UNIVERSIDAD DE ORIENTE
NÚCLEO DE SUCRE
ESCUELA DE CIENCIAS
DEPARTAMENTO DE INFORMATICA

SISTEMA DE MONITOREO AMBIENTAL BASADO EN ARDUINO Y NAGIOS PARA EL CENTRO DE DATOS DEL RECTORADO DE LA UNIVERSIDAD DE ORIENTE

(Modalidad: Pasantía de grado)

DISBELYS DEL CARMEN DIÁZ MILLÁN

TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA OPTAR AL TÍTULO DE LICENCIADO EN INFORMÁTICA

CUMANÁ, 2017

SISTEMA DE MONITOREO AMBIENTAL BASADO EN ARDUINO Y NAGIOS PARA EL CENTRO DE DATOS DEL RECTORADO DE LA UNIVERSIDAD DE ORIENTE

APROBADO POR:

_____

Prof. José Mendoza.
Asesor Académico

_____

Lic. José Luna
Asesor Institucional

_____

Prof. Carmelys Rodríguez
(Jurado)

_____

Prof. José Romero
(Jurado)

INDICE

DEDICATORIA

Este logro en mi vida profesional se lo dedico a Dios y al Divino Niño Jesús

A mis padres, a quienes amo como a nada en este mundo, Arquímedes Rafael Díaz y Carmen Saturnina Millán de Díaz; por darme todo su apoyo, amor y fuerza.

A mi hermano Daniel Rafael Díaz por apoyarme siempre que lo necesitaba.

A mis amigos y a todas aquellas personas especiales que estuvieron a mi lado en los momentos más difíciles de este largo recorrido.

AGRADECIMIENTOS

A:

Mis padres Arquímedes Rafael Díaz y Carmen Saturnina Millán de Díaz por brindarme, sus consejos y regaños, cada aspecto me hizo ser la persona que soy.

Mi hermano Daniel Rafael Díaz por estar siempre conmigo cuidándome.

La Universidad de Oriente, por ser mi segunda casa durante todos estos años.

Mi asesor académico Prof. José Mendoza, por su asesoría con este proyecto.

Mi asesor institucional Lic. José Luna y a la Coordinación de teleinformática de la Universidad de Oriente por su colaboración y apoyo en este arduo trabajo.

Carlos Rodríguez por asesorarme y prestarme su apoyo cada vez que lo necesitaba.

Luis Fernando Rivero por su colaboración en la parte de modelado 3D y por ayudarme en este tramo.

Marizel Gil, Humberto Gil, Carlos Aguilar, Beatriz García, Luis Freites y Pedro Cova por ayudarme cuando más los necesitaba y brindarme su cariño y amistad, más que amigos familia.

Cada una de las personas, fuera y dentro de la Universidad, que pusieron su granito de arena en mi aprendizaje.

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág.

RESUMEN

Se desarrolló un sistema de monitoreo ambiental basado en Arduino (hardware) y Nagios (software) para el Centro de Datos de la Universidad de Oriente (UDO) Venezuela, de bajo costo y adaptable. Primero se analizó las necesidades y requerimientos del Centro de Datos UDO, luego se realizó un estudio de la distribución de temperatura, se determinó cuantos sensores y de qué forma estarían ubicados, después se realizó el *script* (instrucciones) para la comunicación entre Arduino y Nagios. Para este proyecto se aplicó la metodología *Rational Unified Process* (RUP, en español Proceso racional unificado), la cual se compone de cuatro (4) fases: inicio, elaboración, construcción y transición. En la fase de inicio se recolectó información para construir la visión del negocio, se recopilaron los requisitos, se plantearon las pruebas, se modelaron los actores y se delimitaron los casos de uso, sobre los cuales se enfocaría el proyecto. En la fase de elaboración se refinaron algunos diagramas, se identificó los equipos necesarios para las actividades de monitoreo, se recolectó información referente a los mismos, se diseñó el *script* para la comunicación entre Nagios y Arduino, también se diseñó la interfaz web para visualizar los cambios de temperatura y humedad a tiempo real. En la fase de construcción se realizó la integración correcta entre Nagios y Arduino, se configuraron los *host* (huésped) y servicios para su buen funcionamiento y fue verificada la correcta adquisición de datos. Al final, se obtuvo el sistema de monitoreo ambiental para el Centro de Datos UDO, el cual capturó datos de temperatura y humedad, para verificar la estabilidad ambiental del mismo, con el fin de corregir problemas y tomar acciones preventivas frente a los diferentes riesgos que se puedan presentar. A este sistema se puede acceder a través de cualquier dispositivo electrónico con acceso a Internet.El sistema de monitoreo ambiental tiene la cualidad de ser configurable y escalable a medida, asegurando como consecuencia la continuidad de los equipos informáticos del Centro de Datos UDO.

Palabras o frases claves: Monitoreo Ambiental, Arduino, Nagios, Centro de Datos.

INTRODUCCIÓN

Según la Real Academia Española (2001), el origen del monitoreo se encuentra en monitor, un aparato receptor que toma las imágenes directamente de las instalaciones filmadoras y sensores, el cual sirve para controlar la transmisión. El monitor, por lo tanto, ayuda a controlar o supervisar una situación. Esto permite inferir que monitoreo es la acción y efecto de monitorear, el verbo que se utiliza para nombrar a la supervisión o el control realizado a través de un monitor. Por extensión, el monitoreo es cualquier acción de ese tipo, más allá de la utilización de un monitor.

Asimismo Rivera (2005), dice que el proceso de monitoreo es cíclico, es decir, rota continuamente en torno a las diferentes etapas funcionales desde la captura de datos hasta la implementación. La siguiente ilustración muestra los elementos del ciclo de monitoreo, y las relaciones que guardan entre sí. En el orden de la ilustración, los elementos del ciclo del monitoreo se describen a continuación:

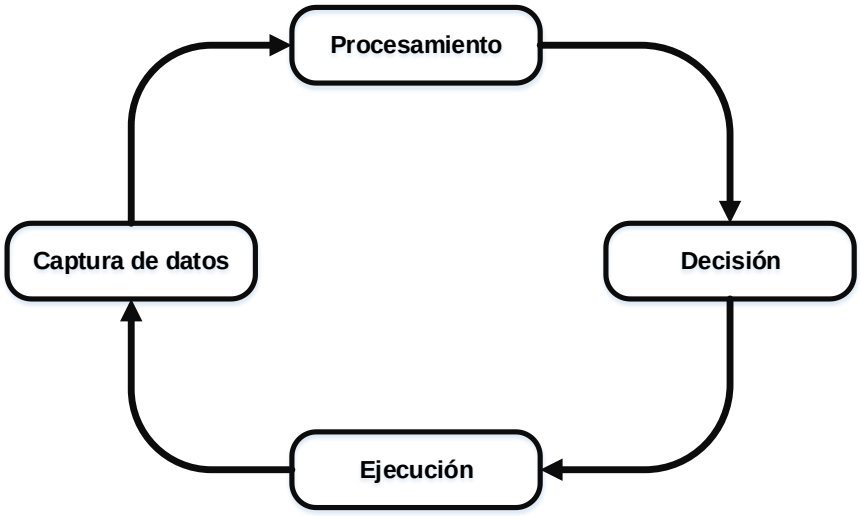


Figura 1. Esquema de la secuencia de monitoreo.
Fuente: Rivera (2005)

En la figura 1, Se representa un sistema de monitoreo en forma general, se inicia con la captura de datos o señales, de la fuente establecida y posterior registro en los instrumentos respectivos; luego pasa al procesamiento de los datos para la obtención de información y así tomar una decisión, con respecto a las acciones correctivas o de retroalimentación necesarias de acuerdo a la información obtenida; por último esta la ejecución, que pondrá en práctica las acciones correctivas o de retroalimentación.

El monitoreo ambiental es la recolección sistemática de datos mediante mediciones u observaciones en series del espacio y tiempo, a través de variables y/o indicadores, estas variables proporcionan una muestra representativa del medio ambiente, por lo tanto la información obtenida es utilizada para evaluar el estado actual del lugar monitoreado (Torres y col, 2012).

Según Torres (2010), el Centro de Datos es aquel lugar donde se concentran los recursos físicos y lógicos necesarios para el procesamiento de la información de una institución u organización.

Durante los últimos años, los Centros de Datos se han convertido en la columna vertebral de cualquier organización o institución, ya que éstos son utilizados para almacenar grandes cantidades de información para la planificación, ejecución y el buen funcionamiento de la organización o institución que depende de un 90% del Centro de Datos.

Entre los principales aspectos que se debe considerar para que el Centro de Datos sea eficiente y eficaz, es tener los equipos adecuados de refrigeración, también un sistema de control y seguridad; la conectividad entre equipos y una distribución de aire de forma fluida.

La técnica más común que se utilizaba para monitorear el entorno de los Centro de Datos se remonta a los días de las computadoras centralizadas hace más de 50 años, e incluyen prácticas como caminar por la habitación con termómetros y confiar en que el personal del área de informática perciba cómo está el ambiente en la habitación, de esta forma los datos que se capturaban no eran confiables y no se podía tomar decisiones seguras a la hora de una situación no esperada (López, 2010).

Sin embargo, a medida que los centros de datos continúan evolucionando y eleven las necesidades de alimentación y refrigeración, el entorno debe controlarse más

Para monitorear el medio ambiente del Centro de Datos UDO se aplicó un sistema que permita medir las variables de temperatura y humedad a través de sensores conectados a una tarjeta Arduino (Arduino, 2010) con un entorno de programación multimedia, la cual permite registrar y acondicionar señales analógicas y digitales.

Para complementar de lo que trata la plataforma Arduino se tiene que:

> "... es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos. Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El micro-controlador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring, plataforma de prototipos electrónicos) y el entorno de desarrollo Arduino (basado en Processing, lenguaje de programación). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Processing, entre otros)" (Banzi, 2012).

El sistema que es usado en el Centro de Datos UDO para seguir el funcionamiento de redes y equipos, es Nagios (2009), que según González (2005), lo define como una aplicación para la monitorización de redes y los recursos de sistemas hardware (carga del procesador, uso de los discos, memoria, estado de los puertos), alertando cuando el

comportamiento de los mismos no es el deseado. Éste sistema también puede aceptar los datos capturados por Arduino.

La importancia de un sistema de monitoreo para las condiciones ambientales en el interior de los Centro de Datos se ha incrementado a lo largo de los años. Desde el objetivo inicial de comprobar si las condiciones satisfacen las necesidades de los equipos instalados, generando alarmas cuando sea necesario, cuando no es así, se han convertido en vital para la planificación de mejoras de los Centros de Datos.

En el Centro de Datos UDO, y en otras similares o más grandes, es muy importante mantener las condiciones óptimas de temperatura y humedad. En consecuencia, es una necesidad el poder monitorear el medio ambiente y ser capaz de activar las alarmas cuando se opera fuera de los valores recomendados. Por este motivo se desea monitorear el medio ambiente del Centro de Datos UDO.

Este trabajo está dividido en tres capítulos, estructurados de la siguiente manera: capítulo I, está compuesto por el planteamiento del problema, alcance y las limitaciones del sistema desarrollado, capítulo II, el cual se compone de dos secciones: Marco teórico, donde se exponen las bases teóricas necesarias para sustentar la investigación, está sección está formada por los antecedentes tanto de la investigación como de la organización y el marco metodológico, que presenta la metodología y el diseño de la investigación, los instrumentos de recolección de datos y explica detalladamente la metodología del área aplicada para el desarrollo del trabajo. Capítulo III, presenta en forma detallada la aplicación de los procedimientos en el marco metodológico para el logro de los objetivos planteados, explicando cada uno de los pasos realizados en el desarrollo del sistema con descripciones, figuras y diagramas que permiten una mejor visualización y entendimiento del sistema desarrollado, Por último, se presentan las conclusiones, recomendaciones, bibliografía, apéndices y anexos.

CAPÍTULO I. PRESENTACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

La Universidad de Oriente (UDO) cuenta con un Centro de Datos, siendo este un espacio acondicionado, que alberga equipos, sistemas y tecnología de la información. Cuando se dice acondicionado, se refiere a que es un lugar que tiene instalado lo siguiente: climatización (aire acondicionado), alimentación eléctrica estabilizada e ininterrumpida, cableado estructurado, sistemas contra incendios, control de acceso y sistemas de cámaras de vigilancia.

Los equipos de refrigeración y la adecuada distribución del aire se encuentran relacionados con la temperatura y humedad del Centro de Datos; los sistemas de control y seguridad, llámese acceso de personal y barreras informáticas, estará relacionados con las características con que está diseñado el Centro de Datos.

Para tener un rendimiento óptimo y un buen funcionamiento de los equipos que se encuentran en el Centro de Datos UDO se debe mantener un constante monitoreo y tener conocimiento de la cantidad de sistemas que estarán funcionando en las horas pico de procesamiento y de los equipos que tienen un elevado consumo de energía, ya que estos generan gran cantidad de calor. Los equipos electrónicos que aquí se encuentran, generalmente están construidos por semiconductores, los cuales se caracterizan por tener la propiedad de que a mayor trabajo, implica aumento de la potencia, esto trae como consecuencia aumento de la temperatura del sistema.

El aumento de la temperatura de los dispositivos semiconductores conlleva, por supuesto, a incrementar el calor, y este calor genera más corriente, ya que en los sistemas semiconductores cada aumento de 10 ºC en la temperatura se duplica la corriente, al duplicarse la corriente aumenta la potencia generando más calor, si este incremento es excesivo e incontrolado, inicialmente provocará una reducción de la vida útil del elemento y en el peor de los casos lo destruirá (Boylestad y Nashelsky, 2003).

