

**Universidad Autónoma Metropolitana
Unidad Azcapotzalco**

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

Reporte del Proyecto

**“Diseño e Implementación de una Aplicación para
la Administración del una Red en el SO Linux”
(Complemento)**

Asesor

Arturo Zúñiga López

Coordinador de Carrera

Francisco Javier Zaragoza Martínez

Alumno

Heladio Arteaga Franco

Matrícula

203305786

Trimestre 010-O

UEA (clave): 115127

**Proyecto Terminal de Ingeniería en
Computación II**

Índice de contenido

Introducción.....	3
Justificación.....	3
Antecedentes.....	4
Objetivo.....	5
Desarrollo.....	6
Resultados y discusión.....	8
Conclusiones.....	10
Áreas de Oportunidad.....	11
Bibliografía.....	12
Apéndices.....	13
A) Escenarios, Funcionamiento y Pruebas de la Aplicación	13
B) Estructura del Proyecto de PHP en Netbeans.....	22
C) Código fuente.....	23
D) Configuración del ambiente para el despliegue de la aplicación.....	24

Introducción

En la actualidad las redes de comunicaciones juegan un papel preponderante en una buena parte de los sistemas productivos de la sociedad, en especial de aquellos en que la eficiencia de los procesos, tales como el almacenamiento, transmisión y acceso seguro a la información, es de gran importancia.

De aquí surge la necesidad de contar con las herramientas que permitan y faciliten la realización de dichas tareas administrativas y que se encuentren al alcance de todos aquellos que así lo requieran.

La decisión de realizar este proyecto fue tomada en base a esa necesidad subyacente de contar con las herramientas que aporten grandes ventajas en la gestión de red y que además cuenten con características amigables para los usuarios (administradores de red), permitiéndoles agilizar sus labores y tomar decisiones en base a una interpretación de la información a la que ahora tendrán acceso.

El presente reporte se refiere a la parte complementaria del proyecto **“Diseño e Implementación de una Aplicación para la Administración del una Red en el SO Linux”** cuya primera parte ha sido ya realizada y reportada por **Héctor Ramiro Pérez Sánchez**. En un principio, se hizo la propuesta del proyecto en conjunto y de este modo, el complemento y parte final del mismo se describe a lo largo de este documento.

Justificación

Aunque hoy en día ya se cuenta con herramientas que realizan tareas de tipo administrativo en todos o casi todos los aspectos, dichas herramientas son de un alto costo económico y por lo cual no siempre se tienen los recursos para poder tener acceso a ellas.

Por consiguiente, una de las ventajas de este proyecto es precisamente proporcionar herramientas que tengan funcionalidades no sólo semejantes sino también equiparables a las aplicaciones propietarias existentes en el mercado.

Con el planteamiento y la implementación de estas aplicaciones se puede sentar precedentes en el campo de las herramientas administrativas de red, que tengan la capacidad de ser ampliadas y mejoradas por los mismos usuarios y que estén disponibles (incluyendo el código fuente) sin costo alguno a todo aquel que desee hacer uso de ellas o hacer un aporte sobre las mismas o a partir de ellas.

Es evidente que con la realización de este proyecto conjunto no se pretende cubrir un espectro demasiado amplio en cuanto a las labores administrativas ya que el número de estas debe ser enorme, pero la simple posibilidad de que se puedan cubrir algunas necesidades básicas nos produce una gran satisfacción y gratificación personal.

Antecedentes

En la actualidad existen herramientas propietarias que asisten en las tareas de administración y monitoreo de la red entre otras tantas y continuamente surgen más aplicaciones del mismo tipo.

La misma literatura de la materia de Redes menciona algunas. Por ejemplo, uno de los más recientes y destacados productos de software para realizar tareas de manejo en la seguridad de una red es el *Cisco Router and Security Device Manager (SDM)*. Dicha aplicación, según se menciona en el sitio de la misma, es una herramienta basada en Web que se encarga de la gestión de dispositivos **Cisco**, en particular *routers* y que además aumenta la productividad en la gestión de las redes, simplifica los cambios a la configuración de los dispositivos y ayuda a encontrar los problemas en redes complejas.

Este software por sí mismo representa un hito en la administración de las redes computacionales, no sólo por la gran cantidad de características que posee sino también por la interfaz gráfica más amigable y de fácil uso.

Como esta, podemos nombrar algunas otras que cumplen alguna función administrativa o auxilian en las mismas:

- NetFlow Analysis**¹
- Cisco Network Assistant**²
- HP Open View**³
- Power SNMP Free Manager for .Net**⁴
- SNMP-Manager**⁵
- LoriotPro**⁶

Para tener más información sobre las mismas se puede consultar el sitio Web correspondiente a través de los enlaces proporcionados. Cabe destacar que las últimas tienen alguna versión “lite”, es decir, con menores características pero sin costo o con uno muy bajo en comparación con las primeras enlistadas, además de que dichas implementaciones usan como base al protocolo **SNMP**.

Desde luego existen muchas más aplicaciones con propósitos similares en el mercado y que se encuentran mencionados en la literatura, sobre todo la correspondiente a la tecnología y los dispositivos de **Cisco**, que representa a una de las principales compañías que proveen servicios y herramientas de red en todos los aspectos a nivel mundial.

¹ <http://www.netflowanalysis.com/>

² <http://www.cisco.com/en/US/products/ps5931/index.html>

³ https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-10^36657_4000_100

⁴ http://www.dart.com/psnet_free.aspx

⁵ <http://www.snmp-manager.com/>

⁶ http://www.loriotpro.com/Products/Features/Summary_Free_Edition_EN.php

Objetivo

Diseñar, desarrollar e implementar una aplicación dedicada para el monitoreo del **tráfico** de una red mediante el apoyo del protocolo **SNMP** en sistema operativo Linux.

Además dicha aplicación servirá para complementar el proyecto iniciado por **Héctor Ramiro Pérez Sánchez** cuya concepción, planteamiento y propuesta original comparte la misma finalidad y fue realizada en conjunto.

Dicha aplicación poseerá características como **una interfase gráfica de usuario agradable, además de que su uso y operación serán de fácil entendimiento para el usuario.**

Con esto se pretende que parte de la información recuperada referente al tráfico sea presentada en forma gráfica y en tiempo real y sirva de apoyo a las labores de gestión de una red para el administrador.

Desarrollo

El desarrollo del proyecto se planteó de manera iterativa, sin apegarse a ninguna metodología de manera estricta, es decir, se definieron objetivos específicos o submódulos de la aplicación, los cuales a una vez llevados a cabo se fueron integrando en la aplicación para luego ser probados. Con esto, la aplicación fue siendo llevada de manera incremental hasta alcanzar su forma final.

Los planteamientos de los submódulos se formularon en el formato de **escenarios**, dicha modalidad es comunmente usada en las metodologías llamadas ágiles para el desarrollo de software.

Los escenarios y la demostración del funcionamiento de los módulos se describen e ilustran con mayor detalle en el Apéndice “**Escenarios y Funcionamiento de la Aplicación**”.

El proceso de desarrollo en un sentido más amplio en cada etapa o iteración, se constituye básicamente de las siguiente actividades:

1. Análisis y definición de la funcionalidad deseada de la aplicación.
2. Definición y búsqueda de las **MIBs**⁷ y los **OIDs**⁸ que se utilizarían para obtener información del dispositivo.
3. Implementación de la funcionalidad e integración de la misma en la aplicación
4. Pruebas unitarias y correcciones.
5. Integración y pruebas generales de la aplicación.
6. En caso de ser necesarias, correcciones.

Este conjunto de actividades se realizó de manera constante a lo largo del desarrollo lo que permitió ir agregando funcionalidad a la aplicación y de este modo conseguir poco a poco el objetivo principal del proyecto.

La aplicación construida para el monitoreo del dispositivo está desarrollada sobre la **plataforma Web** y se implementó utilizando el lenguaje **PHP**⁹ y un servidor instalado de manera local **Apache HTTP**¹⁰ en ambiente Linux, en particular se usó la distribución **Open Suse 11.2**¹¹. Además el se utilizó **Netbeans**¹² (**versión 6.9**) como IDE (Integrated Development Environment) para apoyar y agilizar la codificación ya que cuenta con un módulo dedicado para el desarrollo con dicho lenguaje de programación y ofrece varias comodidades adicionales que un editor simple de texto como **gedit** ofrece.

⁷Pueden ser consideradas bases de datos de objetos que almacenan información estadística sobre los dispositivos. “Essential SNMP 2nd Edition”(2005).

⁸Se refiere al identificador de los objetos almacenados en las MIB.

⁹<http://www.php.net/>

¹⁰<http://httpd.apache.org/>

¹¹http://es.opensuse.org/Bienvenidos_a_openSUSE.org

¹²<http://netbeans.org/>

Los detalles de la configuración del ambiente de desarrollo son prácticamente iguales los requeridos para el despliegue de la aplicación y están incluidos en el anexo correspondiente “**Configuración del ambiente para el despliegue de la aplicación**”.

Es necesario mencionar que el lenguaje PHP tiene un **módulo para el protocolo SNMP**¹³, que es fundamental para el desarrollo e implementación del proyecto. Dicho módulo está documentado lo suficientemente para brindar apoyo y cuenta con la información necesaria para llevar a cabo las tareas requeridas.

Puesto que la aplicación operará sobre la plataforma Web, se hizo uso de tecnologías relativas a la misma y que comúnmente se usan en este ambiente, como lo es la biblioteca de funciones para el lenguaje **JavaScript jQuery**¹⁴, naturalmente el lenguaje de etiquetas **HTML**, el estándar **CSS** para hojas de estilo, además de la biblioteca que se empleó para generar los archivos gráficos escrita en PHP llamada **Libchart**¹⁵.

La definición de la tecnología fue realizada con el apoyo y orientación del asesor del proyecto. Para lo cual se buscó la mejor opción disponible de acuerdo al proceso de desarrollo planteado y que facilitara la implementación del mismo, además puesto que se tenían algunos conocimientos previos básicos trabajando con dichas tecnologías, la curva de aprendizaje de los nuevos aspectos relativos a la misma se facilitaron notablemente.

Como apoyo para las pruebas de la aplicación, se utilizó el simulador gráfico de redes **GNS3**¹⁶. Los detalles del apoyo de esta herramienta y la ayuda prestada por la misma se muestran en el anexo de pruebas “**Pruebas del sistema**”.

Una herramienta fundamental en la búsqueda de los OID's utilizados fue sin duda el **SNMP Object Navigator**¹⁷ disponible en el portal de Cisco. Además se hizo uso constante de la suite de aplicaciones **Net-SNMP**¹⁸ que se instaló en el sistema operativo y permitió la comunicación con el dispositivo de pruebas desde la línea de comando de Linux.

¹³<http://mx2.php.net/manual/es/book.snmp.phpg>

¹⁴<http://jquery.com/>

¹⁵<http://naku.dohcrew.com/libchart/pages/introduction/>

¹⁶<http://www.gns3.net/>

¹⁷<http://tools.cisco.com/Support/SNMP/do/BrowseOID.do>

¹⁸<http://www.net-snmp.org/>

Resultados y discusión

Al final del proyecto se tiene una herramienta capaz de comunicarse con un dispositivo de red (*router* o *switch*) por medio de la dirección IP del mismo utilizando el protocolo SNMP y el módulo correspondiente del lenguaje PHP para poder recuperar la información disponible en el dispositivo.

Una vez que se tiene acceso al dispositivo o unidad administrada, se procesa la información recuperada para ser presentada de manera adecuada a través de una interfase de usuario en forma de sitio web.

Además se otorga la posibilidad de activar o desactivar los puertos o interfases del dispositivo mediante una llamada al mismo utilizando la tecnología Ajax. Para lograr modificar dicha información en el dispositivo, se debe ingresar una dirección IP válida y el nombre de una comunidad existente en el dispositivo que permita la **escritura** de información en la unidad administrada.

De forma adicional se puede visualizar una serie de gráficas que presentan información existente en la unidad administrada en tiempo real, dicha información es adquirida en intervalos de tiempo de 5 segundos, después de lo cual las gráficas se refrescan de manera automática y reflejan los cambios correspondientes a la nueva información recuperada.

Dichas gráficas implican también la realización de consultas a objetos específicos en el dispositivo administrado, dicha información es procesada y almacenada de manera temporal en la aplicación del lado del servidor de aplicación. El proceso que reciben los datos obtenidos permite la generación de las gráficas cada vez que se refresca la página correspondiente, por lo cual se recomienda tener esta página abierta en una pestaña u en su caso en una ventana independiente para poder observar la evolución del dispositivo en un periodo de tiempo determinado.

La aplicación ha sido desarrollada utilizando tecnologías capaces de funcionar tanto en un ambiente Unix como Windows, como es el caso de PHP, HTML, JavaScript y CSS. Siempre y cuando el navegador soporte los estándares empleados. Durante el desarrollo, se procuró el apego a dichas normas soportadas por la mayoría de los navegadores, principalmente Mozilla Firefox. Por lo que se espera que funcione el cualquier ambiente y sistema operativo en donde se tenga una adecuada configuración del entorno para el uso de dichas tecnologías y estándares.

Finalmente lo que se buscaba con el desarrollo de esta aplicación de monitoreo del tráfico sobre una red computacional era mostrar información fidedigna y en tiempo real existente en el dispositivo de una manera agradable y fácil de leer, labor que desempeñan las gráficas. Y en general la disposición de la información a lo largo de la aplicación.

Con esta información disponible, se espera que el administrador de la red -que es el personaje principal al que está dirigida esta aplicación-, tome las medidas correspondientes para preservar, o en su caso recuperar el buen funcionamiento de la red.

Conclusiones

Podemos concluir que se ha conseguido el objetivo principal de este proyecto, al llevar a cabo el diseño e implementación de estas aplicaciones complementarias tanto para la gestión de la seguridad en el dispositivo, como para el monitoreo del tráfico de la red a través de consultas al mismo.

