

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**IMPLEMENTACIÓN DE PROTOTIPOS DE SISTEMAS DE  
MONITOREO DE REDES EMPLEANDO DISTINTAS SOLUCIONES  
A NIVEL DE SOFTWARE**

**IMPLEMENTACIÓN DEL PROTOTIPO EMPLEANDO NEMS SOBRE  
UN RASPBERRY PI**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**BRYAN ANDRÉS ENRIQUEZ AUZ**

**DIRECTOR: LEANDRO ANTONIO PAZMIÑO ORTIZ**

**DMQ, agosto 2022**

## CERTIFICACIONES

Yo, BRYAN ANDRES ENRIQUEZ AUZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



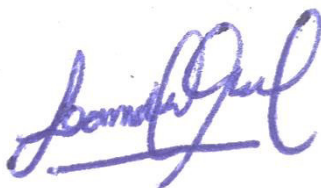
---

**BRYAN ANDRES ENRIQUEZ AUZ**

**Enriquez.bryan@epn.edu.ec**

**Bryan.enriquez@sgtechnologic.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por BRYAN ANDRES ENRIQUEZ AUZ, bajo mi supervisión.



---

**LEANDRO ANTONIO PAZMIÑO ORTIZ**

**Leandro.pazmino@epn.edu.ec**

## DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

A handwritten signature in black ink, appearing to be 'Bryan Andrés Enriquez Auz', written in a cursive style.

BRYAN ANDRÉS ENRIQUEZ AUZ

## **DEDICATORIA**

Este trabajo se lo dedico principalmente a mis padres que nunca cuestionaron mis acciones y siempre me apoyaron profesionalmente, que me alentaron todo el camino que he recorrido a través de mi formación profesional, les agradezco por cada abrazo que me dieron y cada palabra de aliento, por verme en las madrugadas aun con la luz prendida tratando de salir adelante, se lo dedico a mis hermanas Estefanía y Dayana que no dejaron de alentarme y apoyarme en mi trayecto y me lo dedico a mí mismo como prueba de constancia y sacrificio, como prueba de que los sueños son para ir tras ellos y que no hay mayor recompensa que haber llegado a la meta sin importar la forma del camino.

## **AGRADECIMIENTO**

Agradezco a Dios desde una posición muy personal por darme salud y sabiduría para culminar con mis proyectos y alcanzar mis metas, agradezco a mis padres por haber formado mi carácter y enseñarme a luchar hasta conseguir mis sueños, por sus enseñanzas silenciosas a través del ejemplo de lucha, constancia y superación, agradezco a mis amigos Katty, Erika, Cinty, Ivo, Andrés, Oswaldo por creer siempre en mí y apoyarme, agradezco a Gabby y José que fueron siempre una mano amiga y un apoyo fundamental durante los momentos más decisivos, agradezco a David y Nubia por extender su mano siempre que les fue posible, y gracias a todas las personas que no confiaron en mí y mucho más a las que jamás dudaron, todos ustedes me hicieron desear jamás rendirme por mucho que lo pensara.

Gracias a quien me hizo ver que soy capaz de comerme el mundo si me lo propongo y me mostró la vida desde su perspectiva, por darme las alas para volar y soltarme cuando fueron firmes para seguir haciéndolo.

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDO .....	V
RESUMEN.....	VI
ABSTRACT .....	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	8
1.1 Objetivo general .....	9
1.2 Objetivos específicos.....	9
1.3 Alcance.....	9
1.4 Marco Teórico.....	9
2 METODOLOGÍA.....	12
3 RESULTADOS .....	13
3.1 Identificación de los requerimientos para el diseño del prototipo .....	13
3.2 Selección del hardware y software acorde a los requerimientos establecidos	14
3.3 Diseñar el prototipo del sistema de monitoreo .....	17
3.4 Implementación el prototipo.....	27
3.5 Realizar pruebas de funcionamiento del prototipo .....	45
4 CONCLUSIONES.....	61
5 RECOMENDACIONES.....	62
6 REFERENCIAS Bibliográficas.....	62
7 ANEXOS.....	65
ANEXO I: Certificado de Originalidad .....	i
Anexo II.I Código QR de la implementación y pruebas de funcionamiento del prototipo	ii

## RESUMEN

La primera sección presenta la introducción del presente proyecto, se plantean los objetivos que se buscan alcanzar desarrollando a la vez, conceptos importantes que permiten el entendimiento adecuado del proyecto desde su implementación hasta su funcionamiento.

En la segunda sección se plantea la metodología empleada con la que se dará cumplimiento a cada uno de los objetivos, dando a detalle los procesos realizados en la implementación del sistema de monitoreo de red mediante el uso del *software NEMS Nagios*.

Los resultados obtenidos, se presentan en la tercera sección, haciendo énfasis en los procesos de selección de *software* y *hardware* partiendo desde los requerimientos necesarios para el prototipo, se muestran las configuraciones realizadas en el servidor de *NEMS* para monitorear los dispositivos de la red, destacando la correcta instalación del *software* de monitoreo en el dispositivo *Raspberry Pi*, además también se presentan las pruebas de funcionamiento del prototipo implementado a fin de demostrar el cumplimiento de lo planteado con los respectivos registros.

La cuarta sección, presenta las conclusiones en concordancia con los resultados que se obtienen del cumplimiento de cada uno de los objetivos planteados; las recomendaciones, se encuentran en la quinta sección del documento, un apartado importante pues contribuirá con el conocimiento necesario para quienes busquen replicar o mejorar el prototipo desarrollado.

Se encontrarán en las dos últimas secciones, las referencias bibliográficas que fueron consultadas para el desarrollo del proyecto, así como los anexos correspondientes según amerite el caso, ya sean documentos o enlaces hacia información relevante.

**PALABRAS CLAVE:** *NAGIOS, Raspberry Pi, monitoreo, hosts, NEMS, servidor.*

## **ABSTRACT**

*The first section presents the introduction of this project, the objectives to be achieved are presented, developing at the same time, important concepts that allow the proper understanding of the project from its implementation to its operation.*

*The second section presents the methodology used to achieve each of the objectives, detailing the processes carried out in the implementation of the network monitoring system through the use of NEMS Nagios software.*

*The results obtained are presented in the third section, emphasizing the software and hardware selection processes starting from the necessary requirements for the prototype, the configurations made in the NEMS server to start monitoring the network devices are shown, highlighting the correct installation of the monitoring software on the Raspberry Pi device, also the performance tests of the implemented prototype are presented in order to demonstrate compliance with the respective records.*

*The fourth section presents the conclusions in accordance with the results obtained from the fulfillment of each of the stated objectives; the recommendations are found in the fifth section of the document, an important section because it will contribute with the necessary knowledge for those who seek to replicate or improve the developed prototype.*

*The last two sections contain the bibliographical references consulted for the development of the project, as well as the corresponding annexes as appropriate, whether they are documents or links to relevant information.*

**KEYWORDS:** *NAGIOS, Raspberry Pi, monitoring, hosts, NEMS, server.*



# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

A través del desarrollo de este proyecto, se logró implementar un sistema de monitoreo de dispositivos en una red empleando *NEMS Linux*, un software que puede ser montado sobre diferentes plataformas, y muy utilizado por sus beneficios en cuanto a uso y facilidad de manejo.

Para desarrollar este proyecto, se utilizó el dispositivo *Raspberry Pi* modelo 3B + según lo recomendado por el fabricante de forma que se logró cubrir los requerimientos básicos y necesarios para el funcionamiento del monitor de red; este prototipo, es capaz de notificar sobre el estado de un dispositivo o *host* dentro de la red, siempre que esté siendo monitoreado por el prototipo; esto se lo realizó mediante el uso del protocolo *Simple Network Management Protocol (SNMP)*, que usa un sistema de tipo llamada y respuesta para determinar el estado de cada dispositivo en la red.

Desde *routers* y *switches* hasta *hosts* de tipo impresoras u ordenadores de escritorio, pueden ser monitoreados, siempre que éstos admitan el protocolo *SNMP*; así mismo se puede hacer uso de agentes como es el caso de (*Nagios Remote Plugin Executor*) NRPE para poder en dispositivos remotos, como lo indica su nombre, o locales, ejecutar scripts que envían de retorno la información para luego ser desplegada o interpretada por el sistema NAGIOS; es importante recalcar que para el uso de un agente como es NRPE y que este pueda ejecutar los scripts en el dispositivo remoto, estos *scripts* deberán existir en dicho dispositivo [1].

En el dispositivo *Raspberry Pi* se montó el sistema *NEMS Linux* específico para este, mismo que puede ser descargado desde la página oficial. Una vez montado el sistema en el dispositivo se procedió a realizar las respectivas configuraciones dentro del *software*, así como añadir los *hosts*, en este caso un Servidor de *Windows*, una impresora de la marca RICOH modelo MP C305 y un ordenador de escritorio con sistema *Linux*.

Se obtuvieron, con el prototipo en funcionamiento, alertas cuando un dispositivo o *host* dentro de la red se encuentra disponible o no, gracias a la función de llamada y respuesta que es propia del protocolo *SNMP* además de otras alertas como la cantidad de usuarios conectados al dispositivo, la carga del *CPU* o el uso de la memoria *RAM* de los dispositivos que son monitoreados, de forma que se obtuvo una visión mucho más clara de la red y del estado de los dispositivos y su funcionamiento, tanto individual como dentro de la misma, así como mejorar el uso de los recursos de TI al ser un trabajo automatizado, pues se reduce la carga laboral que se realizaba de forma manual al monitorear los dispositivos.

## 1.1 Objetivo general

Implementar un prototipo de sistema de monitoreo de red empleando una solución a nivel de *software* mediante el uso de *NEMS*.

## 1.2 Objetivos específicos

- Identificar los requerimientos para el diseño del prototipo.
- Seleccionar el *hardware* y *software* acorde a los requerimientos establecidos.
- Diseñar el prototipo del sistema de monitoreo.
- Implementar el prototipo.
- Realizar pruebas de funcionamiento del prototipo.

## 1.3 Alcance

Por medio del presente proyecto se busca implementar un prototipo de sistema para recopilar información que permita realizar el monitoreo de dispositivos dentro de una red.

Por tal motivo, tendrán la capacidad de realizar las siguientes acciones:

- Configuración de diferentes elementos dentro de la red para proporcionar información que permita tener un monitoreo del funcionamiento de estos.
- Recopilación información de gestión de los dispositivos dentro de la red.
- Despliegue de la información recopilada para permitir una fácil interpretación.

## 1.4 Marco Teórico

### *Raspberry Pi*

Considerado como un ordenador de bolsillo el *Raspberry Pi* es una opción muy amigable para la realización de proyectos *Do It Yourself (DIY)* siendo desde sus inicios en 2012 una herramienta de hardware muy popular, cuya popularidad y uso solo han ido incrementando con el paso del tiempo y el desarrollo de nuevas tecnologías. Una de sus características y quizá la que más ha contribuido con su popularidad, ha sido el hecho de ser de distribución libre y que el *software* que utiliza de igual manera se basa en distribuciones libres de GNU/Linux, *software* que se conoce como *Raspberry OS*, antes llamado Raspbian, por ser basado en una distribución de Debian [2].

Actualmente *Raspberry Pi* se encuentra en su versión 4B que es la última placa de desarrollo concebida hasta el momento, aunque aún se comercializan sus antecesoras puesto que su uso está orientado a diferentes proyectos que dependerán de más o menos características dependiendo de la funcionalidad del proyecto a implementar, las placas más comunes o las más usadas en la actualidad de *Raspberry Pi* son los modelos 2B, 3B, 3B+ y la última disponible por supuesto, la 4B [2].

### **Monitor de red**

Debido a la gran cantidad de dispositivos que actualmente están conectados a la red cada vez su administración se vuelve más complicada y de difícil ejecución, si bien no por la complejidad, más bien por el tiempo que demanda la realización del trabajo, lo que no solo supone recursos de tiempo sino también recursos de personal y demás herramientas necesarias para la labor, en vista de ello se ha visto la necesidad de implementar soluciones informáticas a nivel de *software* que permitan realizar el monitoreo de una red de forma un poco más automatizada, tener reportes de los eventos que suceden dentro de la red cuando nadie la está supervisando [3].

La tarea de monitoreo de una red se ha convertido en la actualidad en una necesidad para las pequeñas, medianas y grandes empresas por lo que los desarrolladores de *software* cada vez han invertido más en el despliegue de plataformas cada una con diferentes ventajas y prestaciones, y claro que *Linux* como *software* de libre distribución no iba a ser la excepción por lo que se encontrarán soluciones informáticas a esta problemática, realmente innovadoras y completas [3].

### **NEMS Nagios**

*NEMS Linux* ha llegado al mercado como una alternativa completa, en el sentido de que no es necesario realizar configuraciones complejas para poner en marcha el monitoreo de la red. Bastará con descargar una imagen preconfigurada y cargarla en un microrordenador como es el caso del dispositivo *Raspberry Pi* para poner en funcionamiento un prototipo que permita realizar el análisis y monitoreo de la red [4].

### **Protocolo SNMP**

*Simple Network Management Protocol (SNMP)*, perteneciente a la capa de aplicación basado en *IP* es un protocolo que realiza el intercambio de información entre un administrador de red como herramienta y cualquier otro dispositivo que permita la habilitación de dicho protocolo. Algunos consideran a este protocolo como un diseño ideal

para la supervisión de la infraestructura de una red, ya que permite la integración de varios dispositivos de distintas marcas, puesto que *SNMP* no es propietario.

Al hablar de *SNMP* es imprescindible mencionar también a la *Management Information Base (MIB)* o Base de Información para Gestión por sus siglas en inglés, que no es más que un documento de texto que contiene de forma jerárquica la información del dispositivo, relativa a los datos que se pueden recopilar de los mismos.

*MIB*, se ve integrado en todos los dispositivos compatibles con *SNMP* y dicho protocolo se encarga de realizar la respectiva conversión para que sea un formato utilizable por el servidor de monitoreo, cualquiera que este fuese [5].

También para *SNMP* se tiene los *plugins* que no son más que comandos que pueden ser escritos en *Python*, o en *powershell* como también en distintos lenguajes de programación, cumpliendo siempre con el objetivo de obtener del dispositivo la información requerida. [6]

### **Agentes de monitoreo Nagios NRPE y NSClient++**

El agente o también llamado módulo de *Nagios Remote Plugin Executor (NRPE)* permite realizar una forma de monitoreo denominado activo con la instalación de dicho agente en los diferentes recursos, donde se ejecutarán *plugins* o *scripts* en dispositivos remotos de forma que se obtengan datos de lecturas que serán enviados desde el cliente al servidor de monitoreo a fin de que dicho servidor recoja dichos datos, sean procesados y mostrados de la forma en la que se haya implementado el servicio de monitoreo al que se haga referencia [6].

Por otra parte, *NSClient++*, es un agente muy utilizado en el entorno de los sistemas *Windows* para el monitoreo de dispositivos que es además compatible tanto con el sistema *NEMS Nagios*, como con otras plataformas también pensadas para el monitoreo de dispositivos, de manera igual a la que opera el agente *NRPE*, *NSClient++* también utiliza *plugins* para la obtención de la información del dispositivo [6].

### **Protocolo Transport Layer Security (TLS)**

El protocolo de comunicación *TLS* denotado por sus siglas en inglés como Seguridad de la Capa de Transporte se constituye como el sucesor y la versión mejorada del protocolo *Security Sockets Layer (SSL)*, mismo que funciona de forma similar siendo protocolos criptográficos, usan el cifrado para proteger la transferencia de información y datos, al adquirir certificados de seguridad, comúnmente el *SSL* se puede hacer uso tanto del protocolo *TLS* como del *SSL*; y es además muy popular su uso en el manejo de correos electrónicos [7].

## 2 METODOLOGÍA

Como primer paso en el desarrollo de este prototipo, se realizó un análisis del escenario donde va a ser implementado y puesto en funcionamiento para las respectivas pruebas, de modo que fueron considerados aspectos técnicos como cantidad de dispositivos, candidatos a ser analizados con el prototipo, uso de *software* que sea amigable con el funcionamiento y que brinde las características necesarias para el correcto monitoreo de la red.

Ya que la implementación y pruebas se realizaron en una instalación relativamente pequeña y con 3 dispositivos, el prototipo a diseñar requirió de un dispositivo de características básicas sin que estas lleguen a igualarse a las de un ordenador de escritorio por lo cual se usó el dispositivo *Raspberry Pi* que se ajustó a nuestras necesidades y sobre éste se montó un *software* ya desarrollado que permita realizar el monitoreo de la red en la que se implementó el prototipo.

Con los requerimientos identificados se procedió a hacer un análisis para seleccionar el *hardware* que satisfaga estas necesidades y para la selección del *software* se realizó un análisis en base a parámetros adicionales a las características propias del dispositivo, considerando la facilidad de uso de dicho *software*. Se comenzó discriminando el manejo del *software* que se usó para el desarrollo del prototipo eligiendo uno que permita no solo tener facilidad al momento de la implementación, sino también que tenga un entorno de uso amigable siendo capaz de ser manipulado de forma intuitiva y analítica lo que permitió cumplir con los próximos objetivos planteados.

Seleccionado tanto el *software* como el *hardware*, éste último en función del anterior, se procedió a implementar el prototipo, comenzando por la instalación del *software* de monitoreo, que en este caso fue *NEMS Linux* en una microSD con la herramienta *Raspberry Pi Imager*, ya con el dispositivo preparado con el *software* y configurado con las credenciales de administración, se procede a añadir los *hosts* que serán monitoreados, pudiendo realizar esto desde la interfaz web con la que se cuenta gracias al uso de *NEMS Linux*.

Se estableció realizar el monitoreo de 3 tipos distintos de dispositivos a por medio del prototipo dentro de una *Local Area Network (LAN)* para determinar el estado de los mismos y a través de notificaciones que son entregadas mediante correo electrónico se pueda evaluar el comportamiento y funcionamiento del dispositivo que está siendo monitoreado. La implementación del prototipo fue el resultado de unir el *hardware* y el *software*

anteriormente seleccionados para lo cual se realizaron todas las configuraciones necesarias para poner en marcha el prototipo con el sistema de monitoreo que incluyó la instalación de *software*, así como también la configuración de parámetros adicionales y montaje del *hardware* para el prototipo.

Finalizada la implementación del prototipo de monitoreo con el sistema *NEMS Linux*, se procedió a realizar las pruebas de funcionamiento con la inclusión de los dispositivos a ser monitoreados, lo que permite validar el correcto desempeño del prototipo. Los resultados obtenidos de las pruebas que se realizaron se emplearon para subsanar errores descubiertos dentro de la etapa y periodo de pruebas del prototipo y de esta manera garantizar que el prototipo cumple con todos los requerimientos planteados previamente.

### **3 RESULTADOS**

#### **3.1 Identificación de los requerimientos para el diseño del prototipo**

En el campo de la industria, cualquiera que sea el rubro de ésta, cada vez es más evidente el uso de dispositivos que están de hecho conectados a la red y de forma implícita con la existencia de estos dispositivos, también está la necesidad de conocer y supervisar el funcionamiento de los mismos, tal actividad supone no solo de recursos económicos sino también de personal que la realice, por lo que un sistema de monitoreo para dichos dispositivos en la red implica una notable mejora en la eficiencia del trabajo [8].

Al ser este prototipo, un dispositivo que monitoree a otros dentro de la red, se debe pensar en aspectos como la eficiencia energética al hablar de consumo y tener el mayor rendimiento posible hablando de la estabilidad en el funcionamiento del prototipo por lo que se deberá elegir un dispositivo lo suficientemente eficiente para garantizar optimización de recursos en el consumo energético, pero al mismo tiempo asegurar el rendimiento para un funcionamiento adecuado.

Habiendo determinado el escenario donde se pensó implementar el prototipo y establecidos los 3 dispositivos que sirvieron de muestra para las pruebas realizadas se estableció que los requisitos para la implementación del prototipo requieren de un mini ordenador con las suficientes características para poder ejecutar de manera adecuada el *software* de monitoreo de *NEMS Linux*, sin llegar a sobredimensionar las características del ordenador, por lo que se pensó en el dispositivo *Raspberry Pi* en su versión 3 y su modelo B+ siendo que éste se ajusta de forma adecuada a lo necesario para utilizar de

forma adecuada el sistema de monitoreo, colocando al dispositivo *Raspberry Pi* en un punto intermedio considerando los aspectos de prestaciones y funcionalidades.

### 3.2 Selección del hardware y software acorde a los requerimientos establecidos

Se realizó la elección tanto del *hardware* como del *software* en función del cumplimiento de las necesidades del prototipo, comenzando por el *software*, se tiene que lograr la implementación de un sistema que permita realizar el monitoreo de dispositivos de la red de la forma más eficiente posible y que al mismo tiempo su uso sea completamente amigable con la mayoría de usuarios que harán uso del mismo.

Desde su implementación, hasta la interpretación de los datos obtenidos deben ser procesos relativamente sencillos en comparación al uso de otro tipo de *software* también compatible con el dispositivo *Raspberry Pi*, por lo que el software seleccionado fue *NEMS Nagios* para *Raspberry* acorde a la Tabla 3.1 donde se establecieron diferencias entre el sistema *NEMS Nagios* y el sistema *Observorium* en base a reseñas escritas por usuarios dedicados de manera profesional al monitoreo de redes [9].

**Tabla 3.1** Tabla de comparativa de puntuación para Nagios y Observorium [9]

	CALIFICACIÓN POR RESEÑAS ESCRITAS (PUNTUACIÓN)	
	NAGIOS CORE	OBSERVORIUM
<b>Probabilidad de recomendar</b>	7.6	2.1
<b>Probabilidad de renovar</b>	9.9	0
<b>Usabilidad</b>	4	0
<b>Calificación de Soporte</b>	7.7	5
<b>Facilidad de configuración</b>	8.5 (inicial)	7 (general)
<b>Notificaciones</b>	7	6
<b>Usuarios</b>	9.5	8.3
<b>Prueba gratuita del sistema</b>	10	10

Para la selección de hardware, ya que el sistema está pensado para implementarse en el dispositivo *Raspberry Pi* se consideraron tres versiones del dispositivo para ser comparadas y elegir la opción más recomendada en función del resultado de la comparativa de las distintas versiones del dispositivo teniendo así en consideración algunas de las características más importantes de cada versión como se muestra en la Tabla 3.2.

