



**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE  
COMUNICACIÓN**

**ARTÍCULO CIENTÍFICO**

**TEMA:**

**“SISTEMA DE MONITOREO DE TEMPERATURA, HUMEDAD Y  
DETECCIÓN DE HUMO PARA EL CENTRO DE DATOS UBICADO EN LA  
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS BASADO EN EL  
ESTÁNDAR ISO/IEC 17799.”**

**AUTOR: LUIS TARQUINO CAMUÉS NARVÁEZ**

**DIRECTOR: MSC. EDGAR ALBERTO MAYA OLALLA**

**Ibarra-Ecuador**

**2017**

# Sistema de monitoreo de temperatura, humedad y detección de humo para el Centro de Datos ubicado en la Facultad de Ingeniería en Ciencias Aplicadas basado en el estándar ISO/IEC 17799

*Autores – Luis Tarquino CAMUÉS NARVÁEZ, Ing Edgar Alberto MAYA OLALLA, MSc.*

Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Avenida 17 de Julio 5-21 y José María Córdova, Ibarra, Imbabura

[lcamuesn@utn.edu.ec](mailto:lcamuesn@utn.edu.ec), [eamaya@utn.edu.ec](mailto:eamaya@utn.edu.ec)

**Resumen.** *EL presente proyecto consiste, en el diseño e implementación de un sistema de monitoreo de temperatura, humedad y humo, para el Centro de Datos, ubicado en la Facultad de Ingeniería en Ciencias Aplicadas, basado en la norma para la seguridad de la información ISO/IEC 17799, la que proporciona recomendaciones de las mejores prácticas, en la gestión de la información. El sistema está conformado, de una parte, hardware para el monitoreo mediante sensores y software de gestión, para la observación en una interfaz web, de los valores obtenidos por los sensores. De igual manera, el sistema de gestión cumple el rol de alertar al administrador, mediante correo electrónico, sobre las variaciones en los valores de temperatura, humedad y humo detectados en el Centro de Datos.*

*Al inicio, se presenta la base teórica, que promueve la realización del trabajo de investigación, convirtiendo una problemática del Centro de Datos, en oportunidad de mejorar las prestaciones del mismo, en donde se definen los objetivos a cumplirse, así como sus limitaciones en alcance, también se expone el fundamento en cuanto al método utilizado, para la comunicación entre el software y hardware, de igual manera se describe una comparativa entre sistemas de gestión, para seleccionar la más adecuada, y se utiliza la norma IEEE 29148 para analizar los requerimientos, posterior se detallan las características de los componentes de hardware a utilizarse.*

*A continuación se inicia el diseño del sistema de monitoreo, con la verificación de funcionamiento de todos los componentes, de igual manera se realizan las instalaciones, de todos los paquetes de software necesarios para el sistema, conjuntamente con sus configuraciones, en la parte final se realiza el acoplamiento de software y hardware para la validación de su funcionamiento, posteriormente se realiza todo el proceso de implementación del sistema de monitoreo dentro del Centro de Datos, en las ubicaciones definidas en el capítulo uno, y al finalizar se realizan todas las pruebas necesarias, para evaluar el rendimiento del sistema de monitoreo, a la vez, realizar el análisis de los registros de los datos obtenidos, para plantear las conclusiones y recomendaciones, que permitan las mejoras en el Centro de Datos.*

## Palabras Claves

Sistema de monitoreo, sensores, Temperatura, humedad, humo, Software de gestión, Sistema embebido, Centro de Datos.

**Abstract.** *This research was about the design and implementation of a temperature, humidity and smoke monitoring system, for the Data Center, located in the Faculty of Engineering in Applied Sciences, based on the standard for information security ISO / IEC 17799, which provides recommendations of best practices in information management. The system consists, on the one hand, hardware for the monitoring by means of sensors and management software, for the observation in a web interface, of the values obtained by the sensors. Likewise, the management system plays the role of alerting the administrator, through e-mail, about the variations in temperature, humidity and smoke values detected in the Data Center.*

*At the beginning, it presents the theoretical basis, which promotes the realization of the research work, converting a data center problem, in an opportunity to improve the performance of the same, which defines the objectives to be fulfilled, as well as its limitations in The scope, also the basis in the method used, for the communication between the software and the hardware, the same way a comparative between management systems is described, to select the most appropriate, and IEEE 29148 is used to analyze The requirements, after the characteristics of the hardware components to be used.*

*A continuation of the design of the monitoring system, with the verification of the functioning of all the components, in the same way the installations are made of all the software packages necessary for the system, together with their configurations, in the final part It carries out the software and hardware coupling for the validation of its operation, the entire process of implementing the monitoring system within the Data Center, and the locations defined in chapter one, and all the necessary tests are carried out. To evaluate the performance of the monitoring system, at the same time, to carry out the analysis of the records of the obtained data,*

to propose the conclusions and recommendations, that allow the improvements in the Data Center.

## Keywords

Monitoring system, sensors, Temperature, humidity, smoke, Management software, Embedded system, Data center

## 1. Introducción

La implementación de nuevos Centro de Datos amerita el monitoreo de las condiciones ambientales que se tiene en el interior de los mismos, con el fin de evitar cambios bruscos en temperatura, humedad o detección de humo. Por tal motivo es necesario la realización del sistema de monitoreo, para que alerte al administrador sobre las condiciones en que se encuentra el Centro de Datos mediante alertas por correo electrónico y así tome medidas de prevención contra daños.

## 2. Conceptos Básicos

### 2.1 Centro de Datos

Centro de Datos es el lugar de procesamiento de datos en donde se alojan los recursos necesarios para el procesamiento de información, debiendo ser concebido de manera que brinde los servicios de forma continua y segura, operando eficientemente en su capacidad normal y que problemas internos o externos no afecten su rendimiento.

### 2.2 Estandar ISO IEC 17799

El estándar "ISO (Organización Internacional de Estandarización) e IEC (Comisión Electrotécnica Internacional) forman el sistema especializado para la estandarización mundial. El estándar tiene por objetivo, ofrecer guías y principios para iniciar, implementar, mantener y mejorar la administración de la seguridad de la información de una organización. Los objetivos proporcionan, una directriz, sobre cómo alcanzar las mejores prácticas para un mejor desempeño en funcionamiento de equipos y procesamiento de la información. [1]

### 2.3 Gestion en internet

En agosto de 1988, aparecieron las primeras recomendaciones del SNMP, así como de la SMI y MIB correspondientes. [2]

El protocolo SNMP, tuvo una gran acogida en la comunidad y se puede decir que el marco de trabajo de protocolo SNMP se basa en tres documentos.

*Structure of Management Information (SMI): RFC1155.*

*Management Information Base (MIB): RFC 1156, RFC 1213.*

*Simple Network Management Protocol (SNMP): RFC 1157.*

### 2.4 Sistemas de Gestion

Un sistema de gestión de red se compone de hardware y software implementados entre los componentes de red existentes, el software que se emplea en las tareas de gestión de red reside: en el host, en los procesadores de comunicaciones y controladores de grupos terminales. Un sistema de gestión de red está diseñado para ver toda la red como una arquitectura unificada con direcciones y etiquetas asignadas a cada punto, atributos específicos de cada elemento y enlaces conocidos por el sistema. Los elementos activos de la red proporcionan una retroalimentación regular de la información de estado al centro de control de red. [3]

#### 2.4.1 Nagios

Nagios supervisa toda la infraestructura de TI para garantizar que los sistemas, las aplicaciones, los servicios y los procesos empresariales funcionen correctamente. En caso de un fallo, Nagios puede alertar al personal técnico del problema, permitiéndoles comenzar procesos de remediación antes de que las interrupciones afecten a los procesos de negocio, usuarios finales o clientes. Con Nagios, nunca se le dejará tener que explicar por qué una interrupción invisible de la infraestructura dañó la línea de fondo de su organización. [4]

### 2.5 Open Nebula

OpenNebula ofrece una solución sencilla, pero rica en funciones y flexible para la gestión integral de los centros de datos virtualizados. La interoperabilidad de OpenNebula hace que la nube sea una evolución aprovechando los activos de TI existentes, protegiendo las inversiones y evitando el bloqueo de proveedores. OpenNebula es una solución para la empresa que incluye todas las características necesarias para ofrecer una oferta en la nube (privada) y para ofrecer servicios en la nube pública. [5]

### 2.6 Sistemas embebidos

Son plataformas electrónicas de código abierto basado en hardware y software fácil de usar. Un sistema embebido es un sistema electrónico que cumple solo determinadas

funciones. Su hardware no es modificable, su software sí. Entre las placas que más se destacan son:

RASPBERRY, INTEL y ARDUINO

### 2.6.1 Potenciometro



Figura 1. Esquema del potenciometro. Fuente: [6]

Un potenciometro posee la resistencia o resistor variable mecánica con cursor y de al menos tres terminales. Conectando los terminales extremos a la diferencia de potencial a regular control de tensión, se obtiene entre el terminal central (cursor) y uno de los extremos una fracción de la diferencia de potencial total se comporta como un divisor de tensión o voltaje.

### 2.6.2 Pantalla LCD

El (Liquid Crystal Display) o pantalla de cristal líquido es un componente empleado para la visualización de información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo. Está dirigido por un microcontrolador el cual maneja todo su funcionamiento



Figura 2. Pantalla LCD. Fuente: [7]

### 2.6.3 Sensores DHT

El DHT es un sensor de temperatura y humedad. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos. Es simple de usar, pero requiere sincronización para tomar datos.

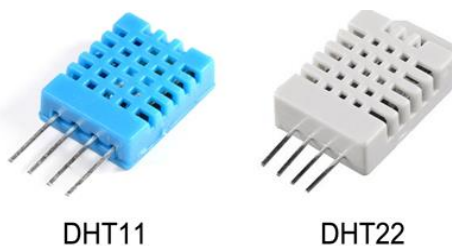


Figura 3. Esquema del potenciometro. Fuente: [8]

### 2.6.4 Sensores de humo MQ

Son sensores muy sencillos de usar, pudiendo detectar diferentes tipos de gases incluido el humo, los sensores son de alta sensibilidad y con un tiempo de respuesta rápido, su salida es una resistencia analógica. Estos sensores son electroquímicos y cuando son expuestos algún tipo de gas varia su resistencia, internamente lleva un calentador el cual aumenta la temperatura interna y por tal motivo el sensor puede reaccionar con distintos tipos de gases entre ellos el humo, provocando una variación en el valor de la resistencia. . [9]



Figura 4. Sensores de humo MQ Fuente: [10]

### Rangos de temperatura y humedad TIA 942

En la norma TIA 942 pagina 34 publicada en el año 2005, se establece como parámetros de operación de un Centro de Datos, un rango de temperatura de 20 °C a 25 °C y una humedad relativa de 40 % a 55 %, información que será relevante al momento de configurar los rangos admitidos por el sistema de monitoreo.

