



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN

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.”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

AUTOR: LUIS TARQUINO CAMUÉS NARVÁEZ

DIRECTOR: MSC EDGAR ALBERTO MAYA OLALLA

IBARRA-ECUADOR

2017



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICIDAD A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

DATOS DEL CONTACTO	
Cédula de identidad	1003439153-3
Apellidos y Nombres	Camués Narváez Luis Tarquino
Dirección	Hermógenes Dávila y Espino
E-mail	ltcamuesn@utn.edu.ec
Teléfono fijo	062-577-022
Teléfono móvil	0989626177
DATOS DE LA OBRA	
Título	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	Camués Narváez Luis Tarquino
Fecha	Abril de 2017
Programa	Pregrado
Título	Ingeniero en Electrónica y Redes de Comunicación
Director	Msc. Edgar Maya

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Camués Narváez Luis Tarquino, con cédula de identidad Nro.100343915-3, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad de material y como apoyo a la educación, investigación y extensión, en concordancia con la ley de Educación Superior Artículo 144.

3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrollo, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume(n) la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.



.....
Firma

Camués Narváez Luis Tarquino
100343915-3
Ibarra, abril de 2017



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A

FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Camués Narváez Luis Tarquino, con cédula de identidad número 100343915-3 manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador artículos 4, 5 y 6, en calidad de la autora del trabajo de grado con el 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. Que ha sido desarrollado para optar por el título de Ingeniero en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

.....
Camués Narváez Luis Tarquino

100343915-3

Ibarra, abril del 2017



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

MAGISTER EDGAR MAYA, DIRECTOR DEL PRESENTE TRABAJO DE
TITULACIÓN CERTIFICA:

Que, el presente trabajo de titulación “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” ha sido desarrollado por el señor Camués Narváez Luis Tarquino bajo mi supervisión.

Es todo en cuanto puedo certificar en honor a la verdad.

Ing. Edgar Maya, Msc.

Director



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

Agradezco a Dios por darme la fortaleza espiritual que he necesitado en todo momento, por ser la fuente de vitalidad para continuar la lucha por el camino de la vida cotidiana y estudiantil. Por ser quien guía mi camino, hasta el cumplimiento de mis metas.

A mi familia, mi padre Tarquino Camués y mi madre Blanca Narváez, mis hermanos y hermanas, por enseñarme e inculcarme el valor del respeto, humildad, y perseverancia, por apoyarme incondicionalmente en todo momento.

Al personal Docente de la Facultad, al personal del Departamento de Informática por transmitir sus conocimientos y ayuda con ética profesional, en especial al Msc. Edgar Maya director de este proyecto de titulación, por su predisposición y paciencia, por haberme orientado con su experiencia profesional y dones de personas.

A Rosita P. por tener un alto espíritu de colaboración y motivación.

Luis Camués N.



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

Este proyecto de titulación se lo dedico a mis padres Tarquino y Blanca, por su esfuerzo y dedicación constante de verme superado en esta gran meta, por brindarme su confianza y ser un soporte fundamental en los momentos difíciles, por transmitirme la perseverancia para continuar en cada una de mis metas, por inculcarme valiosos valores con los que se forjan las grandes personas, pero a la vez sencillas ante la sociedad.

Luis Camués N.

CONTENIDO

AUTORIZACIÓN DE USO Y PUBLICIDAD A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	iv
CERTIFICACIÓN	v
AGRADECIMIENTO.....	vi
DEDICATORIA.....	vii
CONTENIDO	viii
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE TABLAS.....	xiii
RESUMEN.....	xiv
ABSTRACT	xv
CAPÍTULO I.....	1
1 INTRODUCCIÓN	1
1.1 Problema	1
1.2 Objetivos.....	2
1.2.1 Objetivo General	2
1.2.2 Objetivos Específicos	2
1.3 Alcance	3
1.4 Justificación	5
1.5 Contexto	6
CAPÍTULO II	8
2 MARCO TEÓRICO	8
2.1 Definición Estándar ISO IEC 17799.....	8
2.1.1 Cláusulas del Estándar	8
2.1.1.1 Política de Seguridad	9
2.1.1.2 Organización de la seguridad de la información	9
2.1.1.3 Gestión de activos	9
2.1.1.4 Seguridad de recursos humanos	10
2.1.1.5 Seguridad física y ambiental	10
2.1.1.6 Gestión de las comunicaciones y operaciones.....	10
2.1.1.7 Control de acceso	11
2.1.1.8 Adquisición, desarrollo y mantenimiento de los sistemas de información	11
2.1.1.9 Gestión de un incidente en la seguridad de la información	11
2.1.1.10 Gestión de la continuidad del negocio.....	12

2.2	Gestión de red	12
2.2.1	Gestión en Internet	13
2.2.1.1	Estructura de la información de gestión.....	13
2.2.1.2	Sintaxis ASN. 1.....	14
2.2.1.3	MIBS.....	15
2.2.1.3.1	MIBs Experimentales	15
2.2.1.3.2	MIBs Privadas	15
2.2.1.4	Simple Network Managment Protocol (SNMP).....	16
2.2.1.4.1	Operaciones de SNMP	17
2.2.1.4.2	Codificación para la transferencia de la información de gestión VER	18
2.2.1.4.3	Marco administrativo.....	19
2.2.1.5	SNMP v2.....	19
2.2.1.5.1	Operaciones de SNMPv2.....	20
2.2.1.6	SNMP v3.....	21
2.3	Sistemas de Gestión.....	22
2.3.1	Análisis comparativo de los sistemas de gestión	23
2.3.1.1	Zennos.....	23
2.3.1.2	Pandora.....	23
2.3.1.3	Nagios	23
2.3.2	Requerimientos de SGSI bajo la norma IEEE 29148:2011	24
2.3.2.1	Propósito.....	25
2.3.2.2	Alcance.....	25
2.3.2.3	Perspectiva del producto	25
2.3.2.4	Funciones del producto	25
2.3.2.5	Características de los usuarios	26
2.3.2.6	Limitaciones	26
2.3.2.7	Suposiciones y dependencias.....	26
2.3.2.8	Requisitos Complementarios	26
2.3.2.9	Requisitos Específicos	27
2.3.2.10	Justificación	27
2.4	OpenNebula.....	27
2.4.1	Funcionalidades de OpenNebula	27
2.5	Sistema Embebido.....	28
2.5.1	Potenciómetro	29
2.5.2	Pantalla LCD	29
2.5.3	Sensores DHT de temperatura y humedad.....	30

2.5.4	Sensores de humo MQ	31
2.6	Situación Actual	32
2.6.1	Centro de Datos.....	32
2.6.1.1	Subsistema de Arquitectura.....	33
2.6.1.2	Subsistema Eléctrico.....	33
2.6.1.2.1	Tablero general de distribución.....	33
2.6.1.2.2	Sistema de alimentación interrumpida (SAI)	34
2.6.1.3	Subsistema Mecánico	34
2.6.1.3.1	Puerta de seguridad.....	34
2.6.1.3.2	Aire acondicionado	35
2.6.1.3.3	Cámaras de seguridad.....	35
2.6.1.4	Subsistema de Telecomunicaciones	35
CAPÍTULO III	42
3	DESARROLLO DEL SISTEMA DE MONITOREO	42
3.1	Selección de placa	42
3.2	Selección de placa Arduino.....	43
3.2.1	Arduino YUN.....	43
3.2.2	IDE de Arduino	46
3.2.3	Selección de sensor de temperatura y humedad.....	46
3.2.4	Selección de sensor de humo.....	47
3.3	Diagrama de bloques del sistema de monitoreo.....	47
3.4	Lectura de temperatura y humedad con el sensor DHT11.....	48
3.4.1	Visualización de lectura de temperatura y humedad por comunicación serial.....	50
3.5	Lectura de humo con el sensor MQ2	50
3.5.1	Visualización de lectura de datos correspondiente a humo por comunicación serial .	52
3.6	Conexión de pantalla LCD al arduino	53
3.7	Escritura en OPENWRT-YUN desde el código de arduino	55
3.8	Desarrollo del Código para el Sistema de Monitoreo	57
3.8.1	Diagrama de flujo del sistema de monitoreo.....	57
3.9	Armado del circuito.....	59
3.10	Diseño de la fuente de energía	60
3.11	Montaje del circuito físico	61
3.12	Instalación y configuración del Agente SNMPD en Arduino.....	62
3.13	Instalación y Configuración de archivos en Nagios 4.2.4	64
3.14	Instalación y configuración de nagiosgraph	73
3.15	Instalación y configuración del servicio de correo.....	74

CAPITULO IV	77
4 IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO	77
4.1 Arquitectura del sistema	77
4.2 Proceso de armado e implementación del sistema.....	78
4.3 Ubicación de los sensores en los Rack	80
4.4 Ingreso al sistema de monitoreo	83
4.4.1 Ingreso al sistema desde internet	83
4.4.2 Ingreso al sistema desde la intranet.....	85
4.5 Visualización de datos registrados mediante gráficos	86
4.6 Propuesta de políticas para la seguridad física y ambiental del Centro de Datos.....	89
CONCLUSIONES.....	90
RECOMENDACIONES	92
GLOSARIO DE TERMINOS.....	93
ANEXO A: CÓDIGO DEL SISTEMA DE MONITOREO.....	95
ANEXO B: INSTALACIÓN Y CONFIGURACIÓN DE SNMPD EN LININO	99
ANEXO C: INSTALACIÓN Y CONFIGURACIÓN DE NAGIOS CORE 4.2.4	102
ANEXO D: INSTALACIÓN Y CONFIGURACIÓN DE NAGIOSGRAPH.....	104
ANEXO E: POLÍTICAS PARA LA SEGURIDAD FÍSICA Y AMBIENTAL DEL CENTRO DE DATOS.	106
ANEXO F: UBICACIÓN DE SENSORES	108
ANEXO G: HOJA DE DATOS SENSOR MQ-2	110
ANEXO H: HOJA DE DATOS SENSOR DHT11.....	112
ANEXO I: HOJA DE DATOS ARDUINO YUN	114
BIBLIOGRAFÍA.....	116

ÍNDICE DE FIGURAS

Figura 1. Ubicación de sensores.....	5
Figura 2. Árbol de Registro SNMP	14
Figura 3. Arquitectura de SNMP.....	16
Figura 4. Secuencia de PDU en el protocolo SNMP	18
Figura 5. Esquema del potenciómetro	29
Figura 6. Pantalla LCD.....	29
Figura 7. Sensores DHT	30
Figura 8. Sensores de humo MQ.....	31
Figura 9. Arquitectura de la red FICA.....	36
Figura 10. Placa Arduino YUN.....	44
Figura 11. Procesador Atmega y procesador LININO	44
Figura 12. Diagrama del sistema de monitoreo.....	47
Figura 13. Conexión de sensor dht11 al arduino.....	48
Figura 14. Visualización de valores de temperatura y humedad.....	50
Figura 15. Conexión del sensor MQ-2 al arduino	51
Figura 16. Visualización de valores del sensor de humo.....	52
Figura 17. Conexión de pantalla LCD al arduino.....	53
Figura 18. Acceso a OpenWrt mediante putty.	56
Figura 19. Ubicación del archivo creado desde arduino.....	57
Figura 20. Diagrama de flujo del código	58
Figura 21. Armado del circuito	59
Figura 22: Diseño de la fuente	60
Figura 23. Armado del prototipo del sistema.....	62
Figura 24. Árbol resultante sistema de monitoreo.....	63
Figura 25. Datos de los sensores sin humo	65
Figura 26. Exposición de humo al sensor MQ-2.....	66
Figura 27. Datos de los sensores con humo.	67
Figura 28. Interfaz web del sistema de gestión.	73
Figura 29. Activación de gráficos para los servicios monitoreados.	74
Figura 30. Alertas de nagios por correo.....	76
Figura 31. Arquitectura del sistema.....	77
Figura 32. Diseño de la placa.....	78
Figura 33. Armado de la placa que encaja en el Arduino Yun.....	78
Figura 34. Ubicación del arduino en la caja.....	79
Figura 35. Ubicación del sistema en el Centro de Datos.	80
Figura 36. Ubicación de los sensores en los rack izquierdo, central, derecho.	81
Figura 37. Ubicación de la pantalla LCD.	82
Figura 38. Colocación de ganchos para los extintores.....	82
Figura 39. Interfaz de la plataforma Cloud de la Universidad.	83
Figura 40. Entorno de administración de la cuenta Cloud.	84
Figura 41. Interfaz principal de Nagios Core.....	85
Figura 42. Interfaz de monitoreo de Nagios Core.	86
Figura 43. Figura para despliegue de monitoreo gráfico.	87
Figura 44. Gráfico de temperatura rack derecho.	88
Figura 45. Gráfico de humedad rack derecho.	88
Figura 46. Gráfico de humo rack derecho.....	89

ÍNDICE DE TABLAS

Tabla 1. Comparativa de sistemas de Gestión.....	24
Tabla 2. Características de sensores DHT	31
Tabla 3. Características de los sensores MQ.....	32
Tabla 4. Equipamiento TIC del centro de datos FICA	37
Tabla 5. Rangos de temperatura y humedad establecidas por fabricante	41
Tabla 6 : Comparativa sistemas embebidos.....	42
Tabla 7. Microcontrolador Atmega	45
Tabla 8. Microprocesador Linux	45
Tabla 9. Conexión pines LCD-ARDUINO.....	54
Tabla 10. Determinación del valor del fusible	60
Tabla 11: Rangos de activación de alarmas	68
Tabla 12. Representación del color de las líneas en los gráficos	87

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.

En los capítulos iniciales, 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.

En los capítulos siguientes 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.

ABSTRACT

This research was about the design and implementation of a temperature, humidity and smoke monitoring system for the Data Center in Applied Sciences Engineering Faculty, it was based on the ISO / IEC 17799, which provides recommendations about the best practices in information management. The system had one hand, hardware to monitor by means of sensors and management software. To observe a web interface, values obtained by the sensors; likewise, the management system plays the role of alerting the administrator by an e-mail above variations in temperature, humidity and smoke values, detected in the Data Center. Theoretical basis is presented in the initial chapter, which promotes this work, converting a data center problem, in an opportunity to improve its performance and it defines the objectives to be fulfilled, as well as its limitations. In scope, it also explains the basis for the method used for communication between software and hardware, similarly describes a comparison between management systems to select the most appropriated and IEEE 29148 is used to analyze the requirements, further details the characteristics of the hardware components to be used. In the following chapters, the design of the monitoring system is started, with the verification of the functioning of all the components, as well as the installations, of all the software packages necessary for the system, together with their configurations, in the part Finally, the software and hardware coupling is performed for the validation of its operation, after which the entire process of implementing the monitoring system is carried out within the Data Center, in the locations defined in chapter one, and at the end all the Necessary tests, to evaluate the performance of the monitoring system, at the same time, to perform the analysis of the records of the obtained data, to propose the conclusions and recommendations, that allow the improvements in the Data Center.

KEYWORDS: Systems, temperature, humidity, smoke, values, software



CAPÍTULO I

1 INTRODUCCIÓN

En este capítulo, se presenta el problema que surge a partir de una nueva dependencia en la facultad, para convertirlo en una oportunidad, que da inicio al trabajo de investigación. Se describe su problemática, se realiza el planteamiento de una solución y se expone una propuesta en objetivos y metas a desarrollarse.

1.1 Problema

La Universidad Técnica del Norte, es una institución de educación superior pública y acreditada, cuenta con varias facultades, entre las cuales se encuentra la Facultad de Ingeniería en Ciencias Aplicadas, en la que se realizó la implementación de un Centro de Datos, ubicado en la planta baja del edificio, estos equipos proveen varios servicios de vital importancia para los docentes y estudiantes de la Facultad.

El Centro de Datos, en su interior cuenta con equipos de comunicaciones de un costo elevado y sensibles a diversas condiciones ambientales, al momento no dispone de un sistema de monitoreo de la seguridad del equipamiento físico, “para reducir las amenazas y peligros ambientales” (ISO/IEC 17799: Tecnología de la información - Técnicas de seguridad- Código para la práctica de la gestión de la seguridad de la información, 2005) que permitan respaldar el buen funcionamiento lógico y protección del hardware, fundamental para un lugar de estas prestaciones, lo que lo hace vulnerable a cambios como, la temperatura que se debe mantener en el interior del cuarto, ya que varía constantemente, debido al calor que produce el funcionamiento de las tarjetas electrónicas, y el aire que fluye de los ventiladores, sin una canalización adecuada, la humedad que puede concentrarse en el ambiente del cuarto, debido a la cantidad de vapor de agua presente en el aire, la enorme cantidad de componentes

electrónicos, cableado y conexiones, que provocan el calentamiento excesivo de los mismos y consecuentemente se pudiera producir un incendio, estos fenómenos pueden ocasionar el mal funcionamiento de los equipos, o del cableado instalado en el Centro de Datos.

Para solucionar los problemas mencionados, se realizará un sistema de monitorización desarrollado con el sistema embebido Arduino, en el que se puede utilizar un paquete que permite la comunicación mediante el protocolo SNMP, con un sistema de gestión de seguridad de la información y monitoreo, en el sistema embebido se acoplarán sensores, que permita tomar las señales analógicas de temperatura, humedad y detección de humo, para poder transformarlas en señales digitales, que posterior se puedan visualizar en una pantalla led ubicada en el centro de datos, de igual manera que se pueda visualizar en el sistema de gestión, y que el sistema pueda alertar al administrador del Centro de Datos sobre alguna variación de los parámetros a censar, así se garanticen las buenas condiciones de funcionamiento en el interior del centro de datos, con lo que se ayuda a la conservación del hardware y por ende la buena ejecución del software, para que brinde los servicios implementados de una forma ininterrumpida en su rango de operaciones normal por dichos inconvenientes a futuro.

1.2 Objetivos

1.2.1 Objetivo General

Implementar un sistema de monitorización, basado en la norma ISO/IEC 17799, utilizando sistemas embebidos y sistema de gestión, para determinar la temperatura, humedad y detección de humo en el Centro de Datos de la Facultad FICA.

1.2.2 Objetivos Específicos

- Analizar la norma ISO/IEC 17799, sobre sistemas de monitorización de tecnologías de la información.

- Realizar un levantamiento de los equipos que se encuentran actualmente en el centro de datos, servicios que proveen y características ambientales, con la finalidad de relacionar con los parámetros ambientales que recomienda el estándar ANSI/EIA/TIA 942.
- Examinar los diferentes sistemas de gestión, sus prestaciones y elección del más adecuado para ser implementado como servidor, utilizando el estándar IEEE 29148
- Determinar las funcionalidades del sistema embebido en sus diferentes placas y selección de la más adecuada para la realización del proyecto.
- Desarrollo del programa en el IDE de Arduino para la implementación en el hardware y su posterior evaluación de funcionamiento.

1.3 Alcance

El proyecto está enfocado en, realizar el diseño e implementación del sistema de monitorización de temperatura, humedad y detección de humo, que este comunicado con un sistema de gestión para el Centro de Datos, ubicado en la Facultad de Ingeniería en Ciencias Aplicadas, fundamentado en la norma ISO/IEC 17799.

En base a los equipos que se encuentren en el Centro de Datos, investigar los servicios que proveen, analizar los parámetros ambientales que establecen los fabricantes, y relacionar con las recomendaciones que la norma establece, para desarrollar el esquema de los sistemas necesarios, posibles soluciones que cumplan con los objetivos del proyecto, utilizando conocimientos adquiridos en las aulas.

Se investigará conforme lo establece la norma de Especificación de requisitos de software IEEE 29148, sobre las herramientas de gestión en software libre como NAGIOS, ZENOSS, PANDORA. Para describir sus: perspectivas, funcionalidades, características, restricciones, suposiciones y dependencias, que permitan validar las herramientas utilizando las

consideraciones técnicas, y seleccionar el sistema de gestión adecuado para la implementación del servidor, en el cual el administrador pueda gestionar el sistema de monitoreo.

El servidor será implementado en la plataforma Cloud de la universidad utilizando como base el sistema operativo CentOS, el cual permite la instalación de los paquetes respectivos a los sistemas de gestión y dependencias como base de datos en donde se almacena la información recopilada, la cual permite saber el historial de eventos y generación de reportes. Los requerimientos mínimos de la imagen a disponer en la plataforma Cloud, que necesita CentOS son:

MEMORIA RAM: 1GB

PROCESADOR: 1GHz

DISCO: 10 GB

Se diseñará el sistema embebido con dispositivos, componentes y sensores, ubicados en lugares estratégicos tomando como ejemplo el Documento Técnico 102, “Monitoreo de Amenazas Físicas en Centro de Datos” (Figura 1), para la recopilación de información analógica e interpretación de dichas señales de manera digital en capa aplicación, esto se lo realizará utilizando el protocolo SNMP, que permite la comunicación entre dispositivos de red, como es el hardware Arduino en el que podemos instalar el paquete SNMPD que servirá de agente para enviar información al sistema de gestión, y que el sistema pueda alertar al administrador mediante un correo electrónico, sobre alguna alarma en cuanto a los parámetros tomados en cuenta en la elaboración del proyecto, con lo cual se permita realizar un monitoreo, de temperatura, humedad y detección de humo, en el interior del Centro de Datos, para que se puedan implementar las recomendaciones de seguridad ambientales.

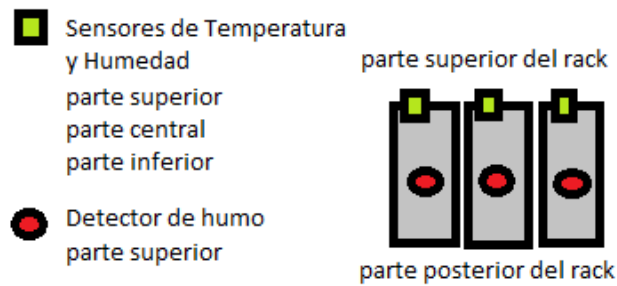


Figura 1. Ubicación de sensores
 Fuente: Elaborado por el autor

En la etapa final se realizará el análisis del hardware Arduino adecuado entre sus diferentes placas y shields, que permitan una buena funcionalidad del proyecto, así como flexibilidad al momento de desarrollar el programa, para su posterior implementación y elaboración de pruebas y conclusiones.

1.4 Justificación

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, causando indisponibilidad de sus prestaciones a los profesores y estudiantes, que son los usuarios directos de los servicios, el sistema de monitoreo permitirá que el administrador pueda saber de manera oportuna y rápida sobre inconvenientes en el Centro de Datos, así poder resguardar el valor de los activos e información, tomando las medidas adecuadas que solucionen los inconvenientes, y que los beneficiarios puedan acceder continuamente a los servicios alojados en los servidores.

Por tal motivo existen varios sistemas de control y seguridad que protejan los equipos y por ende la información almacenada, como pueden ser cámaras de seguridad, UPS, controles de acceso, algo con lo que ya dispone el centro de datos, sin embargo, uno de los factores importantes es la monitorización de temperatura, humedad y detección de humo, que aún no se ha implementado en el Centro de Datos, tomando como referencia lo que establece la norma

ISO/IEC 17799 en la sección 9.2.1 literal (f), es necesario la monitorización de los medios de procesamiento de la información. La implementación de un sistema de monitoreo permite saber en tiempo real los aspectos críticos y fundamentales, conocer el estado de los equipos, programar las tareas de mantenimiento, por tal motivo es necesario ubicar los dispositivos adecuados en puntos estratégicos que permitan censar los parámetros establecidos en el proyecto, garantizándonos que obtendremos la información adecuada, que nos permita entender el comportamiento del centro de datos, para poder realizar las adecuaciones necesarias y mejorar las condiciones de funcionamiento.

Un aspecto importante para la elaboración del proyecto es, ser usuario de los servicios que provee el Centro de Datos, así como inconvenientes en base a información recopilada que han sucedido en el Centro de Datos del Edificio Central a inicios del año en curso, por tal motivo poder aportar con la implementación del sistema, en base a conocimientos obtenidos en clase de temas importantes como, las normas de cableado estructurado y Centro de Datos, que hacen mención a temas relacionados con tecnologías de la información, de igual manera administración y gestión de redes en donde se estudió sistemas de monitorización y control, así mismo los conocimientos adquiridos sobre sistemas embebidos, que permitirán el desarrollo del proyecto.

1.5 Contexto

En la tesis “OPTIMIZACIÓN DE LA ADMINISTRACIÓN EN LA RED DE DATOS DE LA UNIVERSIDAD TÉCNICA DEL NORTE IMPLEMENTADO UN SISTEMA DE MONITORIZACIÓN DE EQUIPOS Y SERVICIOS UTILIZANDO SOFTWARE LIBRE” elaborada por Jeanpierre Wenceslao Castro Flores se realiza la implementación de un sistema de monitoreo utilizando software libre, en donde se describen fundamentos de administración, mapeo de la red, políticas de administración, rendimiento de la red, control de fallas en la red, aplicado para la red de datos de la UTN, siendo concretamente monitoreo lógico de la red. El

tema propuesto se diferencia del realizado porque se monitoreará la temperatura, humedad, detección de humo y esto corresponde a nivel físico, los dos temas tienen relación porque convergen en el mismo punto que es la monitorización para alertar sobre algún problema y dar soluciones que permitan la continuidad de funcionamiento. Adicionalmente el proyecto propuesto pretende la implementación del sistema de monitoreo en sistemas embebidos, que contara con una pantalla que muestre los valores de los parámetros a censar y también con un sistema de alerta mediante correo electrónico al administrador de la red, que diferenciara al proyecto realizado.

CAPÍTULO II

2 MARCO TEÓRICO

En el actual capítulo, se realiza un breve resumen de la norma ISO/IEC 17799, en la que se fundamenta el trabajo, también se realiza la recopilación de información, sobre gestión de red según internet. A lo posterior, se hace una comparativa de sistemas de gestión, utilizando la norma IEEE 29148, para la selección del más adecuado, de igual manera, se explica sobre la situación actual en la que se encuentra el Centro de Datos, para finalizar el capítulo, se describen los elementos, que constituirán el sistema de monitorización en cuanto a hardware.

2.1 Definición Estándar ISO IEC 17799

El estándar “ISO (Organización Internacional de Estandarización) e IEC (Comisión Electrotécnica Internacional) forman el sistema especializado para la estandarización mundial.” (ISO/IEC 17799: Tecnología de la información - Técnicas de seguridad- Código para la práctica de la gestión de la seguridad de la información, 2005, pág. 7) 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. La seguridad de la información garantiza la preservación de: la confidencialidad, permitiendo que solo las personas autorizadas tengan acceso a la información, la integridad de los datos y disponibilidad de los servicios o acceso a la información en el momento que así se lo requiera.

2.1.1 Cláusulas del Estándar

Las organizaciones deben evaluar los riesgos de la seguridad de los sistemas y sus equipos, para poder adoptar controles necesarios, que disminuyan las posibles amenazas. Entre los

controles que la guía proporciona para la gestión de la seguridad de la información, se detalla las cláusulas más importantes.

2.1.1.1 Política de Seguridad

Objetivo: brindar a la administración, las directrices y soporte para mejorar la seguridad de la información, en base a los requisitos comerciales, leyes y reglamentación importante.

La administración, debe otorgar de una manera clara, el sentido de la política a establecer en base a los objetivos comerciales, comprometer su apoyo, con la seguridad de la información, mediante el acogimiento de la política.

El folleto de políticas de seguridad debe ser de conocimiento de todo el personal involucrado, así como de personas externas si así lo requiere.

2.1.1.2 Organización de la seguridad de la información

Objetivo: establecer un cuadro de referencia gerencial, para comenzar y controlar la implementación de la seguridad de la información en la organización.

La administración aprobará, la política de seguridad de la información asignará las funciones de seguridad, planeará y revisará la implementación de la seguridad en toda la organización. Se debería controlar el acceso por parte de grupos externos a medios de procesamiento.

2.1.1.3 Gestión de activos

Objetivo: adoptar una adecuada protección de los activos de la organización, así como llevar un inventario de los mismos y asignación de responsables.