Como se menciona en la sección anterior, la administración de la red se verá favorecida en gran medida con el empleo de estas herramientas, y permitirán al usuario, en este caso al administrador de la red, realizar operaciones de configuración y tomar decisiones en base a la información que podrá consultar por medio de las aplicaciones con las que ahora cuenta.

Se debe destacar por un lado la **interfase más agradable** con la que ahora interactúa el administrador, la cual ya no es tan rudimentaria como lo son la consola de comandos de los dispositivos administrados o la línea de comando del sistema operativo. Y el otro aspecto en que se debe hacer énfasis es que **el tratamiento que se le da a la información** permite una lectura e interpretación más rápida y visualmente atractiva, como lo es el caso de las **gráficas**. La presencia de las mismas le representa al usuario un ahorro de tiempo y de pasos en el proceso de monitoreo del tráfico (en este caso).

Con todo esto sólo resta señalar que se ha conseguido elevar el nivel y la calidad de interacción entre el administrador y los dispositivos con la finalidad de auxiliar a la gestión de los recursos existentes dentro de la red. Y esto se puede lograr de distintas maneras gracias a las herramientas y las tecnologías existentes en la actualidad.

Áreas de Oportunidad

Si bien se ha hecho un gran esfuerzo por construir herramientas que facilitan en gran medida la interacción y la gestión de algunos aspectos básicos de un dispositivo de red, apenas podemos decir que se han sentado los precedentes de lo que puede ser un área explotada y desarrollada mucho más allá de este punto.

Podría pensarse que el esfuerzo es pequeño o mínimo, pero si se reflexiona un poco en torno al asunto, se ha demostrado de manera muy sutil y hasta un tanto minimalista que se pueden realizar aplicaciones que desahoguen un poco la dependencia en las grandes aplicaciones de tipo propietarias.

Por medio del ahondamiento en las investigaciones e implementaciones realizadas hasta este punto y con un trabajo continuo que dé seguimiento a las líneas tratadas (seguridad y tráfico) y otras que aún esperan ser tomadas en cuenta pudiera extenderse y escalarse la funcionalidad de lo ya construido.

También existe la posibilidad de abrir nuevas brechas sobre áreas que son de gran interés pero que por alguna razón siguen quedando en el rezago y el desinterés.

Bibliografía

- **“Essential SNMP”**, Douglas Mauro y Kevin Schmidt, O'Reilly, 2nd Edition, Septiembre 2005, ISBN-0: 0-596-00840-6
- **“Cisco Cookbook”**, Ian J. Brown, Kevin Doodley, O'Reilly, Julio 2003, ISBN-0: 0-596-00367-6
- **Información de cisco SDM**
<http://www.cisco.com/en/US/products/sw/secursw/ps5318/>
Consultado el día 07.012011
- **Libchart, a free chart creation PHPLlibrary**
<http://naku.dohcrew.com/libchart/pages/introduction/>
Consultado el 17.12.2010
- **jQuery, a fast and concise JavaScript Library**
<http://docs.jquery.com/Tutorials>
Consultado el 17.12.2010
- **Módulo SNMP de PHP**
<http://mx2.php.net/manual/es/book.snmp.php>
Consultado el 17.10.2010
- **Servidor HTTP Apache**
<http://httpd.apache.org/>
Consultado el 10.10.2010
- **Free Templates**
<http://www.free-templates.me/>
Consultado 17.10.2010
- **SNMP Object Navigator**
<http://tools.cisco.com/Support/SNMP/do/BrowseOID.do>
Consultado 10.10.2010

Apéndices

A) Escenarios, Funcionamiento y Pruebas de la Aplicación

A lo largo del proyecto se realizó un análisis de las posibles situaciones que se pueden presentar para un administrador de la red en un momento determinado. Por medio de la detección de tales situaciones se pudieron plantear los **escenarios** correspondientes que ilustraran de un modo no tan formal como se hace con un *Caso de Uso*, las condiciones que existen alrededor de un determinado tema o problemática y cómo se podría abordar ésta para su posible solución.

A continuación se describen dichos escenarios y cómo quedaron representados en el funcionamiento final de la aplicación. Además se incluyen las pruebas de funcionalidad ilustradas con la consola del dispositivo utilizado para dicho propósito.

Conexión con el dispositivo administrado

En un principio, se debe establecer la conexión con el equipo administrado para poder acceder a la información contenida en las MIBs. Para realizar esta tarea, se requieren dos datos:

- dirección IP del dispositivo
- nombre de una comunidad SNMP existente en el mismo

El primero se refiere a la dirección a través de la cuál se puede establecer comunicación con el equipo dentro de la red. El segundo indica la comunidad SNMP que se utilizará para en conjunto para acceder a los objetos que serán sensados a lo largo de la aplicación.

Debido a que existen dos tipos de comunidades, de “**Sólo lectura**” (Read Only, **RO**) y de “**Lectura/Escritura**” (Read/Write, **RW**), es crucial conocer el tipo de comunidad que se utilizará a lo largo de la aplicación. Si no se va a realizar alguna modificación en el equipo, puede ingresarse el nombre de una comunidad de sólo lectura. En caso de que se pretenda alterar la configuración del dispositivo, es necesaria una comunidad con privilegios de escritura. Se espera que este último sea el caso más común.



Ambos datos son necesarios para establecer la comunicación con el dispositivo. Se muestra a continuación la configuración del dispositivo al que se conectará la aplicación:

```
interface FastEthernet0/0
ip address 10.0.0.1 255.255.255.0
duplex auto
speed auto
!
interface Serial0/0
no ip address
shutdown
!
interface FastEthernet0/1
ip address 10.0.1.2 255.255.255.0
duplex auto
speed auto

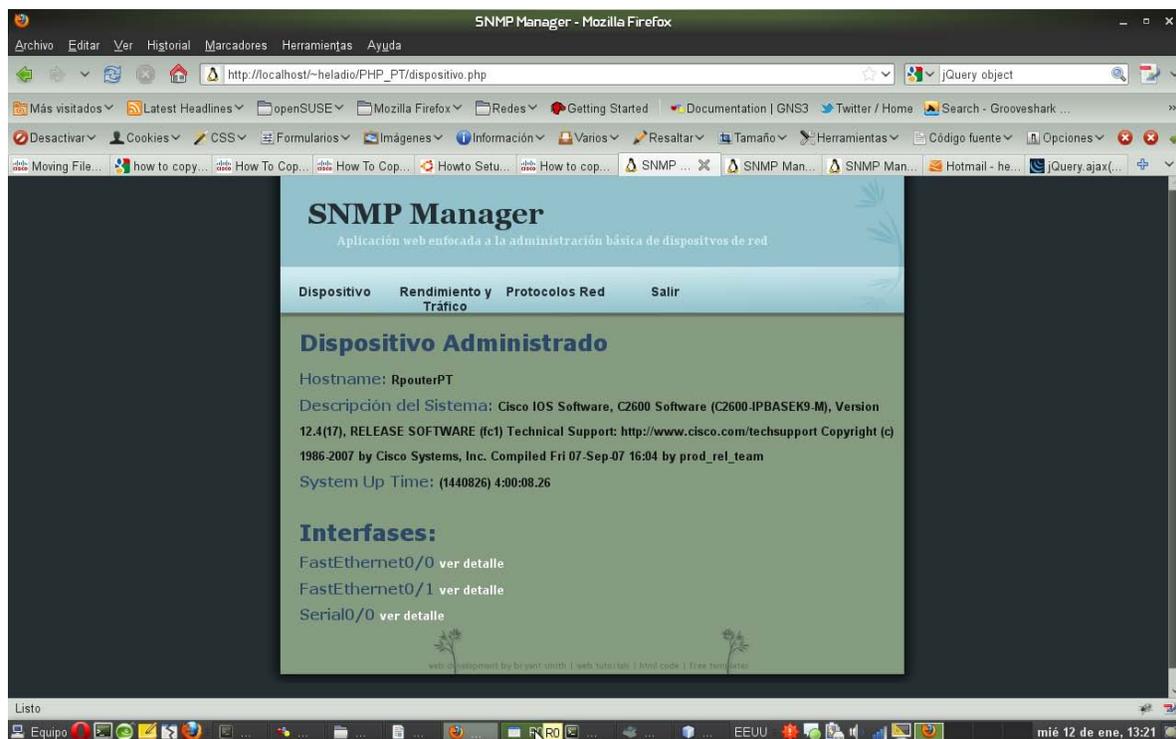
snmp-server community private RW
snmp-server community public RO
```

Se puede observar que se cuenta con las comunidades necesarias y con una configuración de red inicial de las interfaces para comunicarse con el dispositivo por medio de la aplicación.

Recuperación y presentación de la información básica del dispositivo administrado

Una vez que se ha establecido la comunicación con el equipo ingresando la dirección IP y comunidad correspondientes, se desea obtener información relativa al dispositivo. Dicho proceso incluye no sólo recupera la información tal cual, sino que se realiza un proceso de adecuación y “limpieza” de dicha información para que no resulte tan rudimentaria visualmente al usuario.

Por lo tanto se debía idear un método que transformara la información cruda traída desde el dispositivo y aplicarle un formato agradable y legible. Una vez que se procesó la información, debe presentarse en la página inicial de la aplicación.



Se puede apreciar la información ya procesada y con una distribución y formato dentro de la página.

Para corroborarlo, la misma consulta desde la línea de comandos es la siguiente:

```
heladio@suse12:/tftpboot/pt
heladio@suse112:/tftpboot/pt
heladio@suse112:/tftpboot/pt> snmpwalk -v 1 -c private 10.0.0.1 sysDesc
SNMPv2-MIB::sysDescr.0 = STRING: Cisco IOS Software, C2600 Software (C2600-IPBAS
EK9-M), Version 12.4(17), RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Fri 07-Sep-07 16:04 by prod_rel_team
heladio@suse112:/tftpboot/pt>
```

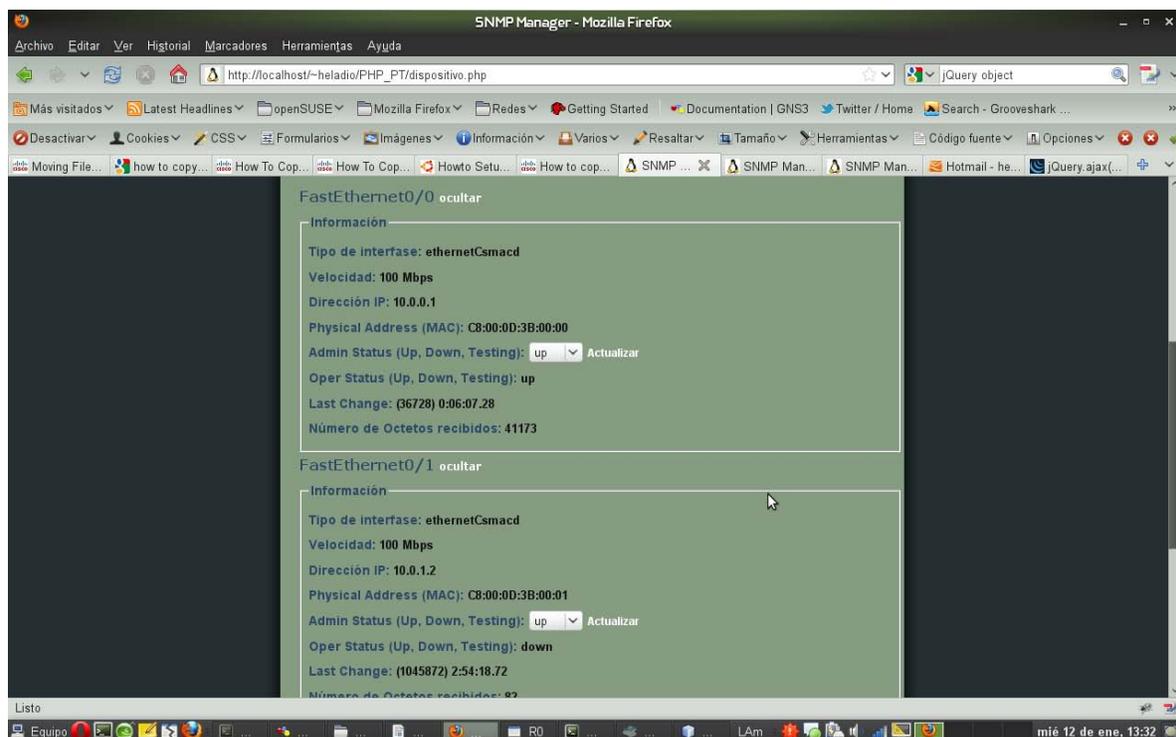
Recuperación y presentación de la información básica de las interfaces del dispositivo administrado

Este escenario es más bien una extensión del anterior ya que, en este caso también se efectúa una recuperación de la información y esta a su vez recibe el tratamiento correspondiente.

Dicha información pertenece a cada una de las interfaces que existen en el dispositivo, tal como son la dirección IP en caso de que la tengan asignada, y la dirección física (MAC) correspondientes.

De este modo, las consultas al equipo debieron especializarse un poco más y esta vez el formato de presentación debe ser un poco más genérico para que la información de cada interfase presente en el dispositivo sea mostrada de manera uniforme y lo más agradable posible.

Para ver los detalles de cada interfase, se debe dar clic en la leyenda “**ver detalle**” que se encuentra junto a la descripción de cada interfase. Con esto se despliega la información obtenida de la consulta.