**Tabla 3.2** Comparativa entre las últimas versiones de Raspberry Pi [10]

	<b>RASPERRY Pi 3B</b>	<b>RASPERRY Pi 3B+</b>	<b>RASPERRY Pi 4B</b>
<b>SoC</b>	Procesador type ARMv8 model Cortex A53 /Arq 64 bit de 4 núcleos / Frec. 1.2 (GHz)	BCM2711 de 4 núcleos 1.5 (GHz) ARM model Cortex A72	BCM2837 de 4 núcleos 1.4 (GHz) ARM model Cortex A53
<b>Conectividad</b>	Wifi incorporado version 802.11 b/g/n Bluetooth v4.1 y BLE	WiFi incorporado versión 802.11AC Bluetooth v4.2 Ethernet de hasta 300 (Mbps)	WiFi version 802.11AC Bluetooth v5.0 Ethernet de hasta 1000 (Mbps)
<b>RAM</b>	1 (GB) LPDDR2	1 (GB) LPDDR2	1 (GB) LPDDR4
<b>USB</b>	Puertos USB v2.0 x4	Puertos USB v2.0 x4	x2 puertos USB v3.0 x2 puertos USB v2.0
<b>Salida de Video</b>	1 x Salida audio/vídeo HDMI	1 puerto HDMI v1.3 definition up to 1080p	x2 puertos micro-HDMI v2.0 hasta 4K Frec. 30 (Hz) o 4K a 60 (Hz) con un solo puerto
<b>Costo</b>	\$70	\$89	\$169

Se consideró además las recomendaciones por parte del desarrollador del sistema que será usado en la implementación del prototipo, para este caso se habla del software NEMS, mismo que se encuentra en su versión 1.5.2 (compilación 1); de la recomendación del fabricante se determinó, para la implementación del prototipo el uso de un dispositivo con ciertas características, tanto mínimas como recomendadas como se muestra en la Tabla 3.3.

**Tabla 3.3** Diferencias entre las versiones 2 y 3 del dispositivo Raspberry [11]

<b>DISEÑO</b>	
Raspberry Pi 3	Raspberry Pi 2
Placa PCB sin case	Placa PCB sin case
<b>CONECTIVIDAD</b>	
Módulo Wifi + Bluetooth	Necesita un USB para obtener WiFi
<b>HARDWARE</b>	
Procesador de 4 núcleos a 1.2 (GHz)	CPU de 900 (MHz)
<b>PLATAFORMA</b>	
Requiere cualquier navegador para mostrar el contenido dinámico	Requiere de cualquier motor de JavaScript para ejecutar el código
<b>ACTUACIÓN</b>	
Aumento de rendimiento muy grande	Menor rendimiento en comparación con el modelo Pi 3
<b>VELOCIDAD DE RELOJ</b>	
Chipset de video core IV a 400 (MHz)	Chipset de video core IV a 250 (MHz)



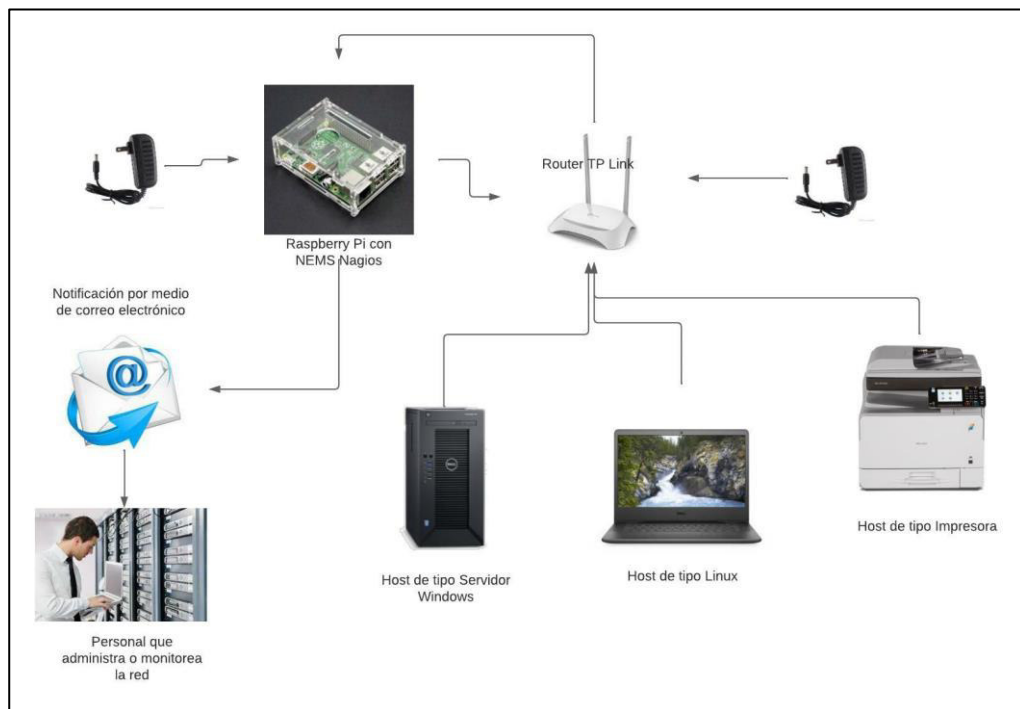
RAM	
Posee una velocidad de RAM de 450 (MHz)	Su velocidad de RAM es de 900 (MHz)

Finalmente, tras realizar las respectivas comparaciones tanto se *software* como de *hardware*, se decidió que el sistema de monitoreo *NEMS Nagios*, se montará sobre el dispositivo *Raspberry Pi* en su modelo 3B+, mismo que cuenta con las siguientes características:

- Procesador Broadcom *model* BCM2837B0, Cortex A53 (*Type* ARMv8) 64-bit SoC Frecuencia de hasta 1.4 (GHz).
- *1 (GB)* memoria tipo *LPDDR2 SDRAM*.
- *2.4 (GHz) / 5 (GHz) IEEE WiFi version 802.11.b/g/n/ac wireless LAN, Bluetooth v4.2.*
- *Gigabit Ethernet over USB v2.0 maximum throughput 300 (Mbps).*
- *Extended 40-pin General Purpose Input/Output (GPIO) header.*
- *High-Definition Multimedia Interface (HDMI).*
- x4 puertos USB v2.0.
- Puerto de cámara para el dispositivo *Raspberry Pi* camera.
- DSI puerto *Display Port* a *Raspberry Pi touchscreen display*.
- Puerto Micro SD para carga de sistema operativo.
- 5 (V)/2.5 (ADC).
- *Power-over-Ethernet (PoE) support [4] [12].*

Para el almacenamiento se tuvo una recomendación mínima de 16 (GB) como lo indican en la página de descarga oficial del sistema *NEMS Linux*, para almacenamiento mediante microSD, pero se utilizó una microSD de 32 (GB) para estar en un límite superior al mínimo recomendado. Como *hardware* adicional se consideró el uso de un *case* para la presentación final del prototipo [2] [4].

### 3.3 Diseñar el prototipo del sistema de monitoreo



**Figura 3.1** Esquema general del diseño del prototipo

Se tiene una red local con varios dispositivos en red de los cuales se eligieron tres de ellos para ser monitoreados con el sistema *NEMS Linux*; teniendo dispositivos de tipo Servidor de Windows, *Host de Linux* y un *host* de tipo impresora; se utilizará el protocolo *SNMP*, así como agentes propios del sistema *NAGIOS* para la obtención de los datos que permitirán verificar el estado y funcionamiento de dichos dispositivos en la red.

Siendo así, para cada tipo de dispositivo se utilizará plantillas y *plugins* específicos para cada tipo de dispositivo, usando *SNMP* para el caso de Windows Server y de dispositivo de tipo Switch y usando el agente *NRPE* para el caso del host de Linux, al mismo tiempo que se monitorean los dispositivos seleccionados, se encuentra monitoreando el propio dispositivo *Raspberry Pi* a fin de controlar la carga de *CPU* y evitar una sobrecarga en el rendimiento del dispositivo por lo que bastará colocar el dispositivo *Raspberry Pi* en una carcasa que lo proteja de posibles impactos durante su manipulación sin mayor modificación respecto al control de temperatura, siempre que el dispositivo no muestre alertas al respecto, en cuyo caso posterior al periodo de pruebas se subsanará dicho error de existir [1].

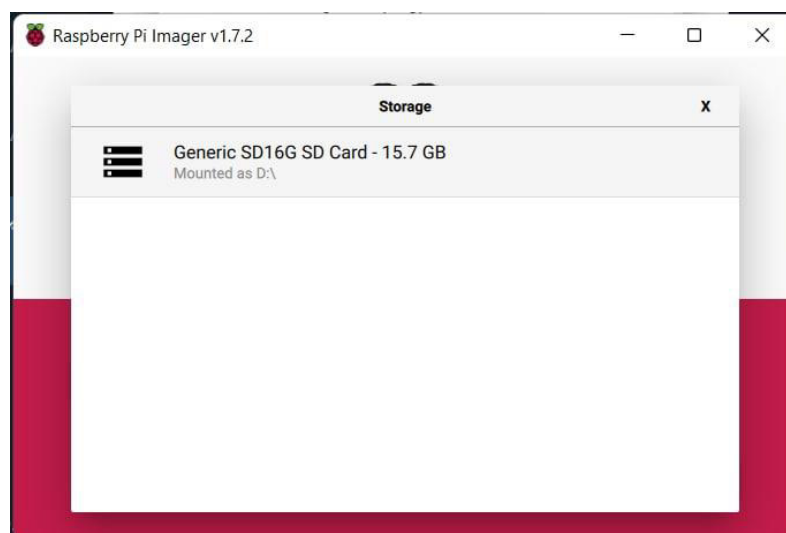
El primer paso para el diseño del prototipo, fue descargar la herramienta de *Raspberry Pi Imager*, con la que se monta el sistema *NEMS* en la tarjeta microSD que se usó en el

dispositivo *Raspberry Pi*; tanto la herramienta de montaje como la imagen del sistema que se usó, fueron descargadas desde sus páginas oficiales respectivamente [4].

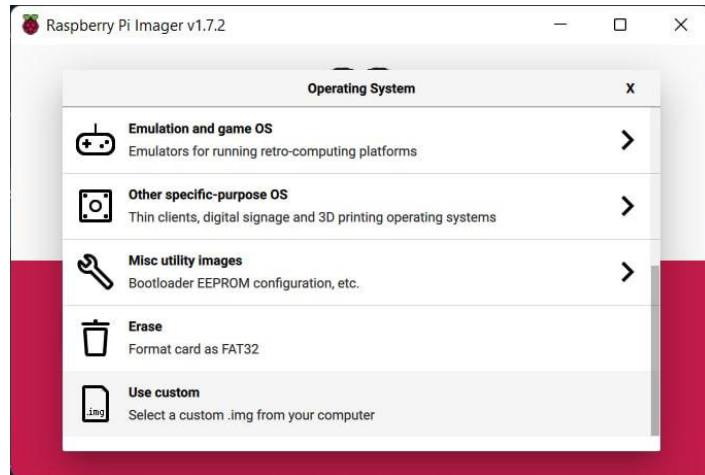
La instalación del sistema de *NEMS Linux* en la microSD fue realmente sencillo, gracias a la intuitiva interfaz de usuario que proporciona la herramienta de *Raspberry Pi Imager*, pues bastó con ejecutar el instalador de la herramienta y seleccionar tanto el sistema operativo que se va a instalar como el dispositivo de almacenamiento de destino tal como se indica en la Figura 3.2; para este caso se utilizó la opción de *custom*, como se muestra en la Figura 3.3

Para el caso del sistema operativo, puesto que *NEMS Linux* se lo instaló a partir de la imagen previamente descargada como se observa en la Figura 3.4. Una vez que se seleccionaron tanto el sistema como el dispositivo de destino se procede a pulsar el botón de escribir para que el sistema se monte en el dispositivo de almacenamiento seleccionado, no sin antes configurar las opciones de *SSH*, así como un usuario y contraseña con la que se iniciará sesión más adelante en la configuración inicial de la forma que se observa en la Figura 3.5 y posteriormente colocarlo en el dispositivo *Raspberry Pi* para poner en marcha el prototipo

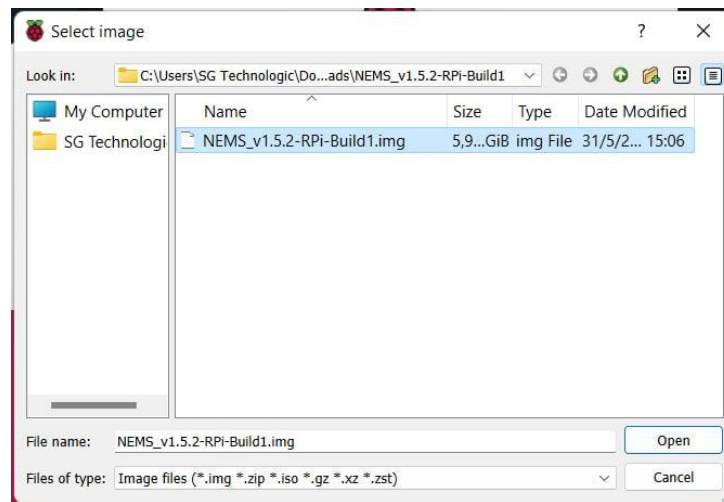
Una vez que la imagen haya sido grabada en el dispositivo de almacenamiento de destino, la herramienta de *Pi Imager*, se mostrará el mensaje que se observa en la Figura 3.6 siempre que la instalación hay asido exitosa.



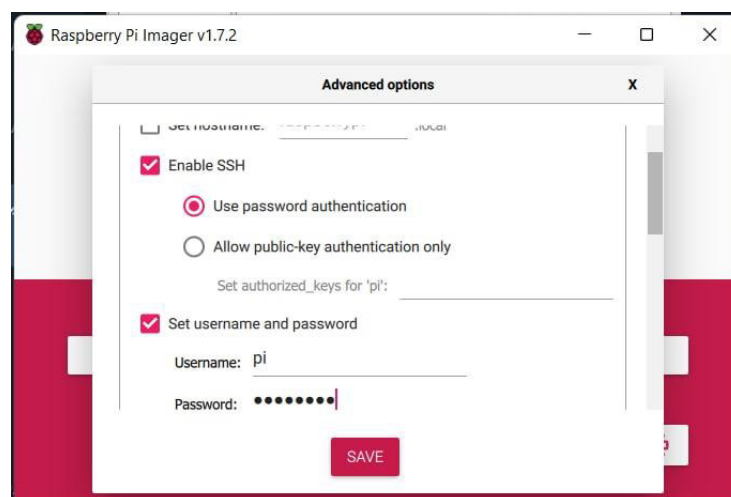
**Figura 3.2** Selección del dispositivo de destino para la instalación del sistema NEMS



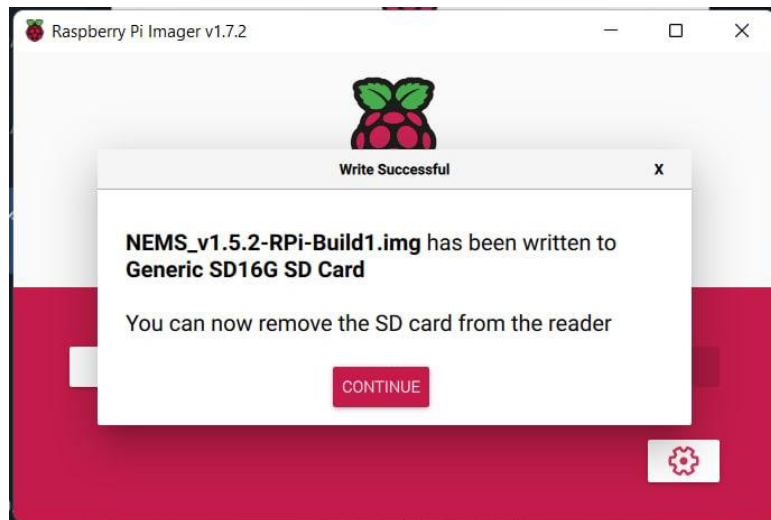
**Figura 3.3** Uso de la opción custom para la elección de imagen a instalar con Pi Imager



**Figura 3.4** Selección de archivo de imagen para montar en la microSD



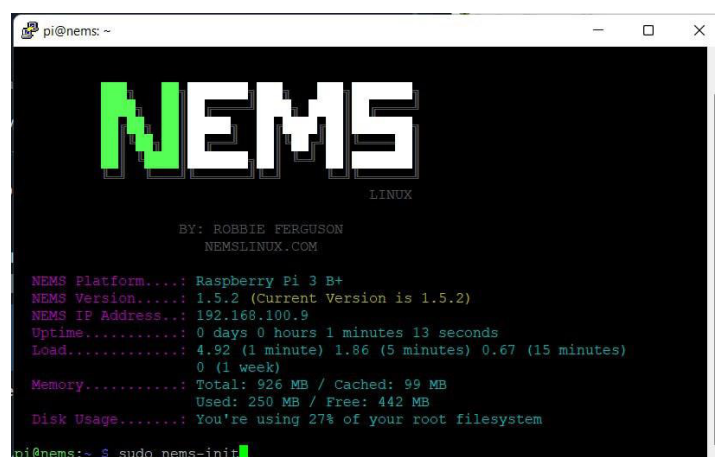
**Figura 3.5** Habilitación del servicio SSH para futuras conexiones



**Figura 3.6** Mensaje de instalación exitosa

Lo primero que se realizó una vez puesto en funcionamiento el prototipo, fue la inicialización mediante el comando “sudo nems-init” tal como se observa en la Figura 3.7 y configuración del sistema cargado en el dispositivo *Raspberry Pi* que previamente fue conectado a la red local dentro de la cual se realizará el monitoreo de los equipos, y del cual se debe conocer su dirección *IgP*, en este caso es la dirección 192.168.100.10, dirección que se obtuvo con la aplicación Fing como lo muestra la Figura 3.8; dicha inicialización se realizó mediante (*Command Line Interface*) *CLI* utilizando el protocolo de conexión (*Secure Shell*) *SSH* que permite una conexión segura y para lo cual se hizo uso de la herramienta PuTTY que se puede observar en la Figura 3.9 que se obtuvo de su página oficial.

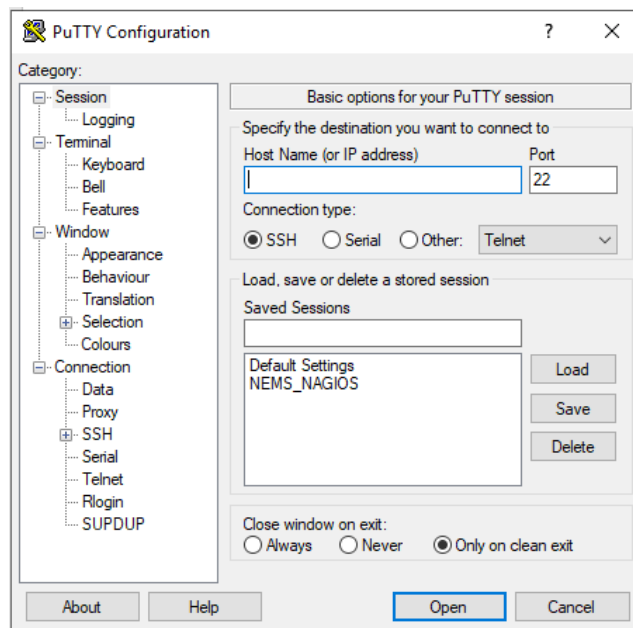
Las configuraciones iniciales que se realizaron fueron, la ubicación geográfica mostrada en la Figura 3.10.