La problemática que se encuentra es que si la temperatura y humedad está fuera de los parámetros señalados pueden dañar los equipos y ocasionar pérdidas de información o error de comunicación. Los dispositivos electrónicos son sensibles a los cambios en la temperatura, siendo esto aún más crítico en equipos que se encuentran trabajando las veinticuatro horas del día.

Este aumento en temperatura puede ocasionar que los servidores salgan de servicio o se "caigan", privando a la comunidad universitaria de ciertos tipos de herramientas, alterando la eficiencia de estos y comprometiendo el nivel de servicio.

El problema que se menciona anteriormente se ha presentado en diversas ocasiones, cuando debido a fallos del sistema de aire acondicionado, la temperatura del cuarto de servidores se ha elevado a rangos inaceptables que comprometen el buen funcionamiento del equipo.

En el de Centro de Datos UDO no existe un monitoreo de temperatura y humedad, aunque Nagios puede registrar el incremento de calor dentro los sistemas de cómputo, sin embargo, esté no puede monitorear la temperatura del medio ambiente donde se encuentran ubicados los equipos.

El sistema de monitoreo ambiental para el Centro de Datos UDO no se ha implementado por falta de tiempo para su desarrollo. Es importante mantener una correcta administración que no solo consista en un simple monitoreo de red, si no que se involucre mucho más, realizando una gestión con herramientas que proporcionen una información más completa, amplia y eficiente que ayude el correcto control, uso y desempeño de los equipos, para que de manera sencilla hagan una observación del comportamiento de todo lo utilizado en la red y la inspección de estado por medio de notificaciones.

Es de suma importancia, mantener en buen estado los equipos que se emplea para brindar los servicios ofrecidos por la Universidad de Oriente, así como evitar todo tipo de situación o escenario que pueda comprometer el buen funcionamiento de los mismos.

Por esto se propone un sistema de monitoreo ambiental basado en Arduino y Nagios para el Centro de Datos del rectorado de la Universidad de Oriente que permita alertar a los encargados, saber cuándo la temperatura y humedad se sale de los estándares normales y tomar decisiones en tiempo real.

1.1.1 Propuesta para solucionar el problema

Se propone un sistema de monitoreo ambiental basado en Arduino y Nagios para el Centro de Datos del rectorado de la Universidad de Oriente. Al existir posibles variaciones de temperatura y humedad en el cuarto de servidores, es indispensable tener un sistema capaz de detectar dichos cambios, y que además pueda de una forma inmediata enviar diferentes señales de alarma, para que las personas a cargo, puedan atender la emergencia suscitada, evitando un problema mayor. Así que teniendo estas consideraciones de por medio, es necesario tener un sistema sensor que sea capaz de reconocer un rango de temperatura aceptable, así como un dispositivo que reciba la información detectada por el sensor y que facilite el envío de datos a un computador personal, y este a su vez posea un software que envíe diferentes mensajes por medio de la red interna así como Internet, alertando que la temperatura ha salido de rango

1.2 JUSTIFICACIÓN DE LA INVESTIGACIÓN

La Coordinacion de Teleinformatica de la UDO no cuenta con un sistema de monitoreo ambiental de temperatura para el Centro de Datos a pesar de que allí alberga toda la informacion necesaria y de suma importancia que requiere estudiantes y personal de la Institucion, debido a la situacion anterior, es necesario diseñar y construir un prototipo de sistema de monitoreo ambiental con la finalidad de que el operador del Centro de Datos pueda mantenerse informado las veinticuatro (24) horas del día durante todo el año acerca de las condiciones ambientales del mismo, a través de correo electrónico y la Internet, el cual le proporcionará información de manera oportuna que le ayudará en la toma de decisiones a la hora de un evento no deseado, y así mantenerlo operativo garantizando un eficiente funcionamiento y un servicio de calidad a la comunidad universitaria, además de resguarda los datos que allí se encuentran.

Cuanta más información se tenga sobre las condiciones del Centro de Datos, mejorará el panorama para tomar decisiones sobre la optimización, expansión y utilización del mismo, por ello se propone un monitoreo ambiental para seguir de forma proactiva las amenazas físicas del mismo.

1.3 OBJETIVOS

GENERAL

Desarrollar un sistema de monitoreo ambiental basado en Arduino y Nagios para el Centro de Datos de la Universidad de Oriente.

ESPECÍFICOS

Analizar los requerimientos y necesidades del Centro de Datos

Realizar un estudio detallado de la distribución de temperatura en todo el cuarto de servidores.

Determinar la cantidad y el tipo de sensores necesarios para obtener una medición correcta de la temperatura y humedad.

Elaborar un script (instrucciones) en lenguaje de alto nivel, un método de comunicación entre la plataforma Arduino y el sistema Nagios.

Elaborar una aplicación web para que se visualice las variables de temperatura y humedad en tiempo real

Probar e integrar los diferentes componentes del sistema.

1.4 ALCANCE Y LIMITACIONES

1.4.1 Alcance

Este sistema de monitoreo va dirigido a los monitores y operadores que se encuentran en el Centro de Datos UDO.

El sistema permitirá el monitoreo ambiental de temperatura y humedad, cuya finalidad es mantener al operador informado las 24 horas del día durante todo el año acerca de las condiciones ambientales del mismo, a través de correo electrónico y la Internet.

1.4.2 Limitaciones

No se cuenta con los recursos económicos necesarios para poseer todos los sensores de temperatura y humedad; y abarcar todo el espacio del Centro de Datos UDO. .

CAPÍTULO II. MARCO DE REFERENCIA

2.1 MARCO TEÓRICO

2.1.1 Antecedentes de la investigación

"Los antecedentes reflejan los avances y el estado actual del conocimiento en un área determinada y sirven de modelo o ejemplo para futuras investigaciones" p. 24, Arias (2006). Se describen a continuación una lista de trabajos cuyo tema está vinculado al de la presente investigación y que cumpliendo con la definición descrita anteriormente, sirven de ejemplo para el desarrollo de este trabajo.

Según Gómez (2012), en su trabajo intitulado: "Desarrollo de un sistema de control de climatización de un edificio, basado en sistemas de inteligencia ambiental y dispositivos moviles", realizado como proyecto de fin de carrera en la Universidad Autónoma de Madrid (UAM), donde describe el diseño y desarrollo de un sistema de control de los equipos de climatización de un edificio de oficinas. Su principal objetivo fue optimizar el confort de los usuarios y reducir el consumo energetico, el sistema se ha diseñado bajo los conceptos y tecnologias propuestas por el paradigma de la inteligencia ambiental, aprovechando los avances de las nuevas tecnologias como son las redes de sensores y los telefonos moviles inteligentes.

Del mismo modo Sánchez (2012), en su trabajo intitulado: "Diseño de un sistema de control domótico basado en la plataforma Arduino". Realizado como proyecto de fin de carrera en la Universidad Politécnica de Valencia (UPV), donde se ha descrito los conocimientos básicos para entender que es y como funciona un sistema domótico y como utilizando el hardware libre de Arduino se puede crear un sistema estable con un presupuesto muy inferior al de las viviendas de alta categoria en este trabajo se explica como estan construidas las placas Arduino y el entono de trabajo que dispusieron para programarla.

Del mismo modo Betancourt (2008), en su trabajo intitulado: “Diseño e implementación de un sistema modular para el control de iluminación, temperatura e inundación, en un ambiente de oficinas”, realizado como trabajo de pregrado, en la Universidad Central de Venezuela (UCV), para la implementación instaló dos (2) controladores lógicos programables, conectados a través de modbus TCP/IP, sensores de presencia, infrarrojo pasivos, temperatura, para efectuar las acciones de control, seleccionados e instalados bajo las recomendaciones propuestas en el diseño, permitiendo la implementación del sistema diseñado y la verificación de su correcto funcionamiento.

Según Becerra (2007), en su trabajo intitulado: “Automatización, control y supervisión remota del sistema central de aire acondicionado (agua helada) para un edificio”. Realizado como trabajo de pregrado en la Universidad Central de Venezuela (UCV), donde realizó la automatización, control y supervisión remota del sistema central de aire acondicionado (agua helada) para un edificio, por medio de Internet, basándose en un autómata programable modelo *QUANTUM* de Schneider Electric bajo el lenguaje UNITY PRO. Esto permite a la empresa encargada de la operación del equipo la visualización del sistema y la realización de maniobras para un óptimo funcionamiento, vía Internet (TCP/IP).

Asi mismo Barreto (2014), en su trabajo intitulado: “sistema de información web para monitoreo y seguridad basado en dispositivo Arduino”. Realizado como trabajo de pregrado, en la Universidad de Oriente (UDO), donde desarrolló un sistema de información web para monitoreo y seguridad del Departamento de la Licenciatura en Informática, el cual tiene como objetivo establecer un sistema de video vigilancia el cual provea de elementos necesarios para la captura de imágenes secuenciales, establecer una relación directa entre vigilancia, estudiantes y los profesores que hacen vida en el edificio de matemática a fin de mejorar el problema de la inseguridad, informar en todo momento los acontecimientos acaecidos en el edificio, llevar un registro cronológico detallado de cada uno de los eventos guardados por el sistema web e informar a cada uno de los usuarios ante cualquier eventualidad que se presente.

2.1.2 Antecedentes de la organización

La Universidad de Oriente fue creada el 21 de noviembre de 1958, mediante el Decreto Ley nº 459 publicado en la gaceta oficial de la República de Venezuela nº 25.831 por la junta provisional de Gobierno presidida por el Dr. Edgard Sanabria, siendo Ministro de Educación el Dr. Rafael Pizani, bajo la conducción de su Rector fundador Dr. Luis Manuel Peñalver Comenzó a funcionar el 12 de febrero de 1960, nacimiento de la Universidad de Oriente, un año después, 113 estudiantes y una docena de profesores, en una vieja casona del sector Caigüire de Cumaná, Venezuela, marcan el camino de la fructífera actividad académica de esta casa universitaria.

En el año 1998 la Universidad de Oriente comienza a integrarse a la Red Académica Universitaria Reacciun (Red Académica Cooperativa entre Centros de Investigación y Universidades Nacionales) y raíz de esa integración nace la Coordinación de Teleinformática que viene a constituirse para manejar y encargarse de todo lo referente a la interconexión de la red de datos UDO.

La Coordinación de Teleinformática UDO, tiene como principal objetivo administrar la red Académica - Administrativa de la institución, asimismo servirá para proteger y garantizar el uso seguro de las redes y de la información que por ella circule, al fin de evitar la pérdida o alteración de dicha información por personas ajenas a la institución

2.1.2.1 Misión

Mantener el desarrollo de una plataforma integral de telecomunicaciones en la UDO, que permita enlazar los distintos núcleos (Anzoátegui, Sucre, Maturín; Bolívar y Nueva Esparta) en el área académica y administrativa, así como la prestación de servicios informáticos a la comunidad.

2.1.2.2. Organigrama de la Organización

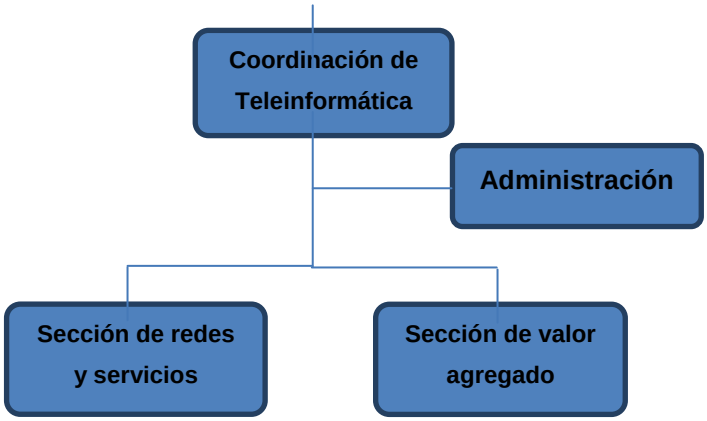


Figura 2. Organigrama de la Coordinación de Teleinformática de la UDO
Fuente: www.udo.edu.ve/ti

2.2 Bases teóricas

Existen muchas áreas dentro de la Licenciatura en informática, pero este proyecto se enfoca en los sistemas de supervisión e instrumentación.

Según Contreras (2001), define supervisión como la habilidad de medir un proceso y actuar en él, formado por componentes interactivos que razonan acerca del comportamiento del proceso para proponer y ejecutar acciones apropiadas para mantener las condiciones de operación normal

Existe una variable manipulada la cual se define como la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada. Normalmente, la variable controlada es la salida del sistema,
Se puede decir que control significa medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar la desviación del valor medido respecto del valor deseado (Ogata, 2010).

Cuyo control será ejecutado por los operadores para prevenir situaciones

A continuación se presentan un conjunto de definiciones fundamentales para un mejor entendimiento:

Un sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado. Un sistema no está necesariamente limitado a los sistemas físicos. El concepto de sistema se puede aplicar a fenómenos abstractos y dinámicos, como los que se encuentran en la economía. Por tanto, la palabra sistema debe interpretarse en un sentido amplio que comprenda sistemas físicos, biológicos, económicos y similares (Ogata, 1998).

Para definir los datos que maneja cualquier sistema, se crean diagramas con el objetivo de que los usuarios puedan entender el trabajo que se realiza y también sirven como apoyo para los desarrolladores a la hora de diseñar y construir.

En las diversas etapas para la creación de diagramas se utilizó el lenguaje UML (*Unified Modeling Language*, siglas en español Lenguaje Unificado De Modelado), definido como un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo tanto las cosas conceptuales, tales como los procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico (Rumbaugh y cols, 1999).