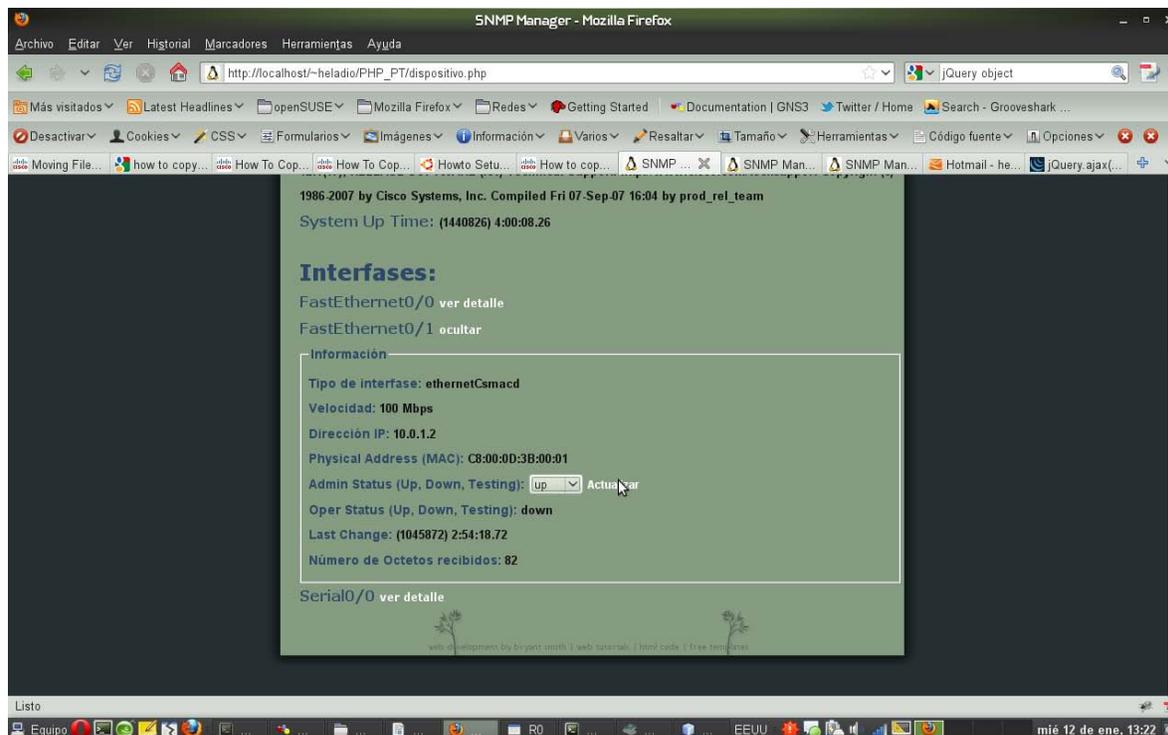
A continuación se ilustra la consulta desde la línea de comandos, que obtiene dicha información sobre el dispositivo y las interfases:

```
heladio@suse112:/tftpboot/pt
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
heladio@suse112:/tftpboot/pt  X  heladio@suse112:/tftpboot/pt
heladio@suse112:/tftpboot/pt> snmpwalk -v 1 -c private 10.0.0.1 ifEntry
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifDescr.1 = STRING: FastEthernet0/0
IF-MIB::ifDescr.2 = STRING: FastEthernet0/1
IF-MIB::ifDescr.3 = STRING: Serial0/0
IF-MIB::ifDescr.4 = STRING: Null0
IF-MIB::ifType.1 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.3 = INTEGER: propPointToPointSerial(22)
IF-MIB::ifType.4 = INTEGER: other(1)
IF-MIB::ifMtu.1 = INTEGER: 1500
IF-MIB::ifMtu.2 = INTEGER: 1500
IF-MIB::ifMtu.3 = INTEGER: 1500
IF-MIB::ifMtu.4 = INTEGER: 1500
IF-MIB::ifSpeed.1 = Gauge32: 100000000
IF-MIB::ifSpeed.2 = Gauge32: 100000000
IF-MIB::ifSpeed.3 = Gauge32: 1544000
IF-MIB::ifSpeed.4 = Gauge32: 4294967295
IF-MIB::ifPhysAddress.1 = STRING: c8:0:2f:de:0:0
IF-MIB::ifPhysAddress.2 = STRING: c8:0:2f:de:0:1
IF-MIB::ifPhysAddress.3 = STRING:
IF-MIB::ifPhysAddress.4 = STRING:
IF-MIB::ifAdminStatus.1 = INTEGER: up(1)
IF-MIB::ifAdminStatus.2 = INTEGER: up(1)
IF-MIB::ifAdminStatus.3 = INTEGER: down(2)
IF-MIB::ifAdminStatus.4 = INTEGER: up(1)
IF-MIB::ifOperStatus.1 = INTEGER: up(1)
IF-MIB::ifOperStatus.2 = INTEGER: up(1)
IF-MIB::ifOperStatus.3 = INTEGER: down(2)
IF-MIB::ifOperStatus.4 = INTEGER: up(1)
IF-MIB::ifLastChange.1 = Timeticks: (8340) 0:01:23.40
IF-MIB::ifLastChange.2 = Timeticks: (3950897) 10:58:28.97
```

Actualización del estado del puerto

Una vez que se ha recuperado la información relativa a cada interfase del dispositivo, se puede modificar el estado del mismo, es decir, se puede activar o desactivar para que deje de estar en funcionamiento y de este modo deje de transmitir y recibir información.

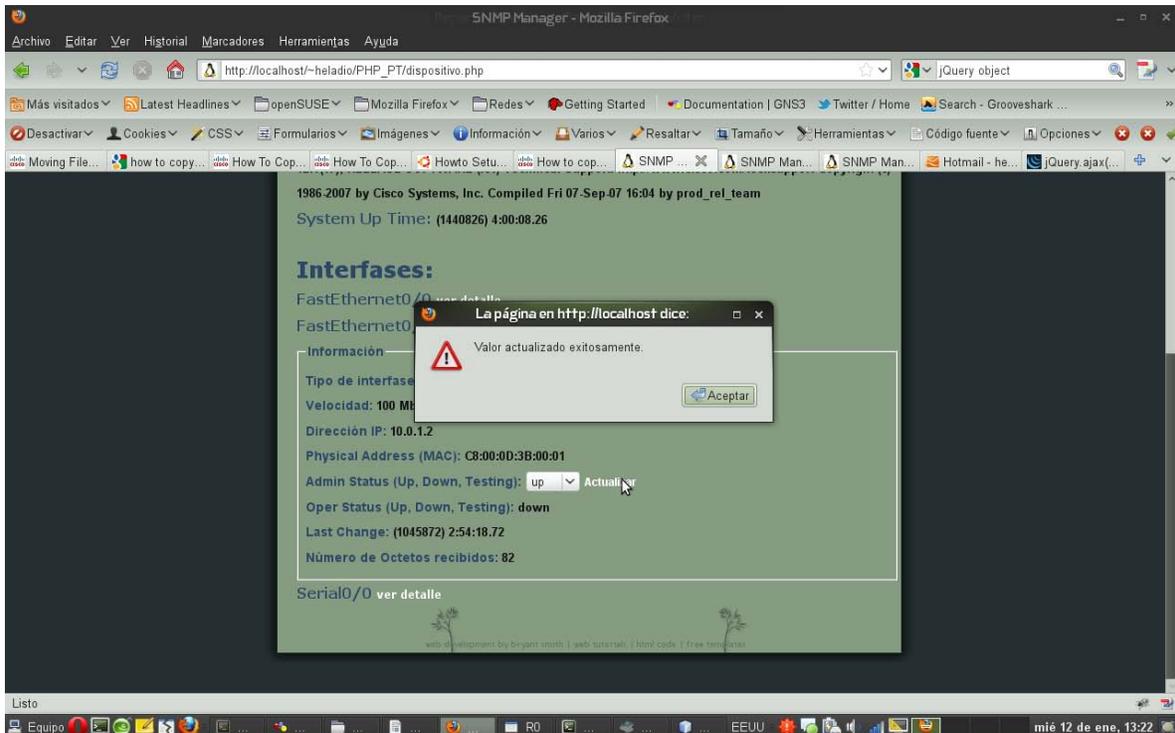
Para realizar esta actualización, se debe ubicar la interfase deseada y luego en los detalles de dicha interfase, seleccionar el nuevo valor. Antes de cualquier modificación (y después de la misma), la aplicación muestra por defecto el valor actual del estado de la interfase.



Como se ilustra a continuación la interfase FastEthernet 0/1 tiene el estatus administrativo en up:

```
RouterHelladyo#show ip interface brief
Interface          IP-Address      OK? Method Status  Protocol
FastEthernet0/0    10.0.0.1        YES manual up      up
Serial0/0          unassigned      YES unset  administratively down down
FastEthernet0/1    10.0.1.2        YES TFTP  up      up
RouterHelladyo#
```

Una vez que se ha seleccionado el nuevo valor, basta con dar clic en “Actualizar” y la aplicación se comunica con el dispositivo para realizar el cambio en la configuración del mismo. Aparecerá una ventana de alerta que informa sobre el resultado de dicha operación.



Una vez realizada la modificación, el estatus del puerto ha cambiado se ilustra cómo en la consola de administración del dispositivo se reflejan los cambios caso de manera instantánea:

```
RouterHelladyo#
*Mar 1 12:45:30.964: %SYS-5-CONFIG_I: Configured from 10.0.0.2 by snmp
*Mar 1 12:45:32.960: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down
*Mar 1 12:45:33.960: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
RouterHelladyo#show ip interface brief
Interface                IP-Address      OK? Method Status      Protocol
FastEthernet0/0          10.0.0.1        YES manual up          up
Serial0/0                unassigned      YES unset  administratively down down
FastEthernet0/1          10.0.1.2        YES TFTP  administratively down down
RouterHelladyo#
RouterHelladyo#
RouterHelladyo#
RouterHelladyo#
```

Ahora el estatus de la interfase actualizada indica que está desactivada, es decir “down”.

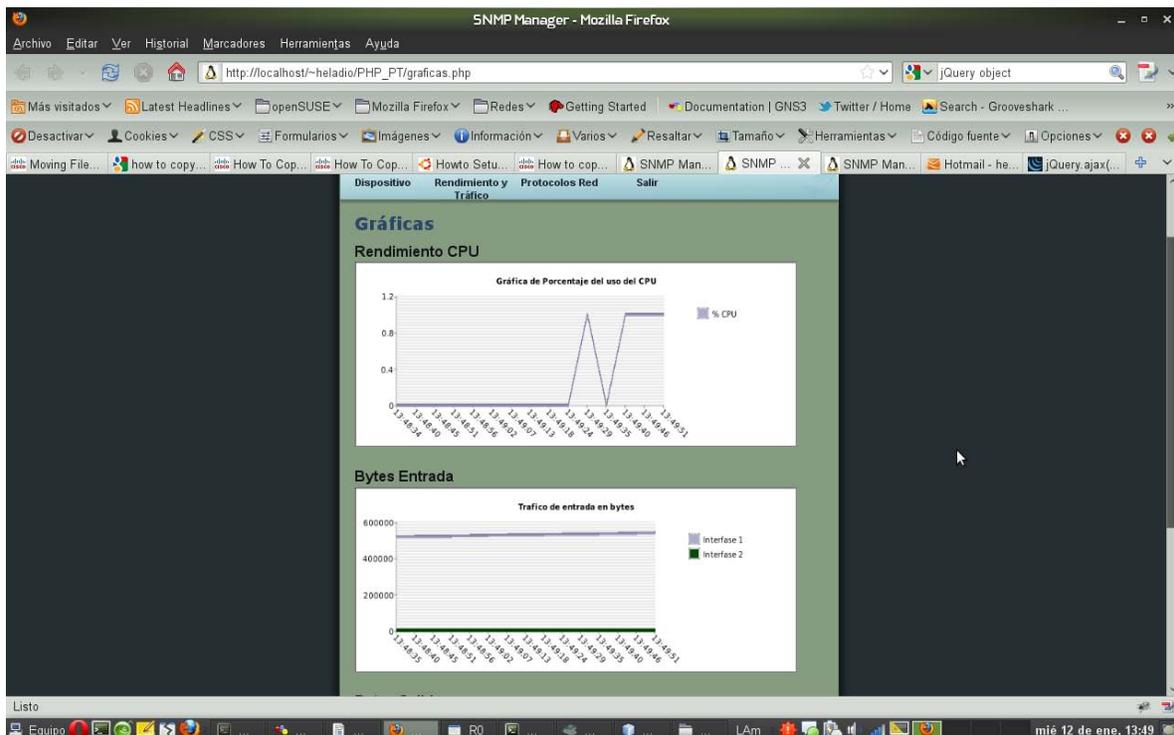
Gráficas de rendimiento y tráfico de los dispositivos

Sin duda una parte crucial de la aplicación, es la presencia de gráficas, las cuales representan una ayuda visual importante y que además son fáciles de leer para la administración del dispositivo de red en cuestión.

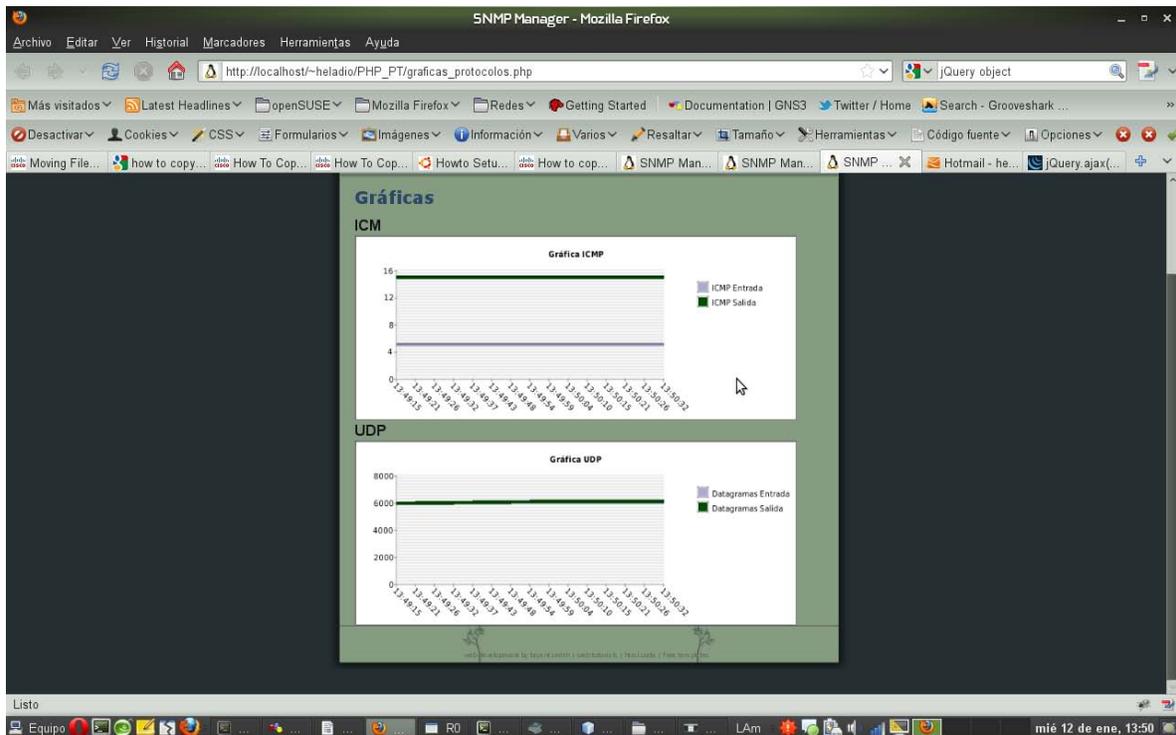
La información para generar dichas gráficas es obtenida directamente del dispositivo en tiempo real, y con esos datos recuperados, se actualizan dichas estadísticas dentro de la aplicación. Cada 5 segundos se hace una consulta y se procesa dicha información para generas las imágenes de las gráficas que serán utilizadas.