Los responsables deben llevar el mantenimiento y protección apropiado de los equipos a su cargo.

2.1.1.4 Seguridad de recursos humanos

Objetivo: verificar que los colaboradores, contratistas y terceros adopten sus responsabilidades, y sean las personas adecuadas para cumplir con las funciones asignadas, disminuyendo las amenazas como robo, fraude o mal uso.

Todo aspirante algún cargo de la organización, estará sujeto a las investigaciones en cuanto a sus antecedentes y desempeño, especialmente para trabajos confidenciales, así como firmar un acuerdo sobre sus obligaciones y responsabilidades con la relación de la seguridad.

2.1.1.5 Seguridad física y ambiental

Objetivo: restringir el acceso a personas no autorizadas, para evitar daño e interferencia con el área de la organización e información.

Los activos de la organización como, medios de procesamiento de información relevante, deberán estar ubicados en áreas seguras y protegidas por los perímetros delimitados, se deben monitorear las condiciones ambientales, tales como; temperatura y humedad, así como evitar amenazas como el fuego o humo.

2.1.1.6 Gestión de las comunicaciones y operaciones

Objetivo: garantizar el funcionamiento correcto y seguro de los medios de procesamiento de la información.

Establecer responsabilidades y lineamientos, para la administración y operación, de todos los medios de procesamiento de la información, implementar medidas para reducir el riesgo de negligencia o mal uso de los sistemas, pudiendo ser procedimientos documentados sobre el manejo y operación de los equipos.

2.1.1.7 Control de acceso

Objetivo: controlar el acceso a la información.

Vigilar el acceso a la información, medios de procesamiento y procesos comerciales, sobre la base de los requerimientos comerciales y de seguridad, establecer lineamientos para controlar la atribución de los derechos de acceso a los sistemas y servicios, con un registro durante todo el periodo de operación de inicio a fin, así como también el uso de claves secretas para los accesos.

2.1.1.8 Adquisición, desarrollo y mantenimiento de los sistemas de información

Objetivo: garantizar que la seguridad forme parte de los sistemas de información

Verificar que los sistemas de información cuenten con la infraestructura adecuada y aplicaciones que soporten el proceso comercial, esto se lo llevara a cabo identificando y adecuando los requerimientos de seguridad en su etapa de desarrollo e implementación de los sistemas de información, así como la integridad de los datos a transmitirse.

2.1.1.9 Gestión de un incidente en la seguridad de la información

Objetivo: verificar que los imprevistos y debilidades de la seguridad de la información, estén ligados con los sistemas de información y sean oportunamente comunicados, de tal manera que permita realizar una corrección oportuna del problema.

Establecer métodos formales de reporte de eventos, todos los involucrados deben estar comunicados sobre los procedimientos a llevar a cabo, cuando suceda algún evento inesperado, así como informar sobre posibles amenazas, que puedan darse en un futuro.

2.1.1.10 Gestión de la continuidad del negocio

Objetivo: contraatacar ante la paralización, de las actividades comerciales y resguardar los procesos comerciales, ante fallas o catástrofes en los sistemas de información, asegurando su pronta recuperación y operación.

Implementar el proceso de administración de la continuidad del negocio, para reducir la conmoción en la organización pudiendo recuperarse tras desastres naturales, fallas de los equipos o accidentes, utilizando controles preventivos y de recuperación.

Evaluar e identificar las posibles amenazas, para implementar planes de continuidad del negocio, garantizando que la seguridad de la información sea la base para su desarrollo.

2.2 Gestión de red

Un sistema de comunicación está definido, como una arquitectura de red interconectada por medios de transmisión y protocolos, los cuales permiten la utilización de recursos informáticos en la red interna. (Huidrobo, 2008, pag. 202) .

“Existen una serie de modelos de gestión normalizados, en los cuales es posible el acceso uniforme a los recursos gestionados. Se normaliza el protocolo de comunicaciones, el modelo de información de gestión y las definiciones de información de gestión. Los modelos de gestión de red tradicionales más importantes son: la arquitectura TMN (ITU-T), el modelo de gestión OSI (ISO) con el protocolo CMIP y el modelo de gestión Internet (IETF), basado en el protocolo SNMP.” (Barba, 1999)

“Las plataformas de gestión, usan un conjunto de aplicaciones, para poder adaptarse al entorno cambiante y complejo de los elementos de red, que se quieran gestionar. Entre las aplicaciones más utilizadas que se incorporan, destacan los *MIB browser* (navegadores de MIB) como interfaces de usuario del protocolo SNMP; el *discover*, que permite auto descubrir equipos y topologías de la red; la programación de exploración de variables de la MIB; la

programación de acciones ante las alarmas; y finalmente los visualizadores gráficos de valores de variables de MIB.” (Barba, 1999)

2.2.1 Gestión en Internet

Las bases cercanas a la gestión de red en Internet iniciaron en marzo de 1987, con una serie de protocolos como el SGMP (*Simple Gateway Monitoring Protocol*), el HEMS (*High-Level Entity Management System*) y el CMOT (versión de CMIP sobre TCP). Estos protocolos de gestión se encargaban, del buen funcionamiento de los nodos de encaminamiento centrales del grupo de redes Internet. Después, en febrero de 1988, se realizaron una serie de revisiones para actualizar el protocolo SGMP, que dieron el inicio a SNMP. Más a largo plazo, existía la versión del CMOT como alternativa. Posteriormente en agosto de 1988, aparecieron realmente las primeras recomendaciones del SNMP, así como de la SMI y MIB correspondientes. (Barba, 1999)

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.

2.2.1.1 Estructura de la información de gestión

El objetivo, consiste en referenciar un recurso en un sistema remoto, en donde se necesita el protocolo IP que permite llegar al sistema remoto, o el protocolo SNMP que permite llegar al proceso de gestión de red del sistema remoto. Para ello se utiliza un método para nombrar los objetos, llamados identificadores de objetos OID.

Los OID son una secuencia de enteros no negativos separados por un punto, de tal manera que forman un árbol, dicho árbol se encuentra estandarizado a nivel mundial, el árbol está

conformado por ramas y nodos, cada nodo puede tener nodos hijos(sub-identificadores). El árbol empieza en el nodo inicial llamado root y se extiende hasta cualquier nivel de profundidad. (Barba, 1999). En la figura 2 se puede observar el árbol de registro y sus ramas.

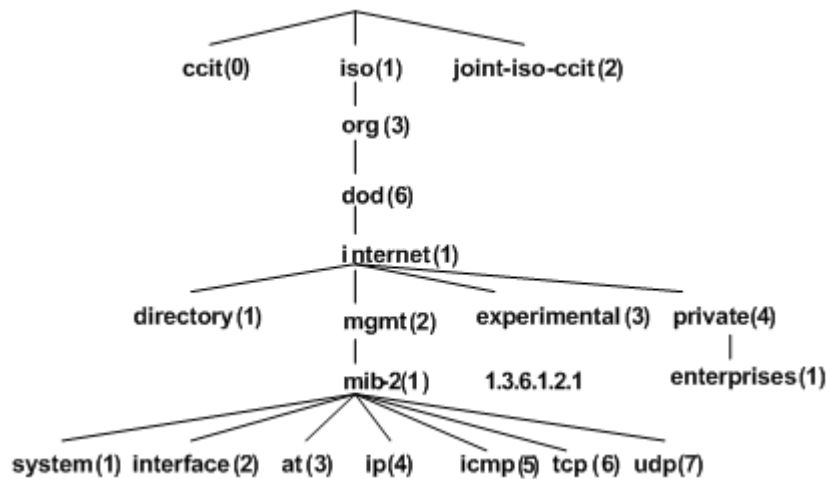


Figura 2. Árbol de Registro SNMP

Fuente: SNMP tecnología. (2017), recuperado de <https://goo.gl/313Jq6>

Los OID permiten alcanzar objetos mediante SNMP, para que el sistema pueda dar respuesta a una petición, se necesita la estructura de los valores, que pueden llegar desde los objetos conocida como la Macro OBJECT-TYPE, de igual forma, conocer el uso de la codificación por línea conocida de los valores (sintaxis de referencia ASN 1).

2.2.1.2 Sintaxis ASN. 1

La sintaxis ASN.1, se representa por el término *Syntax*, en la macro de tipo de objetos, define el tipo de datos que modela el objeto. La sintaxis abstracta, es utilizada para describir las estructuras de datos, que intercambian las entidades del protocolo SNMP, así como la información de gestión que contienen estas estructuras de datos. La sintaxis de transferencia permite a partir de la definición de las estructuras de datos, utilizar una manera determinada de transmitir los datos a través de la red. La sintaxis de transferencia que se utiliza junto con la ASN.1 son las reglas de codificación básicas (*Basic Encoding Rules*, BER).

Una colección de descripciones ASN.1 relacionadas con un mismo tema se conoce como *módulo*. Se definen tres tipos de objetos con ASN.1:

Tipos (*types*), que definen nuevas estructuras de datos.

Valores (*values*), que son realizaciones (variables) de un tipo.

Macros, que se utilizan para cambiar la gramática ASN.1.

2.2.1.3 MIBS

Las bases de información de gestión (MIB), representan un conjunto de objetos gestionados de un recurso, que se publican para ofrecer interoperabilidad de gestión. Existen tipos MIB I y MIB II, están conformados con objetos de la torre de protocolos TCP/IP

2.2.1.3.1 MIBs Experimentales

Las MIB experimentales son consideradas en fase de desarrollo por los grupos de trabajo de Internet. existen MIBs para:

IEEE 802.4 Token Bus (RFC 1230)

IEEE 802.5 Token Ring (RFC 1231)

IEEE 802.3 Repeater Devices (RFC 1368)

Ethernet (RFC 1398)

FDDI (RFC 1285)

RMON (RFC 1271)

Bridges (RFC 1286)

2.2.1.3.2 MIBs Privadas

Las MIB privadas, corresponden a las MIBs de productos específicos, generadas por los distintos fabricantes, y añaden funcionalidad a las MIB estándar. Generalmente, los fabricantes las hacen públicas, poniéndolas accesibles por Internet (deposito común en venera.isi.edu). Por

ejemplo, se pueden encontrar MIBs para productos, de Cabletron, Synoptics, Proteon, ATT, Cisco.

2.2.1.4 Simple Network Management Protocol (SNMP)

El protocolo SNMP (RFC 1157) surge a partir del protocolo SGMP para gestión de *routers* IP. El *Simple Network Management Protocol* (SNMP) es un protocolo de aplicación, que ofrece servicios de gestión de red al conjunto de protocolos Internet. SNMP define una arquitectura basada en cliente-servidor. El programa cliente (llamado el gestor de red) realiza conexiones virtuales a un programa servidor (llamado el agente SNMP) ejecutando en un dispositivo de red remoto. La base de datos controlada por el agente SNMP, se denomina *Management Information Base* (MIB), y es un conjunto estándar de valores estadísticos y de control de status. SNMP permite también extensiones de esta MIB a agentes particulares para el uso de MIB privadas. Los mensajes enviados por el cliente (gestor de red) a los agentes SNMP están formados de identificadores de objetos MIB, junto con instrucciones, a fin de cambiar u obtener un valor. (Barba, 1999). En la figura 3 se puede observar cómo está constituida la arquitectura de SNMP.

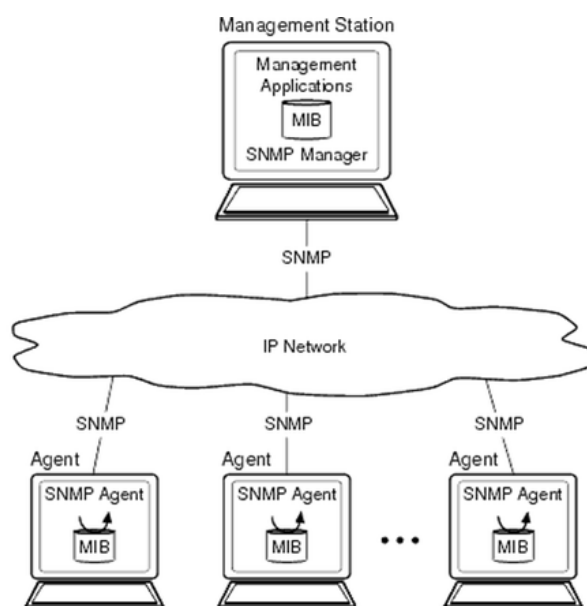


Figura 3. Arquitectura de SNMP

Fuente: Servicios TCP/IP. (2017). recuperado de <https://goo.gl/y1ybWm>

2.2.1.4.1 Operaciones de SNMP

El protocolo SNMP es un protocolo simple y de monitorización. A continuación, se definen los tipos de operaciones permisibles con sus objetos:

- **GetRequest:** petición de valores específicos de la MIB.
- **GetNextRequest:** brinda un medio para moverse por la MIB. Petición del objeto siguiente a uno dado de la MIB (orden lexicográfico).
- **GetResponse:** devuelve los valores solicitados por las operaciones anteriores.
- **SetRequest:** permite asignar un valor a una variable. Debido a posibles problemas de seguridad esta función suele estar desactivada.
- **Traps:** permite a los agentes informar de sucesos inusuales. (p.e. ColdStart, WarmStart, LinkDown, LinkUp, AuthenticationFailure, EGPNeighborLoss, EnterpriseSpecific).

Secuencia de transmisión de un mensaje SNMP

1. Construcción de la PDU, usando estructuras ASN.1.
2. La PDU se procesa por el servicio de autenticación junto a las direcciones correspondientes.
3. La entidad de protocolo construye el mensaje, consistiendo de una versión de campo, el nombre de la comunidad y el resultado del paso dos.
4. Este nuevo objeto ASN.1 es entonces codificado, usando las reglas de codificación básicas y pasado al servicio de transporte.

En la práctica, las autenticaciones no se invocan.

Secuencia de recepción de mensajes SNMP

1. Chequeo básico de la sintaxis del mensaje, descartándolo si es erróneo.
2. Verificación del número de versión. Se descarta el mensaje si no es coherente.
3. La entidad de protocolo pasa al usuario, el fragmento PDU del mensaje y las direcciones de traslado de emisor y receptor al servicio de autenticación.

3.1 Si la autenticación falla, el servicio de autenticación señala a la entidad de protocolos SNMP, para que genere una *Trap* y descarte el mensaje.

3.2 Si la autenticación tiene éxito, el servicio de autenticación devuelve la PDU en la forma de objeto ASN.1 definido en RFC 1157.

4. La entidad de protocolo hace un chequeo básico de la sintaxis del mensaje, descartándolo si es erróneo. En cualquier caso, la comunidad nombrada con la adecuada política de acceso SNMP seleccionada, finalmente procesa la PDU.

En la figura 4. Se puede observar cada paso de la secuencia de PDU en el protocolo SNMP

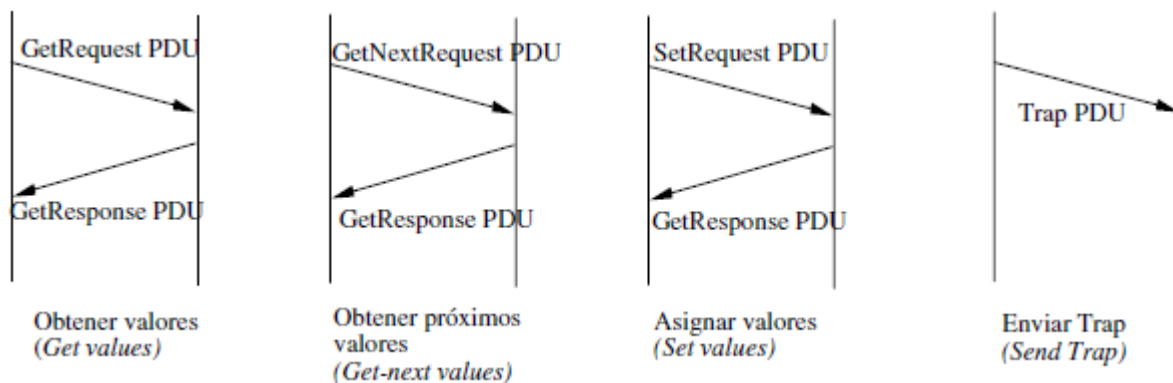


Figura 4. Secuencia de PDU en el protocolo SNMP
Fuente: (Barba, 1999)

2.2.1.4.2 Codificación para la transferencia de la información de gestión VER

La información de gestión se define a partir de la ASN.1, que establece, que para la codificación se utilizarán las BER (*Basic Encoding Rules*). Las BER permiten traducir una estructura de datos cualquiera en una secuencia de bytes y viceversa. Las BER son un algoritmo recursivo, que puede producir una secuencia de bytes a partir de cualquier valor ASN.1. El protocolo SNMP sólo utiliza un subconjunto de estas reglas.

Las BER codifican los tipos utilizando los siguientes tres campos de longitud arbitraria:

- *Tag*: indica el tipo de ASN.1

- *Length*: indica el tamaño de la codificación del valor que sigue
- *Value*: indica propiamente la codificación del valor.

2.2.1.4.3 Marco administrativo

Para una gestión adecuada con SNMP, se define un nombre de comunidad (*community*) que afecta, tanto al agente, como al conjunto de gestores que lo administran, y un perfil de comunidad que delimita las propiedades, así como el modo de acceso al sistema. De esta forma, la comunidad puede ser pública (*public*), es decir, de libre acceso. Esta información se almacena en cada MIB.

De esta forma, una comunidad es una relación entre gestor y agente, y el nombre de comunidad, es una serie de octetos transmitida en los mensajes SNMP. Para la determinación de políticas de autenticación y autorización, se trabaja con autenticación simple, en la que el nombre de comunidad se transmite en claro, de ahí la necesidad de ampliar la seguridad del entorno de gestión mediante otros servicios de seguridad. En cuanto a la autorización, cada comunidad tiene asociado una vista (conjunto de objetos), de forma que para cada objeto se define un modo de acceso: *read-only*, *read-write*. Finalmente, el nombre de comunidad junto al perfil de comunidad marca la política de acceso al sistema. (Barba, 1999)

2.2.1.5 SNMP v2

Este protocolo de gestión que se definió en 1993, es una versión más avanzada del SNMP. SNMPv2 Aporta una serie de ventajas respecto a la primera versión, entre las cuales pueden destacarse:

- Permite mayor eficiencia en la transmisión de información.
- Soporta mecanismos de seguridad como autenticación y cifrado frente al SNMP (no implementado).

- Admite la comunicación entre estaciones de gestión.
- Parte de un modelo de comunicaciones extendido considerablemente.
- Posibilita una señalización extendida de errores.
- tolera el uso de varios servicios de transporte.

El sistema basado en SNMPv2, resuelve varios de los problemas de su versión anterior, SNMP; sin embargo, su mayor complejidad está coartando su desarrollo. Los mensajes emitidos por la plataforma de gestión de red a los agentes SNMP, están integrados por identificadores de objetos MIB, junto con instrucciones, a fin de cambiar u obtener un valor. (Barba, 1999)

2.2.1.5.1 Operaciones de SNMPv2

El protocolo SNMPv2, que incluye los mensajes de la primera versión SNMP, dispone de los siguientes tipos de mensajes:

- GetRequest: petición de valores específicos de la MIB.
- GetNextRequest: proporciona un medio para desplazarse por la MIB. Petición del objeto siguiente a uno dado de la MIB (orden lexicográfico).
- GetBulkRequest: petición de múltiples valores.
- Response: devuelve los valores requeridos por las solicitudes anteriores gestor a gestor o agente a gestor).
- SetRequest: permite asignar un valor a una variable. Debido a posibles problemas de seguridad esta función suele estar desactivada.
- InformRequest: transmite información no solicitada (gestor a gestor).
- SNMPv2-Traps: permite a los agentes informar de sucesos inusuales (p.e. ColdStart, WarmStart, LinkDown, LinkUp, AuthenticationFailure, EGPNeighborLoss, EnterpriseSpecific). (Barba, 1999)

Sin embargo, el hecho de su incompatibilidad con la versión SNMP y la mayor complejidad añadida a las plataformas están desestimando su futura implementación.

La disconformidad en el consorcio sobre las recomendaciones de seguridad propuestas en SNMPv2, ha propiciado finalmente su incorporación en una nueva versión SNMPv3.

2.2.1.6 SNMP v3

El protocolo SNMPv3 es un desarrollo de la serie de modelos de gestión anteriores, las áreas a las que SNMPv3 va dirigidas son básicamente, mejorar la seguridad y la administración respecto a SNMPv2. Respecto a la estructura de la información de gestión, la SMI está dividida en tres partes: definiciones de módulos, definiciones de objetos y definiciones de notificaciones. Las definiciones de módulos (macros ASN.1: MODULE-IDENTITY) se utilizan para describir semánticamente los módulos de información. Para las definiciones sintácticas y semánticas de objetos se usan macros ASN.1: OBJECT-TYPE. Las definiciones de notificaciones usan macros NOTIFICATION-TYPE y describen transmisiones no solicitadas de información de gestión.

Entre los tipos de datos que se incorporan a la SMI (RFC 1902) están:

- IMPORTS para permitir la especificación de items que se usan en un módulo MIB, pero definidos en otro módulo MIB.
- MODULE-IDENTITY especifica una descripción e información administrativa para un módulo MIB.
- OBJECT-IDENTITY y los OID se asignan para especificar un valor OID.
- OBJECT-TYPE especifica el tipo de dato, estatus y la semántica de los objetos gestionados.
- SEQUENCE es un tipo que permite asignar los objetos de una lista en una tabla.
- NOTIFICATION-TYPE se especifica para construir una notificación de eventos.

En SNMPv3 se provee también, aumentar el mapeo de mensajes tipo SNMP a otros tipos de protocolos de transporte. Desde el punto de vista de arquitectura de gestión se extiende el nombrado de:

- Motores y aplicaciones.
- Entidades (proveedores de servicio tales como motores en agentes y gestores).
- Identidades (usuarios de servicio).
- Información de gestión, incluido soporte para múltiples contextos lógicos.

Los cinco tipos de aplicaciones que se prevé asociar con un motor SNMP son: generadores de comandos, receptores de comandos (generadores de respuestas), originadores de notificaciones, receptores de notificaciones y envío de *proxies*. Respecto a las mejoras en seguridad, SNMPv3 utilizará MD5 y algoritmos de Hash para firma digital y proteger contra la modificación de la información proporcionando integridad de datos, autenticación de origen y de usuario. (Barba, 1999).

2.3 Sistemas de Gestión

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. (Stallings, 2004, p.263)

El modelo de gestión de red que es usado en el protocolo SNMP (simple network manager system) incluye los siguientes elementos:

- Estación de gestión
- Agente de gestión
- Base de información de gestión
- Protocolo de gestión de red

2.3.1 Análisis comparativo de los sistemas de gestión

Entre los diferentes sistemas de gestión existentes, se han seleccionado los más adecuados para analizar sus características, de entre ZENNOS, PANDORA y NAGIOS se seleccionará uno para su implementación.

2.3.1.1 *Zennos*

Zenoss Core es un producto de código abierto de monitorización, que proporciona las funcionalidades necesarias, para gestionar eficazmente la configuración, la estabilidad, el rendimiento de las redes, servidores y aplicaciones a través de un único paquete de software integrado. (Zenoss, 2016)

2.3.1.2 *Pandora*

Pandora FMS es un software de monitorización, para gestión de infraestructura TI. Esto incluye equipamiento de red, servidores Windows y Unix, infraestructura virtualizada y todo tipo de aplicaciones. Pandora FMS tiene multitud de funcionalidades, lo cual lo convierte en un software de nueva generación que cubre todos los aspectos de monitorización necesarios. (Pandora, 2016)

2.3.1.3 *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. (Nagios, 2016)

Tabla 1. Comparativa de sistemas de Gestión

	Zenoss	Pandora	Nagios
Graficas	●	●	●
informes sla		●	●
Estadísticas	●	●	●
Agentes	●	●	●
Snmp	●	●	●
Syslog	●	●	●
scripts externos	●	●	●
Complementos (plugins)	●	●	●
Alertas	●	●	●
método de almacenamiento de datos	mysql	mysql	sql
Licencia	gpl	gpl	gpl
requerimientos de cpu y ram	alto	alto	medio

Fuente: Elaborado por el autor

En base a la tabla 1 comparativa el sistema de gestión más viable es el software Nagios. debido a que cumple con todas las características y no consume muchos recursos, teniendo como plataforma una máquina virtual.

2.3.2 Requerimientos de SGSI bajo la norma IEEE 29148:2011

En la Norma IEEE 29148 sección 9.5 REQUISITOS DE SOFTWARE se define el contenido a seguir de la especificación de requisitos de software, para la selección del sistema de gestión de seguridad de la información.

2.3.2.1 Propósito

El propósito que debe cumplir este software es, permitir el acceso mediante interfaz web con autenticación, en donde se pueda visualizar los valores de temperatura, humedad y humo en el cuarto del centro de Datos, y que permita la notificación de alertas al administrador.

2.3.2.2 Alcance

- Instalación del sistema de gestión en una máquina virtual de la plataforma Cloud de la universidad y configuración de autenticación.
- Activación del protocolo SNMP y definición del host a identificar para la recepción de información.
- Levantamiento de la página web, que permita la visualización de los datos de temperatura, humedad y humo, recopilados mediante los sensores instalados en el Centro de Datos.

2.3.2.3 Perspectiva del producto

Implementar un sistema de monitoreo de temperatura, humedad y humo, el que consta de un arduino que recopila información a través de sensores, y esa información la muestra en una pantalla LCD, de igual manera la misma información la envía mediante el protocolo SNMP al software de gestión, el cual tiene la capacidad de mostrar la información mediante una interfaz web, así como también pueda alertar mediante correo electrónico sobre algún evento no deseado en el interior de centro de datos.

2.3.2.4 Funciones del producto

Las funciones que tendrá el sistema de gestión serán:

- Recibir la información obtenida mediante el protocolo SNMP, para mostrarlos cuando el administrador los solicite.

- Mostrar una Interfaz web en donde se proporciona una visión general de los parámetros medidos con los sensores del sistema embebido.
- Despliegue de gráficos para que los administradores puedan observar estadísticas.
- Notificaciones a los contactos cuando ocurren fallas o variación en los límites establecidos de los valores censados.

2.3.2.5 Características de los usuarios

El administrador podrá ingresar al sistema utilizando una cuenta a la que se accede por autenticación y será el quien tenga privilegios de administración y configuración de los archivos, así como los contactos para el reporte de alertas.

2.3.2.6 Limitaciones

Las limitaciones pueden suceder debido a:

Mala conectividad o indisponibilidad de internet

Los datos mostrados serán con un pequeño intervalo de tiempo y no en tiempo real

Datos mostrados con un porcentaje mínimo de error.

2.3.2.7 Suposiciones y dependencias

Es necesario que la infraestructura del proyecto CLOUD implementado en el centro de datos de la FICA este activa

Se necesita una máquina virtual alojada en el servidor CLOUD con el sistema operativo CENTOS

Se requiere que la máquina virtual tenga conexión a internet y que pueda comunicarse con el sistema embebido.

2.3.2.8 Requisitos Complementarios

Entorno de escritorio en la máquina virtual Centos.

Navegador web servicio http en la máquina virtual Centos

2.3.2.9 *Requisitos Específicos*

Los requerimientos necesarios son los siguientes:

- El servidor web Apache 2.
- PHP.
- El compilador GCC y las librerías de desarrollo.
- GD Graphics Library versión

2.3.2.10 *Justificación*

Nagios contiene las características apropiadas, además es estable en funcionamiento y desempeño, cuenta con varios componentes que son necesarios para la monitorización, dispone de una amplia gama de plugins, desarrollados para usos específicos, así como también la alternativa de generar nuestro propio código para implementar en el sistema de monitoreo.

