

# Reporte de Práctica 1

## Administración de Servicios en Red. 4CM1.

---

Huerta Martínez Jesús Manuel, Monteón Valdés Raúl Kevin, Olivares García Marco Antonio.  
Escuela Superior de Cómputo, IPN

30 de septiembre de 2018

<b>1. Introducción</b>	<b>1</b>
1.1. SNMP	1
1.1.1. ¿Qué es la MIB	2
<b>2. Desarrollo</b>	<b>5</b>
2.1. Arquitectura básica de SNMP	5
2.1.1. Instalación de la estación de gestión "Observium" en una máquina virtual	5
2.2. Operaciones SNMP	15
2.2.1. Ejercicio MIB	15
2.3. Modelo de administración de red de SNMP	19
<b>3. Modelo de Casos de Uso</b>	<b>30</b>
3.1. CU1 Asignar Membresía	31
3.1.1. Descripción completa	31
3.1.2. Atributos importantes	31
3.1.3. Trayectorias del Caso de Uso	31
3.2. CU2 Registrar ingreso a área	33
3.2.1. Descripción completa	33
3.2.2. Atributos importantes	33
3.2.3. Trayectorias del Caso de Uso	33
3.3. CU3 Registrar salida de un área	34
3.3.1. Descripción completa	34
3.3.2. Atributos importantes	34
3.3.3. Trayectorias del Caso de Uso	34
3.4. CU4 Registrar pago de membresía	36
3.4.1. Descripción completa	36
3.4.2. Atributos importantes	36
3.4.3. Trayectorias del Caso de Uso	36
3.5. CU5 Consultar uso de área	38
3.5.1. Descripción completa	38
3.5.2. Atributos importantes	38
3.5.3. Trayectorias del Caso de Uso	38

<b>4. Modelo de la Interacción</b>	<b>39</b>
4.0.1. IU23 Pantalla de Control de Acceso . . . . .	39
<b>5. Modelo del Dominio del problema</b>	<b>41</b>

En el mundo actual, en el que la informática gira en torno al concepto de red, el trabajo de los administradores de sistemas es muy complejo. Su misión consiste en mantener en funcionamiento recursos tales como encaminadores (routers), concentradores (hubs), switches, servidores, así como cada dispositivo crítico que conforma la red. Hay gran cantidad de motivos por los cuales un administrador necesita monitorizar entre otros : la utilización del ancho de banda, el estado de funcionamiento de los enlaces, la detección de cuellos de botella, detectar y solventar problemas con el cableado, administrar la información de encaminamiento entre máquinas, etc. La monitorización de la red es también un buen punto desde el que comenzar el estudio de los problemas de seguridad.

En muchos casos, la red de una organización está enlazada mediante costosos enlaces a redes de área extensa (WAN) o con la Internet, y cuyos costes dependen del volumen de tráfico. Es muy importante mantener un registro estadístico del tráfico que circula por estos enlaces. Ésta situación es bastante común en Europa, donde los enlaces X.25 son todavía de uso corriente. La tarificación de este tipo de líneas se realiza en función del número de paquetes enviados y recibidos. La respuesta a lo anteriormente planteado es el protocolo SNMP. [1]

### 1.1. SNMP

El protocolo llamado Simple Network Management Protocol (SNMP), es un protocolo diseñado en los años 80, su principal objetivo fue el integrar la gestión de diferentes tipos de redes mediante un diseño sencillo y que produjera poca sobrecarga en la red. SNMP opera en el nivel de aplicación, utilizando el protocolo de transporte TCP/IP, por lo que ignora los aspectos específicos del hardware sobre el que funciona. La gestión se lleva a cabo al nivel de IP, por lo que se pueden controlar dispositivos que estén conectados en cualquier red accesible desde la Internet, y no únicamente aquellos localizados en la propia red local. Evidentemente, si alguno de los dispositivos de encaminamiento con el dispositivo remoto a controlar no funciona correctamente, no será posible su monitorización ni reconfiguración.

El protocolo SNMP está compuesto por dos elementos: el agente (agent), y el gestor (manager). Es una arquitectura cliente-servidor, en la cual el agente desempeña el papel de servidor y el gestor hace el de cliente. El agente es un programa que ha de ejecutarse en cada nodo de red que se desea gestionar o monitorizar. Ofrece un interfaz de todos los elementos que se pueden configurar. Estos elementos se almacenan en unas estructuras de datos llamadas "Management Information Base" (MIB), se explicarán más adelante. Representa la parte del servidor, en la medida que tiene la información que se desea gestionar y espera comandos por parte del cliente.

El gestor es el software que se ejecuta en la estación encargada de monitorizar la red, y su tarea consiste en consultar los diferentes agentes que se encuentran en los nodos de la red los datos que estos han ido obteniendo.

En esencia, el SNMP es un protocolo muy sencillo puesto que todas las operaciones se realizan bajo el paradigma de carga-y-almacenamiento (load-and-store), lo que permite un juego de comandos reducido. Un gestor puede realizar sólo dos tipos diferentes de operaciones sobre un agente: leer o escribir un valor de una variable en el MIB del agente. Estas dos operaciones se conocen como petición-de-lectura (get-request) y petición-de-escritura (set-request). Hay un comando para responder a una petición-de-lectura llamado respuesta-de-lectura (get-response), que es utilizado únicamente por el agente. La posibilidad de ampliación del protocolo está directamente relacionado con la capacidad del MIB de almacenar nuevos elementos. Si un fabricante quiere añadir un nuevo comando a un dispositivo, como puede ser un encaminador, tan sólo tiene que añadir las variables correspondientes a su base de datos (MIB). Casi todos los fabricantes implementan versiones agente de SNMP en sus dispositivos: encaminadores, hubs, sistemas operativos, etc. Linux no es una excepción, existen varios agentes SNMP disponibles públicamente en la Internet. [1]

La versión original, ahora conocido como SNMPv1, es ampliamente difundida. SNMPv2 añade funcionalidad a la versión original, pero no se ocupa de sus limitaciones de seguridad; esta norma relativamente reciente no ha alcanzado mucha aceptación. La versión SNMPv3 que conserva las mejoras funcionales de SNMPv2 y añade potentes funciones de privacidad y autenticación. El protocolo simple de administración de red (SNMP), publicado en 1988, fue diseñado para proporcionar una implementación sencilla, así como facilitar el trabajo de gestión de redes de múltiples proveedores (enrutadores, servidores, estaciones de trabajo y otros recursos de la red).

La especificación de SNMP tiene como objetivo:

- Definir un protocolo para el intercambio de información entre uno o más sistemas de gestión y un número de agentes,
- Proporcionar un marco para dar formato y almacenamiento de información de gestión y
- Define una serie de variables de información de gestión de propósito general, u objetos.

La versión original de SNMP (ahora conocido como SNMPv1) se convirtió rápidamente en el esquema de gestión de la red más utilizado. Sin embargo, como el uso del protocolo se generalizó, se hicieron evidentes sus deficiencias. Estas incluyen la falta de comunicación-manager-manager, la incapacidad para hacer la transferencia de datos a granel, y la falta de seguridad.

SNMPv2 no ha recibido la aceptación que sus diseñadores anticiparon. Mientras que las mejoras funcionales han sido bienvenidas, los desarrolladores encontraron las modificaciones de seguridad para SNMPv2 demasiado complejas. En consecuencia, el grupo de trabajo SNMPv2 se reactivó para proporcionar una mejora de los documentos SNMPv2. El resultado de este esfuerzo ha sido un éxito menor y un gran fracaso. El éxito de menor importancia es la mejora de los aspectos funcionales de SNMPv2. El gran fracaso radica en el área de la seguridad. El grupo de trabajo fue incapaz de resolver el problema, y surgieron dos enfoques diferentes. Con esta mejora, la parte funcional de SNMPv2 progresó de una propuesta a un estándar de Internet a partir de 1996. Luego, en 1997, empezó a trabajar en SNMPv3, lo que hace cambios funcionales menores e incorpora un nuevo enfoque de seguridad.

### 1.1.1. ¿Qué es la MIB

SNMP define un estándar separado para los datos gestionados por el protocolo. Este estándar define los datos mantenidos por un dispositivo de red, así como las operaciones que están permitidas. Los datos están estructurados en forma de árbol; en el que sólo hay un camino desde la raíz hasta cada variable. Esta estructura en árbol

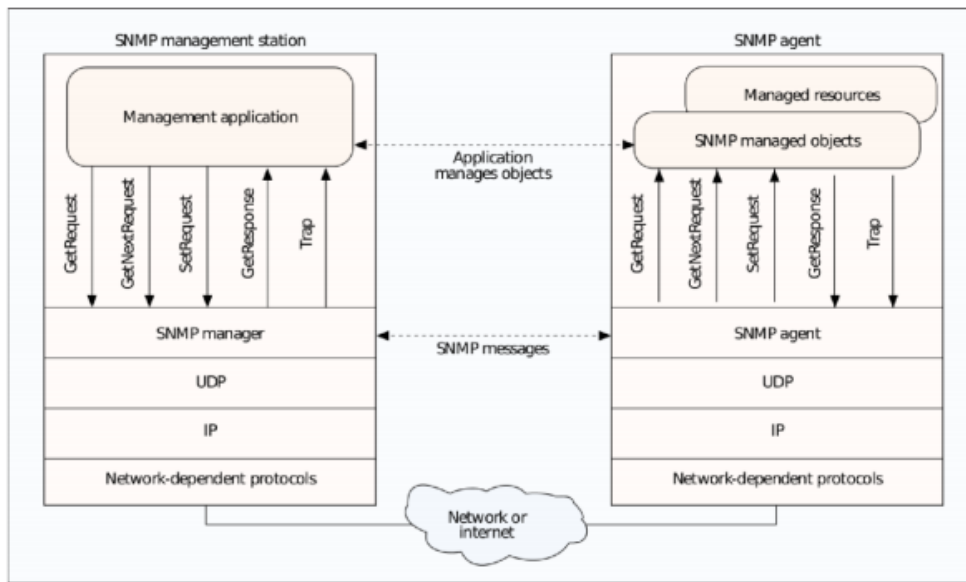


Figura 1.1: Agente y gestor SNMP.

se llama Management Information Base (MIB) y se puede encontrar información sobre ella en varios RFC's. La versión actual de TCP/IP MIB es la 2 (MIB-II) y se encuentra definida en el RFC-1213. En ella se divide la información que un dispositivo debe mantener en ocho categorías (ver Tabla 1). Cualquier variable ha de estar en una de estas categorías.

Objeto.	Descripción.
System	Información del host del sistema de encaminamiento.
Interfaces	Información de los interfaces de red.
Address Translation	Información de traducción de direcciones.
IP	Información sobre el protocolo IP.
ICMP	Información sobre el protocolo ICMP.
TCP	Información sobre el protocolo TCP.
UDP	Información sobre el protocolo UDP.
UGP	Información sobre el protocolo (Exterior Gateway).

Cuadro 1.1: Objetos de la MIB.

La definición de un elemento concreto MIB implica la especificación del tipo de dato que puede contener. Normalmente, los elementos de un MIB son enteros, pero también pueden almacenar cadenas de caracteres o estructuras más complejas como tablas. A los elementos de un MIB se les llama "objetos". Los objetos son los nodos hoja del árbol MIB, si bien, un objeto puede tener más de una instancia, como por ejemplo un objeto tabla. Para referirse al valor contenido en un objeto, se ha de añadir el número de la instancia. Cuando sólo exista una instancia del objeto, está es la instancia cero. Por ejemplo, el objeto ifNumber de la categoría interfaces<sup>es</sup> un entero que representa el número de interfaces presentes en el dispositivo; mientras el objeto ipRoutingTable de la categoría ip contiene la tabla de encaminamiento del dispositivo. Hay que acordarse de utilizar el número de la instancia para leer el valor de un objeto. En este caso, el número de interfaces presentes en un encaminador puede ser observado mediante la instancia ifNumber.0. En el caso de ser un objeto tabla, se ha de utilizar el índice a la tabla como último número para especificar la instancia (fila de la tabla).

Existe otro estándar que define e identifica las variables MIB, llamado "Structure of Management Information" (SMI). SMI especifica las variables MIB, éstas se declaran empleando un lenguaje formal ISO llamado ASN.1, que hace que tanto la forma como los contenidos de estas variables sean no ambiguos. El espacio de nombres ISO (árbol) está situado dentro de un espacio de nombres junto con otros árboles de otros estándares de otras organizaciones. Dentro del espacio de nombres ISO hay una rama específica para la información MIB. Dentro de esta rama MIB, los objetos están a su vez jerarquizados en subárboles para los distintos protocolos y aplicaciones, de forma que esta información puede representarse unívocamente. La siguiente figura muestra el espacio de nombres del MIB del TCP/IP, éste está situado justo bajo el espacio del IAB "mgmt". La jerarquía también especifica el número para cada nivel.