## 3. Desarrollo del sistema de monitoreo

En este capítulo se realiza el desarrollo del sistema de monitoreo, iniciando con las respectivas pruebas y conexiones de los sensores al sistema embebido, desarrollo de códigos, configuraciones y acoplamiento de hardware y software hasta la fase de preinstalación.

### 3.1 Selección de placa

Dentro de los modelos de placa mencionados, las placas Raspberry e Intel satisfacen las necesidades del proyecto sin embargo cuentan con periféricos adicionales como audio y video que no son necesarios y por tal motivo son de mayor costo, en tal virtud la placa adecuada que cumple los requisitos mínimos como pines de entrada, salida y puerto ethernet es la placa arduino, así como la facilidad de programación, ameritan la selección de la mencionada plataforma.

#### 3.1.1 Selección de placa Arduino

En la gama de arduino solo existen 2 placas que traen incorporado un puerto ethernet, a excepción de la shield ethernet que puede acoplarse con varias placas arduino, estándose funcionalidades en algunos pines a las mismas, Estas dos placas son el arduino ethernet y el arduino YUN, Sin embargo solo el arduino YUN cuenta con un microprocesador linux en el que se puede trabajar con el paquete snmpd, esencial en la realización del sistema de monitoreo por tal razón, es la placa seleccionada para el sistema de monitoreo.

#### 3.1.2 Arduino YUN

El Arduino Yun es una placa electrónica basada en el ATmega32u4 y el Atheros AR9331. El procesador Atheros compatible con una distribución Linux basada en OpenWrt llamado OpenWrt-Yun. La placa se ha incorporado con Ethernet y soporte WiFi, un puerto USB-A, ranura para tarjeta microSD, 20 entradas digitales / pines de salida (de los cuales 7 se pueden utilizar como salidas PWM y 12 como entradas analógicas), un cristal de 16 MHz oscilador, una conexión micro USB, una cabecera ICSP y un 3 botones de reinicio. [11]

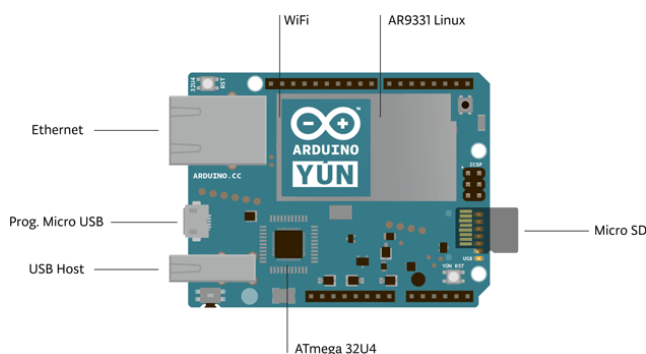


Figura 5. Placa Arduino YUN  
Fuente: [12]

Pines digitales I/O	20
Canales PWM	7
Pines de Entrada Analógica	12
Corriente DC por I/O Pin	40 mA
Corriente DC por 3.3V Pin	50 mA
Memoria Flash	32 KB (of which 4 KB used by bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Tabla1. Microcontrolador Atmega

Fuente: [13]

Procesador	Atheros AR9331
Arquitectura	MIPS @400MHz
Voltaje de Operación	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host
Lector de tarjeta	Micro-SD
RAM	64 MB DDR2
Flash Memoria	16 MB
SRAM	2.5 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Tabla2. Microprocesador Linux

Fuente: [14]

#### 3.1.3 IDE de Arduino

El entorno de desarrollo integrado es el software que permite crear el programa para dar el funcionamiento del sistema embebido.

Microcontrolador	ATmega32U4
Voltaje de operación	5V

### 3.2 Diagrama de bloques del sistema de monitoreo

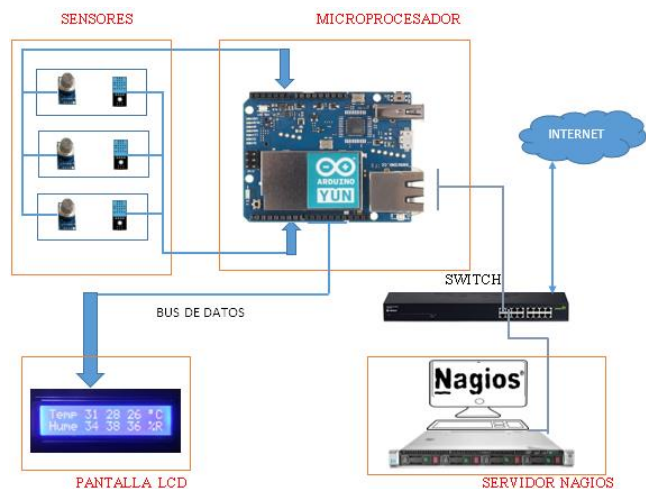


Figura 6. Diagrama del sistema de monitoreo  
Fuente: Propia

En el diagrama de bloques, podemos observar la secuencia del funcionamiento del sistema de monitoreo, desde la recopilación de datos análogo-digitales en los sensores, datos que son recopilados por el arduino y mostrados en una pantalla LCD, solo de temperatura y humedad, paralelamente aquellos datos el arduino los envía al sistema de gestión para ser desplegados mediante una interfaz web, y en base a los datos alertar por correo electrónico al administrador.

### 3.3 Escritura en OPENWRT-YUN desde el código de arduino

Es necesario comprender que el arduino YUN, está compuesto por dos procesadores como se muestra en la figura 11, el procesador ATMEGA y el procesador ATHEROS, para que estos dos procesadores puedan comunicarse, es necesario que en el código se incluya la librería bridge, que funciona como puente de comunicación entre los dos.

Mediante el siguiente código, se muestra un ejemplo, de cómo escribir desde el procesador ATMEGA al procesador ATHEROS. En el cual se incluye aparte de la librería BRIDGE, la librería <FileIO.h>, para el ejemplo se escribirá un archivo de nombre “script.sh” en la dirección /tmp.

```
#include <FileIO.h> // incluye la libreria de archivos
void setup() {
    Bridge.begin(); // iniciamos la comunicacion AVR -
    LININO
    FileSystem.begin(); // iniciamos la libreria de archivos
}
void loop() {
```

```
File script =
FileSystem.open("/tmp/script.sh",FILE_WRITE);
    script.print("#!/bin/sh\n"); // declaramos al archivo
de tipo shell
    script.print("prueba "); // el archivo contendrá la
palabra prueba
    script.close(); // cerramos el archivo
}
```

### 3.4 Armado del circuito

El armado del prototipo, se lo realiza utilizando todos los componentes que conformaran el sistema de monitoreo, en la figura 7 se puede observar la conexión general de hardware, realizado en el programa Proteus basándose en las conexiones descritas.

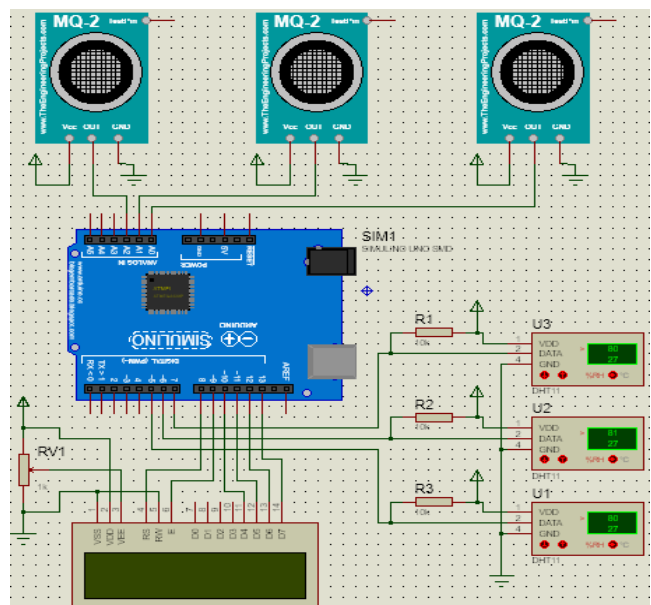


Figura 7. Armado del circuito  
Fuente: Propia

Para protección del circuito se ubicará un fusible en la entrada de alimentación de energía, por lo cual se calcula la corriente necesaria que consumirá el sistema.

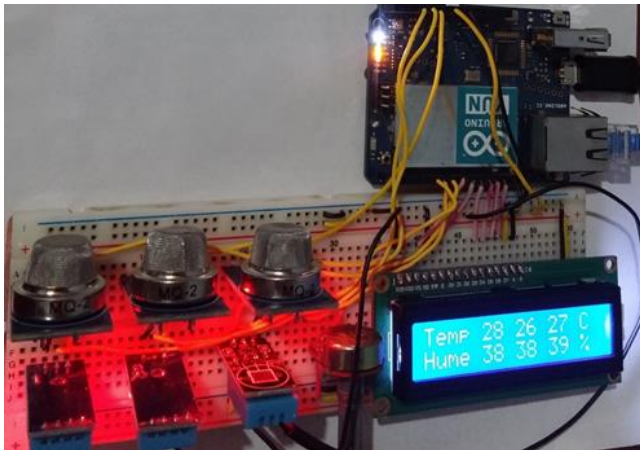
Componente	Corriente consumo	Cantidad	Total
Arduino	260 mA	1	260 mA
Dht11	3.9 mA	3	11.7 mA
MQ-2	137 mA	3	411 mA
LCD	20 mA	1	20 mA
total			702.7 mA

Tabla3. Determinación del valor del fusible

Fuente: Propia

### 3.5 Montaje del circuito físico

En la figura 8 se puede observar el circuito montado en un protoboard para las pruebas de funcionamiento del código, como lo es, la lectura de temperatura, humedad y humo, los valores mostrados en la LCD corresponden a la posición de los sensores correspondientemente.