Por comodidad, se recomienda abrir las pantallas de las gráficas en una pestaña o ventana independiente. De este modo, podrá visualizarse y monitorear la evolución del dispositivo en ciertos aspectos. Y sin importar el tiempo que dichas gráficas estén activas, sólo se almacena al interior de la aplicación, una porción representativa de información suficiente para mostrar las gráficas y sin saturar demasiado al servidor.

Se debe tomar en cuenta que una vez que se cierran las pestañas o se pierde la conexión con el dispositivo, la información referente al mismo recupera hasta ese momento, se pierde, es decir, sólo se almacena temporalmente y si se regresa al módulo de gráficos, la generación de éstos comenzará de nuevo a partir de este punto.



En la imagen de arriba se muestra el primer conjunto de gráficos generado, relativas al uso del CPU y al tráfico total de entrada y salida de las interfaces del dispositivo que cuentan con una dirección IP asignada.



Finalmente en la imagen de arriba, se puede ver el conjunto de gráficas complementarias que reflejan el tráfico de entrada y salida de los distintos protocolos de red.

Debido a que estas consultas se hacen en tiempo real y un número repetido de ocasiones, no se agregarán las imágenes que ilustren dichas operaciones desde la consola del sistema operativo.

B) Estructura del Proyecto de PHP en Netbeans

La estructura del proyecto se ilustra a continuación, de este modo está organizada la aplicación dentro del servidor en donde se corre la aplicación:



C) Código fuente

El código se anexará en el documento intitulado “Código Fuente.doc”. Aparte se entregará en la carpeta correspondiente al proyecto en los archivos originales y guardando la estructura que se utilizó durante la etapa de implementación de la aplicación.

D) Configuración del ambiente para el despliegue de la aplicación

Para el despliegue de la aplicación se requiere que la aplicación resida en un servidor accesible desde la estación del administrador, para el desarrollo de este proyecto en particular se utilizó al gestor de aplicaciones de la distribución openSuse (YaST) para instalar la mayoría de los paquetes, a continuación se muestra la lista de los paquetes que pueden instalarse a través de cualquier gestor de software que el servidor donde resida la aplicación pueda tener.

- **Instalación Apache HTTP Server.**



Si así se desea, se puede consultar la guía de instalación de la aplicación disponible en <http://httpd.apache.org/docs/2.0/es/install.html>.

- **PHP y Módulo SNMP**

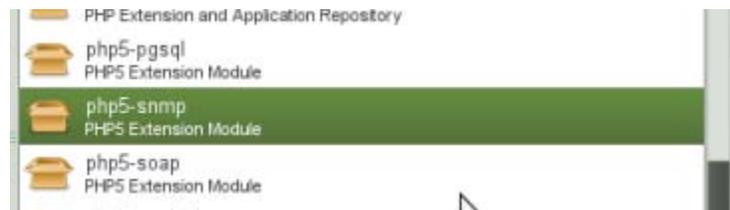
PHP y su módulo para Apache:



Para mayor información acerca de esta configuración de PHP con el servidor apache, se puede consultar la página del lenguaje, que contiene tutoriales de instalación para ambientes Unix:

<http://www.php.net/manual/es/install.unix.apache2.php>

Módulo SNMP para PHP:



Para consultar detalles sobre la configuración de este módulo se puede visitar el sitio del mismo:

<http://www.php.net/manual/es/snmp.setup.php>

- Utilería Net-SNMP



Para mayores detalles de esta aplicación, se sugiere visitar el sitio de la misma:
<http://www.net-snmp.org/docs/INSTALL.html>

- **Bibliotecas jQuery y Libchart**

En el caso de estas bibliotecas, es necesario descargar los archivos relativos a cada una de ellas y posteriormente ubicarlos en algún lugar adecuado dentro de la estructura del proyecto.

Posteriormente, se hará uso de sus servicios por medio de las correspondientes referencias a lo largo de las páginas de la aplicación siempre que así se requiera.

Página de descarga de jQuery http://docs.jquery.com/Downloading_jQuery :



Para jQuery se puede consultar alguno de los tantos tutoriales en línea, por ejemplo uno de ellos ubicado en la página de la biblioteca es el siguiente:

http://docs.jquery.com/Tutorials:How_jQuery_Works

The following is an example of what the complete HTML file might look like if you were to use the script in your own file. Note that it links to Google's [CDN](#) to load the jQuery core file. Also, while the custom script is included in the <head>, it is generally preferable to place it in a separate file and refer that file with the script element's src attribute

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.3/jquery.min.js" type="text/javascript">
  <script type="text/javascript">
    $(document).ready(function(){
      $("a").click(function(event){
        alert("As you can see, the link no longer took you to jquery.com");
        event.preventDefault();
      });
    });
  </script>
</head>
<body>
  <a href="http://jquery.com/">jQuery</a>
</body>
</html>
```

En el caso de Libchart, la página de descarga es:
<http://naku.dohcrew.com/libchart/pages/download/>



The screenshot shows the Libchart website interface. At the top, there is a navigation bar with links for 'Introduction', 'Download' (highlighted in red), 'Sample Code', 'Reference API', and 'Links'. Below the navigation bar is a banner for 'FusionCharts' with the text 'Need more chart types, gauges and maps?'. The main content area is titled 'Download' and contains the following information:

Download
Libchart for PHP5. ([changelog](#))

- Get libchart-1.2.2.tar.gz (100 kB)

Old Versions
Every older version of Libchart can be found on the [Google Code Project](#).

A continuación se ilustra el uso de esta biblioteca para gráficos
<http://naku.dohcrew.com/libchart/pages/samplecode/> :

1. How to Create a vertical bar chart

1.1 Create a bar chart object

We first include the chart library in our project, and then create a chart object of dimensions 500 x 250 pixels. If not specified, the image dimensions will be defaulted to sensible values. There are currently 4 types of charts : *VerticalBarChart*, *HorizontalBarChart*, *LineChart* and *PieChart*.

```
include "libchart/libchart.php";  
$chart = new VerticalBarChart(500, 250);
```

como se puede apreciar, sólo se incluye el archivo correspondiente en donde se requiera su uso dentro de la aplicación.

Código fuente de la Aplicación

El código se presenta por archivo desde el nivel superior de la aplicación (raíz) y continuando por los contenidos de las carpetas.

Nivel: / (raíz)

Archivo: [index.html](#)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
  <head>
    <title>SNMP Manager</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="css/asubtlegreen.css" />
    <script type="text/javascript" src="js/jquery-1.4.3.js" ></script>
    <script type="text/javascript" src="js/scripts.js" ></script>
  </head>
  <body>
    <div id="page">
      <div id="header">
        <div class="title">SNMP Manager </div>
        <div class="subText">Aplicación web enfocada a la administración
&acute;sica de dispositivos de red</div>
        <!--<div class="subText">A web design by Bryant Smith</div-->
      </div>
      <div id="bar" style="display: none;">
        <div class="menuLink"><a href="index.php">Inicio</a></div>
        <div class="menuLink"><a href="graficas.php">Gr&acute;ficas</a></div>
        <div class="menuLink"><a href="index.php">Recursos</a></div>
        <!--<div class="menuLink"><a href="index.html">Services</a></div-->
        <!--<div class="menuLink"><a href="index.html">Contacto</a></div-->
      </div>
      <div id="pageContent">

        <div class="articleTitle">Administración de dispositivos con SNMP</div>

        <div class="articleContent">

          <div class="rightLinks">
            <div class="linkTitle">Enlaces*:</div>
            <p class="links">
              <a href="http://www.bryantsmith.com">Web Design</a><br />
              <a href="http://www.bryantsmith.com">Templates</a><br />
              <a href="http://www.bryantsmith.com">Marketing</a><br />
              <a href="http://www.bryantsmith.com">SEO</a><br />
              <a href="http://www.bryantsmith.com">Programming</a><br />
              <a href="http://www.bryantsmith.com">Consulting</a><br />
            </p>
          </div>
          <!--
            <p>You may use this template on any site, anywhere, for free just please leave the link
back to me in the footer. This template validates XHTML Strict 1.0, CSS Validates as well; enjoy
```

```

:) </p><br />
    <p>Esta es una plantilla gratuita, por lo que se dejará el enlace del pie de
página como único requisito para su uso.</p><br />
-->
    <form method="post" action="dispositivo.php">
        <div>
            <fieldset>
                <legend><b>Información del dispositivo</b></legend>
                <label><b>Dirección IP:</b></label>
                <input type="text" id="ip_dispositivo" name="ip_dispositivo" maxlength="15"/>
                <label><b>Comunidad:</b></label>
                <input type="text" id="comunidad" name="comunidad" maxlength="15"/>
                <input type="button" value="Conectar" id="btn_conectar"/>
            </fieldset>
        </div>
    </form>

    </div>
</div>
</div>
<div id="footer">
    <a href="http://www.aszx.net">web development</a> by <a href="http://www.bryantsmith.com">bryant smith</a> | <a href="http://www.quackit.com">web tutorials</a> | <a href="http://www.htmlcodes.me">html code</a> | <a href="http://www.free-templates.me">free templates</a>
</div>
</body>
</html>

```

Nivel: / (raíz)
Archivo: [index.php](#)

```

<?php
    require_once ('session/session.php');
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title></title>
        <script type="text/javascript" src="js/jquery-1.4.3.js"> </script>
        <script type="text/javascript" src="js/scripts.js"> </script>
    </head>
    <body>
        Hola
        <br/>
        <?php
            print "ip: " . $_SESSION['ip_dispositivo'] . " comunidad: " . $_SESSION['comunidad'];
            if (isset($_SESSION['ip_dispositivo'], $_SESSION['comunidad'])) {
                print "ip: " . $_SESSION['ip_dispositivo'] . " comunidad: " . $_SESSION['comunidad'];
                //Limpia las sesiones:
                unset($_SESSION['ip_dispositivo']);
                unset($_SESSION['comunidad']);
            }
        </?php
    </body>
</html>

```

```
unset($_SESSION['rendimiento']);
unset($_SESSION['bytesEntrada']);
unset($_SESSION['bytesSalida']);
unset($_SESSION['icmpOutEchoReps']);
unset($_SESSION['icmpInEchos']);
unset($_SESSION['udpInDatagrams']);
unset($_SESSION['udpOutDatagrams']);
```

```
print <<<HERE
<body onLoad="location.href = 'index.php'">
  </body>
```

HERE;

```
  } else {
    print <<<HERE
    <body onLoad="location.href = 'index.html'">
      </body>
```

HERE;

```
  }
  // Redirect to index.html :)
  ?>
```

```
</body>
```

```
</html>
```

Nivel: / (raiz)

Archivo: dispositivo.php

```
<?php
```

```
require_once ('session/session.php');
```

```
//Si toma los valores del POST
```

```
if (filter_has_var(INPUT_POST, "ip_dispositivo") && filter_has_var(INPUT_POST, "comunidad")) {
    $ip = filter_input(INPUT_POST, "ip_dispositivo");
    $comunidad = filter_input(INPUT_POST, "comunidad");
```

```
} else if (isset($_SESSION["ip_dispositivo"], $_SESSION["comunidad"])) {
    //print " antes: session->. $_SESSION["ip_dispositivo"]. ",com.->. $_SESSION["comunidad"];
    $ip = $_SESSION['ip_dispositivo'];
    $comunidad = $_SESSION['comunidad'];
    //print " antes: ambiente ip->. $ip.",com.->. $comunidad;
}
```

```
if (isset($ip, $comunidad)) {
    $time_start = microtime(true);
    if (snmpget($ip, $comunidad, "sysDescr.0", 2000000, 2) == NULL) {
        //print "<br/>ip->. $ip.", comun.->. $comunidad;
        header('Location: error.php');
    }
}
```

```
$time_end = microtime(true);
$time = $time_end - $time_start;
```

```
//echo "<br/>Start in $time_start secondes¥n";
```

```
//echo "<br/>End in $time_end secondes¥n";
```

```
//echo "<br/>Delay in $time secondes¥n";
```

```
}
```

```

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
<head>
  <title>SNMP Manager</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <link rel="stylesheet" type="text/css" href="css/asubtlegreen.css" />
  <script type="text/javascript" src="js/jquery-1.4.3.js" ></script>
  <script type="text/javascript" src="js/scripts.js" ></script>
</head>
<body>
  <?php

  //Inicializar la variable de session para limpiar arreglos de graficas
  //if(!isset($_SESSION['reinicia_graficos'])) {
  //  $_SESSION['reinicia_graficos'] = TRUE;
  //}

  ?>

  <div id="page">
    <div id="header">
      <div class="title">SNMP Manager</div>
      <div class="subText">Aplicaci&oacute;n web enfocada a la administraci&oacute;n
b&aacute;sica de dispositivos de red</div>
      <!--<div class="subText">A web design by Bryant Smith</div-->
    </div>
    <div id="bar">
      <div class="menuLink"><a href="dispositivo.php">Dispositivo</a></div>
      <div class="menuLink"><a href="graficas.php">Rendimiento y Tr&aacute;fico</a></div>
      <div class="menuLink"><a href="graficas_protocolos.php">Protocolos Red</a></div>
      <div class="menuLink"><a href="index.php">Salir</a></div>
      <!--<div class="menuLink"><a href="index.html">Services</a></div-->
      <!--<div class="menuLink"><a href="index.html">Contacto</a></div-->
    </div>
    <div id="pageContent">
      <div class="articleTitle">
        Dispositivo Administrado
      <?php
      if (isset($ip, $comunidad)) {
        $sysName = snmpget($ip, $comunidad, "sysName.0"); // . 1. 3. 6. 1. 2. 1. 1. 5. 0
        if (strpos(strtolower($sysName), "router") > 0) {
          print "<img src=%Y"img/router.png%Y" alt=%Y"ro%Y"/>";
        } else if (strpos(strtolower($sysName), "switch")) {
          print "<img src=%Y"img/switch.png%Y" alt=%Y"sw%Y"/>";
        }
      }
      ?>
    </div>

    <div class="articleContent">