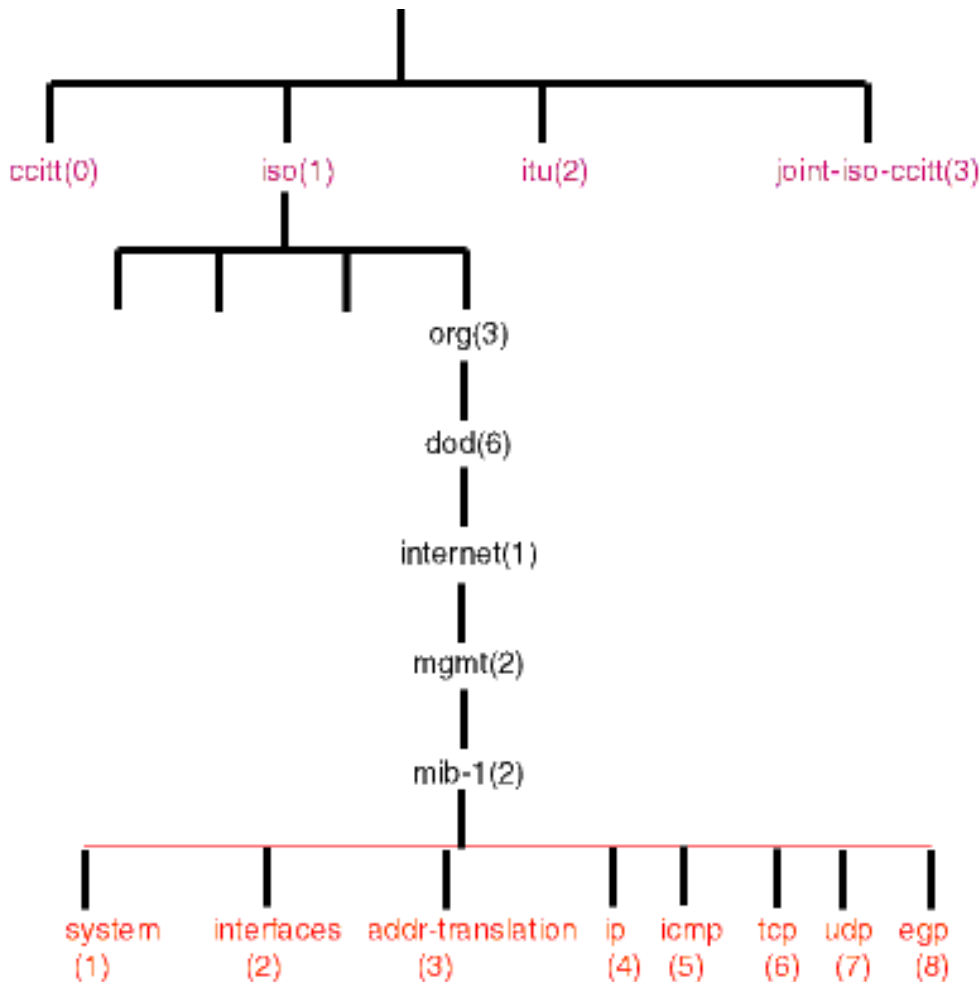


Figura 1.2: Jerarquía del árbol.

Es importante constatar que la mayor parte del software necesita el punto raíz (.) para localizar el objeto en el MIB. Si no se incluye el punto raíz, se asume que el path es relativo desde .iso.org.dod.internet.mgmt.mib-2. De esta forma, el objeto ifNumber de la categoría "interfaces" se puede llamar: **.iso.org.dod.internet.mgmt.mib-2.interfaces.ifnumber** o el equivalente numérico: **.1.3.6.1.2.1.2.1**. Y la instancia es: **.iso.org.dod.internet.mgmt.mib-2.interfaces.ifnumber.0** o el equivalente numérico: **.1.3.6.1.2.1.2.1.0**. Adicionales MIB se pueden añadir a este árbol conforme los vendedores definen nuevos objetos y publican los correspondientes RFC.

## 2.1. Arquitectura básica de SNMP

El modelo de administración de red que se utiliza para SNMP incluye los siguientes elementos:

- Estación de gestión,
- Agente de Gestión,
- Base de información de gestión y
- Protocolo de Gestión de redes.

En las secciones se enlistan las tareas para realizar la instalación y configuración de dichos elementos

### 2.1.1. Instalación de la estación de gestión “Observium” en una máquina virtual

Para realizar la instalación de Observium sobre la máquina virtual debe tenerse previamente instalado un software de virtualización, para el caso utilizamos Virtual Box, así como la imagen ISO del sistema Observium a instalar.

**Paso 1:** Configuramos la red de la máquina virtual de Ubuntu, para eso debemos elegir el adaptador tipo puente, y dentro de las opciones avanzadas elegir el modo promiscuo. Como se ve a continuación:



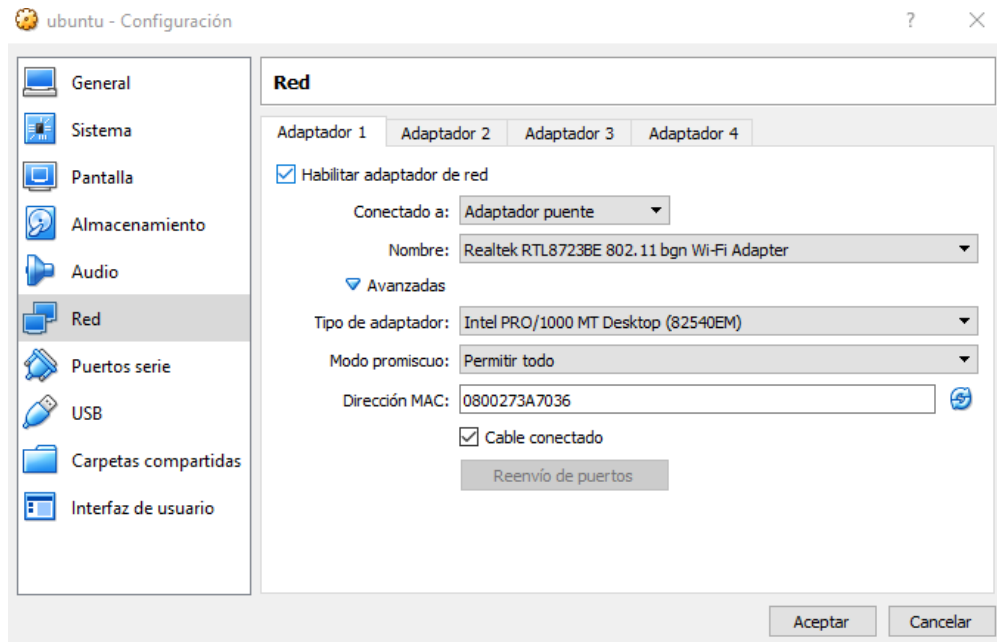


Figura 2.1: Instalación de observium: Paso 1.

**Paso 2:** De las opciones de Virtual Box seleccionamos "Almacenamiento", luego "Controlador IDE y seleccionamos la imagen ISO del sistema Observium. Como se muestra debajo:

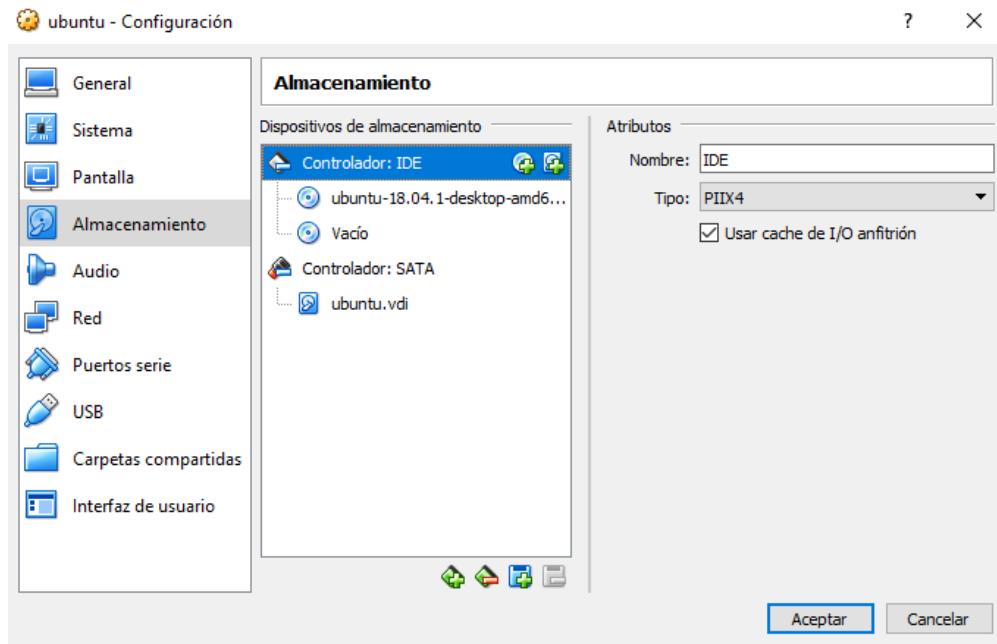


Figura 2.2: Instalación de observium: Paso 2.

**Paso 3:** Creamos una nueva máquina virtual, e introducimos nombre, tipo y versión del sistema operativo a instalar.

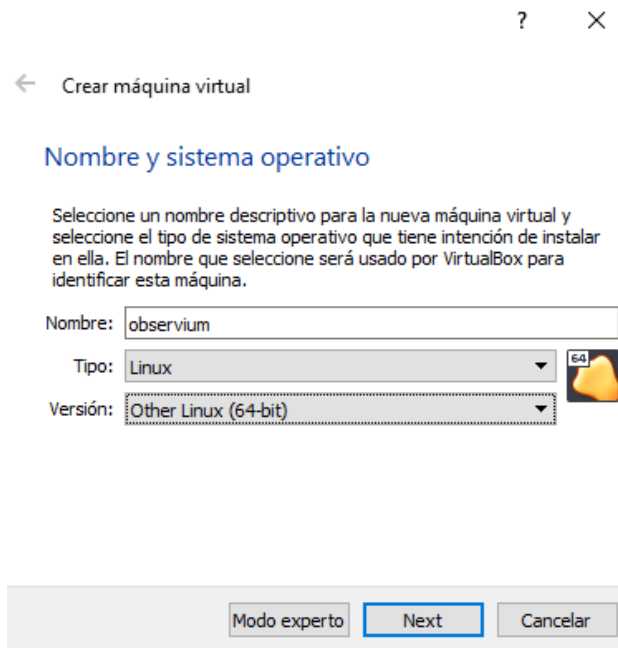


Figura 2.3: Instalación de observium: Paso 3.

**Paso 4:** Seleccionamos las características que tendrá nuestra máquina virtual para el sistema Observium. Tales como la memoria RAM o la capacidad de Disco duro. Como se muestra en las imágenes debajo.

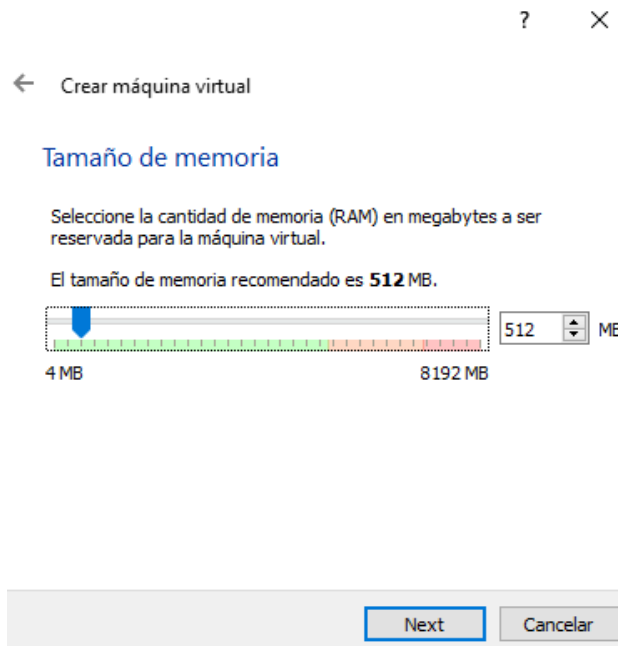


Figura 2.4: Instalación de observium: Paso 4a.

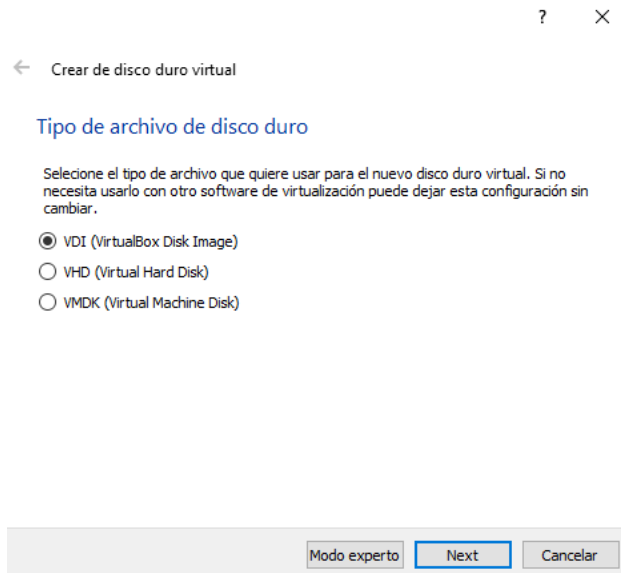


Figura 2.5: Instalación de observium: Paso 4b.

**Paso 5:** Configuramos el adaptador de red, con las configuraciones que se observan en la imagen siguiente.

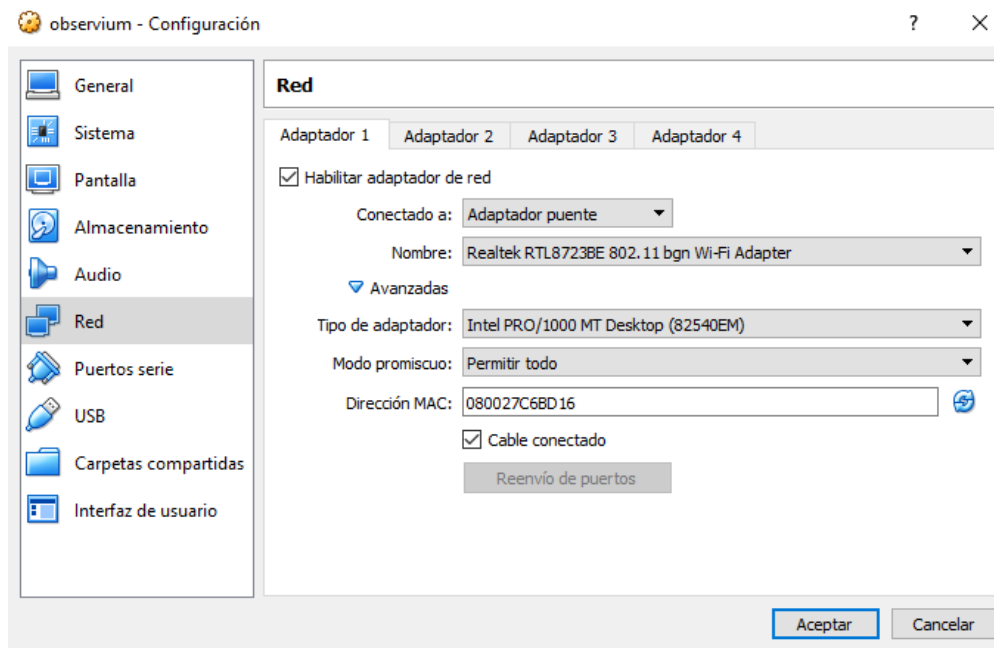


Figura 2.6: Instalación de observium: Paso 5.