**Figura 8.** Armado del prototipo del sistema  
Fuente: Propia

### 3.6 Instalacion y configuración del agente SNMPD en Arduino

Encendemos el Arduino y nos conectamos a su red inalámbrica, verificar que la interfaz de red se encuentre conectada a un router con acceso a internet. Desde la máquina que tiene acceso a la red inalámbrica del Arduino, accedemos mediante el protocolo SSH al mini-servidor LININO. Pudiendo acceder desde PUTTY o desde linux con el comando.

```
ssh -p 22 root@192.168.240.1
```

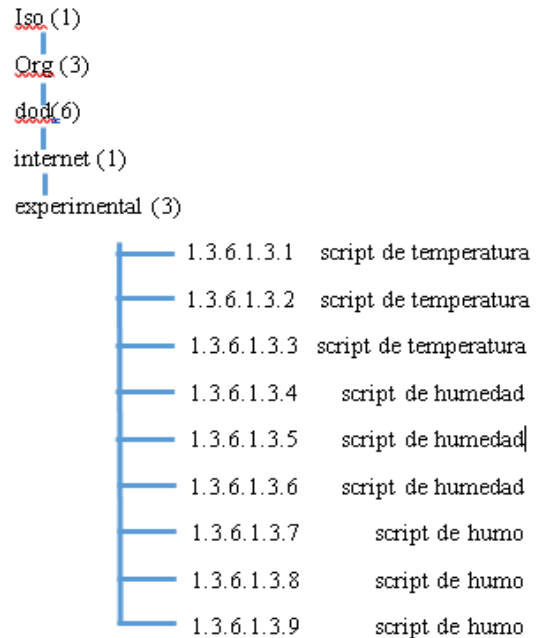
Nos autenticamos con las credenciales y procedemos a la instalación de los paquetes

Como Las MIB son las bases de datos establecidas por los fabricantes, no se pueden utilizar los mismos OID, por tal motivo se emplean en el código los OID utilizados para fines experimentales que forma parte de la estructura SMI. Ubicado en el árbol de identificadores mediante la rama.

- 1 iso
- 1.3 org
- 1.3.6 dod
- 1.3.6.1 internet
- 1.3.6.1.3 experimental

Partiendo de esa rama podemos ubicar varias subdirecciones, existiendo una que puede almacenar variables de tipo string Quedando el identificador

1.3.6.1.3.x.101.1, siendo la x el número correspondiente a cada sensor, de esa manera, se almacena el valor escrito desde el código de arduino en el OID correspondiente a un archivo, el árbol resultante se lo muestra en la figura9.



**Figura 9.** Arbol resultante sistema de monitoreo  
Fuente: Propia

### 3.7 Instalacion y configuración de archivos en Nagios Core 4.2.4

La instalación del sistema de gestión nagios 4.2.4, se lo realiza conjuntamente con la instalación del paquete snmpd en la máquina virtual alojada en el CLOUD OpenNebula, realizado como proyecto de tesis en la facultad FICA.

En la tabla 11 se muestra los umbrales de temperatura y humedad a configurar en el archivo de nagios de manera permanente tras las pruebas realizadas durante varios días de funcionamiento.

Parámetro	Warning	Critical
Temperatura	20 °C a 27 °C	18 °C a 29 °C
Humedad	32% a 55% HR	30% a 60% HR
Humo	0 a 400 ppm	0 a 500 ppm
Nota:	Estos valores pueden ser modificables después de la instalación y adaptación de los sensores al ambiente interno del Centro de Datos.	

**Tabla3.** Rangos de activación de ALARMAS

Fuente: Propia

### 3.7.1 Visualización de servicios monitoreados en interfaz web

Iniciamos el servicio httpd, nagios y en un navegador web accedemos a la dirección 127.0.0.1/nagios, nos autenticamos con las credenciales y en la sección servicios, se desplegará la información requerida.

Para verificar el historial del host tenemos que dar permisos de lectura escritura y ejecución al archivo nagios.log, lo hacemos con el siguiente comando.

```
chmod 607 /usr/local/nagios/var/nagios.log
```

En la figura 28 se puede observar la interfaz web en donde se muestran los distintos servicios monitoreados.

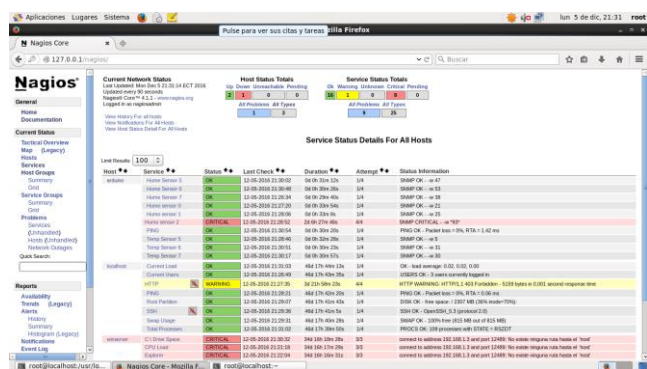


Figura 10. Arbol resultante sistema de monitoreo  
Fuente: Propia

### 3.7.2 Nagiosgraph

Nagiosgraph es un complemento de código libre, desarrollado por sourceforge, para graficar los valores obtenidos por el sistema de gestión Nagios, en el que se puede visualizar mediante línea del tiempo por horas, días o meses los resultados de monitoreo.

### 3.7.3 Instalacion y configuración del servicio de correo

Iniciamos con la instalación del servicio de correo postfix y lo instalamos con el comando yum

```
yum -y install postfix
```

Ingresamos al directorio /etc/postfix y editamos el archivo main.cf

Para el ejemplo, utilizamos una cuenta de correo Gmail debido a que es un correo seguro e ingresamos el servidor y el puerto a usar y la configuración TLS requerida.

```
relayhost=[smtp.gmail.com]:587  
activamos la autenticación sasl  
smtp_sasl_auth_enable=yes  
smtp_sasl_password_maps=hash:/etc/postfix/sasl_passwd  
smtp_sasl_security_options=noanonymous
```

```
smtp_tls_CAfile=/etc/postfix/cacert.pem  
smtp_use_tls=yes
```

A continuación, generamos el fichero vi /etc/postfix/sasl\_passwd

En el agregamos las siguientes lineas en donde la palabra “contraseña” hace referencia a la contraseña de la cuenta. [smtp.gmail.com]:587 ltctonny@gmail.com:contraseña

```
Damos permisos de lectura y escritura al archivo  
chmod 600 /etc/postfix/sasl_passwd  
postmap /etc/postfix/sasl_passwd  
chmod 600 /etc/postfix/sasl_passwd.db
```

Copiamos el contenido del certificado Equifax\_Secure\_CA.pem a cacert.pem  
cat /etc/ssl/certs/Equifax\_Secure\_CA.pem | tee -a /etc/postfix/cacert.pem

Reiniciamos el servicio service postfix restart

Para probar el envío de correos realizamos una prueba mediante línea de comandos echo “mensaje de prueba” | mail -s “asunto prueba” ltctonny@gmail.com

Si el correo se envió y receipto satisfactoriamente todo está bien, caso contrario verificar DNS y restricciones del correo. Agregar el contacto en el archivo de nagios, al que va a redirigir las alertas del correo.

```
vim /usr/local/nagios/etc/contacts.cfg  
email nagios@localhost, ltctonny@gmail.com
```

Procedemos a reiniciar el servicio service nagios restart

En la figura 11, se puede apreciar el servicio de notificaciones por parte del sistema Nagios, en la cual se indica, que en el rack izquierdo se ha detectado un valor de 37% de humedad relativa, lo que está por debajo del valor definido en el archivo de configuración

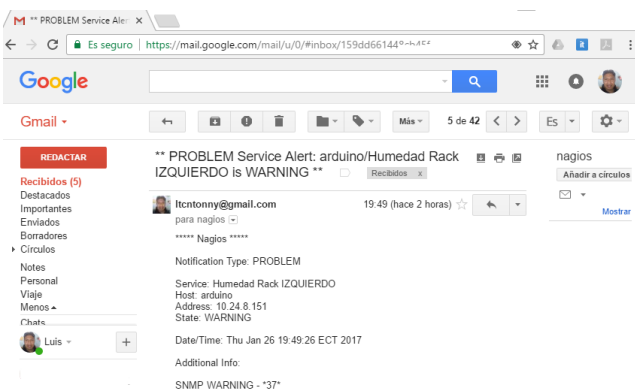
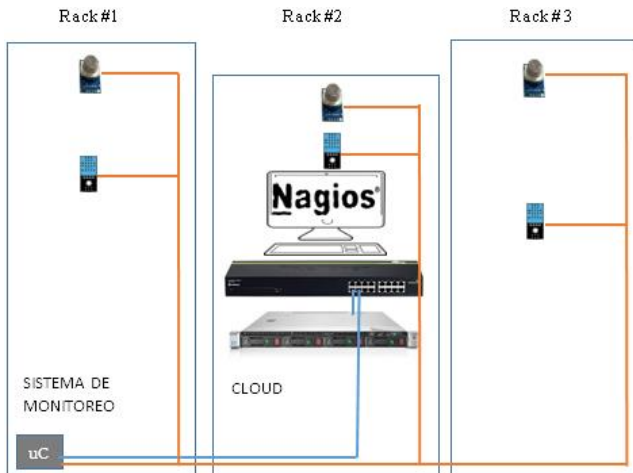


Figura 11. Alertas de Nagios por correo  
Fuente: Propia



## 4. Desarrollo del software.

En la figura 12, se puede observar la arquitectura de la conexión mediante cable de red entre arduino y servidor nagios (cloud), también se puede visualizar la ubicación de los sensores que son en la puerta trasera de cada rack.



**Figura 12** Arquitectura del sistema.  
Fuente: Propia.

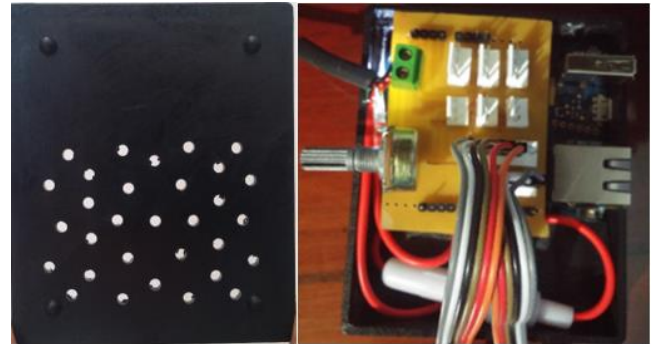
### 4.1 Proceso de armado e implementación.