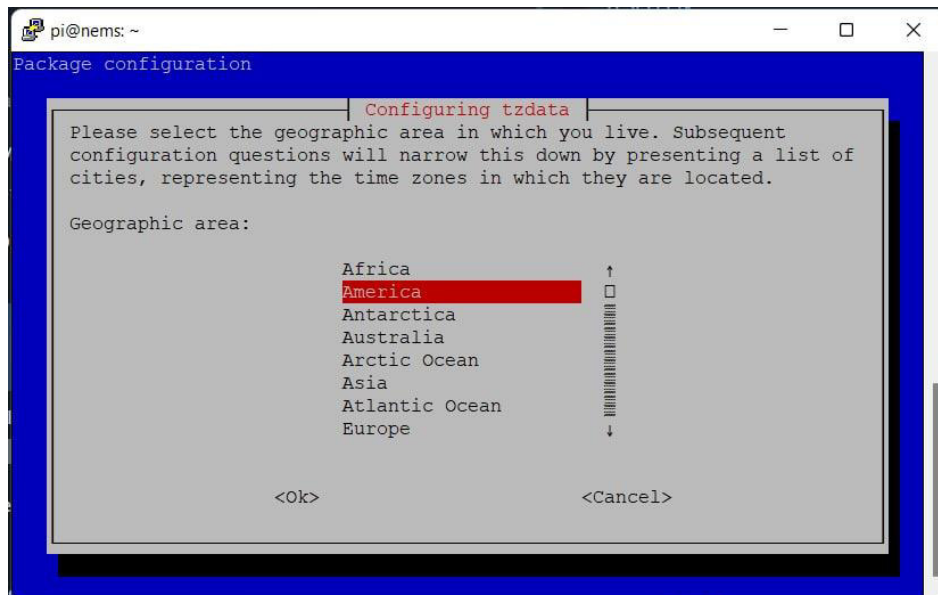
**Figura 3.7** Inicialización del sistema NEMS por medio de CLI



**Figura 3.8** Obtención de la dirección del dispositivo Raspberry Pi con la herramienta FING

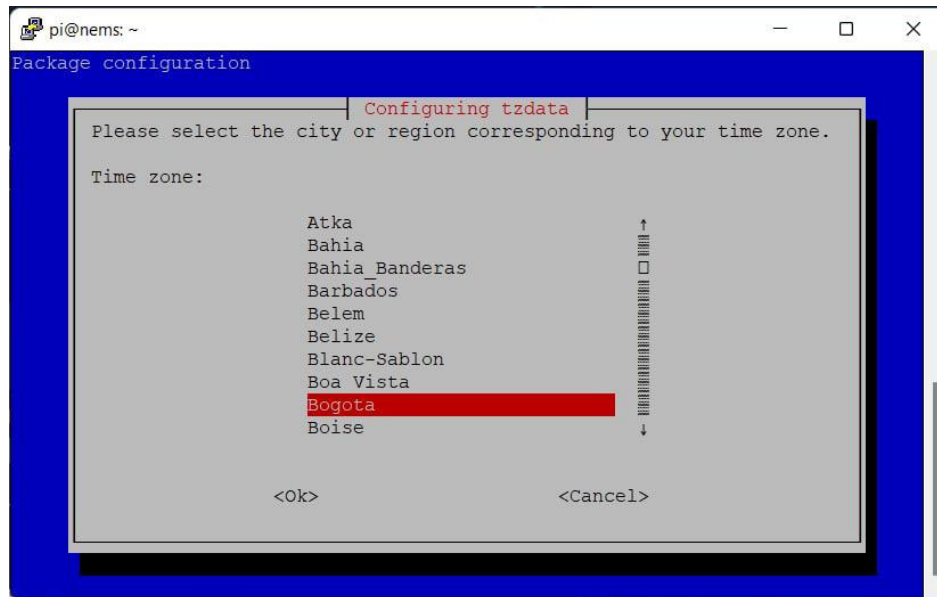


**Figura 3.9** Interfaz de la herramienta PuTTY para conexiones mediante SSH



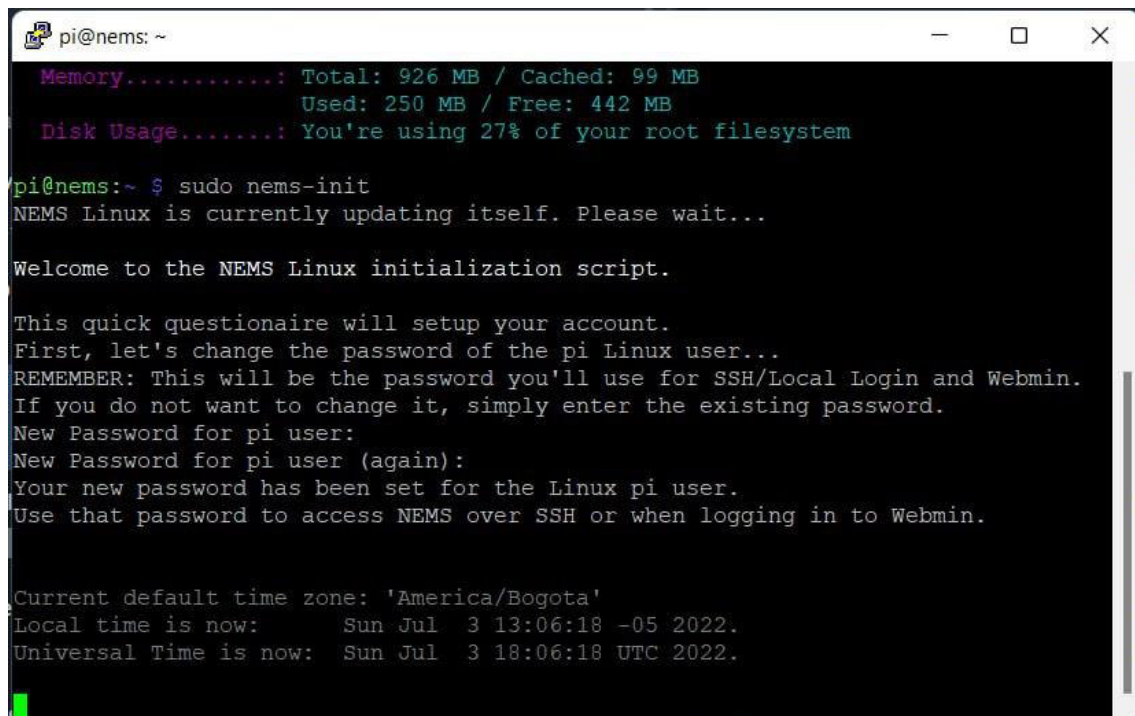
**Figura 3.10** Especificación de Zona geográfica para configuración inicial.

También se debe especificar una zona horaria, y puesto que no está disponible en el menú la región específica para Ecuador, se seleccionó Bogotá por pertenecer al mismo huso horario tal como se observa en la Figura 3.11.



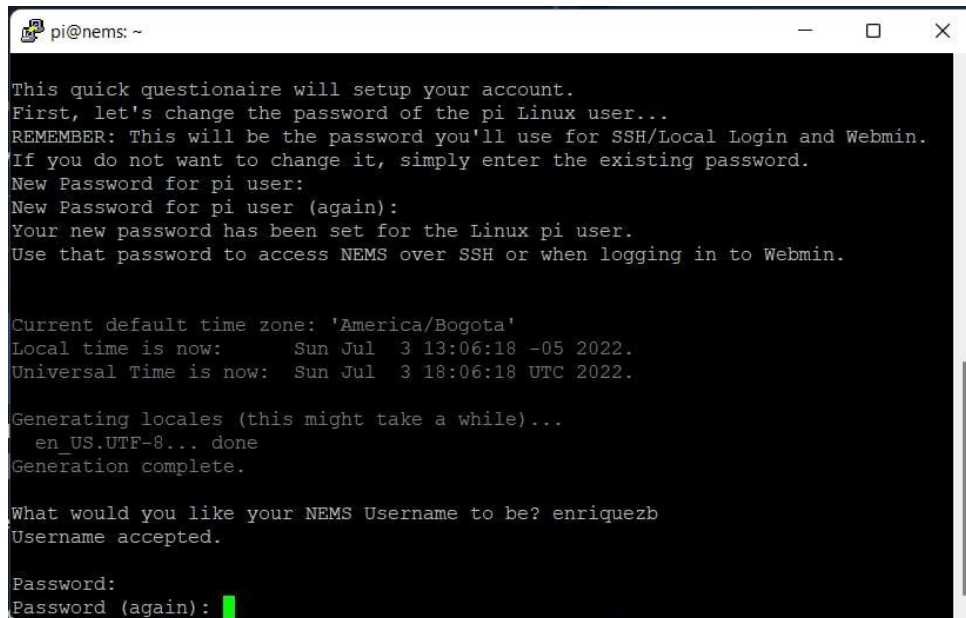
**Figura 3.11** establecimiento de zona horaria en la configuración inicial

Por medidas de seguridad, durante la configuración inicial se solicita nuevamente la contraseña para el usuario pi como se muestra en la Figura 3.12 que se configuró en los pasos anteriores durante la instalación del sistema NEMS.



**Figura 3.12** Establecimiento de credenciales de inicio de sesión para el usuario pi

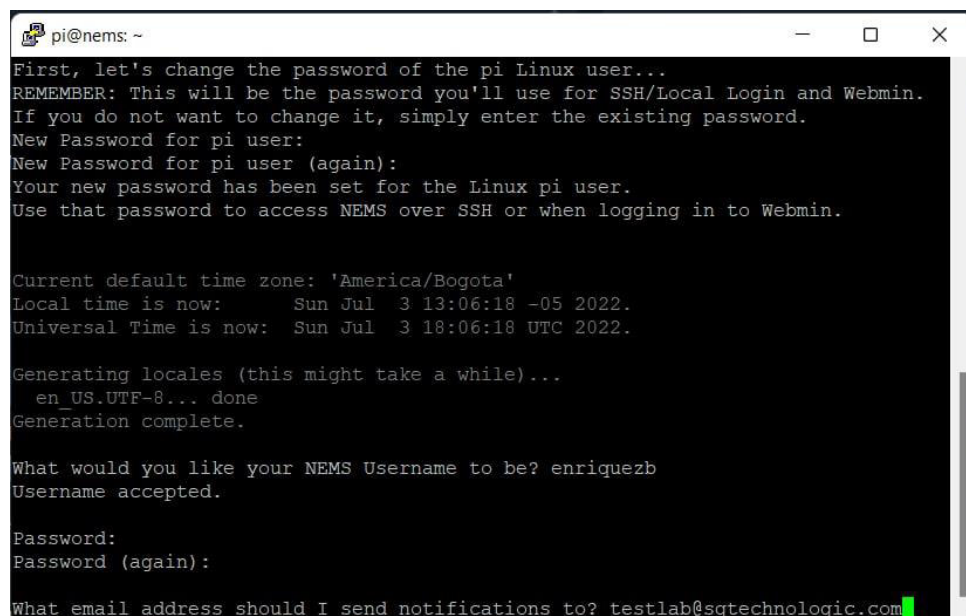
El siguiente paso será definir las credenciales del inicio de sesión para el administrador permitiendo elegir tanto el nombre de usuario como la contraseña de la forma que se observa en la Figura 3.13.



```
pi@nems: ~  
  
This quick questionnaire will setup your account.  
First, let's change the password of the pi Linux user...  
REMEMBER: This will be the password you'll use for SSH/Local Login and Webmin.  
If you do not want to change it, simply enter the existing password.  
New Password for pi user:  
New Password for pi user (again):  
Your new password has been set for the Linux pi user.  
Use that password to access NEMS over SSH or when logging in to Webmin.  
  
Current default time zone: 'America/Bogota'  
Local time is now:      Sun Jul  3 13:06:18 -05 2022.  
Universal Time is now: Sun Jul  3 18:06:18 UTC 2022.  
  
Generating locales (this might take a while)...  
  en_US.UTF-8... done  
Generation complete.  
  
What would you like your NEMS Username to be? enriquezb  
Username accepted.  
  
Password:  
Password (again):
```

**Figura 3.13** Establecimiento de credenciales de inicio de sesión para usuario administrador

Habiendo definido las credenciales de inicio de sesión, se solicita una dirección de correo electrónico válida, que será usada para las notificaciones que el sistema realiza al existir advertencias con los dispositivos monitoreados. Esta configuración la se observa en la Figura 3.14.



```
pi@nems: ~  
  
First, let's change the password of the pi Linux user...  
REMEMBER: This will be the password you'll use for SSH/Local Login and Webmin.  
If you do not want to change it, simply enter the existing password.  
New Password for pi user:  
New Password for pi user (again):  
Your new password has been set for the Linux pi user.  
Use that password to access NEMS over SSH or when logging in to Webmin.  
  
Current default time zone: 'America/Bogota'  
Local time is now:      Sun Jul  3 13:06:18 -05 2022.  
Universal Time is now: Sun Jul  3 18:06:18 UTC 2022.  
  
Generating locales (this might take a while)...  
  en_US.UTF-8... done  
Generation complete.  
  
What would you like your NEMS Username to be? enriquezb  
Username accepted.  
  
Password:  
Password (again):  
  
What email address should I send notifications to? testlab@sgtechnologic.com
```

**Figura 3.14** establecimiento de email para la recepción de notificaciones



Al finalizar la configuración de los parámetros anteriores el sistema arrojará un mensaje de notificación indicando que el sistema ha sido configurado de manera correcta como lo muestra la Figura 3.15 y está listo para su uso considerando para el próximo inicio de sesión las credenciales que fueron configuradas anteriormente para el usuario administrador como se observa en la Figura 3.16.

```
pi@nems: ~
Local time is now:    Sun Jul  3 13:06:18 -05 2022.
Universal Time is now: Sun Jul  3 18:06:18 UTC 2022.

Generating locales (this might take a while)...
 en_US.UTF-8... done
Generation complete.

What would you like your NEMS Username to be? enriquezb
Username accepted.

Password:
Password (again):

What email address should I send notifications to? testlab@sgtechnologic.com

Adding password for user enriquezb
Adding user 'enriquezb' ...
Adding new group 'enriquezb' (1002) ...
Adding new user 'enriquezb' (1001) with group 'enriquezb' ...
Creating home directory '/home/enriquezb' ...
Copying files from '/etc/skel' ...
Added user enriquezb.
Initializing new Nagios user
Updating contact: enriquezb

Generating unique SSL Certificates...
Generating unique SSH Certificates...
Creating SSH2 RSA key; this may take some time ...
2048 SHA256:Hslcr6rQGI52cGdsdg46223rcfrURj9HvFgllZyvHtc root@nems (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:sxLhT2WR6DXsUTrybbgM69SrfU195ZwTVQHL8eXkG3s root@nems (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:2adlycBzH3wbKR+2F2nDsRbLD/XFUmbIbzjA43+QlPK root@nems (ED25519)
rescue-ssh.target is a disabled or a static unit, not starting it.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@ WARNING: NEMS SERVER IDENTIFICATION HAS CHANGED! @
@ Next time you connect, you'll need to re-import! @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Done.
Restarting NEMS services...
Done.

You can now visit https://192.168.100.9/ to get started.

Enjoy NEMS Linux!

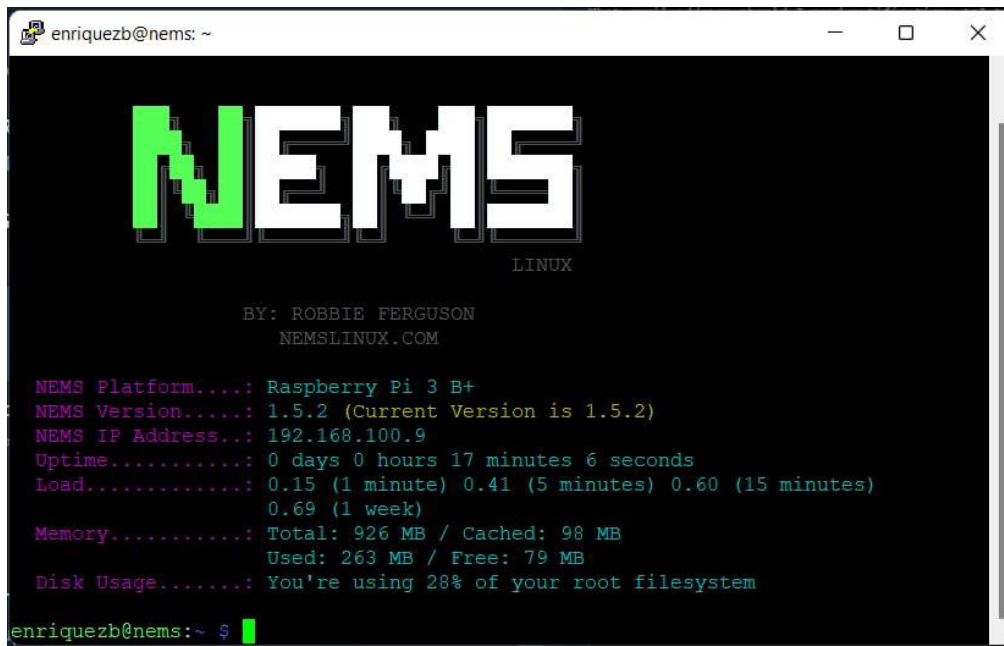
pi@nems: ~
```

**Figura 3.15** Finalización de configuración inicial del sistema NEMS

```
192.168.100.9 - PuTTY
login as: enriquezb
enriquezb@192.168.100.9's password:
```

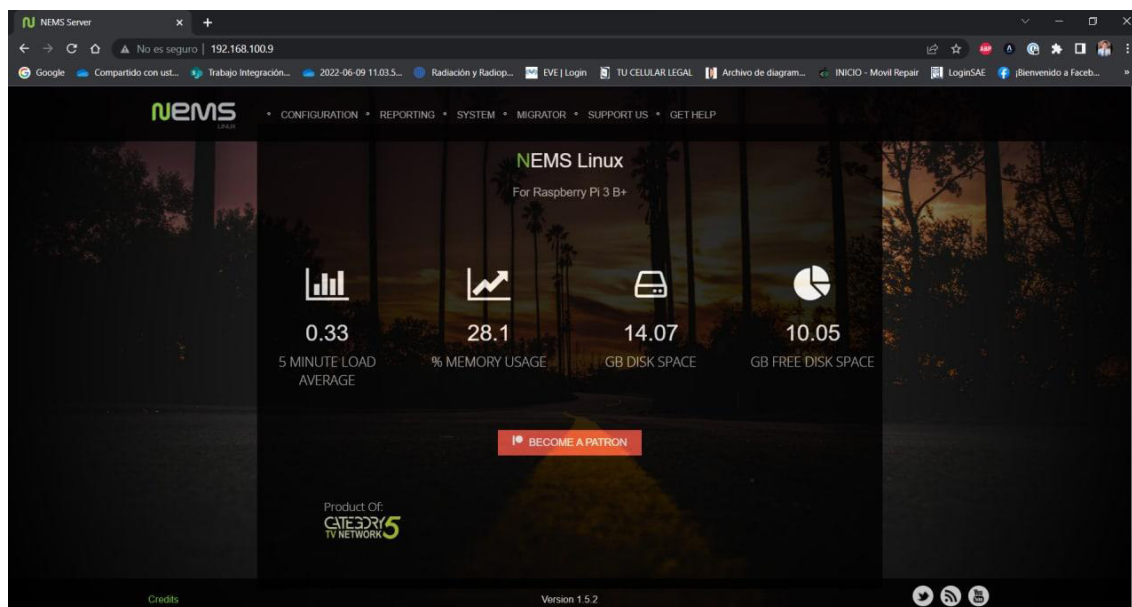
**Figura 3.16** Primer inicio de sesión con las nuevas credenciales de administrador

Si la instalación y configuración inicial del sistema *NEMS Linux* fue correcta incluyendo el uso de las nuevas credenciales de inicio de sesión, se puede observar nuevamente la pantalla principal de línea de comandos del sistema *NEMS Linux*, pero conectados con el nuevo usuario como se ve en la Figura 3.17.



**Figura 3.17** Ventana principal de CLI para el sistema NEMS

Tras haber verificado que toda la configuración inicial fue exitosa, se puede conectar al sistema de monitoreo mediante un navegador, digitando en la barra de direcciones la dirección *IP* del mismo que se obtuvo previamente y acceder a la interfaz web del sistema como se observa en la Figura 3.18.

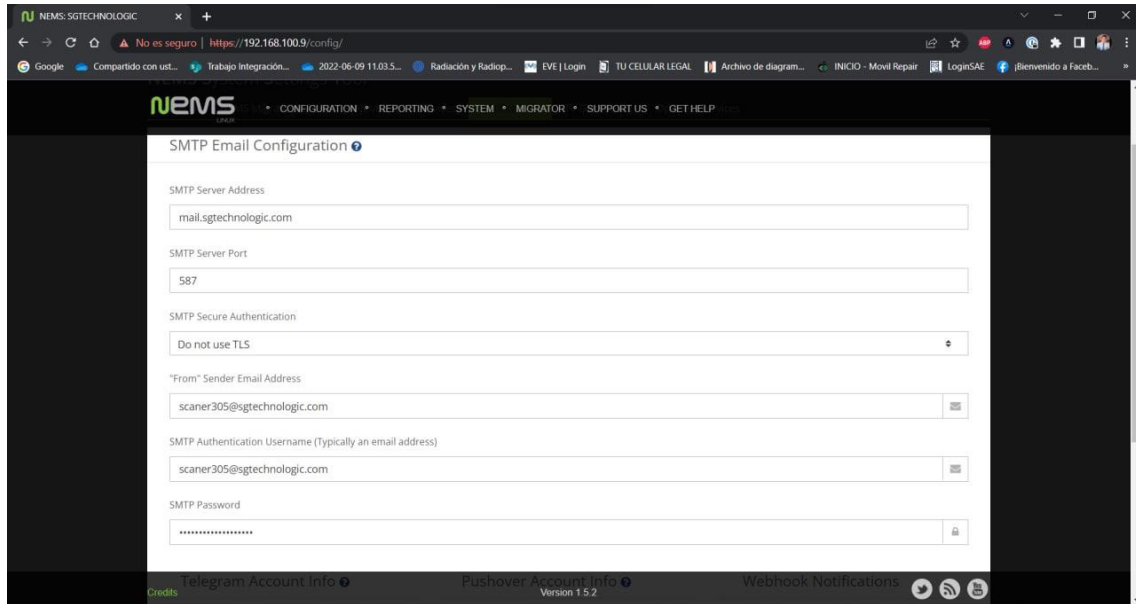


**Figura 3.18** Interfaz web del sistema NEMS

Para hacer uso de las notificaciones a través de *email*, es necesario especificar en la configuración del sistema *NEMS Linux* los parámetros del protocolo SMTP como el nombre del servidor, el tipo de seguridad que se va a utilizar, el puerto de comunicación del mismo

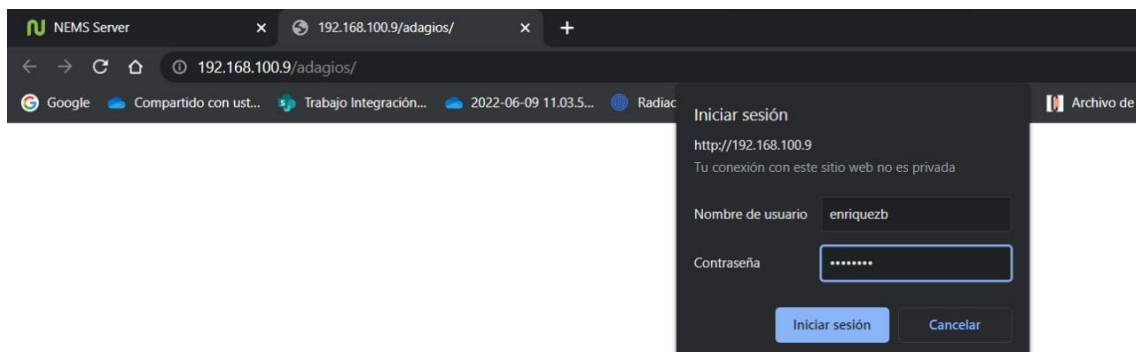
y las credenciales de un usuario válido que pertenezca al dominio, estos parámetros se especifican en la Figura 3.19.

Es importante mencionar que por el momento el *core* de *NAGIOS* solo admite el protocolo *TLS* sin seguridad por lo que el puerto a utilizar es el 587 en lugar del puerto 465 [7].



**Figura 3.19** Configuración del servicio SMTP para las notificaciones por email

El siguiente paso para el sistema de monitoreo de *NEMS Linux* es ir añadiendo los *hosts* que serán monitoreados, así como los servicios en cada uno de los *hosts* que se añadan; esta acción debe ser realizada desde el menú de NConf del sistema *NAGIOS* para lo cual se solicitará al usuario que inicie sesión con sus credenciales como se observa en la Figura 3.20, una vez iniciada la sesión de forma correcta se podrá tener acceso al menú de *hosts* donde se permite añadir, eliminar o modificar cada uno de los *hosts* según sea lo requerido. La ventana de la sección de *hosts* se la observa como se muestra en la Figura 3.21.