**Paso 6:** Se selecciona la imagen ISO con la cual va a correr el sistema.

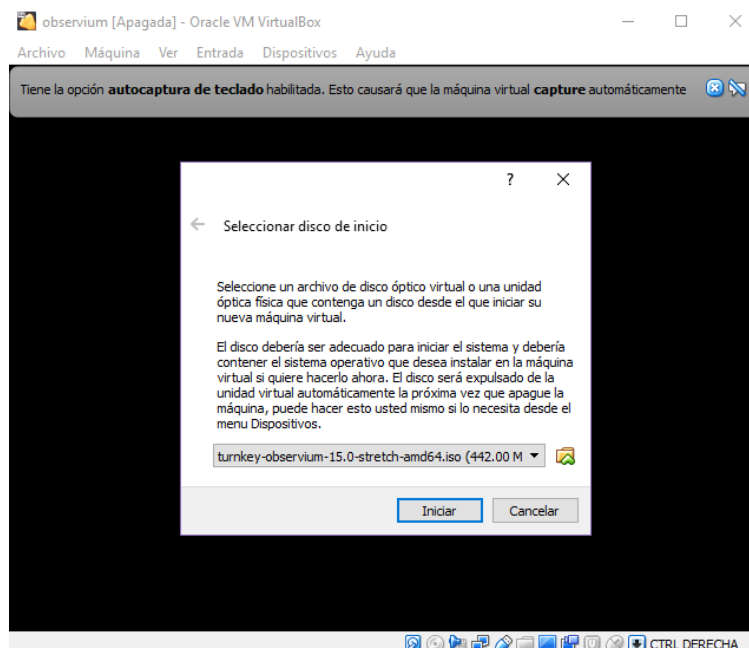


Figura 2.7: Instalación de observium: Paso 6.

**Paso 7:** Seleccionamos la opción de "instalar" mostrada en la siguiente pantalla.

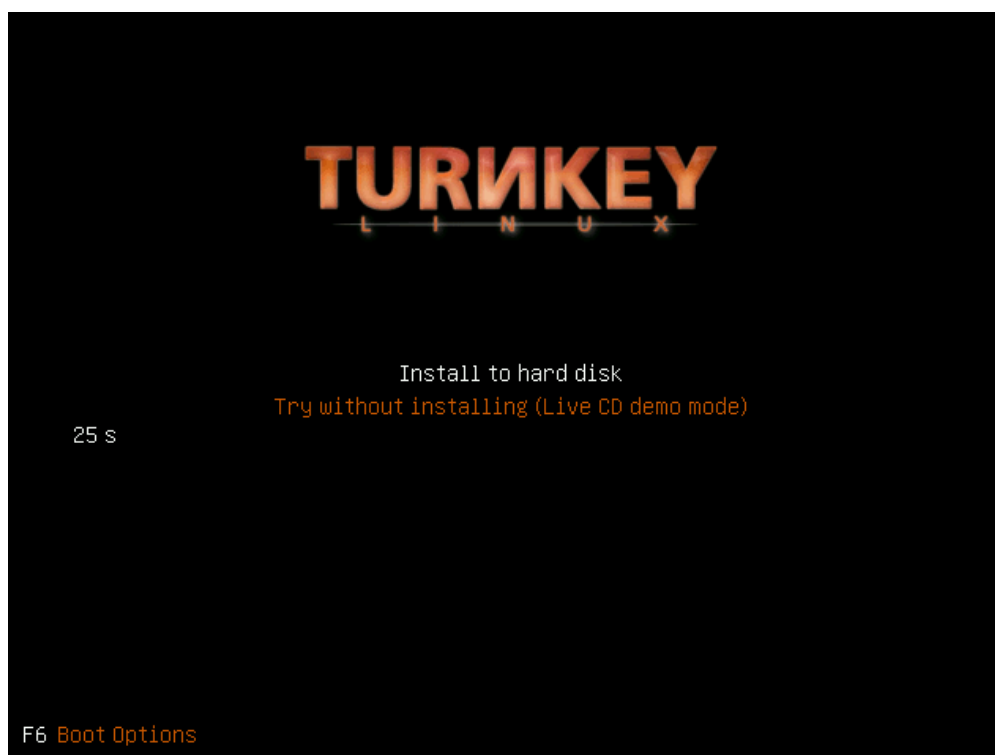


Figura 2.8: Instalación de observium: Paso 7.

**Paso 8:** De las pantallas a continuación siga el flujo, seleccionando las opciones marcadas.

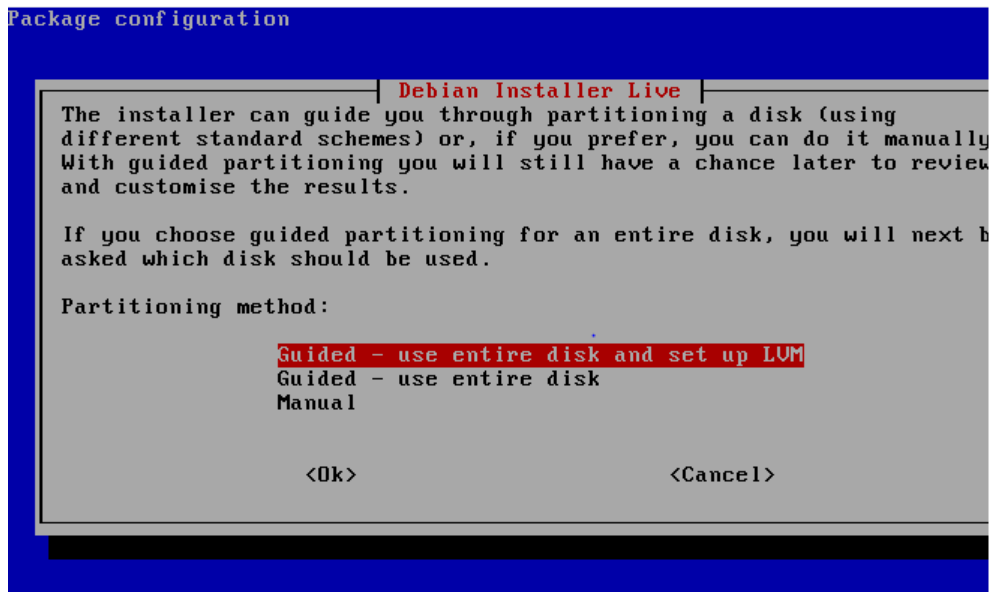


Figura 2.9: Instalación de observium: Paso 8.

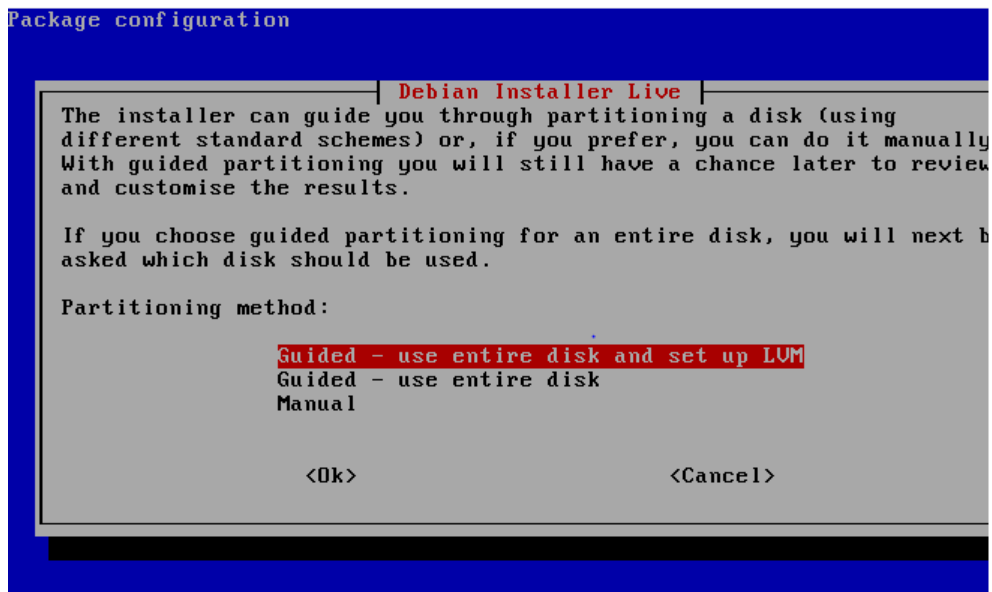


Figura 2.10: Instalación de observium: Paso 8.

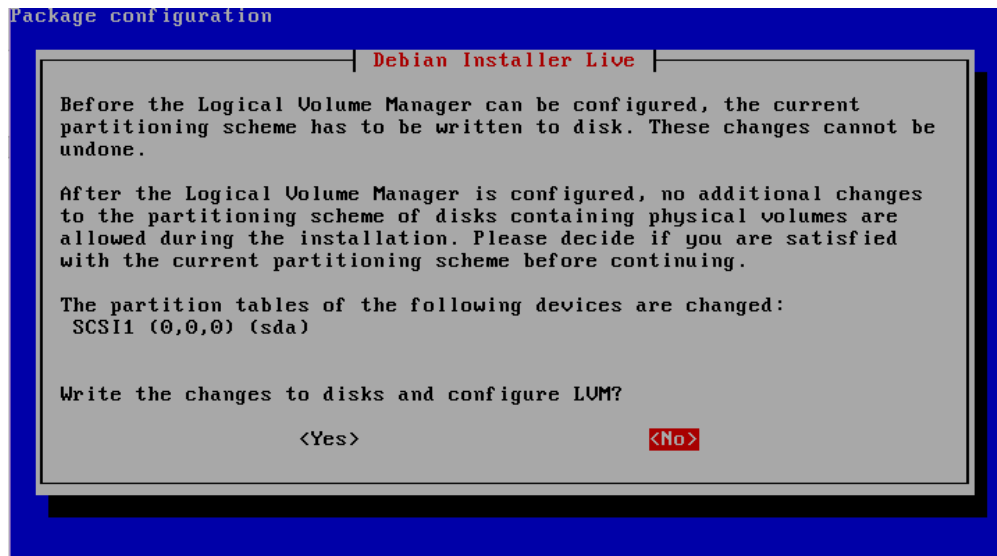


Figura 2.11: Instalación de observium: Paso 8b.

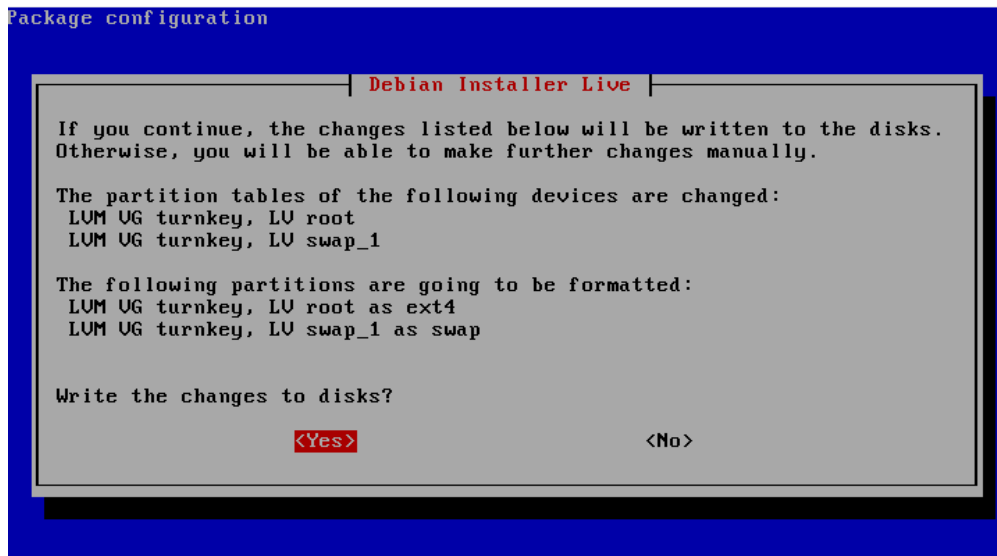


Figura 2.12: Instalación de observium: Paso 8c.

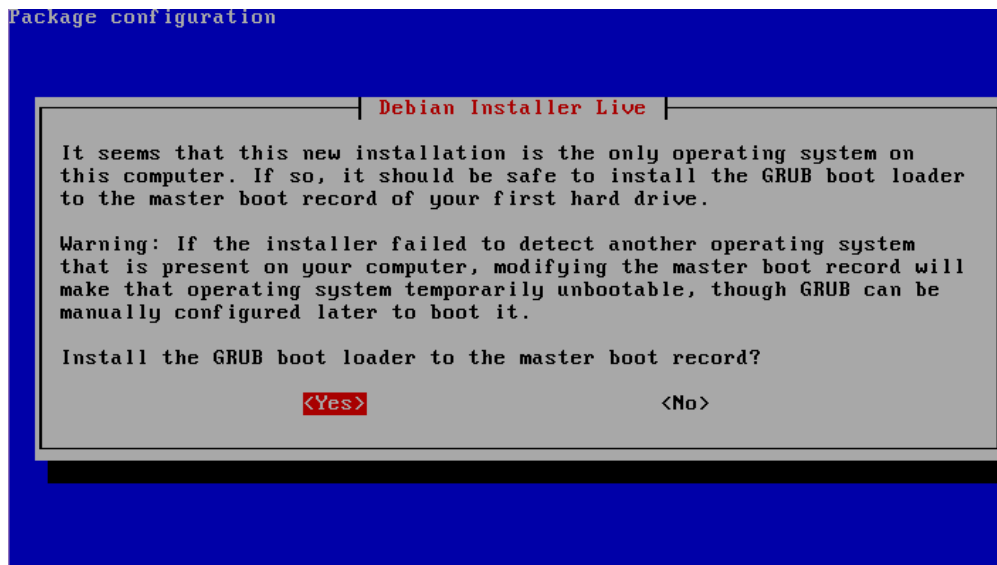


Figura 2.13: Instalación de observium: Paso 8d.

