcas usadas en el proyecto.

Como se puede apreciar, el DeviceView en su primera versión incluye un listado de 36 OID; por lo tanto, si se requiere ampliar el alcance de la herramienta y soportar otros dispositivos activos de tipo *switch* gestionable, el DeviceView debe tener los listados de las OID por cada uno de los objetos susceptibles de ser gestionados.

Al aplicar el proceso RUP y adaptar sus fases a los requerimientos específicos del proyecto, se analizaron y diseñaron diagramas UML [20] como el de clases, flujo, secuencia, actividad y otros, los cuales permiten especificar el desarrollo del *software* y los requerimientos funcionales de la solución.

El modelado de las clases y paquetes del DeviceView mediante diagramas permite definir la estructura del código fuente, en el cual se programan los *scripts* con las funcionalidades finales del DeviceView.

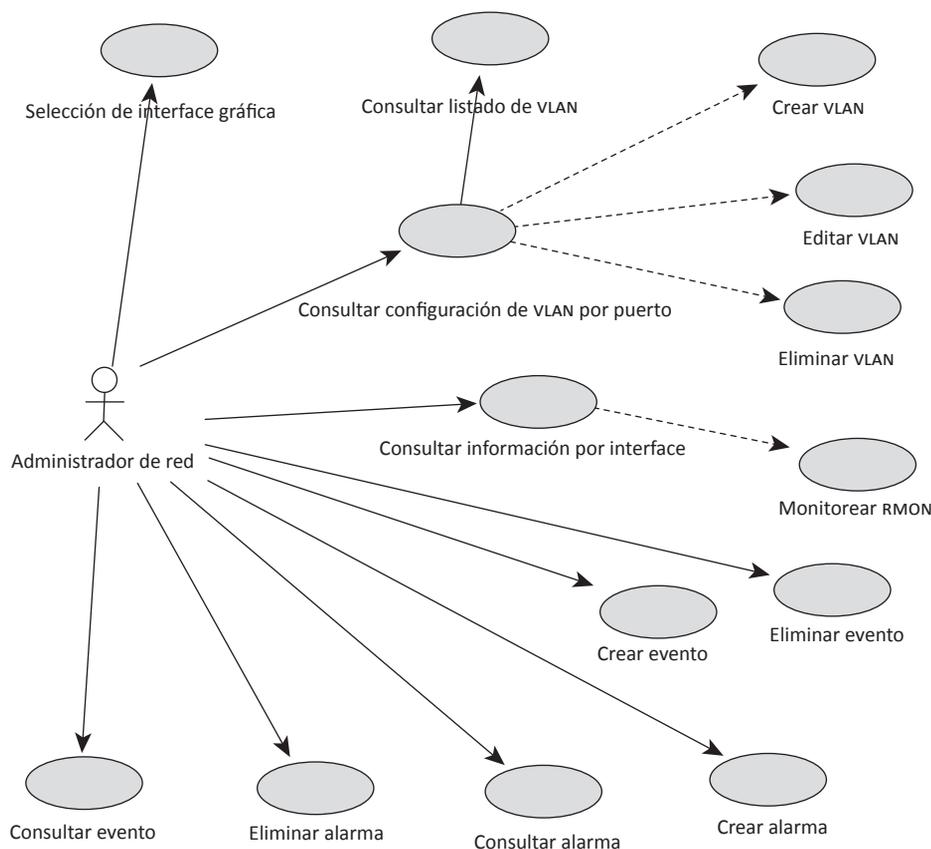


Figura 1. Casos de uso parciales del usuario Administrador de red

Fuente: elaboración propia



Figura 2 . Listado de OID usadas por el DeviceView
Fuente: elaboración propia

Como insumo fundamental del desarrollo se incluyó el paquete SNMP Package compuesto por métodos ya definidos que permiten ejecutar comandos SNMP, los cuales realizan acciones específicas a cada uno de los dispositivos activos de red. La lógica programada en el DeviceView permite guiar al usuario administrador de red desde el proceso de selección del dispositivo, pasar por la ejecución de acciones indicadas por el usuario y culminar en el registro de auditoría o trazabilidad de las acciones realizadas por el usuario.