2.4 OpenNebula

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. (opennebula.org, 2017)

2.4.1 Funcionalidades de OpenNebula

Ofrece funciones en las dos capas de Data Center Virtualization y Cloud Infrastructure:

- **Administración de Virtualización de Data Center.** OpenNebula permite administrar la virtualización del centro de datos, consolidar servidores e integrar los activos de TI existentes para la computación, el almacenamiento y la creación de redes. OpenNebula

se integra directamente con los hipervisores (como KVM, Xen o VMware ESX) y tiene un control completo sobre los recursos físicos y virtuales, proporcionando funciones avanzadas de gestión de capacidad, optimización de recursos, alta disponibilidad y continuidad de negocio, posee funciones de administración y aprovisionamiento en la nube de OpenNebula.

- **Gestión de nubes.** OpenNebula permite proporcionar una capa de aprovisionamiento multi-arrendatario, similar a una nube encima de una solución de administración de infraestructura existente (como VMware vCenter). OpenNebula tiene características de provisión, elasticidad y multi-tenancy cloud como aprovisionamiento de centros de datos virtuales, federación de datacenter o cloud computing híbrido para conectar infraestructuras internas con clouds públicos, mientras que la infraestructura es administrada por herramientas conocidas para el manejo y operación de infraestructura.

2.5 Sistema Embebido

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

Partes del sistema embebido:

a) Entradas

- Son los datos que traemos de nuestro entorno de la vida real, para poder realizar una actividad en el sistema embebido.

b) Unidad central de procesamiento

- Recibe las señales de las entradas y las procesa para generar algún tipo de respuesta.
- Es el lugar donde se almacena nuestra programación.

c) Salidas

- Son las formas auditivas, visuales, motoras, que generan las respuestas de la unidad central de procesamiento.

2.5.1 Potenciómetro

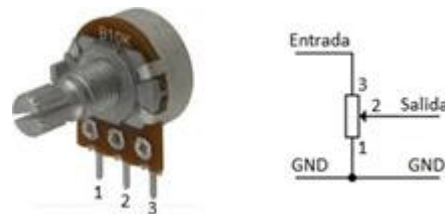


Figura 5. Esquema del potenciómetro
Fuente: <https://goo.gl/oLjrgl>

Un potenciómetro 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.

2.5.2 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.

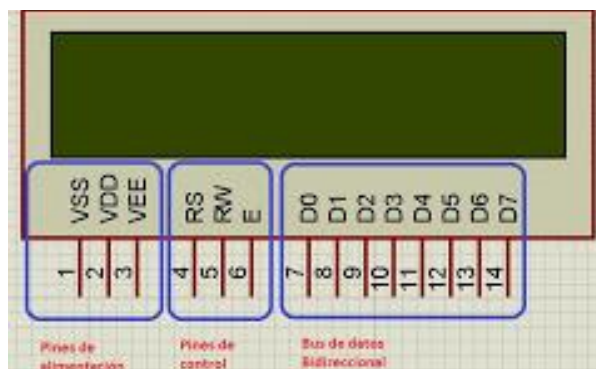


Figura 6. Pantalla LCD
Fuente: (<http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>, s.f.)

Pines de alimentación:

Vss: Gnd

Vdd: +5 voltios

Vee: corresponde al pin de contraste, regulable con un potenciómetro de 10K conectado a Vdd.

Pines de control:

RS: es el pin de selección de registro de control de datos (0) o registro de datos (1). El pin RS trabaja paralelamente a los pines del bus de datos.

RW: Es el pin de Escritura (0) o de Lectura (1). Permite escribir o leer un dato en la pantalla.

E: Es el pin Enable o de habilitación. Si E (0) quiere decir que el LCD no está activado para recibir datos, pero si E (1) está activo y podemos escribir o leer desde el LCD.

Pines de Bus de datos:

El Bus de datos bidireccional engloba desde los pines D0 a D7. Para realizar la comunicación con el LCD se realiza utilizando los 8 bits del bus de datos (D0 a D7) o empleando los 4 bits más significativos del bus de datos (D4 a D7). (todoelectrodo, 2013).

2.5.3 Sensores DHT de temperatura y humedad

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. En la tabla 2 se muestra una comparativa entre los dos sensores.

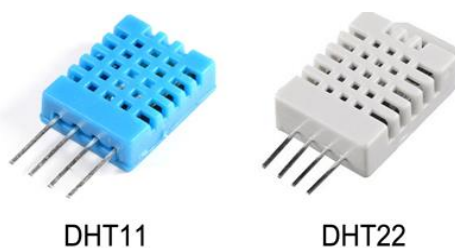


Figura 7. Sensores DHT
Fuente: (datasheet4u, 2017)

Tabla 2. Características de sensores DHT

DHT11		DHT22	
Alimentación	3Vdc a 5Vdc	Alimentación	3Vdc a 5Vdc
Rango de medición de temperatura	0 a 50 °C	Rango de medición de temperatura	-40 a 125 °C
Precisión de medición de temperatura	±2.0 °C.	Precisión de medición de temperatura	±0.5 °C.
Rango de medición de humedad	20% a 80% RH.	Rango de medición de humedad	0% a 100% RH.
Precisión de medición de humedad	+5% RH	Precisión de medición de humedad	+2 a 5% RH
Tiempo de censado	1 segundo	Tiempo de censado	2 segundos

Fuente: (datasheet4u, 2017)

2.5.4 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 varía 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. (naylampmechatronics.com, s.f.) a continuación, en la tabla 3 se muestra una comparativa de las características de los sensores.



Figura 8. Sensores de humo MQ
Fuente: (yapo, 2017)

Tabla 3. Características de los sensores MQ

MQ-2		MQ-4	
Alimentación	5V DC ó AC	Alimentación	5V DC ó AC
Corriente de operación	160 mA	Corriente de operación	160 mA
Disipación de potencia	750mW	Disipación de potencia	750mW
Temperatura de funcionamiento	de -20 a 50 °C	Temperatura de funcionamiento	de -20 a 50 °C
Rango de medición	300-10000ppm	Rango de medición	200-10000ppm
Tiempo de respuesta	<10 segundos	Tiempo de respuesta	<10 segundos
Precisión	95%	Precisión	95%

Fuente: Adaptado del datasheet de los sensores

2.6 Situación Actual

La Facultad de Ingeniería en Ciencias Aplicadas FICA, cuenta con un espacio físico adecuado para el alojamiento de servidores y equipos de red, con lo que satisface las necesidades de conectividad a los usuarios, teniendo en cuenta que el diseño e implementación del Centro de Datos es reciente, no se cuenta con un sistema de alerta temprana, que ayude a prevenir posibles problemas en cuanto a los parámetros de temperatura humedad y detección de humo, habiendo antecedentes como el incendio ocurrido en el año 2015, en donde se pudo evidenciar las falencias en cuanto a prevención ante algún problema, se planteó el sistema de monitoreo para el Centro de Datos ubicado en la facultad FICA.

2.6.1 Centro de Datos

El Centro de Datos es un área física de prestaciones definidas donde convergen equipos de networking, para ofrecer servicios IT. (Pacio, 2014), en su libro Data centers Hoy, lo define como “un espacio con determinadas características físicas especiales de refrigeración, protección y redundancia cuyo objetivo es alojar todo el equipamiento tecnológico de la compañía, brindando seguridad y confiabilidad” (pág. 1).

Está dividido en 4 subsistemas:

- Subsistema de arquitectura
- Subsistema eléctrico
- Subsistema mecánico
- Subsistema de telecomunicaciones

2.6.1.1 Subsistema de Arquitectura

Es el espacio físico, ambiental y de seguridad del edificio en donde se colocará el Centro de Datos. En la Facultad, el Centro de Datos se encuentra ubicado en el primer piso, al fondo del pasillo principal, la puerta de acceso a esta área se encuentra ubicada dentro de la oficina CIERCOM, lo que funcionaría como filtro para llegar al mismo. El Centro de Datos cuenta con un piso falso a una altura de 35 cm, de esta manera cumple con la norma de altura de los racks y el nivel del techo de (2.6 m).

2.6.1.2 Subsistema Eléctrico

Suministra y distribuye de manera correcta y eficiente la alimentación eléctrica tanto de AC como en DC al equipamiento activo y pasivo del Centro de Datos.

En la parte externa de la facultad se cuenta con una caja de revisión, desde donde se extienden los cables eléctricos pasando por el piso técnico y terminan en el tablero principal.

2.6.1.2.1 Tablero general de distribución

Instalado a una altura de 2m del piso técnico y sus características son:

- Dispone de 12 espacios independientes para circuitos, incluido breaker de protección.
- Voltaje de 127/220[VAC]
- Límite de capacidad máxima de carga en corriente 125[A] en terminales principales

2.6.1.2.2 Sistema de alimentación interrumpida (SAI)

El centro de datos cuenta con un sistema de alimentación ininterrumpida, el cual brinda un respaldo en caso de alguna falla del suministro eléctrico, el sistema UPS es el encargado de aportar el flujo eléctrico provisional a todo el conjunto TIC.

El sistema de UPS instalado cuenta con las siguientes características:

- Fuente nominal: 10KWA
- Potencia nominal: 7Kw
- Voltaje nominal de salida: 230 V
- Tiempo de respaldo de energía con 75% de la carga nominal: 9 minutos
- Temperatura de operación: intervalo de 0 y 40 °C

2.6.1.3 Subsistema Mecánico

Hace referencia a los mecanismos a instalar como refrigeración, sistema de control de acceso, mecanismos anti fuego, de prevención y respuesta ante desastres del CDP.

2.6.1.3.1 Puerta de seguridad

El ingreso al Centro de Datos está restringido por una puerta de seguridad realizada en estructura metálica de 4cm de espesor con una mirilla de 5 láminas de vidrio, la estructura es totalmente sellada con lo que se evita la fuga de aire refrigerado.

En la puerta de ingreso se encuentra instalado:

- control biométrico de acceso
- barra anti-pánico
- brazo cierra-puertas
- cerradura electromagnética.

2.6.1.3.2 Aire acondicionado

Para la refrigeración interna del centro de datos, se cuenta con un sistema tipo “confort” cuyo electrodoméstico se encuentra ubicado a una altura de 2m en la zona del pasillo caliente, tiene las siguientes características:

- Refrigerante: r410a/1130 gr
- Capacidad de enfriamiento 24000 BTU/h
- Potencia de consumo (nominal):2170 w
- Presión de diseño: alta 550PSIG/BAJA 340 psig
- Clase de resistencia de unidad externa: IP2

2.6.1.3.3 Cámaras de seguridad

La video-vigilancia interna del Centro de Datos, está a cargo de una cámara compatible con el protocolo IP (Soporte para conexiones 802.3x), manejo del estándar 802.11 con seguridad WEP, WPA, WPA2. Dicha cámara está instalada a 2,30 m de altura encima de los tableros eléctricos, se encuentra operativa enfocando la puerta de ingreso.

2.6.1.4 Subsistema de Telecomunicaciones

Brinda Soporta y garantiza los servicios de red interna (local) y externamente (internet)

Los espacios de telecomunicaciones se encuentran establecidos de la siguiente manera:

Rack # 1, izquierdo

Rack de telecomunicaciones y respaldo de la red de fibra óptica de la Universidad

Acometidas eléctricas del CDP conjuntamente bandeja de ODF.

Switch CORE de alta disponibilidad

Patch panel de conexión cruzada para el cableado horizontal

Rack #2, central

Rack para servidores de proyectos de desarrollo Cloud.

Switch de distribución de cableado horizontal a cada planta

Rack #3, derecho

Rack de servidores de la facultad

Switch de distribución de cableado horizontal a cada planta.

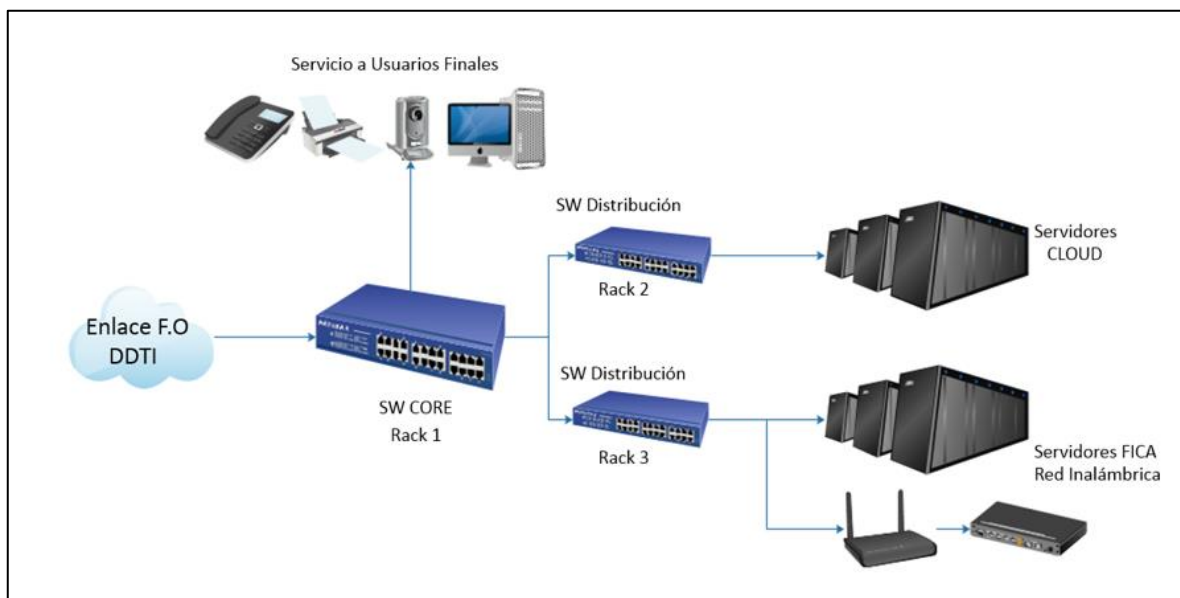


Figura 9. Arquitectura de la red FICA
Fuente: Diseño del centro de datos FICA

En la figura 9, se puede observar la arquitectura de la red de la Facultad de Ingeniería en Ciencias Aplicadas. En base a la información recopilada actualmente el centro de datos cuenta con los siguientes equipos.

En la tabla 4 se muestra de una forma organizada el equipamiento de equipos existentes en el centro de datos.

Tabla 4. Equipamiento TIC del centro de datos FICA

EQUIPO	MARCA MODELO (UBICACIÓN)	FUNCIÓN	No. SERIE	INVENTARIO UTN	RESPONSABLE	DIRECCIÓN IP	CARACTERÍSTICAS
SERVIDORES	IBM System x3500 M4 (RACK 3)	Repositorio DSpace	KQ6M81T	1410103.327.0079	Ing. Fernando Garrido	10.24.8.61 172.16.3.61	RAM: 7,6 GB Procesador: Intel Xeon (R) CPU E5-2630 2,30 GHz x 12 HDisk: 610 GB OS: CentOS 6.5
	IBM System x3500 M4 (RACK 3)	Reactivos	KQ6M81V	1410103.327.0078	Ing. Fernando Garrido	10.24.8.66 172.16.3.66	RAM: 7,6 GB Procesador: Intel Xeon (R) CPU E5-2630 2,30 GHz x 12 HDisk: 135 GB OS: CentOS 6.5
	IBM System x3200 M2 (RACK 2)	DHCP FICA (Inactivo)	103QCPYOF560	1410107.001.5050	Ing. Fernando Garrido	-----	RAM: Procesador: HDisk: OS:

HP Proliant DL360 G9 (RACK 2)	Proyecto Cloud-Open Stack	MXQ51704F7	1410107.018.02017	Ing. Edgar Jaramillo	10.24.8.76 172.16.3.76	RAM: Procesador: HDisk: OS: Ubuntu Server 14.043 LTS
HP Proliant ML150 G5 (RACK 2)	Proyecto Cloud SDN	MXS8460A5J	1410107.001.728	Ing. Edgar Jaramillo	10.24.8.77 172.16.3.77	RAM: 4.8 GB Procesador: Intel Xeon(R) CPU E5405 2.0GH x 4 Almacenamiento: 232 GB OS: Ubuntu Server 14.043 LTS
HP Proliant DL360 G9 (RACK 2)	Proyecto Cloud- Eucalyptus	MXQ51500L9	1410107.018.02015	Ing. Edgar Jaramillo	10.24.8.75 172.16.3.75	RAM: 32 GB Procesador: HDisk: OS: Ubuntu Server 14.043 LTS
HP Proliant DL360 G9 (RACK 2)	Proyecto Cloud-Nebula	MXQ51704F9	1410107.018.02016	Ing. Edgar Jaramillo	10.24.8.74 172.16.3.74	RAM: Procesador: HDisk: OS: Ubuntu Server 14.043 LTS
IBM System x3250 M3	SIN SERVICIO	KQCMXW0	Etiqueta rota	Ing. Edgar Jaramillo	Ninguna	RAM: Procesador:

	(RACK 2)						HDisk: OS:
	IBM System x3650 M3	CISIC	KQ14RFK	Etiqueta rota	Ing. Edgar Jaramillo	-----	RAM: Procesador: HDisk: OS: CentOS
	PC tipo "Clon"	Administración biométricos FICA	Modelo del sistema:H81M-S1	NO TIENE	Ing. Carlos Vásquez	172.16.7.24	RAM: 4 GB Procesador: Core (TM) i3-4150 CPU 3.5 GHz Almacenamiento: OS: Windows 7 Profesional
	HP Proliant ML150 G5	Servicio de encuestas y evaluación- OPINA	MXS8460A5P	1410107.001.727	Ing. Edgar Jaramillo	10.24.8.67 172.16.3.67	RAM: 4.8 GB Procesador: Intel Xeon (R) CPU E54405 GHz x 4 Almacenamiento:150 GB OS: Ubuntu 12.10
SWITHCS	CISCO Catalyst 4506-E	Enlace Principal Distribución de Red de la Facultad	-----	Etiqueta rota	Ing. Vinicio Guerra		Puertos: Características de Capa 3 administrable
	CISCO Linksys SR224G	Distribución de red	REP20FB00241	No tiene registro UTN	Ing. Edgar Jaramillo		Puertos: 24
	Cloud Router Switch Mikrotik	Router de red inalambrica	-----	No tiene registro UTN	Ing. Edgar Jaramillo		Puertos: 24 puertos

	CAS125-24G-15AM					
	Switch QPCOM QP-1240R	Switch de Distribución de conexiones AP – red inalámbrica FICA	121005919	No tiene registro UTN	Ing. Edgar Jaramillo	24 puertos fast Ethernet 10/100/1000 Mbps
	3COM	Distribución de red	73MF4XD03C0A 0	1410107.031.0001	Ing. Edgar Jaramillo	----- Puertos: 24
OTROS EQUIPOS	EQUIPO		FUNCIÓN		No. SERIE	CARACTERÍSTICAS
	Rack 1		Distribución switch CORE y cableado de datos FICA		-----	42 UR, tipo Gabinete, metálico, color negro
	Rack 2		Distribución servidores Proyecto Cloud		-----	36 UR, tipo Gabinete, metálico, color negro

Fuente: (Diseño de la infraestructura física de un Data Center TIER I basado en el estándar TIA 942, para la Facultad de Ingeniería de la Universidad Técnica del Norte, 2016)

En base a la información descrita en la tabla 4, se han obtenido los valores de temperatura y humedad en sus respectivos datasheets, de los equipos localizados en el centro de datos de la facultad, en la tabla 5 observamos los valores obtenidos de los datasheet de los equipos.

Tabla 5. Rangos de temperatura y humedad establecidas por fabricante

EQUIPO	TEMPERATURA ON HUMEDAD RELATIVA	
IBM System x3500 M4	10.0 a 28.0 °C	8% a 80%
IBM System x3200 M2	10.0 a 28.0 °C	8% a 80%
IBM System x3250 M3	10.0 a 28.0 °C	8% a 80%
HP Proliant DL360 G9	10.0 a 27.0 °C	8% a 90%
HP Proliant ML150 G5	10.0 a 27.0 °C	8% a 90%
HP Proliant DL360 G9	10.0 a 27.0 °C	8% a 90%
CISCO Catalyst 4506-E	0 a 40.0 °C	10% a 90%
CISCO Linksys SR224G	0 a 40.0 °C	10% a 90%
Cloud Router Switch Mikrotik CAS125-24G-15AM	-35 a 65 °C	5 % a 85%
Switch QPCOM QP-1240R	0 a 40.0 °C	10% a 90%
3 COM	0 a 45.0 °C	10% a 90%
Adaptador CISCO POE80U-580G-CR	-40 a 55.0 °C	5% a 90%
UPS	0 a 40.0 °C	

Fuente: datasheet de los equipos

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.

CAPÍTULO III




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. En la tabla 6 se muestra la comparativa de las características de las placas seleccionadas para el análisis.

Tabla 6 : Comparativa sistemas embebidos

PLACA	Arduino YUN	Raspberry PI 1 B+	Intel Galileo
			
Voltaje	5 voltios	5 voltios	5 voltios
Procesador	Atheros AR9331 400MHZ	ARM 700 MHZ	Intel Quark SoC 400MHz
Pines E/S Y analógicos	20	40	20
RJ-45	1 Ethernet	1 Ethernet	1 Ethernet

USB	1	4	1
Audio	NO	1 conector	1 conector
Video	NO	1 entrada MIPI 1 salida RCA	NO
PCIe	NO	NO	1 conector
Micro SD	1 slot	1 slot	1 slot

Fuente: Elaborado por el autor

3.2 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, restándole 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.

3.2.1 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. (ARDUINO, 2016)

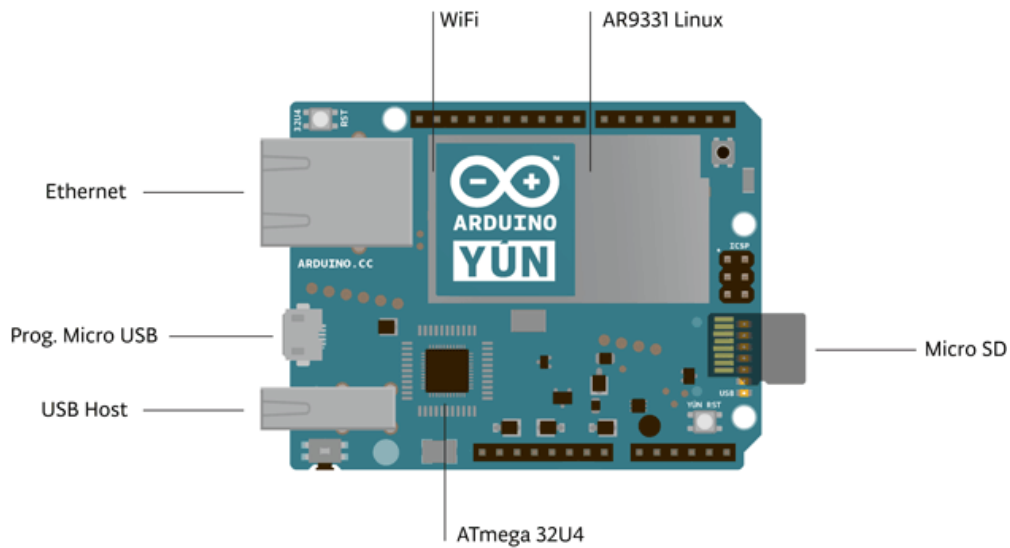


Figura 10. Placa Arduino YUN
 Fuente: <https://www.arduino.cc/en/uploads/Guide/YunParts.png>

El Arduino YUN se distingue de otras placas en que se puede comunicar con la distribución de Linux a bordo, que ofrece un ordenador en red de gran alcance con la facilidad de Arduino.

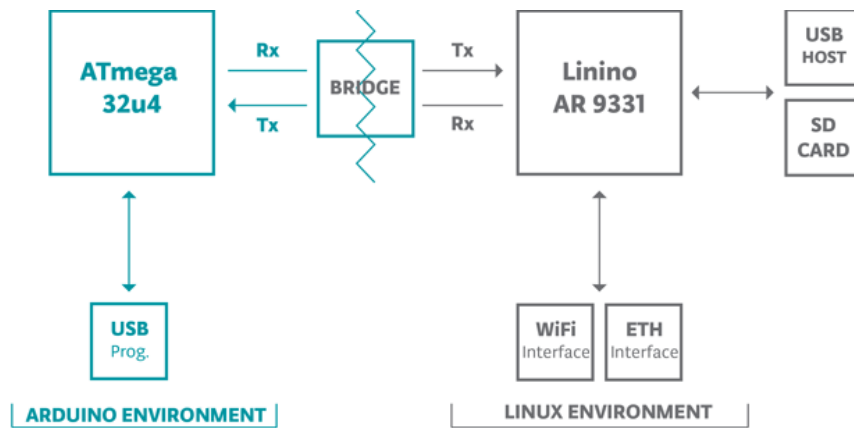


Figura 11. Procesador Atmega y procesador LININO
 Fuente: <https://www.arduino.cc/en/uploads/Guide/BridgeBlockDiag.png>

La biblioteca Bridge facilita la comunicación entre los dos procesadores, y permite comunicarse con las interfaces de red. El anfitrión del USB, interfaces de red y la tarjeta SD no están conectados a la 32U4, pero sí al AR9331. En la tabla 7 y 8 podemos observar las características del microcontrolador Atmega y micro procesador Linux correspondientemente.

Tabla 7. Microcontrolador Atmega

Microcontrolador	ATmega32U4
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

Fuente: (Arduino, 2017)

Tabla 8. Microprocesador Linux

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

Fuente: (Arduino, 2017)

El Arduino YUN tiene una serie de paquetes para la comunicación con un ordenador, otro Arduino, u otros microcontroladores. El ATmega32U4 proporciona una comunicación serie dedicado UART TTL (5V). El microcontrolador 32U4 también permite la comunicación en serie (CDC) a través de USB y aparece como un puerto COM virtual para el software en el ordenador. El chip también actúa como un dispositivo completo USB 2.0 de alta velocidad. El software de Arduino incluye un monitor de serie que permite mostrar los datos de texto simples para ser emitidos hacia y desde la placa Arduino. Una biblioteca Software Serial permite la comunicación en serie en cualquiera de los pines digitales del Yun. Pines 0 y 1 deben evitarse ya que son utilizados por la biblioteca de puente. Las interfaces Ethernet y Wi-Fi a bordo están expuestos directamente al procesador AR9331. Para enviar y recibir datos a través de ellos, utilizar la biblioteca de puente. La tarjeta se puede programar con el software Arduino (IDE). Los ATmega32U4 en el Arduino Yun viene precargado con un cargador de arranque que le permite cargar nuevo código a ella sin el uso de un programador de hardware externo. Se comunica mediante el protocolo AVR109. (Arduino, 2017).

3.2.2 IDE de Arduino

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

3.2.3 Selección de sensor de temperatura y humedad

En base a la información descrita en la tabla 2, se puede evidenciar que los dos sensores varían un poco en sus características de funcionamiento, el sensor DHT 22 es un sensor de mayor precisión en la medición y soporta mayores condiciones de operatividad, el sensor DHT11 es un sensor de precisión adecuada y soporta condiciones de funcionamiento, a las que se encuentran en el centro de datos, sin embargo, en base a la investigación realizada, se puede mencionar que el sensor

DHT11 es accesible en tiendas de la localidad a diferencia del sensor DHT22 que solo se los encuentra en la capital.

3.2.4 Selección de sensor de humo

En base a la información descrita en la tabla 3, se puede observar que los dos sensores son de características similares, sin embargo, en base a la investigación realizada, se puede mencionar que el sensor MQ-2 está elaborado específicamente para la detección de humo, así como su disponibilidad es accesible en tiendas de la localidad a diferencia del sensor MQ-4.