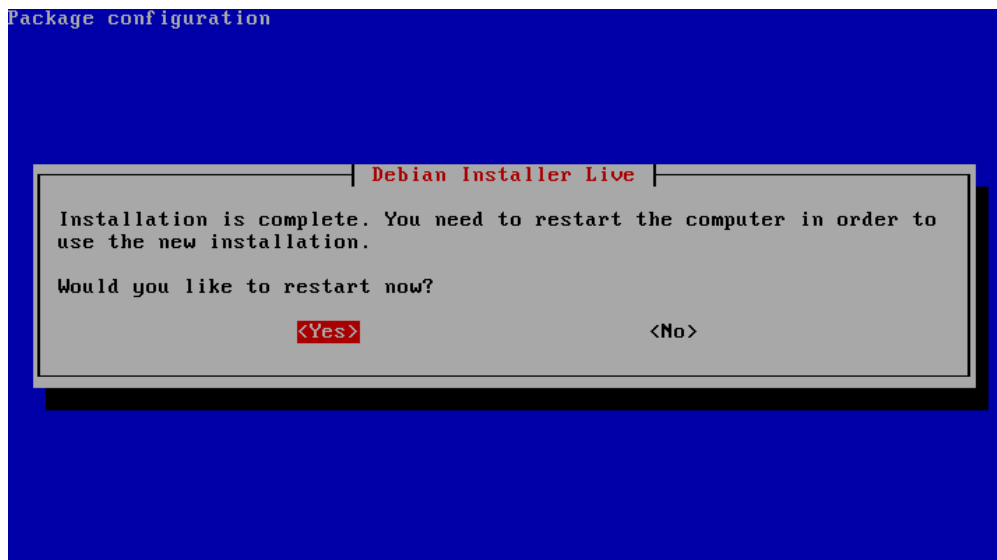


Figura 2.14: Instalación de observium: Paso 8e.

**Paso 9:** Se establecen las contraseñas para la cuenta y la base de datos, así como un correo electrónico, como se observa en las imágenes debajo.

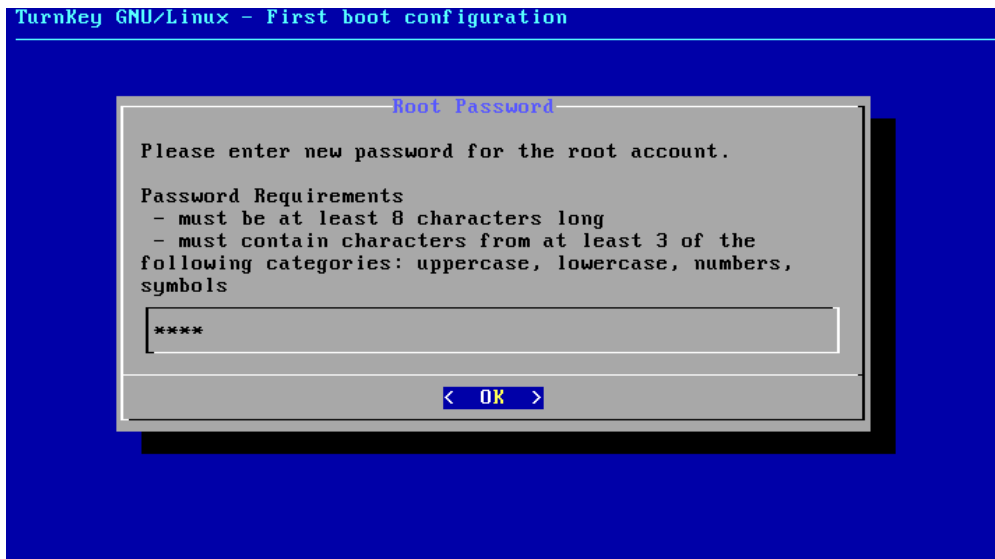


Figura 2.15: Instalación de observium: Paso 9a.

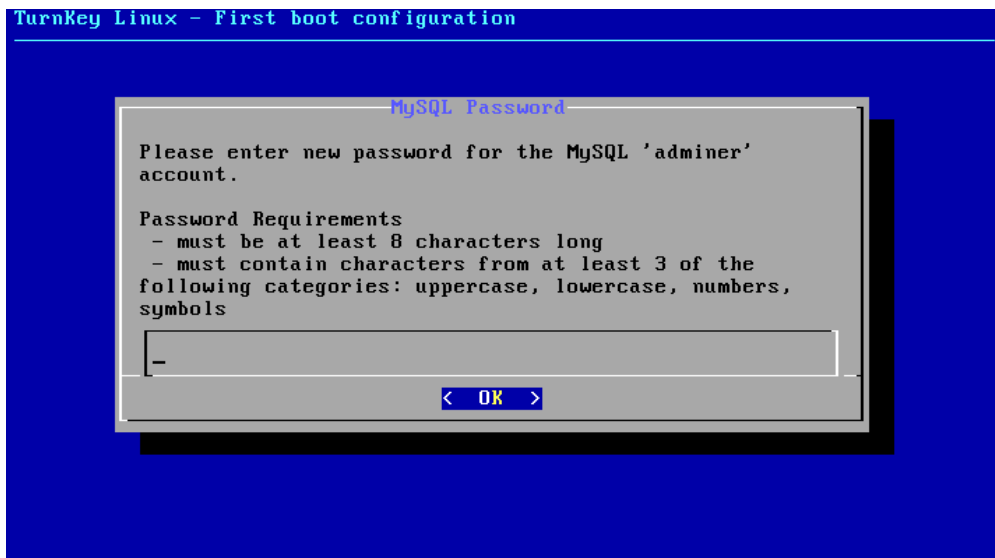


Figura 2.16: Instalación de observium: Paso 9b.



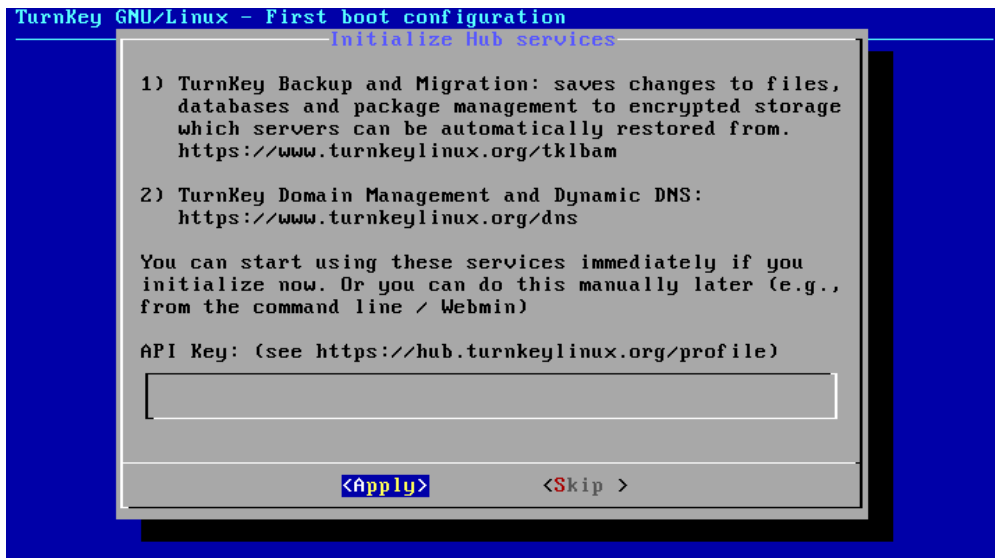


Figura 2.17: Instalación de observium: Paso 9c.

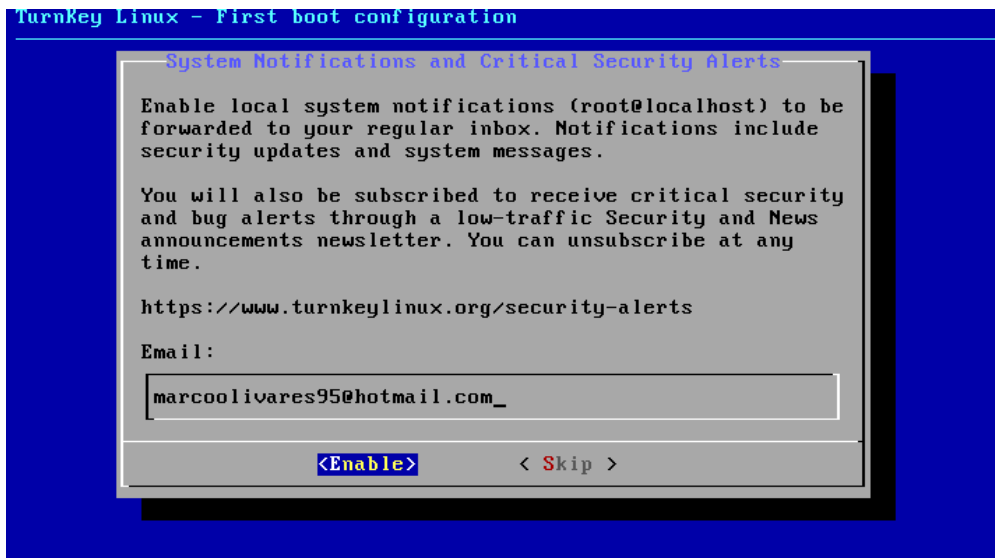


Figura 2.18: Instalación de observium: Paso 9d.

**Paso 10:** Se selecciona la opción "instalar" y posteriormente "reboot". Como se muestra a continuación.

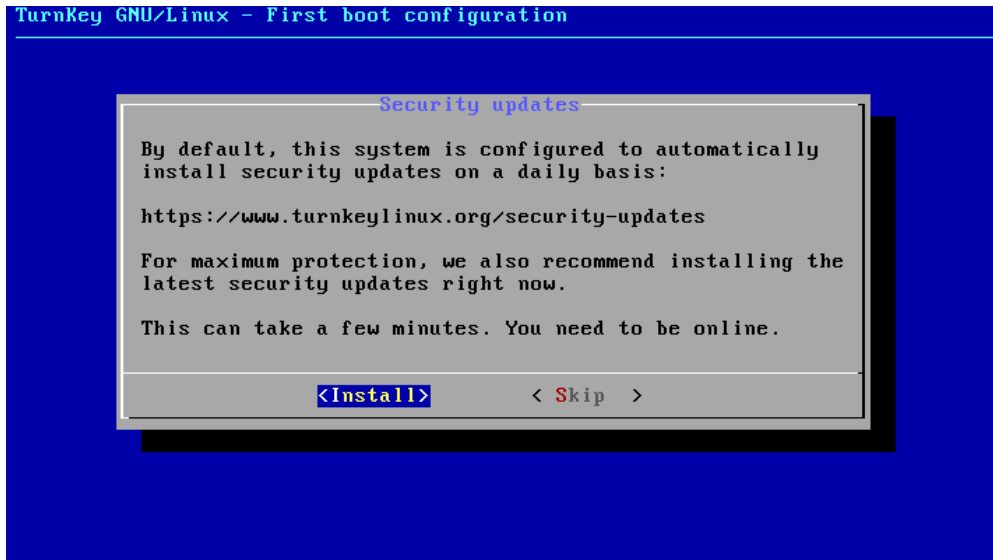


Figura 2.19: Instalación de observium: Paso 10a.

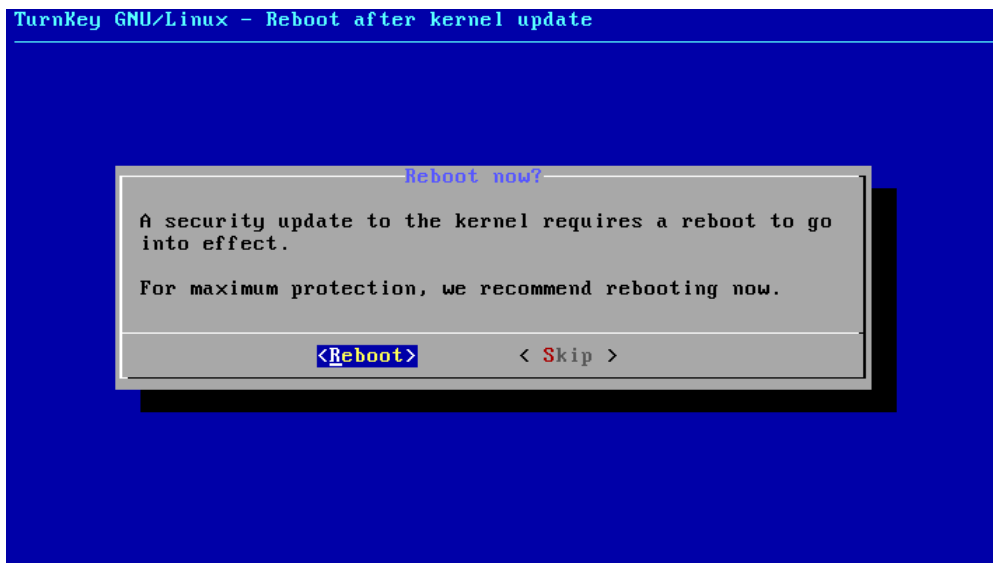


Figura 2.20: Instalación de observium: Paso 10b.

## 2.2. Operaciones SNMP

### 2.2.1. Ejercicio MIB

Para demostrar las consultas SNMP y la forma para acceder a los objetos de la MIB se realizan las siguientes consultas.

**consulta 1:** ¿Cuándo fue el último reinicio (Día, hora y minuto) de los agentes?

```
marco@marco-VirtualBox:~$ snmpget -v2c -c comunidad3 192.168.0.5 1.3.6.1.2.1.1.3.0
iso.3.6.1.2.1.1.3.0 = Timeticks: (19877377) 2 days, 7:12:53.77
marco@marco-VirtualBox:~$ snmpget -v2c -c unidadASR localhost 1.3.6.1.2.1.1.3.0
iso.3.6.1.2.1.1.3.0 = Timeticks: (1541825) 4:16:58.25
```

Figura 2.21: Consulta SNMP1.