Las figuras 3 y 4, en su orden respectivo, muestran un diagrama de actividades (diagrama de flujo) y la implementación de estos dos diagramas representados en un fragmento de su código fuente escrito en lenguaje Java. En el diagrama de actividades se muestra el procedimiento que debe realizar un usuario para crear una VLAN; en el código fuente se muestra la función “numero_vlan()” que construye los elementos propios del formulario respectivo, valida la existencia de los parámetros ingresados por el usuario y guarda los cambios en el dispositivo. Hay otros diagramas como, por ejemplo, el diagrama de secuencia, el cual muestra la interacción

de mensajes entre las instancias de las clases, componentes y actores de una forma secuencial.

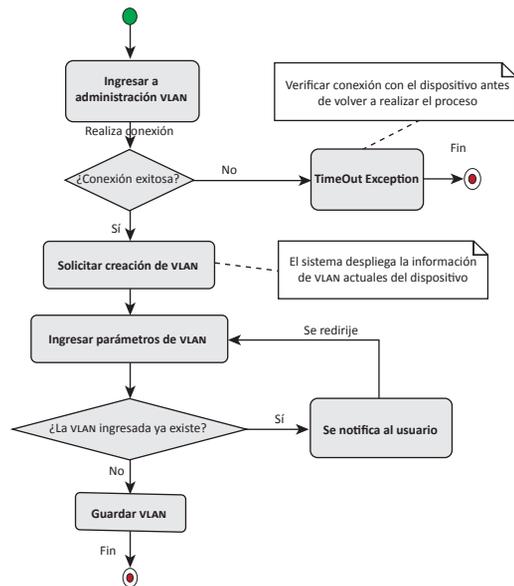


Figura 3. Diagrama de flujo “Crear VLAN”
Fuente: elaboración propia

```

public void numero_vlan(){
    this.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
    JScrollPane scroll = new JScrollPane();
    final JPanel etiqueta = new JPanel(new GridBagLayout ());
    final GridBagConstraints constraints = new GridBagConstraints();
    JLabel titulo=new JLabel("Crear VLAN");
    constraints.gridx=0; constraints.gridy=0; constraints.gridheight=1; constraints.gridwidth=3;
    constraints.insets = new Insets(20, 20, 20, 20); etiqueta.add(titulo, constraints);
    JButton btn1=new JButton("Crear VLAN");
    constraints.gridx=2; constraints.gridy=2; constraints.gridheight=1; constraints.gridwidth=1;
    constraints.insets = new Insets(20, 20, 20, 20); etiqueta.add(btn1, constraints);
    etiqueta.revalidate();
    btn1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        validar_vlan obj = null; Construye con = null; String desc;
        try {
            con=new Construye();
            obj = new validar_vlan();
        } catch (SQLException ex) {
            Logger.getLogger(inter_crear_vlan.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(inter_crear_vlan.class.getName()).log(Level.SEVERE, null, ex);
        }
        boolean pregunta=obj.validar(num_id.getText().trim()); // ¿existe esta vlan?
        System.out.println("RESPUESTA==> "+pregunta);
        if(pregunta==true){
            JOptionPane.showMessageDialog(null, "ID asignado");
        }else{
            try {
                Crear_vlan obj2=new Crear_vlan(num_id.getText().trim(), nom_id.getText().trim());
                desc = "Creada VLAN nombre: "+nom_id.getText().trim()+" , numero de VLAN: "+num_id.getText().trim()+" ";
                String res; res = con.inserta_registro(username,direccionIP,"Crear",desc);
            } catch (SQLException ex) {
                Logger.getLogger(inter_crear_vlan.class.getName()).log(Level.SEVERE, null, ex);
            } catch (ClassNotFoundException ex) {
                Logger.getLogger(inter_crear_vlan.class.getName()).log(Level.SEVERE, null, ex);
            }
            JOptionPane.showMessageDialog(null, "VLAN creada");
        } //fin else
    } //fin action btn1
    });
    this.getContentPane().add(scroll);
    scroll.setViewportView(etiqueta);
    this.setPreferredSize(new Dimension(450, 300));
    this.setTitle("Configuración VLAN");
    this.pack();
    this.setVisible(true);
}

```

Figura 4. Fragmento de código fuente para la creación de VLAN
Fuente: elaboración propia

La arquitectura de alto nivel del proyecto fue representada en un diagrama de componentes (figura 5), el cual muestra la estructura y la relación entre los componentes o elementos de diseño del DeviceView a un alto nivel. Así, en la capa de aplicación se tiene tanto el protocolo HTTP (para la presentación web de la interfaz de usuario y del Applet que permite desplegar las funcionalidades del DeviceView) como Nagios en conjunto con el DeviceView (que utilizan los protocolos SNMP y RMON a fin de acceder a los objetos gestionables de los dispositivos activos). Por último, se encuentra como otro componente el árbol de objetos OID del dispositivo activo y administrable.