3.3 Diagrama de bloques del sistema de monitoreo

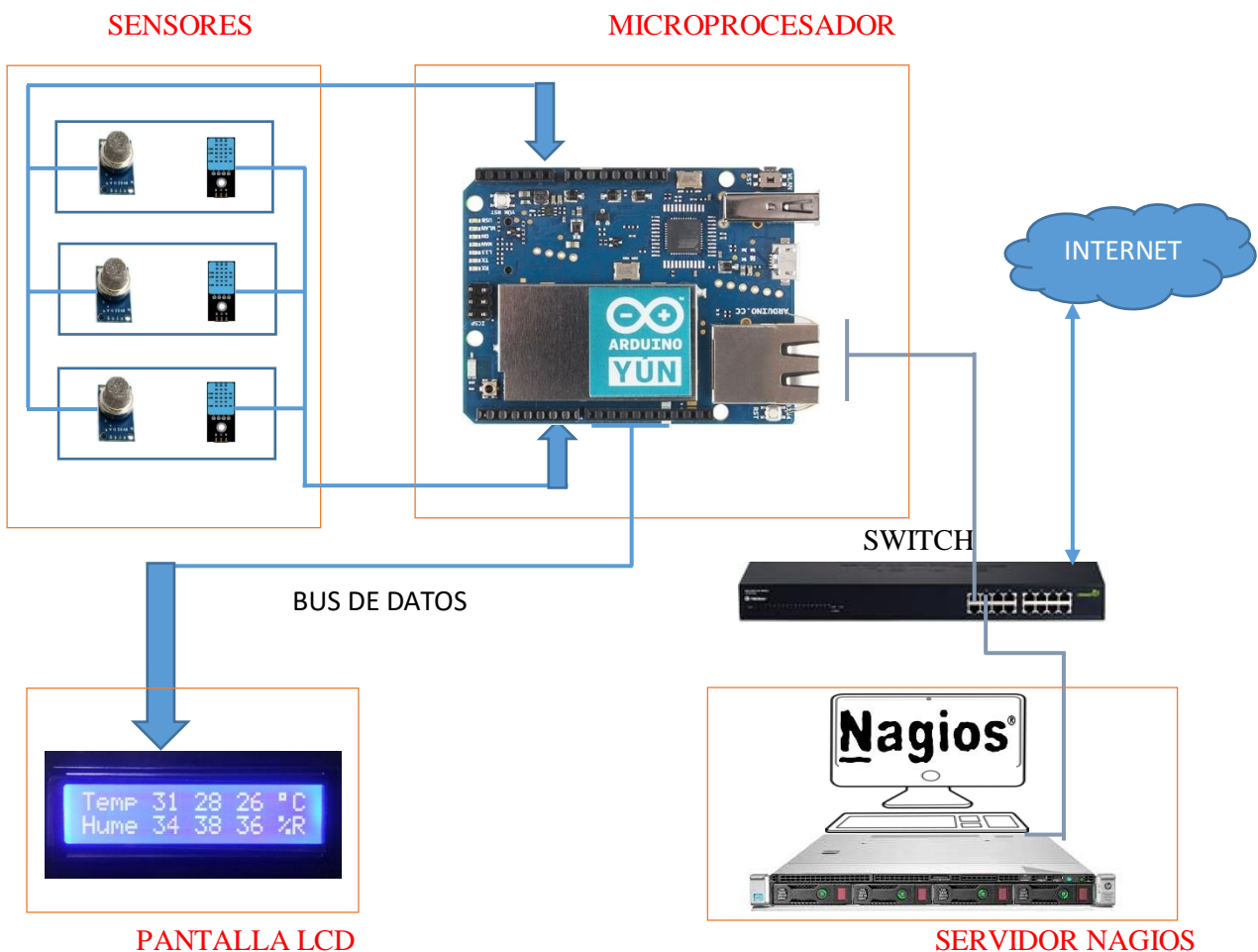


Figura 12. Diagrama del sistema de monitoreo
Fuente: Elaborado por el autor

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.

3.4 Lectura de temperatura y humedad con el sensor DHT11

Para realizar la lectura de temperatura y humedad utilizamos el sensor DHT11, es un sensor muy versátil que permite leer las dos variables simultáneamente, la conexión del sensor al arduino se lo realiza a un pin digital del arduino como se muestra en la figura 13.

En el sensor se muestran las letras V, D, G en cada pin, que significan:

V: es el pin que va conectado a voltaje

D: es el pin de transmisión de datos

G: es el pin que va conectado a tierra

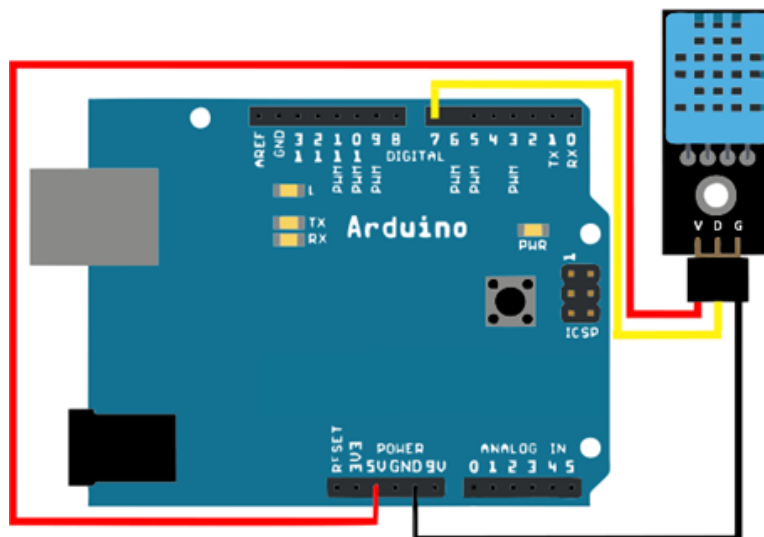


Figura 13. Conexión de sensor dht11 al arduino
Fuente: Elaborado por el autor

El desarrollo del código se lo realiza en el entorno de desarrollo integrado de arduino (interfaz gráfica IDE de arduino), para leer las variables de temperatura y humedad que recopila el sensor se muestra a continuación.

```
#include "DHT.h"           // se incluye la librería dht

#define DHTPIN 7           // se define el pin en el arduino al que va conectado el pin da datos

#define DHTTYPE DHT11     // se define el tipo de sensor que se usara

DHT dht(DHTPIN, DHTTYPE); // definimos la variable dht

void setup() {

  Serial.begin(9600);      // inicializamos la comunicación serial

  dht.begin();            // inicializamos la variable dht

}

void loop() {

  delay(1000);            // lectura del sensor durante cada segundo

  int t = dht.readTemperature(); //Leemos la temperatura en grados Celsius

  int h = dht.readHumidity(); //Leemos la Humedad Relativa

  Serial.print("Temperatura: "); // se imprime la palabra temperatura

  Serial.print(t);        // se imprime el dato de temperatura

  Serial.print(" C ");    // se imprime la letra C

  Serial.print("Humedad "); // se imprime la palabra humedad

  Serial.print(h);        // se imprime el dato de humedad

  Serial.println(" %HR "); // se imprime %HR

}
```


3.4.1 Visualización de lectura de temperatura y humedad por comunicación serial

Para visualizar los datos obtenidos a través del sensor, abrimos la conexión mediante comunicación serial y se desplegara el grafico de la figura 14. En donde nos indica los valores de temperatura y humedad, que el arduino está recopilando en ese momento de la lectura.

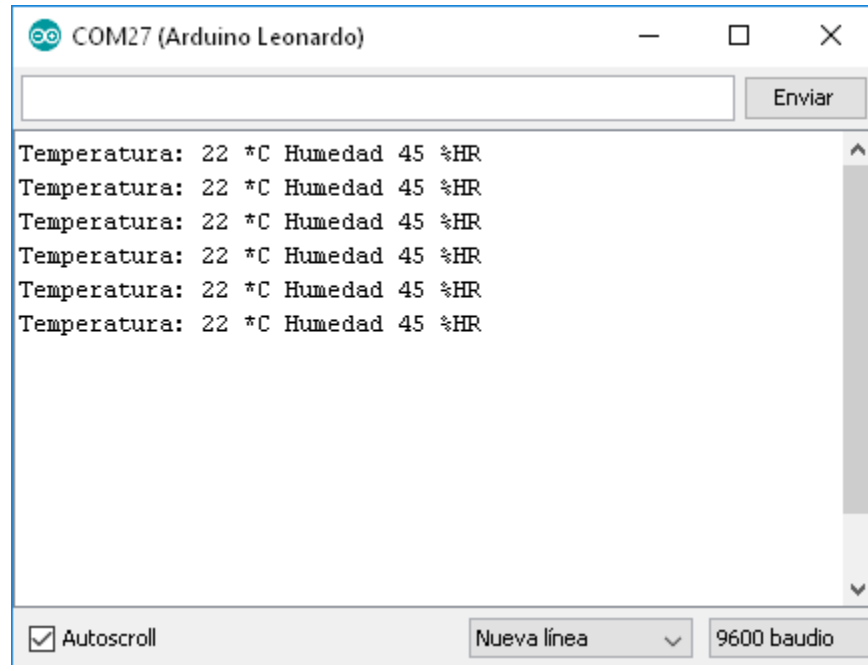


Figura 14. Visualización de valores de temperatura y humedad.
Fuente: Elaborado por el autor

3.5 Lectura de humo con el sensor MQ2

EL sensor MQ-2, es un sensor capaz de distinguir varios tipos de gases entre ellos el humo, son fáciles de implementar con la placa arduino, este sensor puede realizar lecturas tanto de forma analógica como digital, sin embargo, para una mejor precisión se utilizará el método de lectura analógica, la conexión se realiza a un pin analógico del arduino, como se muestra en la figura 15.

En el sensor la distribución de pines es de la siguiente manera:

V: pin para conectar al voltaje de arduino

G: pin para conectar a tierra del arduino

D0: pin de salida digital al arduino

A0: pin de salida analógica al arduino

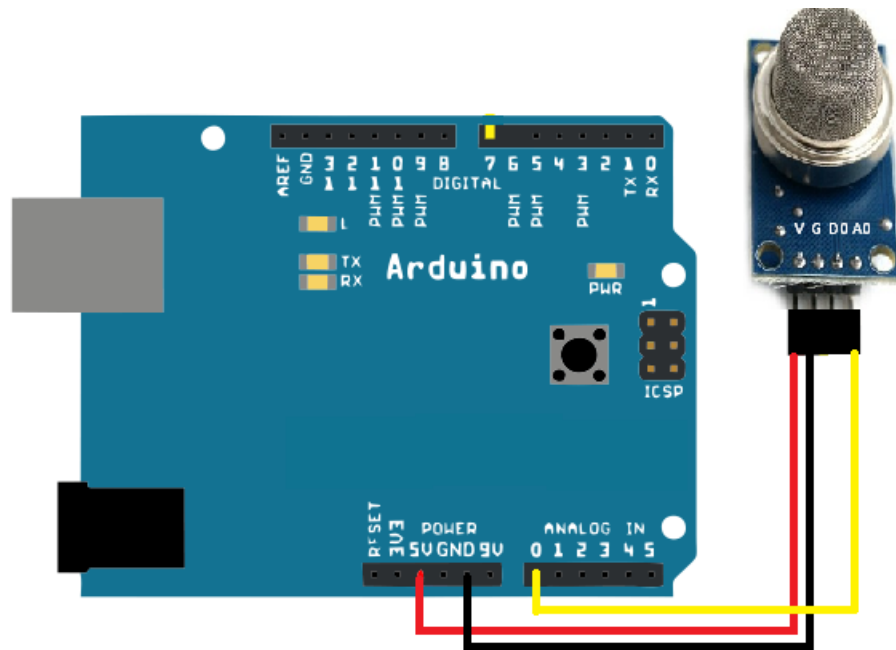


Figura 15. Conexión del sensor MQ-2 al arduino
Fuente: Elaborado por el autor

El desarrollo del código se lo realiza en el entorno de desarrollo integrado de arduino (interfaz gráfica IDE de arduino), con el objetivo de leer el dato analógico que se genera a partir del calentamiento del circuito interno del sensor, el que hace variar su resistencia. El código se muestra a continuación.

```
void setup() {  
  Serial.begin(9600); // iniciamos la comunicación serial  
}  
  
void loop() {  
  int adc_MQ = analogRead(A0); //Lemos la salida analógica del MQ
```

```
float voltaje = adc_MQ * (5.0 / 1023.0); //Convertimos la lectura en un valor de voltaje

Serial.print("adc:"); // imprime la palabra adc

Serial.print(adc_MQ); // imprime el dato obtenido de la lectura

Serial.print(" voltaje:"); // imprime la palabra voltaje

Serial.println(voltaje); // imprime el dato del voltaje

delay(100); // retardo de 1 segundo

}
```

3.5.1 Visualización de lectura de datos correspondiente a humo por comunicación serial

Para visualizar los datos obtenidos a través del sensor, abrimos la conexión mediante comunicación serial. Se desplegará una ventana donde mostrará el valor analógico que censa en ese momento, en la figura 16 se muestra la salida de valores analógicos detectados por el sensor.

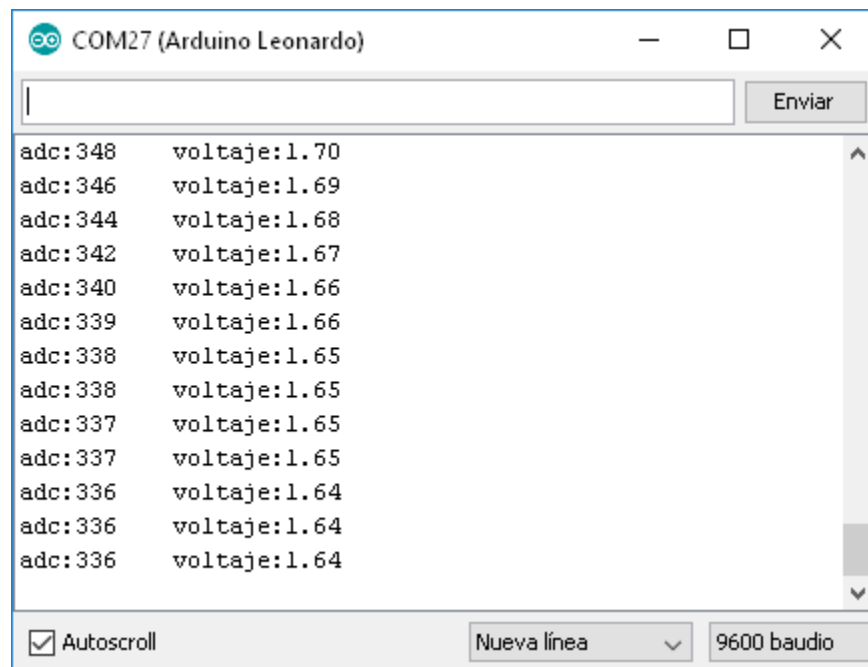


Figura 16. Visualización de valores del sensor de humo.

Fuente: Elaborado por el autor

3.6 Conexión de pantalla LCD al arduino

Para realizar la conexión de la pantalla LCD al arduino, es necesario definir mediante líneas de código, en donde declararemos los pines que funcionarán como salidas del arduino que corresponderán a cada pin del LCD. La conexión de pines del LCD se muestra en la figura 17.

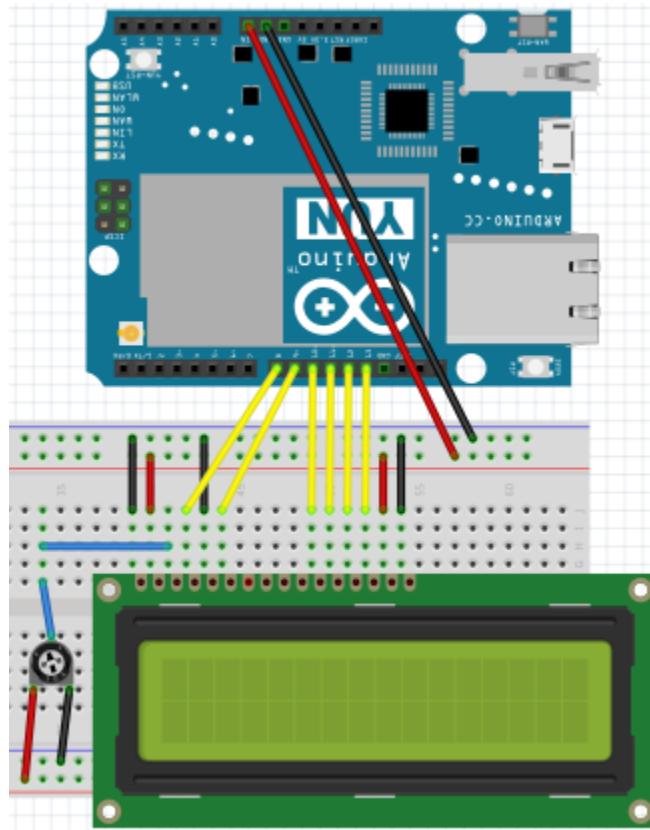


Figura 17. Conexión de pantalla LCD al arduino.

Fuente: Elaborado por el autor

Para realizar la visualización en la pantalla LCD en el código se debe incluir la librería <LiquidCrystal.h>, definir los pines del arduino que irán conectados a los pines del LCD, e inicializar la LCD con el número de columnas y filas (16,2) correspondientemente, para imprimir los símbolos o letras en alguna ubicación se indica en que fila y columna de la LCD

Desarrollo del código para la conexión del LCD al arduino.

```

#include <LiquidCrystal.h> //incluye libreria LCD

LiquidCrystal lcd(8,9,10,11,12,13); // (rs , enable, pines D4 D5 D6 D7)

void setup() {

  lcd.begin(16,2); //inicializamos la lcd

}

void loop() {

lcd.setCursor(0,0); lcd.print("Temp"); //imprime en la posición 0,0 del LCD la palabra Temp.

}

```

En la tabla 9 se indica las conexiones pin a pin LCD – Arduino.

Tabla 9. Conexión pines LCD-ARDUINO

# DE PIN	PINES LCD
1	VSS ----- conectado a tierra
2	VDD ----- conectado a voltaje
3	V0 ----- conectado a potenciómetro
4	RS ----- conectado a pin 8 en arduino
5	RW ----- conectado a tierra
6	E ----- conectado a pin 9 en arduino
11	D4 ----- conectado a pin 10 en arduino
12	D5 ----- conectado a pin 11 en arduino
13	D6 ----- conectado a pin 12 en arduino
14	D7 ----- conectado a pin 13 en arduino
15	A ----- conectado a voltaje
16	K ----- conectado a tierra
NOTA:	El resto de pines del LCD no se conectan, el pin intermedio del potenciómetro se conecta al pin de V0, los otros voltaje y tierra.

Fuente: Elaborado por el autor

3.7 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

}
```

Con el código escrito y grabado en el arduino, pasaremos a comprobar que el archivo “script.sh” se encuentre en la dirección /tmp, para ello ingresamos a la red inalámbrica del arduino y

accedemos mediante PUTTY (software para acceder mediante el protocolo SSH) a la dirección 192.168.240.1 puerto 22. Como lo muestra la figura 18.

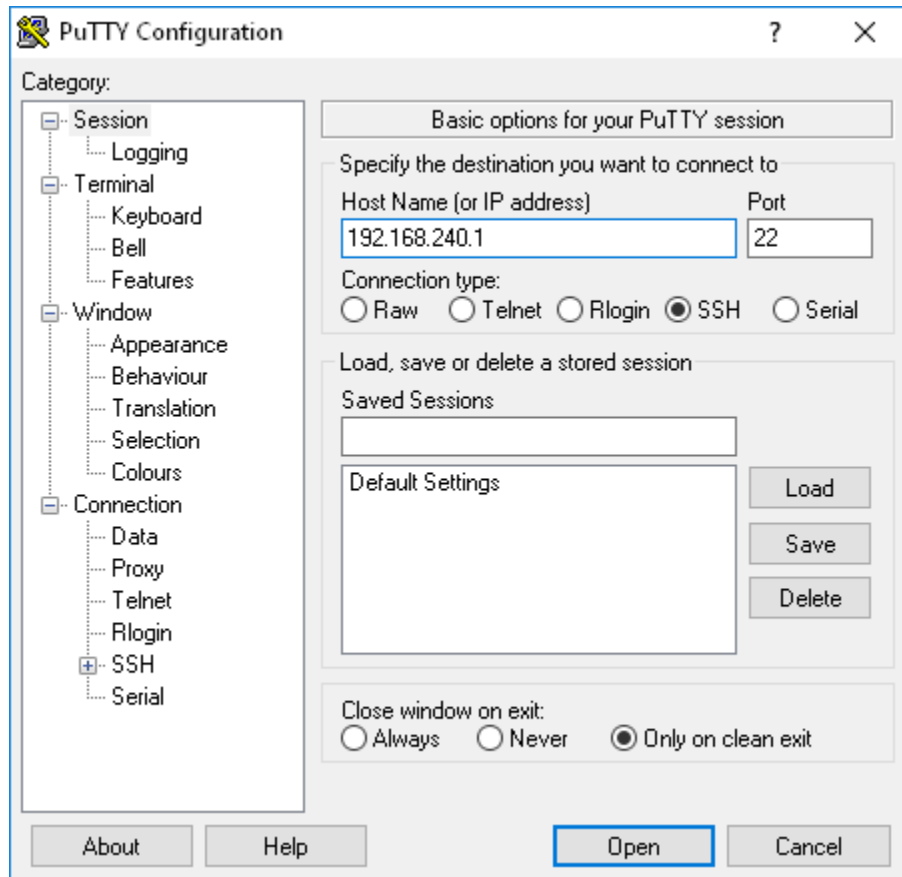


Figura 18. Acceso a OpenWrt mediante putty.
Fuente: Elaborado por el autor

Una vez ingresado nos autenticamos con las credenciales por defecto, (mirar figura 19). Y nos dirigimos mediante comandos de navegación en linux a la dirección /tmp en donde ubicaremos el archivo script.sh.

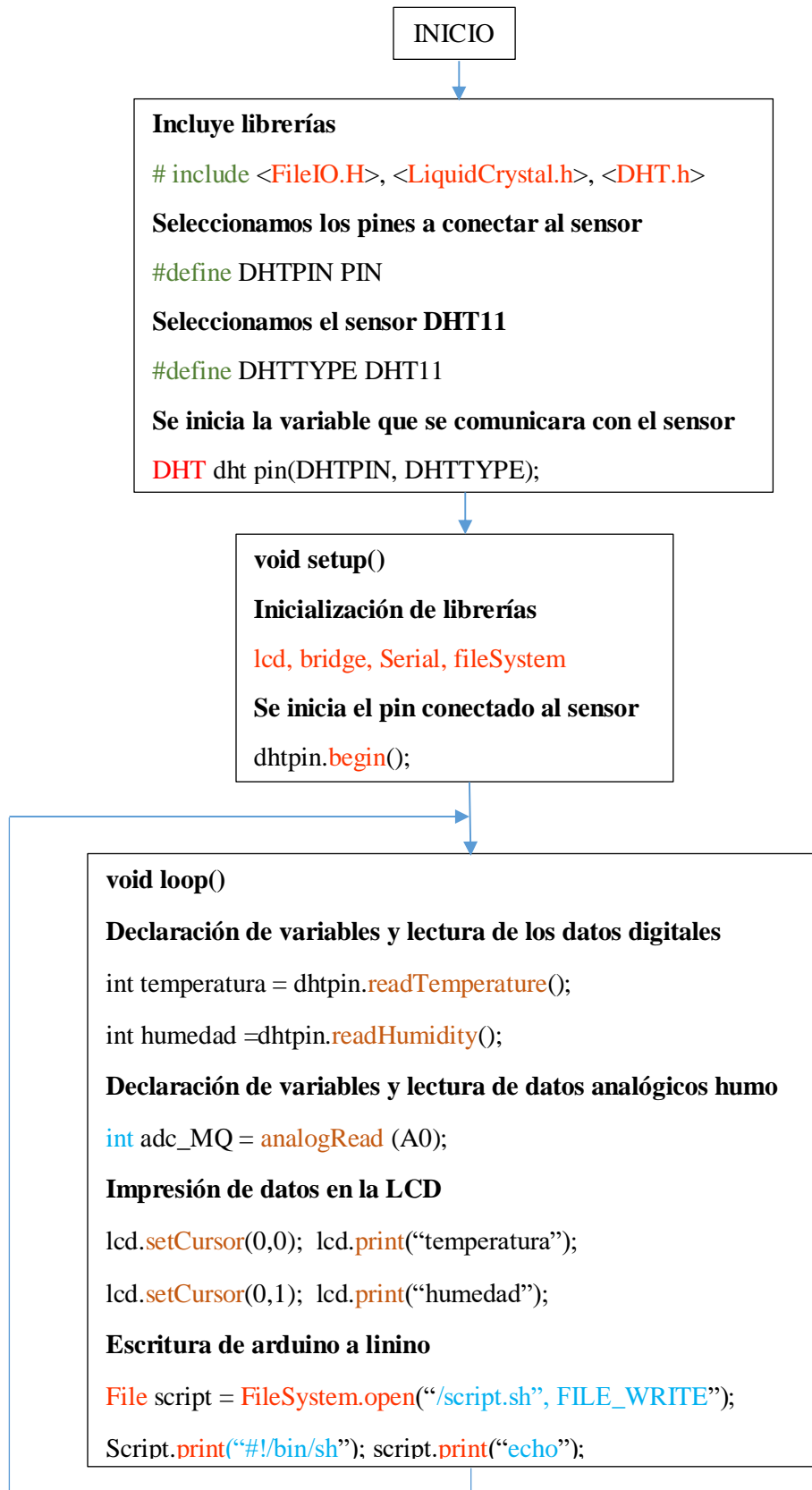


Figura 20. Diagrama de flujo del código

El desarrollo del código se lo realiza en el IDE de Arduino, teniendo como parámetros a medir la temperatura, humedad y humo. Ya descritos su funcionamiento y explicada su conexión en los numerales anteriores, incluyendo la creación de los archivos OID (objetos identificadores), en los que se almacenara los valores obtenidos de las lecturas de los parámetros a ser monitoreados. Todo el código está adjuntado en el anexo A.

3.9 Armado del circuito

El armado del prototipo, se lo realiza utilizando todos los componentes que conformaran el sistema de monitoreo, en la figura 21 se puede observar la conexión general de hardware, realizado en el programa Proteus basándose en las conexiones descritas.

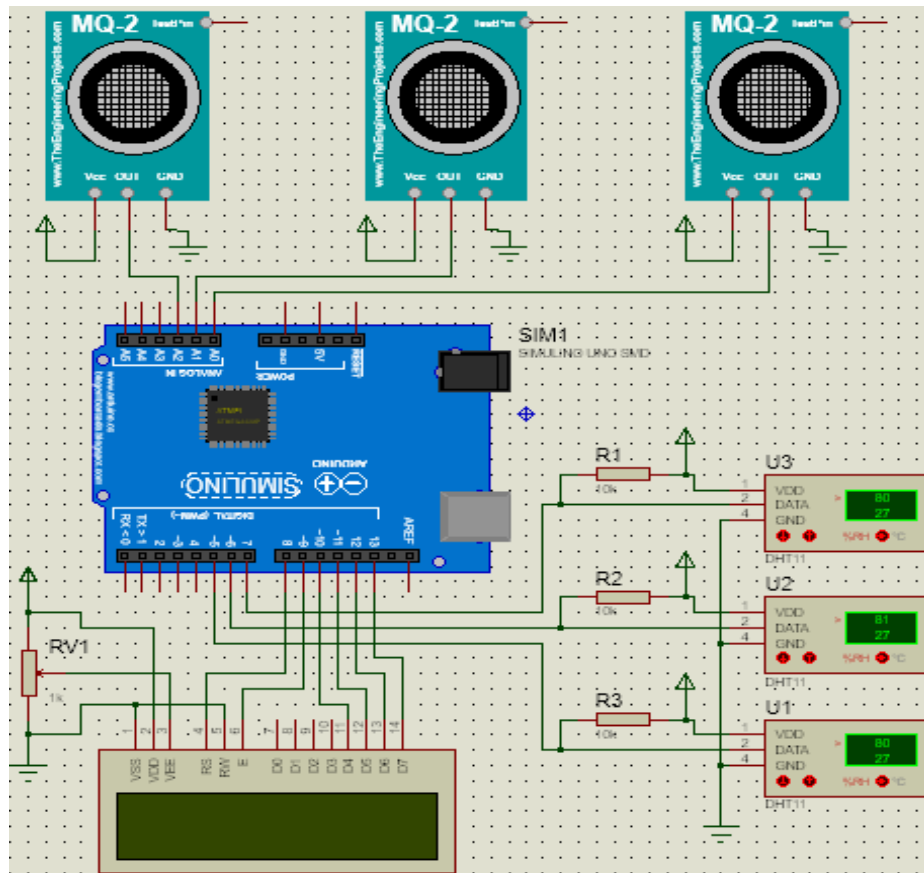


Figura 21. Armado del circuito
Fuente: Elaborado por el autor

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.

Tabla 10. Determinación del valor del fusible

Componente	Corriente consumo unitario	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

Fuente: Elaborado por el autor

En base al cálculo realizado en la tabla 10, se ubicará el fusible más cercano al valor obtenido, siendo el de 0.75 A.

3.10 Diseño de la fuente de energía

En base al cálculo de la corriente necesaria para el funcionamiento del circuito, es indispensable el diseño de una fuente de energía, que brinde más corriente que la que provee el puerto USB para el montaje del circuito final. El esquema de la fuente de energía se lo muestra en la figura 22.

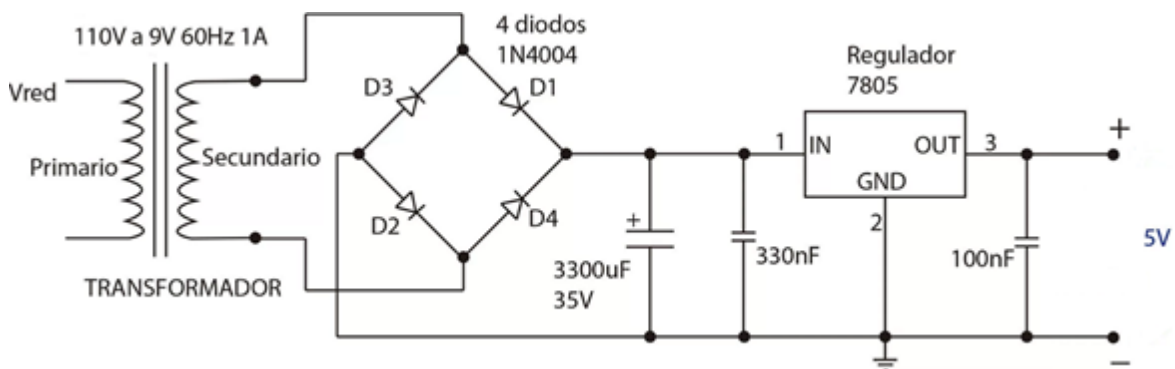


Figura 22: Diseño de la fuente
Fuente: datasheet 7805

A continuación, se detalla los cálculos y elementos para la elaboración de la fuente de energía.

1 transformador con primario de 110V y secundario de 9 voltios a 1 Amperio.

4 Diodos 1N4004 para la etapa de rectificación.

1 capacitor de 330 nF

1 capacitor de 100 nF

1 regulador de voltaje LM7805

Cálculo del capacitor para disminuir el voltaje de rizado

$$V = 9V \quad V_p = 9\sqrt{2} \quad V_{max} = V_p - 1,4 = 9\sqrt{2} - 1,4 = 11,32V$$

“por lo general se diseña el condensador para conseguir una tensión de rizado correspondiente al 10% de la tensión pico máxima” (Alcalde, 2014)

$$V_r = 10\%(V_{max}) = 10\% (11,32) = 1,13V$$

$$V_r = \frac{I}{2fC} \quad 1,13 = \frac{0,8}{2(60HZ)C} \quad C = \frac{0,8}{135,6} = 5800\mu F$$

En base a los cálculos realizados se utilizará un capacitor de 6000uF y se recomienda el doble del voltaje máximo, siendo 25 voltios, con esto se obtiene un voltaje DC a la entrada del regulador, considerando que el voltaje de entrada promedio es de 10 voltios, la potencia disipada será:

$$P = (V_{entrada} - V_{salida}) * I = (10 - 5) * 0,8 = 4Watts$$

Para disminuir el calentamiento del regulador se le ubicara un disipador adecuado.

3.11 Montaje del circuito físico

En la figura 23 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.

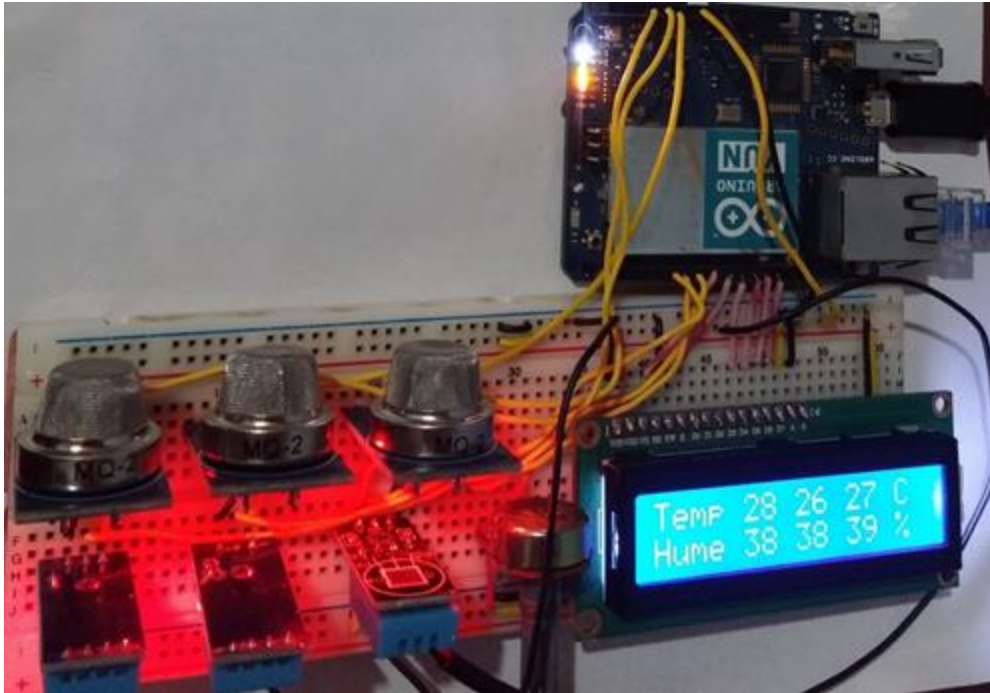


Figura 23. Armado del prototipo del sistema.
Fuente: Elaborado por el autor

3.12 Instalación 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. Todo el procedimiento se lo muestra en el apéndice B.

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 figura 24.

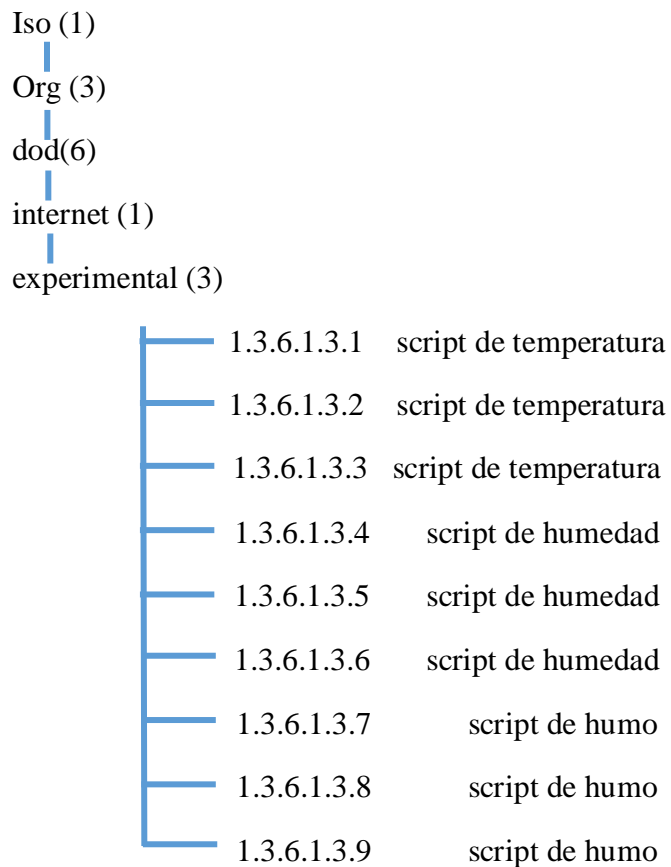


Figura 24. Árbol resultante sistema de monitoreo
Fuente: Elaborado por el autor

3.13 Instalación y Configuración de archivos en Nagios 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, todo el procedimiento se lo realiza en el anexo C.

Culminada la instalación, procedemos a crear el archivo de configuración, que va a contener la información del sistema embebido Arduino, mediante comandos creamos el archivo, a partir de un archivo de ejemplo contenido en la siguiente dirección.