A continuación se presentan algunos conceptos de los diferentes tipos de diagramas. Los diagramas de casos de uso son una colección de situaciones respecto al uso de un sistema. Cada escenario describe una secuencia de eventos. Cada secuencia se inicia por una persona, otro sistema, una parte del hardware o por el paso del tiempo. A las entidades que inician secuencias se les conoce como actores. El resultado de la secuencia debe ser algo utilizable ya sea por el actor que la inició, o por otro que la

inicio (Schmuller, 2002)

En un modelo de caso de uso, una figura agregada representa a un actor, un eclipse a un caso de uso y una línea asociativa representa la comunicación entre el actor y el caso de uso

Los diagramas de clases son un conjunto de objetos que comparten una estructura y comportamientos comunes (Schmuller, 2002). Un diagrama de clases se encuentra formado por varios rectángulos conectados por líneas que muestran la manera en que las clases se relacionan entre sí.

Los diagramas de secuencia son un conjunto de objetos que se representan del modo usual: rectángulos con nombres, mensajes representados por líneas continuas con punta de flechas y el tiempo representado como una progresión vertical. El objetivo de este tipo de diagrama UML es mostrar la forma en que los objetos se comunican entre sí al transcurrir el tiempo (Schmuller, 2002).

En el caso de esta investigación es importante destacar algunos conceptos relacionados con el tema, como lo son aplicación web, páginas y servidores.

Una aplicación web es aquella que los usuarios usan accediendo a un servidor web a través de Internet o de una intranet, las mismas son populares debido a la practicidad del navegador web como cliente de poco consumo de recursos de hardware (Montilva, 2007).

Un servidor web es un programa que corre sobre el servidor que escucha las peticiones que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor web buscará una página web o bien ejecutará un programa en el servidor. Una página de Internet o página web es un documento electrónico que contiene información específica de un tema en particular y que es almacenado en algún sistema de cómputo que se encuentre

conectado a la red mundial de información denominada Internet, de tal forma que este documento pueda ser consultado por cualquier persona que se conecte a esta red mundial de comunicaciones y que cuente con los permisos apropiados para hacerlo. Una página web es la unidad básica del *World Wide Web* (WWW). De cualquier modo un servidor, siempre devolverá algún tipo de resultado HTML, siglas de *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto), al cliente o navegador que realizó la petición (Millar, 1998). HTML es un lenguaje de marcas que define el formato de las páginas que se publican en la web. Una página web está compuesta por distintos elementos (texto, dibujos, tablas, listas) que permiten mostrar información estructurada en los navegadores de los clientes. HTML también facilita la interacción con el usuario a graves de los formularios, éstos dan la posibilidad de que el usuario introduzca datos y genere órdenes para que se procese la información (Gayo, 2000).

Todo el proyecto tiene una planificación en un tiempo estipulado, por este motivo se usa el diagrama de Gantt que permite visualizar la presentación de las actividades, sus duraciones y los momentos en que deben comenzar y terminar cada una de ellas. Es un sencillo gráfico compuesto de barras, donde cada una simboliza una tarea del proyecto (Fowler, 1999).

2.3 MARCO METODOLÓGICO

"El marco metodológico tiene como objeto proporcionar un modelo de verificación que permite constatar hechos con teorías, y su forma es la de una estrategia o plan general que determina las operaciones necesarias para hacerlo"p.17 (Sabino, 2007).

2.3.1 Metodología de la investigación

Se tomó como referencia para el desarrollo de este trabajo de grado la metodología planteada por Sabino (2007).

2.3.1.1 Nivel de investigación

Las investigaciones descriptivas utilizan criterios sistemáticos que permiten poner de manifiesto la estructura o el comportamiento de los fenómenos en estudio, proporcionando de ese modo información sistemática y comparable con la de otras fuentes (Sabino, 2007).

Esta investigación es de tipo descriptiva, ya que comprende la descripción, registro, análisis e interpretación de los procesos y actividades que se llevan a cabo en el Centro de Datos UDO, las cuales se tomarán como asiento para la investigación.

2.3.1.2 Diseño de investigación

El diseño de la investigación es de campo, debido a que los datos de interés se recogen en forma directa de la realidad, mediante el trabajo concreto del investigador y su equipo. Se empleó este tipo de investigación, ya que la información fue recolectada directamente en el Centro de Datos UDO a través de la aplicación de entrevistas a los operadores que ahí se encuentran y la observación directa en la misma.

2.3.1.3 Técnicas de recolección de datos

Entre las técnicas para recolectar los datos están: entrevistas no estructuradas al personal que labora en el Centro de Datos UDO, que permitió recopilar información general acerca del mismo. También se utilizó la técnica de la observación directa, la cual

permite al investigador “observar y recoger datos mediante su propia observación” (Sabino, 2000); ésta técnica se aplicó para verificar e interpretar la manera como se realizan las actividades en Centro de Datos UDO.

2.3.2 Metodología del área aplicada

El desarrollo de este proyecto está basado en la metodología *Rational Unified Process* (RUP), es un proceso iterativo e incremental que permite la construcción del sistema paso a paso mediante una secuencia de iteraciones, donde cada iteración aborda una parte de la funcionalidad total del sistema. RUP es un proceso de desarrollo de software que utiliza el Lenguaje Unificado de Modelado (*Unified Modeling Languaje*, UML), que es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, porque permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender (Schmuller, 2002). La metodología RUP en el desarrollo de este proyecto contemplará las siguientes fases (Gómez, 2007):

2.2.2.1 Fase de inicio

En esta fase se concreta la visión del producto, define el ámbito y objetivos del proyecto, así como las funcionalidades y capacidades del producto.

2.2.2.2 Fase de elaboración

En esta fase tanto la funcionalidad como el dominio del problema se estudian en profundidad. Se define una arquitectura básica, a la vez que se planifica el proyecto considerando los recursos disponibles.

2.2.2.3 Fase de construcción

El producto se desarrolla mediante iteraciones en las cuales se involucran tareas de análisis, diseño e implementación. En esta fase se refina de manera incremental la arquitectura básica producto de las fases anteriores, gran parte del trabajo es

programación y pruebas donde se documenta tanto el sistema construido como el manejo del mismo.

2.2.2.4 Fase de transición

En esta fase se contempla la preparación de planes de capacitación para los usuarios, elaboraciones de manual, configuración y parametrizado de las cuentas de usuarios, migración de los datos del sistema actual al nuevo sistema y puesta en marcha del sistema.

2.4 Definición de términos.

Este glosario ayudará a entender los términos usados en este proyecto. Se divide en parte software como su palabra lo dice, es la parte blanda (programas) que se utilizan en un computador y hardware es la parte tangible (física).

2.4.1 Software

Alarma

Una alarma es aquella que se genera de manera automática por una herramienta utilizada para monitorear equipos. Un aviso de alarma es una señal por medio del cual se informa al personal autorizado para que sigan instrucciones específicas de emergencia debido a la presencia real o inminente de una amenaza (Barrientos y Beites, 2006).

Bus SPI (Serial Peripheral Interface)

Es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

Explorador web

Es un software encargado de interpretar los lenguajes de programación web que crea en la pantalla de la computadora un menú único, y proporciona una interfaz gráfica con la web. El menú enlaza al usuario con recursos de la Internet, incluyendo elementos de textos, gráficas, archivos de sonidos, entre otros (Tanenbaum, 2003).

Firmware

Es el conjunto de instrucciones de un programa informático que se encuentra registrado en una memoria ROM (Read Only Memory, siglas en español, Memoria De Solo Lectura). Estas instrucciones fijan la lógica primaria que ejerce el control de los circuitos de alguna clase de artefacto.

Hipertexto

Es una herramienta de software con escritura no secuencial, un texto que se bifurca y que permite al lector elegir su trayectoria y su discurso, principalmente sobre una pantalla de un dispositivo electrónico que lo soporte (Tanenbaum, 2003). Las características más destacadas de los hipertextos es la facilidad de acceso y consulta, el almacenamiento de un gran volumen de información, el uso de todas las tecnologías de la información, presentación del contenido de una forma más agradable, permiten una navegación personal, dinamismo e interactividad y es multiplataforma

Hipermedia

Es una ampliación de hipertexto. Hipermedia le permite buscar y manipular formas de datos en multimedia; esto es, gráficas, sonido, video (Scolari, 2008).

HTTP (HyperText Preprocessor)

También denominado protocolo de transferencia de hipertexto, es el protocolo utilizado para cada transacción o consulta en la WWW. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor (Tanenbaum, 2003).

Host

Máquina conectada a una red. Tiene un nombre que la identifica, el hostname. La máquina puede ser una computadora, un dispositivo de almacenamiento por red, una impresora, etc

Nagios

Es un sistema de monitorización de redes ampliamente utilizado, de código abierto, que monitorea los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado. Entre sus características principales la monitorización de servicios de red (HTTP, SNMP, entre otros.), la monitorización de los recursos de sistemas hardware, independencia de sistemas operativos, posibilidad de monitorización remota mediante túneles SSL cifrados o SSH, y la posibilidad de programar *plugins* (instrucciones) específicos para nuevos sistemas.

Se trata de un software que proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas por los responsables correspondientes mediante (entre otros medios) correo electrónico y mensajes SMS, cuando estos parámetros exceden de los márgenes definidos por el administrador de red. Nagios fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix, Cuando se detecta un error es capaz de notificar en diferentes modos, informando el estado del servicio que ha provocado el error, incluyendo informes de estado e históricos Web. Su arquitectura es abierta la cual permite una rápida y fácil adaptación de sus funciones según las necesidades que se tenga.

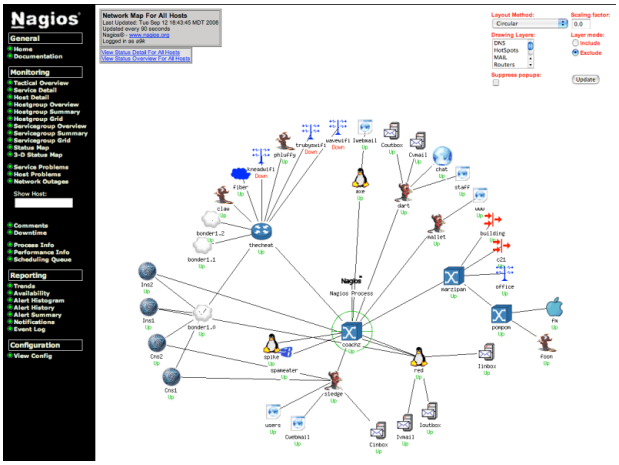


Figura 3. Ejemplo de Interfaz de Nagios
Fuente: https://www.Nagios.org/

Objeto de datos

Un objeto de datos es una representación de determinada información compuesta que el software debe entender. Un objeto de datos puede ser una entidad externa (por ejemplo, cualquier elemento que produzca o consuma información), elemento (un reporte o despliegue), suceso (llamada telefónica) o evento (alarma), rol (vendedor), una unidad organizacional (departamento de contaduría), lugar (almacén), o una estructura (archivo).

Página web

Es el resultado en hipertexto que proporciona un navegador de la WWW después de obtener la información solicitada. Su contenido puede ir desde un texto corto a un enorme conjunto de textos, gráficos estáticos o en movimiento, sonido, entre otros (Tanenbaum, 2003).

Python

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional (Knowlton, 2009).

Protocolo

Es la descripción formal de formatos de mensaje y de reglas que dos o más computadores deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina a máquina o intercambios de alto nivel entre programas de asignación de recursos (Castells, 2001). Es un conjunto de reglas que determina cómo se envían, interpretan y procesan los mensajes entre computadoras. Los protocolos permiten que las computadoras interactúen en una red y trabajen a diferentes niveles del modelo OSI.

Protocolo TCP/IP

Los protocolos TCP e IP, en realidad son dos protocolos distintos, son los encargados de organizar físicamente el tráfico por dentro de la red (Postel, 2012).

Cada uno de ellos desempeña una actividad diferente:

Protocolo IP (*Internet Protocol*):

Es el primero en ponerse a trabajar. En cuanto se interactúa con Internet de alguna forma, el protocolo IP se activa, su labor consiste en dividir en lo que se llama paquetes IP toda la información que hay que remitir.

Protocolo TCP (*Transfer Control Protocol*):

Cuando los paquetes ya se han creado, se pone en marcha este segundo protocolo, cuya labor es la de transmitir los paquetes desde el ordenador emisor hasta su destino.

Red

Es un conjunto de equipos informáticos y software conectados entre sí por medio de dispositivos físicos, con la finalidad de compartir información, recursos y ofrecer servicios. La finalidad principal para la creación de una red de computadoras es compartir los recursos y la información en la distancia, asegurar la confiabilidad y la disponibilidad de la información (Groth-Skandier, 2005).

SSH (*Secure SHell*, siglas en español, Intérprete de Órdenes Seguro)

Es el nombre de un protocolo y del programa que lo implementa, sirve para acceder a otras máquinas de forma segura mediante un sistema de cifrado de datos (López, 2010).

SSL (*Secure Sockets Layer*, siglas en español, Capa de Puertos Seguros)

Son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet (López, 2010).

SMS (*Short Message Service*, siglas en español, Servicio de Mensaje Cortos)

Mensaje corto de texto que se puede enviar entre teléfonos celulares o móviles. (RAE,2001).

SNMP (*Simple Network Management Protocol*, siglas en español, Protocolo Simple de Administración de Red).

Es un protocolo que les permite a los administradores de red administrar dispositivos de red y diagnosticar problemas en la red (Barban, 1999).