El desarrollo como tal fue realizado como un proyecto Java Applet [21], ya que su finalidad es lograr un acceso por parte de los usuarios a través de un cliente o navegador web. Asimismo, el desarrollo se realizó bajo el paradigma de la orientación

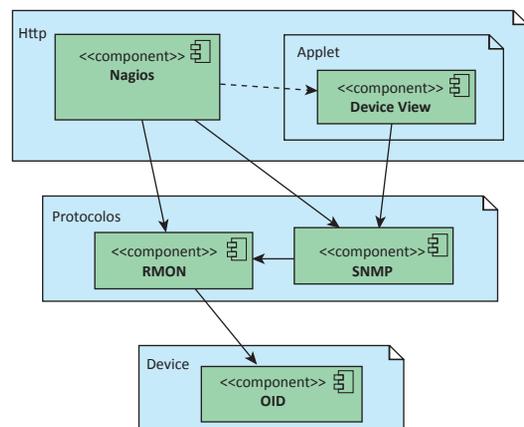


Figura 5 . Diagrama de componentes
Fuente: elaboración propia

a objetos y el patrón de arquitectura de *software* Modelo Vista-Controlador, a fin de que cada fragmento de código con una función determinada se encuentre claramente separado en el manejo o acceso a datos y a los dispositivos mediante SNMP y RMON (modelo), la lógica o reglas de la aplicación (controlador) y la parte que es visible por el usuario (vista).

El Applet debió ser firmado digitalmente para que el cliente habilitara su contenido y el usuario interactúe con el DeviceView. La firma digital es un proceso criptográfico para la distribución del Applet mismo; esta firma garantiza a los destinatarios que el Applet proviene de una fuente de confianza y que su código fuente no ha sido alterado o manipulado por terceros.

3. Resultados

El *software* resultante denominado DeviceView es un Applet escrito en el lenguaje de programación Java, el cual está firmado digitalmente para ser ejecutado en un navegador web. Con el fin de incrementar las capacidades de gestión a los administradores de red, el DeviceView fue incorporado a la interfaz gráfica de Nagios y se presenta como una más de sus funcionalidades; para tal efecto, el módulo está dividido en dos opciones: una permite el acceso a la administración de dispositivos activos de red, y la otra lleva el usuario a la configuración de parámetros generales.

El DeviceView muestra la imagen del dispositivo a gestionar (como se puede apreciar en la figura 6), lo que facilita al administrador de red manipular los puertos gestionables del dispositivo y acceder a la gestión de este. La imagen del *switch* desplegada depende del resultado de la ejecución de una primitiva SNMP realizada a la dirección IP del dispositivo indicado por el usuario; el resultado de esta OID obtenida es comparada con la OID definida previamente en el DeviceView que corresponde a la imagen de uno u otro *switch* (de 24 o 48 puertos), en caso de no encontrar ninguno de los *switch* de las marcas: Hp Networking (3com) (etiqueta 1 y 2), Allied Telesis (etiqueta 3 y 4) y Cisco (etiqueta 5 y 6) definidos previamente, el sistema muestra una imagen de un *switch* genérico (etiquetas 7 y 8).

El DeviceView extiende las capacidades de Nagios y permite a los administradores de red gestionar de manera gráfica familias específicas de equipos activos de las marcas Cisco, HP Networking

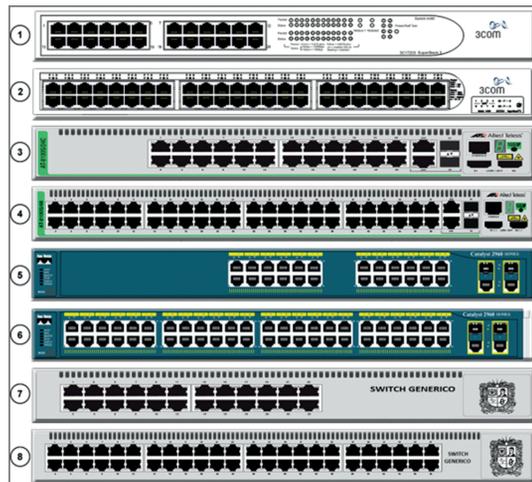


Figura 6 . Diseño de la interfaz gráfica del DeviceView

Fuente: elaboración propia

y Allied Telesis. Entre las tareas administrativas que podemos ejecutar en los equipos activos se encuentran:

- Administrar los segmentos lógicos de la red de los dispositivos activos. Las tareas administrativas relacionadas con la configuración de VLAN que ejecuta son: visualización de las VLAN existentes, creación y eliminación de VLAN, edición de VLAN como cambiar nombre, adición y eliminación de puertos a la VLAN.
- Mostrar información del sistema referente al equipo activo a administrar; dicha información corresponde a: nombre del *host*, descripción del sistema, estado administrativo, estado operativo y espacio disponible en memoria *flash*.
- Utiliza el protocolo RMONV1 para el monitoreo de red, abordando los grupos: estadísticas, historia, alarma y eventos.
- Crear eventos y alarmas del protocolo RMON, a fin de monitorear la utilización de la red y optimizar el rendimiento, así como asistir en el diagnóstico de fallos de la red.
- Acceder a los datos individuales de cada puerto dentro de los equipos activos de red, visualizando los siguientes datos: dirección física, velocidad y tabla ARP.
- Administración del área funcional de “gestión de cambios”, para lo cual implementa un sistema que permite visualizar: fecha de instalación del dispositivo, características de su configuración, puertos que estén presentando fallas físicas.

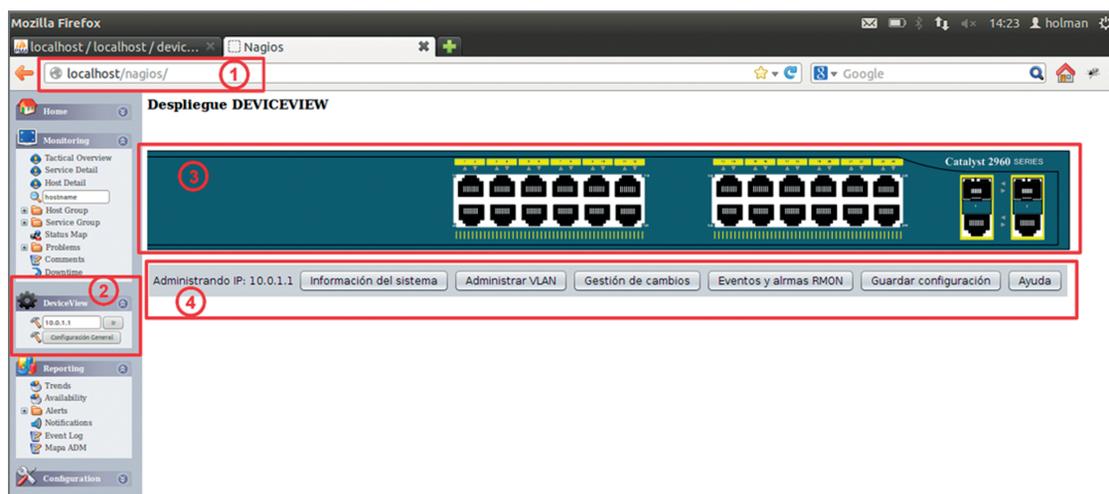


Figura 7. Despliegue del módulo DeviceView

Fuente: elaboración propia

La interfaz gráfica ofrecida es muy amigable para el usuario. A fin de usar el módulo DeviceView, el administrador de red debe ingresar a la dirección del servidor de Nagios (véase recuadro 1), luego debe ingresar a la configuración general del módulo (véase recuadro 2) para establecer algunos parámetros iniciales como el nombre de la comunidad, establecer ubicación del archivo de configuración, cargar los dispositivos activos definidos en Nagios (véase la figura 7) y administrar usuarios. Con el propósito de realizar la administración de los dispositivos activos el usuario administrador de red debe monitorear frecuentemente los servicios y dispositivos de red desde las herramientas dispuestas por Nagios; cuando el administrador de red lo requiera, puede ingresar la dirección IP del dispositivo que necesite gestionar y el sistema carga la interfaz correspondiente al dispositivo (véase recuadro 3), habilitando la selección gráfica de puertos y las demás herramientas de la barra de menú (véase recuadro 4).

```

1 # Define the switch that we'll be monitoring
2
3 define host{
4     use         generic-switch      ; Inherit default values from a template
5     host_name   Sw_piso1            ; The name we're giving to this switch
6     alias       Switch_piso_1      ; A longer name associated with the switch
7     address    10.0.1.50           ; IP address of the switch
8     hostgroups  switches           ; Host groups this switch is associated with
9 }