```
cd /usr/local/nagios/etc/objects
cp templates.cfg arduino.cfg
```

La información que va a contener el archivo es información de host y de grupo que pertenece

```
define host {
    use          linux-server; Inherit default values from a template
    host_name    arduino; The name we're giving to this host
    alias        arduino; A longer name associated with the host
    address      x.x.x.x; IP address of the host}

define hostgroup {
    hostgroup_name arduino-servers; The name of the hostgroup
    alias          Linux Servers; Long name of the group
}
}
```

A continuación, se detalla la configuración de los servicios que van a ser monitoreados, entre los que tenemos temperatura, humedad y humo.

Para definir los valores warning y critical en los sensores de humo, se ha realizado varias pruebas con el sensor MQ-2 en condiciones normales sin humo y en condiciones sometiendo a humo en la que se ha recopilado los siguientes datos.

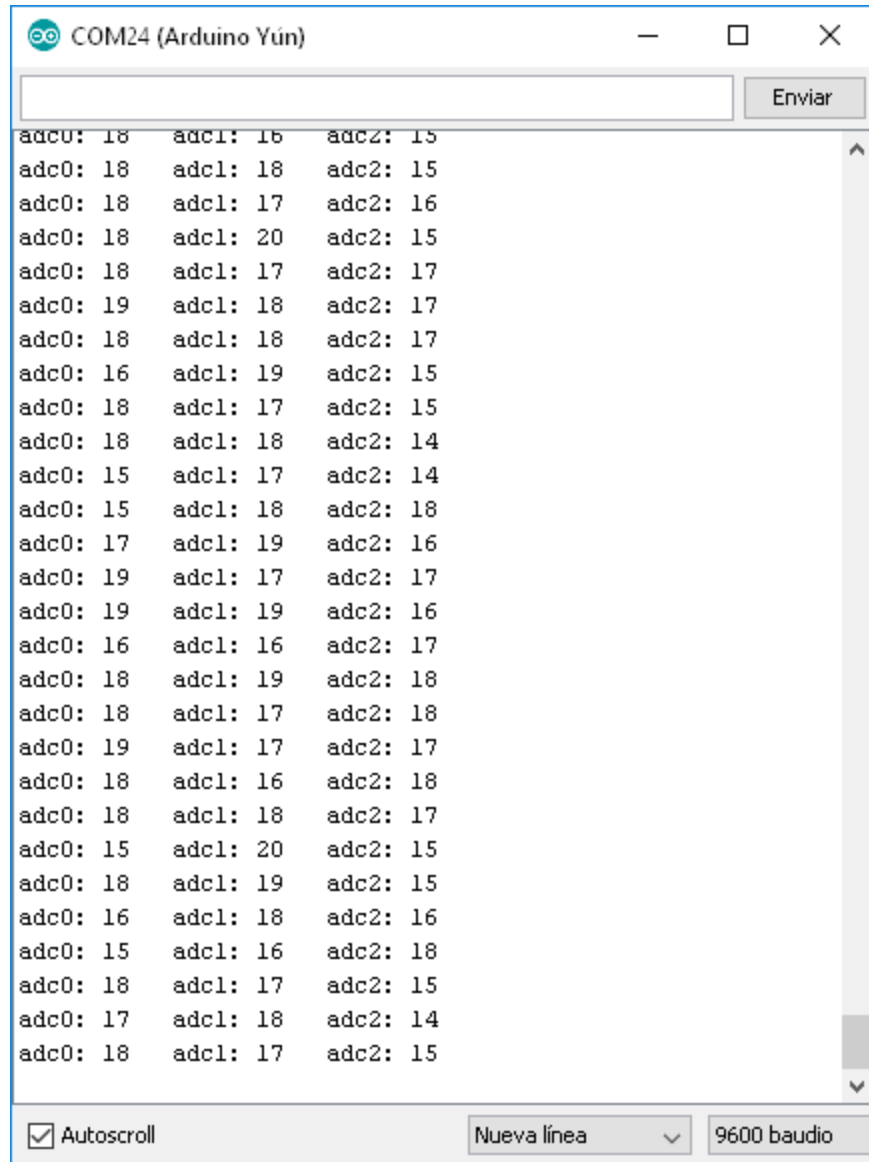


Figura 25. Datos de los sensores sin humo

Fuente: Elaborado por el autor

En la figura 25, podemos observar que los valores son muy similares y tienden a tener un valor casi igual en todos los casos. Para la siguiente prueba se ha sometido el sensor a pequeñas cantidades de humo como lo muestra la figura 26.

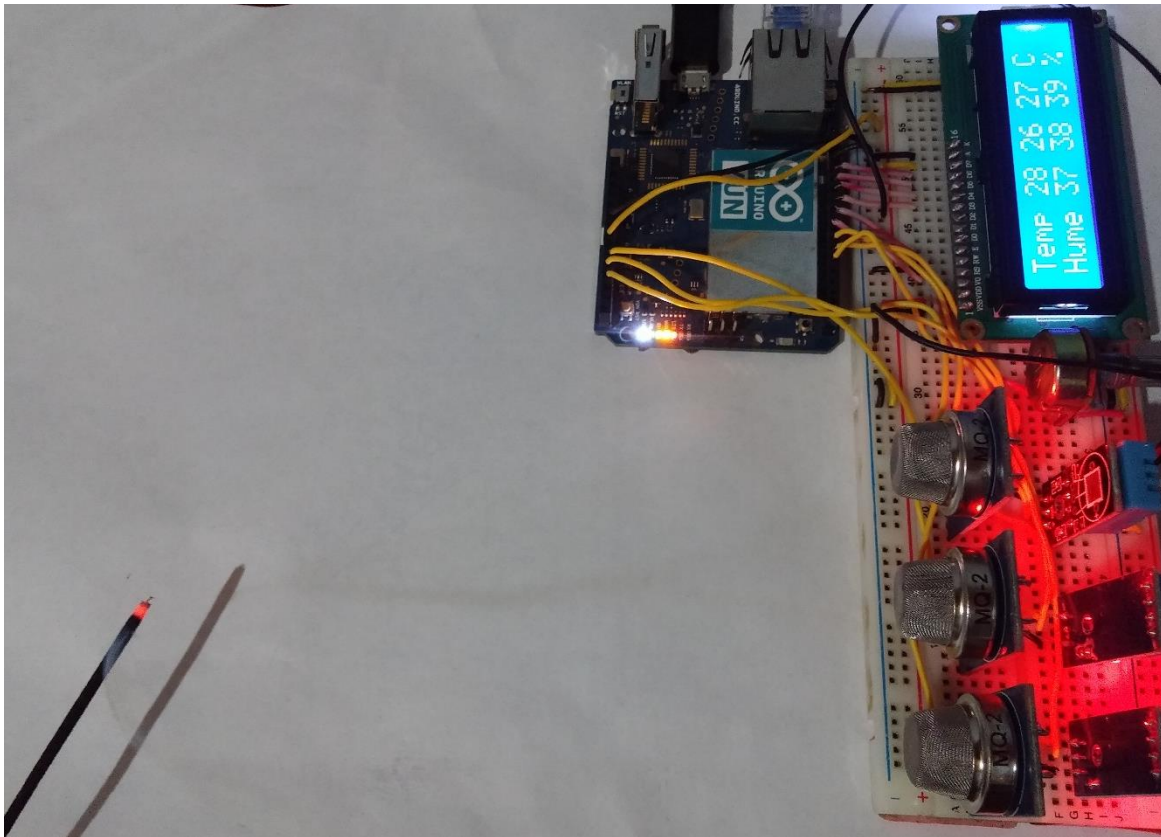


Figura 26. Exposición de humo al sensor MQ-2
Fuente: Elaborado por el autor

La cantidad de humo aplicada al sensor es mínima, por tal motivo los cambios de valores en los datos obtenidos del sensor no serán tan significativas y serán por un corto periodo de tiempo. Como se visualiza en la figura 27.

Los datos obtenidos en condiciones de humo, en la que se puede mirar la reacción de los datos analógicos en presencia de humo, no es en largos periodos de tiempo, llegando a una lectura analógica de 82 el pico más alto para el sensor adc2. Estos valores están representados en partes por millón (ppm).

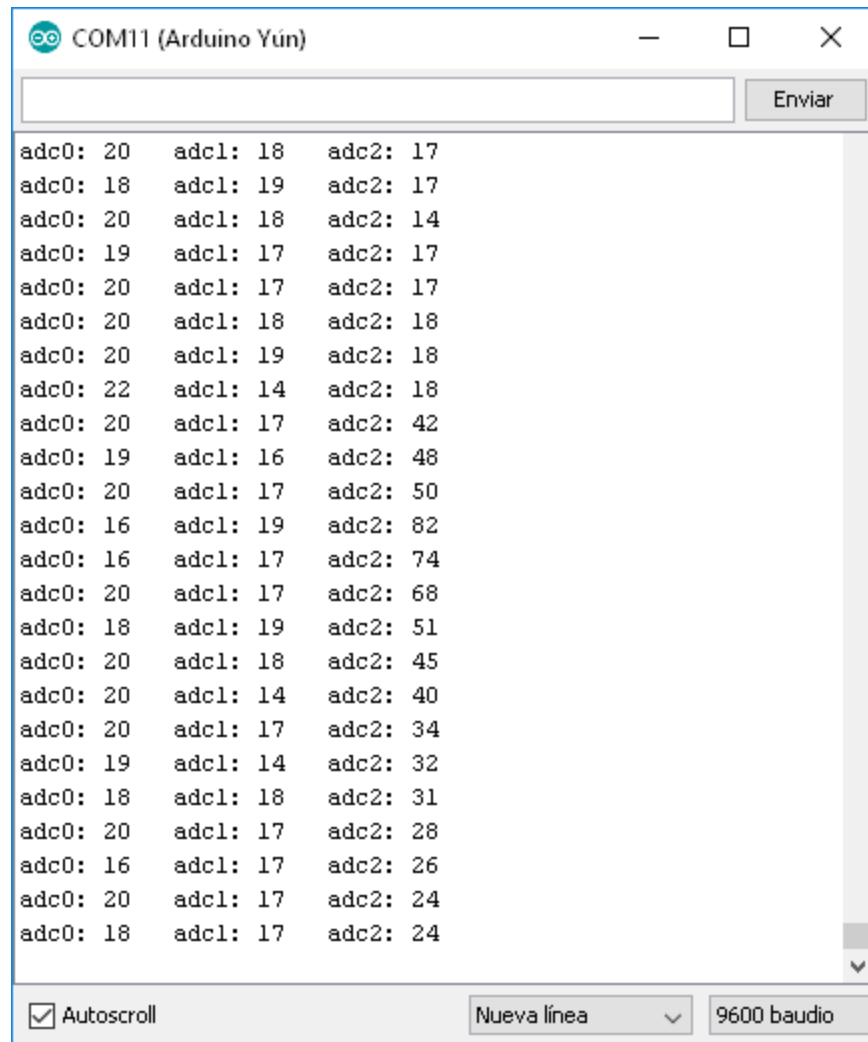


Figura 27. Datos de los sensores con humo.
Fuente: Elaborado por el autor

para la activación de la alarma de humo, en base a las pruebas realizadas se necesitaría, de por lo menos 500 ppm de humo presente en el aire cercano a una distancia máxima de 1 metro del sensor, la bocanada de humo que se absorbe de un tabaco equivale aproximadamente a 200 ppm, según las pruebas realizadas con el sensor, mediante lo cual se necesitaría de 3 bocanadas de humo arrojadas en dirección al sensor para que lo pueda detectar y emitir la alarma.

Con los análisis anteriores se procede a configurar los valores de warning y critical para temperatura, humedad y humo en el archivo de nagios.

Para definir los rangos de temperatura y humedad se guía en la norma TIA-942, en la que recomienda una temperatura de entre 20 °C a 25 °C, con la cual se define el rango en estado warning entre 20 °C a 25 °C y en base a los valores obtenidos de los datasheet en la tabla 5. Se define el estado critical entre 18 °C a 27 °C. Para definir el estado warning en humedad se adopta los valores de la norma en 40% a 55% de humedad relativa y el estado critical en 35% a 60% de humedad relativa. Para definir los rangos de valores de humo, se basa en las pruebas realizadas con el sensor, esto debido a que no existen alguna norma que indique porcentajes de humo admitidos en centros de datos, quedando los valores de warning en 0:400 ppm y critical de 0:500 ppm.

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.

Tabla 11: Rangos de activación de alarmas

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.	

Fuente: Elaborado por el autor

En el siguiente código se muestra la configuración del archivo de nagios en donde podemos apreciar el OID correspondiente a cada parámetro a ser monitoreado, a continuación, se describe el código.

```
define service { // se define un servicio a ser monitoreado
```

```
use // se define el tipo de servicio en este caso un service local y un servicio de gráficos
```

```
host_name // el nombre del equipo a ser monitoreado en el ejemplo es el arduino
```

```
check_command // es el comando que se usara para monitorear
```

```
check_snmp// es el plugin que se utilizara para la comunicación servidor-agente
```

```
-o 1.3.6.1.3.x.101.1 OID a ser monitoreado específicamente archivo script
```

```
-C public // se utiliza la contraseña o comunidad de tipo publica
```

```
-P 2c // representa la versión de snmp
```

```
-w valor1:valor2 // es el rango de estado alerta, si el valor se encuentra entre valor 1 y valor 2 no genera alerta warning, si el valor no se encuentra entre valor1 y valor2 , genera una alerta de tipo warning.
```

```
-c valor3:valor4 // es el rango de estado peligro, si el valor se encuentra entre valor 3 y valor 4 no genera alerta de peligro, si el valor no se encuentra entre valor3 y valor4 , genera una alerta de tipo peligro.
```

```
-u " °C" // etiqueta que se mostrara a lado del valor de temperatura
```

Consideraciones: valor 3 < valor1 y valor4 > valor2

```
#OID correspondiente a temp7
```

```
define service {
    use          local-service,graphed-service
    host_name    arduino
    service_description Temperatura Rack CENTRO
    check_command check_snmp! -o 1.3.6.1.3.1.101.1 -C public -P 2c -w 20:27 -c 18:29 -u " °C"
}

```

#OID correspondiente a temp6

```
define service {
    use          local-service,graphed-service
    host_name    arduino
    service_description Temperatura Rack IZQUIERDO
    check_command check_snmp! -o 1.3.6.1.3.2.101.1 -C public -P 2c -w 20:27 -c 18:29 -u " °C"
}

```

#OID correspondiente a temp5

```
define service {
    use          local-service,graphed-service
    host_name    arduino
    service_description Temperatura Rack DERECHO
    check_command check_snmp! -o 1.3.6.1.3.3.101.1 -C public -P 2c -w 20:27 -c 18:29 -u " °C"
}

```

#OID correspondiente a hume7

```
define service {
    use          local-service,graphed-service
    host_name    arduino
    service_description Humedad Rack CENTRO
    check_command check_snmp! -o 1.3.6.1.3.4.101.1 -C public -P 2c -w 32:55 -c 30:60 -u " %
HR"
}

```

#OID correspondiente a hume6

```
define service {

```

```

use          local-service,graphed-service
host_name    arduino
service_description  Humedad Rack IZQUIERDO
check_command check_snmp! -o 1.3.6.1.3.5.101.1 -C public -P 2c -w 32:55 -c 30:60 -u " %
HR"
}
#OID correspondiente a hume5
define service {
use          local-service,graphed-service
host_name    arduino
service_description  Humedad Rack DERECHO
check_command check_snmp! -o 1.3.6.1.3.6.101.1 -C public -P 2c -w 32:55 -c 30:60 -u " %
HR"
}
#OID correspondiente a smok0
define service {
use          local-service,graphed-service
host_name    arduino
service_description  Humo Rack CENTRO
check_command  check_snmp! -o 1.3.6.1.3.7.101.1 -C public -P 2c -w 0:400 -c 0:500 -u " %
humo "
}
#OID correspondinte a smok1
define service {
use          local-service,graphed-service
host_name    arduino
service_description  Humo Rack DERECHO
check_command  check_snmp! -o 1.3.6.1.3.8.101.1 -C public -P 2c -w 0:400 -c 0:500 -u "
% humo"
}
#OID correspondiente a smok2

```

```
define service {
    use          local-service,graphed-service
    host_name    arduino
    service_description Humo Rack IZQUIERDO
    check_command check_snmp! -o 1.3.6.1.3.9.101.1 -C public -P 2c -w 0:400 -c 0:500 -u "
% humo"
}
```

Para que el servidor pueda aceptar la configuración realizada, agregamos la dirección del archivo en el que se definieron los servicios, para ello nos dirigimos a:

```
cd /usr/local/nagios/etc
```

```
vim nagios.cfg
```

Y agregamos la siguiente línea

```
# Definición para monitorear Arduino
```

```
cfg_file=/usr/local/nagios/etc/objects/arduino.cfg
```

Reiniciamos el servicio 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.

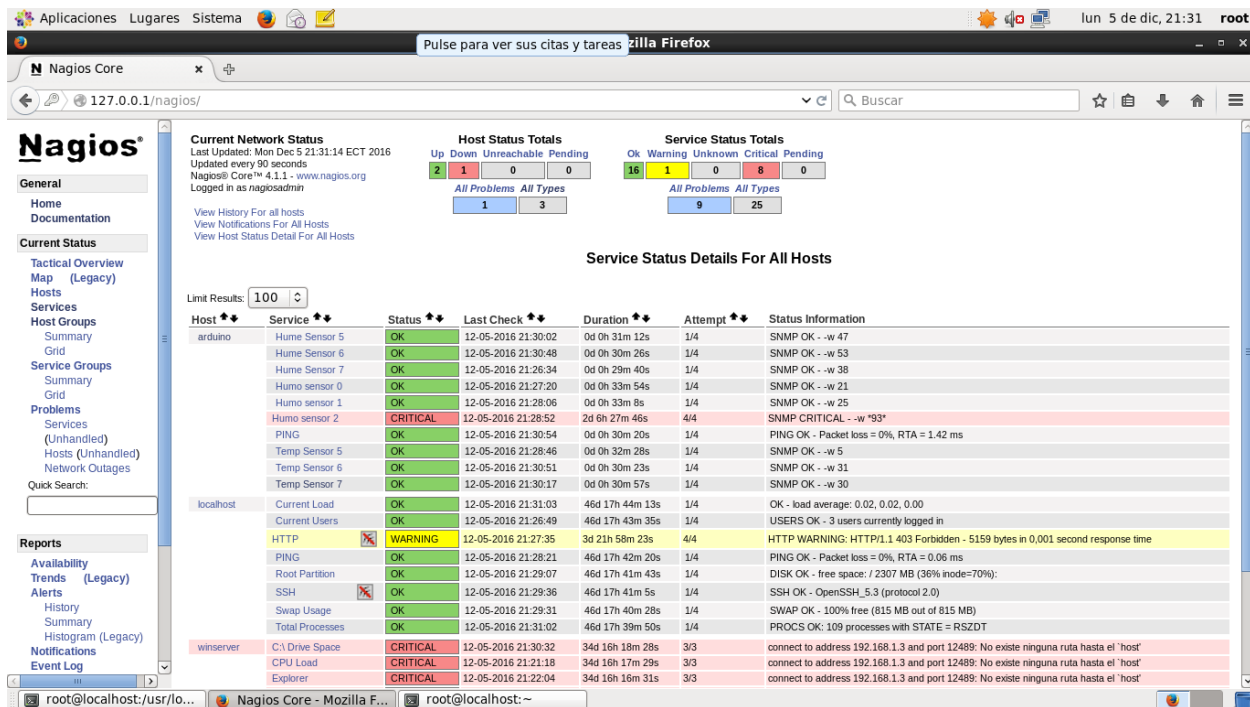


Figura 28. Interfaz web del sistema de gestión.
Fuente: Elaborado por el autor

3.14 Instalación y configuración de 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. La instalación se lo realiza en el apéndice D.

En la figura 29, a diferencia de la figura 28 podemos observar que se activó el servicio de visualización de gráficos. Para la visualización de estadísticas gráficas, es necesario dar clic en el grafico del servicio correspondiente y se desplegara en una nueva pestaña, en la que podemos seleccionar varias modalidades de estadística.

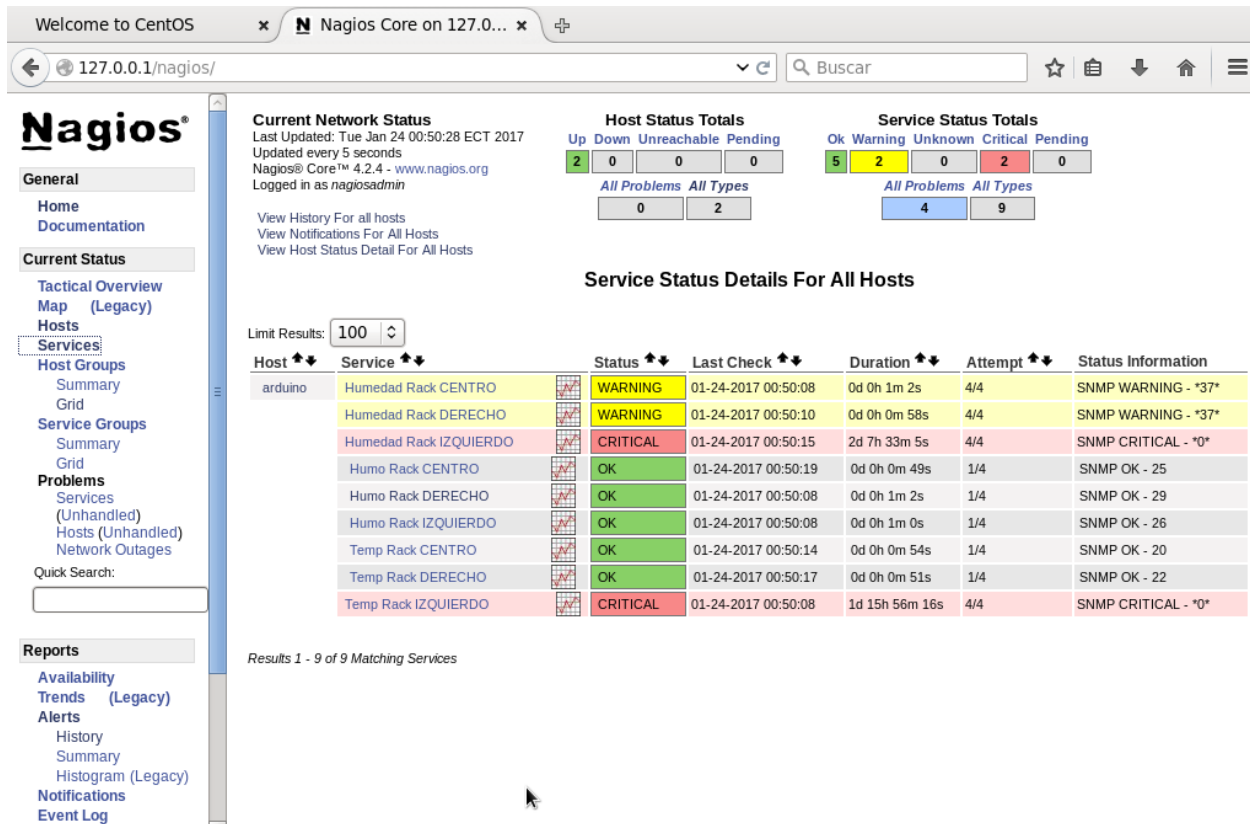


Figura 29. Activación de gráficos para los servicios monitoreados.
Fuente: Elaborado por el autor

3.15 Instalación 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
```

y en el agregamos las siguientes líneas en donde la palabra “contraseña” hace referencia a la contraseña de la cuenta.

```
[smtp.gmail.com]:587 ltcntonny@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” ltcntonny@gmail.com
```

si el correo se envió y recibió 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, ltcntonny@gmail.com
```

procedemos a reiniciar el servicio

```
service nagios restart
```

En la figura 30, 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

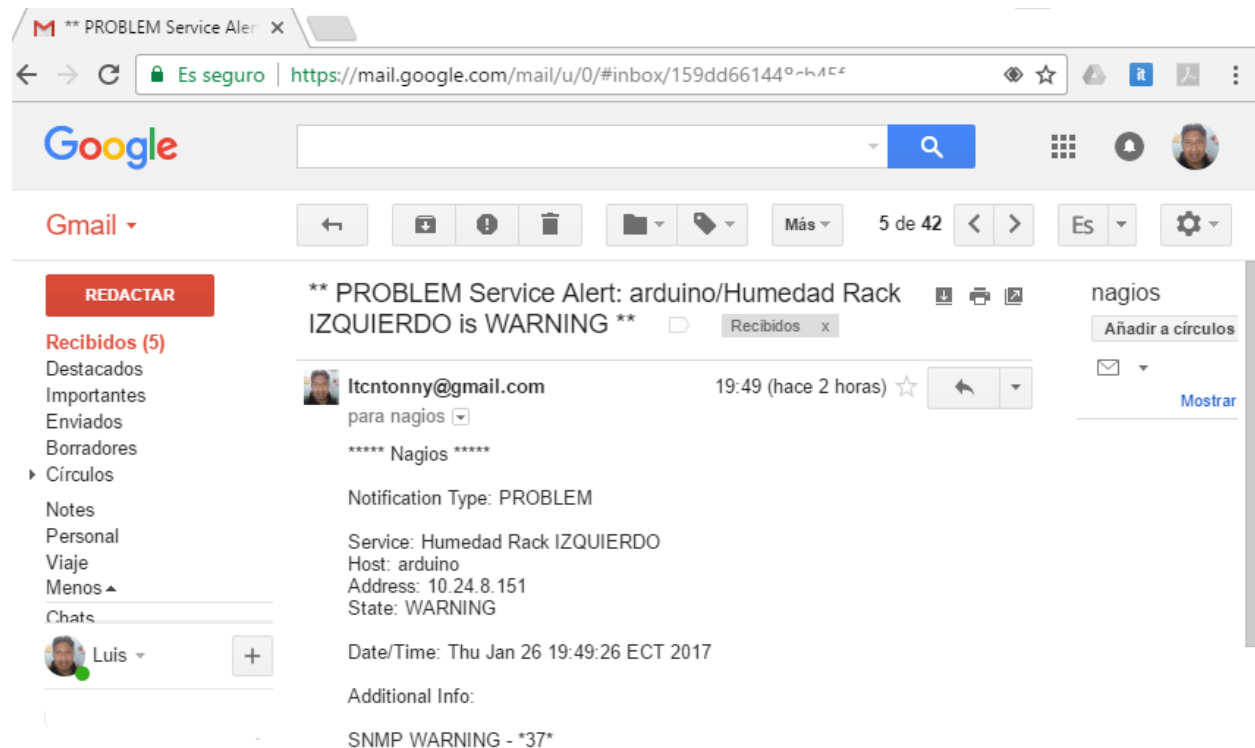


Figura 30. Alertas de nagios por correo
Fuente: Elaborado por el autor

CAPITULO IV

4 IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO

En el actual capítulo se realiza la elaboración de la placa, armado e instalación de todo el sistema de monitoreo, hasta su puesta en funcionamiento para empezar a registrar los valores que obtengan los sensores ubicados en cada rack, de esta manera hacer el análisis de resultados, conclusiones y recomendaciones, así como planteamiento de políticas para la seguridad física y ambiental.

4.1 Arquitectura del sistema

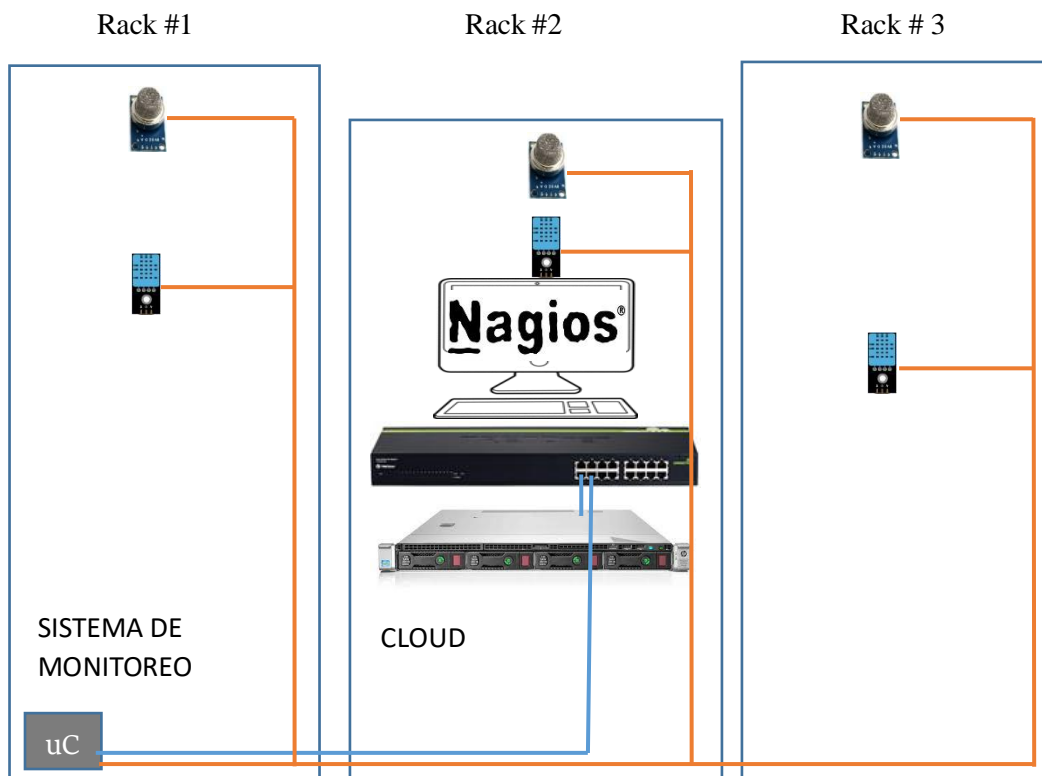


Figura 31. Arquitectura del sistema
Fuente: Elaborado por el autor

En la figura 31, 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.

4.2 Proceso de armado e implementación del sistema

El diseño de la placa se lo realiza en base a los componentes descritos en la figura 21.

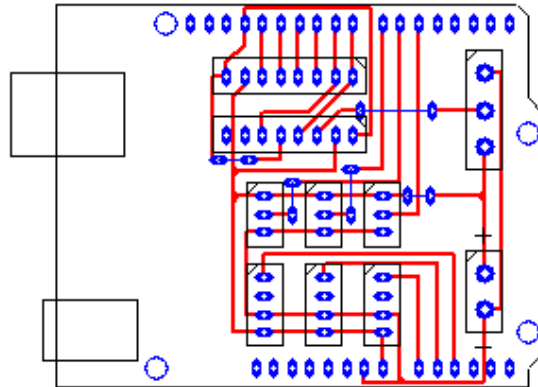


Figura 32. Diseño de la placa
Fuente: Elaborado por el autor

En la figura 33, 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.

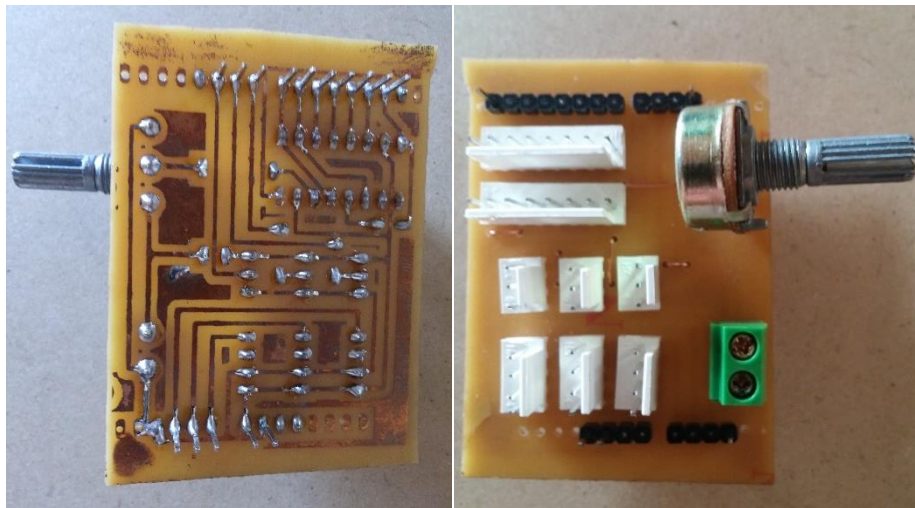


Figura 33. Armado de la placa que encaja en el Arduino Yun.
Fuente: Elaborado por el autor

A continuación, 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.

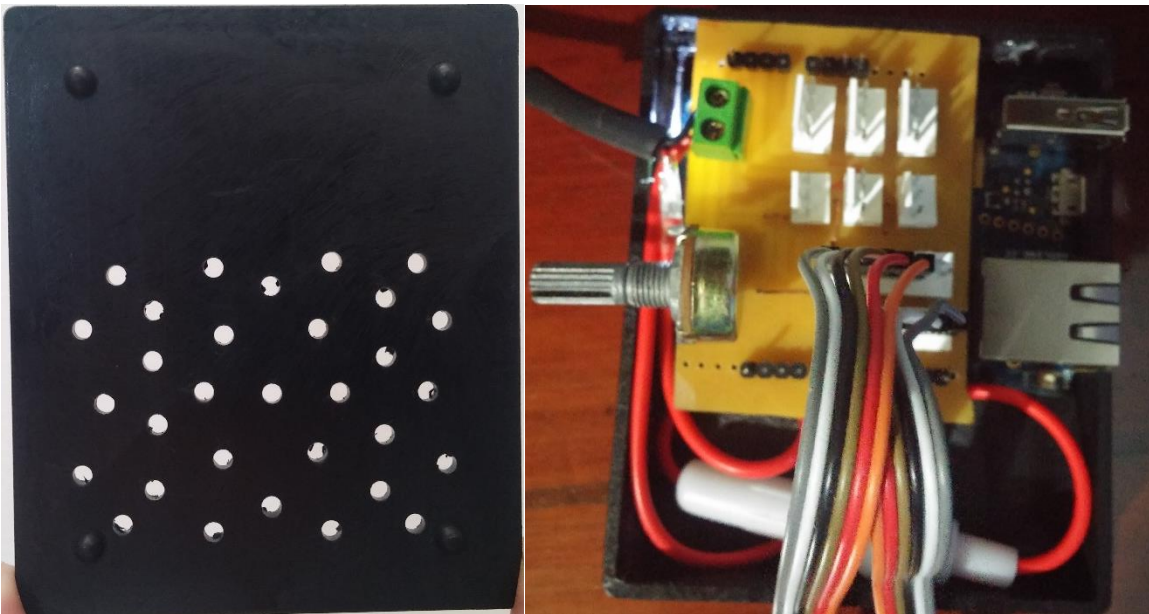


Figura 34. Ubicación del arduino en la caja.
Fuente: Elaborado por el autor

Para la instalación del sistema embebido, se conecta el cortapicos adquirido para el sistema de monitoreo en la multitoma de energía del rack 1, En la figura 35 se conectan los cables a la placa, los mismos que terminan en el otro extremo con los sensores. Todos los cables cuentan con el etiquetado correspondiente en cada extremo, con etiqueta de color amarillo correspondiente para propósitos misceláneos en cableado estructurado.

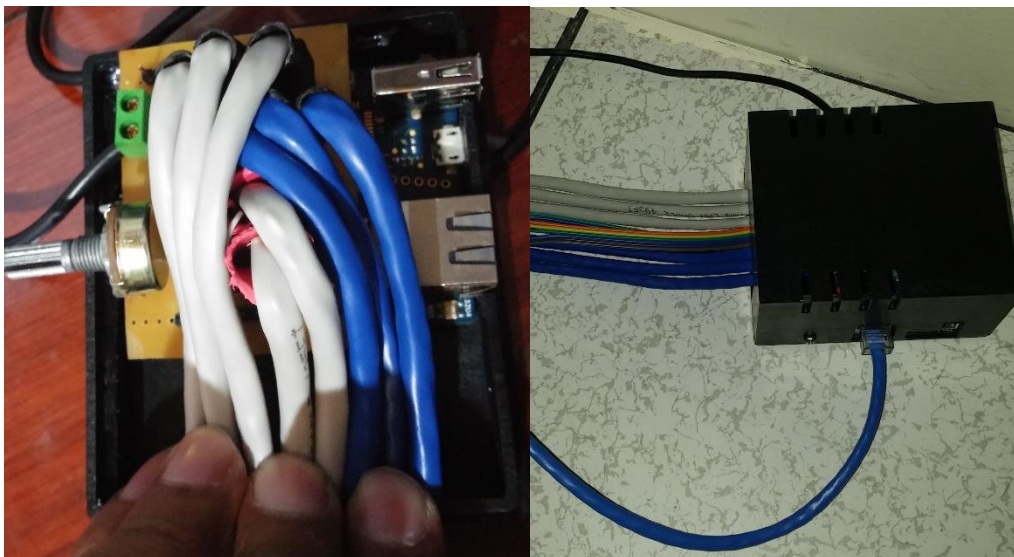


Figura 35. Ubicación del sistema en el Centro de Datos.
Fuente: Elaborado por el autor

4.3 Ubicación de los sensores en los Rack

Los sensores son ubicados en la puerta de atrás de cada rack, en la figura 36, se puede visualizar la colocación de los sensores de temperatura, humedad y humo.

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.

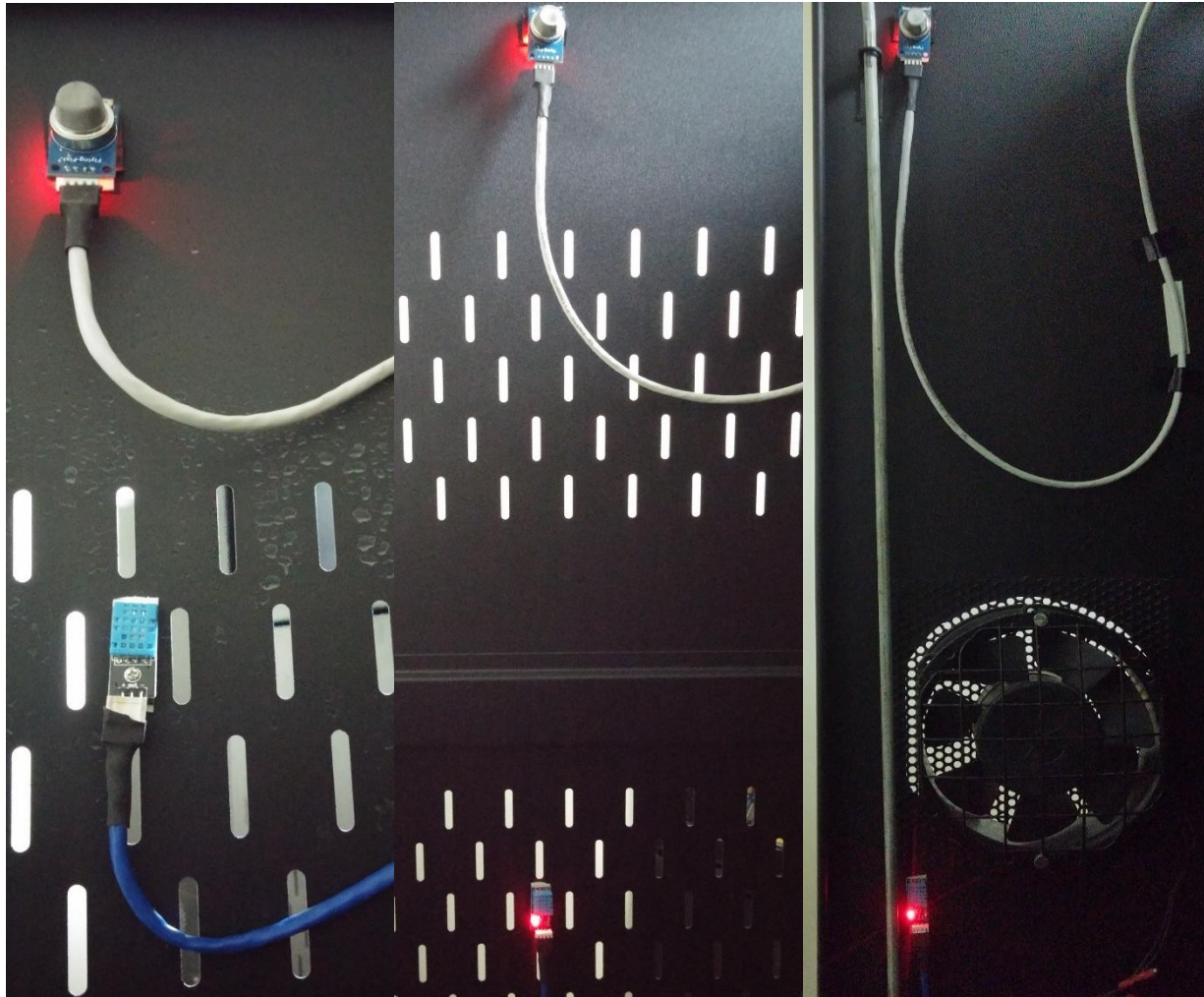


Figura 36. Ubicación de los sensores en los rack izquierdo, central, derecho.
Fuente: Elaborado por el autor

La figura 37, 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 37. Ubicación de la pantalla LCD.
Fuente: Elaborado por el autor

Para culminar el trabajo se realiza la colocación de ganchos, para la ubicación de los extintores que se encontraban en lugares no apropiados dentro del Centro de Datos, debiendo estar colocados a una altura promedio de 1.70 metros, visible y accesible de ser posible siempre cerca de los puntos de evacuación y de los puntos en los que existen más probabilidades de que se inicie fuego.



Figura 38. Colocación de ganchos para los extintores.
Fuente: Elaborado por el autor

4.4 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 muestran los dos procedimientos.

4.4.1 Ingreso al sistema desde internet

El siguiente 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. Abrimos un navegador e 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 39.

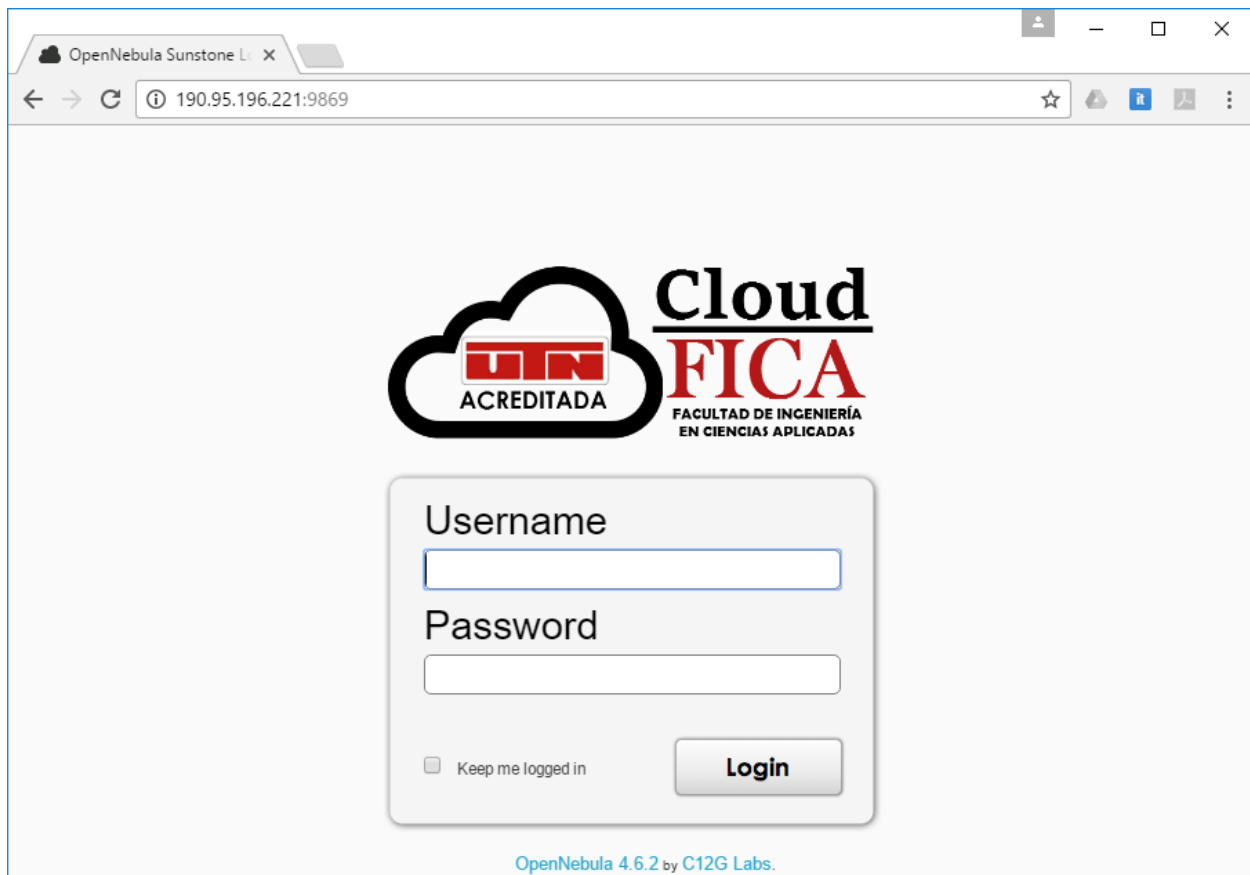


Figura 39. Interfaz de la plataforma Cloud de la Universidad.
Fuente: Elaborado por el autor

Una vez registrado, accederemos al entorno de administración de la cuenta, presionamos en el botón de opciones de la máquina, señalado en la figura 40.

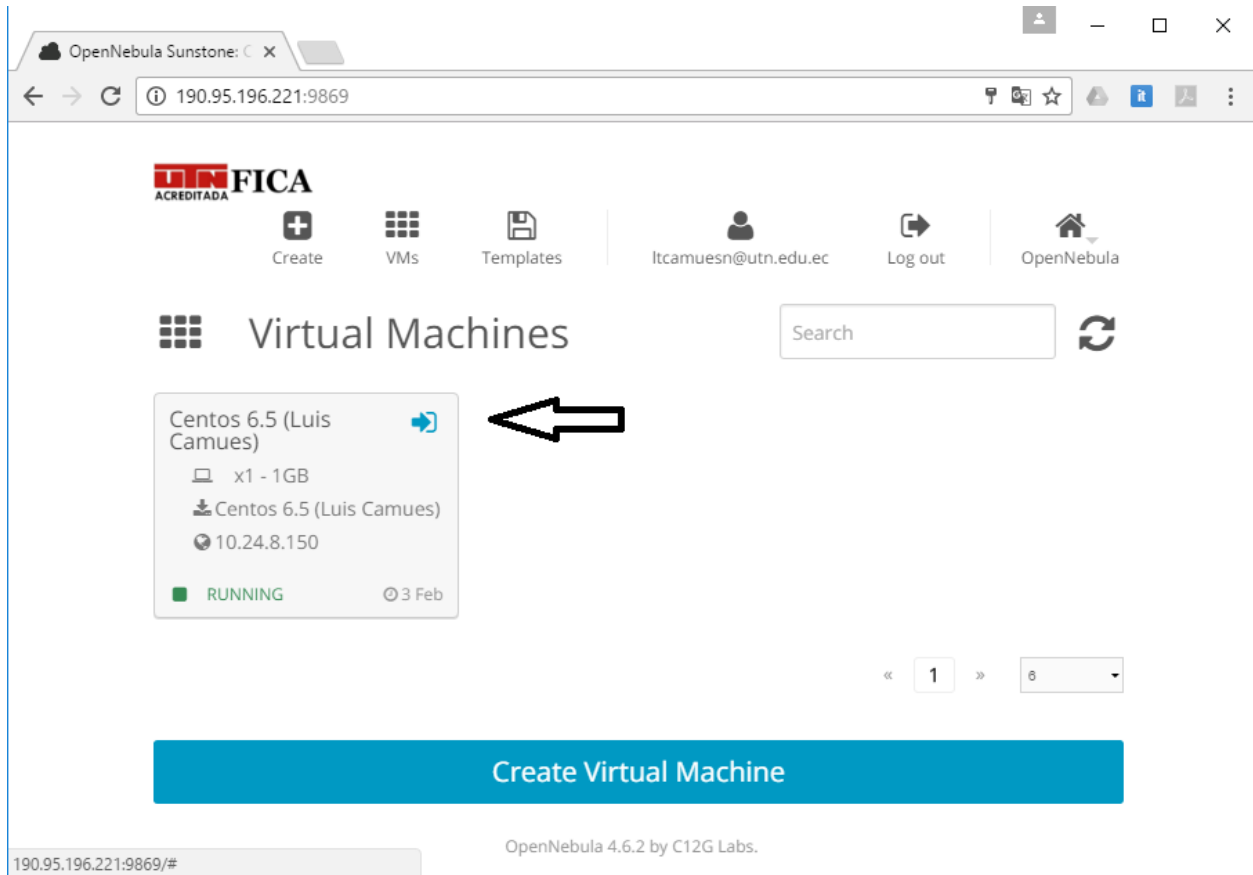


Figura 40. Entorno de administración de la cuenta Cloud.
Fuente: Elaborado por el autor

Se desplegará un menú de opciones, en el que debemos presionar la opción consola, esto abrirá en una ventana emergente el escritorio de la máquina que aloja el sistema de monitoreo, el procedimiento para el ingreso a la interfaz web del sistema pudiera ser ingresando a la dirección 127.0.0.1/nagios en donde nos autenticaremos con las credenciales de usuario y contraseña del sistema, dando como resultado el ítem 4.4.2. ingreso al sistema desde intranet.

4.4.2 Ingreso al sistema desde la intranet

Ubicándonos en una PC conectada a la red de la Universidad, abrimos un navegador web en donde se escribirá la dirección x.x.x.x/nagios, la cual pedirá autenticarse con los datos de usuario y contraseña.



Figura 41. Interfaz principal de Nagios Core.

Fuente: Elaborado por el autor

Una vez dentro se mostrará la pantalla mostrada en la figura 41, en donde nos dirigiremos al menú izquierdo y presionaremos la opción services, la cual mostrará la interfaz de monitoreo ubicada en la figura 42.

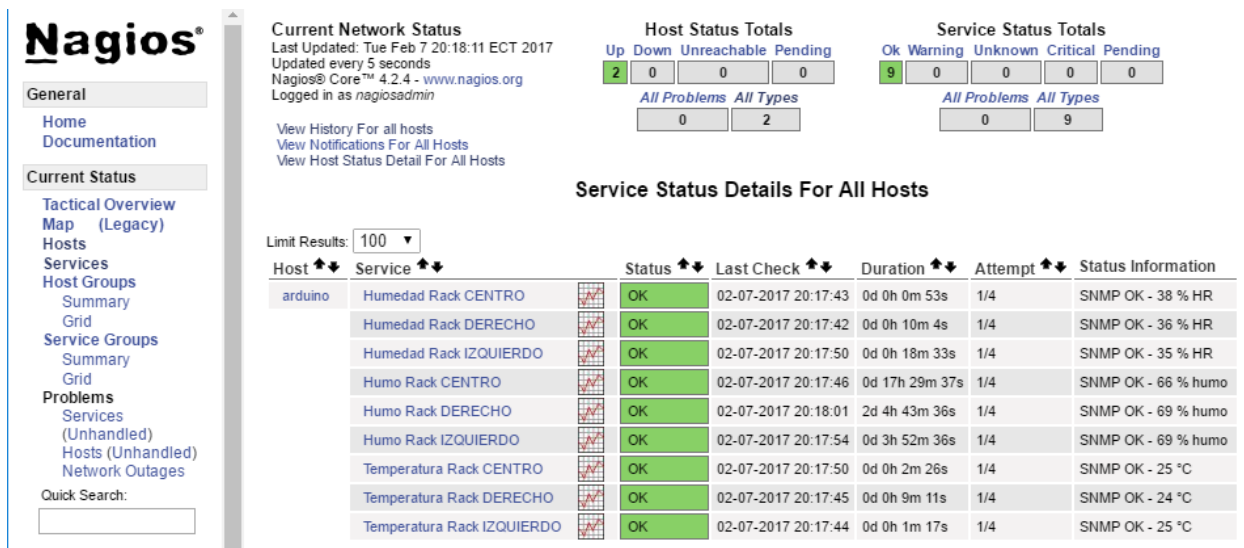


Figura 42. Interfaz de monitoreo de Nagios Core.
 Fuente: Elaborado por el autor

En la figura 42, se pueden visualizar los valores de temperatura humedad y humo correspondientes a cada rack, al momento de la captura todos los indicadores de estado se encuentran con un color verde, esto significa, que los valores mostrados están dentro del rango definido en la sección configuraciones de archivos de nagios. Si algún indicador se torna de color amarillo, significa que el valor correspondiente al indicador pasó de un estado “OK” a un estado “warning”, lo que es una advertencia. Si algún indicador se torna de color rojo, significa que el valor del indicador pasó de un estado “OK” a “warning” y luego a “critical”, lo que requiere una mayor atención o supervisión en el rack que se detectó el incremento o decremento de valor.

4.5 Visualización de datos registrados mediante gráficos

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 gráfico correspondiente a cada sensor como se indica la flecha en la figura 43.

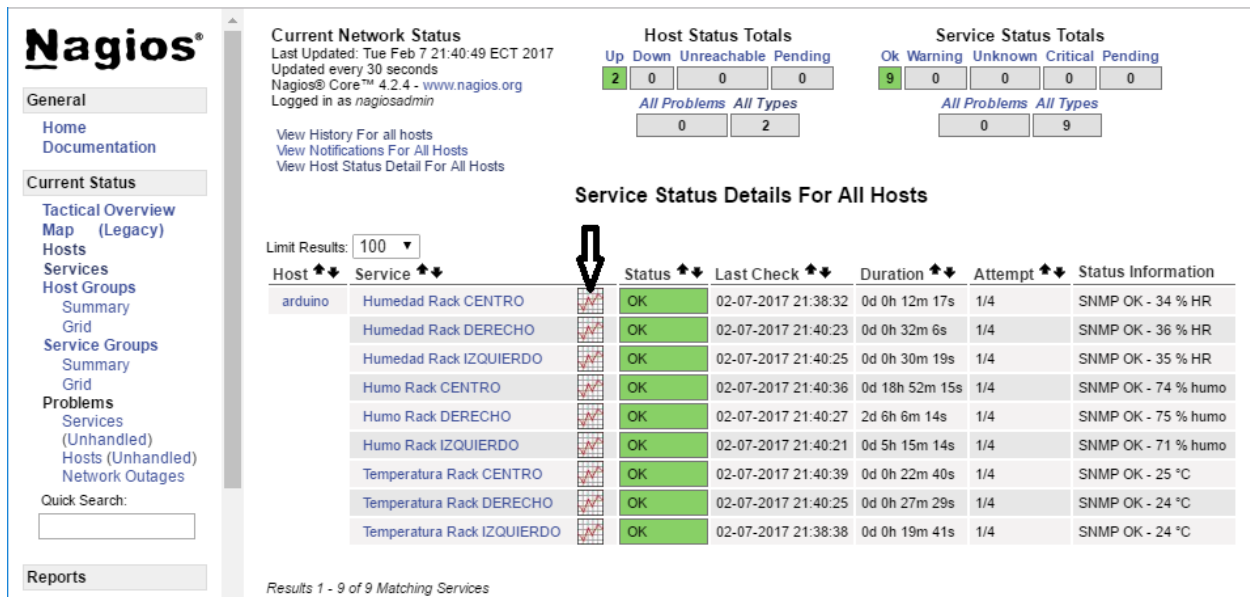


Figura 43. Figura para despliegue de monitoreo gráfico.
 Fuente: Elaborado por el autor

Los siguientes gráficos corresponden al registro de valores en unas determinadas horas del rack # 3 (rack derecho).

Tabla 12. Representación del color de las líneas en los gráficos

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

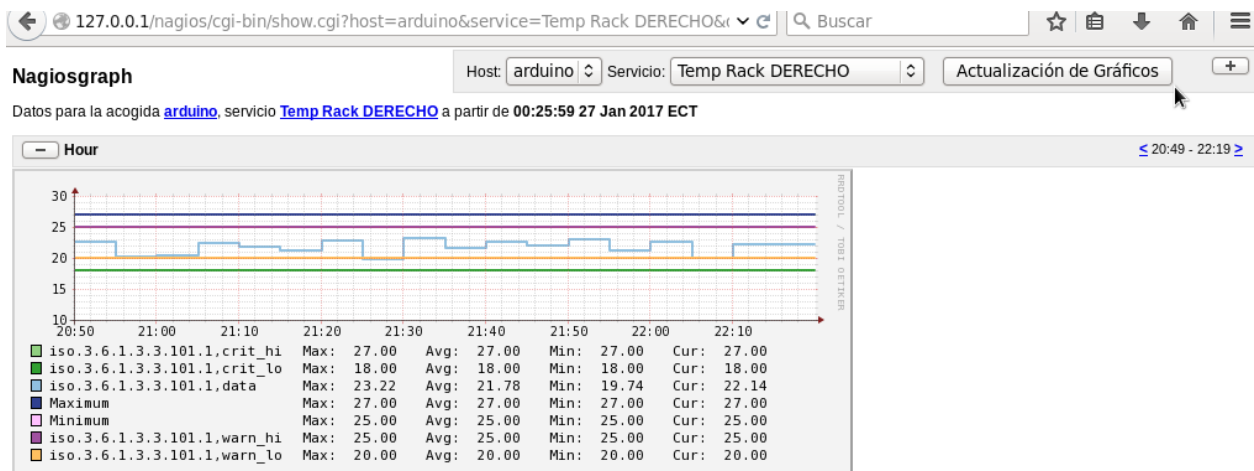


Figura 44. Gráfico de temperatura rack derecho.
Fuente: Elaborado por el autor

En la figura 44, 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.

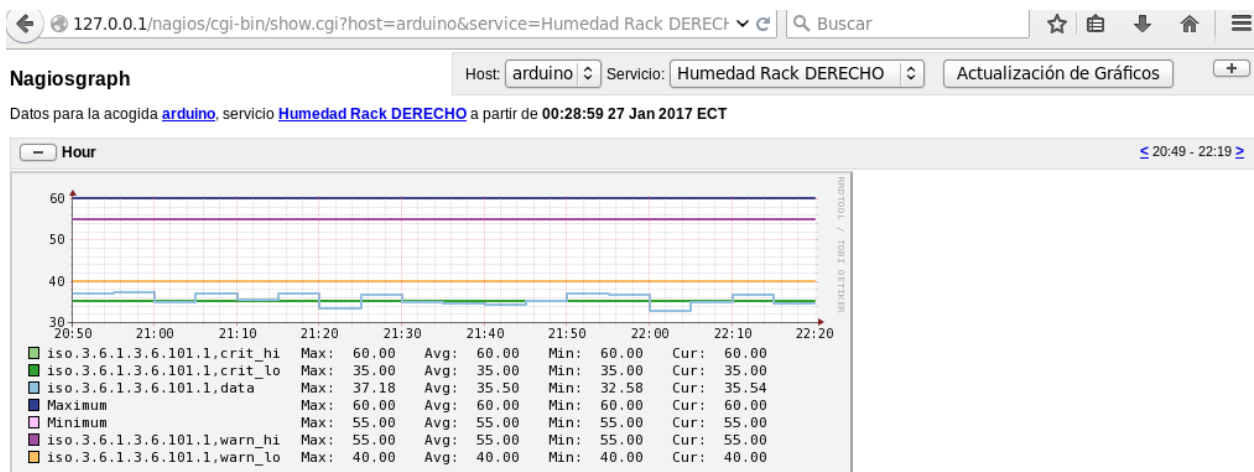


Figura 45. Gráfico de humedad rack derecho.
Fuente: Elaborado por el autor

En la figura 45, 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.

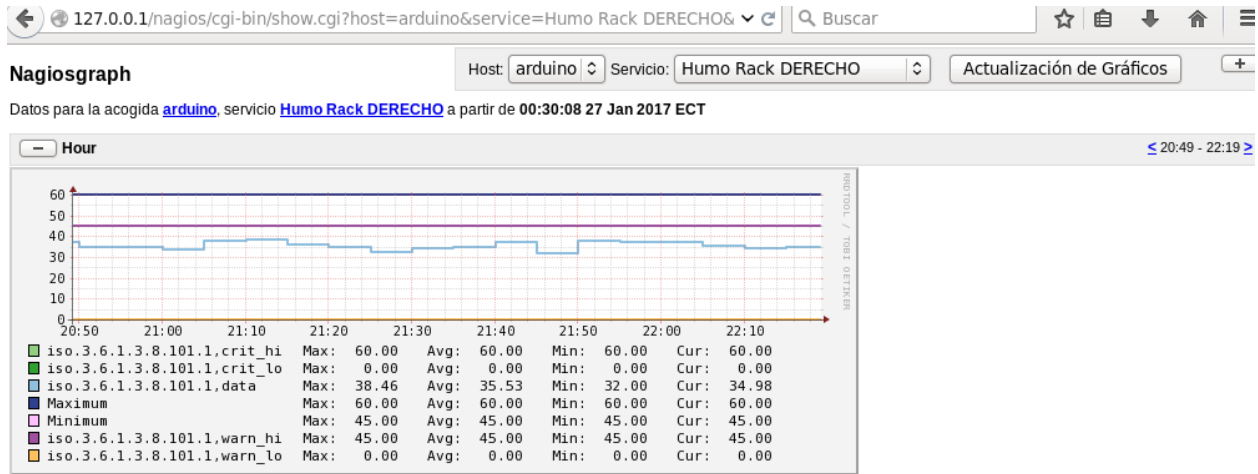


Figura 46. Gráfico de humo rack derecho.
Fuente: Elaborado por el autor

En la figura 46, 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 limites asignados en ese intervalo de tiempo, por tal motivo no ha generado alertas.

4.6 Propuesta de políticas para la seguridad física y ambiental del Centro de Datos.

Las políticas de seguridad, es un documento escrito en donde se establecen los lineamientos a seguir para reducir los riesgos físicos y ambientales en el Centro de Datos, en él se detallan procedimientos para ingreso de personas, manejo adecuado del sistema de monitoreo. Todo el documento se adjunta en el apéndice E.

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, varía 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.

RECOMENDACIONES

- En el diseño de la placa al momento de quemar la baquelita, realizarlo de manera profesional revisando que no existan imperfecciones, ya que al instalar el sistema funcionara por algunos días, pero a mayor tiempo perderá su sincronización y se mostraran caracteres no correspondientes a los reales.
- Para la utilización del cableado, utilizar de preferencia par trenzado debido a su composición de un solo hilo, de otra manera utilizando cable multi-hilo se pierde una pequeña fiabilidad en los datos.
- En la elaboración de los extremos de los cables, se debe procurar que sean de fácil acople y desacople a la placa y sensores, tomar en cuenta usar cable flexible, solo en los extremos del cable correspondientes al LCD-placa en donde necesitan mayor número de conductores debido a sus pines y es muy propenso al quiebre de sus hilos, lo que no permite la visualización en la LCD debido al corto circuito.
- En las configuraciones es recomendable realizar un censado no tan seguido pudiendo ser de cada minuto ya que de serlo muy seguido las variaciones harían que genere muchas alertas y sature el buzón de los contactos establecidos.
- En la configuración de nagiosgraph, es recomendable realizarlo al final de todo el procedimiento y no modificar mucho los rangos de valores ya que esto genera inconvenientes en las gráficas y deja de funcionar.
- En caso de pérdida de energía del servidor físico, el servidor nagios virtual pasara a estado desconocido, por tal motivo contactar con el administrador de la plataforma cloud, para que habilite el servidor nagios.

GLOSARIO DE TERMINOS

Datasheet (Hoja de especificaciones). Es un documento que resume el rendimiento y otras características técnicas de un producto.

Linino OS. Es un tipo de software libre que utilizan las placas Arduino de características avanzadas.

Linux. Tipo de Software libre y código abierto, su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU,) y otra serie de licencias libres.

Open Source License. Es un tipo de licencia para software de computadora y otros productos que permite obtener el código fuente de manera gratuita.

OpenWrt. Es un firmware basado en una distribución de Linux empotrada en dispositivos tales como routers personales.

Sistema embebido. Es un sistema de computación diseñado para realizar una o algunas funciones dedicadas en tiempo real.

Potenciómetro. Resistencia regulable en un circuito eléctrico.

Prototipo. Ejemplar original o primer molde en que se fabrica una figura u otra cosa.

RJ45. Es una interfaz física comúnmente utilizada para conectar redes de computadoras.

Sketch. Se denomina Sketch a una parte de código fuente listo para abrir con el entorno de desarrollo integrado de Arduino.

SRAM (Static Random Access Memory). Memoria estática de acceso aleatorio, basada en semiconductores, capaz de mantener los datos, mientras siga alimentada, sin necesidad de circuito de refresco.

UART (Universal Asynchronous Receiver-Transmitter). Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie.

USB (Universal Serial Bus). Es un bus estándar industrial que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos

ATMEGA: es un chip microcontrolador creado por la empresa de fabricación ATMEL

ATHEROS: es un chip microcontrolador, que permite trabajar con el estándar IEEE 802.11 de redes inalámbricas.

CLOUD COMPUTING: computación en la nube, es un modelo de prestación de servicios de negocio y tecnología

ANEXO A: CÓDIGO DEL SISTEMA DE MONITOREO

Código realizado en el IDE de arduino para el censado de los parámetros de temperatura humedad y humo en cada rack del Centro de Datos, cargado al Arduino YUN.