Scripts

Es un archivo de órdenes, archivo de procesamiento por lotes o guion es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con la página web o con el usuario (Scolari, 2008).

Socket

Es un mecanismo que permite la conexión entre distintos procesos, habitualmente se utilizan para establecer comunicaciones entre distintas máquinas que estén conectadas a través de la red. Cuando utilizamos Sockets para comunicar procesos nos basamos en la arquitectura cliente y servidor.

2.4.2 Hardware.

MicroSD

Fue desarrollada por SanDisk, y en julio de 2005 fue adoptada por la Asociación de Tarjetas SD con el nombre microSD

Tarjeta Arduino

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa de circuito impreso que contiene un microcontroladores de la marca "ATMEL" que cuenta con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación *processing*. El dispositivo conecta el mundo físico con el mundo virtual, o el analógico con el digital controlando, sensores, alarmas, sistemas de luces, motores, sistemas de comunicaciones y actuadores físicos (Banzi, 2012).

Lenguaje de programación Arduino

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel *Processing*. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, debido a que usa la transmisión serial de datos soportada por la mayoría de los lenguajes. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida. Los principales lenguajes de programación compatibles con Arduino son: C C++ (mediante libSerial o en Windows) C# Java Matlab Python Visual Basic .NET

Entorno Arduino

El entorno de desarrollo de Arduino contiene un editor de texto para escribir los códigos, un área de mensajes, una consola de texto para el puerto serial, una barra de tareas con botones para las funciones más comunes y una serie de menús para interactuar con el usuario. Este se conecta al hardware del Arduino para comunicarse y cargar programas.

Figura 4. Ejemplo del Entorno Arduino

Instrumento de medición

Es un dispositivo empleado con el propósito de comparar magnitudes físicas distintas a través de un procedimiento de medición. Para esto los componentes deben estar interconectados de manera que mantengan una relación funcional. (Dulhoste, 2010),

Elementos funcionales de un instrumento de medición

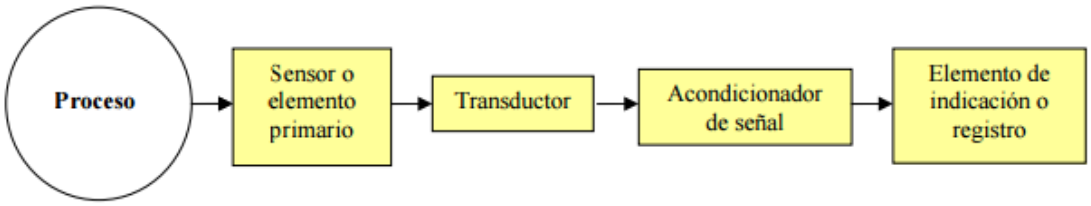


Figura 5. Elementos de un instrumento de medición.
Fuente: Dulhoste, 2010

Sensor.

Según la RAE (2001) define un sensor como un dispositivo que capta magnitudes físicas (variaciones de luz, temperatura, sonido, etc.) u otras alteraciones de su entorno

El transductor.

Es capaz de traducir el valor de una magnitud física o química, en una señal eléctrica, ya sea en forma analógica o digital (Balcelis y Romeral, 1997).

Acondicionador de señal

La señal de salida del sensor de un sistema de medición en general se debe procesar de una forma adecuada para la siguiente etapa de la operación. La señal puede ser, por ejemplo, demasiado pequeña, y sería necesario amplificarla; podría contener interferencias que eliminar; ser analógica y requerir su digitalización; ser digital y convertirla en analógica; ser un cambio en el valor de la resistencia, y convertirla a un cambio en corriente; consistir en un cambio de voltaje y convertirla en un cambio de corriente de magnitud adecuada, etcétera. A todas estas modificaciones se les designa en general con el término acondicionamiento de señal. Los componentes comunes de un acondicionador de señal son: Amplificador, filtro, convertidor Analógico/Digital.

Elemento de indicación o registro

Es el elemento que permite la lectura o registro de la variable medida. Este puede ser una aguja en una escala graduada, una pantalla digital, una pluma en un registrador, una computadora etc.

Clasificación de sensores.

Los sensores se clasifican dependiendo de la señal o magnitud de entrada al dispositivo, la señal de salida, y la naturaleza de la señal generada (Ortega, 2014).

Según Mendoza (2012), define señal como una cantidad física que varıa con el tiempo, el espacio o cualquier otra variable independiente que transmite información acerca del comportamiento o la naturaleza de algún fenómeno

Según el tipo de señal de entrada:

Mecánica: Son aquellos sensores que miden una propiedad extrínseca de la materia, o la interacción entre dos fuerzas de distinto tipo. Ejemplo la longitud, masa, velocidad, aceleración, fuerza, torque, presión, intensidad acústica, longitud de onda.

Química: Son los sensores que pueden identificar las propiedades intrínsecas de la materia. Ejemplo concentración, composición potencial de oxidación/reducción, pH

Eléctrica: Son los sensores que son capaces de captar la interacción de los electrones en una corriente. Ejemplo está el voltaje, corriente eléctrica, constante dieléctrica, polarización, campo eléctrico.

Magnética: Son los sensores que miden cambios en la intensidad de un campo magnético, como también su densidad y permeabilidad.

Térmica: En este grupo están los sensores que detectan la variación de la temperatura en el ambiente o en un determinado material, así como también el flujo de calor que se puede transmitir de un cuerpo a otro.

Según el tipo de señal de salida

Sensores Analógicos: Componen la mayor parte de sensores disponibles en el mercado, y son aquellos que entregan su señal continua en el tiempo. Ejemplo los sensores generadores de señal. Entregan una salida de nivel variable en función del parámetro que midan, por ejemplo, un sensor de temperatura de -20º a +50º con salida 0-10V.

Sensores Digitales: Son aquellos sensores que poseen una salida de carácter discreto. Ejemplo los sensores de posición, sensores codificadores incrementales, sensores auto resonantes, entre otros. Dan la información relativa a la medida con un protocolo de comunicaciones específico que el fabricante facilita: por ejemplo el sensor de temperatura y humedad.

Según la naturaleza de la señal generada

Sensores pasivos: Son aquellos que utilizan una señal externa o auxiliar para poder realizar la medición de la magnitud. No precisan de alimentación: Resistencias que cambian de valor según luz o temperatura. Ejemplos de estos sensores son los de luz, de temperatura, entre otros

Sensores activos: Se los conoce también como sensores generadores de señal y son aquellos que no requieren de una señal externa para poder realizar la medición de una magnitud, ya que son capaces de emitir una señal propia. Tienen circuitos electrónicos que alimentar y necesitan una fuente de energía

CAPÍTULO III. DESARROLLO

El proceso unificado conocido como RUP, es un modelo de software que permite mediante un proceso continuo de pruebas y retroalimentación, garantizar el cumplimiento de ciertos estándares de calidad. Su objetivo es asegurar un producto de alta calidad que satisfaga la necesidad del usuario final dentro de un tiempo y presupuesto previsible.

Planificación del proyecto.

A través de la planificación del proyecto se lograron establecer los objetivos y las delimitaciones del proyecto. Igualmente se determinó la necesidad principal de la Coordinación de Teleinformática UDO.

3.1 Descripción del proyecto

3.1.1 Planificación del tiempo

Durante esta etapa se determinaron las actividades y el número de iteraciones para el desarrollo del proyecto. Para cada una de ellas se establecieron las actividades y los tiempos de ejecución. Luego se elaboró el cronograma de actividades para cada iteración, apéndice A.

3.1.2 Gestión de riesgos

Según Salvador y cols. (2003) define los riesgos como eventos o condiciones inciertas que, si se producen, tienen un efecto positivo o negativo sobre al menos un objetivo del proyecto, como tiempo, coste, alcance o incluso la calidad.

Partiendo de esta definición la gestión de riesgo es el proceso sistémico de identificación, análisis y respuesta a los riesgos de los proyectos, este consiste entonces en aumentar la probabilidad de impacto de los eventos positivos y disminuir la probabilidad e impacto de los eventos adversos al proyecto.

Todo desarrollo tiende a presentar una serie de riesgos que de no ser solucionados en el momento adecuado, trae como consecuencia problemas posteriores que con el transcurrir del tiempo son más difíciles de solucionar, razón por la cual es necesario tomar en cuenta dichos riesgos, desde los inicios del desarrollo.

Mediante la gestión de riesgos se lograron reconocer, analizar y estimar el impacto de todos aquellos riesgos que pudieran influir negativamente en el normal desenvolvimiento del proyecto.

Para el desarrollo del sistema de monitoreo ambiental basado en Arduino y Nagios para el Centro de Datos UDO, se determinó la siguiente lista de riesgos:

Tabla 1. Riesgos identificados.

ID	Riesgo
R1	No contar con el equipo necesario
R2	La falta de conocimientos de las nuevas herramientas de desarrollo hace que el proyecto tome más tiempo del esperado para su culminación
R3	Alcanzar el ámbito del producto requiere más tiempo del esperado.
R4	Añadir requisitos extra.
R5	Desconocimiento de las herramientas a utilizar.
R6	Pérdida accidental de la información referente al desarrollo.

Para los riesgos que resultaron predominantes en el desarrollo del sistema de monitoreo ambiental, se planteó un plan de prevención y contingencia para los más predominantes durante el desarrollo del proyecto, los cuales se encuentran enumerados en la Tabla 2.

Tabla 2. Plan de prevención y contingencia para los riesgos más predominantes

ID	Probabilidad	Impacto	Plan de Prevención	Plan de contingencia
R1	40%	Critico	Tener el equipo necesario antes de realizar el sistema.	Probar el funcionamiento de los equipos a utilizar
R2	30%	Marginal	Elegir herramientas que no sean difíciles de aplicar	Llevar a cabo una serie de talleres de adiestramiento en las herramientas de desarrollo
R3	25%	Marginal	Efectuar un estudio detallado del proyecto para contemplar todas las funcionalidades que debe tener el sistema	Redefinir el ámbito del proyecto incorporando todas las funcionalidades y reorganizando el cronograma de actividades
R4	40%	Marginal	Identificar todos y cada uno de los requerimientos.	Ajustar el cronograma de actividades
R5	50%	Marginal	Buscar orientación de expertos que usen las herramientas.	Consultar información de las herramientas a utilizar.
R6	40%	Critico	Establecer un esquema de seguridad y control de versiones para la información valiosa del proyecto	Buscar respaldos anteriores y analizar la pérdida y a partir de allí replanificar el tiempo e invertir más horas para lograr recuperar la información perdida.

3.2 FASE I: Inicio.

En esta fase se realizó un análisis objetivo donde se comprendió lo que debía abarcar el sistema a desarrollar, en consecuencia se creó una arquitectura del mismo, que cumplió con los requerimientos de la institución; estos requerimientos fueron expresados en un modelo inicial de casos de uso. Durante la fase de inicio las iteraciones ponen mayor énfasis en actividades del modelado del negocio y de requerimientos.

En esta fase inicial se realizó una iteración, con una duración de 30 días, conformada por requisitos y análisis. A continuación se procede a describir los elementos que fueron utilizados en esta fase.

3.2.1 Contexto del Sistema

Se llevó a cabo un estudio de las actividades más importantes realizadas en la Coordinación de Teleinformática UDO con el fin de comprender los procesos de mayor relevancia y ofrecer una visión representativa del contexto del sistema. Para entender mejor el contexto se elaboró un plano del Centro de Datos UDO.

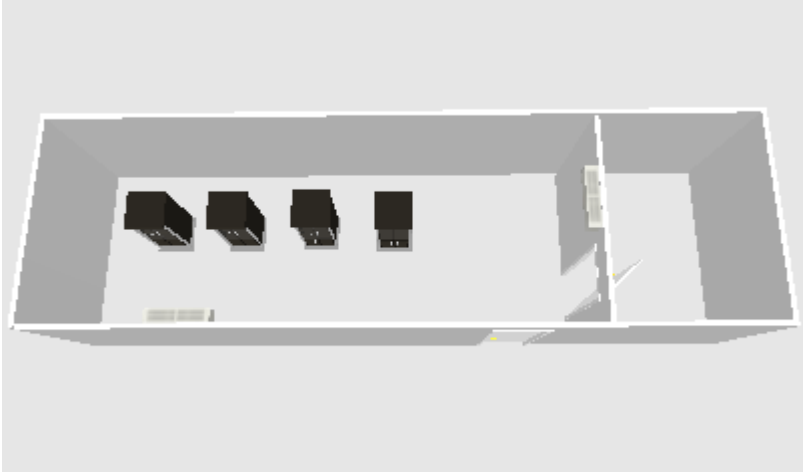


Figura 6. Vista aérea del Centro de Datos UDO

3.2.2 Modelado del negocio

Según la Fundación Premio Nacional de Tecnología (2011), el modelado de negocio describe la lógica con la cual una organización crea, entrega y captura valor. Definiendo los roles de sus actores, las interacciones entre las distintas partes del sistema y las reglas en las que se basan las actividades de la empresa.

Para realizar el modelado del negocio de la Coordinación de Teleinformática UDO se hizo un estudio de su misión, estructura y funciones del mismo con la finalidad de capturar los procesos al más alto nivel y asegurar que todos los involucrados tengan una visión común del proyecto a desarrollar. Todos estos aspectos dieron como resultado el modelado de objetivos, procesos, actores, roles y reglas. Se presenta a continuación:

3.2.2.1 Misión de la Coordinación de Teleinformática UDO

Es la de mantener el funcionamiento óptimo de la plataforma integral de telecomunicaciones de la UDO, que permita enlazar los 5 núcleos principales en los estados (Bolívar, Anzoátegui, Sucre, Nueva Esparta y Monagas) y sus extensiones, en el área académica y administrativa, así como la prestación de servicios informáticos a la comunidad.