```

```

<p>
<?php
include "libs/utileriasSNMP.php";

if (isset($ip, $comunidad)) {
    /* print <<<Here
        IP: $ip <br/>
        Comunidad: $comunidad <br/>
        Here; */

    //print "Sesion IP: ".$_SESSION["ip_dispositivo"]. "\n";
    //$sysName = snmpget($ip, $comunidad, "sysName.0"); //1.3.6.1.2.1.1.5.0
    $sysDescr = snmpget($ip, $comunidad, "sysDescr.0"); //1.3.6.1.2.1.1.1.0
    $sysUpTime = snmpget($ip, $comunidad, "sysUpTime.0"); //1.3.6.1.2.1.1.3.0
    $sysContact = snmpget($ip, $comunidad, "sysContact.0"); //1.3.6.1.2.1.1.4.0
    $sysServices = snmpget($ip, $comunidad, "sysServices.0"); //1.3.6.1.2.1.1.7.0

    $ifNumber = snmpget($ip, $comunidad, "ifNumber.0"); //número de interfaces:
.1.3.6.1.2.1.2.1.0
    //$ifNumber = stripos($ifNumber, ": ") > 0 ? substr($ifNumber, stripos($ifNumber, ":
") + 2) : $ifNumber;

    $sysDescr = UtileriasSNMP::limpiaValor($sysDescr);
    $sysUpTime = UtileriasSNMP::limpiaValor($sysUpTime);
    $sysName = UtileriasSNMP::limpiaValor($sysName);
    $sysContact = UtileriasSNMP::limpiaValor($sysContact);
    $sysServices = UtileriasSNMP::limpiaValor($sysServices);
    $ifNumber = UtileriasSNMP::limpiaValor($ifNumber);

    //<p><span class='span_label'><b>Hostname: </b></span><b>" . $sysName .
"</b><br/>";
    print UtileriasSNMP::procesaDatoValor("Hostname", $sysName);
    print UtileriasSNMP::procesaDatoValor("System Contact", $sysContact);
    print UtileriasSNMP::procesaDatoValor("Descripci&ocute;n del Sistema", $sysDescr);
    print UtileriasSNMP::procesaDatoValor("System Up Time", $sysUpTime );
    //print UtileriasSNMP::procesaDatoValor("Interfases", $ifNumber - 1);

    print "<br/><span class=¥"articleTitle_sub¥">Interfases:</span><br/>"; //ifNumber
    //print "S&ocute;olo hay las que marca el ifNumber, si se trata de acceder una que
no existe, manda un error.<br/>";
    //el ifEntry es uan rama con informacion de cada interface del dispositivo. No se
puede aceder a este OID: 1.3.6.1.2.1.2.2.1
    //print "<b>Informaci&ocute;n de las interfaces:</b></legend><br/>";

    //Reuni&ocaron; de los &iacute;ndices y las correspondientes IP registradas.
    $lista_IPs = UtileriasSNMP::obtenerIPs($ip, $comunidad);

    $codigoHml = ""; //codigo de las interfases

```

```

for ($i = 1; $i < $ifNumber; $i++) {
    //Con las comillas se indica que es un arreglo asociativo
    $direccion_IP = isset($lista_IPs["$i"]) ? $lista_IPs["$i"] : "No asignada.";
    $codigoHml .= UtileriasSNMP::procesaInterfase($ip, $comunidad, $i, $direccion_IP );
}
print $codigoHml;
//if comunidad e IP

//Esta forma manda la información por medio del método POST
print <<<THERE
<form action="graficas.php" method="post" id="graficas">
    <input id="ip" name="ip" type="hidden" value="$ip"/>
    <input id="comunidad" name="comunidad" type="hidden" value="$comunidad"/>
    <input id="reinicia_graficos" name="reinicia_graficos" type="hidden" value="true"/>
</form>
<form action="graficas_protocolos.php" method="post" id="graficas_protocolos">
    <input id="ip" name="ip" type="hidden" value="$ip"/>
    <input id="comunidad" name="comunidad" type="hidden" value="$comunidad"/>
    <input id="reinicia_graficos" name="reinicia_graficos" type="hidden" value="true"/>
</form>
THERE;
    ?>
    </p>
</div>
</div>
</div>
</div>
<?php
    } else {
        print <<<HERE
<body onLoad="location.href = 'index.html'">
    </body>
HERE;
        //header('Location: index.html');
    }
    ?>
    <div id="footer">
        <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a> | <a href="http://www.quackit.com">web
tutorials</a> | <a href="http://www.htmlcodes.me">html code</a> | <a href="http://www.free-
templates.me">free templates</a>
    </div>
</body>
</html>

```

Nivel: / (raíz)

Archivo: ActualizaEstatus.php

```
<?php
session_start();
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
if (!isset($_SESSION["ip_dispositivo"], $_SESSION["comunidad"])) {
    $respuesta['msg'] = "Los valores de IP y comunidad RW no están disponibles.";
} else {
    $ip = $_SESSION['ip_dispositivo'];
    $comunidad = $_SESSION['comunidad'];

    if (isset($_POST['indice'], $_POST['valor'])) {
        $indice = $_POST['indice'];
        $nuevoValor = $_POST['valor'];

        if(snmpset($ip, $comunidad, "ifAdminStatus.$indice", 'i', (int)$nuevoValor)) {
            $respuesta['msg'] = "Valor actualizado exitosamente.";
        } else {
            $respuesta['msg'] = "Error al momento de actualizar el valor.";
        }
        //$respuesta['msg'] = "El indice es $indice y el nuevo valor es $nuevoValor. \n".
        // "La IP es: $ip y la comunidad es: $comunidad.";
    } else {
        $respuesta['msg'] = "El indice de la interfase/puerto o el valor nuevo no llegó. \nNo se
realizó la modificación.";
        // "Pero la IP es: $ip y la comunidad es: $comunidad.";
    }
}

print json_encode($respuesta);

?>
```

Nivel: / (raíz)

Archivo: validaConexion.php

```
<?php
//session_start();
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
$respuesta['msg'] = "Conexión no establecida.";

if (isset($_POST['ip_disp'], $_POST['com'])) {
    $ip = $_POST['ip_disp'];
    $comunidad = $_POST['com'];
    $get_aux = snmpget($ip, $comunidad, "sysDescr.0");

    if (isset($get_aux)) {
```

```

    $respuesta['msg'] = "sysDesc accesado exitosamente.";
} else {
    $respuesta['msg'] = "No se pudo corroborar la conexi3n.";
}
//$respuesta['msg'] = "El indice es $indice y el nuevo valor es $nuevoValor. ¥n".
//"La IP es: $ip y la comunidad es: $comunidad.";
} else {
    $respuesta['msg'] = "No llegaron la ip ni la comunidad.";
    //"Pero la IP es: $ip y la comunidad es: $comunidad.";
}
print json_encode($respuesta);
?>

```

Nivel: / (raíz)

Archivo: error.php

```

<?php
    require_once ('session/session.php');
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Error de conexi3n</title>
        <link rel="stylesheet" type="text/css" href="css/asubtlegreen.css" />
        <script type="text/javascript" src="js/jquery-1.4.3.js"> </script>
        <script type="text/javascript" src="js/scripts.js"> </script>
    </head>
    <body>
        <div id="page">
            <div id="header">
                <div class="title">SNMP Manager</div>
                <div class="subText">Aplicaci3n web enfocada a la administraci3n
b&aacute;sica de dispositivos de red</div>
                <!--<div class="subText">A web design by Bryant Smith</div-->
            </div>
            <div id="pageContent">
                <div class="articleTitle">
                    <b>Error:</b> verifica que la direcci3n IP sea correcta y que la comunidad
                    exista en la configuraci3n del dispositivo.
                </div>
                <br/>
                <div class="articleContent">
                    <?php
                        //print "ip: " . $_SESSION['ip_dispositivo'] . " comunidad: " . $_SESSION['comunidad'];
                        if (isset($_SESSION['ip_dispositivo'], $_SESSION['comunidad'])) {
                            //print "ip: " . $_SESSION['ip_dispositivo'] . " comunidad: " .
$_SESSION['comunidad'];
                            //Limpia las sesiones:
                            unset($_SESSION['ip_dispositivo']);
                            unset($_SESSION['comunidad']);
                            unset($_SESSION['rendimiento']);
                            unset($_SESSION['bytesEntrada']);
                        }
                    }
                </div>
            </div>
        </div>
    </body>
</html>

```

```

unset($_SESSION['bytesSalida']);
unset($_SESSION['icmpOutEchoReps']);
unset($_SESSION['icmpInEchos']);
unset($_SESSION['udpInDatagrams']);
unset($_SESSION['udpOutDatagrams']);

print <<<HERE
<!--body onLoad="location.href = 'index.php'">
  </body-->
  <body onLoad="setTimeout('redirecciona( )', 5000)">
    <b>Regresando a la pantalla de conexi&oacute;n,
      <a href="index.php">redireccionando</a> en 5 segundos.
    </b>
  <h3>

  </h3>

```

HERE;

```

} else {
  print <<<HERE
  <!--body onLoad="location.href = 'index.html'">
    </body-->
    <body onLoad="setTimeout('redirecciona( )', 5000)">
      <b>Regresando a la pantalla de conexi&oacute;n,
        <a href="index.php">redireccionando</a> en 5 segundos.
      </b>
    <h3>

    </h3>

```

HERE;

```

}
// Redirect to index.html :)
?>
</div>
</div>
</div>
<div id="footer">
  <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a> | <a href="http://www.quackit.com">web
tutorials</a> | <a href="http://www.htmlcodes.me">html code</a> | <a href="http://www.free-
templates.me">free templates</a>
  </div>
</body>
</html>

```

Nivel: / (raíz)

Archivo: funcionesGraficas.php

```
<?php
```

```
/**
```

```

* Genera la gráfica correspondiente al uso del CPU.
* @return string Regresa una cadena con el código para incluir la imagen de la
* gráfica generada.
*/

```

```

function graficaRendimiento() {
    $consultas_permitidas = CONSULTAS_PERMITIDAS;
    $chart = new LineChart(); //1. GEnerar la grafica

    $rendimiento = $_SESSION['rendimiento'];

    $serie = new XYDataSet();
    foreach ($rendimiento as $hora => $value) {
        $serie->addPoint(new Point($hora, $value));
    }

    //Ajuste del tamaño del arreglo en session
    if ((count($rendimiento)) >= $consultas_permitidas) {
        array_shift($_SESSION['rendimiento']);
    }

    $dataSet = new XYSeriesDataSet();
    $dataSet->addSerie("% CPU", $serie);

    $chart->setDataSet($dataSet);
    $chart->setTitle("Gráfica de Porcentaje del uso del CPU");

    $chart->getPlot()->setGraphCaptionRatio(0.62);
    $chart->render("img_graficas/grafica_rendimiento_CPU.png");
    //print "<img alt=¥"ICMP¥" src=¥"img_graficas/grafica_rendimiento_CPU.png¥" style=¥"border: 1px
solid gray;¥"/>";
    return "<div>
        <img alt=¥"ICMP¥" src=¥"img_graficas/grafica_rendimiento_CPU.png¥" style=¥"border: 1px
solid gray;¥"/>
        </div>";
}

/**
 * Genera la gráfica correspondiente al trafico total de entrada para cada interfase.
 * @return string Regresa una cadena con el código para incluir la imagen de la
 * gráfica generada.
 */
function graficaEntrada() {
    $chart = new LineChart(600, 250); //1. GEnerar la grafica
    $consultas_permitidas = CONSULTAS_PERMITIDAS;
    $bytesEntrada = $_SESSION['bytesEntrada'];
    //aqui se declara el arreglo que contendra la informacion para formar
    //las series para la grafica
    $bytes_If = array();

    //<br/>>>><br/>Normalizacion del arreglo de los datos para la grafica<br/>";
    //for ($i = 0; $i < count($bytesEntrada); $i++) {
    foreach ($bytesEntrada as $hora => $ifBytes) {
        //Se cambiaron los indices de la consulta por la hora en que se realizó
        foreach ($ifBytes as $indiceIf => $bytes) {
            $bytes_If[$indiceIf][$hora] = $bytes;
            if (count($bytes_If[$indiceIf]) > $consultas_permitidas) {