```
/*Sensor de Temperatura, Humedad y Humo con snmpd*/
#include <FileIO.h> //incluye la libreria de archivos
#include <LiquidCrystal.h> //incluye libreria LCD
LiquidCrystal lcd(8,9,10,11,12,13); // (rs, enable, pines)
#include <DHT.h>
#define DHTPIN7 7 //sensor DHT 11
#define DHTPIN6 6
#define DHTPIN5 5
#define DHTTYPE DHT11
DHT dht7(DHTPIN7, DHTTYPE);
DHT dht6(DHTPIN6, DHTTYPE);
DHT dht5(DHTPIN5, DHTTYPE);
void setup() {
  lcd.begin(16,2); //inicializamos el lcd
  Bridge.begin(); //iniciamos la comunicacion AVR - LININO
  Serial.begin(9600); //iniciamos la comunicacion serial a 9600 baudios
  FileSystem.begin(); //iniciamos la libreria de archivos
  dht7.begin();
  dht6.begin();
  dht5.begin();
}
void loop() {
  int temp7 = dht7.readTemperature();//Lectura de temperatura rack centro
  int hume7 = dht7.readHumidity(); //Lectura de humedad rack centro
```

```
int temp6 = dht6.readTemperature();//Lectura de temperatura rack izquierda
int hume6 = dht6.readHumidity(); //Lectura de humedad rack izquierda
int temp5 = dht5.readTemperature();//Lectura de temperatura rack derecha
int hume5 = dht5.readHumidity(); //Lectura de humedad rack derecha
```

```
lcd.setCursor(0,0); lcd.print("Temp"); //imprime en la posicion
lcd.setCursor(5,0); lcd.print(temp6);
lcd.setCursor(8,0); lcd.print(temp7);
lcd.setCursor(11,0); lcd.print(temp5);
lcd.setCursor(14,0); lcd.print((char)223);
lcd.setCursor(15,0); lcd.print("C");
lcd.setCursor(0,1); lcd.print("Hume");
lcd.setCursor(5,1); lcd.print(hume6);
lcd.setCursor(8,1); lcd.print(hume7);
lcd.setCursor(11,1); lcd.print(hume5);
lcd.setCursor(14,1); lcd.print("%R");
```

```
File script = FileSystem.open("/etc/snmp/temp7.sh",FILE_WRITE);
script.print("#!/bin/sh\n");
script.print("echo ");
script.print(temp7);
script.close();
File script1 = FileSystem.open("/etc/snmp/temp6.sh",FILE_WRITE);
script1.print("#!/bin/sh\n");
script1.print("echo ");
script1.print(temp6);
script1.close();
File script2 = FileSystem.open("/etc/snmp/temp5.sh",FILE_WRITE);
script2.print("#!/bin/sh\n");
script2.print("echo ");
```

```
script2.print(temp5);
script2.close();
```

```
File script3 = FileSystem.open("/etc/snmp/hume7.sh",FILE_WRITE);
script3.print("#!/bin/sh\n");
script3.print("echo ");
script3.print(hume7);
script3.close();
```

```
File script4 = FileSystem.open("/etc/snmp/hume6.sh",FILE_WRITE);
script4.print("#!/bin/sh\n");
script4.print("echo ");
script4.print(hume6);
script4.close();
```

```
File script5 = FileSystem.open("/etc/snmp/hume5.sh",FILE_WRITE);
script5.print("#!/bin/sh\n");
script5.print("echo ");
script5.print(hume5);
script5.close();
```

```
int adc_MQ0 = analogRead(A0); //Leemos la salida analógica del MQ
float voltaje0 = adc_MQ0 * (5.0 / 1023.0);
int x = adc_MQ0-100;
Serial.print("adc0: ");
Serial.print(x);
/*Serial.print(adc_MQ0);*/
int adc_MQ1 = analogRead(A1); //Leemos la salida analógica del MQ
float voltaje1 = adc_MQ1 * (5.0 / 1023.0);
Serial.print("  adc1: ");
Serial.print(adc_MQ1);
int adc_MQ2 = analogRead(A2); //Leemos la salida analógica del MQ
```



```
float voltaje2 = adc_MQ2 * (5.0 / 1023.0);
Serial.print("  adc2: ");
Serial.println(adc_MQ2);

File script6 = FileSystem.open("/etc/snmp/smok0.sh",FILE_WRITE);
script6.print("#!/bin/sh\n");
script6.print("echo ");
script6.print(x);
/*script6.print(adc_MQ0);*/
script6.close();
File script7 = FileSystem.open("/etc/snmp/smok1.sh",FILE_WRITE);
script7.print("#!/bin/sh\n");
script7.print("echo ");
script7.print(adc_MQ1);
script7.close();
File script8 = FileSystem.open("/etc/snmp/smok2.sh",FILE_WRITE);
script8.print("#!/bin/sh\n");
script8.print("echo ");
script8.print(adc_MQ2);
script8.close();

delay(1000);
}
```

ANEXO B: INSTALACIÓN Y CONFIGURACIÓN DE SNMPD EN LININO

El siguiente procedimiento es la instalación del paquete snmpd y configuración de los archivos.

Desde la maquina linux accedemos mediante el protocolo SSH al mini-servidor LININO

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

Nos autenticamos con las credenciales y procedemos a la instalación de los paquetes. Todo el procedimiento se lo muestra en el apéndice B.

```
opkg update
```

```
opkg install snmpd
```

```
opkg install snmp-utils
```

Una vez instalado el agente SNMP ingresamos en la dirección `cd /etc/config`, ubicamos el archivo `snmpd` en donde se configurará los objetos identificadores para las variables a monitorear.

Con el editor `vi` abrimos el archivo `snmpd` y adjuntamos las líneas que declaran a los objetos.

```
config exec
```

```
option name temp7.sh
```

```
option prog /etc/snmp/temp7.sh
```

```
option miboid 1.3.6.1.3.1
```

```
config exec
```

```
option name temp6.sh
```

```
option prog /etc/snmp/temp6.sh
```

```
option miboid 1.3.6.1.3.2
```

```
config exec
```

```
option name temp5.sh
```

```
option prog /etc/snmp/temp5.sh
```

```
option miboid 1.3.6.1.3.3
```

```
config exec
```

```
option name hume7.sh
option prog /etc/snmp/hume7.sh
option miboid 1.3.6.1.3.4
```

config exec

```
option name hume6.sh
option prog /etc/snmp/hume6.sh
option miboid 1.3.6.1.3.5
```

config exec

```
option name hume5.sh
option prog /etc/snmp/hume5.sh
option miboid 1.3.6.1.3.6
```

config exec

```
option name smok0.sh
option prog /etc/snmp/smok0.sh
option miboid 1.3.6.1.3.7
```

config exec

```
option name smok1.sh
option prog /etc/snmp/smok1.sh
option miboid 1.3.6.1.3.8
```

config exec

```
option name smok2.sh
option prog /etc/snmp/smok2.sh
option miboid 1.3.6.1.3.9
```

Guardamos el archivo y nos dirigimos al directorio en donde agregaremos los OID y dirección de los objetos identificadores

```
vim /etc/snmp/snmpd.conf
```

```
exec 1.3.6.1.3.1 temp7.sh /etc/snmp/temp7.sh
```

```
exec 1.3.6.1.3.2 temp6.sh /etc/snmp/temp6.sh
```

```
exec 1.3.6.1.3.3 temp5.sh /etc/snmp/temp5.sh
```

```
exec 1.3.6.1.3.4 hume7.sh /etc/snmp/hume7.sh
exec 1.3.6.1.3.5 hume6.sh /etc/snmp/hume6.sh
exec 1.3.6.1.3.6 hume5.sh /etc/snmp/hume5.sh
exec 1.3.6.1.3.7 smok0.sh /etc/snmp/smok0.sh
exec 1.3.6.1.3.8 smok1.sh /etc/snmp/smok1.sh
exec 1.3.6.1.3.9 smok2.sh /etc/snmp/smok2.sh
```

Iniciamos el servicio snmpd

```
/etc/init.d/snmpd start
```

Para que se pueda realizar la lectura del valor almacenado en el objeto identificador con el comando snmpget desde terminal, vamos a la dirección /etc/snmp en donde se encuentran los archivos creados desde el código de Arduino y los convertimos en archivos ejecutables.

```
chmod +x *.sh
```

Verificamos el funcionamiento

```
root@Arduino:~# snmpget -v 2c -c public 127.0.0.1 1.3.6.1.3.1.101.1
```

Donde nos devolverá la temperatura del primer sensor.

```
iso.3.6.1.3.1.101.1 = STRING: "30"
```

ANEXO C: INSTALACIÓN Y CONFIGURACIÓN DE NAGIOS CORE 4.2.4

Antes de comenzar nuestra instalación, necesitamos inhabilitar SELinux. setenforce 0

Modificamos en la dirección / etc / selinux / config y cambiamos el enforcing a disabled.

Requisitos de Instalación

Ahora necesitamos instalar las dependencias que Nagios Core necesitará para funcionar, utilizamos el comando yum.