La Coordinación de Teleinformática UDO, cumple con varios objetivos dentro de la institución, el principal es el de dirigir la red Académica - Administrativa, asimismo servirá para proteger y garantizar el uso seguro de las redes y de la información que por ella circule, al fin de evitar la pérdida o alteración de dicha información por personas ajenas a la institución.

Dentro de sus otros objetivos se encuentra el desarrollo de la plataforma de comunicaciones en la UDO, la aplicación de técnicas de telecomunicación e informática, a la transmisión de información computarizada e impulsar el desarrollo de un servicio de

alta calidad en la búsqueda de información vía Intranet e Internet en la institución y comunidad en general

3.2.2.2 Estructura

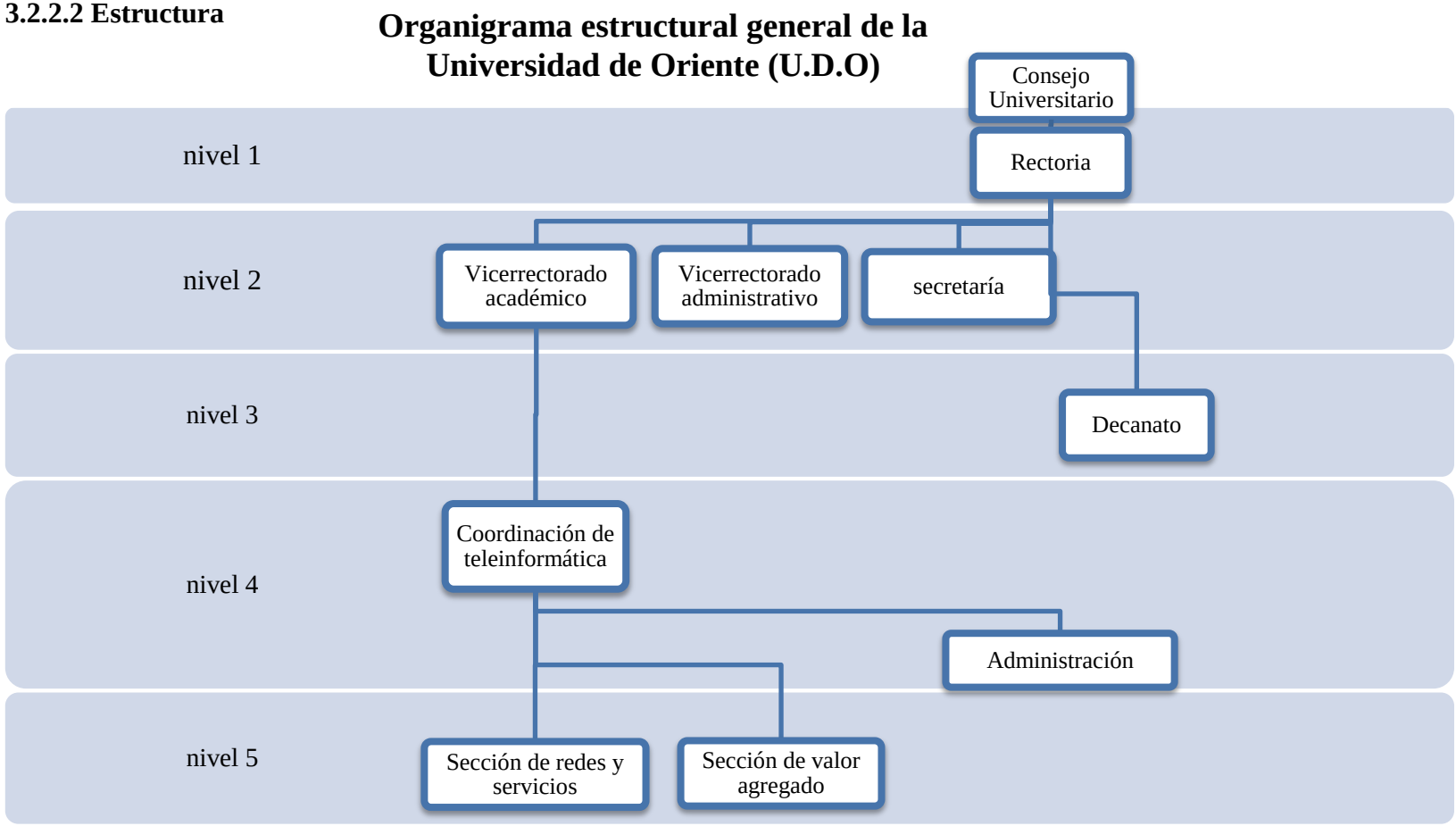


Figura 7.Organigrama UDO.
Fuente: www.udo.edu.ve

3.2.2.3 Funciones

La coordinación de teleinformática cumple con varias funciones la primera es la de proveer los servicios (Correo Electrónico, Web y otros) de Internet a todos los funcionarios y estudiantes de la UDO, bajo las restricciones que sean aplicables.

También administra y mantiene en funcionamiento los sistemas y dispositivos electrónicos, localizados en el Centro de Datos UDO

Desarrolla labores de monitoreo y ejecución de procedimientos en situaciones normales y de emergencia que se presenten en el Centro de Datos UDO

3.2.2.3.1 Funciones de la Sección de Redes y Servicios

Monitorear el comportamiento de la red de datos, así como darle el óptimo funcionamiento y mantenimiento necesario.

Otra de las funciones que tiene esta sección es la de realizar los mantenimientos preventivos y/o correctivos a las instalaciones de telecomunicaciones.

También se encarga de recibir y resolver las fallas detectadas en los servicios proporcionados; administra los servidores de red perteneciente a la Red Académica y Administrativa de la Universidad de Oriente (RAUDO), estableciendo mecanismos de respaldo de información

Establece planes de mantenimiento de los equipos de comunicaciones de voz, dato y video existentes en la institución.

Luego de recolectar información concerniente a la coordinación de teleinformática y sus actividades, se generó un análisis en el cual se pueden apreciar las actividades básicas del negocio.

El proceso manual se genera de la siguiente manera: el operador es el encargado de observar cualquier situación que suceda en todos los equipos del Centro de Datos UDO, está observación se hace a través del sistema Nagios que es el delegado de supervisar los servicios (servidores, redes). Estos servicios al presentar fallas generan alarmas, estas fallas son verificadas por el operador y monitor, quienes son los encargado de tomar las acciones pertinentes para solucionar los problemas.

En cuanto al monitoreo de las variables ambientales temperatura y humedad, solo comprueban si el clima esta frio y si están en funcionamiento los aires acondicionados en el Centro de Datos UDO.

Así concluye el proceso manual que conlleva a retardos en los tiempos de respuesta del personal a la hora de algún inconveniente técnico.

3.2.2.4 Modelado de casos de uso del negocio.

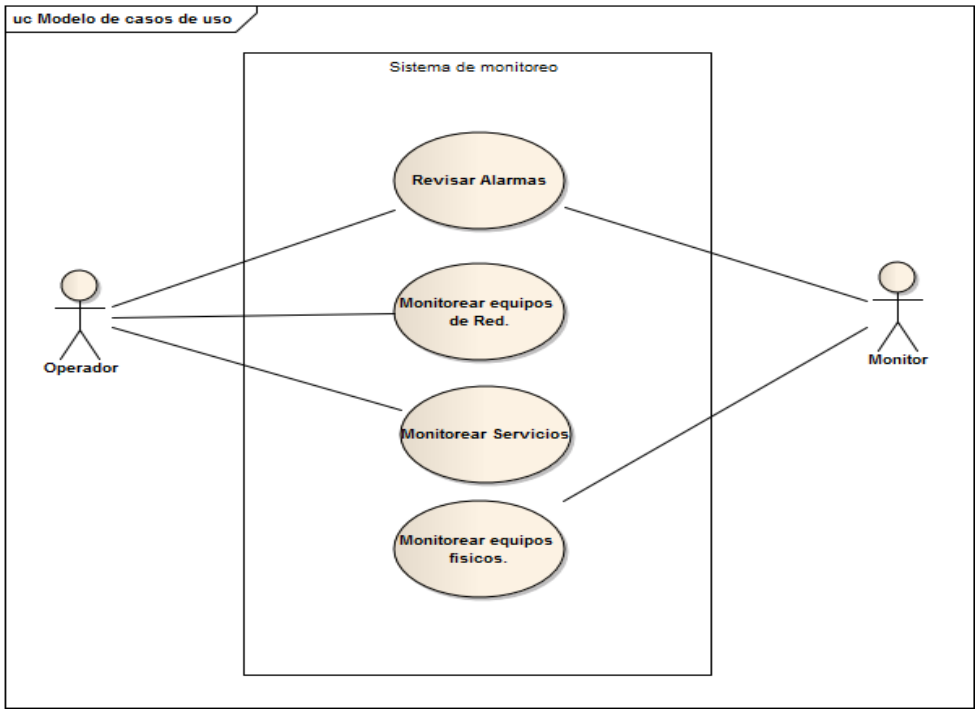


Figura 8. Modelado de caso de uso del negocio.

3.2.2.5 Modelado de objetivos

Una vez realizada la descripción del sistema de negocio, se elaboró el modelo de objetivos del negocio, que es representado en la figura 9 definiendo así la misión, visión y objetivos que contribuyen a alcanzar las metas establecidas por la Coordinación de Teleinformática UDO.

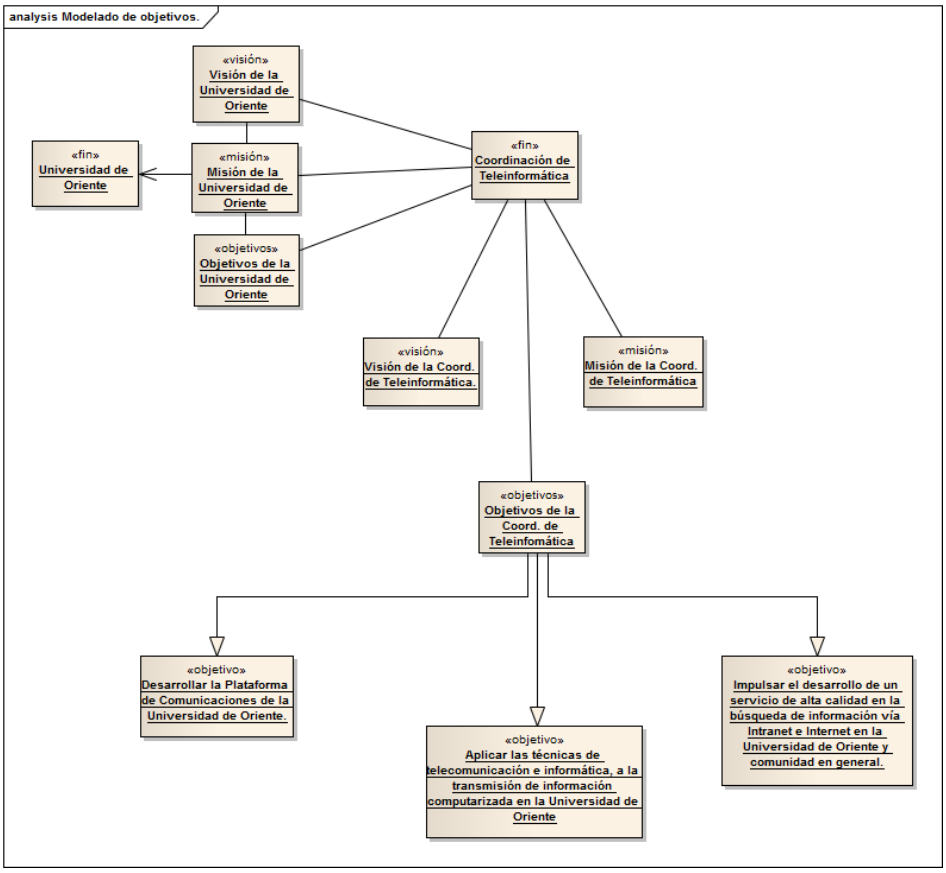


Figura 9. Modelo de objetivos de la Coordinación de Teleinformática UDO.

3.2.2.6 Modelo de Análisis del Negocio

El modelo de procesos del negocio se comenzó con la elaboración de la cadena de valor figura 10. La cadena de valor muestra aquellos procesos que son la razón de ser del sistema de negocio estudiado, clasificados en procesos fundamentales (PF) y procesos de apoyo (PA).

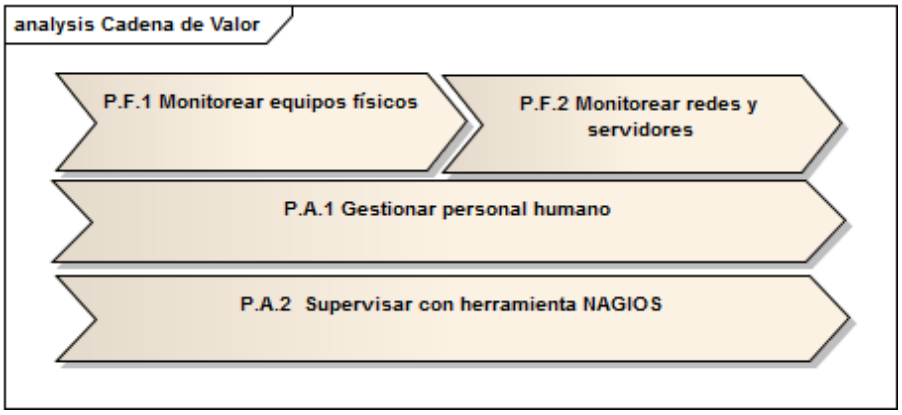


Figura 10. Cadena de valor de los procesos de la Coordinación de Teleinformática UDO.