**Figura 3.20** Inicio de sesión en el *dashboard* de configuración del sistema NEMS

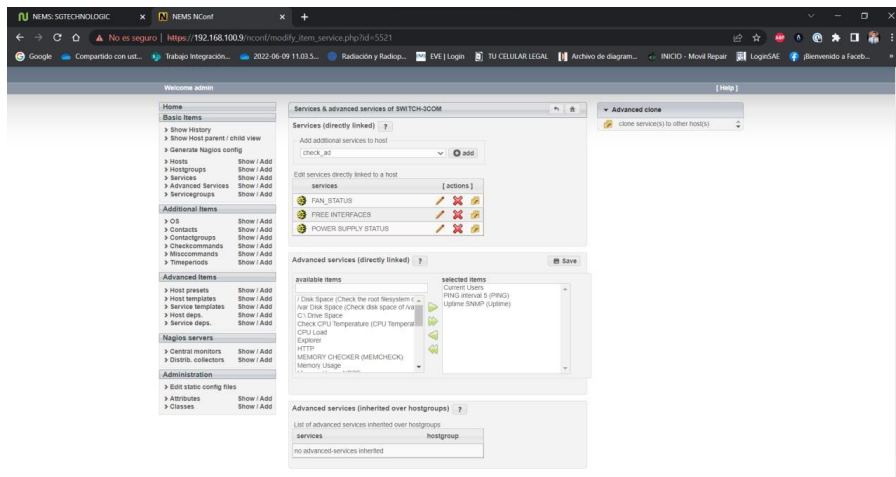


Figura 3.21 Ventana principal de la sección hosts del sistema NEMS

### 3.4 Implementación el prototipo

El prototipo diseñado se implementó en la red local de la empresa *Technologic Services Group* en el área de servicio técnico, donde se disponen de los *hosts* a monitorearse:

- *Host* de tipo servidor con *Windows Server* versión 2019.
- *Host* de tipo ordenador con sistema operativo *Linux Ubuntu v20.0*.
- *Host* de tipo impresora de la marca *RICOH* modelo *MP C305 SPF*.

Se empezó la implementación del prototipo con el armado y ensamble del dispositivo en su carcasa para protección de agentes externos o ambientales durante el periodo que se encuentre en funcionamiento dentro de las instalaciones.

Se dispone de una carcasa modular de plástico, el dispositivo *Raspberry Pi*, y la *microSD* con el *software* de monitoreo previamente cargado como se muestra en la Figura 3.22.



Figura 3.22 Prototipo desensamblado para su implementación

Una vez que el dispositivo fue ensamblado de forma correcta se procedió a conectarlo a la red eléctrica a través de su adaptador de alimentación y a la red local a través de su puerto *ethernet* con un *patch cord* de la forma en que se muestra en la Figura 3.23.

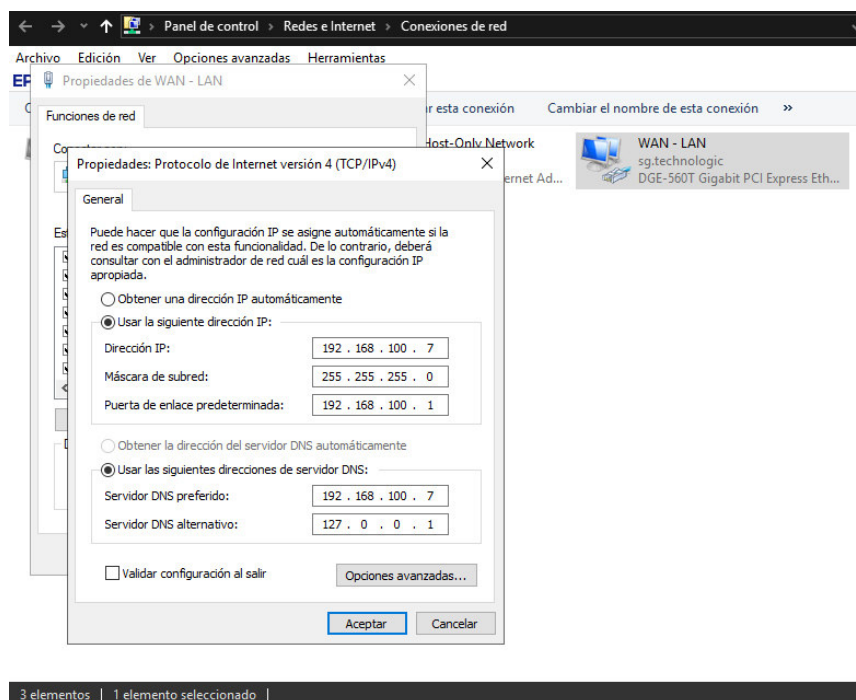


**Figura 3.23** Dispositivo *Raspberry Pi* conectado a la corriente y a la red

Para iniciar con el monitoreo de los diferentes dispositivos, se precisó realizar ciertas configuraciones en cada uno para poder realizar la actividad de monitoreo; a continuación, se detallan las configuraciones realizadas en cada *host* para cada caso puntual.

### Host de *Windows* con sistema operativo *windows server 2019*

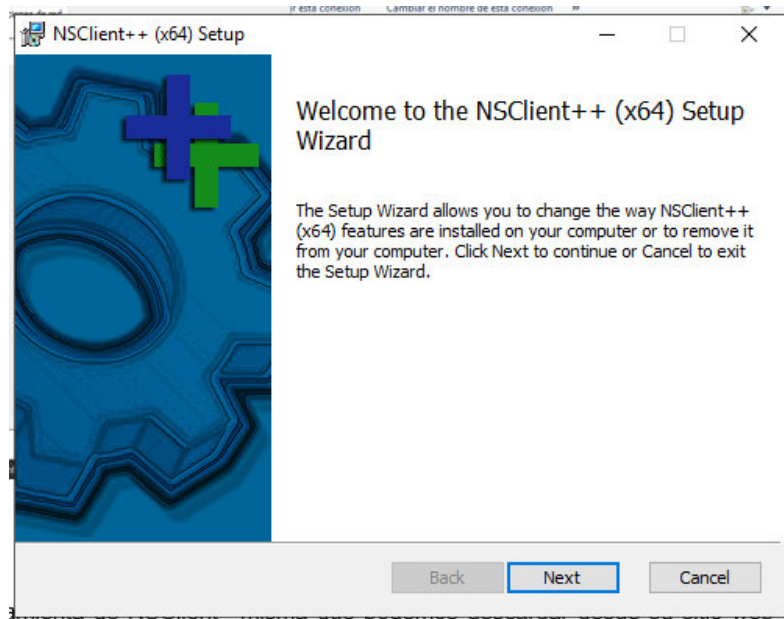
Las configuraciones que se realizaron en el cliente de *Windows*, fueron verificar que la dirección *IP* asignada a este *host* sea estática, lo que se puede verificar en las propiedades del adaptador de red, a través del panel de control en la ventana que se muestra en la Figura 3.24.



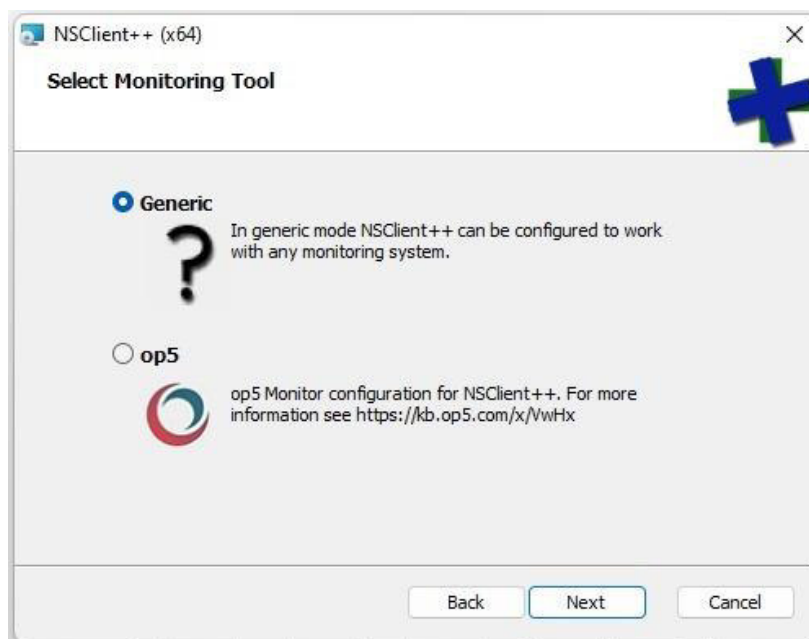
**Figura 3.24** Verificación de la asignación de *IP* fija para el host de *Windows*

Una vez verificada la asignación de una *IP* fija al *host* se procede a instalar la herramienta de *NSClient++* misma que se puede descargar desde su sitio web oficial, para proceder con la instalación de dicho agente en este *host*, para lo cual se debió seguir los siguientes pasos [13].

Al ejecutar el instalador de *NSClient++* se presenta la pantalla de inicio que se observa en la Figura 3.25, sobre la cual se debe presionar sobre el botón de *Next* que avanzará a la pantalla de la Figura 3.26, que constituye el primer paso de la instalación de tipo genérica.

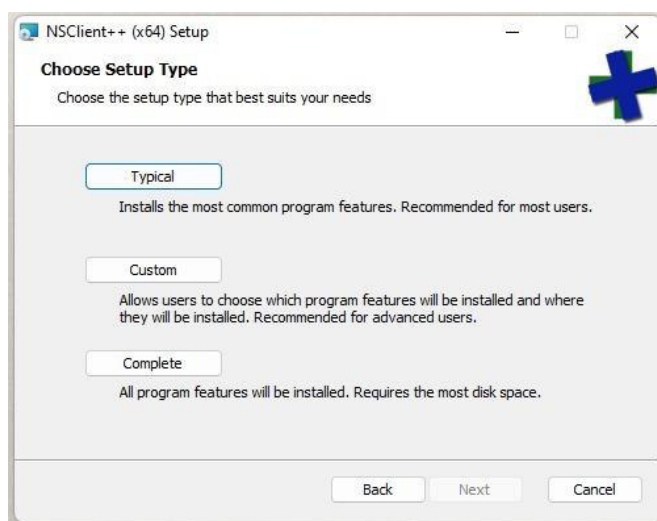


**Figura 3.25** Inicio del Instalador de NSClient++



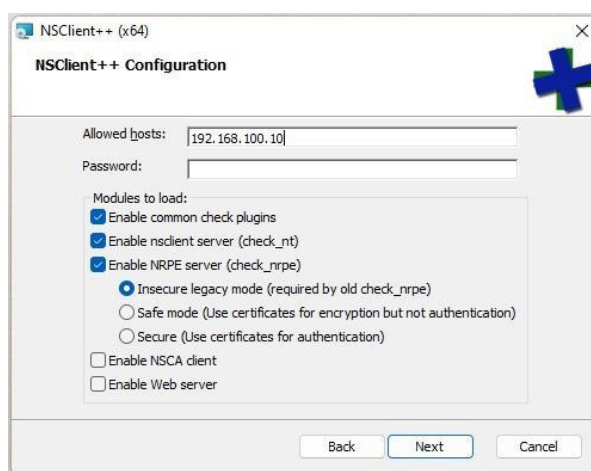
**Figura 3.26** Paso 1 de la instalación de NSClient++

Tras pulsar en el botón de *Next* se deberá elegir el tipo de instalación que se desea realizar, para este caso, se debe seleccionar la opción de “*Complete*” que se observa en la Figura 3.27, realizando así el paso número 2.



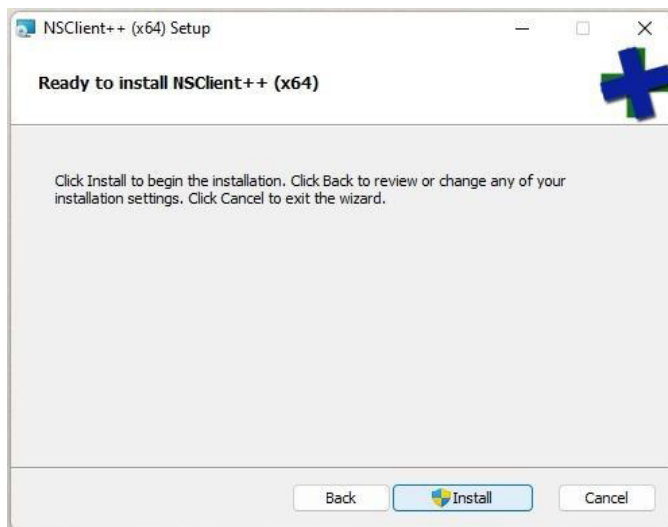
**Figura 3.27** Paso 2 de la instalación de *NSClient++*

El paso número 3 fue seleccionar en la ventana que se aprecia en la Figura 3.28, la dirección *IP* del servidor de monitoreo, para este caso es la dirección 192.168.100.10 que previamente fue obtenida y asignada al servidor mediante la reserva de dicha dirección en el *router* principal, usando la dirección *MAC* del dispositivo, puesto que al ser un sistema de monitoreo no es conveniente que su *IP* cambie, siendo esto una posible causa de desconexión entre los dispositivos y el servidor de monitoreo; también se deberán marcar las opciones de acuerdo a como se muestra en la figura dejando el campo de contraseña en blanco.



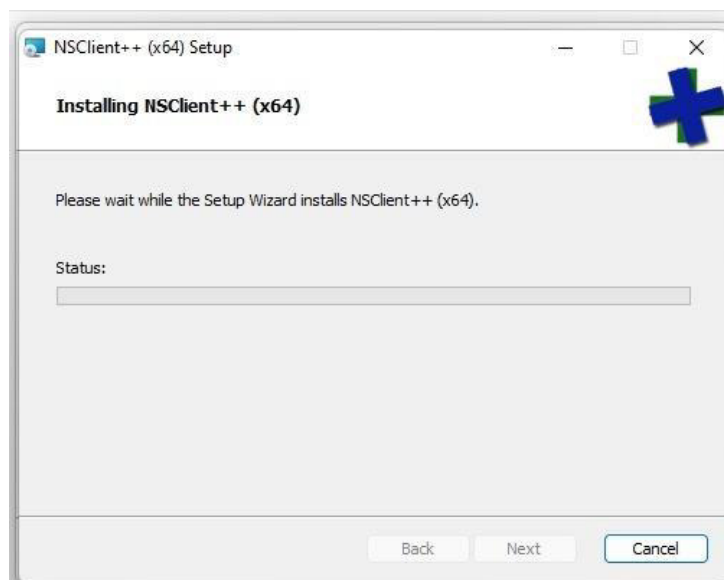
**Figura 3.28** Paso 3 de la instalación de *NSClient++*

En el paso número 4 de la instalación simplemente se debe pulsar el botón de “Next” para avanzar a la siguiente pantalla de configuración como muestra la Figura 3.29.



**Figura 3.29** Paso 4 de la instalación de NSClient++

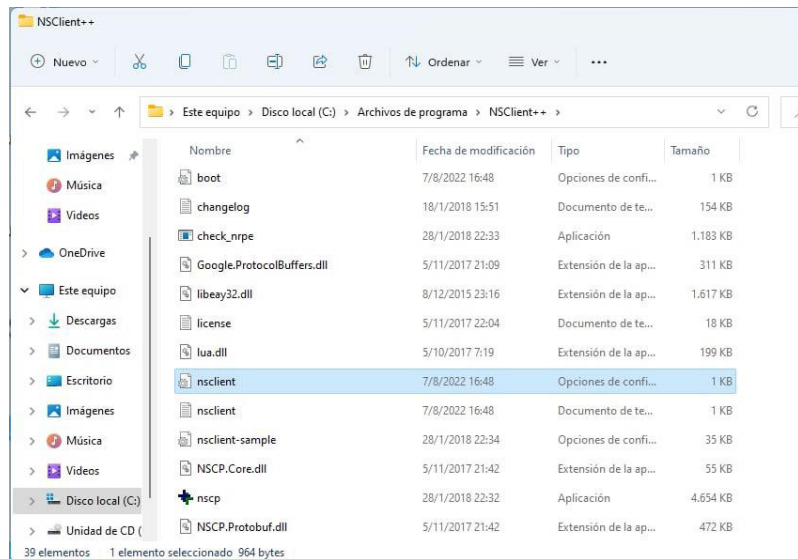
Tras pulsar el botón “Install” de la ventana anterior bastará con esperar que finalice el proceso de instalación como se muestra en la Figura 3.30.



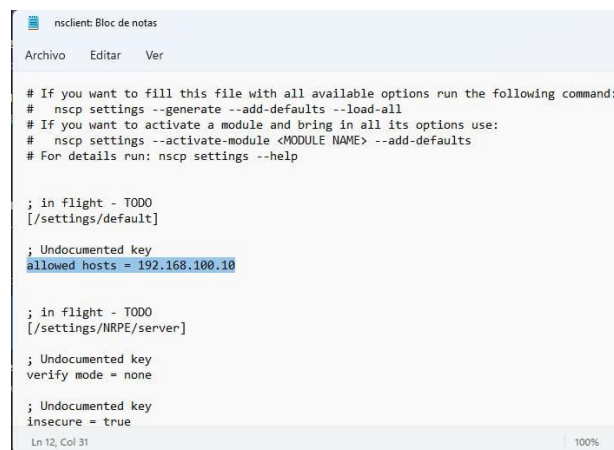
**Figura 3.30** Finalización del proceso de instalación de NSClient++

Finalizada la instalación del agente *NSClient++* es importante verificar la asignación y los permisos que tiene configurado el servicio para la dirección *IP* del servidor *NEMS*, para lo cual se dirige a la ruta que se observa en la Figura 3.31 en busca del archivo de configuración “*nsclient*”, el mismo que se debe abrir y encontrar la línea que contiene la instrucción de “*allowed hosts = XX.XX.XX.XX*” y verificar que corresponda a la dirección *IP* del servidor *NEMS* tal como se muestra en la Figura 3.32 [14].



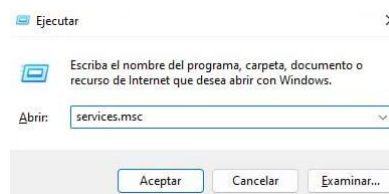


**Figura 3.31** Ubicación del archivo de configuración de NSClient++

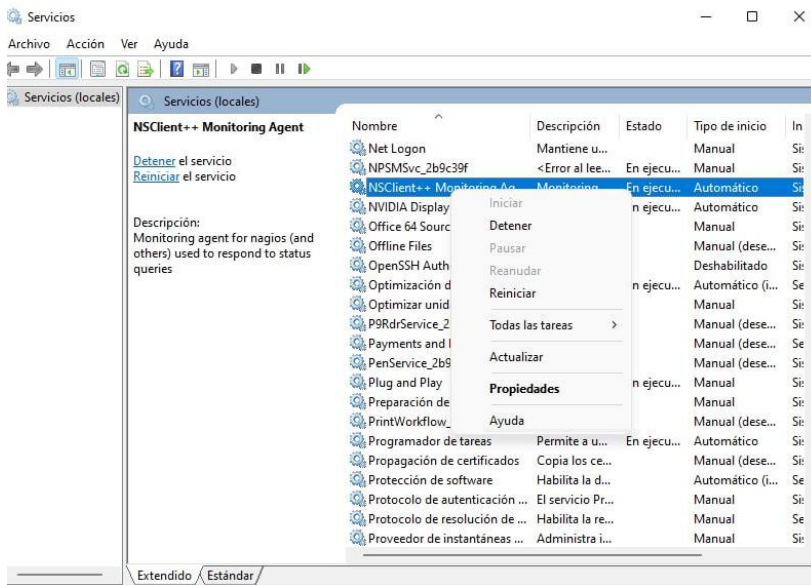


**Figura 3.32** Verificación de permiso para la IP del servidor NEMS en NSClient++

Si la verificación anterior es correcta, solo basta reiniciar el servicio de NSClient++ para lo cual por medio del comando Win+R se abre la ventana de ejecutar para escribir “services.msc” como se observa en la Figura 3.33, para acceder a la lista de servicios en la cual se debe encontrar el servicio de NSClient++ Monitoring y reiniciarlo de la manera que se muestra en la Figura 3.34.