```
yum install httpd php php-cli gcc glibc glibc-common gd gd-devel net-snmp net-snmp-utils  
openssl-devel wget unzip -y
```

Continuamos con la creación de un usuario y un grupo para usar Nagios

```
useradd nagios  
groupadd nagcmd  
usermod -a -G nagcmd nagios  
usermod -a -G nagcmd apache
```

Instalación de Nagios

Descargamos los archivos .tar.gz necesarios para los complementos Nagios y Nagios Plugins

Ingresamos a la dirección

```
cd / tmp  
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.2.4.tar.gz  
wget http://www.nagios-plugins.org/download/nagios-plugins-2.1.4.tar.gz  
tar xzf nagios-4.2.4.tar.gz  
tar xzf nagios-plugins-2.1.4.tar.gz
```

Una vez extraídos los archivos de los dos paquetes ingresamos a la carpeta nagios y digitamos los comandos de instalación.

```
cd nagios-4.2.4  
./configure --with-command-group = nagcmd  
make all  
make install  
make install-init  
make install-config  
make install-commandmode  
make install-webconf
```

Creación de una contraseña para nagiosadmin

Procedemos a crear una contraseña para el usuario nagiosadmin. Esto se utilizará para iniciar sesión en la interfaz web principal.

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Instalar complementos de Nagios

Ahora que Nagios está instalado, necesitamos instalar los plugins para que pueda utilizarlos.

Ingresamos a la dirección e instalamos los plugins.

```
cd /tmp/nagios-plugins-2.1.4
./configure --with-nagios-user = nagios --con-nagios-group = nagios --with-openssl
make all
make install
```

Iniciar Nagios

Para iniciar Nagios, debemos iniciar los servicios httpd y nagios.

```
service httpd start
service nagios start
```

Ahora en un navegador accedemos a <http://nagiosip/nagios/>

Solución de problemas

Si no se puede acceder a la página web de Nagios, puede estar relacionado con reglas de firewall o puerto.

ANEXO D: INSTALACIÓN Y CONFIGURACIÓN DE NAGIOSGRAPH

Para la instalación de las dependencias necesarias utilizamos el comando yum.

```
yum -y install perl-GD php-gd rrdtool-perl rrdtool-php rrdtool perl-CGI perl-Time-HiRes
```

creamos un directorio en donde descargaremos nagiosgraph 1.5.2

```
mkdir -p /soft/nagios
```

```
cd /soft/nagios
```

```
wget https://sourceforge.net/projects/nagiosgraph/files/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz
```

extraemos el contenido de los paquetes

```
cd /soft/nagios
```

```
tar xvf nagiosgraph-1.5.2.tar.gz
```

procedemos a iniciar la instalación

```
cd /soft/nagios/nagiosgraph-1.5.2
```

```
./install.pl --check-prereq
```

```
./install.pl --layout overlay --prefix /usr/local/nagios
```

En el siguiente paso instalamos en las direcciones por defecto.

Configuración de archivos de nagiosgraph

al finalizar el procedimiento definimos el comando al cual va a ser referencia los gráficos.

```
vi /usr/local/nagios/etc/objects/commands.cfg
```

en el archivo agregamos las siguientes líneas.

```
define command {  
    command_name process-service-perfdata-for-nagiosgraph  
    command_line /usr/local/nagios/libexec/insert.pl  
}
```

Verificamos que no existan errores en las configuraciones de Nagios

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Reiniciamos los servicios Nagios y http

```
service nagios restart
```

service httpd restart

editamos el archivo de ejemplos para definir el servicio

```
vi /usr/local/nagios/etc/objects/templates.cfg
```

al final del archivo agregamos las siguientes líneas

```
define service {  
    name          graphed-service  
    action_url    /nagios/cgi-bin/show.cgi?host=$HOSTNAME&&service=$SERVICEDESC'  
onMouseOver='showGraphPopup(this)' onMouseOut='hideGraphPopup()' rel='/nagiosgraph/cgi-  
bin/showgraph.cgi?host=$HOSTNAME&&service=$SERVICEDESC&&period=week&rrdopts=  
-w+450+-j  
    register     0  
}
```

Para hacer referencia al servicio de gráficos lo llamamos desde el archivo de servicio de chequeo.

```
vi /usr/local/nagios/etc/objects/arduino.cfg
```

y en el archivo agregamos el servicio

```
define service {  
    use          local-service,graphed-service    ; Name of service template to use  
    host_name    localhost  
    service_description HTTP  
    check_command check_http  
    notifications_enabled 0  
}
```


ANEXO E: POLÍTICAS PARA LA SEGURIDAD FÍSICA Y AMBIENTAL DEL CENTRO DE DATOS.

El administrador o responsable del Centro de Datos deberá ser la persona que garantice el cumplimiento de las siguientes políticas.

Políticas para la seguridad física del Centro de Datos

- Las personas que requieran acceso deberán realizar una solicitud dirigida al personal que administra el Centro de Datos, en donde conste los detalles de nombre, hora ingreso, hora de salida, fecha y motivo de acceso.
- Las personas que resulten autorizadas la solicitud por el administrador deberán realizarlo bajo la supervisión y recomendaciones del administrador.
- El ingreso al Centro de Datos es exclusivamente para procesos de investigación o desarrollo, descartándose todo tipo de ocio dentro del mismo.
- En el proceso de Desarrollo o implementación, procurar que las condiciones de limpieza sean las más adecuadas, sin dejar desperdicios en el interior del Centro de Datos
- Por ningún motivo la puerta de acceso deberá permanecer entre-abierta ya que esto permite la salida del aire refrigerado y afecta el sistema de climatización interna.
- Queda prohibido utilizar el Centro de Datos como almacén bodega, para elementos ajenos al campo de redes y telecomunicaciones.
- Toda visita académica deberá ser bajo supervisión y procurar que los visitantes posean guantes o pulseras antiestáticas durante la manipulación de algún equipo o componente.
- Queda prohibido el ingreso de líquidos, al interior del Centro de Datos ya que el derrame accidental puede provocar severos daños en el sistema eléctrico.

Políticas para el funcionamiento y uso del sistema de monitoreo.

- Las personas autorizadas en el ingreso al Centro de Datos por ningún motivo deberán apagar el sistema de enfriamiento (aire acondicionado).
- Queda terminantemente prohibido encender elementos que produzcan calor, fuego o humo ya que esto activara las alarmas implementadas.
- En caso de no visualización de caracteres en la pantalla LCD encendida, verificar el control de luminosidad (potenciómetro), ubicado en la caja del sistema de monitoreo.
- En caso de mal funcionamiento y remplazo de algún sensor, verificar las conexiones establecidas en el capítulo 3, sección conexiones de sensores.
- En caso de remplazo de algún sensor o mantenimiento del sistema, es de preferencia detener el sistema de monitoreo, de no hacerlo, seguirá emitiendo alertas.
- En caso de pérdida de energía en el servidor físico, contactar al administrador de la plataforma Cloud, para el restablecimiento de la máquina virtual, debido al paso a un estado desconocido no modificable desde la máquina virtual.
- Verificar por lo menos una vez a la semana la interfaz web del sistema de monitoreo, para evidenciar el funcionamiento y emisión de alertas en los registros.

Supervisión de amenazas físicas en el centro de datos

White Paper 102

Revisión 3

Por Christian Cowan y Chris Gaskins

> Resumen Ejecutivo

Las metodologías tradicionales para la supervisión del entorno de los centros de datos ya no son suficientes. Las nuevas tecnologías (como los servidores Blade) elevan las necesidades de refrigeración, y las nuevas leyes (como la Ley Sarbanes-Oxley) aumentan los requisitos de seguridad. Por lo tanto, el entorno físico del centro de datos debe controlarse más de cerca. Aunque existen protocolos bien conocidos para la supervisión de dispositivos físicos, como sistemas SAI, equipos de aire acondicionado de salas de ordenadores y sistemas de extinción de incendios, hay una serie de puntos de supervisión distribuidos que no suele tenerse en cuenta. En este documento se describen este tipo de amenazas, se sugieren estrategias para la implementación de dispositivos de supervisión y se ofrecen prácticas recomendadas para aprovechar los datos recopilados con el fin de reducir el tiempo de inactividad.

Contenido

haga clic en una sección para saltar a ella

Introducción	2
¿Qué son las amenazas físicas distribuidas?	2
Ubicación de los sensores	5
Recopilación de datos de los sensores	8
Acción "inteligente"	8
Método de diseño	12
Ejemplo de distribución de sensores	13
Conclusión	14
Recursos	15

Imagen 1

Amenazas para el centro de datos

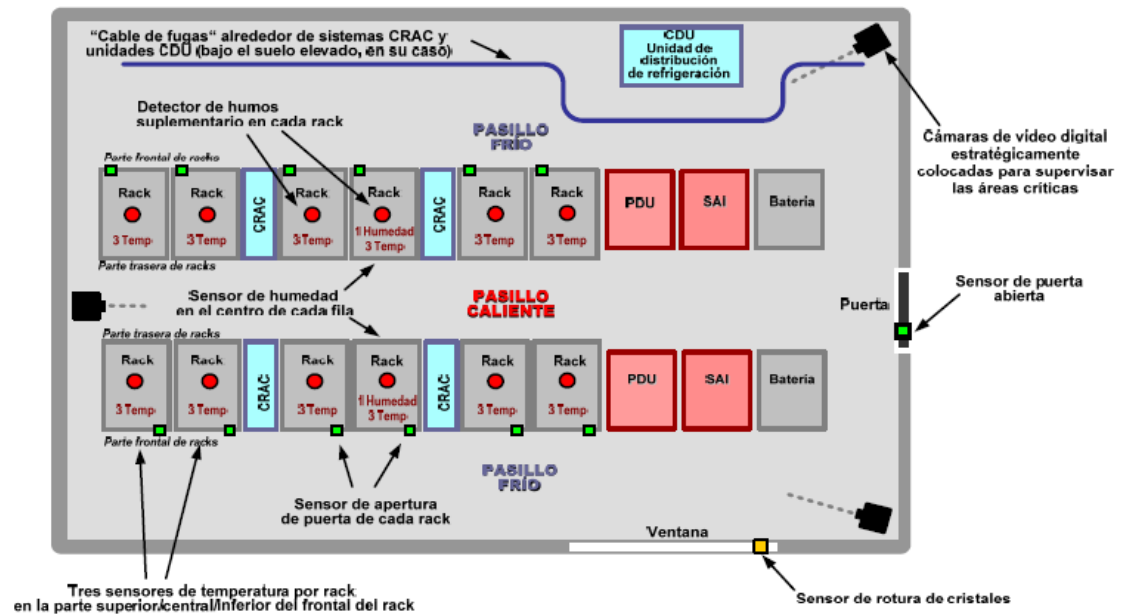


Ejemplo de distribución de sensores

En la **Imagen 3** se muestra un ejemplo de diseño de centro de datos en el que se ilustra la ubicación de dispositivos de supervisión de acuerdo con las prácticas recomendadas descritas en este documento.

Imagen 3

Ejemplo de distribución de sensores



ANEXO G: HOJA DE DATOS SENSOR MQ-2

HANWEI ELETRONICS CO.,LTD

MQ-2

<http://www.hwsensor.com>

TECHNICAL DATA

MQ-2 GAS SENSOR

FEATURES

Wide detecting scope
Stable and long life

Fast response and High sensitivity
Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 800mw	

B. Environment condition

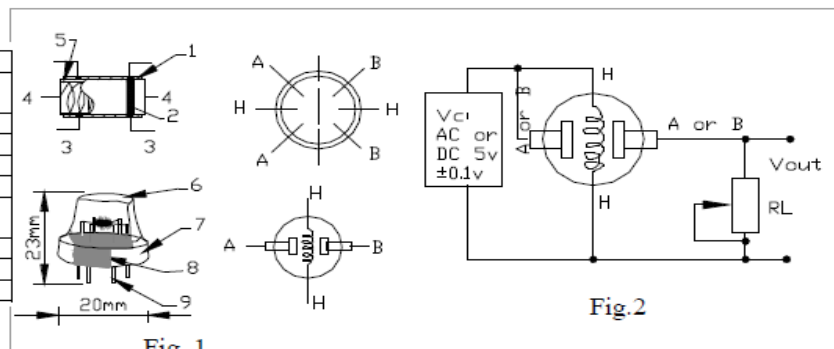
Symbol	Parameter name	Technical condition	Remarks
T _{ao}	Using Tem	-20℃-50℃	
T _{as}	Storage Tem	-20℃-70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

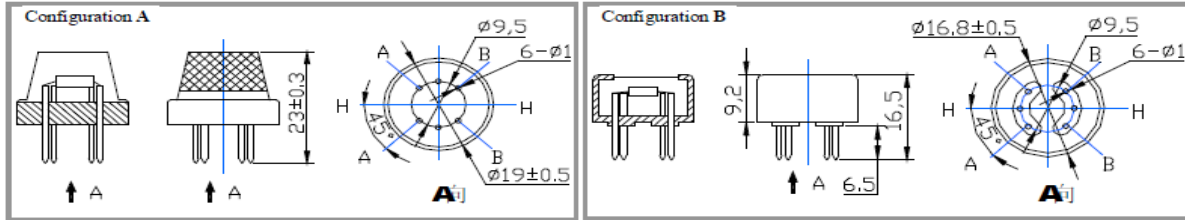
C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
R _s	Sensing Resistance	3K Ω -30K Ω (1000ppm iso-butane)	Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm H ₂ 100ppm-2000ppm Alcohol
α (3000/1000) isobutane	Concentration Slope rate	≤0.6	
Standard Detecting Condition	Temp: 20℃±2℃ Humidity: 65%±5%	V _c :5V±0.1 V _H : 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

Parts	Materials
1 Gas sensing layer	SnO ₂
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceranic	Al ₂ O ₃
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni





Structure and configuration of MQ-2 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a

crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

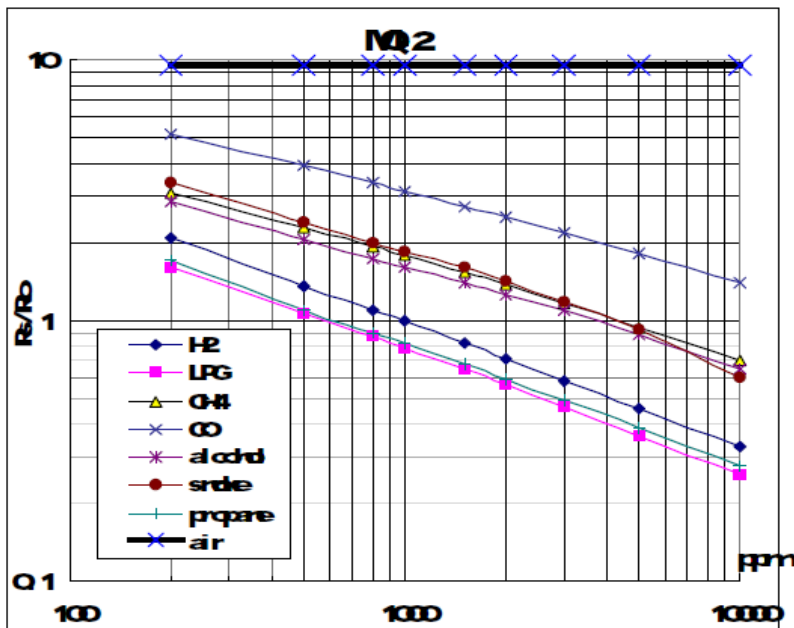


Fig.2 sensitivity characteristics of the MQ-2

Fig.3 is shows the typical sensitivity characteristics of the MQ-2 for several gases.

in their: Temp: 20°C,
Humidity: 65%,
 O_2 concentration 21%
 $R_L=5k\Omega$

R_0 : sensor resistance at 1000ppm of H_2 in the clean air.
 R_s : sensor resistance at various concentrations of gases.

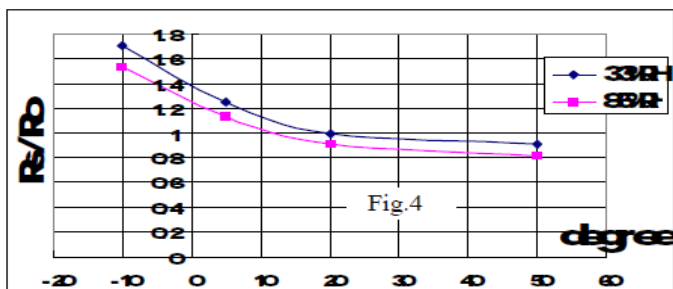


Fig.4 is shows the typical dependence of the MQ-2 on temperature and humidity.

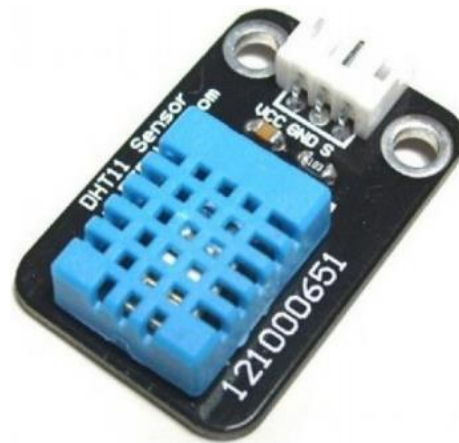
R_0 : sensor resistance at 1000ppm of H_2 in air at 33%RH and 20 degree.
 R_s : sensor resistance at 1000ppm of H_2 at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-2 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 1000ppm liquified petroleum gas<LPG>, or 1000ppm iso-butane< $i-C_4H_{10}$ > concentration in air and use value of Load resistance that (R_L) about 20 K Ω (5K Ω to 47 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

ANEXO H: HOJA DE DATOS SENSOR DHT11



1. Introduction

This DFRobot DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	±5%RH	±2°C	1	4 Pin Single Row

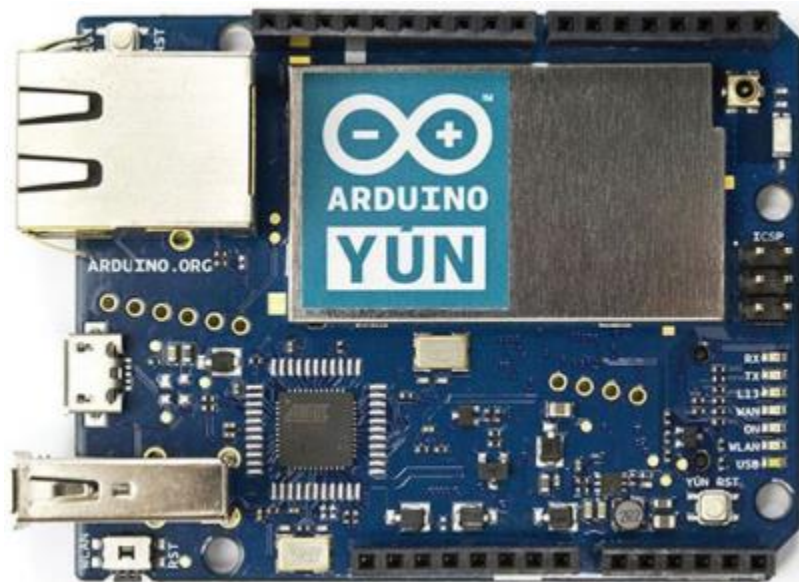
Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			± 1%RH	
Accuracy	25°C		± 4%RH	
	0-50°C			± 5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
Temperature				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			± 1°C	
Accuracy		± 1°C		± 2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

VDD=5V, T = 25°C (unless otherwise stated)

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

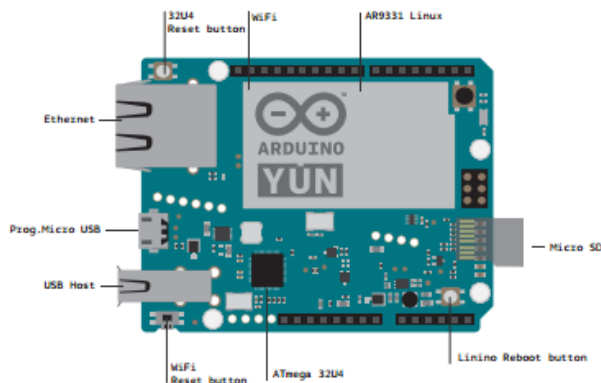
Note: Sampling period at intervals should be no less than 1 second.

ANEXO I: HOJA DE DATOS ARDUINO YUN



Overview

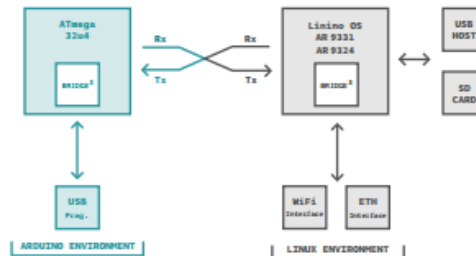
Arduino Yún is a microcontroller board based on the ATmega32u4 (datasheet) and the Atheros AR9331. The Atheros processor supports a Linux distribution based on OpenWrt named Linino OS. The board has built-in Ethernet and WiFi support, a USB-A port, micro-SD card slot, 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, an ICSP header, and a 3 reset buttons.



The Yún distinguishes itself from other Arduino boards in that it can communicate with the Linux distribution onboard, offering a powerful networked computer with the ease of Arduino. In addition to Linux commands like cURL, you can write your own shell and python scripts for robust interactions.

The Yún is similar to the Leonardo in that the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Yún to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port.

The Bridge library facilitates communication between the two processors, giving Arduino sketches the ability to run shell scripts, communicate with network interfaces, and receive information from the AR9331 processor. The USB host, network interfaces and SD card are not connected to the 32U4, but the AR9331, and the Bridge library also enables the Arduino to interface with those peripherals.



Description

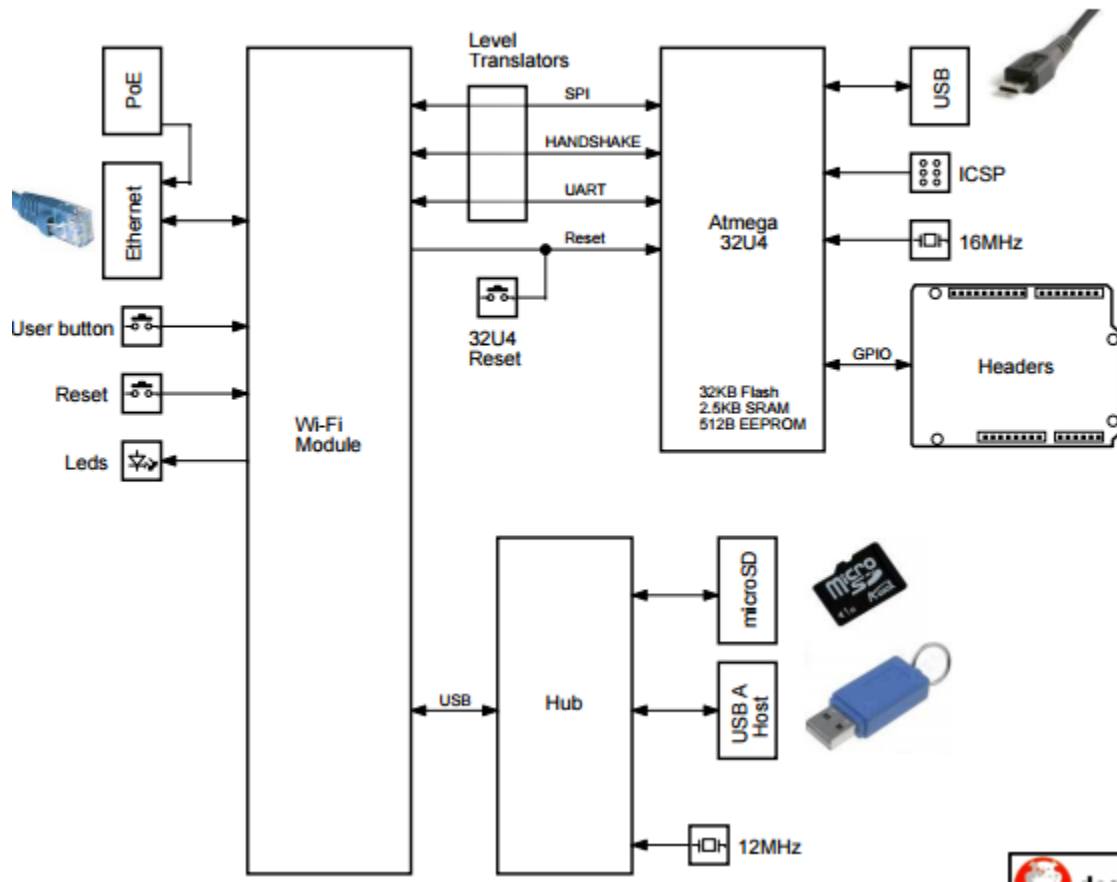
AVR Microcontroller

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage	5V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (of which 4 KB used by bootloader)
SRAM	2.5 KB
EEPROM	1 KB
Clock Speed	16 MHz

Linux microprocessor

Processor	Atheros AR9331
Architecture	MIPS @400MHz
Operating Voltage	3.3V
Ethernet	IEEE 802.3 10/100Mbit/s
WiFi	IEEE 802.11b/g/n
USB Type-A	2.0 Host
Card Reader	Micro-SD only
RAM	64 MB DDR2
Flash Memory	16 MB
PoE compatible 802.3af card support (see the note below)	
Length	73 mm
Width	53 mm
Weight	32 g

Block Diagram



BIBLIOGRAFÍA

- ARDUINO. (s.f.). Obtenido de www.arduino.cc
- ARDUINO. (2016). Obtenido de <https://www.arduino.cc/en/Guide/ArduinoYun>
- Arduino. (2017). Obtenido de <https://www.arduino.cc/en/Main/ArduinoBoardYun>
- Arduino. (2017). *Arduino*. Obtenido de <https://www.arduino.cc/en/Main/ArduinoBoardYun>
- Barba, A. (1999). *Gestion de red*.
- datasheet4u*. (2017). Obtenido de <http://www.datasheet4u.com/pdf/DHT22-pdf/792209>
- Diseño de la infraestructura física de un Data Center TIER I basado en el estándar TIA 942, para la Facultad de Ingeniería de la Universidad Técnica del Norte. (2016). Ibarra, Imbabura.
- electronilab.co*. (s.f.). Obtenido de <http://electronilab.co/tienda/sensor-de-temperatura-y-humedad-dht11/>
- Gaskins, C. (2016). *apc*. Obtenido de http://www.apc.com/salestools/JMON-5ZLP8M/JMON-5ZLP8M_R3_LS.pdf
- <http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>. (s.f.). Obtenido de <http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>:
<http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>
- Huidrobo, B. &. (2008, pag. 202).
- ISO/IEC 17799: Tecnología de la información - Técnicas de seguridad- Código para la práctica de la gestión de la seguridad de la información*. (15 de Junio de 2005). Obtenido de <file:///C:/Users/HOUSE/Downloads/iso-17799-2005-espa%C3%B1ol.pdf>
- Nagios. (2016). Obtenido de <https://www.nagios.org/>
- naylampmechatronics.com*. (s.f.). Obtenido de http://www.naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html
- opennebula.org*. (2017). Obtenido de <https://opennebula.org/about/technology/>
- Pacio, G. (2014). *Data Centers Hoy*. Buenos Aires: Alfaomega.
- Pandora. (2016). Obtenido de <https://pandorafms.org/es/>
- Stallings. (2004, p.263). *Fundamentos de seguridad en redes*.
- tienda.bricogeek.com*. (s.f.). Obtenido de <http://tienda.bricogeek.com/sensores-gas/469-sensor-de-gas-metano-mq-2.html>

todoelectrodo. (s.f.). Obtenido de todoelectrodo: <http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>

todoelectrodo. (2013). Obtenido de todoelectrodo:
<http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html>

yapo. (2017). Obtenido de
http://www.yapo.cl/biobio/computadores/sensor_de_gas__combustible_y_humo_mq_2_2_8300330.htm

Zenoss. (2016). Obtenido de <https://serviciosderednoona.wordpress.com/sistema-de-monitoreo-zenoss-4-2/>