**consulta 2:** ¿Cuántas interfaces Ethernet tienen?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.2.2.1.2
iso.3.6.1.2.1.2.2.1.2.1 = STRING: "lo"
iso.3.6.1.2.1.2.2.1.2.2 = STRING: "Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller"
iso.3.6.1.2.1.2.2.1.2.3 = STRING: "Qualcomm Atheros AR9485 Wireless Network Adapter"
```

Figura 2.22: Consulta SNMP2.

**consulta 3:** ¿Cuál es la velocidad (en MBPS) de esas interfaces?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.2.2.1.5
iso.3.6.1.2.1.2.2.1.5.1 = Gauge32: 10000000
iso.3.6.1.2.1.2.2.1.5.2 = Gauge32: 10000000
iso.3.6.1.2.1.2.2.1.5.3 = Gauge32: 0
```

Figura 2.23: Consulta SNMP3.

**consulta 4:** ¿Cuál es la interfaz que ha recibido el mayor número de octetos?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.2.2.1.10
iso.3.6.1.2.1.2.2.1.10.1 = Counter32: 144603
iso.3.6.1.2.1.2.2.1.10.2 = Counter32: 0
iso.3.6.1.2.1.2.2.1.10.3 = Counter32: 1924780
```

Figura 2.24: Consulta SNMP4.

**consulta 5:** Indica cuál interfaz de red ha recibido el mayor número de octetos

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.2.2.1.10
iso.3.6.1.2.1.2.2.1.10.1 = Counter32: 144603
iso.3.6.1.2.1.2.2.1.10.2 = Counter32: 0
iso.3.6.1.2.1.2.2.1.10.3 = Counter32: 1924780
```

Figura 2.25: Consulta SNMP5.

**consulta 6:** ¿Cuál es la MAC de esa interfaz?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.2.2.1.6
iso.3.6.1.2.1.2.2.1.6.1 = ""
iso.3.6.1.2.1.2.2.1.6.2 = Hex-STRING: 2C 56 DC 0A 61 DF
iso.3.6.1.2.1.2.2.1.6.3 = Hex-STRING: 80 A5 89 2A 4E B4
```

Figura 2.26: Consulta SNMP6.

**consulta 7:** ¿Cuántos mensajes ICMP ha recibido el agente?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.5.1.0
iso.3.6.1.2.1.5.1.0 = Counter32: 80
```

Figura 2.27: Consulta SNMP7.

**consulta 8:** ¿Cuántas entradas tiene la tabla de enrutamiento IP?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.4.21
iso.3.6.1.2.1.4.21.1.1.0.0.0.0 = IpAddress: 0.0.0.0
iso.3.6.1.2.1.4.21.1.1.169.254.0.0 = IpAddress: 169.254.0.0
iso.3.6.1.2.1.4.21.1.1.192.168.0.0 = IpAddress: 192.168.0.0
iso.3.6.1.2.1.4.21.1.2.0.0.0.0 = INTEGER: 3
iso.3.6.1.2.1.4.21.1.2.169.254.0.0 = INTEGER: 3
iso.3.6.1.2.1.4.21.1.2.192.168.0.0 = INTEGER: 3
iso.3.6.1.2.1.4.21.1.3.0.0.0.0 = INTEGER: 1
iso.3.6.1.2.1.4.21.1.3.169.254.0.0 = INTEGER: 0
iso.3.6.1.2.1.4.21.1.3.192.168.0.0 = INTEGER: 0
iso.3.6.1.2.1.4.21.1.7.0.0.0.0 = IpAddress: 192.168.0.1
iso.3.6.1.2.1.4.21.1.7.169.254.0.0 = IpAddress: 0.0.0.0
iso.3.6.1.2.1.4.21.1.7.192.168.0.0 = IpAddress: 0.0.0.0
iso.3.6.1.2.1.4.21.1.8.0.0.0.0 = INTEGER: 4
iso.3.6.1.2.1.4.21.1.8.169.254.0.0 = INTEGER: 3
iso.3.6.1.2.1.4.21.1.8.192.168.0.0 = INTEGER: 3
iso.3.6.1.2.1.4.21.1.9.0.0.0.0 = INTEGER: 2
iso.3.6.1.2.1.4.21.1.9.169.254.0.0 = INTEGER: 2
iso.3.6.1.2.1.4.21.1.9.192.168.0.0 = INTEGER: 2
iso.3.6.1.2.1.4.21.1.11.0.0.0.0 = IpAddress: 0.0.0.0
iso.3.6.1.2.1.4.21.1.11.169.254.0.0 = IpAddress: 255.255.0.0
iso.3.6.1.2.1.4.21.1.11.192.168.0.0 = IpAddress: 255.255.255.0
iso.3.6.1.2.1.4.21.1.13.0.0.0.0 = OID: ccitt.0
iso.3.6.1.2.1.4.21.1.13.169.254.0.0 = OID: ccitt.0
iso.3.6.1.2.1.4.21.1.13.192.168.0.0 = OID: ccitt.0
```

Figura 2.28: Consulta SNMP8.

**consulta 9:** ¿Cuántos datagramas UDP ha recibido el agente?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.7.1
iso.3.6.1.2.1.7.1.0 = Counter32: 1711
```

Figura 2.29: Consulta SNMP9.

**consulta 10:** ¿El agente ha recibido mensajes TCP? ¿Cuántos?

```
maestroluminuscom@maestro:~$ snmpwalk -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.6.10  
iso.3.6.1.2.1.6.10.0 = Counter32: 1787
```

Figura 2.30: Consulta SNMP10.

**consulta 11:** ¿Cuántos mensajes EGP ha recibido el agente?

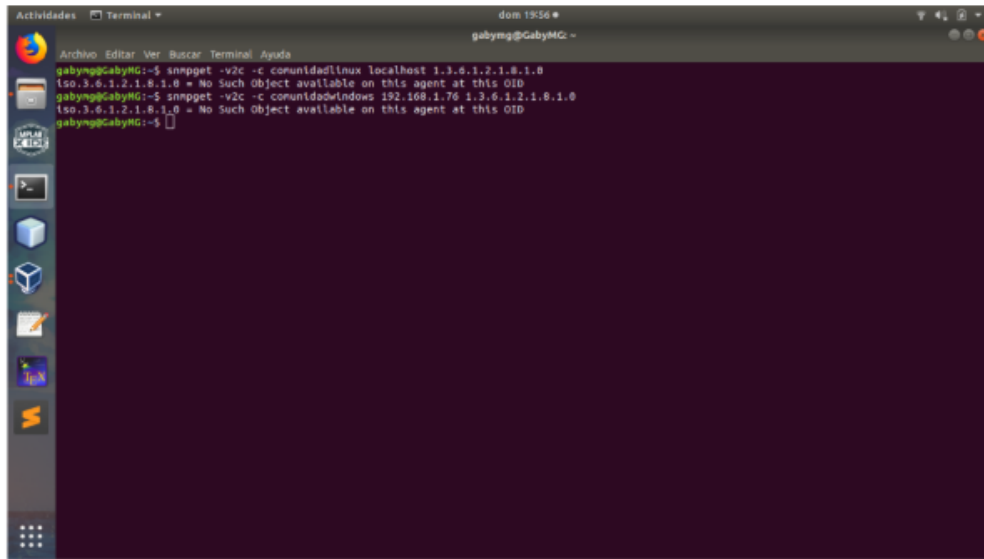


Figura 2.31: Consulta SNMP11.

**consulta 12:** Indica el Sistema Operativo que del agente.

```
maestroluminuscom@maestro:~$ snmpget -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.1.1.0  
iso.3.6.1.2.1.1.1.0 = STRING: "Linux maestro.luminus.com 4.15.0-34-generic #37-Ubuntu SMP Mon Aug 27 15:21:48 UTC 2018 x86_64"
```

Figura 2.32: Consulta SNMP12.

**consulta 13:** Modifica el nombre del contacto o la ubicación del sistema de un agente

```
maestroluminuscom@maestro:~$ snmpget -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.1.1.0  
iso.3.6.1.2.1.1.1.0 = STRING: "Linux maestro.luminus.com 4.15.0-34-generic #37-Ubuntu SMP Mon Aug 27 15:21:48 UTC 2018 x86_64"
```

Figura 2.33: Consulta SNMP13.

**consulta 14:** Dibuja la MIB del agente.

```
maestroluminuscom@maestro:~$ snmpget -v2c -c comunidadEquipo3_grupo4CM1 192.168.0.7 1.3.6.1.2.1.1.1.0  
iso.3.6.1.2.1.1.1.0 = STRING: "Linux maestro.luminus.com 4.15.0-34-generic #37-Ubuntu SMP Mon Aug 27 15:21:48 UTC 2018 x86_64"
```

Figura 2.34: Consulta SNMP14.

## 2.3. Modelo de administración de red de SNMP

A continuación se muestran los resultados de la herramienta creada por medio de las siguientes operaciones.

**Operación 1:** Se agrega un agente LINUX con el nombre "comunidadEquipo3grupo4CM1". Haciendo una consulta SNMP con la comunidad dada de alta obtenemos la información de ésta misma, como se muestra en la imagen debajo.

```
maestro@luminuscom:~/PruebaFlask/practicairedes3$ snmpget -v2c -c comunidadEquipo3_grupo4CM1 8.12.0.213 1.3.6.1.2.1.1.1.0  
iso.3.6.1.2.1.1.1.0 = STRING: "Linux maestro.luminus.com 4.15.0-34-generic #37-Ubuntu SMP Mon Aug 27 15:21:48 UTC 2018 x86_64"
```

Figura 2.35: Operación 1a.

Para el agente de Windows, se configuró con el mismo nombre del agente para Linux, la diferencia radica en que la ip de ambos agentes es distinta. En la siguiente figura se puede observar que al hacer la consulta snmp, el resultado arroja información de software "Windows".

```
maestro@luminuscom:~/PruebaFlask/practicairedes3$ snmpget -v2c -c comunidadEquipo3_grupo4CM1 8.12.0.217 1.3.6.1.2.1.1.1.0  
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: Intel64 Family 6 Model 78 Stepping 3 AT/AT COMPATIBLE - Software: Windows Version 6.3 (Build 15063 Multiprocessor Free)"
```

Figura 2.36: Operación 1b.

**Operación 2:** Se agrega el agente windows previamente creado como agente a observium, como se puede ver a continuación.

Adding host agentWindows community comunidadEquipo3\_grupo4CM1 port 161  
Trying v2c: comunidadEquipo3\_grupo4CM1 ...  
Device added (id = 4):

**Basic Configuration**

Hostname: agentWindows

Skip RING: ☐ Skip ICMP echo checks

Protocol Version: v2c

Transport: UDP

Port: 161

Timeout: 1

Retries: 5

Ignore existing RRDs: ☒ Ignore pre-existing RRD directory and files

**Authentication Configuration**

SNMP Community: comunidadEquipo3\_grupo4CM1

Figura 2.37: Operación 2a.

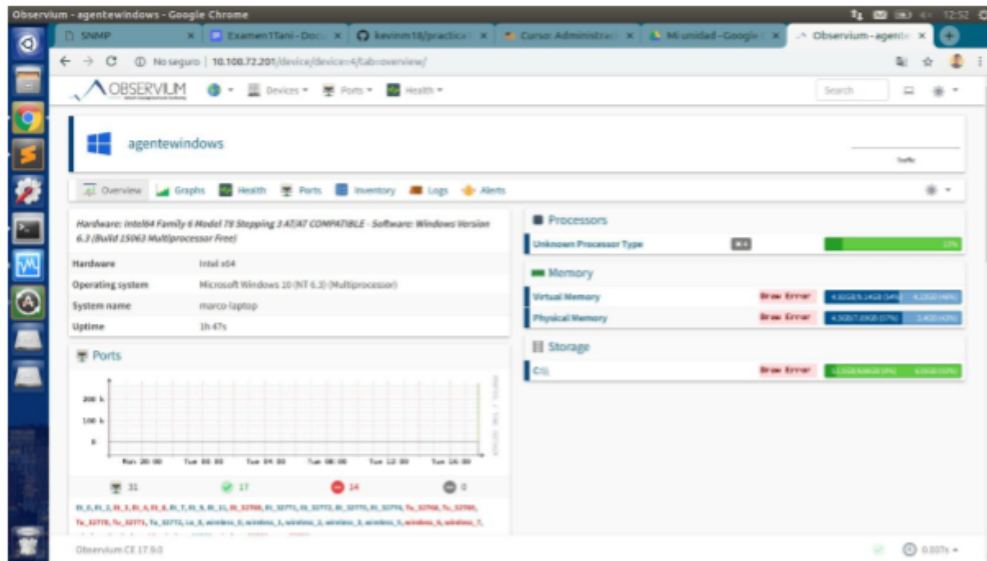


Figura 2.38: Operación 2b.



Figura 2.39: Operación 2c.

**Operación 3:** Se agrega un nuevo agente usando nuestra herramienta creada, para ello se requieren los siguientes datos del agente: Nombre (UbuntuExamenEquipo3), host (8.12.0.123), comunidad (comunidadEquipo3\_grupo4CM1), version SNMP (v2) y puerto (8012), al presionar el botón “agregar agente”.

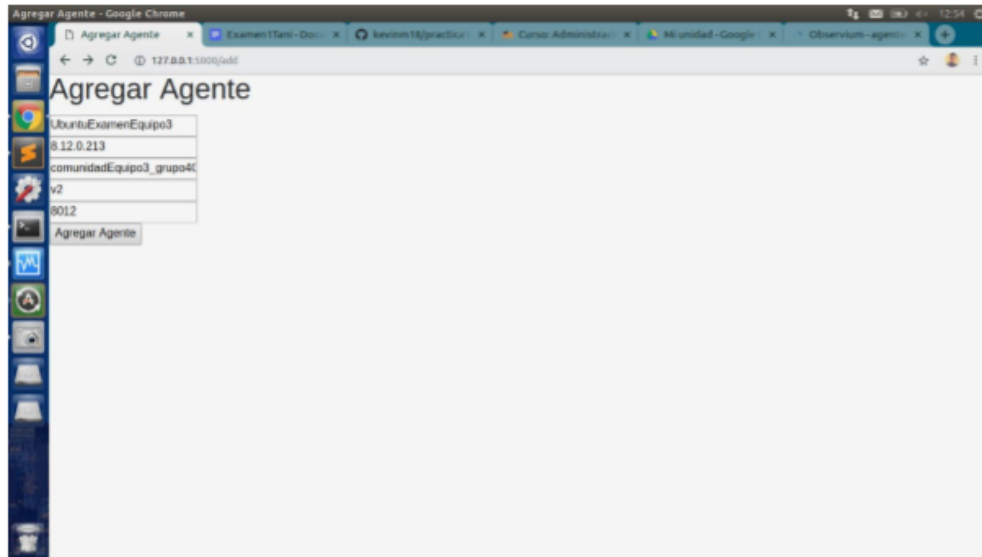


Figura 2.40: Operación 3c.