**Figura 3.33** Acceso a la lista de servicios desde el comando ejecutar de Windows



**Figura 3.34** Reinicio del servicio de NSClient++ para actualizar los cambios  
**Host de tipo Linux con sistema operativo Ubuntu v20.04**

Para el *host* de Linux se debe instalar también un agente de monitoreo, en este caso usando *NRPE* y *plugins*, por lo que se debe instalarlos mediante *CLI* con el comando que se muestra en la Figura 3.35 [15].

```
bryan@bryan-VirtualBox: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bryan@bryan-VirtualBox:~$ sudo apt install nagios-nrpe-server nagios-plugins
```

**Figura 3.35** Instalación del cliente *NRPE* mediante *CLI*

Se puede observar el proceso de instalación en porcentaje mientras este se realiza tal como se puede apreciar en la Figura 3.36.

```
bryan@bryan-VirtualBox: ~
libdbi1 libldb1 libmysqlclient20 libnet-snmp-perl libpq5 libsmbclient
libsnmp30 libtirpc1 libwbclient0 monitoring-plugins monitoring-plugins-basic
monitoring-plugins-common monitoring-plugins-standard mysql-common
python-crypto python-ldb python-samba python-tdb rpcbind samba-common
samba-common-bin samba-libs smbclient snmp
Suggested packages:
libcrypt-des-perl icinga | icinga | nagios3 nagios-plugins-contrib fping
postfix | sendmail-bin | exim4-daemon-heavy | exim4-daemon-light qstat
python-crypto-dbg python-crypto-doc heimdal-clients cifs-utils
The following NEW packages will be installed:
libdbi1 libmysqlclient20 libnet-snmp-perl libpq5 libtirpc1
monitoring-plugins monitoring-plugins-basic monitoring-plugins-common
monitoring-plugins-standard mysql-common nagios-nrpe-server nagios-plugins
python-crypto python-ldb python-samba python-tdb rpcbind samba-common
samba-common-bin smbclient snmp
The following packages will be upgraded:
libdbi1 libsmbclient libsnmp30 libwbclient0 samba-libs
5 upgraded, 21 newly installed, 0 to remove and 183 not upgraded.
Need to get 10,1 MB of archives.
After this operation, 22,1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ec.archive.ubuntu.com/ubuntu xenial-updates/main amd64 samba-libs a
md64 2:4.3.11+dfsg-0ubuntu0.16.04.34 [5.179 kB]
13% [1 samba-libs 1.690 kB/5.179 kB 33%]
```

**Figura 3.36** Proceso de instalación de *plugins* de *NRPE* mediante *CLI*

Tras el proceso de instalación de los *plugins* para *NRPE* es necesario configurar también para este *host* la dirección *IP* del servidor por lo que se debe editar el archivo de configuración de *NRPE* con el comando que se muestra en la Figura 3.37, se deberá encontrar dentro del archivo la línea que contenga la sentencia “*allowed\_hosts=xx.xx.xx.xx*” y verificar que la dirección *IP* configurada, corresponda a la del servidor de *NEMS* como se puede observar en la Figura 3.38 [15].

```
Creating config file /etc/nagios-plugins/config/snmp.cfg with new version
Setting up monitoring-plugins (2.1.2-2ubuntu2) ...
Setting up nagios-plugins (2.1.2-2ubuntu2) ...
Processing triggers for libc-bin (2.23-0ubuntu11.2) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
bryan@bryan-VirtualBox:~$ sudo nano /etc/nagios/nrpe.cfg
```

**Figura 3.37** Comando para editar el archivo de configuración del agente *NRPE*

```
bryan@bryan-VirtualBox: ~
GNU nano 2.5.3      File: /etc/nagios/nrpe.cfg      Modified
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1, 192.168.100.10
#
# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow clients
# to specify arguments to commands that are executed. This option only works
# if the daemon was configured with the --enable-command-args configure script
# option.
```

**Figura 3.38** Concesión de permiso para la *IP* del servidor *NEMS*

De la misma manera que para el *host* de *Windows*, luego de realizar la verificación de las configuraciones en los clientes, bastará con realizar un reinicio del servicio de *NRPE* para dar por terminada la configuración del *host* de *Linux* como se muestra en la Figura 3.39.

```
Processing triggers for libc-bin (2.23-0ubuntu11.2) ...
Processing triggers for ureadahead (0.100.0-19.1) ...
Processing triggers for systemd (229-4ubuntu21.28) ...
bryan@bryan-VirtualBox:~$ sudo nano /etc/nagios/nrpe.cfg
bryan@bryan-VirtualBox:~$
bryan@bryan-VirtualBox:~$ sudo systemctl restart nagios-nrpe-server
```

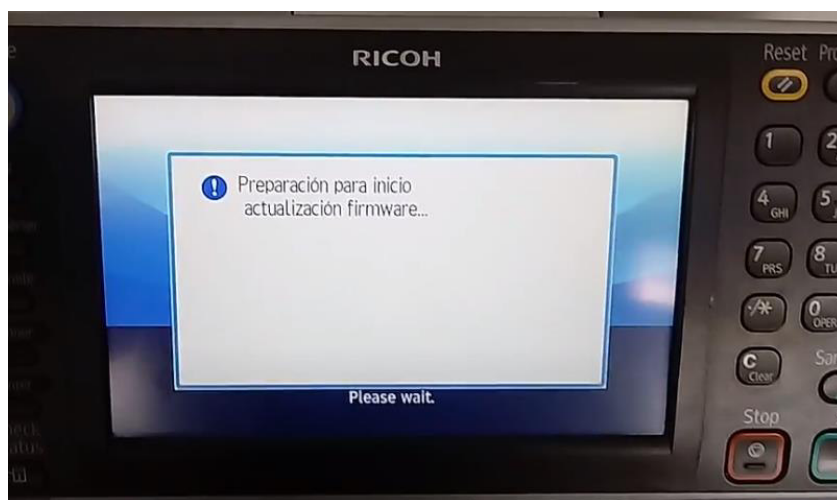
**Figura 3.39** Reinicio del servicio de *NRPE* para actualizar cambios

### **Host de tipo impresora marca Ricoh MP C305 spf**

Para monitorear con el prototipo un *host* de tipo impresora de la marca RICOH específicamente, se debe considerar un aspecto importante, que es contar con el software más actualizado por lo que en este caso el primer paso fue hacer un *upgrade* o actualización mediante el uso de una tarjeta de memoria como la que se presenta en la Figura 3.40, por ser de uso exclusivo para dispositivos de la marca RICOH.

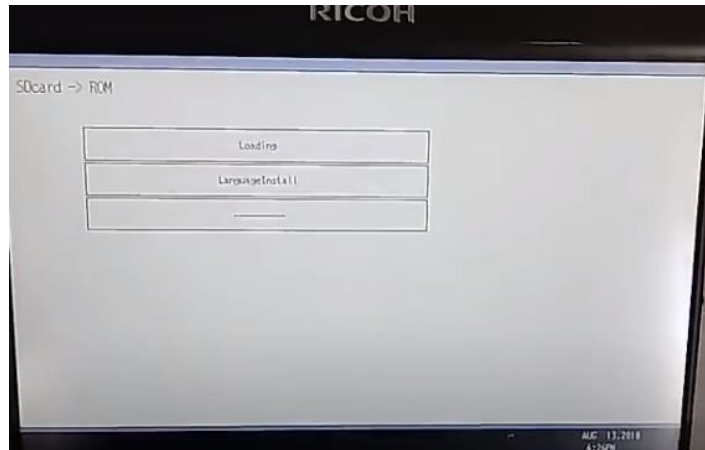


**Figura 3.40** Tarjeta de memoria de uso exclusivo para dispositivos de la marca RICOH. Apenas se introduce la tarjeta de memoria en la ranura de servicio del equipo (estando apagado), al encenderlo comenzará de manera automática la actualización del sistema, mostrando un aviso en la pantalla del dispositivo como se muestra en la Figura 3.41.

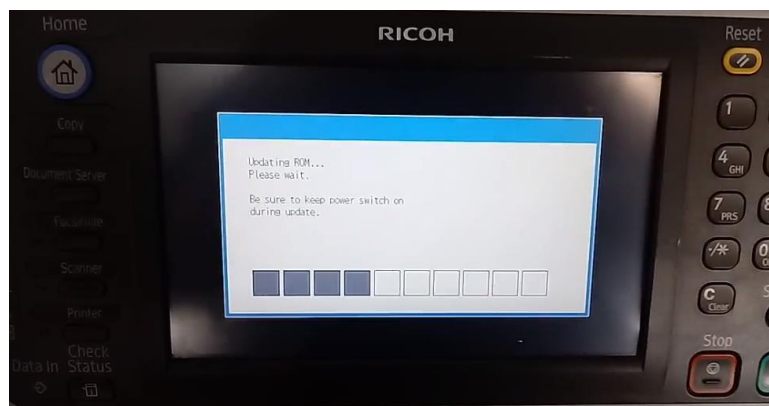


**Figura 3.41** Inicio de actualización de software para RICOH MP C305

Una vez preparada la actualización se desplegará en pantalla un menú para seleccionar las opciones que se desea actualizar como lo indica la Figura 3.42, tras lo cual se podrá verificar el avance de actualización mediante una barra de progreso como se observa en la Figura 3.43.



**Figura 3.42** Menú de actualización para RICOH MP C305



**Figura 3.43** Proceso de actualización de firmware para RICOH MP C305

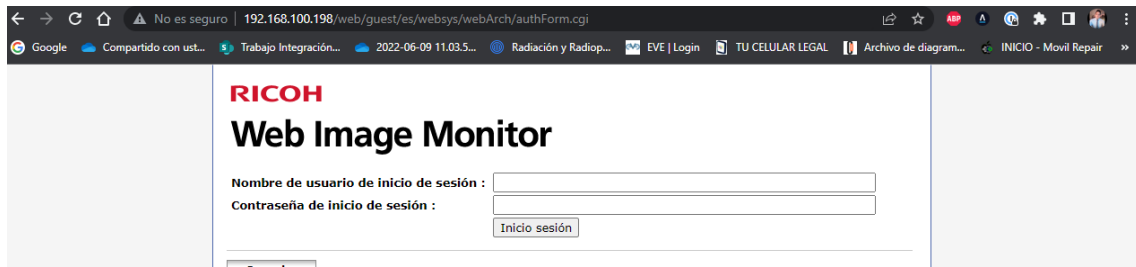
Si la instalación de la nueva actualización se realizó con éxito se observará que la impresora inicia de forma normal en su página de *home*, esto se muestra en la Figura 3.44.



**Figura 3.44** Pantalla principal para RICOH MP C305

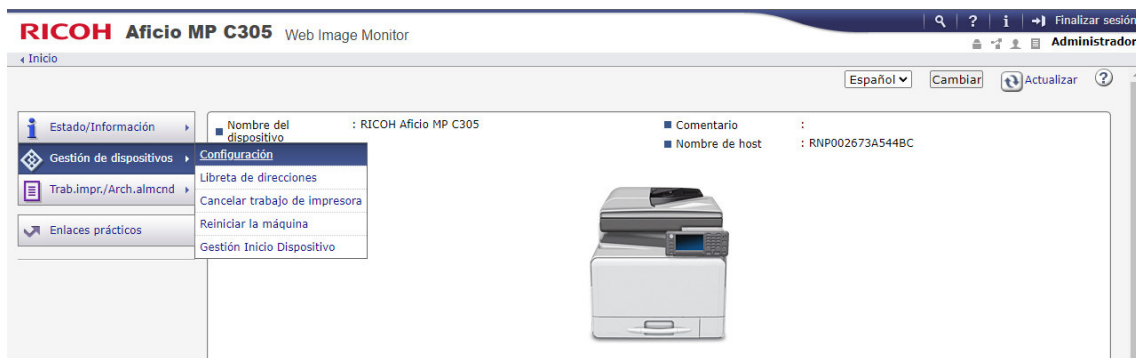
Las configuraciones a realizarse en el *host* tras la actualización del sistema del mismo son bastante simples y se las realizará a través de su interfaz *web* denominada *Web Monitor*, a la cual se puede acceder a través de su dirección *IP*, previamente configurada de manera

estática; dentro de la interfaz, se debe iniciar sesión para realizar las configuraciones como lo muestra la *Figura 3.45*.



**Figura 3.45** Interfaz web para RICOH MP C305

Una vez que se ha establecido el inicio de sesión, se debe dirigir al menú de la izquierda en el apartado de gestión de dispositivo para entrar en la opción de configuración de la forma que se muestra en la *Figura 3.46*, para seleccionar posteriormente de las opciones de red la parte de configuración *SNMP* que se observa en la *Figura 3.47*.



**Figura 3.46** Menú de configuración para RICOH MP C305 en la *Web Monitor*



**Figura 3.47** Configuración del protocolo *SNMP* para RICOH MP C305

Una vez dentro de la configuración del protocolo *SNMP* bastará con llenar los campos solicitados como el nombre de la comunidad a utilizarse, así como establecer la dirección *IP* perteneciente al servidor de *NEMS* que realizará el monitoreo. Esta configuración se constata en la *Figura 3.48*.

Comunidad					
Nº	Nombre de comunidad	Tipo de acceso	Tipo de protocolo	Activo/Inactivo	Dirección gestor
1	public	lectura-escritura	IPv4	<input checked="" type="radio"/> Activo <input type="radio"/> Inactivo	192.168.100.10
			IPv6	<input checked="" type="radio"/> Activo <input type="radio"/> Inactivo	::

**Figura 3.48** Creación de comunidad *SNMP* y establecimiento de *IP* para el servidor **Servidor *NEMS* de monitoreo**

Para realizar las configuraciones finales, se debe realizar en el servidor de *NEMS* al cual se deberá conectar mediante una sesión de *PUTTY* con el protocolo *SSH* como lo indica la Figura 3.49.

```

enriquezb@nems: ~
Last login: Sun Jul 31 13:28:43 2022 from 192.168.100.7

  NEMS
  LINUX

BY: ROBBIE FERGUSON
NEMSLINUX.COM

NEMS Platform....: Raspberry Pi 3 B+
NEMS Version....: 1.5.2 (Current Version is 1.5.2)
NEMS IP Address..: 192.168.100.10
Uptime.....: 0 days 5 hours 11 minutes 31 seconds
Load.....: 0.08 (1 minute) 0.17 (5 minutes) 0.22 (15 minutes)
           0.18 (1 week)
Memory.....: Total: 923 MB / Cached: 94 MB
           Used: 256 MB / Free: 96 MB
Disk Usage.....: You're using 17% of your root filesystem

enriquezb@nems:~ $

```

**Figura 3.49** Inicio de sesión en el servidor *NEMS* por *SSH*

Tras realizar el inicio de sesión se deberá acceder a la ruta especificada en la Figura 3.50 para poder acceder a los archivos de configuración que son necesarios modificar en el servidor.

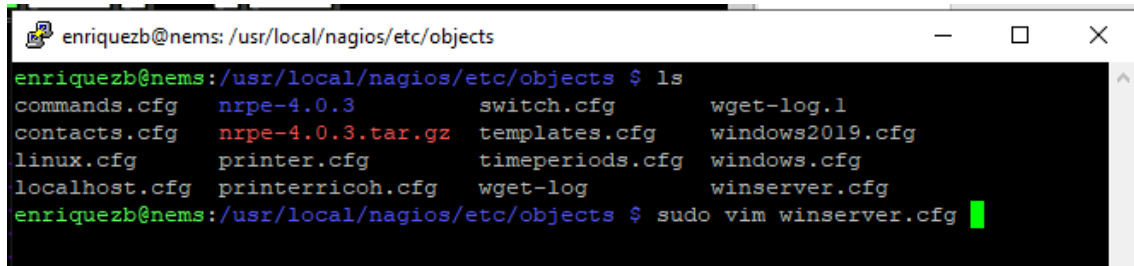
```

enriquezb@nems: /usr/local/nagios/etc/objects
enriquezb@nems:/usr/local/nagios/etc/objects $ ls
commands.cfg  nrpe-4.0.3      switch.cfg      wget-log.1
contacts.cfg  nrpe-4.0.3.tar.gz  templates.cfg  windows2019.cfg
linux.cfg     printer.cfg     timeperiods.cfg  windows.cfg
localhost.cfg  printerricoh.cfg  wget-log       winserver.cfg
enriquezb@nems:/usr/local/nagios/etc/objects $

```

**Figura 3.50** Ubicación de plantillas para administrar diferentes *hosts*

Dentro del directorio se procede a crear el archivo plantilla de configuración para el *host* de tipo servidor mediante el comando “sudo vim” para acceder con permisos avanzados, como se evidencia en la Figura 3.51.

A terminal window showing the user 'enriquezb' at 'nems' in the directory '/usr/local/nagios/etc/objects'. The user runs 'ls' and lists several configuration files including 'winserver.cfg'. Then, the user runs 'sudo vim winserver.cfg' to open the file for editing.

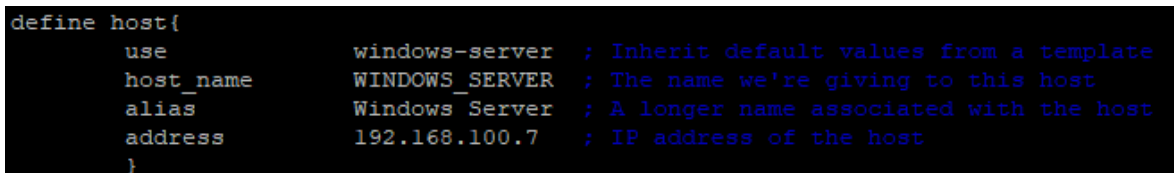
```
enriquezb@nems: /usr/local/nagios/etc/objects
enriquezb@nems:/usr/local/nagios/etc/objects $ ls
commands.cfg  nrpe-4.0.3      switch.cfg      wget-log.1
contacts.cfg  nrpe-4.0.3.tar.gz templates.cfg   windows2019.cfg
linux.cfg     printer.cfg     timeperiods.cfg windows.cfg
localhost.cfg printerricoh.cfg wget-log        winserver.cfg
enriquezb@nems:/usr/local/nagios/etc/objects $ sudo vim winserver.cfg
```

**Figura 3.51** Creación de la plantilla winserver.cfg

Los comandos dentro del archivo creado deben tener un formato que el servidor de *NEMS* sea capaz de leer la estructura de los mismos debe respetar dicho formato, comenzando por la definición del *host*, donde se especifica el nombre, el alias, y su dirección *IP* como se observa en la Figura 3.52 así como también se van a definir los servicios con lo que será monitorizado el *host*, mismos que deberán conservar también una estructura de datos estandarizada. Los servicios que se implementaron fueron:

- UPTIME.
- DISK USAGE.
- MEMORY USAGE.
- PING.
- CPU LOAD.

Cada uno de los ítems anteriores, bien definidos y acorde a la estructura establecida como se muestra en la Figura 3.53. donde también se debe especificar el nombre del *host*, el nombre del servicio, la estructura del comando que se va a ejecutar, independiente en cada caso y si se desea algo más personalizado, se puede añadir parámetros extras como es el caso del intervalo de tiempo en que se ejecutan dichos comandos [16].

A terminal window showing the configuration for a host in the 'winserver.cfg' file. The configuration uses a 'define host' block with fields for 'use', 'host\_name', 'alias', and 'address', each with a descriptive comment.

```
define host{
    use                windows-server ; Inherit default values from a template
    host_name          WINDOWS_SERVER ; The name we're giving to this host
    alias              Windows Server ; A longer name associated with the host
    address             192.168.100.7 ; IP address of the host
}
```

**Figura 3.52** Formato para la definición de *host* en la plantilla winserver.cfg.



```

define service{
    use                generic-service
    host_name          WINDOWS_SERVER
    service_description UPTIME
    check_command      check_nt!UPTIME
    check_interval     1
}

# Create a service for monitoring CPU load
# Change the host_name to match the name of the host you defined above

define service{
    use                generic-service
    host_name          WINDOWS_SERVER
    service_description CPU LOAD
    check_command      check_nt!CPULOAD!-1 5,80,90
    check_interval     1
}

# Create a service for monitoring memory usage
# Change the host_name to match the name of the host you defined above

define service{
    use                generic-service
    host_name          WINDOWS_SERVER
    service_description MEMORY USAGE
    check_command      check_nt!MEMUSE!-w 80 -c 90
    check_interval     1
}

# Create a service for monitoring C:\ disk usage
# Change the host_name to match the name of the host you defined above

define service{
    use                generic-service
    host_name          WINDOWS_SERVER
    service_description DISK USAGE
    check_command      check_nt!USEDISKSPACE!-1 c -w 80 -c 90
    check_interval     1
}

# Create a service for monitoring the W3SVC service

define service{
    use                generic-service
    host_name          WINDOWS_SERVER
    service_description PING
    check_command      check_ping!500.0,20%!800.0,60%
    check_interval     1
}

```

**Figura 3.53** Definición de servicios para administrar el *host* de *Windows*

Para la configuración que corresponde al *host* de *Linux* se deberá crear la respectiva plantilla a la que se le llamó “linux.cfg”, también creada con el comando “sudo vim” tal como se aprecia en la Figura 3.54 [16].

```

enriquezb@nems:~ $ sudo vim linux.cfg

```

**Figura 3.54** Creación de la plantilla linux.cfg

La configuración y definición tanto del *host* como de los servicios establecidos se realizan de la misma manera que para el caso del *host* de *Windows* puesto que deberá ser leído por el servidor de *NEMS* por lo que lleva la misma estructura de datos, pero con la información del *host* de *Linux* como se observa en la Figura 3.55 [16].

```

##DEFINICION DE SERVIDORES

define host{
    use                linux-server
    host_name          CLIENTE_LINUX
    alias              UbuntuClient
    check_interval     1
    address            192.168.100.19
}

```

**Figura 3.55** Formato para la definición de *host* en la plantilla *linux.cfg*

Los servicios que se definieron para este *host* fueron:

- CPU LOAD.
- DISK USAGE.
- PING.
- CURRENT USERS.