Los procesos fundamentales de la cadena de valor se descomponen en subprocesos. A partir de los procesos fundamentales anteriormente descritos se obtuvieron los subprocesos que se muestran en la Figura 11.

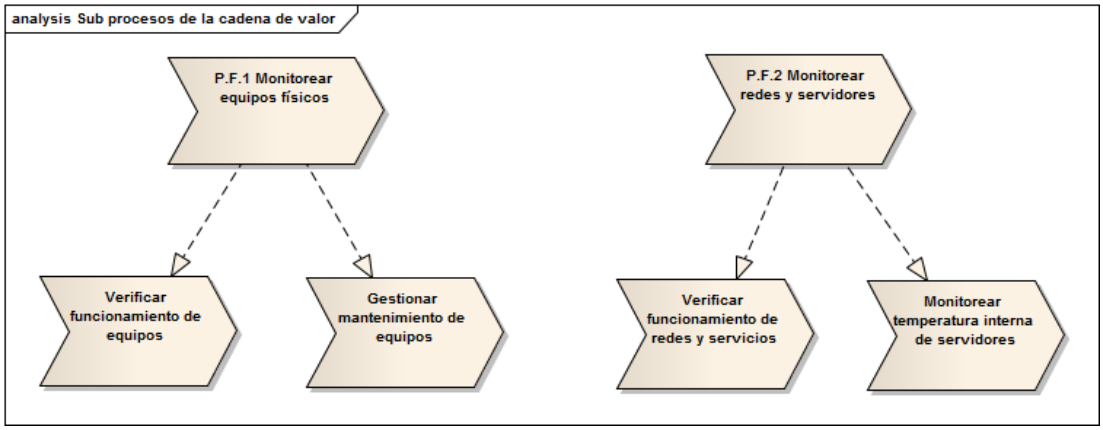


Figura 11. Sub-procesos de la cadena de valor del sistema de negocio estudiado.

En la figura 12 y figura 13, se muestran los diagramas de sub-procesos. En el Apéndice B, se encuentran los diagramas de los subprocesos restantes.

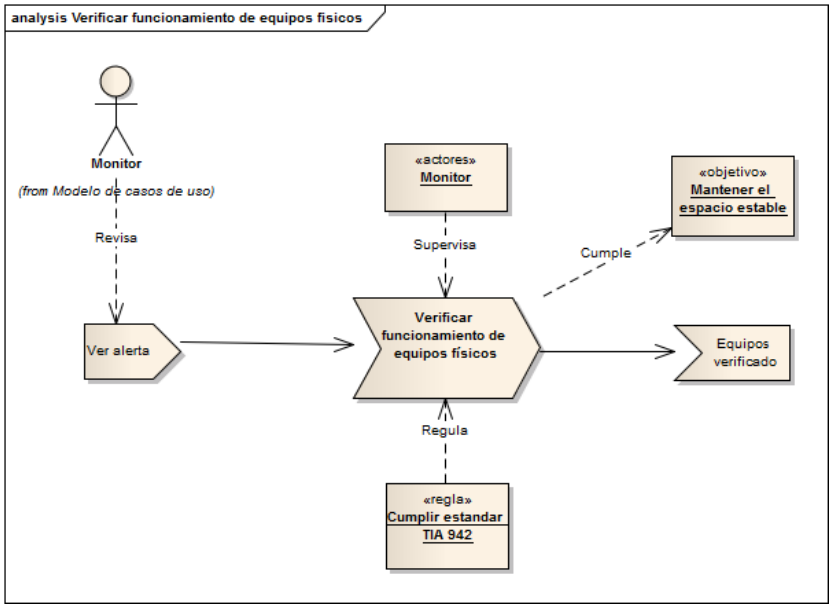


Figura 12. Diagrama de proceso del subproceso Verificar funcionamiento de equipos físicos

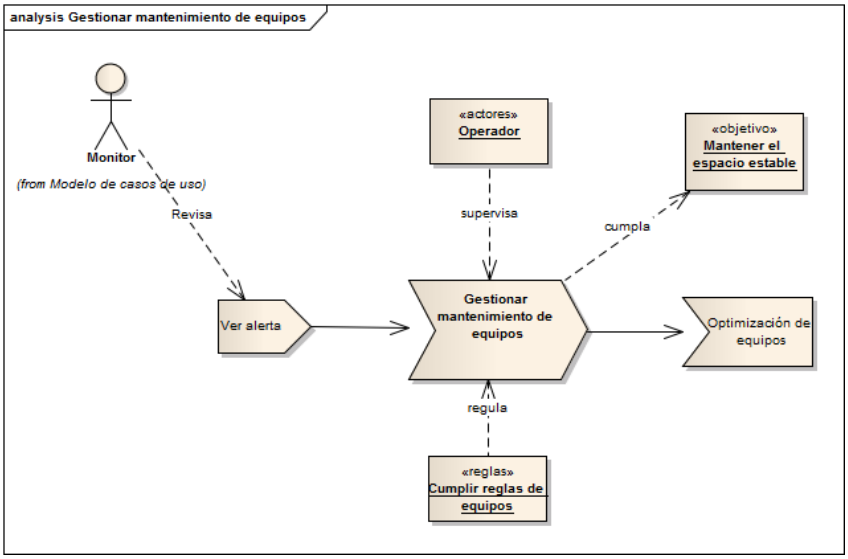


Figura 13. Diagrama de proceso del subproceso gestionar mantenimiento de equipos

3.2.2.7 Modelado de actores

Los actores son personas o sistema externo que interactúan con el sistema de monitoreo. Pueden proveer funcionalidades al sistema, así como utilizar las ya provistas por éste, por tanto pueden ingresar o capturar datos al sistema.

Cada actor tiene asignado un rol dentro de la aplicación, lo que determina las acciones que este puede realizar en el sistema

Operador: Personal encargado de gestionar toda la información referente a las alarmas, los eventos, servicios y generar reportes. Cuentan con claves de acceso y nombre de usuario

Monitor: Es el encargado de verificar las alarmas y los equipos físicos que conforman el Centro de Datos UDO.

3.2.3 Requisitos

Los requerimientos son características con las cuales el sistema debe contar o restricciones que se deben satisfacer para ser aceptadas por los clientes (Quiroga, 2008).

La captura de requisitos fue realizada mediante entrevistas al personal de la gerencia encargada del departamento de teleinformática. A continuación en la se presentan dichos requisitos.

Tabla 3. Lista de requisitos.

ID	Requisito
1	Establecer clave de acceso al personal asociado, para mantener la seguridad del mismo.
2	Implementar una comunicación vía Ethernet para lograr un acceso remoto al sistema.
3	Realizar una interfaz visual, fácil y cómoda para despliegue de los datos en tiempo real.
4	Programación del *plugin* que cumpla la función de comunicación entre Arduino y Nagios
5	Visualizar reportes.
6	Configurar los *hosts* y servicios adecuados para monitorización de la red.
7	Enviar notificaciones al operador encargado
8	Comprobar el funcionamiento correcto del sistema de monitorización

3.2.3.1 Análisis de requisitos

El análisis consistió en clasificar y agrupar los requisitos recolectados en funcionales y no funcionales.

Según Falgueras (2003), los requisitos funcionales son aquellos que describen lo que debe realizar el software para los usuarios. Estos requisitos quedan recogidos en los casos de uso.

Los requisitos no funcionales no van asociados a casos de usos concretos y consiste en restricciones impuestas por el entorno, y la tecnología. Algunos de ellos son sobre interfaz, facilidad de mantenimiento, entre otros. La Tabla 4 muestra los requisitos recolectados anteriormente debidamente clasificados.

Tabla 4. Lista de requisitos clasificados

ID	Requisito	Tipo de requisito
1	Establecer claves de acceso al personal asociado, para mantener la seguridad del mismo.	Funcional
2	Implementar una comunicación vía Ethernet para lograr un acceso remoto al sistema.	Funcional
3	Visualizar reportes	Funcional
4	Programación del *plugin* para la comunicación entre Arduino y Nagios	Funcional
5	Realizar una interfaz visual, fácil y cómoda para despliegue de los datos en tiempo real	No Funcional
6	Configurar los *hosts* y servicios adecuados para monitorización de la red	No Funcional
7	Enviar notificaciones al operador encargado	No Funcional
8	Comprobar el funcionamiento correcto del sistema de monitorización	No Funcional

3.2.4 Análisis y diseño

En esta disciplina se procesó la información recolectada permitiendo describir el sistema con mayor profundidad, y se determinó el alcance del sistema

Durante las siguientes fases se refinan los requerimientos, diagramas y arquitectura hasta obtener una estructura desarrollada y resistente que ofrece una visión clara de las actividades a realizar antes de iniciar con la fase de construcción

3.2.5 Prueba

Las pruebas ayudan a los desarrolladores a determinar si el sistema satisfacen todos los requerimientos solicitados por los clientes, del mismo modo, ayuda a detectar errores. A continuación en la Tabla 5 se muestran las pruebas planificadas para el sistema.

Tabla 5. Plan de pruebas del sistema

Prueba	**Descripción**
Pruebas de adquisición de datos	Verificar la correcta adquisición de datos desde un computador externo. Dicha prueba se realiza tomando la temperatura externa del Centro de Datos UDO y visualizando por pantalla la correcta adquisición.
Pruebas de interfaz	Valida si las interfaces son aceptables para los usuarios. Para determinar esto, dichas pruebas se realizan a través de entrevistas a los expertos en el sistema.

3.3 FASE II: Elaboración.

El objetivo principal de esta fase es alcanzar la línea base de la arquitectura, recopilando la mayoría de los requisitos que aún quedan pendientes. En esta fase se transforman y refinan los modelos de la fase de inicio en otra serie de modelos que vayan perfilando una solución más cercana al mundo real.

Al final de esta fase se tiene la recopilación de todos los requisitos funcionales y la elaboración de un modelo de análisis y de diseño, lo suficientemente sólidos como para construir la arquitectura base del sistema.

3.3.1 Hardware

Se describe el proceso de diseño del hardware para el sistema de monitorización

3.3.1.1 Modelado de negocio.

Se realizó el diagrama del sistema de monitoreo (hardware) figura 14. Este sistema consiste de una placa Arduino que se encarga de:

Establecer la conexión con el protocolo Ethernet

Comunicación con el *plugin* de Nagios

Lectura de la página web contenida en la memoria micro SD

Lectura del sensor de temperatura

A Continuación se procederá a explicar en detalle los elementos más importantes del sistema de monitoreo (Hardware)

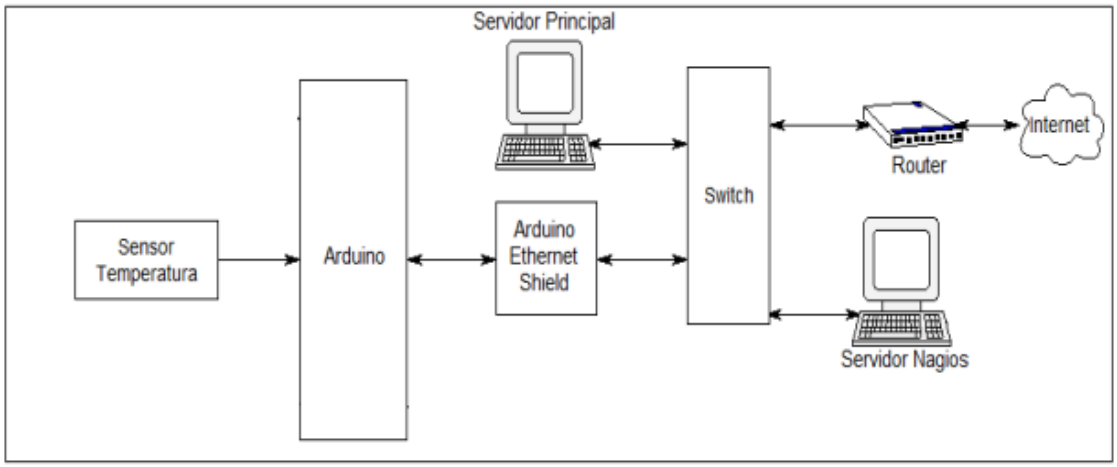


Figura 14. Diagrama general del sistema de monitoreo (hardware)

Sensor de temperatura: Dadas las dimensiones de la habitación, es necesario un estudio de los cambios de temperatura dentro de dicho cuarto, para asegurarse que existe una uniformidad de las lecturas. Para esto se hizo un muestreo de la temperatura en diferentes lugares del Centro de Datos UDO durante un tiempo de 30 días, para así definir la cantidad de sensores a emplear, y su respectiva ubicación. Se investigó también sobre los diferentes métodos de medición de temperatura y humedad para decidir cuál tipo de dispositivos se apegaba más a las necesidades de la institución y los requerimientos del proyecto. Una vez hecho este estudio se adquirieron los sensores de temperatura y humedad. Con estos dispositivos seleccionados se diseñó un circuito que permitió minimizar las alteraciones que puede producir el ruido en el valor de temperatura y humedad que se obtenga. Este muestreo de temperatura externa y temperaturas internas (servidores), que se encuentran en el Centro de Datos UDO se puede observar en el Apéndice D

Sistema transductor y de envío de datos (Arduino y Arduino Ethernet shield): para el análisis de las variables de temperatura y humedad, es indispensable convertirlos a código digital, se empleó una placa Arduino que aparte de gobernar la conversión analógica-digital, tuvo a cargo también la captura de los valores provenientes del sensor, así como la eventual transmisión al computador.