Como se puede ver en la lista de agentes agregados, se muestra el agente recién añadido en el último renglón, con los datos arriba descritos, mostrando que el agente se añadió correctamente. Asimismo, se muestra el estado de las interfaces del agente agregado.

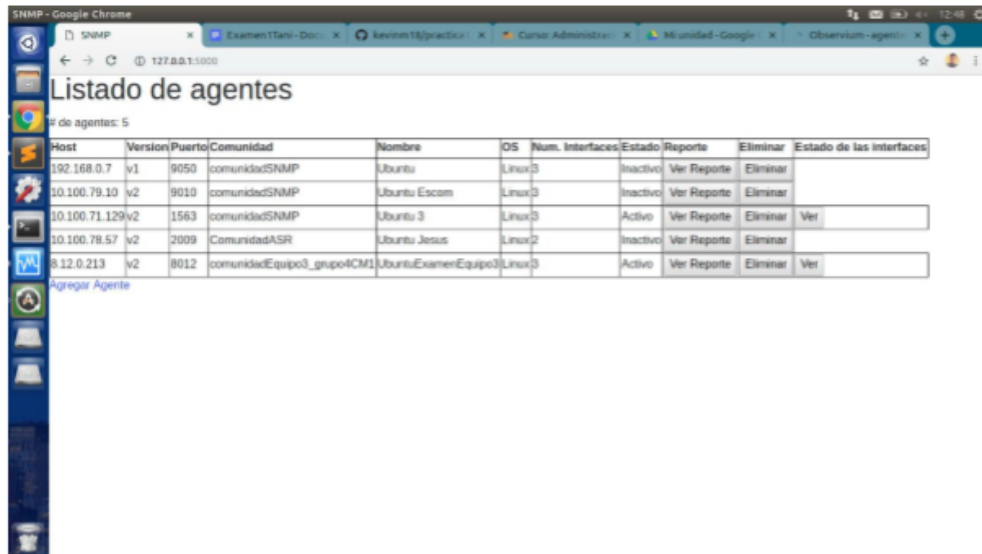


Figura 2.41: Operación 3b.

Finalmente, aquí se muestran las funciones (de nuestra herramienta) que hacen posible agregar nuevos agentes. Primeramente tenemos la función “agregar” que realiza consultas SNMP para comprobar que los datos introducidos sean correctos, de ser esto cierto y al lograr exitosamente las consultas SNMP, se inserta el agente,



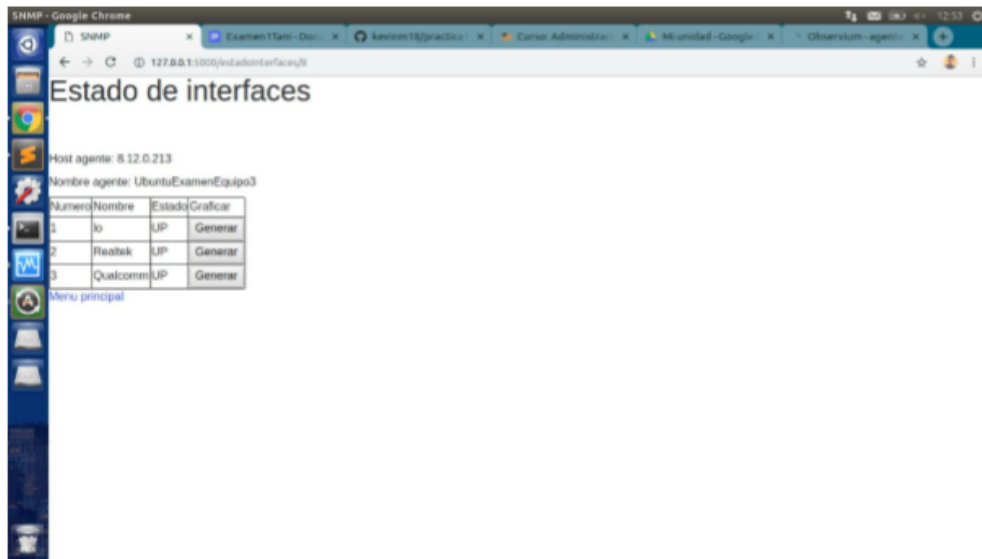


Figura 2.42: Operación 3c.

por medio de la función “insertar”, ambas funciones se muestran en las imágenes debajo.

```
''' Metodos de formulario '''

# Metodo ejecutado por el formulario al agregar un nuevo agente
@app.route('/addAgent', methods = ['post', 'get'])
def agregar():
    host = request.form['host']
    comunidad = request.form['comunidad']
    nombre = request.form['nombre']
    version = request.form['version']
    puerto = request.form['puerto']
    if host and comunidad and nombre and version and puerto:
        sistemaOperativo = parseResultAfterEquals(snmGet(comunidad, host, OID_SYSINFO))
        interfaces = parseResultAfterEquals(snmGet(comunidad, host, OID_INTERFACES))
        insertar(host, comunidad, nombre, version, puerto, sistemaOperativo, interfaces)
        return redirect(url_for('main'))
    else:
        return json.dumps({'html': '<span> Llene todos los campos </span>'})
```

Figura 2.43: Operación 3d.

```
# Inserta un nuevo dispositivo a la base de datos, así como sus interfaces.
def insertar(host, comunidad, nombre, version, puerto, sistemaOperativo, interfaces):
    cursor.execute('INSERT INTO agente(hostname, version, puerto, comunidad, nombre, os, num_interfaces) VALUES
    cursor.execute('SELECT MAX(id) from agente')
    idAgente = cursor.fetchone()[0]
    for i in range (int(interfaces)):
        cursor.execute('INSERT INTO interfaz_agente(id_agente, id_interfaz) VALUES (%s, %s)', (idAgente, i + 1))
    conn.commit()
```

Figura 2.44: Operación 3e.

También se agregan 5 consultas SNMP sobre objetos de la MIB. Las imágenes debajo muestran las gráficas de cada una, mostrando además el nombre de la variable analizada debajo de la gráfica. Dichas gráficas se genera con la herramienta realizada.

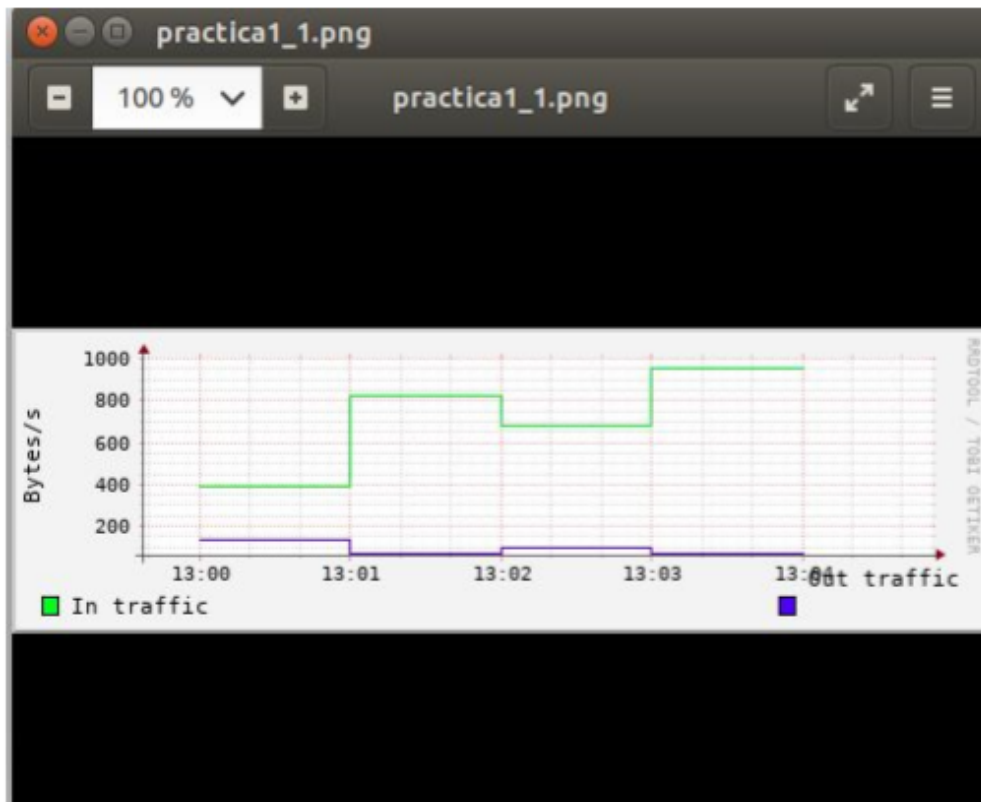


Figura 2.45: Operación 3f.

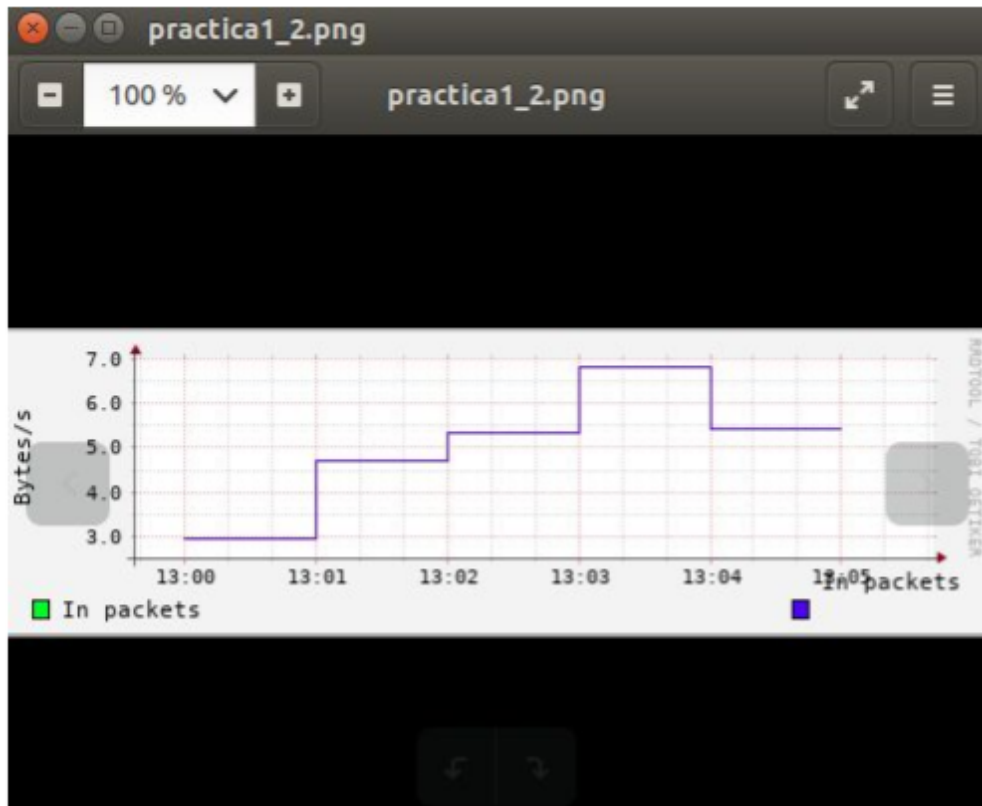


Figura 2.46: Operación 3g.

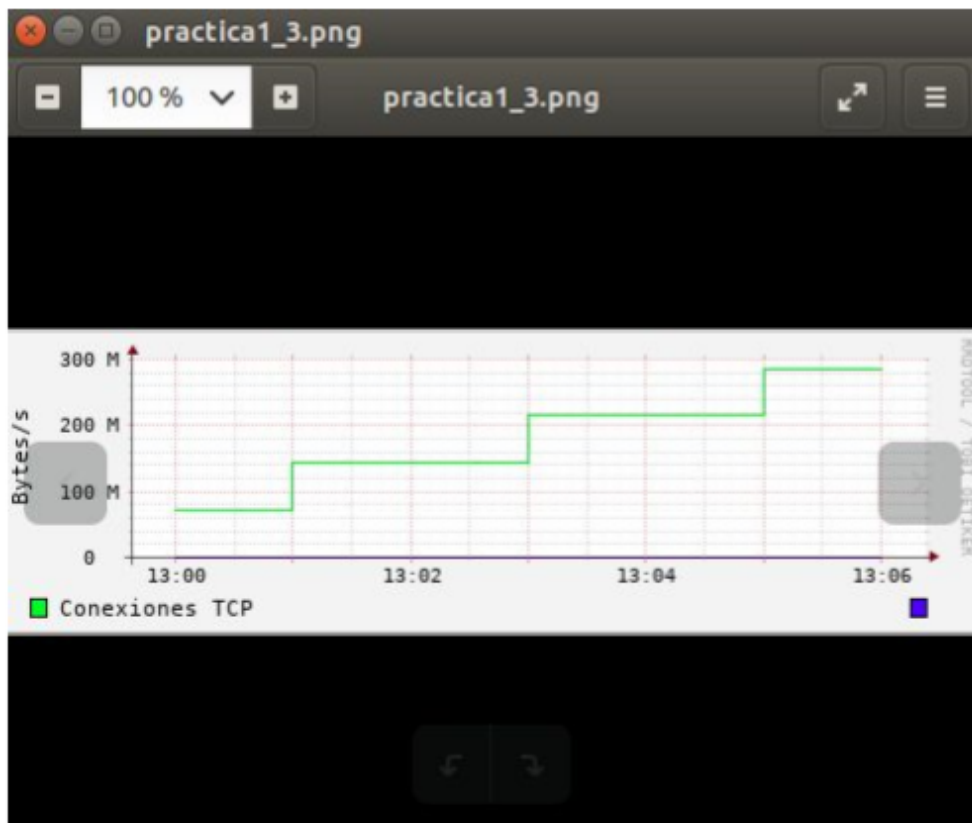


Figura 2.47: Operación 3h.