La definición de los servicios antes mencionados se aprecian en la Figura 3.56 [16].

```

##### DEFINICION DE SERVICIOS

define service{
    use                generic-service
    host_name          CLIENTE_LINUX
    service_description DISK USAGE
    check_command      check_nrpe!check_hdal
    check_interval     1
}

define service{
    use                generic-service
    host_name          CLIENTE_LINUX
    service_description UPTIME
    check_command      check_nrpe!check_uptime -H 192.168.100.19
    check_interval     1
}

define service{
    use                generic-service
    host_name          CLIENTE_LINUX
    service_description CPU LOAD
    check_command      check_nrpe!check_load
    check_interval     1
}

define service{
    use                generic-service
    host_name          CLIENTE_LINUX
    service_description PING
    check_command      check_ping!500.0,20%!800.0,60%
    check_interval     1
}

define service{
    use                generic-service
    host_name          CLIENTE_LINUX
    service_description CURRENT USERS
    check_command      check_nrpe!check_users
    check_interval     1
}

```

**Figura 3.56** Definición de servicios para administrar el *host* de *Linux*

En el caso específico del agente *NRPE* para el monitoreo de dispositivos con *Linux* se requiere de una configuración adicional, la misma que se hará en el archivo de “*commands.cfg*” para realizar la definición del comando y que el servidor de *NEMS* pueda interpretar los datos entregados por el agente de la forma en que se observa en la Figura 3.57.

```
define command {
    command_name check_nrpe
    command_line $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

```

**Figura 3.57** Modificación del archivo `commands.cfg` para *NRPE*

Ya que los *hosts* de tipo *Linux* y *Windows* fueron añadidos de forma manual realizando cambios en los archivos fuente del servidor *NEMS* fue necesario editar también el archivo “`nagios.cfg`”, donde se le indicó al servidor los archivos que deben ser leídos para añadir así los *hosts* a monitorear como se indica en la Figura 3.58.

```
enriquezb@nems: /usr/local/nagios/etc
#####
#
# NAGIOS.CFG - Sample Main Config File for Nagios
#
#####

# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!

log_file=/var/log/nagios/nagios.log

# We're using NConf, so all defaults are removed and instead, we'll use:
cfg_dir=/etc/nems/conf/global
cfg_dir=/etc/nems/conf/Default_collector

cfg_file=/usr/local/nagios/etc/objects/windows2019.cfg
cfg_file=/usr/local/nagios/etc/objects/linux.cfg
#cfg_file=/usr/local/nagios/etc/objects/printerricoh.cfg

# OBJECT CACHE FILE
# This option determines where object definitions are cached when
# Nagios starts/restarts. The CGIs read object definitions from
# this cache file (rather than looking at the object config files
# directly) in order to prevent inconsistencies that can occur
# when the config files are modified after Nagios starts.

object_cache_file=/usr/local/nagios/var/objects.cache

```

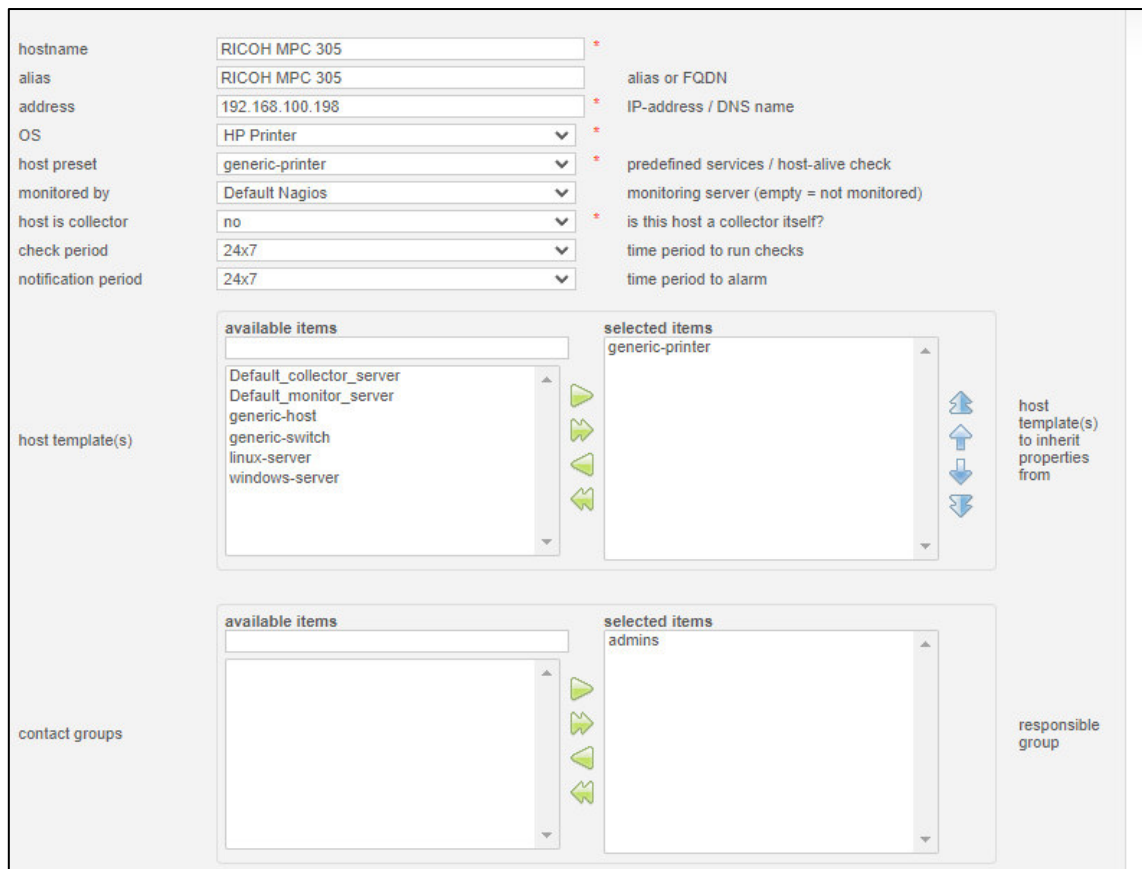
**Figura 3.58** modificación del archivo `nagios.cfg` para uso de plantillas personalizadas

El *host* de tipo impresora fue añadido mediante el uso de la herramienta *NConf* propia de *NEMS Nagios* en su interfaz *web*, desde donde se puede realizar algunas configuraciones específicas referente a los *host* y servicios implementados, como se muestra en la Figura 3.59, siendo así que al pulsar en la opción de “*add*” en el apartado de *hosts* permite configurar un nuevo dispositivo para ser monitoreado, en este caso una impresora de la marca *RICOH* modelo *MP C305*, de la cual se deberá llenar la información en los campos

solicitados, como por ejemplo su dirección *IP* indispensable para la tarea de monitorear dicho *host*. Las configuraciones y opciones que se deben elegir se pueden apreciar en la Figura 3.60.

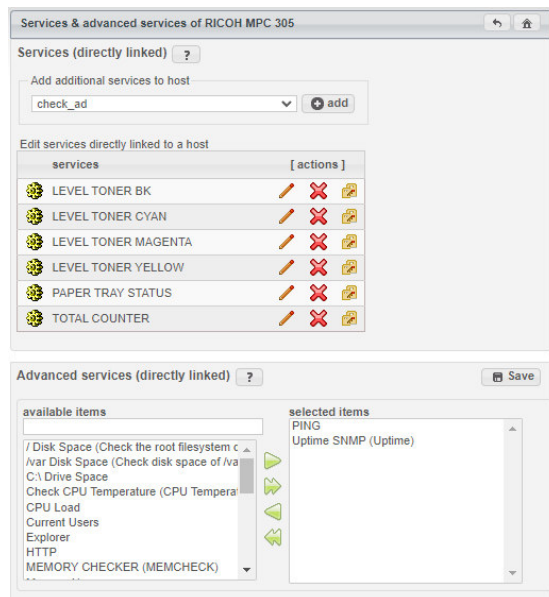


**Figura 3.59** Menú de *NEMS NConf* para administrar *hosts*



**Figura 3.60** Añadiendo *host* de tipo impresora desde NConf

Una vez que se ha añadido el *host* se deberán configurar los servicios que se van a monitorear del mismo, para este caso se consideró los siguientes que se evidencian en la **¡Error! No se encuentra el origen de la referencia..**



**Figura 3.61** Añadiendo servicios predefinidos y personalizados al *host*

### Servicios Predefinidos

- *UPTIME.*
- *PING.*

### Servicios Personalizados

- *LEVEL TONER BK.*
- *LEVEL TONER CYAN.*
- *LEVEL TONER MAGENTA.*
- *LEVEL TONER YELLOW.*
- *PAPER TRY STATUS.*
- *TOTAL COUNTER.*

En específico los servicios personalizados deben cumplir con una estructura en la definición de su comando que se construye de la siguiente manera.

```
check_nt -H $HOSTADDRESS$ -C <ComunityName> -O <OID> -u <value> -w <value> -c <value>
```

Dicha estructura deberá ser configurada para cada servicio con su respectivo *OID* como se observa en la Figura 3.62, misma que para el caso de las IMPRESORAS RICOH puede encontrarse en la documentación oficial de RICOH.

The screenshot shows the Nagios service configuration page. It includes various configuration options such as 'notes', 'notes URL', 'action URL', 'max check attempts', 'check interval', 'retry interval', 'first notification delay', 'notification interval', 'notification options', 'active checking', 'passive checking', 'notification enabled', 'check freshness', and 'freshness threshold'. There are also 'available items' and 'selected items' lists, and a 'service parameters' field with a command: ARG1: -C public -o 1.3.6.1.4.1.367.3.2.1.2.24.1.1.5.1 -u % -w 30:21 -c 20:10 (BK) [17].

**Figura 3.62** Ejemplo de estructura de servicio personalizado para *host* tipo impresora

Las sentencias para los comandos que se utilizaron en los servicios implementados para el *host* de tipo impresora son:

- -C public -o 1.3.6.1.4.1.367.3.2.1.2.24.1.1.5.1 -u % -w 30:21 -c 20:10 (BK) [17].
- -C public -o 1.3.6.1.4.1.367.3.2.1.2.24.1.1.5.2 -u % -w 30:21 -c 20:10 (C) [17].
- -C public -o 1.3.6.1.4.1.367.3.2.1.2.24.1.1.5.3 -u % -w 30:21 -c 20:10 (M) [17].
- -C public -o 1.3.6.1.4.1.367.3.2.1.2.24.1.1.5.4 -u % -w 30:21 -c 20:10 (Y) [17].
- -C public -o 1.3.6.1.4.1.367.3.2.1.2.20.2.2.1.10.2.1 -c 0:-2 (PAPER TRY STATUS) [17].
- -C public -o 1.3.6.1.4.1.367.3.2.1.2.19.5.1.9.12 -w 37920:37910 -c 37930:37921 (TOTAL COUNTER) [17].

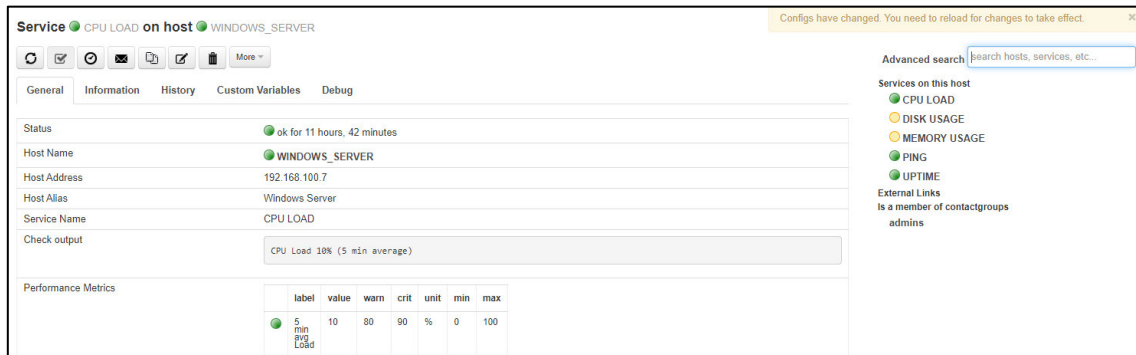
### 3.5 Realizar pruebas de funcionamiento del prototipo

Para garantizar el correcto funcionamiento del prototipo se procede a realizar pruebas aleatorias que logren evidenciar cada uno de los servicios mediante la comparación de lecturas tanto en el servidor de *NEMS* como en el cliente en el que se está realizando la consulta.

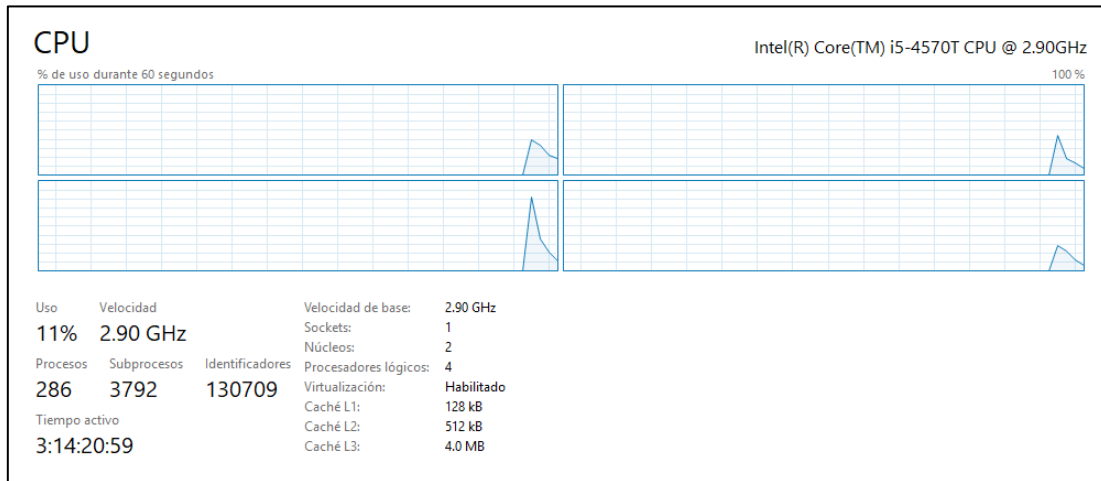
#### Pruebas de funcionamiento para el *host* de tipo *Windows*

Como se puede observar en la Figura 3.63 el sistema de monitoreo muestra una carga de la *CPU* de 10% en promedio, mientras que el *host*, a través del administrador de tareas

muestra una lectura del 11% para la carga de la *CPU*, como lo muestra la Figura 3.64 con lo que se evidencia el correcto funcionamiento para el servicio de *CPU LOAD*.



**Figura 3.63** Estado del servicio *CPU LOAD* para *host Windows* en el servidor



**Figura 3.64** Estado del servicio *CPU LOAD* para *host Windows* en el cliente

El servidor de *NEMS Nagios* muestra los valores de uso de memoria del disco duro tales como cantidad de memoria total, memoria usada y memoria libre como se observa en la Figura 3.65, valores que coinciden con la información obtenida en el cliente mediante el administrador de discos, como se muestra en la Figura 3.66.



**Figura 3.65** Estado del servicio *DISK USAGE* para *host Windows* en el servidor

Volumen	Distribución	Tipo	Sistema de ...	Estado	Capacidad	Espacio ...	% disponible
(C:)	Simple	Básico	NTFS	Correcto (...)	222.97 GB	32.78 GB	15 %

**Figura 3.66** Estado del servicio *DISK USAGE* para *host Windows* en el cliente

En el caso del servicio de *MEMORY USAGE* de puede observar en el sistema de *NEMS Nagios*, los valores correspondientes a memoria total (confirmada), memoria usada y memoria libre según se observa en la Figura 3.67, y cuya información se ratifica en el cliente con los valores obtenidos del administrador de tareas como se aprecia en la Figura 3.68.

Service: MEMORY USAGE on host WINDOWS\_SERVER

Status: warning for 5 hours, 50 minutes (not acknowledged)

Host Name: WINDOWS\_SERVER

Host Address: 192.168.100.7

Host Alias: Windows Server

Service Name: MEMORY USAGE

Check output: Memory usage: total:21457.45 MB - used: 18797.45 MB (88%) - free: 2659.99 MB (12%)

label	value	warn	crit	unit	min	max
Memory usage	18797.45	17165.96	19311.70	MB	0.00	21457.45

**Figura 3.67** Estado del servicio *MEMORY USAGE* para *host Windows* en el servidor

Administrador de tareas

Memoria: 12.0 GB DDR3

Uso de memoria: 8.2/12.0 GB (68%)

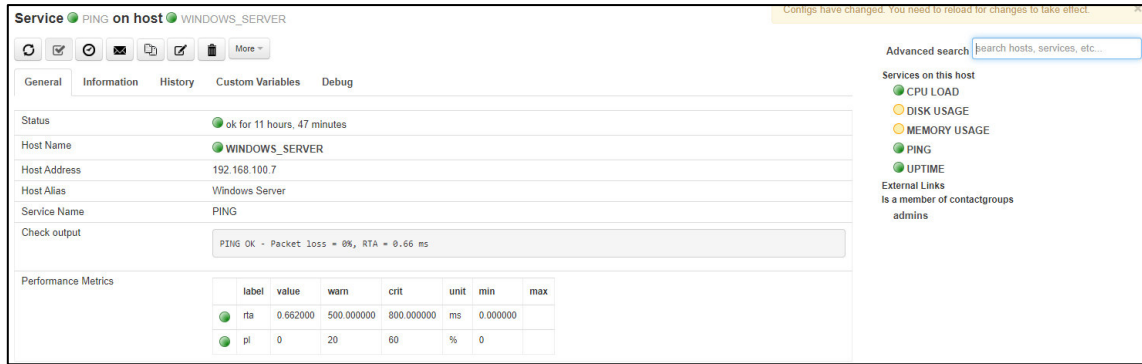
Composición de memoria:

En uso (comprimido)	Disponibles	Velocidad:	1333 MHz
8.1 GB (0 MB)	3.8 GB	Ranuras usadas:	2 de 4
Confirmada	En caché	Factor de forma:	DIMM
18.4/21.0 GB	3.8 GB	Reservada para hardware:	46.7 MB
Bloque paginado	Bloque no paginado		
436 MB	208 MB		

**Figura 3.68** Estado del servicio *MEMORY USAGE* para *host Windows* en el cliente

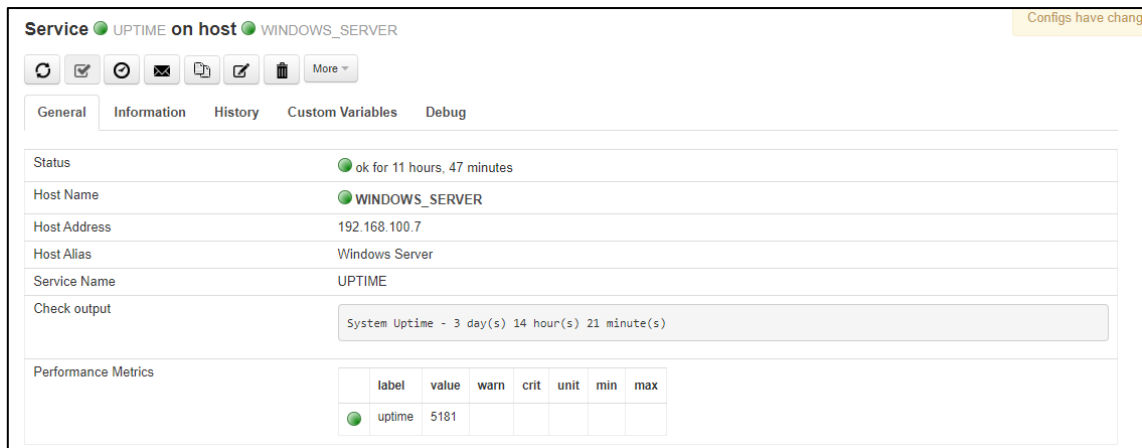
En la Figura 3.69 se puede observar el estado *OK* del servicio de *PING* para el *host* con 0% de paquetes perdidos con lo que se garantiza que el *host* se encuentra en estado *UP*.





**Figura 3.69** Estado del servicio *PING* para *host Windows* en el servidor

Para el servicio de *UPTIME* se puede observar en la Figura 3.70 el tiempo que el dispositivo ha permanecido encendido, mismo que se puede verificar con la información obtenida en el cliente como se muestra en la Figura 3.71.



**Figura 3.70** Estado del servicio *UPTIME* para *host Windows* en el servidor

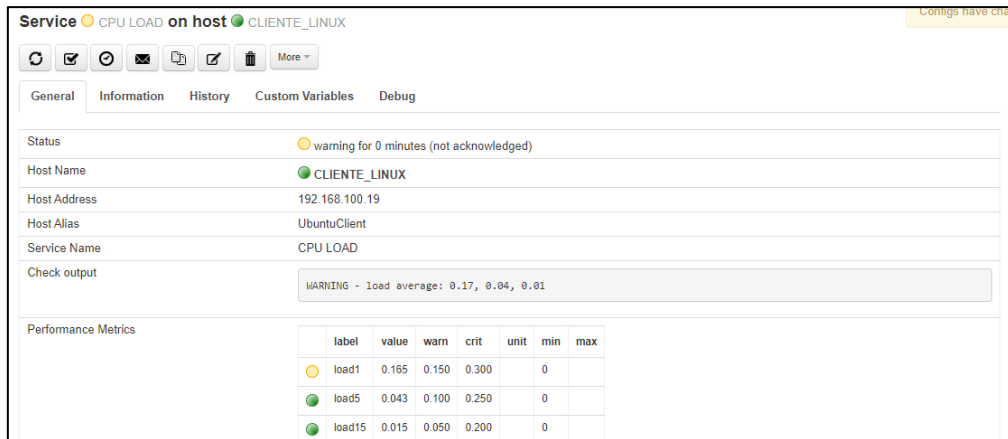


**Figura 3.71** Estado del servicio *UPTIME* para *host Windows* en el cliente

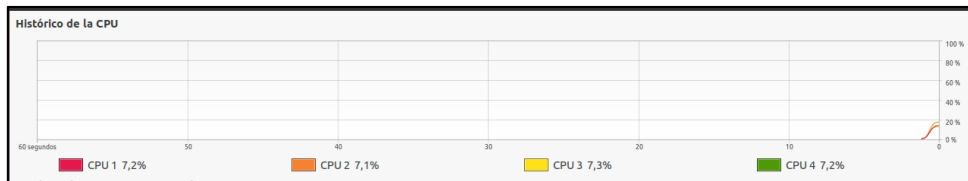
### Pruebas de funcionamiento para el *host* de tipo *Linux*

Para el servicio de *CPU LOAD*, el sistema de *NEMS Nagios* entrega 3 valores que corresponden a las lecturas del servicio en promedio para 1 minuto, 5 minutos y 15 minutos.

De considerar a la lectura de un minuto como la lectura más cercana a una instantánea se observa un valor de 0.17 o su equivalente a 17% como carga de la *CPU* como se puede identificar en la Figura 3.72 y corroborar de manera visual en la gráfica que se obtiene en el cliente mediante la aplicación de “*System*” propia de *Linux* donde se observa que dicha carga no supera el umbral del 20% según lo que se observa en la Figura 3.73, por lo que se considera una lectura correcta en el sistema de monitoreo.

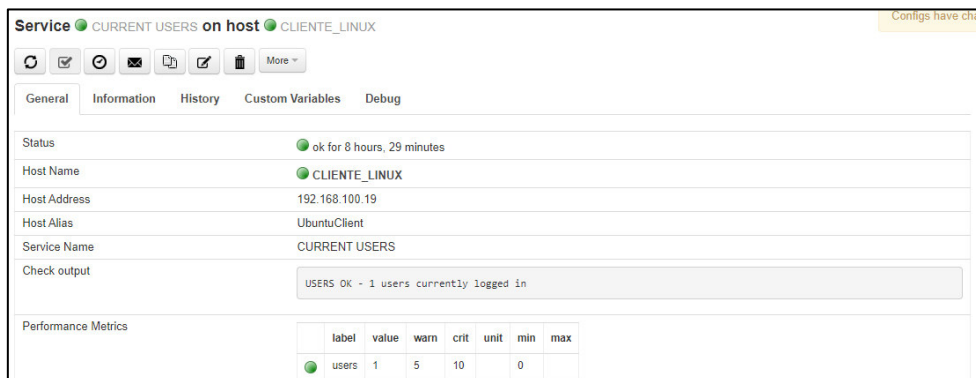


**Figura 3.72** Estado del servicio *CPU LOAD* para *host Linux* en el servidor

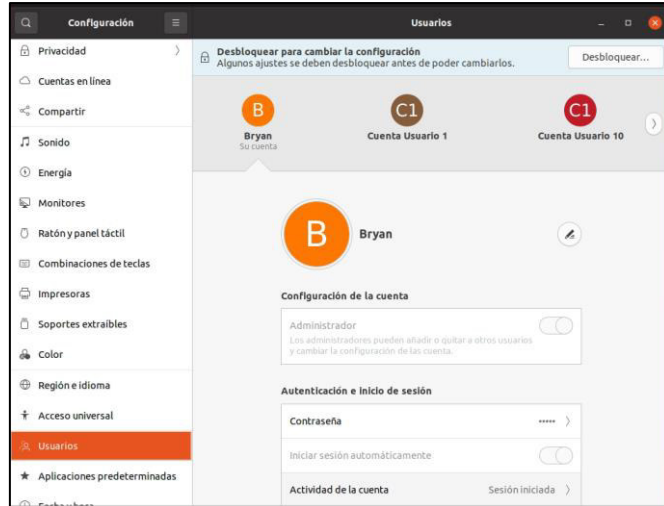


**Figura 3.73** Estado del servicio *CPU LOAD* para *host Linux* en el cliente

El servicio de *CURRENT USERS* dentro del sistema de *NEMS Nagios* muestra un usuario actualmente conectado según lo muestra la Figura 3.74 y que se evidencia en el lado del cliente, en el apartado de configuración reiterando que solamente un usuario tiene una sesión iniciada en el dispositivo como se ve en la Figura 3.75.