```

Figura 8. Fragmento de código de definición de switch en Nagios

Fuente: elaboración propia

Por medio de esta interfaz el usuario puede realizar toda la administración del dispositivo activo consultado. El prototipo ha sido probado con éxito en ambientes prácticos de redes LAN con los dispositivos de las marcas y referencias mencionadas para realizar las siguientes actividades: ingreso al sistema, selección de la interfaz gráfica, consultar información del sistema, consultar, crear, eliminar y configurar VLAN, consultar información por interfaz, monitoreo RMON, gestión de usuarios y contraseñas, establecer ubicación del fichero Switch.cfg y gestión de acceso al sistema, entre otros.

El trabajo de grado que hace parte del grupo de investigación Orión obtuvo una calificación de 286/300, que equivale a 4,933 y fue reconocido como tesis meritosa.

4. Discusión y conclusiones

De acuerdo con los antecedentes de la investigación y la revisión bibliográfica realizada, la infraestructura de red en ambientes heterogéneos puede presentar desafíos cuando las operaciones dinámicas se realizan a través de los sistemas, pero el uso de una herramienta NMS que admita múltiples proveedores, definitivamente puede ser de gran utilidad para los ingenieros durante la implementación y operación de la red y su proceso de expansión.

Las infraestructuras de red se vuelven cada día más complejas a fin de estar en capacidad de soportar

la operación del negocio y sus requerimientos. En entornos de red con dispositivos activos *multivendor*, para efectuar tareas como administrar VLAN y puertos, crear eventos y alarmas RMON y otros mencionados en los resultados, los administradores de red deben acceder a cada dispositivo a través del *software* propietario de cada fabricante, o bien pueden acceder por medio de la línea de comandos haciendo uso del protocolo SNMP y RMON, lo que implica pérdida de tiempo y de oportunidad en la gestión, lo cual puede ser determinante para la continuidad de la operación de los servicios de red.

La administración de una red requiere de una herramienta que no solo proporcione monitoreo y gestión centralizada, sino que incorpore la administración de dispositivos activos de red configurables para evitar utilizar diferentes herramientas de gestión, una por cada fabricante de los dispositivos. El uso del DeviceView, adherido al NMS Nagios, confiere al administrador de red y a la organización que lo utiliza capacidades para la gestión gráfica de dispositivos y monitoreo de servicios y recursos de manera centralizada, al utilizar una sola interfaz de usuario y de manera oportuna, así como al atender los incidentes de red y reduciendo el tiempo apropiado para el aprendizaje y acceso a cada herramienta propietaria.

DeviceView permite establecer mecanismos de control, eventos, alarmas y seguimiento en un repositorio de datos, mediante el uso de una amigable interfaz gráfica para la administración de equipos activos, lo cual proporciona ayuda para la gestión y facilita su configuración.

En el uso la herramienta se debe tener en cuenta que los dispositivos gestionados se deben encontrar dentro de una misma comunidad SNMP para su administración, y que el prototipo está pensado para la administración de redes en las que no existe un único NMS, debido a que utiliza el protocolo SNMPv2. Adicionalmente, se debe considerar que las tareas de monitoreo son realizadas consultando las OID correspondientes a la extensión de MIB-RMONv1.

DeviceView solo puede gestionar los dispositivos activos definidos en su alcance —como se describió en la parte de “Metodología” y de “Resultados”—, por lo tanto, se debe trabajar como una oportunidad de mejora en su desarrollo, a fin de ampliar sus funcionalidades, especialmente en las fuentes y en la obtención y forma de almacenamiento de las OID de los dispositivos, y así soportar

un mayor número de equipos activos de diferentes marcas y familias que se puedan gestionar.