**Figura 13** Armado de placa que encaja en el arduino YUN.  
Fuente: Propia.

En la figura 13, se muestra la placa del circuito que encaja en el Arduino Yun, sirve como base sólida para la ubicación de los cables que conecta a los sensores en cada uno de los racks, de igual forma tiene un potenciómetro que sirve para la calibración de luminosidad de la pantalla LCD.

A continuación, en la figura 14, se realiza la elaboración de la caja que va a contener dentro el sistema embebido placa y cables conectores. En la base donde reposara la placa, se ha realizado perforaciones, para evitar el calentamiento excesivo del arduino debido al procesamiento de información, también se incluye el fusible para protección del circuito.



**Figura 14** Colocacion del Arduino en la caja.  
Fuente: Propia.

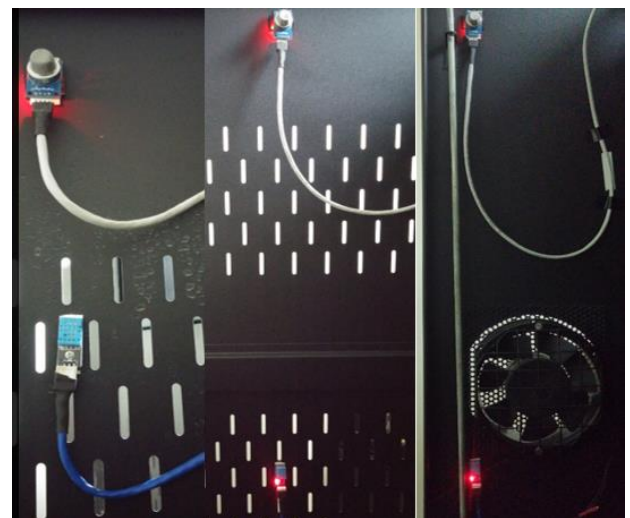
En la figura 15, se realiza la conexión del cableado de sensores, cable de red, cable de energía a la placa arduino



**Figura 15** Colocacion del Arduino en la caja.  
Fuente: Propia

### 4.2 Ubicación de los sensores en los RACK.

Los sensores son ubicados en la puerta de atrás de cada rack, en la figura 16, se puede visualizar la colocación de los sensores de temperatura, humedad y humo.



**Figura 16** Colocación de los sensores en los RACK.  
Fuente: Propia.

Finalizando el proceso de implementación se ubica el cableado de la pantalla LCD desde el interior de Centro de Datos, hacia el exterior, pasándolo por el cielo falso hacia el pasillo de la facultad, e ingresando a la oficina, para terminar cerca del lector de huellas, en total se utilizó 30 metros de cable, dos cables de red de 15 metros, que funcionan como bus de datos debido a la cantidad de cables para a conexión entre Arduino-LCD.

La figura 17, indica los valores de temperatura y humedad en los rack izquierdo, central y derecho correspondientemente tanto en la ubicación de los sensores por rack a los valores mostrados en la pantalla LCD. La pantalla LCD se ubica a lado derecho de la puerta de acceso al Centro de Datos.



Figura 17 Ubicación de la pantalla LCD. Fuente: Propia.

### 4.3 Ingreso al sistema de monitoreo.

Para poder ingresar a la interfaz gráfica que muestran los distintos valores que son obtenidos con los sensores, se puede acceder desde fuera de la universidad (internet) y dentro de la universidad (intranet), a continuación, se muestra el procedimiento desde internet.

#### 4.3.1 Ingreso al sistema desde internet.

El procedimiento se lo debe realizar desde internet, para acceder a la plataforma cloud y a la máquina virtual, en donde está instalado el sistema de monitoreo. En un navegador ingresamos a la dirección 190.95.196.221:9869, donde se mostrará la interfaz en la que debemos ingresar nuestras credenciales de administrador de la cuenta en el cloud, mostrada en la figura 18.

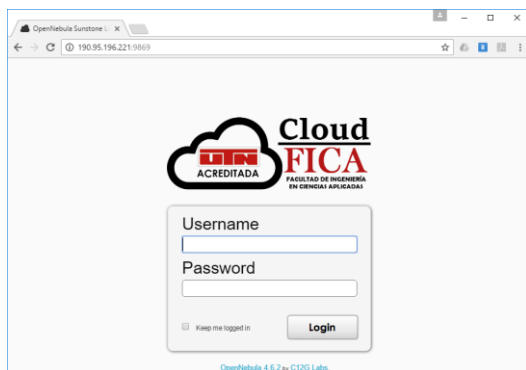


Figura 18 Ingreso a la plataforma CLOUD. Fuente: Propia

Una vez dentro del sistema en un navegador escribimos la dirección 127.0.0.1/nagios en donde es necesario autenticarse con las credenciales, y se desplegara la opción de inicio de la pagina, en el panel izquierdo presionar el enlace que lleva a servicios y mostrara los servicios monitoreados como lo indica la figura 19.

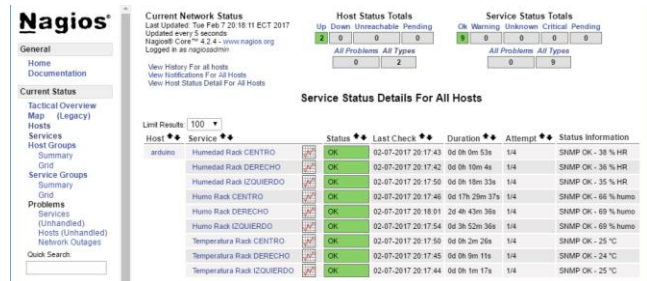


Figura 19 Interfaz de monitoreo Nagios Core. Fuente: Propia.

#### 4.3.2 Visualización de datos registrados mediante graficos.

Cada indicador está configurado en los archivos de nagios, los que permiten mostrar los valores registrados de manera gráfica, para observar el registro de valores es necesario pulsar en el grafico correspondiente a cada sensor como se indica la flecha en la figura 20.

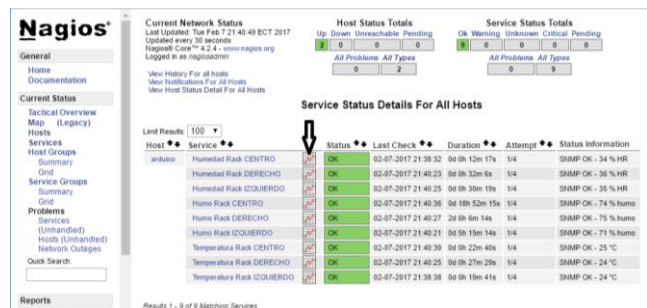
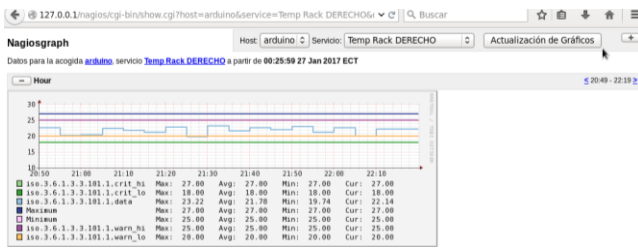


Figura 20 Interfaz de monitoreo Nagios Core. Fuente: Propia.

Color de línea	representación
azul	límite máximo del estado critical
morado	límite máximo del estado warning
celeste	valor correspondiente a la variable monitoreada
tomate	límite mínimo del estado warning
verde	límite mínimo del estado critical

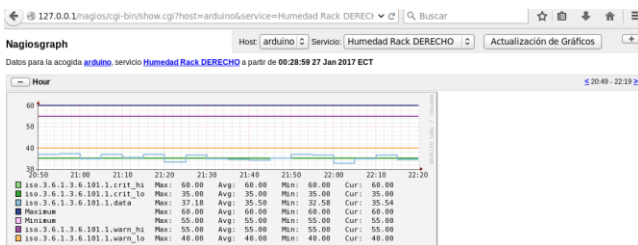
Tabla3. Representación del color en las líneas de los graficos

Fuente: Propia



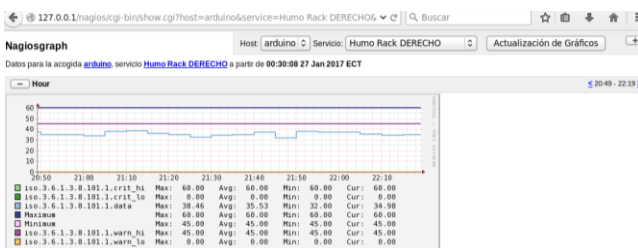
**Figura 21** Grafico de temperatura Rack derecho.  
Fuente: Propia.

En la figura 21, podemos observar el comportamiento del valor de temperatura del rack derecho, representado por la línea de color celeste, en la que nos indica que la temperatura en ese intervalo de tiempo no ha generado alertas, porque no se ha salido de los límites asignados.



**Figura 22** Grafico de humedad Rack derecho.  
Fuente: Propia.

En la figura 22, podemos observar el comportamiento del valor de humedad del rack derecho, representado por la línea de color celeste en la que nos indica que la humedad en ese intervalo de tiempo, si ha generado alertas, porque siempre se mantuvo debajo del valor warning delimitado en los archivos de configuración, además debió generar alertas por estado critical ya que también decayó esos umbrales inferiores



**Figura 23** Grafico de humo Rack derecho.  
Fuente: Propia.

En la figura 23, podemos observar el comportamiento del valor de humo del rack derecho, representado por la línea celeste en la que nos indica que el valor correspondiente a humo, nunca supero los límites asignados en ese intervalo de tiempo, por tal motivo no ha generado alertas

## 5. Conclusiones.

Se ha utilizado la norma ISO/IEC 17799, porque determina la información como un activo muy relevante que posee valor para la institución y por tal motivo, requiere de una protección y monitorización adecuada dentro de las posibilidades, el objetivo de la seguridad de la información es resguardar dicho activo permitiendo la continuidad operativa, por lo tanto, se espera minimizar los riesgos a través del sistema de monitoreo.