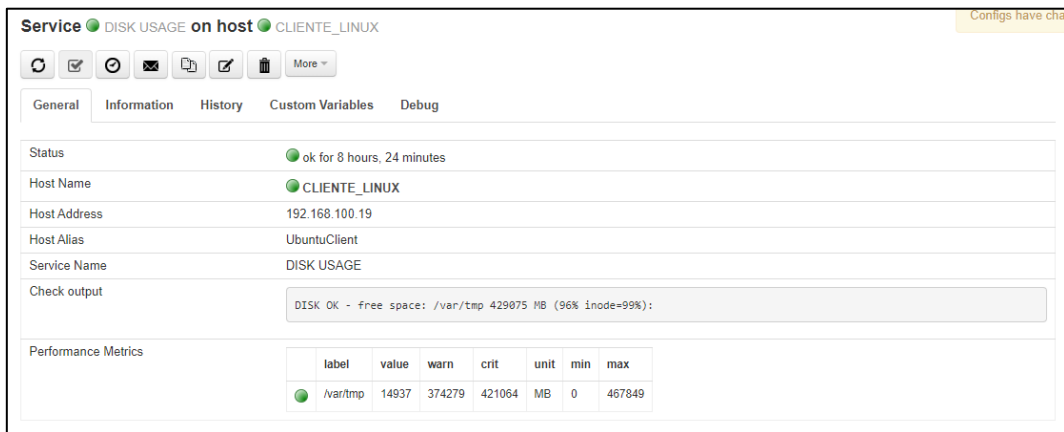


**Figura 3.74** Estado del servicio *CURRENT USERS* para *host Linux* en el servidor



**Figura 3.75** Estado del servicio CURRENT USERS para *host Linux* en el cliente

En este caso, el servicio de *DISK USAGE* hace referencia a la lectura del espacio libre en el disco a partir de lo cual se observa en la Figura 3.76 una lectura de memoria libre del 96% de un total de 99% lo que hace referencia a un uso del 3%, valor que se confirma en el cliente mediante el apartado del sistema de archivos de la aplicación “System” según se presenta en la Figura 3.77.

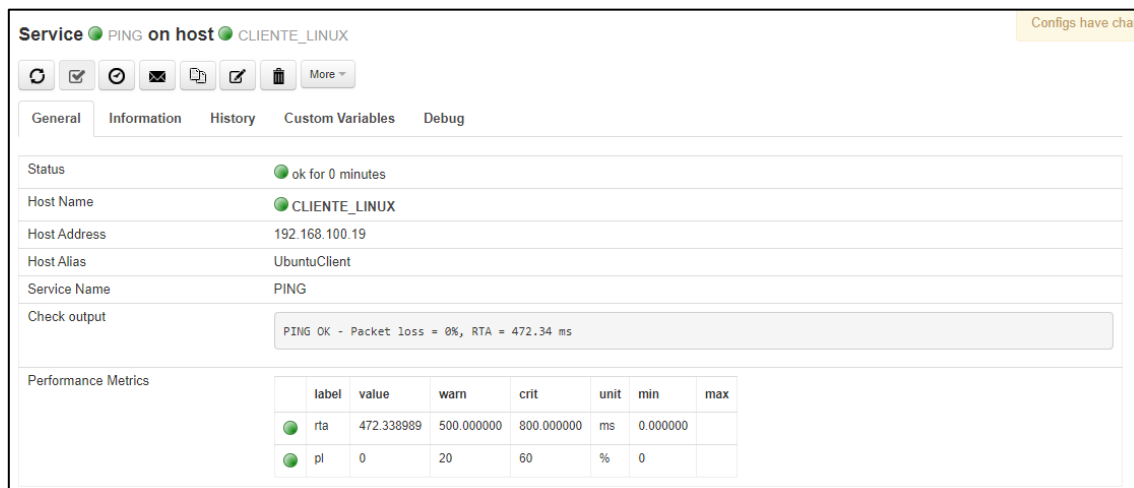


**Figura 3.76** Estado del servicio DISK USAGE para *host Linux* en el servidor



**Figura 3.77** Estado del servicio DISK USAGE para *host Linux* en el cliente

Se puede verificar en la Figura 3.78 el estado *OK* del servicio de *PING* con una pérdida de paquetes de 0% con lo que se puede establecer al *host* en estado *UP*.



Service **PING on host CLIENTE\_LINUX** Configs have changed

General Information History Custom Variables Debug

Status ● ok for 0 minutes

Host Name ● CLIENTE\_LINUX

Host Address 192.168.100.19

Host Alias UbuntuClient

Service Name PING

Check output PING OK - Packet loss = 0%, RTA = 472.34 ms

Performance Metrics

	label	value	warn	crit	unit	min	max
<span style="color: green;">●</span>	rta	472.338989	500.000000	800.000000	ms	0.000000	
<span style="color: green;">●</span>	pl	0	20	60	%	0	

**Figura 3.78** Estado del servicio *PING* para *host Linux* en el servidor

### Prueba de Funcionamiento para el *host* de tipo impresora

La verificación del correcto funcionamiento de los servicios que se monitorean en el *host* de tipo impresora como lo es, el nivel de cada uno de los cartuchos de tinta y demás, se realizó mediante la interfaz web de la que dispone el propio *host* pudiendo observarse en la Figura 3.79, Figura 3.80, Figura 3.81 y Figura 3.82 los valores que se van a corroborar tanto en la interfaz *web* del dispositivo como en el panel físico de la impresora según se observa en la Figura 3.83 para el caso de la interfaz *web* y en la Figura 3.84, para el caso del dispositivo físico.



Service **LEVEL TONER BK on host RICOH MPC 305** Configs have changed

General Information History Custom Variables Debug

Status ● ok for 1 day, 4 hours

Host Name ● RICOH MPC 305

Host Address 192.168.100.198

Host Alias RICOH MPC 305

Service Name LEVEL TONER BK

Check output SNMP OK - 60 %

Performance Metrics

	label	value	warn	crit	unit	min	max
<span style="color: red;">●</span>	iso.3.6.1.4.1.367.3.2.1.2.24.1.1.5.1	60	30:21	20:10			

**Figura 3.79** Estado del servicio *LEVEL TONER BK* en *host* tipo impresora en el servidor

Service ● LEVEL TONER CYAN on host ● RICOH MPC 305 Configs have changed

🔄 📧 🕒 📄 📝 🗑️ More ▾

General | Information | History | Custom Variables | Debug

Status ● critical for 1 day, 4 hours (not acknowledged)

Host Name ● RICOH MPC 305

Host Address 192.168.100.198

Host Alias RICOH MPC 305

Service Name LEVEL TONER CYAN

Check output SNMP CRITICAL - \*20\* %

Performance Metrics

	label	value	warn	crit	unit	min	max
<span style="color: red;">●</span>	iso.3.6.1.4.1.367.3.2.1.2.24.1.1.5.2	20	30:21	20:10			

**Figura 3.80** Estado del servicio *LEVEL TONER CYAN* en *host* tipo impresora en el servidor

Service ● LEVEL TONER MAGENTA on host ● RICOH MPC 305 Configs have changed

🔄 📧 🕒 📄 📝 🗑️ More ▾

General | Information | History | Custom Variables | Debug

Status ● ok for 1 day, 4 hours

Host Name ● RICOH MPC 305

Host Address 192.168.100.198

Host Alias RICOH MPC 305

Service Name LEVEL TONER MAGENTA

Check output SNMP OK - 70 %

Performance Metrics

	label	value	warn	crit	unit	min	max
<span style="color: red;">●</span>	iso.3.6.1.4.1.367.3.2.1.2.24.1.1.5.3	70	30:21	20:10			

**Figura 3.81** Estado del servicio *LEVEL TONER MAGENTA* en *host* tipo impresora en el servidor

Service ● LEVEL TONER YELLOW on host ● RICOH MPC 305 Configs have changed

🔄 📧 🕒 📄 📝 🗑️ More ▾

General | Information | History | Custom Variables | Debug

Status ● ok for 12 hours, 7 minutes

Host Name ● RICOH MPC 305

Host Address 192.168.100.198

Host Alias RICOH MPC 305

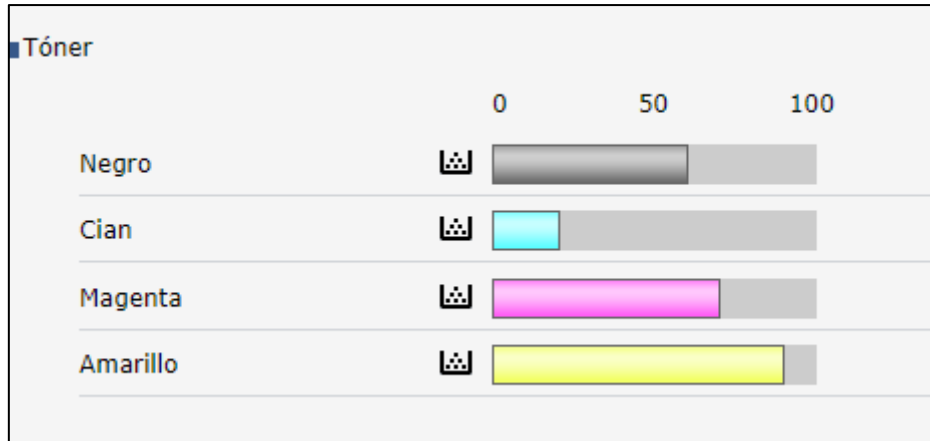
Service Name LEVEL TONER YELLOW

Check output SNMP OK - 90 %

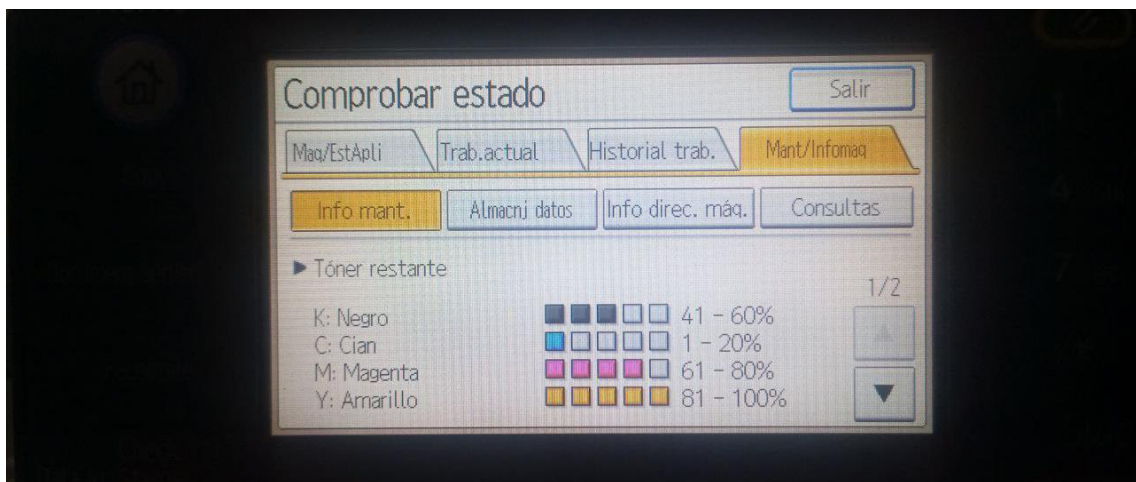
Performance Metrics

	label	value	warn	crit	unit	min	max
<span style="color: red;">●</span>	iso.3.6.1.4.1.367.3.2.1.2.24.1.1.5.4	90	30:21	20:10			

**Figura 3.82** Estado del servicio *LEVEL TONER YELLOW* en *host* tipo impresora en el servidor



**Figura 3.83** Niveles de toner BK, C, M, Y lectura de la *Web Monitor*



**Figura 3.84** Niveles de toner BK, C, M, Y lectura en el cliente en vivo

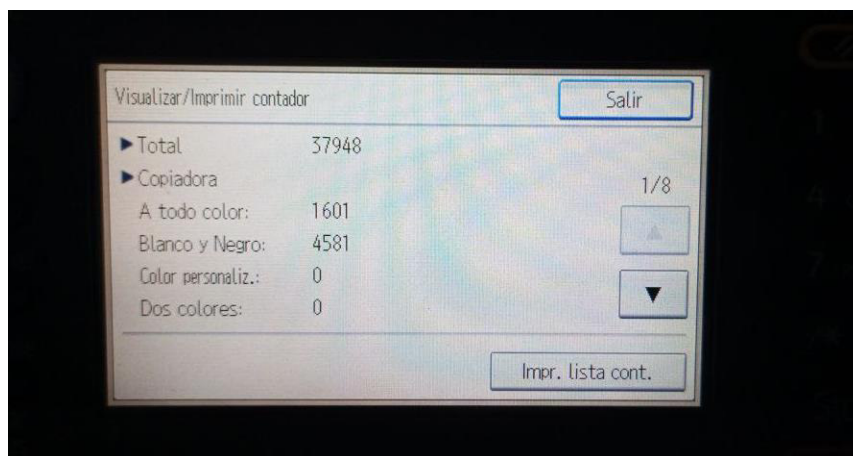
La lectura del servicio *TOTAL COUNTER* que se muestra en la Figura 3.85 es verificado de manera exacta mediante el uso de la interfaz web propia del dispositivo que también entrega la lectura correspondiente al contador del equipo validando que los dos valores corresponden de forma correcta como se muestra en la Figura 3.86.



**Figura 3.85** Estado del servicio *TOTAL COUNTER* del *host* tipo impresora en el servidor



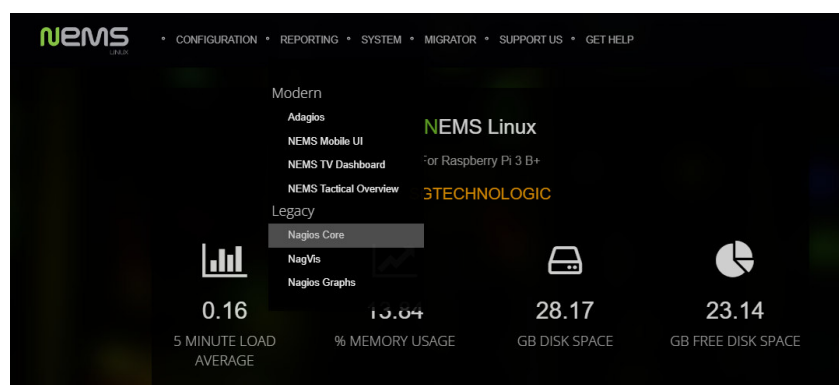
**Figura 3.86** Estado del servicio *TOTAL COUNTER* del *host* tipo impresora en el cliente Web Monitor



**Figura 3.87** Estado del servicio *TOTAL COUNTER* del *host* tipo impresora en el cliente en vivo

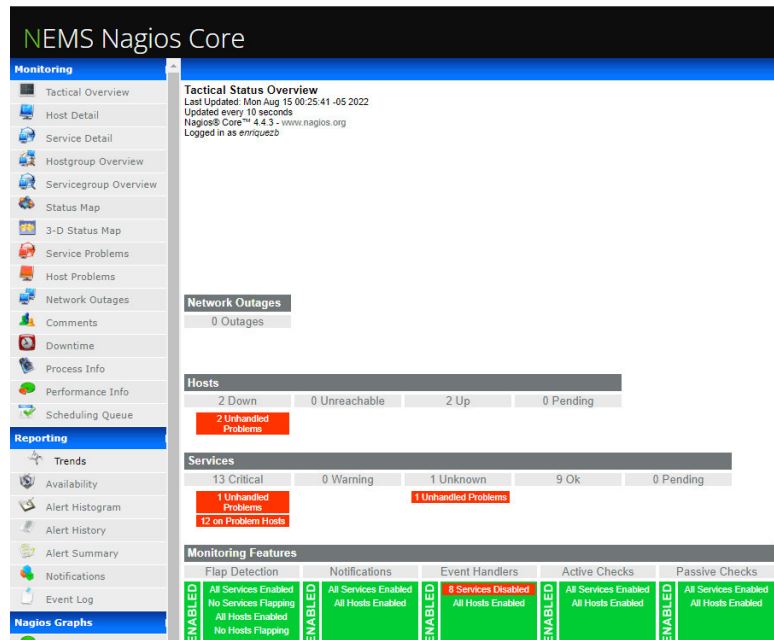
### Reportes estadísticos y notificaciones entregados por el sistema de monitorea para cada *host*

Una de las herramientas más útiles que ofrece el sistema de monitoreo de *NEMS Nagios* es el acceso a reportes gráficos que son generados de forma automática por el mismo en tiempo real y a los cuales se puede acceder desde la página de *Nagios Core* de la pestaña *Reporting* de la página principal del sistema de monitoreo como lo muestra la Figura 3.88.



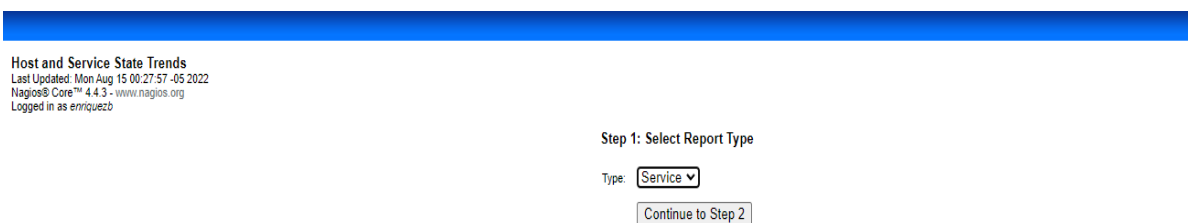
**Figura 3.88** Acceso a la página de *Nagios Core* desde la página principal

Para visualizar los reportes que se han generado para cada *host* se debe dirigir a la sección *trends* del menú que aparece a la izquierda de la página de *Nagios Core*, como se puede ver en la Figura 3.89.



**Figura 3.89** Acceso a los reportes estadísticos del sistema desde la opción *trends*

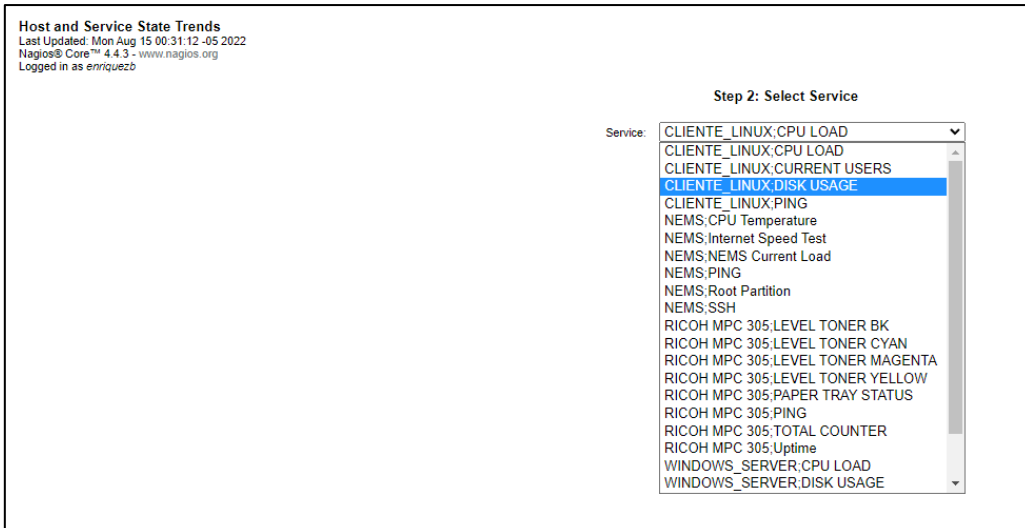
El sistema permite visualizar los reportes por *host* o por servicio, siendo la segunda opción la más recomendada puesto que entrega información más detallada y útil al momento de realizar el monitoreo de los *hosts*, por lo que se deberá elegir preferentemente la opción de *service* y continuar hacia el paso 2 como se aprecia en la Figura 3.90.



**Figura 3.90** Selección del tipo de reporte que se desea visualizar

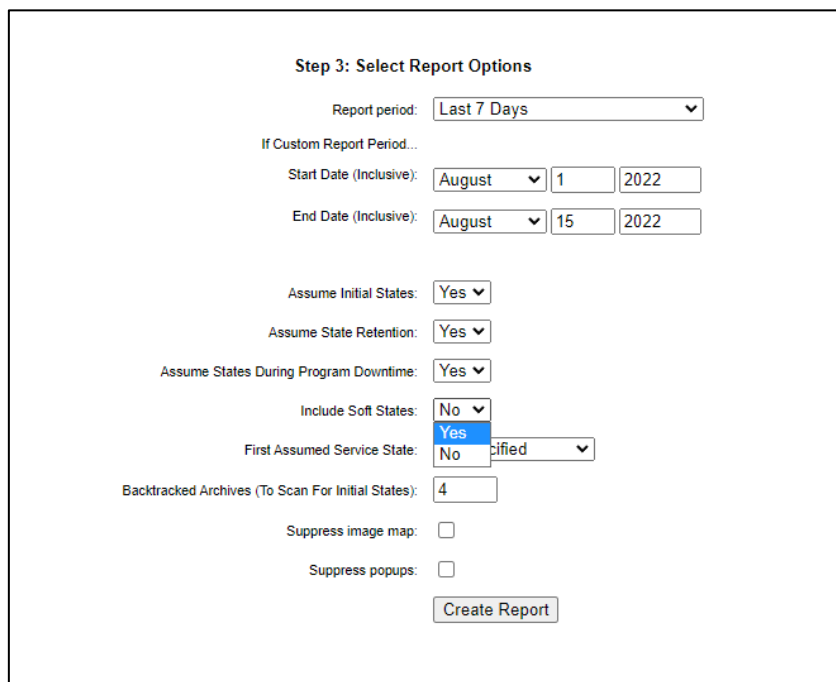
En la siguiente ventana se puede realizar una selección de cualquiera de los servicios que se están monitoreando en los *hosts* para poder visualizar el reporte histórico y estadístico del mismo, teniendo disponible un menú desplegable con todos los *hosts* y servicios que se están monitoreando, de la forma en que se muestra en la Figura 3.91.