Las mediciones proporcionadas por el sensor pasarán por una etapa de acondicionamiento de señal, para luego ir al computador.

Instrumentación

Es el proceso de sensar y procesar la información proveniente de variables físicas (temperatura, humedad, entre otros), que a partir de aquí se emplea para comenzar el monitoreo, empleando particularmente para este caso dispositivos electrónicas. Para su funcionamiento la instrumentación se divide en tres etapas:

Los sensores: Son aquellos que transforman la magnitud de una variable física que se desea medir a una señal eléctrica.

Acondicionamiento: Como su nombre lo indica, acondiciona los niveles de la señal de salida de un sensor. Esto se debe a que en la mayoría de los casos la señal de los sensores suele no ser adecuada para su procesamiento. En general los sensores entregan señales muy pequeñas y vulnerables a ruido, por lo que el acondicionamiento generalmente consiste de dos sub etapas: la etapa de amplificación y la etapa de filtrado. Dichas sub etapas se realizan a través de un amplificador operacional de instrumentación.

Digitalización: La digitalización proporciona un código digital (binario) equivalente a la señal de entrada proveniente de la etapa de acondicionamiento. Dicho código permite el procesamiento de las señales a través de sistemas basados en procesadores o computadoras

3.3.1.2 Requerimientos

Selección del hardware para el sistema de monitoreo.

A la hora de escoger la plataforma Arduino que se empleó en el diseño concreto, se tuvo en cuenta multitud de factores, como la documentación, herramientas de desarrollo

disponibles, precio y las características principales del Arduino (tipo de memoria de programa, número de temporizadores, interrupciones, entre otros).

A continuación se detallan algunas de las características que se tomaron en cuenta para el diseño de este sistema.

Bajo costo

Conexión de red Ethernet

Puertos de entrada y salida digitales

Manejo de interrupciones

Para cumplir con las exigencias de usuario, existe una amplia gama de dispositivos que permiten la implementación de este proyecto, sin embargo unos cumplen con mayores ventajas. A continuación se muestra la tabla 6 las ventajas y desventajas de Arduino + Shield Ethernet.

Tabla 6. Ventajas y desventajas de Arduino + Shield Ethernet

	Ventajas	**Desventajas**
Arduino + Shield Ethernet	Herramienta de desarrollo para Linux, lenguaje programación se basa en C/C++. Librerías para manejo del Ethernet. Disponible en el país a un precio accesible Módulo Ethernet cuenta con puerto de conexión para tarjeta microSD	No se presentan mayores desventajas con respecto a este dispositivo

Descripción del hardware seleccionado.

Arduino y módulo Ethernet

Arduino es una plataforma libre, esta placa de desarrollo se basa en un microprocesador ATmega328, el cual posee 14 pines de entrada/salida digitales, 6 entradas analógicas. Se caracteriza por su simplicidad de conexión y la posibilidad de ser programado directamente desde cualquier computador a través de un cable USB. El IDE (*Integrated Development Environment*, en español, Entorno de Desarrollo Integrado) de código abierto se puede descargar de forma gratuita (en la actualidad para Mac OS X, Windows y Linux). En la figura 15 se muestra una imagen de la plataforma física del Arduino.

Figura 15. Arduino Uno, vista superior e inferior
Fuente: www.arduino.cc

Una de las principales ventajas del Arduino es que admite agregar módulos (*shields*) en cascada para diferentes aplicaciones específicas. Para el diseño del sistema de monitoreo el requisito principal es la conexión a la red LAN (*Local Area Network*, en español, Red de área local), por lo cual se utiliza el Arduino Ethernet Shield, este dispositivo se puede observar en la figura 16.

Figura 16. Arduino Ethernet Shield
Fuente: www.arduino.cc

El Arduino Ethernet Shield permite a una placa Arduino conectarse a internet utilizando un cable de red RJ45. Se basa en el chip Wiznet W5100 Ethernet. El Wiznet W5100 provee un stack de red IP (*Internet Protocol*, en español, protocolo de internet) y admite protocolos TCP (*Transmission Control Protocol*, en español, Protocolo de Control de Transmisión) y UDP (*User Datagram Protocol*, en español, Protocolo de Datagrama de Usuario). Soporta hasta cuatro conexiones de socket simultáneas. Utilizando la librería de Ethernet en el entorno de programación.

El Arduino Ethernet Shield cuenta con una ranura micro-SD, que puede ser usado para almacenar archivos que luego se puedan compartir a través de la red. El lector de tarjetas es accesible a través del protocolo SPI (Serial Peripheral Interface, en español, protocolo de comunicación serial).

Las características se determinan en la tabla 7.

Tabla 7. Características del Arduino Ethernet Shield

Microprocesador	ATmega328
Voltaje de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12 V
Voltaje enchufe de entrada (limites)	6-20 V
Voltaje de entrada	36-57 V
Digital pines E/S digitales	14
SRAM (Static Random Access Memory, en español, memoria estática de acceso aleatorio)	2 KB (ATmega328)
EEPROM (Electrically Erasable Programmable Read-Only Memory, en español, ROM programable y borrable eléctricamente)	1KB (ATmega328)
Velocidad del reloj	
Ethernet Controller Embedded W5100 TCP/IP	16 MHz

Se eligió utilizar un sensor DHT11, puesto que es un dispositivo que permite una fácil manipulación en la maqueta, además de poseer una respuesta de 10s, capacidad anti-interferencia y ventajas de costo.

Sensor DHT11

Este es un sensor digital, tiene tres pines los cuales son: Tierra, Vcc y la señal de datos. En la figura 17 se muestra el sensor de temperatura.

Figura 17. Sensor de temperatura DHT11
Fuente: www.prometec.net

Este sensor se alimenta de una tensión en un rango entre 3,5V y 5,5V. Puede medir temperaturas entre 0ºC y 80ºC con una precisión de ±2ºC, y una humedad relativa entre el 20% y el 95% con una precisión del 5%. El valor típico del tiempo de respuesta es de 10s. El sensor se comunica con el Arduino, que se implementa en el firmware del mismo.

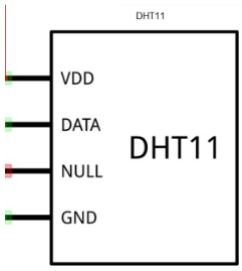


Figura 18. Distribución de terminales del Sensor DHT11
Fuente: www.prometec.net

Tabla 8. Descripción de los pines del sensor DHT11

Pin	Nombre	Descripción
1	VDD	suministro de energía 3 - 5.5 V
2	DATA	salida de datos en serie
3	NC	No conectado
4	GND	Tierra

3.3.1.3. Análisis y diseño

Diseño del sistema de monitoreo

El Arduino se comunica con el Ethernet Shield por medio del protocolo SPI, este protocolo es un estándar de comunicaciones que permite el intercambio de datos entre dispositivos (Maestro-Esclavo) de manera síncrona y bidireccional, a una alta velocidad de transmisión. Para seleccionar a cada uno de los esclavos existe una línea, denominada "Slave Select", en este caso el sistema está compuesto de un maestro (Arduino) y un esclavos (Ethernet Shield) Las señales del protocolo SPI son las siguientes:

• SCLK: Señal de reloj, impuesta y generada por el dispositivo maestro, encargado de sincronizar la comunicación.

• MOSI: (Master Output – Slave Input) Maestro envía los datos al dispositivo esclavo.

• MISO: (Master Input – Slave Output) Esclavos envían datos al dispositivo maestro.

• SS: (Slave Select) Línea que el maestro lo activará para indicar al esclavo con el que se va a establecer la comunicación.

El sensor de temperatura requiere un solo cable de conexión con el Arduino, sin embargo, la comunicación entre ellos es por medio del protocolo 1-wire, el cual es un protocolo de comunicaciones en serie diseñado por Dallas Semiconductor. Está basado en un bus, un maestro y varios esclavos de una sola línea de datos en la que se alimentan. Por supuesto, necesita una referencia a tierra común a todos los dispositivos.

Ubicación de los sensores en el Centro de Datos UDO.

Se pueden utilizar diversos tipos de sensores para obtener una advertencia anticipada de distintos problemas. Aunque el número y el tipo específico de sensores puede variar en función del presupuesto y el riesgo de la amenaza. En este trabajo se utilizó específicamente el sensor DHT11.

La Tabla 9 contiene las pautas de instalación básica recomendada.

Tabla 9. Ubicación de sensores en el Centro de Datos UDO

Tipo de sensor	Ubicación	Práctica recomendada general	Normas del sector aplicables
Sensores de temperatura	Rack	En la parte superior, central e inferior de la puerta frontal de cada rack, para supervisar la temperatura de entrada de los dispositivos instalados en el rack.	Normas ASHRAE
Sensores de humedad	Fila	Uno por cada pasillo frío; se instala en la parte frontal de un rack del centro de la fila.	Normas ASHRAE

Para entender mejor la distribución de los sensores en el Centro de Datos UDO se elaboró un diseño de ubicación en el mismo, la cual se aprecia en la figura 19

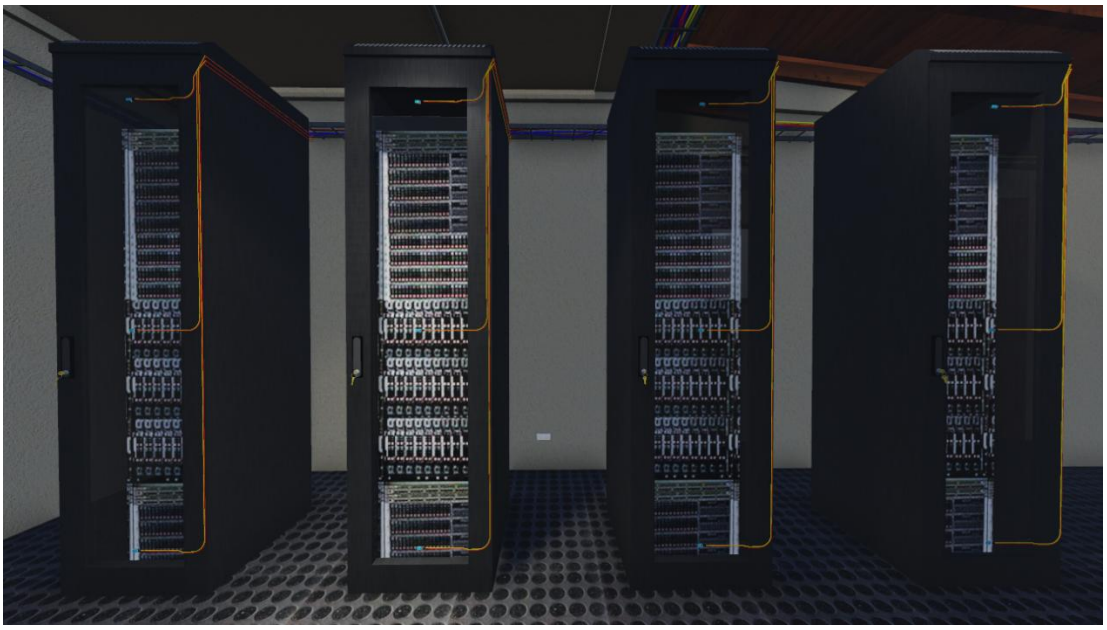


Figura 19. Distribución de los sensores DHT11 en el Centro de Datos UDO

En las siguientes figuras se muestra el sensor DHT11 ubicado en las distintas partes del Rack, en el Centro de Datos UDO.