Se optó por la norma IEEE 29148 por ser una guía que permite una selección adecuada en cuanto a requerimientos de software, de esta manera satisfacer las necesidades del sistema, en la comparativa de software resulta elegido el sistema NAGIOS por su eficacia y eficiencia en funcionamiento además de poseer un manejo de reportes, alertas, generación de gráficas y fácil administración.

Como base hardware del sistema se optó por la placa Arduino YUN, ya que es versátil por tener incorporado el microcontrolador Atmega y el microprocesador Atheros, el cual dentro de sus funciones esta, la conectividad mediante un puerto ethernet, lo que facilita la comunicación con el software de monitoreo Nagios alojado en el CLOUD.

Una de las partes fundamentales dentro del sistema, es que se active SNMP para las pruebas iniciales de comunicación, tanto en la máquina que aloja el software NAGIOS, como en el arduino a través del agente snmpd, y que estos se mantengan en la misma comunidad y la misma contraseña para que puedan intercambiar información, correspondiente en autenticación y a valores de los parámetros a monitorear. Basándose en los valores de temperatura y humedad que recomienda la norma TIA 942, no resultaron muy eficientes esos rangos en la configuración, ya que la temperatura tiende a subir y la humedad a bajar, es decir, varia más durante el día con respecto a la noche y esto se debe al incremento de usuarios, lo que conlleva más procesamiento de los equipos, por tal motivo se han definido nuevos intervalos de valores más adecuados a la realidad.

Como conclusión general en base a la realidad evidenciada en el Centro de Datos, podemos decir que fue realizado, siguiendo las guías que dictan las normas de diseño de Centro de Datos, sin embargo, aún falta un sistema de climatización más preciso, porque las corrientes de aire caliente y frío no están muy bien definidas.

## Referencias Bibliográficas.

- 1] *Tecnología de la información - Técnicas de seguridad- Código para la práctica de la gestión de la seguridad de la información*, 2005.
- 2] A. Barba, *Gestión de red*, 1999.
- 3] Stallings, *Fundamentos de seguridad en redes*, 2004, p.263.
- 4] Nagios, 2016. [En línea]. Available: <https://www.nagios.org/>.
- 5] «opennebula.org.» 2017. [En línea]. Available: <https://opennebula.org/about/technology/>.
- 6] [En línea]. Available: <https://goo.gl/oljrgl>.
- 7] [En línea]. Available: (<http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>), s.f.
- 8] [En línea]. Available: [datasheet4u](#), 2017.
- 9] «[naylampmechatronics.com](http://www.naylampmechatronics.com),» [En línea]. Available: [http://www.naylampmechatronics.com/blog/42\\_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html](http://www.naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html).
- 10] [En línea]. Available: [yapo](#), 2017.
- 11] ARDUINO, 2016. [En línea]. Available: <https://www.arduino.cc/en/Guide/ArduinoYun>.
- 12] [En línea]. Available: <https://www.arduino.cc/en/uploads/Guide/YunParts.png>.
- 13] [En línea]. Available: [Arduino](#), 2017.
- 14] [En línea]. Available: [Arduino](#), 2017.

## Sobre los Autores.



Luis T. CAMUÉS NARVÁEZ. Nació en Ibarra el 03 de julio de 1987. Realizó sus estudios primarios en la Escuela “Sebastian de Benalcazar” Los estudios secundarios los realizó en el “Colegio Universitario UTN” donde finalizó en el año 2006, obteniendo el título de Bachiller en Ciencias Especialización Físico Matemático. Actualmente, está realizando su proceso de titulación en Ingeniería en Electrónica y Redes de Comunicación, Universidad Técnica del Norte – Ecuador.



Edgar A. MAYA OLALLA. Nació en Ibarra – Ecuador el 22 de abril del año 1980. Ingeniero en Sistemas Computacionales en la Universidad Técnica del Norte en el año 2006. Magíster en Redes de Comunicaciones en la Pontificia Universidad Católica del Ecuador en el año 2014.

Actualmente es docente de la Carrera de Ingeniería en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte.



# System of monitoring of temperature, humidity and smoke detection for data center located at the Faculty of engineering in applied sciences based on the standard ISO/IEC 17799

*Authors - Tarquino Luis CAMUES NARVAEZ, Ing Edgar Alberto MAYA OLALLA, MSc.*

Faculty of engineering in applied sciences, Technical University of the North, Avenue 17 of July 5-21 and José María Córdova, Ibarra, Imbabura

[lcamuesn@utn.edu.ec](mailto:lcamuesn@utn.edu.ec), [eamaya@utn.edu.ec](mailto:eamaya@utn.edu.ec)

**Abstract.** *This research was about the design and implementation of a temperature, humidity and smoke monitoring system, for the Data Center, located in the Faculty of Engineering in Applied Sciences, based on the standard for information security ISO / IEC 17799, which provides recommendations of best practices in information management. The system consists, on the one hand, hardware for the monitoring by means of sensors and management software, for the observation in a web interface, of the values obtained by the sensors. Likewise, the management system plays the role of alerting the administrator, through e-mail, about the variations in temperature, humidity and smoke values detected in the Data Center.*

*At the beginning, it presents the theoretical basis, which promotes the realization of the research work, converting a data center problem, in an opportunity to improve the performance of the same, which defines the objectives to be fulfilled, as well as its limitations in The scope, also the basis in the method used, for the communication between the software and the hardware, the same way a comparative between management systems is described, to select the most appropriate, and IEEE 29148 is used to analyze The requirements, after the characteristics of the hardware components to be used.*

*A continuation of the design of the monitoring system, with the verification of the functioning of all the components, in the same way the installations are made of all the software packages necessary for the system, together with their configurations, in the final part It carries out the software and hardware coupling for the validation of its operation, the entire process of implementing the monitoring system within the Data Center, and the locations defined in chapter one, and all the necessary tests are carried out. To evaluate the performance of the monitoring system, at the same time, to carry out the analysis of the records of the obtained data, to propose the conclusions and recommendations, that allow the improvements in the Data Center.*

## Keywords

Monitoring system, sensors, Temperature, humidity, smoke, Management software, Embedded system, Data center

## 1. Introducción

The implementation of new data center requires the monitoring of environmental conditions that is inside of them, in order to avoid sudden changes in temperature, humidity or smoke detection. For this reason it is necessary to realization of the system of monitoring, alert to the administrator about the conditions in which the data center is through alerts by email and thus take measures to prevent damage.

## 2. Básic concepts

### 2.1 Data center

Data Center is the place of data processing where staying the resources necessary for the processing of information, and must be designed so that services provide continuous and securely, efficiently operating in its normal capacity and that internal or external problems do not affect its performance.

### 2.2 ISO IEC 17799 standard

The standard "ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized for global standardization system. The standard aims, offer guidelines and principles to initiate, implement, maintain and improve the security of an organization's information management. The objectives provide a guideline, about how to achieve best practices for

better performance in operation of equipment and information processing. [1]

### 2.3 management on the internet

In August 1988, appeared the first SNMP as well as the SMI and MIB corresponding recommendations. [2]

The SNMP protocol, had a great reception in the community and we can say that the SNMP protocol framework is based on three documents.

*Structure of Management Information (SMI): RFC1155.*

*Management Information Base (MIB): RFC 1156, RFC 1213.*

*Simple Network Management Protocol (SNMP): RFC 1157.*

### 2.4 management systems

A network management system is composed of hardware and software deployed between the existing network components, software that is used in the tasks of network management resides: the host, drivers of Terminal groups and communications processors. A network management system is designed to look across the network as an architecture unified with directions and labels assigned to each point, specific attributes of each item and links known to the system. The active elements of the network provide regular feedback of status information to the network control center. [3]

#### 2.4.1 Nagios

Nagios monitors across the IT infrastructure to ensure systems, applications, services, and business processes to function properly. In the event of a failure, Nagios can alert technical staff of the problem, allowing them to begin remediation processes before outages affect the processes of business, end users or customers. With Nagios, you will never leave you having to explain why an invisible infrastructure outage damaged the baseline of your organization. [4]

### 2.5 open Nebula

Occi offers a simple, but rich in features and flexible solution for the integral management of virtualized data centers. Occi interoperability makes the cloud evolution leveraging existing it assets, protecting investment and avoiding providers blocking. Occi is a solution for the enterprise that includes all the features necessary to offer a (private) cloud and to provide public cloud services. [5]

### 2.6. embedded systems

Are open source hardware and easy-to-use software-based electronic platforms. An embedded system is an electronic system that only certain functions. Your hardware is not modifiable, their software Yes. Between the plates that stand out most are:

RASPBERRY, INTEL and ARDUINO

#### 2.6.1 Potentiometer

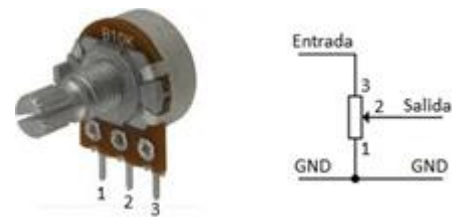


Figure 1. Scheme of the potentiometer.

Source: [6]

A pot has the strength or resistor mechanical variable with cursor and at least three terminals. By connecting the ends to the regular control of voltage potential difference, gets between the central terminal (cursor) and one end a fraction of the potential difference total behaves as a voltage or voltage divider.

#### 2.6.2 Display LCD

(Liquid Crystal Display) or liquid crystal display is a component used to display information in a graphical way, through characters, symbols, or small drawings depending on the model. It is directed by a microcontroller which handles all your operation.



Figure 2. LCD display.

Source: [7]

### 2.6.3 Sensors DHT

The DHT is a sensor of temperature and humidity. Uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and displays the data using a digital signal on the data pin. It is simple to use, but requires synchronization to take data.