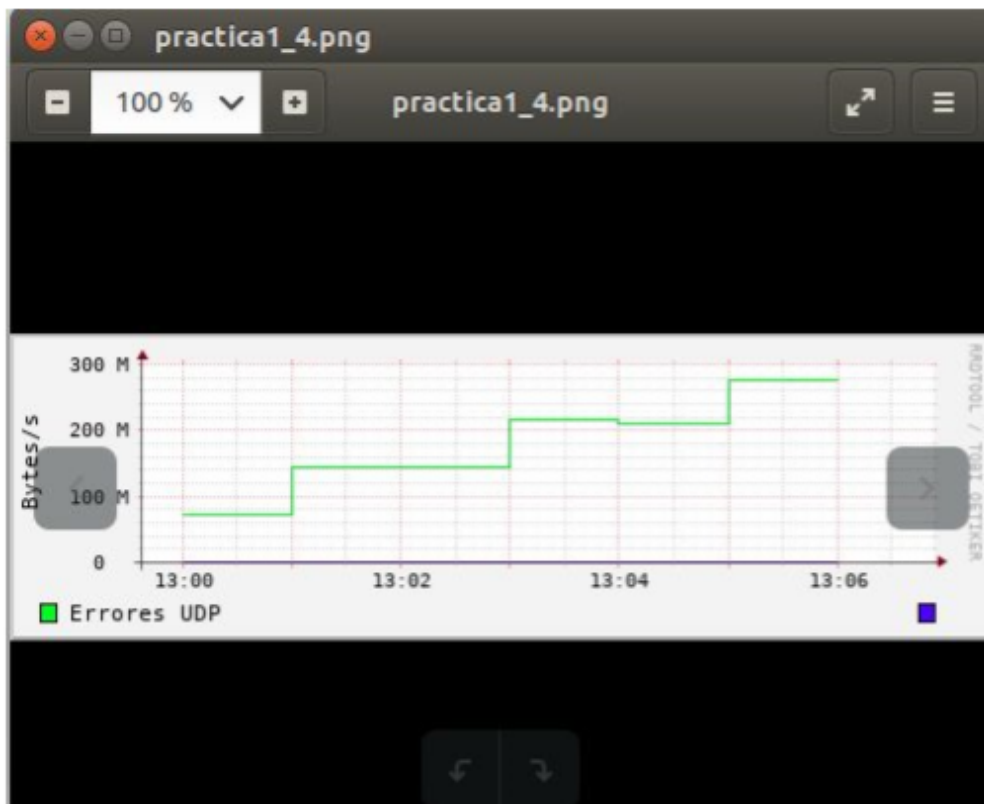


Figura 2.48: Operación 3i.

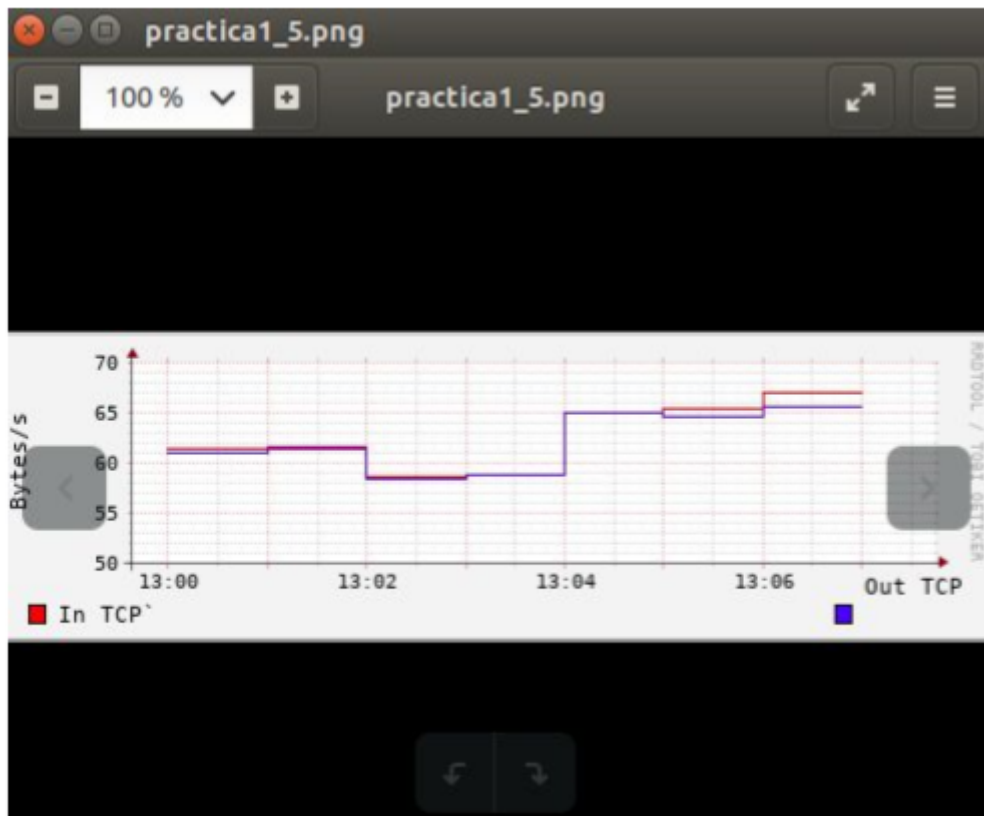


Figura 2.49: Operación 3j.

**Operación 4:** Se agrega agente3 de la maestra.

SNMP - Google Chrome

127.0.0.1:5000

### Listado de agentes

Nº de agentes: 6

Host	Version	Puerto	Comunidad	Nombre	OS	Num. Interfaces	Estado	Reporte	Eliminar	Estado de las interfaces
192.168.0.7	v1	9050	comunidadSNMP	Ubuntu	Linux3		Inactivo	Ver Reporte	Eliminar	
10.106.79.10	v2	9010	comunidadSNMP	Ubuntu Escom	Linux3		Inactivo	Ver Reporte	Eliminar	
10.106.71.129	v2	1563	comunidadSNMP	Ubuntu 3	Linux3		Activo	Ver Reporte	Eliminar	Ver
10.106.78.57	v2	2009	ComunidadASR	Ubuntu Jesus	Linux2		Inactivo	Ver Reporte	Eliminar	
8.12.0.213	v2	8012	comunidadEquipo3_grupo4CM1	UbuntuExamenEquipo3	Linux3		Activo	Ver Reporte	Eliminar	Ver
8.12.0.100	v2	1024	variation/virtualtable	agente3	No	0	Activo	Ver Reporte	Eliminar	Ver

[Agregar Agente](#)

Figura 2.50: Operación 4a.

Se muestra el OID que construimos para nuestro apartado (apartado 3 del la tabla escrita en el examen en el problema 4), ifInUcastPkts

```
maestroluminuscom@maestro:~/PruebaFlask/practicairedes3$ snmpget -v2c -c variation/virtualtable 8.12.0.100:1024 1.3.6.1.2.1.2.2.1.12.1
iso.3.6.1.2.1.2.2.1.12.1 = Counter32: 2758261
```

Figura 2.51: Operación 4b.

Aquí se muestra la imagen de lo antes analizado.

Debajo se muestra el código de muestra que todo se hace de manera concurrente.

**Operación 5:** Debajo se muestra el nodo antes de la información

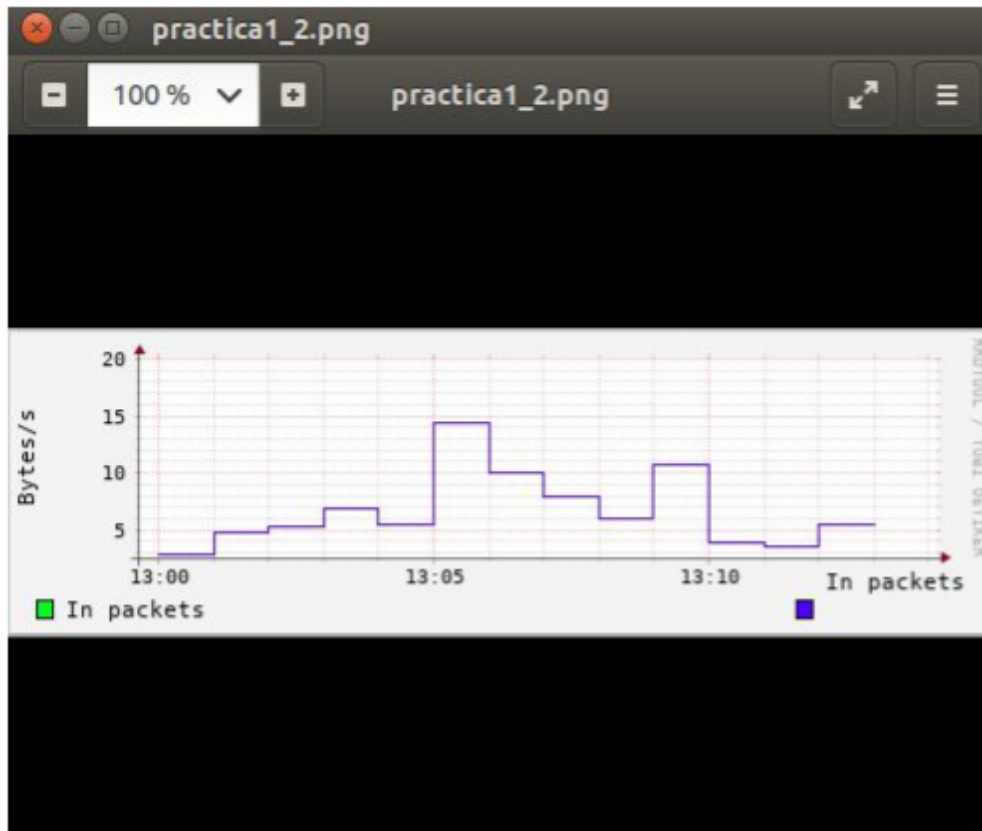


Figura 2.52: Operación 4c.

```
app.route('/generarGrafica/<idInterfaz>.<idAgente>', methods = ['post', 'get'])
def generarGrafica(idInterfaz, idAgente):
    cursor.execute('SELECT hostname, comunidad FROM agente WHERE id = ' + idAgente + ' ')
    dataAgente = cursor.fetchone()
    rrd1()
    hiloGrafica = threading.Thread(target = graphRRD, args = (idInterfaz, dataAgente[0], dataAgente[1],))
    hiloGrafica.start()
    hiloRRD3 = threading.Thread(target = executeRRD3, args = ())
    hiloRRD3.start()
    return redirect(url_for('main'))
```

Figura 2.53: Operación 4d.

```
maestroluminuscom@maestro: ~
maestroluminuscom@maestro:~$ snmpget -v2c -c comunidadEquipo3_grupo4CM1 10.100.7
1.161 1.3.6.1.2.1.1.4
Error in packet
Reason: (noSuchName) There is no such variable name in this MIB.
Failed object: iso.3.6.1.2.1.1.4

maestroluminuscom@maestro:~$ snmpget -v2c -c comunidadEquipo3_grupo4CM1 10.100.7
1.161 1.3.6.1.2.1.1.4.0
iso.3.6.1.2.1.1.4.0 = ""
maestroluminuscom@maestro:~$
```

Figura 2.54: Operación 5.



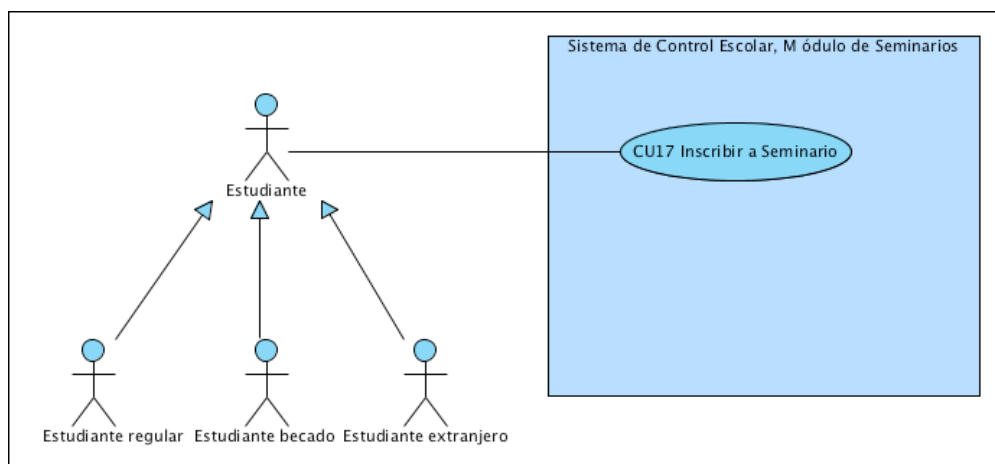


Figura 3.1: Diagrama de Casos de Uso del sistema.



## 3.1. CU1 Asignar Membresía

### 3.1.1. Descripción completa

Ayudar al ejecutivo vendedor a asignar un tipo de membresía a un usuario.