```

```

        array_shift($bytes_If[$indiceIf]);
    }
}
}
//Ajuste del tamaño del arreglo de los bytes
if ((count($_SESSION['bytesEntrada'])) >= $consultas_permitidas) {
    array_shift($_SESSION['bytesEntrada']);
}

//Obtener el maximo numero de valores para poder rellenar los arreglos.
$consultas_maximo = 0;
foreach ($bytes_If as $indiceIf => $iteracion) {
    if ($consultas_maximo < count($iteracion))
        $consultas_maximo = count($iteracion);
}
//print "El maximo es : $consultas_maximo.<br/>";

$series = array(); //Las series para graficar los puertos

foreach ($bytes_If as $indiceIf => $iteracion) {
    //print "numero de consultas/iteraciones para la interfase $indiceIf: " . count($iteracion) .
'<br/>';
    if (count($iteracion) < $consultas_maximo) {
        $iteracion = array_pad($iteracion, $consultas_maximo, "");
    }
    $temp_serie = new XYDataSet();
    foreach ($iteracion as $key => $value) {
        //print $indiceIf .") $key => $value<br/>";
        $temp_serie->addPoint(new Point($key, $value));
    }
    $series[$indiceIf] = $temp_serie;
}

$dataSet = new XYSeriesDataSet();
foreach ($series as $indiceIf => $serie) {
    $dataSet->addSerie("Interfase $indiceIf", $serie);
}

$chart->setDataSet($dataSet);
$chart->setTitle("Trafico de entrada en bytes");

$chart->getPlot()->setGraphCaptionRatio(0.60);
$chart->render("img_graficas/TraficoEntrante.png");
//print "<img alt=¥"Line chart CPU¥" src=¥"img_graficas/TraficoEntrante.png¥" style=¥"border: 1px
solid gray;¥"/>";
return "<div>
    <img alt=¥"Line chart CPU¥" src=¥"img_graficas/TraficoEntrante.png¥" style=¥"border:
1px solid gray;¥"/>
    </div>";
}

/**

```

```

* Genera la gráfica correspondiente al trafico total de salida para cada interfase.
* @return string Regresa una cadena con el código para incluir la imagen de la
* gráfica generada.
*/
function graficaSalida() {
    $chart = new LineChart(600, 250); //1. Generar la grafica
    $consultas_permitidas = CONSULTAS_PERMITIDAS;
    $bytesSalida = $_SESSION['bytesSalida'];
    //aquí se declara el arreglo que contendrá la información para formar
    //las series para la grafica
    $bytes_If = array();

    //<br/>>>><br/>Normalización del arreglo de los datos para la grafica<br/>";
    //for ($i = 0; $i < count($bytesSalida); $i++) {
    foreach ($bytesSalida as $hora => $ifBytes) {
        //Se cambiaron los índices de la consulta por la hora en que se realizó
        foreach ($ifBytes as $indiceIf => $bytes) {
            $bytes_If[$indiceIf][$hora] = $bytes;
            if (count($bytes_If[$indiceIf]) > $consultas_permitidas) {
                array_shift($bytes_If[$indiceIf]);
            }
        }
    }
    //Ajuste del tamaño del arreglo de los bytes
    if ((count($_SESSION['bytesSalida'])) >= $consultas_permitidas) {
        array_shift($_SESSION['bytesSalida']);
    }

    //Obtener el máximo número de valores para poder rellenar los arreglos.
    $consultas_maximo = 0;
    foreach ($bytes_If as $indiceIf => $iteracion) {
        if ($consultas_maximo < count($iteracion))
            $consultas_maximo = count($iteracion);
    }
    //print "El máximo es : $consultas_maximo.<br/>";

    $series = array(); //Las series para graficar los puertos

    foreach ($bytes_If as $indiceIf => $iteracion) {
        //print "número de consultas/iteraciones para la interfase $indiceIf: " . count($iteracion) .
'<br/>';
        if (count($iteracion) < $consultas_maximo) {
            $iteracion = array_pad($iteracion, $consultas_maximo, "");
        }
        $temp_serie = new XYDataSet();
        foreach ($iteracion as $key => $value) {
            //print $indiceIf ." $key => $value<br/>";
            //$temp_serie->addPoint(new Point($key, $value));
            $temp_serie->addPoint(new Point($key, $value));
            //date("H:i:s");
        }
        $series[$indiceIf] = $temp_serie;
    }
}

```

```

}

$dataSet = new XYSeriesDataSet();
foreach ($series as $indiceIf => $serie) {
    $dataSet->addSerie("Interfase $indiceIf", $serie);
}

$chart->setDataSet($dataSet);
$chart->setTitle("Trafico de salida en bytes");

$chart->getPlot()->setGraphCaptionRatio(0.60);
$chart->render("img_graficas/TraficoSalida.png");
//print "<img alt=¥"Line chart CPU¥" src=¥"img_graficas/TraficoSalida .png¥" style=¥"border: 1px
solid gray;¥"/>";
return "<div>
    <img alt=¥"Line chart CPU¥" src=¥"img_graficas/TraficoSalida.png¥" style=¥"border: 1px
solid gray;¥"/>
    </div>";
}

/**
 * Genera la gráfica correspondiente al trafico total de entrada y salida para relativo al
 protocolo
 * ICMP.
 * @return string Regresa una cadena con el código para incluir la imagen de la
 * gráfica generada.
 */
function icmpEntradaSalida() {
    $consultas_permitidas = CONSULTAS_PERMITIDAS;
    $chart = new LineChart(); //1. GGenerar la grafica

    $icmpEntrada = $_SESSION['icmpInEchos'];
    $icmpSalida = $_SESSION['icmpOutEchos'];

    $icmpInOut = array("Entrada" => $icmpEntrada, "Salida" => $icmpSalida);
    foreach ($icmpEntrada as $hora => $inEchos) {
        //print $hora ." => $inEchos<br/>";
    }

    $series = array();

    foreach ($icmpInOut as $direccion => $serie) {
        $temp_serie = new XYDataSet();
        foreach ($serie as $hora => $value) {
            $temp_serie->addPoint(new Point($hora, $value));
        }
        $series[$direccion] = $temp_serie;
    }

    //Ajuste del tamaño del arreglo en session
    if ((count($icmpEntrada)) >= $consultas_permitidas) {

```

```

    array_shift($_SESSION['icmpInEchos']);
    array_shift($_SESSION['icmpOutEchos']);
}

$dataSet = new XYSeriesDataSet();
foreach ($series as $direccion => $serie) {
    //$k = key($serie);
    $dataSet->addSerie("ICMP $direccion", $serie);
}

$chart->setDataSet($dataSet);
$chart->setTitle("Gráfica ICMP");

$chart->getPlot()->setGraphCaptionRatio(0.62);
$chart->render("img_graficas/grafica_ICMP.png");
//print "<img alt=¥"ICMP¥" src=¥"img_graficas/grafica_ICMP.png¥" style=¥"border: 1px solid
gray;¥"/>";
return "<img alt=¥"ICMP¥" src=¥"img_graficas/grafica_ICMP.png¥" style=¥"border: 1px solid
gray;¥"/>";
}

/**
 * Genera la gráfica correspondiente al trafico total de entrada y salida para relativo al
 protocolo
 * UDP.
 * @return string Regresa una cadena con el código para incluir la imagen de la
 * gráfica generada.
 */
function udpEntradaSalida() {
    $consultas_permitidas = CONSULTAS_PERMITIDAS;
    $chart = new LineChart(); //1. Generar la grafica

    $udpEntrada = $_SESSION['udpInDatagrams'];
    $udpSalida = $_SESSION['udpOutDatagrams'];

    $udpInOut = array("Entrada" => $udpEntrada, "Salida" => $udpSalida);
    foreach ($udpEntrada as $hora => $inEchos) {
        //print $hora ." => $inEchos<br/>";
    }

    $series = array();

    foreach ($udpInOut as $direccion => $serie) {
        $temp_serie = new XYDataSet();
        foreach ($serie as $hora => $value) {
            $temp_serie->addPoint(new Point($hora, $value));
        }
        $series[$direccion] = $temp_serie;
    }

    //Ajuste del tamaño del arreglo en session

```

```

if ((count($udpEntrada)) >= $consultas_permitidas) {
    array_shift($_SESSION['udpInDatagrams']);
    array_shift($_SESSION['udpOutDatagrams']);
}

$dataSet = new XYSeriesDataSet();
foreach ($series as $direccion => $serie) {
    //$k = key($serie);
    $dataSet->addSerie("Datagramas $direccion", $serie);
}

$chart->setDataSet($dataSet);
$chart->setTitle("Gráfica UDP");

$chart->getPlot()->setGraphCaptionRatio(0.62);
$chart->render("img_graficas/grafica_UDP.png");
//print "<img alt='ICMP' src='img_graficas/grafica_UDP.png' style='border: 1px solid
gray;' />";
return "<img alt='ICMP' src='img_graficas/grafica_UDP.png' style='border: 1px solid
gray;' />";
}

```

?>

Nivel: / (raíz)

Archivo: graficas.php

```

<?php
require_once ('session/session.php');
include ('libs/utileriasSNMP.php');
//require_once ('html/top.php');
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="refresh" content="5; url=graficas.php"> </head>
    <title>SNMP Manager</title>
    <script type="text/javascript" src="js/jquery-1.4.3.js"> </script>
    <link rel="stylesheet" type="text/css" href="css/asubtlegreen.css" />
    <script type="text/javascript" src="js/scripts.js"> </script>
</head>

<?php
if (isset($_SESSION['ip_dispositivo'], $_SESSION['comunidad'])) {
    $ip = $_SESSION['ip_dispositivo'];
    $comunidad = $_SESSION['comunidad'];

    if (isset($_POST['reinicia_graficos'])) {
        $reinicio = (boolean)$_POST['reinicia_graficos'];
        //$vienede = "POST";
        $_SESSION['reinicia_graficos'] = FALSE; //Para que ya no se reinicie
    } else if (isset($_SESSION['reinicia_graficos'])) {

```

```

    $reinicio = $_SESSION['reinicia_graficos'];
    //$vienede = "Session.";
}
//print "reinicio->". (boolean)$reinicio. ". viene de: $vienede";
//Grafica del rendimiento de CPU
if (!isset($_SESSION["rendimiento"]) || $reinicio) {
    $_SESSION['rendimiento'] = array();
    $busyPer = snmpget($ip, $comunidad, '.1.3.6.1.4.1.9.2.1.56.0');
    $busyPer = (int)UtileriasSNMP::limpiaValor($busyPer);
    $_SESSION['rendimiento'][date("H:i:s")] = $busyPer;
} else {
    $busyPer = snmpget($_SESSION['ip_dispositivo'], $_SESSION['comunidad'],
'.1.3.6.1.4.1.9.2.1.56.0');
    $busyPer = UtileriasSNMP::limpiaValor($busyPer);
    $_SESSION['rendimiento'][date("H:i:s")] = $busyPer;
}

//Recoleccion de info. para para la grafica de los bytes de entrada y salida.
if (!isset($_SESSION["bytesEntrada"], $_SESSION["bytesSalida"]) || $reinicio) { //ifInOctets
para las interfases que tengan una ip asignada
    $_SESSION["bytesEntrada"] = array();
    $_SESSION["bytesSalida"] = array();

    $ipAdEntIfIndex = snmpwalk($ip, $comunidad, "ipAdEntIfIndex");

    $ifInOctets = snmpwalk($ip, $comunidad, "ifInOctets");
    $ifOutOctets = snmpwalk($ip, $comunidad, "ifOutOctets");

    $sess_ifInOctets = NULL;
    $sess_ifOutOctets = NULL;

    for ($i = 0; $i < count($ipAdEntIfIndex); $i++) {
        $temp_indice = UtileriasSNMP::limpiaValor($ipAdEntIfIndex[$i]);
        $temp_inOct = UtileriasSNMP::limpiaValor($ifInOctets[$temp_indice-1]);
        $temp_outOct = UtileriasSNMP::limpiaValor($ifOutOctets[$temp_indice-1]);
        // print "$temp_indice -> $temp_inOct";
        //Se guardan ahora en el arreglo temporal de sesion
        $sess_ifInOctets[$temp_indice] = $temp_inOct; //primer elemento con los arreglos
        $sess_ifOutOctets[$temp_indice] = $temp_outOct; //primer elemento con los arreglos
    }

    $_SESSION["bytesEntrada"][date("H:i:s")] = $sess_ifInOctets; //Se inicializa el arreglo de
la session
    $_SESSION["bytesSalida"][date("H:i:s")] = $sess_ifOutOctets; //Se inicializa el arreglo de
la session

} else {
    /*Si ya existe el arreglo en la session, se realiza la consulta y se actualiza dicho
arreglo*/
    $ipAdEntIfIndex = snmpwalk($ip, $comunidad, "ipAdEntIfIndex");

    $ifInOctets = snmpwalk($ip, $comunidad, "ifInOctets");

```

```

$ifOutOctets = snmpwalk($ip, $comunidad, "ifOutOctets");

$sess_ifInOctets = NULL;
$sess_ifOutOctets = NULL;

$consultas_permitidas = CONSULTAS_PERMITIDAS;
for ($i = 0; $i < count($ipAdEntIfIndex); $i++) {
    $temp_indice = UtileriasSNMP::limpiaValor($ipAdEntIfIndex[$i]);
    $temp_inOct = UtileriasSNMP::limpiaValor($ifInOctets[$temp_indice-1]);
    $temp_outOct = UtileriasSNMP::limpiaValor($ifOutOctets[$temp_indice-1]);
    //print "$temp_indice -> $temp_inOct";
    //Se guardan ahora en el arreglo de sesion
    $sess_ifInOctets[$temp_indice] = $temp_inOct;
    $sess_ifOutOctets[$temp_indice] = $temp_outOct;
}
$_SESSION["bytesEntrada"][date("H:i:s")] = $sess_ifInOctets;
$_SESSION["bytesSalida"][date("H:i:s")] = $sess_ifOutOctets;
/*
if ((count($_SESSION['bytesEntrada'])) >= $consultas_permitidas) {
    array_shift($_SESSION['bytesEntrada']);
}
*/
*/
}

?>
<body>
    <div id="page">
        <div id="header">
            <div class="title">SNMP Manager</div>
            <div class="subText">Aplicaci&oacute;n web enfocada a la administraci&oacute;n
b&aacute;sica de dispositivos de red</div>
            <!--<div class="subText">A web design by Bryant Smith</div-->
        </div>
        <div id="bar">
            <div class="menuLink"><a href="dispositivo.php">Dispositivo</a></div>
            <div class="menuLink"><a href="graficas.php">Rendimiento y Tr&aacute;fico</a></div>
            <div class="menuLink"><a href="graficas_protocolos.php">Protocolos Red</a></div>
            <div class="menuLink"><a href="index.php">Salir</a></div>
            <!--<div class="menuLink"><a href="index.html">Services</a></div-->
            <!--<div class="menuLink"><a href="index.html">Contacto</a></div-->
        </div>
        <div id="pageContent">
            <div class="articleTitle">
                Gr&aacute;ficas
            </div>

            <div class="articleContent">
                <!--h2>Rendimiento CPU</h2>
                <div>
                    
                </div>
            </div>
        </div>
    </div>