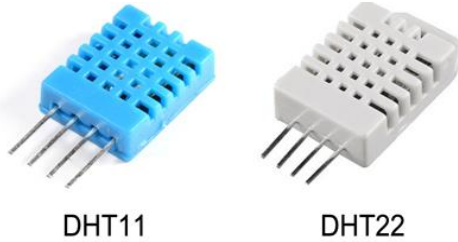


Figure 3. Scheme of the potentiometer. Source: [8]

### 2.6.4 MQ smoke sensors

Are very simple sensors to use, and can detect different types of gases including smoke, sensors are highly sensitive and with a response time fast, its output is an analog resistance. These sensors are electrochemical and when they are exposed some kind of gas varies its resistance, internally carries a heater which increases the internal temperature and for this reason the sensor can react with different types of gases including smoke, causing a variation in the value of the resistance. [9]



Figure 4. MQ smoke sensors Source: [10]

### Ranges of temperature and humidity TIA 942

In the standard TIA 942 page 34 posted in 2005, is set to operation parameters of a data center, a temperature range of 20 °C to 25°C and humidity from 40% to 55%, information that will be relevant at the time of setting up the ranges supported by the system of monitoring.

## 3 Development of the system of monitoring

In this chapter is the development of the system of monitoring, starting with the respective tests and connections from sensors to the embedded system,

development of codes, configurations, and hardware and software to the pre-installation phase coupling.

### 3.1 selection of plate

Within plate models mentioned above, Raspberry and Intel boards meet the needs of the project however there are additional audio peripherals and video that are not needed and for this reason are higher cost, accordingly the proper plate which meets the minimum requirements as input, output, and ethernet port-PIN is the Arduino as well as the ease of programming, they require the selection of the above-mentioned platform.

#### 3.1.1 Selection of Arduino board

In the arduino single range there are 2 plates which bring incorporated an ethernet port, with the exception of the shield ethernet which can be attached with multiple arduino boards, minus some pins to the same features, these two plates are the arduino ethernet and YUN, however only the YUN arduino arduino has a micro-processor linux that can work with the package snmpd essential in the implementation of the system of monitoring for this reason, it is the Board selected for the system of monitoring.

#### 3.1.2 Arduino YUN

Arduino Yun is an electronic based on the ATmega32u4 and Atheros AR9331. Processor Atheros compatible with a Linux distribution based on OpenWrt called OpenWrt-Yun. The plate has been incorporated with Ethernet and support WiFi, a USB-A port, microSD card slot, 20 digital inputs / output pins (of which 7 can be used as PWM outputs and analog inputs 12), a 16 MHz crystal oscillator, a micro USB, an ICSP header, and a 3-button Reset. [11]

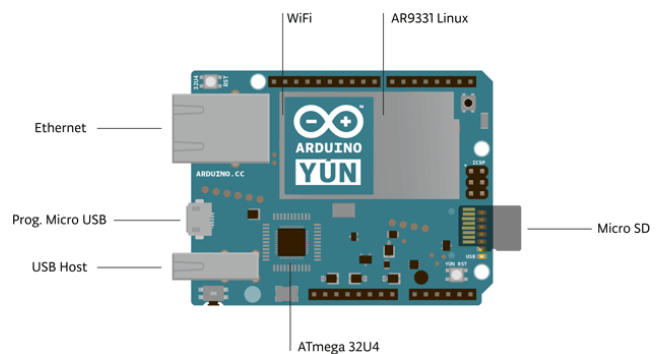


Figure 5. Plate Arduino YUN Source: [12]

<b>Microcontrolador</b>	<b>ATmega32U4</b>
Voltaje de operación	5V
Pines digitales I/O	20



Canales PWM	7
Pines de Entrada Analógica	12
Corriente DC por I/O Pin	40 mA
Corriente DC por 3.3V Pin	50 mA
Memoria Flash	32 KB (of which 4 KB used by bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Tabla1. Microcontrolador Atmega

Fuente: [13]

Procesador	Atheros AR9331
Arquitectura	MIPS @400MHz
Voltaje de Operación	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host
Lector de tarjeta	Micro-SD
RAM	64 MB DDR2
Flash Memoria	16 MB
SRAM	2.5 KB
EEPROM	1 KB
Velocidad de reloj	16 MHz

Tabla2. Microprocesador Linux

Fuente: [14]

### 3.1.3 IDE de Arduino

The integrated development environment is the software that allows you to create the program to give the performance of the embedded system.

## 3.2 Block diagram of the system of monitoring

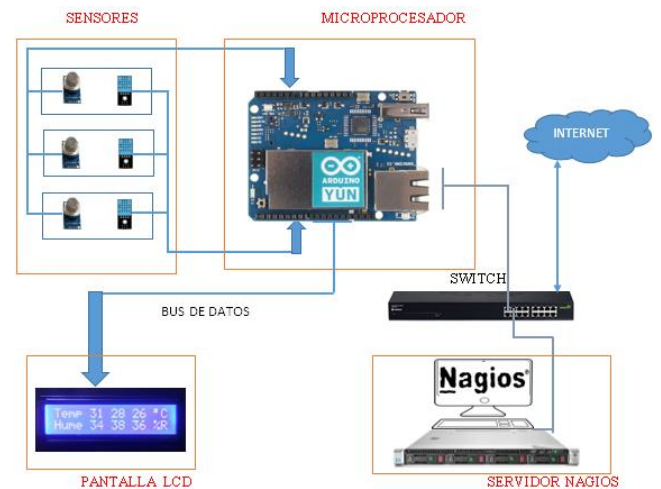


Figure 6. Diagram of the system of monitoring

Source: Own

In the block diagram, we can see the sequence of operation of the system of monitoring, from either the sensor data collection, data that are collected by the arduino and displayed on an LCD screen, only temperature and humidity, at the same time those data the arduino sends them to the system of management to be deployed using a web interface, and based on the data alert the administrator by email.

## 3.3 write OPENWRT-YUN from the arduino code

It is necessary to understand that the arduino YUN, is composed of two processors as shown in Figure 11, the ATMEGA processor and processor ATHEROS, so these two processors can communicate, it is necessary that code be included in the library bridge, which functions as a bridge of communication between the two.

Using the following code shows an example of how writing from the ATMEGA processor to processor ATHEROS. Which includes apart from the BRIDGE, the < FileIO.h > library bookcase, for example will be written to a file called "script.sh" address/tmp.

```
#include < FileIO.h > // include the library files
void setup() {}

Bridge.begin (); We started AVR - LININO
communication

FileSystem.begin (); We started the library files
}

void loop() {}

File script
FileSystem.open("/tmp/script.sh",FILE_WRITE);
```

```

script.print("#!/bin/sh\n"); We declare to the shell type
file
script.print ("test"); the file will contain the word test
script.close (); We close the file
}
    
```

### 3.4 Assembly of the circuit

The Assembly of the prototype, it is using all the components that comprise the system of monitoring, the general connection of hardware, in Proteus based on connections described program may be seen in Figure 7.

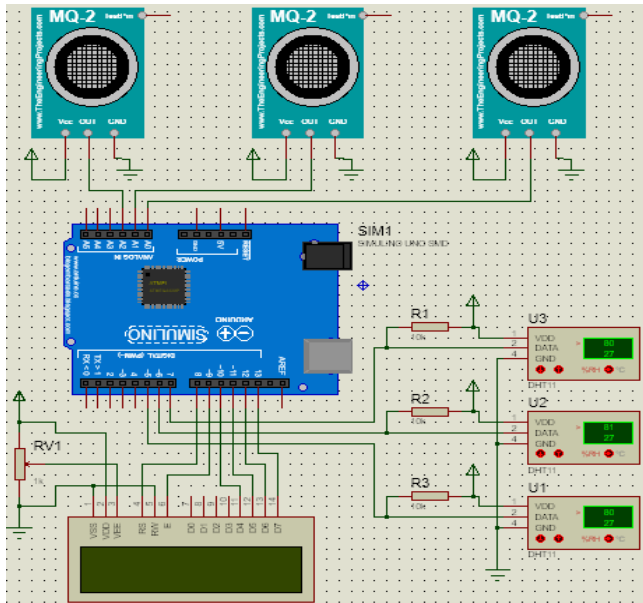


Figure 7. Assembly of the circuit

Source: Own

For circuit protection fuse will be located in power supply input, which estimated the current that will consume the system.

Componente	Corriente consumo	Cantidad	Total
Arduino	260 mA	1	260 mA
Dht11	3.9 mA	3	11.7 mA
MQ-2	137 mA	3	411 mA
LCD	20 mA	1	20 mA
total			702.7 mA

Table3. Determination of the value of the fuse

Source: Own

### 3.5 mounting of the physical circuit

Circuit mounted on a Breadboard for the test runs of the code, as it is, the reading of temperature, humidity and smoke may be seen in Figure 8, the values displayed on the LCD correspond to the position of the sensors accordingly.

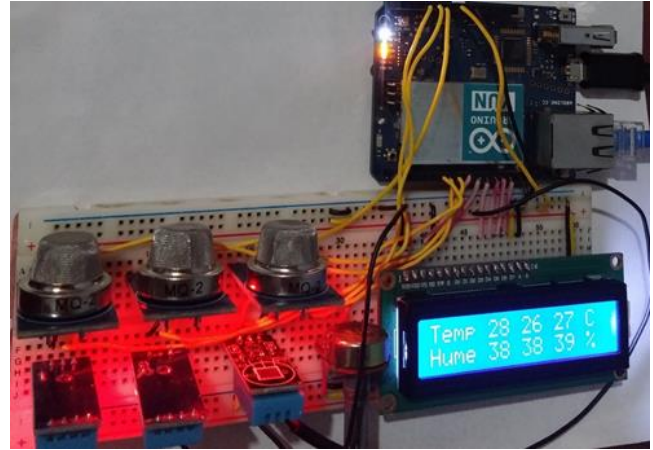


Figure 8. Assembly of the prototype system

Source: Own

### 3.6 installation and configuration of the agent SNMPD on Arduino

We light the Arduino and connect to your wireless network, verify that the network interface is connected to a router with internet access. From the host that has access to the wireless network of the Arduino, we access the mini-Server LININO using the SSH protocol. Being able to access from PUTTY or linux with the command.

```
SSH - p 22 root@192.168.240.1
```

We authenticate credentials and proceed to the installation of the packages

As the MIB are the databases established by the manufacturer, the same OID cannot be used, for this reason they are used used for experimental purposes the OID code that is part of the SMI structure. Located in the tree of identifiers by the branch.

- 1 iso
- 1.3 org
- 1.3.6 dod
- 1.3.6.1 internet
- 1.3.6.1.3 experimental

On the basis of this branch can be located several sub-directorates, there is one that can store variables of type string and the identifier 1.3.6.1.3. x. 101.1, being the number corresponding to each sensor, that way, it is stored written from the arduino code value in the OID corresponding to a file, the resulting tree displays it in the figura9.