### 3.1.2. Atributos importantes

<b>Caso de Uso:</b>	CU1 Asignar Membresía
<b>Versión:</b>	0.1
<b>Actor:</b>	Ejecutivo Vendedor
<b>Propósito:</b>	Que el ejecutivo vendedor asocie un tipo de membresía para un determinado usuario, cuando éste desee hacer uso de las instalaciones, para que se pueda generar una orden de pago y se active la membresía del usuario.
<b>Entradas:</b>	Cliente al que se le asignará la membresía, tipo de membresía, duración de la membresía
<b>Origen:</b>	Teclado y mouse
<b>Salidas:</b>	Membresías disponibles, información sobre las membresías (servicios, cursos, áreas), mensaje de venta exitosa, tipo de membresía que se le asignó al usuario, servicios y cursos de los que dispone el usuario, orden de pago indicando el número de cuenta y monto a pagar, fecha límite para pagar, referencia.
<b>Destino:</b>	Pantalla
<b>Precondiciones:</b>	El usuario debe estar registrado en el sistema y no debe contar con una membresía vigente.
<b>Postcondiciones:</b>	El usuario quedará asociado a un tipo de membresía
<b>Errores:</b>	E1: El usuario ya posee con una membresía vigente.
<b>Tipo:</b>	Caso de uso que extiende del CU1
<b>Observaciones:</b>	
<b>Autor:</b>	Marco Antonio Olivares García.

### 3.1.3. Trayectorias del Caso de Uso



**Trayectoria principal:** Principal

- 1 Da clic en el botón **Asignar membresía** en el caso de uso CU17Login mediante la IU23 Pantalla de Consultas de Clientes
- 2 Verifica que el usuario no tenga una membresía activa. [Trayectoria A]
- 3 Consulta las membresías registradas.
- 4 Muestra las membresías registradas mediante la IU23 Pantalla de Membresías disponibles.
- 5 Da clic en en boton **Información**.
- 6 Consulta la información de la membresía seleccionada.
- 7 Muestra la información de la membresía seleccionada.
- 8 Cierra la información mostrada acerca de la membresía.

- 9  Selecciona la duración de la membresía.
  - 10  Calcula el monto a pagar.
  - 11  Muestra el monto a pagar.
  - 12  Da clic en en boton Confirmar.
  - 13  Informa que se asignó de forma exitosa la membresía al usuario, muestra el nombre de la membresía, la duración de la misma, monto a pagar, número de cuenta bancaria y fecha límite para pagar mediante la  IU24 Pantalla de asignación de membresía exitosa.
  - 14  Da clic en en boton Finalizar.
- - - Fin del caso de uso.

### Trayectoria alternativa A:

**Condición:** El usuario tiene una membresía activa.

- A1  Muestra el Mensaje “El usuario ya posee una membresía activa”.
  - A2  Oprime el botón Salir.
  - A3 Termina el caso de uso.
- - - Fin de la trayectoria.



## 3.2. CU2 Registrar ingreso a área

### 3.2.1. Descripción completa





Registrar cuando un usuario acceda a una determinada área, así como la hora a la que ingresa.

### 3.2.2. Atributos importantes

Caso de Uso:	CU2 Registrar ingreso a área
Versión:	0.1
Actor:	Usuario
Propósito:	Almacenar en un registro la hora de entrada del usuario a un área, para saber si tiene acceso a la misma, y en dado caso contrario, se le cobre un cargo extra por el uso del área
Entradas:	Curp del usuario
Origen:	Tarjeta del usuario
Salidas:	
Destino:	
Precondiciones:	El usuario debe estar registrado en el sistema y debe contar con una tarjeta de usuario
Postcondiciones:	El área a la que ingrese contará con un usuario más.
Errores:	
Tipo:	Caso de uso primario
Observaciones:	
Autor:	Marco Antonio Olivares García.


### 3.2.3. Trayectorias del Caso de Uso

#### Trayectoria principal: Principal

- 1  Pasa por los sensores ubicados en la entrada del área.
  - 2  Consulta la información correspondiente del usuario al que está asociada la tarjeta que pasó por los sensores.
  - 3  Registra la hora en la cual el usuario está ingresando al área y el curp de usuario de la tarjeta.
  - 4  Verifica que el usuario tenga acceso al área. [Trayectoria A]
- - - Fin del caso de uso.

#### Trayectoria alternativa A:

Condición: El usuario no tiene acceso al área

- A1  En el caso de uso CU3 se calculará el costo adicional que tendrá el uso del área.
  - A2 Termina el caso de uso.
- - - Fin de la trayectoria.



### 3.3. CU3 Registrar salida de un área

#### 3.3.1. Descripción completa






Registrar cuando un usuario sale de una determinada área, así como la hora a la que sale.

#### 3.3.2. Atributos importantes

Caso de Uso:	CU3 Registrar salida de un área
Versión:	0.1
Actor:	Usuario
Propósito:	Almacenar en un registro la hora de salida del usuario de un área, actualizar el tiempo de acceso disponible para esa área en su membresía y en algunos casos, realizar un cargo extra por hacer uso de un área no incluida en su membresía
Entradas:	Curp del usuario
Origen:	Tarjeta del usuario
Salidas:	
Destino:	
Precondiciones:	El usuario debe haber ingresado previamente a un área
Postcondiciones:	El área a la que ingrese contará con un usuario menos.
Errores:	
Tipo:	Caso de uso primario
Observaciones:	
Autor:	Marco Antonio Olivares García.



#### 3.3.3. Trayectorias del Caso de Uso

##### Trayectoria principal: Principal

- 1  Pasa por los sensores ubicados en la entrada del área.
  - 2  Consulta la información correspondiente al usuario que está asociado a la tarjeta que pasó por los sensores.
  - 3  Busca en el registro el curp del usuario de la tarjeta y registra la hora de salida.
  - 4  Calcula el tiempo que el usuario estuvo dentro del área.
  - 5  Resta el tiempo que el usuario pasó dentro del área al tiempo disponible en la membresía. [Trayectoria A]
- - - Fin del caso de uso.

##### Trayectoria alternativa A:

Condición: El usuario no tiene acceso al área

- A1  Se calcula el costo adicional que tendrá el haber usado el área durante el tiempo que permaneció el usuario dentro del área.
- A2  Se suma el costo adicional del uso de área no autorizada al monto a pagar de la membresía para el próximo mes.

**A3** Termina el caso de uso.  
- - - *Fin de la trayectoria.*



### 3.4. CU4 Registrar pago de membresía

#### 3.4.1. Descripción completa

Ayudar al ejecutivo vendedor a registrar el pago de una membresía efectuado por el cliente.


#### 3.4.2. Atributos importantes

<b>Caso de Uso:</b>	CU4 Registrar pago de membresía
<b>Versión:</b>	0.1
<b>Actor:</b>	Ejecutivo Vendedor
<b>Propósito:</b>	Que el ejecutivo vendedor registre el pago de una membresía una vez que el usuario haya efectuado el pago correspondiente, para que su membresía quede activa y pueda hacer uso de las instalaciones y servicios del gimnasio.
<b>Entradas:</b>	nombre del cliente, forma de pago, persona que registra el pago, monto pagado, número de referencia, fecha en que se realizó el pago
<b>Origen:</b>	Teclado y mouse.
<b>Salidas:</b>	Mensaje de pago exitoso, vendedores ejecutivos que trabajan en la sucursal.
<b>Destino:</b>	Pantalla
<b>Precondiciones:</b>	El usuario debe estar registrado y debe haber comprado una membresía.
<b>Postcondiciones:</b>	La membresía del usuario quedará activa.
<b>Errores:</b>	
<b>Tipo:</b>	Caso de uso que extiende del CU1 Consultar clientes.
<b>Observaciones:</b>	
<b>Autor:</b>	Marco Antonio Olivares García.

#### 3.4.3. Trayectorias del Caso de Uso








##### Trayectoria principal: Principal

- 1 Da clic en el botón **Registrar pago** en el caso de uso CU17Login mediante la IU23 Pantalla de Consultas de Clientes
- 2 Consulta los ejecutivos vendedores que trabajan en la sucursal.
- 3 Despliega los ejecutivos vendedores que trabajan en la sucursal mediante la IU23 Pantalla de Registro de pagos de membresías.
- 4 Selecciona la forma de pago. [Trayectoria A]
- 5 Solicita el monto pagado.
- 6 Introduce la cantidad pagada.
- 7 Solicita el nombre del ejecutivo vendedor que está registrando el pago.
- 8 Selecciona el nombre del ejecutivo vendedor que está registrando el pago.
- 9 Da clic en el botón **Confirmar**
- 10 Verifica que los datos de entrada correspondan al tipo de dato solicitado. [Trayectoria B]
- 11 Informa que se registró el pago de forma exitosa.

- 12  Da clic en en boton Finalizar.  
- - - - *Fin del caso de uso.*



### Trayectoria alternativa A:

**Condición:** El ejecutivo vendedor elige la forma de pago por depósito bancario.

- A1  Solicita la fecha de pago.  
A2  Introduce la fecha de pago.  
A3  Solicita el número de transacción.  
A4  Introduce el número de transacción.  
A5  Solicita el monto pagado.  
A6  Introduce el monto pagado.  
A7  Continúa en el paso 7 del CU4  
- - - - *Fin de la trayectoria.*

### Trayectoria alternativa B:

**Condición:** Los datos de entrada no corresponden con el tipo de dato solicitado

- B1  Muestra el Mensaje “Error en los datos de entrada”.  
B2  Continúa en el paso 4 del CU4  
- - - - *Fin de la trayectoria.*





### 3.5. CU5 Consultar uso de área.

#### 3.5.1. Descripción completa

Facilitar el control de los usuarios que se encuentran usando un área.

#### 3.5.2. Atributos importantes

<b>Caso de Uso:</b>	CU5 Consultar uso de área.
<b>Versión:</b>	0.1
<b>Actor:</b>	Encargado de área.
<b>Propósito:</b>	Que el encargado de área pueda saber cuántos y qué usuarios se encuentran usando una determinada área.
<b>Entradas:</b>	Nombre del área
<b>Origen:</b>	Mouse
<b>Salidas:</b>	Número de usuarios que se encuentran usando el área, usuarios que están usando el área.
<b>Destino:</b>	Pantalla
<b>Precondiciones:</b>	Que el área exista
<b>Postcondiciones:</b>	
<b>Errores:</b>	
<b>Tipo:</b>	Caso de uso primario
<b>Observaciones:</b>	
<b>Autor:</b>	Marco Antonio Olivares García

#### 3.5.3. Trayectorias del Caso de Uso

**Trayectoria principal:** Principal

- 1 Muestra la lista de áreas registradas en la sucursal.
  - 2 Muestra la lista de áreas registradas en la sucursal.
  - 3 Elige el área a consultar vía la IU23 Pantalla de Control de Acceso.
  - 4 Oprime el botón .
  - 5 Consulta el registro de acceso al área seleccionada.
  - 6 Cuenta el número de usuarios que registraron su entrada al área pero que aún no han registrado su salida.
  - 7 Muestra todos los usuarios que registraron su entrada al área pero que aún no han registrado su salida, así como el número de usuarios que se encuentran usando el área.
- - - Fin del caso de uso.

Esta sección se queda deliberadamente en blanco debido a que el diseño de las interfaces dependerá de la plataforma a utilizar por cada equipo.

#### 4.0.1. IU23 Pantalla de Control de Acceso

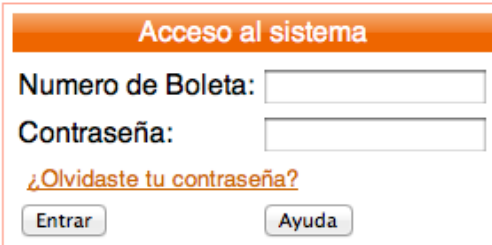
##### Objetivo

Controlar el acceso al sistema mediante una contraseña a fin de que cada usuario acceda solo a las operaciones permitidas para su perfil.

##### Diseño

Esta pantalla aparece al iniciar el sistema. Para ingresar al mismo se debe escribir el Número de Boleta del estudiante y la contraseña de acceso.

### Acceso al sistema



Acceso al sistema

Numero de Boleta:

Contraseña:

[¿Olvidaste tu contraseña?](#)

Figura 4.1: IU23 Pantalla de Control de Acceso.



## Salidas

Ninguna.

## Entradas

Número de Boleta y Contraseña del Estudiante.

## Comandos

- **Entrar**: Verifica que el Estudiante se encuentre registrado y la contraseña sea la correcta. Si la verificación es correcta, se muestra la  IU32 Pantalla de Selección de Seminario.
- **Ayuda**: Muestra la ayuda de esta pantalla  IU50 Pantalla de Ayuda.

## Mensajes

- **MSG5** Error al verificar los datos de acceso, vuelva a intentarlo.

## Modelo del Dominio del problema

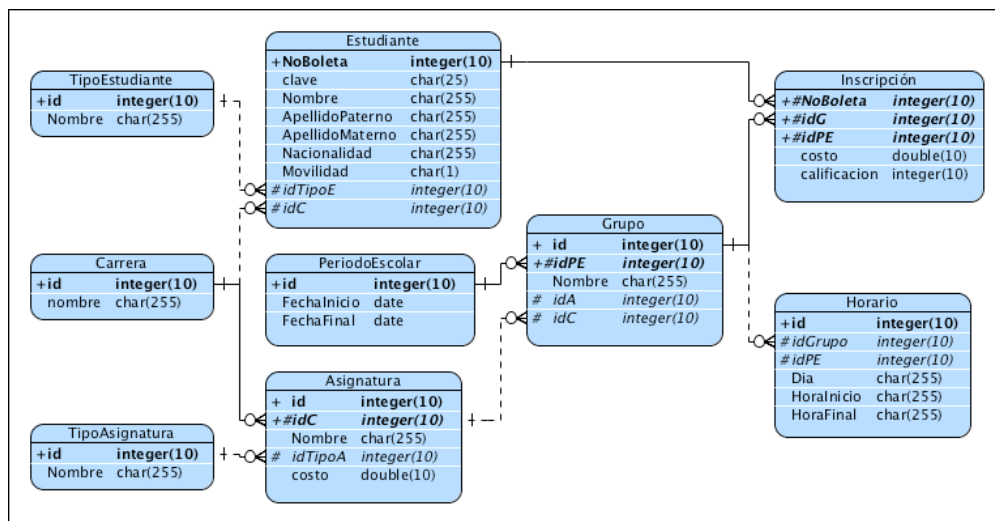


Figura 5.1: Diseño de la Base de Datos.

---

## Bibliografía

---

- [1] D. Guerrero *SNMP: Administración y Mantenimiento de Redes con Linux* [En línea]. LinuxFocus. (S/L), 1998.  
Disponible en <http://redesdecomputadores.umh.es/aplicacion/snmp.htm>