```

```

<br/-->
<div>
  <?php
    include 'libchart/classes/libchart.php';
    require_once 'funcionesGraficas.php';

    print "<h2>Rendimiento CPU</h2>";
    $codigo_grafica_rendimiento = graficaRendimiento();
    print $codigo_grafica_rendimiento;
    print "<br/>";
    print "<h2>Bytes Entrada</h2>";
    $codigo_grafica_entrada = graficaEntrada();
    print $codigo_grafica_entrada;
    print "<br/>";
    print "<h2>Bytes Salida</h2>";
    $codigo_grafica_salida = graficaSalida();
    print $codigo_grafica_salida;
  ?>
</div>

<form action="graficas.php" method="post" id="graficas">
  <input id="ip" name="ip" type="hidden" value="$ip"/>
  <input id="comunidad" name="comunidad" type="hidden" value="$comunidad"/>
  <input id="reinicia_graficos" name="reinicia_graficos" type="hidden" value="true"/>
</form>

<form action="graficas_protocolos.php" method="post" id="graficas_protocolos">
  <input id="ip" name="ip" type="hidden" value="$ip"/>
  <input id="comunidad" name="comunidad" type="hidden" value="$comunidad"/>
  <input id="reinicia_graficos" name="reinicia_graficos" type="hidden" value="true"/>
</form>

</div>
</div>
<?php
  } else {
    print <<<HERE
    <body onLoad="location.href = 'index.html'">
      </body>
  HERE;
      //header('Location: index.html');
    }
  ?>

</div>
<div id="footer">
  <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a> | <a href="http://www.quackit.com">web
tutorials</a> | <a href="http://www.htmlcodes.me">html code</a> | <a href="http://www.free-
templates.me">free templates</a>
  </div>
</body>
</html>

```

Nivel: / (raíz)

Archivo: graficas_protocolos.php

```
<?php
require_once (' session/session.php');
include (' libs/utileriasSNMP.php');
//require_once (' html/top.php');
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta http-equiv="refresh" content="5; url=graficas_protocolos.php"> </head>
    <title>SNMP Manager</title>
    <script type="text/javascript" src="js/jquery-1.4.3.js"> </script>
    <link rel="stylesheet" type="text/css" href="css/asubtlegreen.css" />
    <script type="text/javascript" src="js/scripts.js"> </script>
  </head>

  <?php
  if (isset($_SESSION[' ip_dispositivo'], $_SESSION[' comunidad'])) {
    $ip = $_SESSION[' ip_dispositivo'];
    $comunidad = $_SESSION[' comunidad'];

    if (isset($_POST[' reinicia_graficos'])) {
      $reinicio = (boolean)$_POST[' reinicia_graficos'];
      //$vienede = "POST";
      $_SESSION[' reinicia_graficos'] = FALSE; //Para que ya no se reinicie
    } else if (isset($_SESSION[' reinicia_graficos'])) {
      $reinicio = $_SESSION[' reinicia_graficos'];
      //$vienede = "Session.";
    }

    //print "reinicio->". (boolean)$reinicio. ". viene de: $vienede";
    //Grafica del ICMP
    if (!isset($_SESSION["icmpInEchos"], $_SESSION["icmpOutEchos"]) || $reinicio) {
      $_SESSION[' icmpInEchos'] = array();
      $_SESSION[' icmpOutEchos'] = array();

      $icmpInEchos = snmpget($ip, $comunidad, ' icmpInEchos.0');
      $icmpInEchos = (int)UtileriasSNMP::limpiaValor($icmpInEchos);

      $icmpOutEchos = snmpget($ip, $comunidad, ' icmpOutEchos.0');
      $icmpOutEchos = (int)UtileriasSNMP::limpiaValor($icmpOutEchos);

      $_SESSION[' icmpInEchos'][date("H:i:s")] = $icmpInEchos;
      $_SESSION[' icmpOutEchos'][date("H:i:s")] = $icmpOutEchos;
    } else {
      $icmpInEchos = snmpget($ip, $comunidad, ' icmpInEchos.0');
      $icmpInEchos = (int)UtileriasSNMP::limpiaValor($icmpInEchos);

      $icmpOutEchos = snmpget($ip, $comunidad, ' icmpOutEchos.0');
```

```

$icmpOutEchos = (int)UtileriasSNMP::limpiaValor($icmpOutEchos);

$_SESSION['icmpInEchos'][date("H:i:s")] = $icmpInEchos;
$_SESSION['icmpOutEchos'][date("H:i:s")] = $icmpOutEchos;
}
//Grafica del UDP
if (!isset($_SESSION["udpInDatagrams"], $_SESSION["udpOutDatagrams"]) || $reinicio) {
    $_SESSION['udpInDatagrams'] = array();
    $_SESSION['udpOutDatagrams'] = array();

    $udpInDatagrams = snmpget($ip, $comunidad, 'udpInDatagrams.0');
    $udpInDatagrams = (int)UtileriasSNMP::limpiaValor($udpInDatagrams);

    $udpOutDatagrams = snmpget($ip, $comunidad, 'udpOutDatagrams.0');
    $udpOutDatagrams = (int)UtileriasSNMP::limpiaValor($udpOutDatagrams);

    $_SESSION['udpInDatagrams'][date("H:i:s")] = $udpInDatagrams;
    $_SESSION['udpOutDatagrams'][date("H:i:s")] = $udpOutDatagrams;
} else {
    $udpInDatagrams = snmpget($ip, $comunidad, 'udpInDatagrams.0');
    $udpInDatagrams = (int)UtileriasSNMP::limpiaValor($udpInDatagrams);

    $udpOutDatagrams = snmpget($ip, $comunidad, 'udpOutDatagrams.0');
    $udpOutDatagrams = (int)UtileriasSNMP::limpiaValor($udpOutDatagrams);

    $_SESSION['udpInDatagrams'][date("H:i:s")] = $udpInDatagrams;
    $_SESSION['udpOutDatagrams'][date("H:i:s")] = $udpOutDatagrams;
}
?>
<body>
    <div id="page">
        <div id="header">
            <div class="title">SNMP Manager</div>
            <div class="subText">Aplicaci&ocaron web enfocada a la administraci&ocaron;
b&aacute;sica de dispositivos de red</div>
            <!--<div class="subText">A web design by Bryant Smith</div-->
        </div>
        <div id="bar">
            <div class="menuLink"><a href="dispositivo.php">Dispositivo</a></div>
            <div class="menuLink"><a href="graficas.php">Rendimiento y Tr&aacute;fico</a></div>
            <div class="menuLink"><a href="graficas_protocolos.php">Protocolos Red</a></div>
            <div class="menuLink"><a href="index.php">Salir</a></div>
            <!--<div class="menuLink"><a href="index.html">Services</a></div-->
            <!--<div class="menuLink"><a href="index.html">Contacto</a></div-->
        </div>
        <div id="pageContent">
            <div class="articleTitle">
                Gr&aacute;ficas
            </div>

            <div class="articleContent">
                <div>

```

```

<?php
    include 'libchart/classes/libchart.php';
    require_once 'funcionesGraficas.php';

    print "<h2>ICM</h2>";
    $codigo_grafica_icmp = icmpEntradaSalida();
    print $codigo_grafica_icmp;
    print "<br/>";
    print "<h2>UDP</h2>";
    $codigo_grafica_udp = udpEntradaSalida();
    print $codigo_grafica_udp;
    // $codigo_grafica_entrada = graficaSalida();
    // print $codigo_grafica_entrada;
    ?>
</div>
<form action="graficas.php" method="post" id="graficas">
    <input id="ip" name="ip" type="hidden" value="$ip"/>
    <input id="comunidad" name="comunidad" type="hidden" value="$comunidad"/>
    <input id="reinicia_graficos" name="reinicia_graficos" type="hidden" value="true"/>
</form>
<form action="graficas_protocolos.php" method="post" id="graficas_protocolos">
    <input id="ip" name="ip" type="hidden" value="$ip"/>
    <input id="comunidad" name="comunidad" type="hidden" value="$comunidad"/>
    <input id="reinicia_graficos" name="reinicia_graficos" type="hidden" value="true"/>
</form>

</div>
</div>
<?php
    } else {
    print <<<HERE
    <body onLoad="location.href = 'index.html'">
    </body>
HERE;
        //header('Location: index.html');
    }
?>

</div>
<div id="footer">
    <a href="http://www.aszx.net">web development</a> by <a
href="http://www.bryantsmith.com">bryant smith</a> | <a href="http://www.quackit.com">web
tutorials</a> | <a href="http://www.htmlcodes.me">html code</a> | <a href="http://www.free-
templates.me">free templates</a>
    </div>
</body>
</html>
Nivel: css/
Archivo: asubtlegreen.css

/* A Free Design by Bryant Smith (bryantsmith.com) */

```

```
html, body {
  text-align: center;
}
p {text-align: left;}

body {
  margin: 0;
  padding: 0;
  /*background: #333333 url(..img/img01.gif) repeat-x;*/
  text-align: justify;
  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
  font-size: 13px;
  color: #666666;
  background-color:#252F33;
}
*
{
  margin: 0 auto 0 auto;
  text-align:left;}

#header {
  position:relative;
  width: 680px;
  height:99px;
  margin-left:29px;
  margin-right:21px;
  background: url(..img/header.png) no-repeat;
}

#page
{
  margin: 0 auto 0 auto;
  display: table;
  height: 100%;
  position: relative;
  overflow: hidden;
  background: #252F33 url(..img/background.png) repeat-y;
  width: 730px;
}

.title
{
  position:relative;
  left:30px;
  top:22px;
  text-align:left;
  font-family:Georgia, "Times New Roman", Times, serif;
  font-size:32px;
  font-weight:bold;
  color:#252F33;
}
```

```
.subText
{
  position:relative;
  left:62px;
  top:26px;
  text-align:left;
  font-family:Georgia, "Times New Roman", Times, serif;
  font-size:12px;
  font-weight:bold;
  color:#CEEAE;
}
```

```
.articleTitle
{
  text-align:left;
  padding-left:25px;
  padding-top:10px;
  padding-bottom:10px;
  color: #2C4969;
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:24px;
  font-weight:bold;
}
```

```
.articleContent
{
  width:auto;
  position:relative;
  padding-left:25px;
  color:#101214;
  text-align:left;
  font-family:Arial, Helvetica, sans-serif;
  line-height:27px;
}
```

```
.rightLinks
{
  width: 102px;
  font-size:11px;
  font-weight:bold;
  line-height:21px;
  height:auto;
  margin-right:-3px;
  background-image:url(../img/links_branch.png);
  background-repeat:no-repeat;
  text-align:right;
  float:right;
  display:none; /*No se utilizará*/
}
```

```
.rightLinks a:hover
{
```

```
    color:#667765;
}
```

```
.rightLinks .linkTitle
{
    font-size:13px;
    font-weight:bold;
    margin-top:27px;
    margin-bottom:32px;
    margin-right:5px;
}
```

```
#bar
{
    position:relative;
    width: 680px;
    height:57px;
    margin-left:29px;
    margin-right:21px;
    background: url(..img/bar.png) no-repeat;
}
```

```
.menuLink
{
    height:36px;
    width: 120px;
    text-align:center;
    float:left;
    font-family:Geneva, Arial, Helvetica, sans-serif;
    font-size:14px;
    font-weight:bold;
    color:#252F33;
    padding-top:19px;
}
```

```
.menuLink:hover
{
    background: url(..img/bar2.png) repeat-x;
}
```

```
/*Para el menu de las graficas*/
```

```
a
{
    text-decoration:none;
    color:#252F33;
}
```

```
#pageContent
{
    width: 680px;
```

```
    height: 80%;
    /*height:800px;*/
}

button
{
    clear:both;
}

label {
    width: 100px;
    /*border:1px solid #000;*/
    float: left;
    clear: left;
    margin-bottom: 5px;
    margin-right: 15px;
    text-align: right;
}

input
{
    float:left;
    margin-right: 15px;
}

#footer {

    width: 730px;
    height:60px;
    background: url(../img/footer.png) no-repeat;
    text-align:center;
    font-size:10px;
    color:#667765;
    padding-top:34px;
}

#footer a
{
    font-size:10px;
    color:#667765;
}

html, body {
    text-align: center;
}

.span_label{
    color: #2C4969;
    font-family:Verdana, Arial, Helvetica, sans-serif;
    font-size:16px;
    text-align: left;
    font-weight: bold;
}
```

```

}

.span_label_interfase1{
  color: #2C4969;
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:14px;
  text-align: left;
}
.span_label_interfase2{
  color: #2C4969;
  font-family:Arial, Helvetica, sans-serif;
  font-size:14px;
  text-align: left;
}
.span_sub {
  font-size: 12px;
  color: #FEFEFE;
  cursor: pointer;
  font-weight: bold;
}
.articleTitle_sub
{
  text-align:left;
  /*padding-left:25px;*/
  padding-top:10px;
  padding-bottom:10px;
  color: #2C4969;
  font-family:Verdana, Arial, Helvetica, sans-serif;
  font-size:24px;
  font-weight:bold;
}