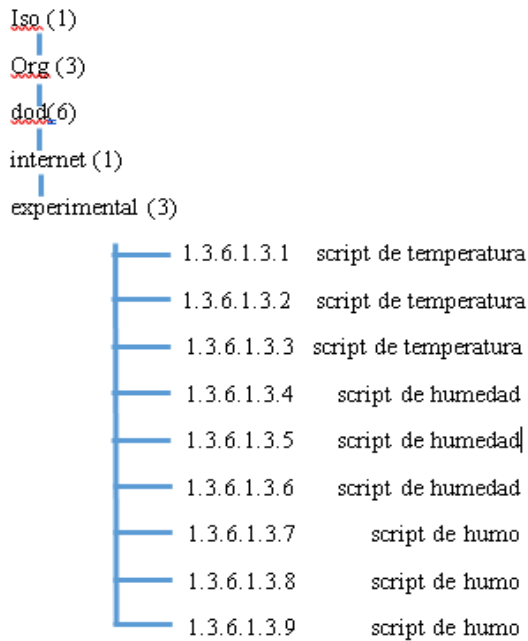


Figure 9. Resulting-tree system of monitoring

Source: Own

### 3.7 installation and configuration of Nagios Core 4.2.4 files

Management 4.2.4 nagios installation, performed together with the installation of the package snmpd on the virtual machine hosted in the Occi CLOUD, made as a thesis project at the Faculty FICA.

Table 11 shows the thresholds of temperature and humidity to configure nagios file permanently after tests conducted during several days of operation.

Parameter	Warning	Critical
Temperature	20 ° c to 27 ° C	18 ° c to 29 ° C
Moisture	32% to 55% RH	30% to 60% RH
Smoke	0 to 400 ppm	0 to 500 ppm

Note: These values can be modified after installation and adjustment of sensors to the internal environment of the data center.

Table3. Activation of alarms ranges

Source: Own

### 3.7.1 display of services monitored in web interface

We started the service httpd, nagios and in a web browser access the address 127.0.0.1/nagios, we authenticate credentials and in the section services, it will display the required information.

To check the history of the host we have to give read permissions writable and executable file nagios.log, we do so with the following command.

```
chmod 607 /usr/local/nagios/var/nagios.log
```

The web interface where the monitored services are displayed can be seen in Figure 10.

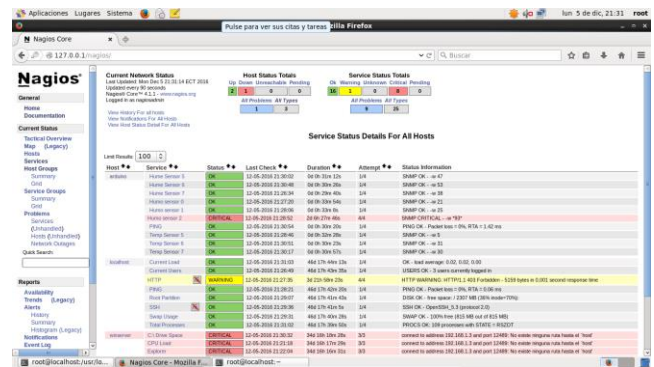


Figure 10. Resulting-tree system of monitoring

Source: Own

### 3.7.2 Nagiosgraph

Nagiosgraph is an open source, developed by sourceforge, to plot the values obtained by the management system Nagios, which can be displayed using the time line for hours, days or months the results of monitoring.

### 3.7.3 installation and configuration of the mail service

We started with the installation of the postfix mail service and install it with the command yum

```
yum - and install postfix
```

We entered the /etc/postfix directory and edit the main.cf file

For example, we use a Gmail account since it is a secure mail and enter the server and the port to use and the required TLS configuration.

```
relayhost = [smtp.gmail.com]: 587
```

We activate the sasl authentication

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash: /
etc/postfix/sasl_passwd
```

```
smtp_sasl_security_options = noanonymous
smtp_tls_CAfile=/etc/postfix/cacert.pem
smtp_use_tls = yes
```

We then generate file  
(vi) /etc/postfix/sasl\_passwd

In the add the following lines where the word "password" refers to the password for the account.  
[smtp.gmail.com]: 587 ltcntonny@gmail.com:contrasena

```
We give read and write permissions to the file
chmod 600 /etc/postfix/sasl_passwd
postmap /etc/postfix/sasl_passwd
chmod 600 /etc/postfix/sasl_passwd.db
```