Referencias

- [1] A. Clemm. *Network management fundamentals*. Indianapolis: Cisco Systems Inc., Cisco Press, 2006, pp. 47-48.
- [2] T. Groves, D. Arnold y Y. He, “In-network, push-based network resource monitoring: scalable, responsive network management”, en *Proceedings of the Third International Workshop on Network-Aware Data Management (NDM '13)*, Nueva York, Article 8, 6 pp. (2013), [En línea]. doi: <http://doi.acm.org/10.1145/2534695.2534704>
- [3] IAP Solutions, “Multivendor Network Management”, *Iapsolutions.com*, 2017. [En línea]. Disponible en: <http://www.iapsolutions.com/es/blog-iapsolutions/Entrada/31-multivendor-network-management>
- [4] W. Stallings, *SNMP, SNMPV2, SNMPV3 and RMON 1 and 2*. Madrid: Addison Wesley, 2006, pp. 171-178.
- [5] D. Mauro y K. Schmitd. *Essential SNMP*. O'Reilly Media, 2009, pp. 23-28. [En línea]. Disponible en: <http://shop.oreilly.com/product/9780596008406.do>
- [6] C. A Pattinson, “Study of the behaviour of the Simple Network Management Protocol”, en *Proceedings of the 12th IFIP/IEEE Workshop on Distributed Systems: Operations and Management (DSOM 2001)*, pp. 1-6, 2001. doi: <http://dx.doi.org/10.3990/2.26>
- [7] R. Presuhn. *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*. RFC, 2002, pp. 2-10. [En línea]. Disponible en: <https://tools.ietf.org/html/rfc3416>
- [8] J-S. Lee y P-L, “Hsu Design and implementation of the SNMP agents for remote monitoring and control via UML and Petri nets”, en *IEEE Transactions on Control Systems Technology*, pp. 293-302, 2004. [En línea]. Disponible en: <https://pdfs.semanticscholar.org/c339/742976e474b8cac1ddef-942355d921a323b5.pdf>
- [9] J. Schönwälder, A. Pras, H. Matus, J. Schippers, R. van de Meent, “SNMP traffic analysis: approaches, tools, and first results”, en *Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007)*, pp. 323-332, Munich, 2007. [En línea]. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.497.3099&rep=rep1&type=pdf>
- [10] Ibm.com, “Tivoli NetView for z/os”, *Ibm.com*, 2017. [En línea]. Disponible en: <http://www-03.ibm.com/software/products/es/tivoli-netview-zos>.

- [11] Grupo Arión, “Monitoreo de infraestructura de TI con BMC ProactiveNet Performance Manager”, *Grupoarion.com*, 2017. [En línea]. Disponible <http://www.grupoarion.com.mx/soluciones-bmc/monitoreo/>
- [12] CA Technologies, “Unified Infrastructure Management”, *Ca.com*, 2017. [En línea]. Disponible en: <https://www.ca.com/ar/products/ca-unified-infra-structure-management.html>
- [13] ManageEngine, “OP Manager”, *Manageengine.com*, 2017. [En línea]. Disponible en: <https://www.manageengine.com/latam/network-monitoring/>
- [14] Imagine Communications, “Magellan NMS Network Manager”, 2017. [En línea]. Disponible en <http://www.imaginecommunications.com/products/networking/command-control/magellan-nms-network-management-system>
- [15] R. Rodríguez, J. Barón y C. Montenegro, “Agentes de software aplicado a gestión de redes basada en web”, *Revista Redes de Ingeniería*, vol. 4, no. 2, 2013. [En línea]. Disponible en: <http://revistas.udistrital.edu.co/ojs/index.php/REDES/article/view/5929/7437>
- [16] A. Martínez, M. Yannuzzi y V. López. *Network management challenges and trends in multi-layer and multi-vendor settings for carrier-grade networks*. IEEE Publications and Tutorials, 2014. pp. 2207-2230. doi: <https://doi.org/10.1109/COMST.2014.2327754>
- [17] Thwck Solarwinds IT Community, “Complexities of multi-vendor network environments and how to manage them!”, *Solarwinds.com*, 2014. [En línea]. Disponible en: https://thwack.solarwinds.com/community/solarwinds-community/geek-speak_tht/blog/2014/01/08/complexities-of-multi-vendor-network-environments-and-how-to-manage-them
- [18] Nagios, “The industry standard in IT infrastructure monitoring”, *Nagios.org*, 2017. [En línea]. Disponible en: <https://www.nagios.org/>
- [19] Metodoss, “Metodología RUP”, *Metodoss.com*, 2017. [En línea]. Disponible en: <https://metodoss.com/metodologia-rup/>
- [20] J. Rumbaugh, I. Jacobson y G. Booch. *El lenguaje unificado de modelado: manual de referencia*. Madrid: Addison Wesley, 1999, pp. 21-34. [En línea]. Disponible en: <https://univdelosandes.on.worldcat.org/oclc/964825492>
- [21] J. S. Harbour. *Beginning Java game programming*. 2ª ed., Boston, MA: Thompson Course Technology, pp. 17-25, 59-50, 2008. [En línea]. Disponible en: <https://univdelosandes.on.worldcat.org/oclc/290576467>