```

Nivel: js/

Archivo: scripts

```

$(document).ready(function() {
  //alert("ready");

  //validación de la conexión
  $('#btn_conectar').click(function () {
    //event.preventDefault();
    //alert('Submit pressed!');
    var ip = $('#ip_dispositivo').val();
    var comunidad = $('#comunidad').val();
    if (ip == null || ip == "") {
      alert ('Se requiere la dirección IP.');
```

```

    }

    var info_valida = valida_conexion(ip, comunidad);

    //alert ('Validacion: ' + info_valida);

    if (info_valida) {
        $('form').submit();
    }
    else {
        alert ('Información de conexión inválida.');
```

```

    });
```

```

    $('div [id^="div_if"]').addClass('span_label');
    $('div [id^="div_det_if"]').hide();
```

```

/**
```

```

 * Mostrar y ocultar la información de los puertos.
```

```

 */
```

```

$('span[id^="span_detalles"]').click(function() {
```

```

    var visibility = $(this).parent().next().is('visibility');
```

```

    var display = $(this).parent().next().css("display");
```

```

    //alert($(this).parent().next().css("display") + " visibility: " + typeof(visibility) + "v: " +
visibility);
```

```

    if(display == "none") {
```

```

        //alert("sipo");
```

```

        $(this).parent().next().show();
```

```

        $(this).text("ocultar");
```

```

    }
```

```

    else { //display == "block"
```

```

        //alert("nope");
```

```

        $(this).text("ver detalle");
```

```

        $(this).parent().next().hide();
```

```

    }
```

```

});
```

```

/**
```

```

 * Actualizar el estado de administración del puerto.
```

```

 */
```

```

$('span[id^="span_actualiza"]').click(function () {
```

```

    var indiceIf = $(this).next().val(); //el indice de la interfase(puerto)
```

```

    //var nuevoValor = $(this).prev("select[option:selected]").val(); //el valor que se asignará
```

```

    var nuevoValor = $(this).prev().val(); //el valor que se asignará
```

```

    //alert('Actualiza: ' + indiceIf + ' valor nevo: ' + nuevoValor);
```

```

    //Llamada asíncrona
```

```

    $.ajax({
```

```

        type: "POST",
```

```

        url: "actualizaEstatus.php",
```

```

        dataType: "json",
        data: {indice: indiceIf, valor: nuevoValor},
    //    beforeSend
    success: function(datos) {
        alert(datos.msg);
    },
    error: function(datos) {
        alert('Error.' + datos.msg);
    }
    });
});

//$("a[href^='graficas']").text("Graphs");
$("a[href^='graficas']").click(function(event) {
    var href = $(this).attr('href');
    var forma = $('form[action*="' + href + '"]');
    event.preventDefault();
    //alert ($(this).text());
    //alert (forma.attr('id'));
    forma.submit();
    //$('form').submit();
});

});

function redirecciona( )
{
    location.href = 'index.php';
}

function valida_conexion(ip, comunidad)
{
    //alert('ip> ' + ip + ' comunidad> ' + comunidad);
    var res;
    //Llamada asíncrona para validar la conexión
    $.ajax({
        type: "POST",
        url: "validaConexion.php",
        dataType: "json",
        data: {ip_disp: ip, com: comunidad},
        async: false,
    //    beforeSend
    success: function(datos) {
        //alert(datos.msg);
        res = true;
    },
    error: function(datos) {
        //alert('Error en la verificación de conexión. ' + datos.msg);
        res = false;
    }
    });
});

```

```

    //alert(' esValida: ' + res);
    return res;

}
Nivel: libs/
Archivo: utileriasSNMP.php
<?php

/**
 * Clase con funciones que se pueden usar a lo largo del sistema.
 *
 * @author heladio
 */
class UtileriasSNMP {
    //put your code here

/**
 * Procesa toda la informacion de cada interfase que será mostrada.
 * @param string $ip La dirección IP del dispositivo.
 * @param string $comunidad La comunidad de lectura.
 * @param int $indice El indice de la interfase que se está procesando.
 * @return string El código HTML que se presentará en la página.
 */
    public static function procesaInterfase($ip, $comunidad, $indice, $direccion_IP) {

        $temp = "";
        $ifTable_ifEntry_ifIndex = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.1." . $indice);
        $ifTable_ifEntry_ifDescr = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.2." . $indice);
        $ifTable_ifEntry_ifType = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.3." . $indice);
        $ifTable_ifEntry_ifSpeed = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.5." . $indice);
        $ifTable_ifEntry_ifPhysAddress = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.6." . $indice);
        $ifTable_ifEntry_ifAdminStatus = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.7." . $indice);
        $ifTable_ifEntry_ifOperStatus = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.8." . $indice);
        $ifTable_ifEntry_ifLastChange = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.9." . $indice);
        $ifTable_ifEntry_ifInOctets = snmpget($ip, $comunidad, ".1.3.6.1.2.1.2.2.1.10." . $indice);

        //Tratamiento de las variables:
        $ifTable_ifEntry_ifDescr = self::limpiaValor($ifTable_ifEntry_ifDescr);
        $ifTable_ifEntry_ifType = self::limpiaValor($ifTable_ifEntry_ifType, true);
        $ifTable_ifEntry_ifSpeed =
self::convierteVelocidad(self::limpiaValor($ifTable_ifEntry_ifSpeed), "Mbps");
        $ifTable_ifEntry_ifPhysAddress = self::limpiaValor($ifTable_ifEntry_ifPhysAddress);
        $ifTable_ifEntry_ifAdminStatus = self::limpiaValor($ifTable_ifEntry_ifAdminStatus, true);

        if (strcmp($ifTable_ifEntry_ifDescr, "Null0") != 0) {
            $temp .=
            "<div id='div_if$indice$'>". $ifTable_ifEntry_ifDescr."</div>".
            " <span id='span_detalles_$indice$' class='span_sub$'>ver detalle</span></div>".
            "<div id='div_det_if$indice$'></div>".
            "<fieldset><legend></legend></fieldset>".
            "<span class='span_label_interfase2'><b></b></span>".

```

```

    " Informaci&oacute;n</b></span></legend>¶n";

    $temp .= self::procesaDatoValor("Tipo de interfase", $ifTable_ifEntry_ifType ,
"span_label_interfase2");

    $temp .= self::procesaDatoValor("Velocidad", $ifTable_ifEntry_ifSpeed,
"span_label_interfase2");

    $temp .= isset($direccion_IP) ? self::procesaDatoValor("Dirección IP", $direccion_IP,
"span_label_interfase2") : "";

    $temp .= self::procesaDatoValor("Physical Address (MAC)", $ifTable_ifEntry_ifPhysAddress,
"span_label_interfase2");

    $temp .= self::procesaDatoValorAdmin("Admin Status (Up, Down, Testing)",
$ifTable_ifEntry_ifAdminStatus, $indice, "span_label_interfase2");

    $temp .= self::procesaDatoValor("Oper Status (Up, Down, Testing)",
self::limpiaValor($ifTable_ifEntry_ifOperStatus, true), "span_label_interfase2");

    $temp .= self::procesaDatoValor("Last Change",
self::limpiaValor($ifTable_ifEntry_ifLastChange), "span_label_interfase2");

    $temp .= self::procesaDatoValor("Número de Octetos recibidos",
self::limpiaValor($ifTable_ifEntry_ifInOctets), "span_label_interfase2");

    $temp .= "</fieldset></div>";
} //if descr == NULL0
return $temp;
}

/**
 * Se encarga de obtener las direcciones IP de las interfaces del dispositivo.
 * Obtiene primero los índices y luego los asocia en un arreglo.
 * @param string $ip
 * @param string $comunidad
 * @return array El arreglo con las direcciones y los índices correspondientes.
 */
public static function obtenerIPs($ip, $comunidad) {
    $ipAdEntIfIndex = snmpwalk($ip, $comunidad, "ipAdEntIfIndex"); //1.3.6.1.2.1.4.20.1.2
    $ipAdEntAddr = snmpwalk($ip, $comunidad, "ipAdEntAddr"); //1.3.6.1.2.1.4.20.1.1
    $direccionesIP = null;

    for ($i = 0; $i < count($ipAdEntIfIndex); $i++) {
        $temp_indice = UtileriasSNMP::limpiaValor($ipAdEntIfIndex[$i]);
        $temp_ip = UtileriasSNMP::limpiaValor($ipAdEntAddr[$i]);
        //Se meten a un arreglo para regresarlo.
        //Con las comillas se implica que es un arreglo asociativo.
        $direccionesIP["$temp_indice"] = $temp_ip;
    }
    return $direccionesIP;
}

```

```

/**
 * Procesa la cadena recuperada del dispositivo y la convierte de bytes a Megabytes/
 * @param string $valor
 * @return int Regresa el valor en MB
 */
public static function bytesToMb($valor_bytes) {
    $valor_Mb = self::limpiaValor($valor_bytes);
    $valor_Mb = round((($valor_Mb)/1024)/1024); //MB
    return $valor_Mb;
}

/**
 * Convierte la velocidad de la interfase a Mega bits por segundo (Mbps) o a
 * Giga Herz (Ghz) según el tipo de interfase.
 * @param int $velocidad
 * @param string $unidad Unidad a la que se realizará la conversión.
 * @return string Regresa la cadena con el dato listo para mostrarse.
 */
public static function convierteVelocidad($velocidad, $unidad="Mbps") {
    if ($unidad == "Mbps") {
        $vel_Mbps = $velocidad / 1000000;
        $velocidad = $vel_Mbps . " Mbps";
    } else {
        $vel_GHz = $velocidad / 1000000000;
        $velocidad = $vel_GHz . "GHz";
    }

    return $velocidad;
}

/**
 * Muestra en código HTML la etiqueta que se pasa y el valor si es que
 * éste no está vacío.
 * @param string $label La etiqueta del campo.
 * @param Object $valor El valor recuperado.
 * @return string El código HTML para ser mostrado.
 */
public static function procesaDatoValor($label, $valor, $class_css="span_label") {
    $codigo = "";
    if (!empty($valor)) {
        $codigo = "<p><span class='\"$class_css\"'><b>$label: </b></span><b>";
        $codigo.= $valor . "</b></p>";
    }
    return $codigo;
}

/**
 * Muestra en código HTML la etiqueta que se pasa y el valor si es que
 * éste no está vacío. Especial para el Admin Status.
 * @param string $label La etiqueta del campo.
 * @param Object $valor El valor recuperado.

```

```

* @return string El codigo HML para ser mostrado.
*/
public static function procesaDatoValorAdmin($label, $valor, $indice, $class_css ="span_label") {
    $posibles_valores = array(
        'up' => 1,
        'down' => 2/*,
        'testing' => 3*/
    );
    $codigo = "";
    if (!empty($valor)) {
        $codigo = "<p><span class=¥'$class_css¥'><b>$label: </b></span>".
            "<select id=¥'sel_AdminStatus_if$indice¥'>¥n";
        foreach ($posibles_valores as $key => $value) {
            if ($valor == $key) {
                $codigo .= "<option value=¥'$value¥' selected>$key</option>¥n";
            } else {
                $codigo .= "<option value=¥'$value¥'>$key</option>¥n";
            }
        }
        $codigo.= "</select>¥n".
            "<span id=¥'span_actualiza$indice¥' class=¥'span_sub¥'>¥n".
            " Actualizar ". // "<input type=¥'button¥' value=¥'Actualizar¥' ></input>".
            "</span><input type=¥'hidden¥' id=¥'txt_interfase_¥$indice¥' value=¥'$indice¥' /></p>";
    }
    return $codigo;
}

/**
* Limpia la cadena que se obtiene con los valores de los OIDs. Particularmente
* quita el tipo de dato al principio y los valores entre paréntesis, además
* de que ajusta los valores de la dirección MAC.
* @param string $cadena La cadena que contiene el valor recuperado.
* @param boolean $quitaParentesis [opcional]<br/>Indicador de que se requiere el ajuste de
paréntesis.
* @return string La cadena tratada.
*/
public static function limpiaValor($cadena, $quitaParentesis = false) {
    $valor = stripos($cadena, ": ") > 0 ? substr($cadena, stripos($cadena, ": ") + 2) : $cadena;

    if ($quitaParentesis) {
        $valor = (stripos($valor, "(") == strlen($valor)-1) ? substr($valor, 0, stripos($valor,
"(")) : $valor;
        //print stripos($valor, "("). ": $valor : len-". strlen($valor). "<br/>";
    }

    //revisión para la dirección MAC
    if (stripos($valor, ":0:") > 0) {
        $temp = explode(":", $valor);
        for ($index = 0; $index < count($temp); $index++) {
            if( strlen($temp[$index]) == 1 ) {
                $temp[$index] = "0".$temp[$index];
            }
        }
    }
}

```

```

    }
}

$valor = implode($temp, ":");
$valor = strtoupper($valor);
}

return $valor;
}
/**
 * Método para convertir el valor hexadecimal de la IP con el formato SS:SS:SS:SS
 * al formato decimal XXX.YYY.ZZZ.WWW
 * @param string $ip_hexadecimal El valor en el formato SS:SS:SS:SS
 * @return string La dirección IP con el formato decimal XXX.YYY.ZZZ.WWW
 */
public static function convierteIP($ip_hexadecimal) {
    $temp_IP = explode(":", $ip_hexadecimal);
    for ($index = 0; $index < count($temp_IP); $index++) {
        $temp_IP[$index] = hexdec($temp_IP[$index]);
    }
    $temp_IP = implode('.', $temp_IP);
    return $temp_IP;
}
}
?>

```

Nivel: session/

Archivo: session.php

```

<?php
/*
 * Contiene el manejo de la session para ser incluido al comienzo de las paginas.
 *
 */
//ini_set('session.use_cookies', '1');
//ini_set('session.use_trans_sid', '0');
//ini_set('session.gc_maxlifetime', '1800');
//session_id(md5($_SERVER['REMOTE_ADDR']));
session_start();
define('CONSULTAS_PERMITIDAS', 15);
if (!isset($_SESSION["ip_dispositivo"], $_SESSION["comunidad"])) {
    if (isset($_POST['ip_dispositivo'], $_POST['comunidad'])) {
        $_SESSION["ip_dispositivo"] = $_POST['ip_dispositivo'];
        $_SESSION["comunidad"] = $_POST['comunidad'];
    } else {
        header('Location: index.html');
        //$_SESSION["ip_dispositivo"] = $_POST['ip_dispositivo'];
        //$_SESSION["comunidad"] = $_POST['comunidad'];
    }
}
}
?>

```