```
Copy the contents of the certificate
Equifax_Secure_CA.pem to cacert.pem
cat /etc/ssl/certs/Equifax_Secure_CA.pem | Tee -
/etc/postfix/cacert.pem
```

Restart the service  
Service postfix restart

To test the sent post conducted a test using command line  
echo "test message" | mail -s "test subject"  
ltcntonny@gmail.com

If the email was sent and I receipt satisfactorily all is well, case contrary to verify DNS and mail restrictions. Add the contact in the file of nagios, which is going to redirect the e-mail alerts.  
Vim /usr/local/nagios/etc/contacts.cfg  
email nagios@localhost, ltcntonny@gmail.com

We proceed to restart the service  
service restart nagios

In Figure 11, you can see the notification service system Nagios, which indicates, that left rack has detected a value of 37% of relative humidity, which is below the value defined in the configuration file

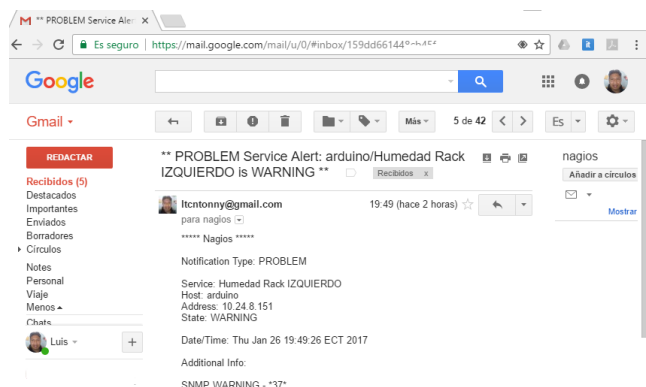


Figure 11. Nagios alerts by email

Source: Own

## 4. Development of the software.

The architecture of the cable network between arduino and server connection may be seen in Figure 12, nagios (cloud), you can also display the location of the sensors that are at the back door of each rack.

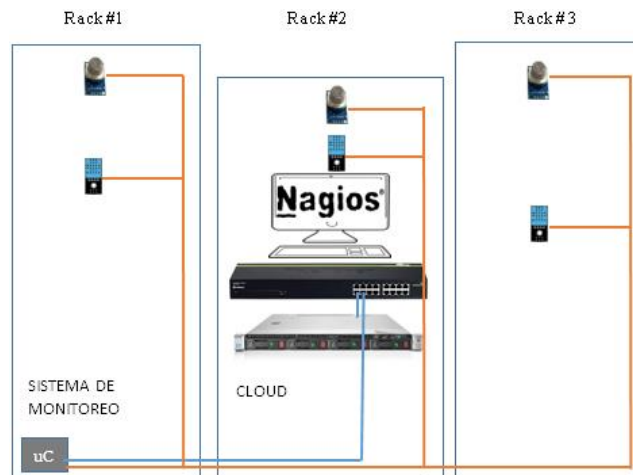


Figure 12 System architecture.

Source: Own.

### 4.1 Assembly and implementation process.

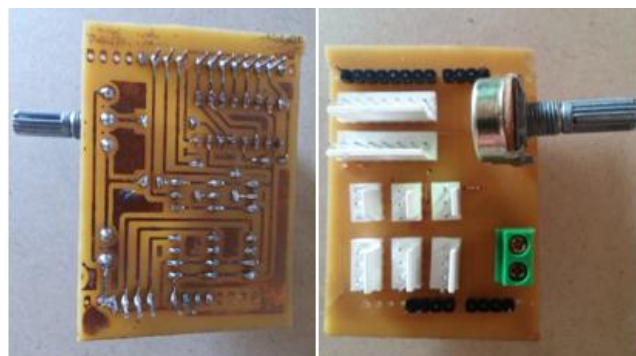
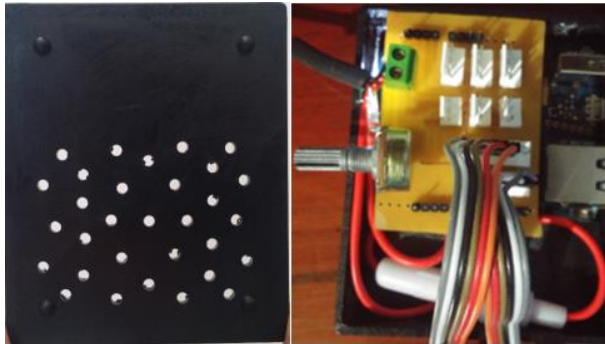


Figure 13 Armored plate that fits into the arduino YUN.

Source: Own.

Figure 13, shows the circuit board that fits into the Arduino Yun, serves as a solid basis for the location of the cables that connects to sensors in each one of the racks, similarly has a potentiometer which serves for calibration of the brightness of the LCD screen.

Below in Figure 14, is the development of the box which will contain inside the embedded system board and connecting cables. At the base where CJD plate, drilling has been, to avoid overheating of the arduino due to information processing, also included the circuit protection fuse.



**Figure 14** Placement of the Arduino in the box.

Source: Own.

In Figure 15, connecting the wiring of sensors, network cable, power to the arduino Board cable

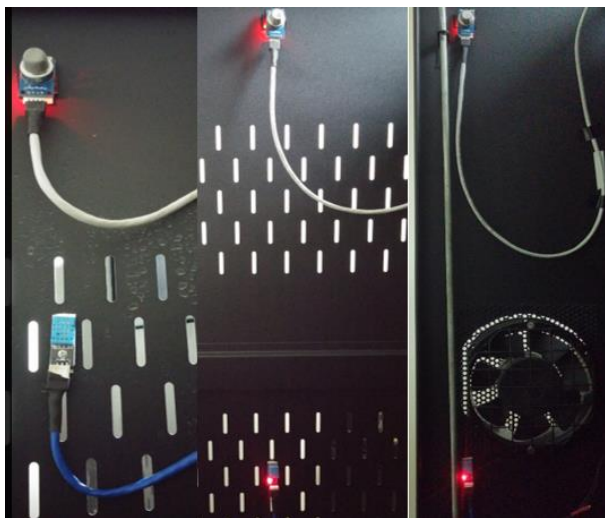


**Figure 15** Placement of the Arduino in the box.

Source: Own

#### 4.2 location of the sensor in the RACK.

The sensors are located in the back door of each rack, in Figure 16, you can display the placement of sensors for temperature, humidity and smoke.



**Figure 16** Placement of the sensor in the RACK.

Source: Own.

Finalizing the process of implementation is located the LCD wiring from inside data center, to the outside, through the false ceiling to the Hall of the faculty, and entering the office to finish close to the fingerprint reader, total was used in 30 meters of cable, two 15 m network cables to work as bus data due to the number of cables for connection between Arduino-LCD.

Figure 17, indicates the values of temperature and humidity in the left, Center and right rack correspondingly the location of sensors per rack to the values shown in the LCD display. The LCD screen is located at right side of the access door to the data center.



**Figure 17** Location of the LCD display.

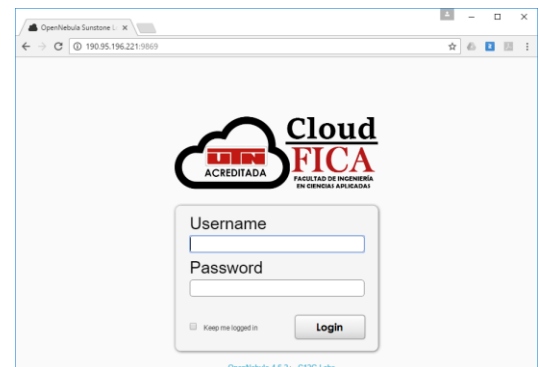
Source: Own.

### 4.3 Joined the system of monitoring.

To enter the GUI that shows the different values that are obtained with sensors, is accessible from outside the University (internet) and within the University (intranet), below, shows the procedure from the internet.

#### 4.3.1 entrance to the system from the internet.

The procedure should be performed are from the internet, to access to the cloud platform and the virtual machine, where the system is installed for monitoring. In a Web browser enter the address 190.95.196.221:9869, where will display the interface in which we must enter our administrator credentials of the account in the cloud, shown in Figure 18.



**Figure 18** Joined the CLOUD platform.

Source: Own

Once inside a browser prompt write the address 127.0.0.1/nagios where is necessary to authenticate credentials and would deploy the boot option of you page, in the left pane, clicking the link leads to services and will show the monitored services, as shown in Figure 19.

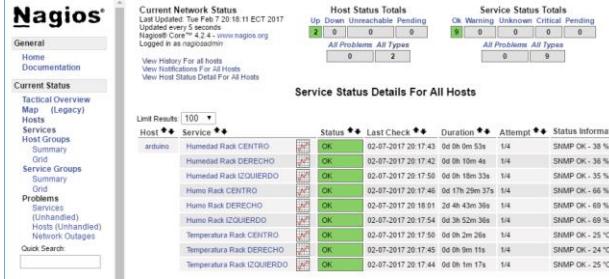


Figure 19 Interface for monitoring Nagios Core.

Source: Own.

### 4.3.2 data registered through graphics display.

Each indicator is set in the files for nagios, which allow to show the values recorded in a graphic way, to observe the registration of values, it is necessary to press the graph corresponding to each sensor as he is indicated by the arrow in Figure 20.

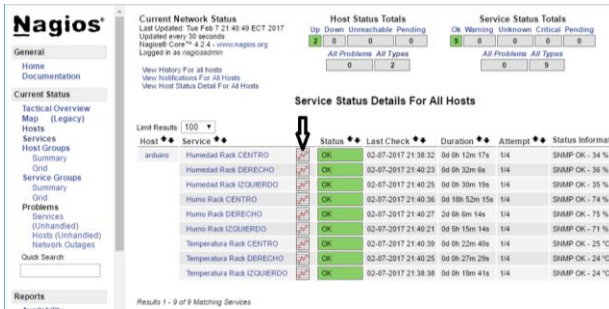


Figure 20 Interface for monitoring Nagios Core.

Source: Own.

Line color	representation
Blue	the critical state threshold
Purple	maximum limit of the State warning
Celeste	corresponding to the monitored variable value
tomato	minimum limit of the State warning
Green	the critical state threshold

Table3. The color in the lines of graphics rendering

Source: Own

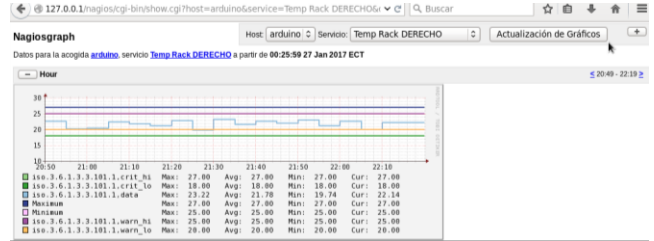


Figure 21 Chart of temperature right Rack.

Source: Own.

In Figure 21, we can observe the behavior of the right rack temperature value, represented by the blue line, which tells us that the temperature in that interval of time has not produced alerts, because it did not come to give them limits allocated.

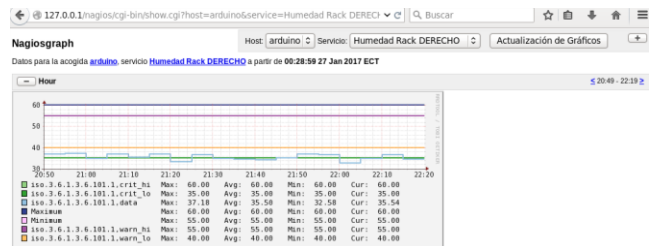


Figure 22 Chart of moisture right Rack.

Source: Own.

In Figure 22, we can observe the behavior of the humidity value of the right rack, represented by the blue line that indicates that moisture in this time interval, if you have generated alerts, because he always kept below the value defined in the configuration files, warning also should generate alerts for critical State already which also declined these lower thresholds

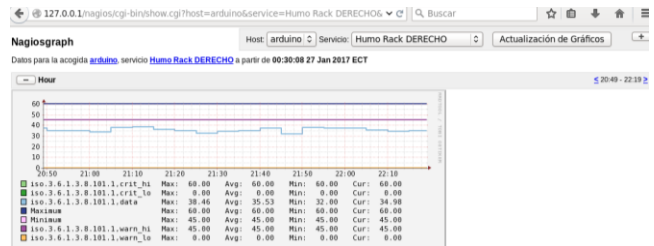


Figure 23 Chart of smoke right Rack.

Source: Own.

In Figure 23, we can observe the behavior of the value of smoke from the right rack, represented by the blue line that indicates that the value of smoke, never exceeded the limits assigned in that interval of time, by such reason has not generated alerts

## 5. Conclusions.

It has been used standard ISO/IEC 17799, because it determines the information as a very important asset that has value for the institution and for that reason, requires a protection and monitoring appropriate within the possibilities, the objective of the information security is safeguard the assets enabling the operational continuity, is therefore expected to minimize risks through the system of monitoring.

Opted for the standard IEEE 29148 for being a guide that allows an appropriate selection with regard to requirements of software, in this way the needs of the system, in the comparison of software is chosen the NAGIOS system for its effectiveness and efficiency in operation as well as own handling of reports, alerts, graphics generation and easy administration.

As base system hardware opted for plate Arduino YUN, it is versatile for having incorporated the Atmega microcontroller and microprocessor Atheros, which its functions of this, connectivity through an ethernet port, allowing easy communication with the software of monitoring Nagios hosted in the CLOUD.

One of the key parts in the system, is to activate SNMP for initial testing of communication, both on the machine that hosts the software NAGIOS the arduino through the snmp agent, and that these are kept in the same community and the same password so that they can exchange information, authentication and monitoring parameter values.

Based on the values of temperature and humidity that recommends the standard TIA 942, were not very efficient those ranges in the configuration, since temperature tends to rise and the humidity down, i.e., various more during the day regarding the evening and this is due to the increase of users, resulting in more processing equipment for this reason have been defined new intervals of values suited to reality.

## Referencias Bibliográficas.

- 1] *Tecnología de la información - Técnicas de seguridad- Código para la práctica de la gestión de la seguridad de la información*, 2005.  
A. Barba, Gestion de red, 1999.
- 2] Stallings, Fundamentos de seguridad en redes, 2004, p.263.
- 3] Nagios, 2016. [En línea]. Available: <https://www.nagios.org/>.
- 4] «opennebula.org,» 2017. [En línea]. Available: <https://opennebula.org/about/technology/>.
- 5] [En línea]. Available: <https://goo.gl/oIjrgl>.
- 6] [En línea]. Available: <http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>, s.f.
- 7] [En línea]. Available: datasheet4u, 2017.
- 8] «naylampmechatronics.com,» [En línea]. Available: [http://www.naylampmechatronics.com/blog/42\\_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html](http://www.naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html).
- 9] [En línea]. Available: yapo, 2017.
- 10] ARDUINO, 2016. [En línea]. Available: <https://www.arduino.cc/en/Guide/ArduinoYun>.
- 11] [En línea]. Available: <https://www.arduino.cc/en/uploads/Guide/YunParts.png>.
- 12] [En línea]. Available: Arduino, 2017.
- 13] [En línea]. Available: Arduino, 2017.
- 14]

## Sobre los Autores.



Luis T. CAMUÉS NARVÁEZ. He was born in Ibarra on July 3, 1987. He completed his primary studies at the "Sebastian de Benalcazar" School. Secondary studies at the "University College UTN" where he finished in 2006, obtaining a bachelor's degree in Sciences Mathematical Physical Specialization. At the moment, it is realizing its process of

titulación in Engineering in Electronics and Communication Networks, Universidad Técnica del Norte - Ecuador



Edgar A. MAYA OLALLA. He was born in Ibarra, Ecuador, on April 22, 1980. He holds a degree in Computer Systems at Universidad Técnica del Norte in 2006. Master's degree in Communication Networks at the Pontifical Catholic University of Ecuador in the year 2014. He is

currently a lecturer at Career of Engineering in Electronics and Communication Networks of the Universidad Técnica del Norte