**Figura 3.91** Menú de selección de *host* y servicio para la entrega del reporte

Habiendo seleccionado el *host* y servicio del cual se desea visualizar el reporte, se debe configurar el periodo a mostrar considerando en esta ventana una opción importante, que es marcar como *yes* la opción de “*Include soft states*”, pues de lo contrario no se podrá visualizar la información solicitada por tener deshabilitada dicha opción, esta configuración se puede observar en la Figura 3.92.



**Figura 3.92** Selección del periodo del cual se desea obtener el reporte

Ya con las opciones pertinentes seleccionadas, se obtendrá un reporte estadístico del servicio seleccionado para determinado *host* como se puede apreciar en la Figura 3.93, Figura 3.94, Figura 3.95, Figura 3.96, Figura 3.97 y Figura 3.98

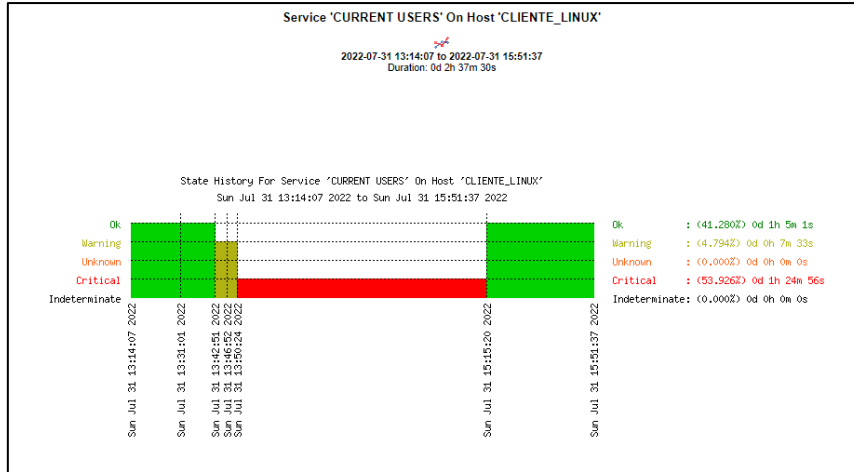


Figura 3.93 Ejemplo de reporte estadístico para servicio del *host Linux*

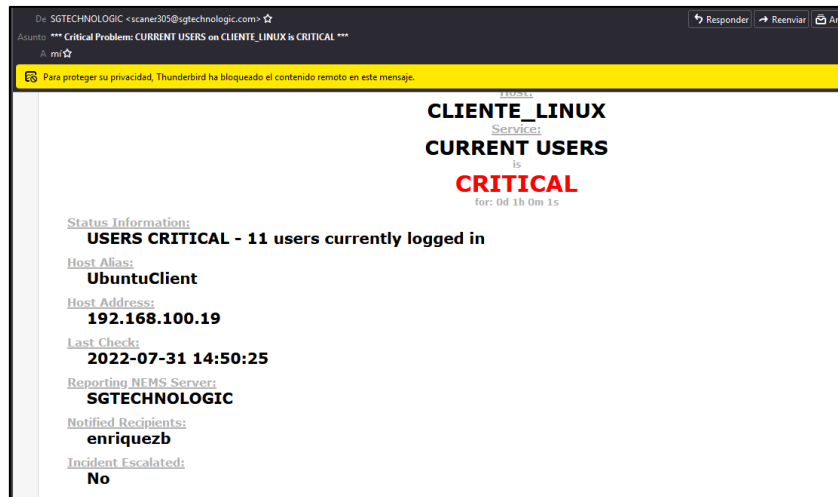


Figura 3.94 Notificación por correo del evento reportado para *host Linux*

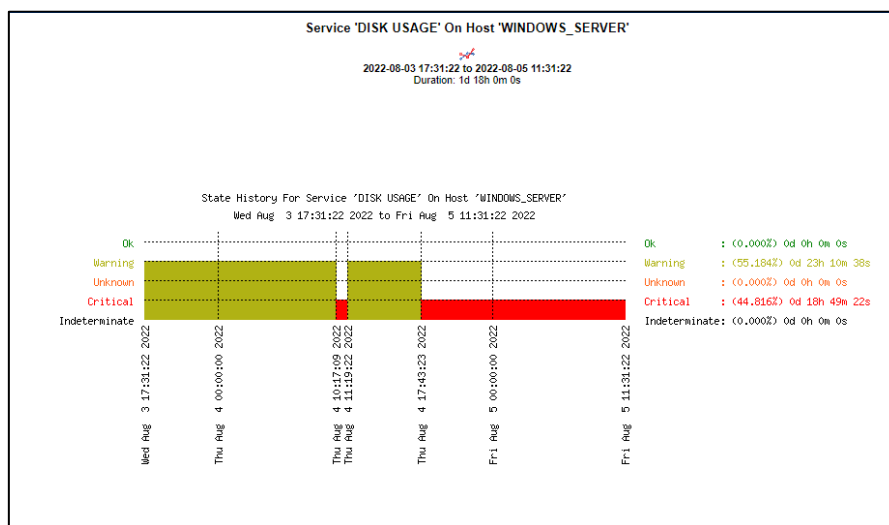


Figura 3.95 Ejemplo de reporte estadístico para servicio del *host Windows*

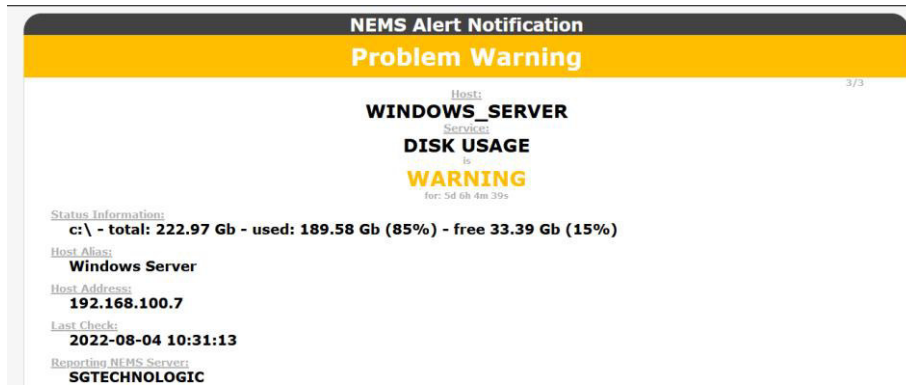


Figura 3.96 Notificación por correo del evento reportado para *host Windows*

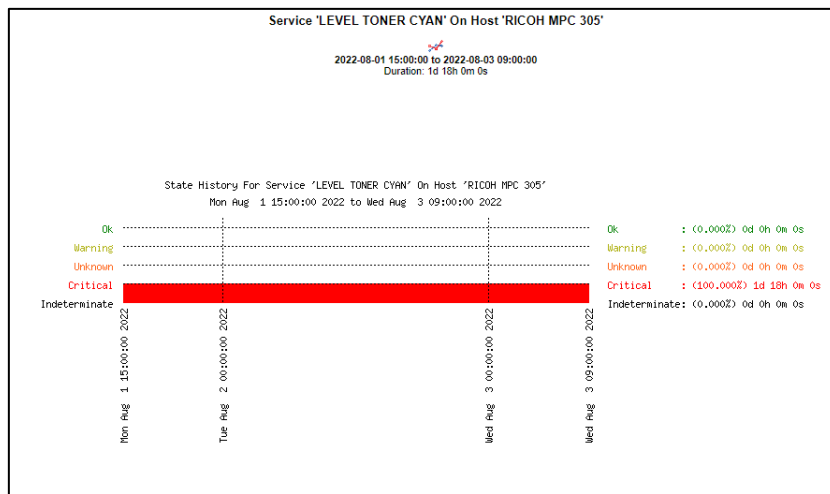


Figura 3.97 Ejemplo de reporte estadístico para servicio del *host* tipo impresora

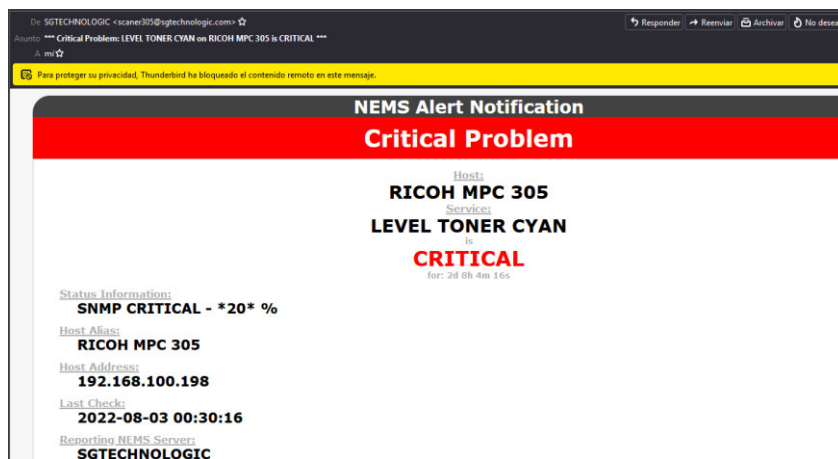


Figura 3.98 Notificación por correo del evento reportado para *host* tipo impresora

El sistema de *NEMS Nagios*, cuenta además como herramientas de visualización las ventanas de *NEMS TC Dashboard*, que se muestra en la Figura 3.99, misma que permite una visualización superficial de aquellos *hosts* que se encuentran con algún tipo de alerta,

permitiendo conocer directamente los problemas activos y actuales presentes en los *hosts* que se están monitoreando.

Cuenta con una interfaz adaptada para para dispositivos móviles llamada *Nems Mobile UI* para tener una visualización más optimizada para la pantalla de un dispositivo móvil en el caso de requerir acceder a dicha información desde un teléfono celular o *Tablet*, como se aprecia en la Figura 3.100.

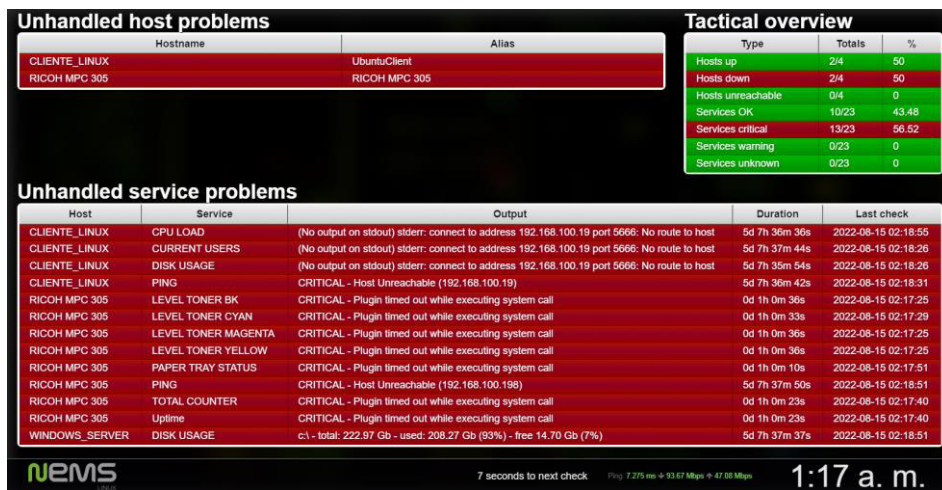


Figura 3.99 Ventana de *NEMS TV Dashboard*



Figura 3.100 Ventana de *NEMS Mobile UI*

Por último, se tiene una ventana llamada *NEMS Tactical overview*, que permite visualizar información un poco más detallada que en el caso de las ventanas anteriores y además

ofrece una alerta sonora y audible que notifica el estado de los *hosts* que están siendo monitoreados como lo muestra la Figura 3.101.



**Figura 3.101** Ventana de *NEMS Tactical overview*

### **Demostración del funcionamiento del prototipo**

En la sección Código QR de la implementación y pruebas de funcionamiento del prototipo se puede observar el código QR que direcciona hacia el video demostrativo del funcionamiento del prototipo, para su apropiada visualización.

### **Mantenimiento del prototipo realizado**

En la sección Código QR del mantenimiento del prototipo se encuentra en código QR que redirige hacia el video que muestra la realización del correcto mantenimiento que se debe aplicar al prototipo y otra información relevante respecto al tema.

### **Costos del Prototipo**

Los costos presentados a continuación están basados en precios referenciales obtenidos de la página Ebay.com, tienda en línea.

**Tabla 3.4** Costos de implementación del Prototipo

Material	Cantidad	Precio Unitario	Precio Total
Raspberry Pi model 3B +	1 (u)	\$ 63.00	\$ 63.00
Case carcasa para Raspberry Pi	1 (u)	\$ 15.00	\$ 15.00
Memoria Kingston 32 (Gb)	2 (u)	\$ 4.40	\$ 8.80
Patch Cord 3 (m)	1 (u)	\$ 3.00	\$ 3.00
Mano de Obra	10 (h)	\$ 30.00	\$ 300.00
		<b>TOTAL</b>	<b>\$ 389.80</b>

## 4 CONCLUSIONES

- Al ser el desempeño, portabilidad y economía, los requerimientos más sobresalientes, se concluyó que el uso del sistema de *NEMS Nagios*, en conjunto con el dispositivo *Raspberry Pi*, permiten cumplir con dichos requerimientos sin tener que sobredimensionar o sobrestimar la expectativa de gastos a considerar por su versatilidad y compatibilidad entre el sistema y el dispositivo.
- De realizar una adecuada comparación entre las diferentes versiones que dispone el dispositivo *Raspberry Pi*, se concluyó que el modelo 3B+ del dispositivo *Raspberry Pi* logra establecer un equilibrio en la relación costo beneficio, pues además de tener alta disponibilidad en el mercado, también cumple con los requerimientos mínimos para la ejecución del sistema *NEMS Nagios*, según lo que recomienda el desarrollador.
- Se concluyó que un trabajo eficiente de monitoreo de dispositivos en red, consiste también en estandarizar en la manera de lo posible los servicios que serán monitoreados en cada *host* considerando la incidencia que tienen las variaciones de un determinado parámetro en el comportamiento de un *host* como la carga de la *CPU* o el uso del disco en el sistema para el caso de ordenadores y servidores, o el estado de los consumibles más importantes en el caso del *host* de tipo impresora.
- Se concluyó que el uso de agentes de monitoreo como *NSClient++* y *NRPE* requiere de cierto grado de conocimiento en el manejo de *Linux* y de los comandos utilizados en este sistema operativo, pero a su vez facilitan en gran manera el proceso de configuración para el monitoreo de diferentes dispositivos.
- La integración del *software NEMS Linux* con el dispositivo *Raspberry Pi* permiten de manera eficiente la implementación de un sistema de monitoreo para dispositivos dentro de una red sin la necesidad de realizar inversiones altas en la adquisición de servidores, y ofreciendo una potente solución portátil para quienes se dedican al monitoreo de redes.
- Se concluyó que el sistema *NEMS Nagios* es completamente amigable con el usuario final puesto que ofrece más de una forma de entregar reportes tales como ventanas optimizadas para dispositivos móviles o ventanas con alertas audibles y narración de eventos como es el caso de *NEMS Tactical overview*.

## 5 RECOMENDACIONES

- Se recomienda realizar la reserva de la dirección *IP* del servidor *NEMS*, así como de los dispositivos a monitorearse, en el *router* de la red a fin de evitar conflictos de comunicación o direcciones duplicadas por conflictos con el servicio de *DHCP*.
- Se recomienda realizar una inspección adecuada del lugar y condiciones donde será implementado el prototipo, a fin de garantizar la integridad del mismo durante el tiempo de funcionamiento.
- Se recomienda el uso de fuentes de alimentación compatibles no genéricas para evitar posibles daños en el dispositivo a cause de fallos electrónicos, así como el uso de un dispositivo de protección contra voltaje como reguladores o *UPS*.
- Se recomienda realizar un análisis previo de las condiciones de trabajo de los *hosts* a ser monitoreados de forma que se puedan establecer de manera adecuada los umbrales de alerta tanto de advertencia como críticos.}
- Se recomienda una vez configurado el sistema y dispositivo, realizar un *backup* de la configuración a fin de disponer de ella, en caso de fallos tras una nueva configuración o actualización del sistema.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Nagios Enterprises, «Exchange Nagios,» 18 marzo 2020. [En línea]. Available: <https://exchange.nagios.org/directory/Addons/Monitoring-Agents/NRPE--2D-Nagios-Remote-Plugin-Executor/details#:~:text=NRPE%20allows%20you%20to%20remotely,remote%20Windows%20machines%20as%20well..> [Último acceso: 2022 julio 14].
- [2] A. Delgado, «GEEKNETIC,» 21 11 2022. [En línea]. Available: <https://www.geeknetic.es/Raspberry-Pi/que-es-y-para-que-sirve>. [Último acceso: 18 05 2022].
- [3] C. Inc, «CISCO,» 2022. [En línea]. Available: [https://www.cisco.com/c/es\\_mx/solutions/automation/what-is-network-monitoring.html](https://www.cisco.com/c/es_mx/solutions/automation/what-is-network-monitoring.html). [Último acceso: 18 05 2022].

- [4] R. Ferguson, «NEMS Linux,» NEMS org, [En línea]. Available: <https://nemslinux.com/download/nagios-for-raspberry-pi-4.php>. [Último acceso: 18 05 2022].
- [5] Progress Whatsup Gold, «Progress Whatsup Gold,» [En línea]. Available: [https://www.whatsupgold.com/es/snmp#:~:text=Simple%20Network%20Management%20Protocol%20\(SNMP,cualquier%20dispositivo%20habilitado%20para%20SNMP..](https://www.whatsupgold.com/es/snmp#:~:text=Simple%20Network%20Management%20Protocol%20(SNMP,cualquier%20dispositivo%20habilitado%20para%20SNMP..) [Último acceso: 14 julio 2022].
- [6] C. Melantuche, «No Solo Hacking,» 22 noviembre 2020. [En línea]. Available: <https://www.nosolohacking.info/nagios-nrpe-introduccion-e-instalacion/#:~:text=Nagios%20NRPE%20nos%20permite%20ejecutar,nagios%20para%20que%20lo%20represente..> [Último acceso: 14 julio 2022].
- [7] DigiCert, «DIGICERT,» [En línea]. Available: [https://www.websecurity.digicert.com/es/es/security-topics/what-is-ssl-tls-https#:~:text=sitio%20instale%20SSL,-,El%20protocolo%20TLS%20\(Transport%20Layer%20Security%2C%20seguridad%20de%20la%20capa,transferencia%20de%20datos%20e%20informaci%C3%B3n..](https://www.websecurity.digicert.com/es/es/security-topics/what-is-ssl-tls-https#:~:text=sitio%20instale%20SSL,-,El%20protocolo%20TLS%20(Transport%20Layer%20Security%2C%20seguridad%20de%20la%20capa,transferencia%20de%20datos%20e%20informaci%C3%B3n..) [Último acceso: 14 julio 2022].
- [8] SYNnex WESTCON-COMSTOR, «Blog SYNnex Westcan-Comstor,» 21 septiembre 2016. [En línea]. Available: <https://digital.la.synnex.com/la-importancia-del-monitoreo-de-la-red-y-analytics-avanzado#:~:text=Ese%20monitoreo%20permite%20obtener%20informaci%C3%B3n,ajustar%20y%20expandir%20la%20red..> [Último acceso: 14 julio 2022].
- [9] Trust Radius, «TrustRadius,» [En línea]. Available: <https://www.trustradius.com/compare-products/nagios-core-vs-observium>. [Último acceso: 2022 julio 14].
- [10] 330ohms, «330ohms,» 21 noviembre 2019. [En línea]. Available: <https://blog.330ohms.com/2019/11/21/retropie-raspberry-pi-4-vs-pi-3b-realmente-es-mejor/>. [Último acceso: 2022 julio 14].
- [11] A. Williams, «Trusted Reviews,» 29 febrero 2016. [En línea]. Available: <https://www.trustedreviews.com/opinion/raspberry-pi-3-vs-pi-2-2936374>. [Último acceso: 14 julio 2022].



- [12] R. Pi, «Raspberry Pi,» Raspberry Pi, 14 03 2018. [En línea]. Available: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>. [Último acceso: 18 05 2022].
- [13] «NSClient++,» 2006. [En línea]. Available: <https://nsclient.org/>. [Último acceso: 13 06 2022].
- [14] M. Meseguer, «Sobrebits,» 21 junio 2018. [En línea]. Available: <https://sobrebits.com/plugin-de-nagios-con-powershell-parte-2-instalacion-de-nsclient/>. [Último acceso: 14 julio 2022].
- [15] R. Kumar, «TecAdmin,» 4 junio 2021. [En línea]. Available: <https://tecadmin.net/how-to-install-nrpe-on-ubuntu-20-04/>. [Último acceso: 14 julio 2022].
- [16] Linuxsysymas, «Linuxsysymas,» 06 mayo 2015. [En línea]. Available: <https://linuxsysymas.wordpress.com/2015/05/06/nagios-server-windows/>. [Último acceso: 14 julio 2022].
- [17] RICOH COMPANY LTD, «Ricoh Private MIB Specification Part 4,» TOKIO, 2012.

## **7 ANEXOS**

La lista de los Anexos se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces

ANEXO III. Conjunto de datos extensos

## **ANEXO I: Certificado de Originalidad**

Quito, D.M. 23 de agosto de 2022

De mi consideración:

Yo, **LEANDRO ANTONIO PAZMIÑO ORTIZ**, en calidad de Director del Trabajo de Integración Curricular titulado: **IMPLEMENTACIÓN DE PROTOTIPOS DE SISTEMAS DE MONITOREO DE REDES EMPLEANDO DISTINTAS SOLUCIONES A NIVEL DE SOFTWARE, COMPONENTE: IMPLEMENTACIÓN DEL PROTOTIPO EMPLEANDO NEMS SOBRE UN RASPBERRY PI** elaborado por el estudiante **BRYAN ANDRÉS ENRIQUEZ AUZ** de la carrera **TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES**, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo producto del Trabajo de Integración Curricular indicado.

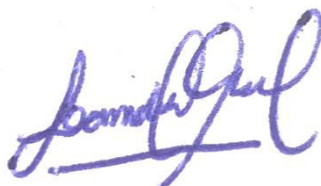
El documento escrito tiene un índice de similitud del 12 %.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

[Reporte Turnitin - BEnriquez.pdf](#)

Atentamente,



---

**DIRECTOR**

Ing. Leandro Antonio Pazmiño Ortiz, MSc.

**Docente Ocasional**

**ESFOT**

## ANEXO II: Enlaces

Código QR de la implementación y pruebas de funcionamiento del prototipo



Código QR del mantenimiento del prototipo