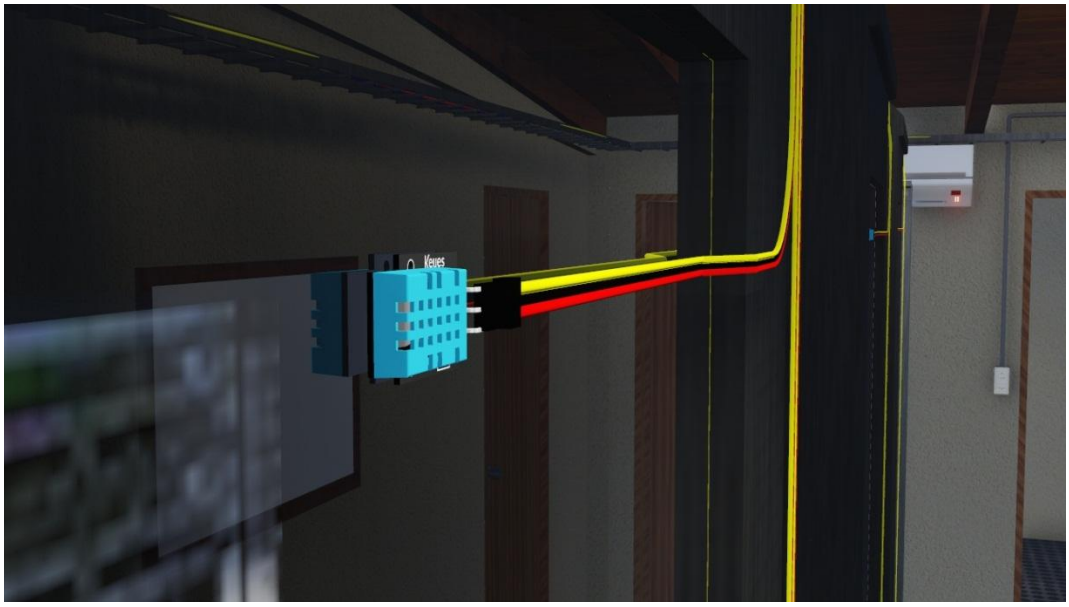


Figura 20. Sensor DHT11 ubicado en la parte superior del Rack

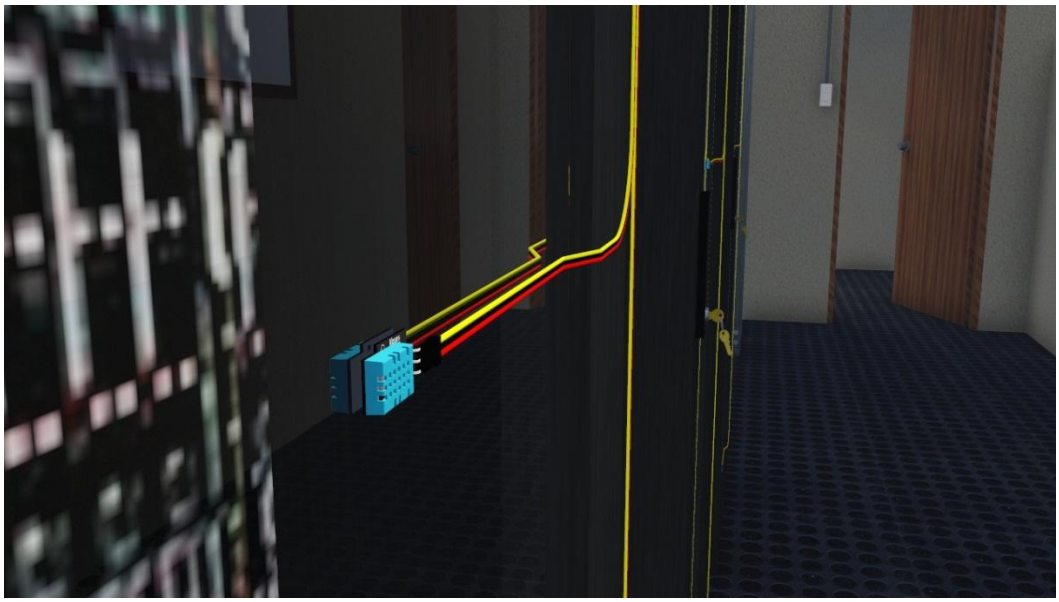


Figura 21. Sensor DHT11 ubicado en la parte central del Rack

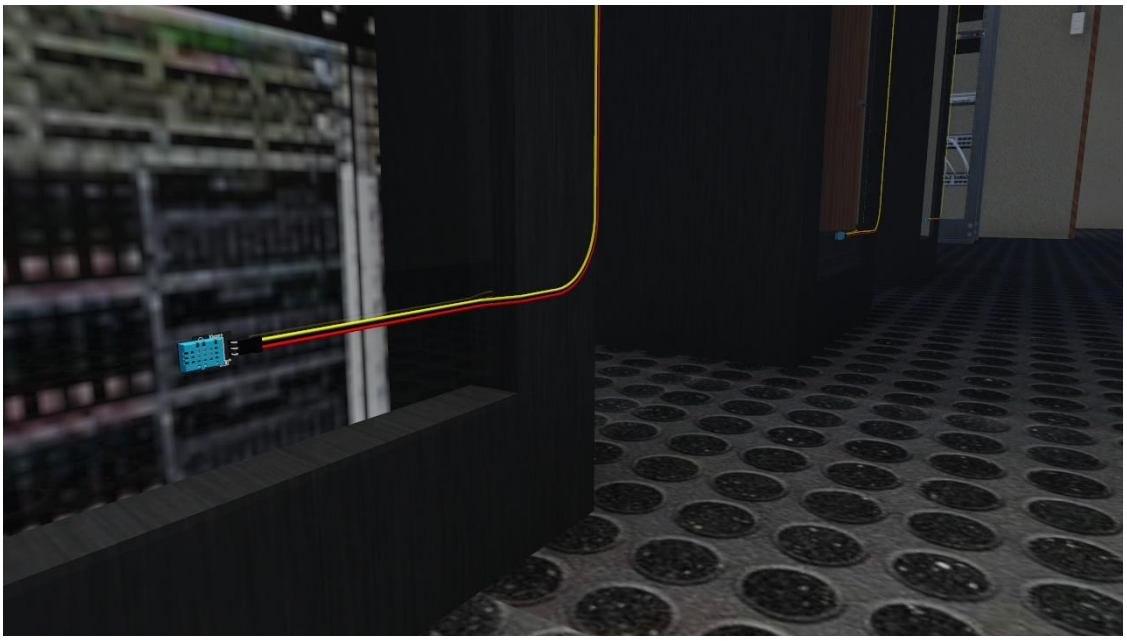


Figura 22. Sensor DHT11 ubicado en la parte inferior del Rack

Los sensores proporcionan datos brutos, pero la interpretación de estos datos es igualmente importante a la hora de generar alertas, notificaciones y correcciones necesarias. A medida que las estrategias de supervisión se vuelven más sofisticadas y se multiplica el número de sensores instalados en el centro de datos, el procesamiento de esta cantidad de datos es esencial.

Es fundamental poder filtrar, correlacionar y evaluar los datos para determinar la mejor forma de reaccionar cuando se producen eventos que están fuera de los límites establecidos. Una reacción eficaz implica alertar a la persona adecuada por medio del método apropiado y con la información oportuna. Hay tres formas posibles de acción:

1. Mediante alertas de condiciones fuera de límites que puedan poner en peligro dispositivos o racks específicos o el centro de datos en su conjunto
2. Mediante la acción basada en alertas y en rangos específicos
3. Mediante el análisis y la notificación para facilitar mejoras, optimización y mediciones de fallos

Alertas

A la hora de configurar las alertas hay que definir tres aspectos: Rango de alarma (los valores que deben registrarse para que se activen las alarmas); métodos de alerta (cómo debe enviarse la alerta y a quién) y prioridad (¿es necesario asignar ciertos tipos de alarmas a un puesto distinto del escalafón empresarial para su resolución?).

Rango de alarma: es necesario determinar para el sensor a utilizar, las condiciones de funcionamiento aceptables. Asimismo, se deben configurar umbrales para que se activen las alarmas cuando las lecturas sobrepasen las condiciones de funcionamiento establecidas. Los rangos se deben fijar con cuidado para garantizar su máxima eficacia. Puede haber umbrales diferentes que disparen alertas distintas en función de la gravedad del incidente.

En la tabla 10 se puede ver los umbrales recomendados de temperatura y humedad para un Centro de Datos según la TIA942 (*Telecomunications Industry Association*, en español, Asociación de la Industria de Telecomunicaciones) (2005)

Tabla 10. Rangos Recomendados de temperatura y humedad para los Centro de Datos.

Sensor	Rango mínimo	Rango máximo
Temperatura del aire	**16ºC**	**25 ºC**
Humedad	**40%**	**55%**

Métodos de alerta: la información de alerta se puede enviar de formas muy distintas como, por ejemplo, a través de mensajes de correo electrónico, mensajes de texto SMS y notificaciones a servidores HTTP. Es importante que los sistemas de alerta sean flexibles y personalizables para poder proporcionar correctamente la cantidad adecuada de información al destinatario oportuno. Las notificaciones de alerta deben incluir varios datos, como el nombre de la variable a sensar, y la fecha/hora de la alarma.

Prioridad de la alerta: es posible que algunas alarmas requieran una atención inmediata. Un sistema de supervisión debe ser capaz de asignar alarmas específicas a un nivel superior de autoridad si el problema no se resuelve dentro de un periodo de tiempo específico

Reacción ante los datos

La recopilación de los datos de los sensores es solo el primer paso, y el responsable del centro de datos debe estar atento a cualquier evento. Para implementar este tipo de automatización debe tenerse en cuenta lo siguiente:

Acciones de alerta: Estas acciones automatizadas pueden ser notificaciones personales, como correos electrónicos.

Visibilidad continúa en tiempo real de los datos de los sensores: la posibilidad de saber las lecturas instantáneas de los sensores a emplear es un requisito básico. La interpretación de estas tendencias permite a los administradores detectar problemas más amplios.

Análisis y notificación

Los sistemas de supervisión no solo deben incluir las tendencias a corto plazo de los datos de los sensores, sino también datos históricos a largo plazo. Los sistemas de supervisión de primera clase deben tener acceso a las lecturas de los sensores de semanas, meses o incluso años anteriores y permitir la elaboración de gráficos e informes a partir de estos datos

3.3.1.4. Prueba

Test de comunicación entre el Arduino y los sensores

Requisitos de software para las pruebas

Para la comunicación entre la tarjeta Arduino y los dispositivos a usarse, se necesitó la instalación del entorno y las librerías que deben estar en la computadora para las pruebas de comunicación y el desarrollo del programa a realizarse.

Se debe ingresar correctamente en el directorio del entorno Arduino las librerías necesarias, para que al armar el código, el programa pueda ser compilado y no presente ningún error de librerías no declaradas.

Conexión de Arduino con el Sensor DHT11

1. Se cargó la librería correspondiente del sensor DHT11 en el directorio del entorno de Arduino.

2. Se inició el entorno Arduino y se verificó que reconozca la librería, mediante el menú Sketch>Importar librería.

3. Se implementó el código de uso del sensor, para obtener una prueba de comunicación. El código tuvo la siguiente estructura. Figura 23

Figura 23. Estructura de código para prueba del sensor DHT11.

4. Se compiló el código para probar que no haya errores en el programa.

5. Se realizó la conexión de los sensores DHT11 y la tarjeta Arduino Uno. Se muestra en la figura 24

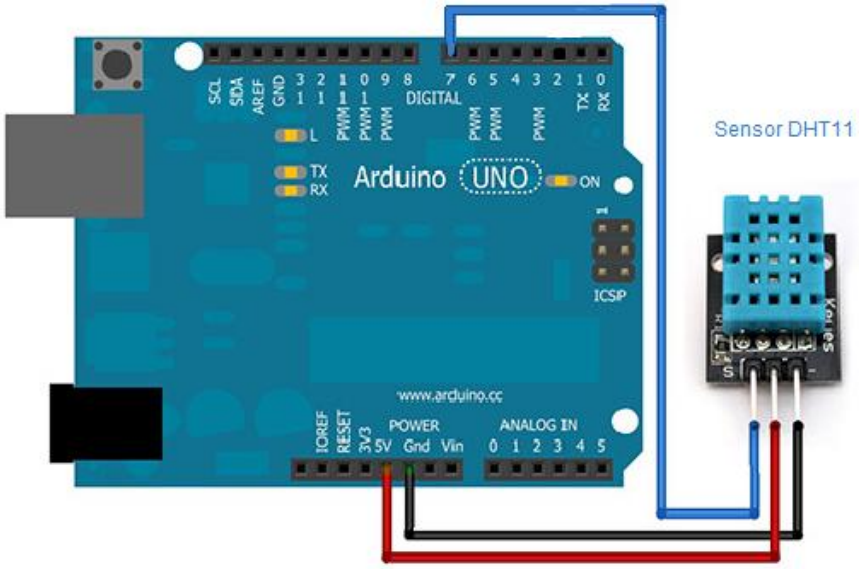


Figura 24. Conexión entre Arduino Uno y sensor DHT11.

6. Se conectó la tarjeta Arduino al computador mediante un cable USB y se verifico que el entorno reconozca al puerto COM (COMmunications, en español, comunicaciones), reseteó el Arduino y cargó nuevamente el programa.

7. Se confirmó la comunicación entre el Arduino y el sensor DHT11. El sensor se encarga de la toma de los datos de la temperatura y la humedad. Figura 25

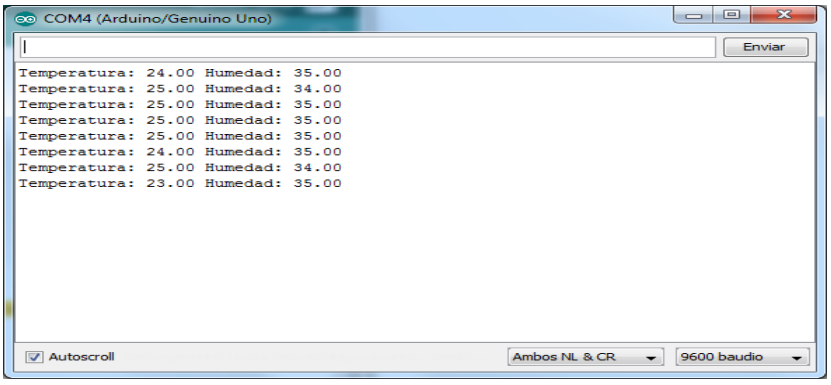


Figura 25. Datos de temperatura y humedad del sensor DHT11.

3.3.2 Software

3.3.2.1. Software para el sistema de monitoreo

A continuación se describe el proceso de diseño del software para el sistema de monitoreo

Modelado de negocio.

El sistema de monitoreo que se desarrolló consistió en el diseño de un circuito utilizando el dispositivo Arduino, el cual está conectado un sensor de temperatura y humedad por medio de uno de sus puertos, donde tomó muestras y estas se enviaron por Ethernet a el servidor Nagios que es capaz de enviar alertas por correo electrónico al operador

El sistema de monitoreo Nagios, se seleccionó principalmente porque aparte de que es software libre, permite el monitoreo de hardware, y servicios de red, cuenta con la posibilidad de programar plugins específicos para nuevos sistemas. En este plugin se le incluyeron rutinas de comparación que permitan determinar en qué momento el valor de la temperatura y humedad ha salido de un rango aceptable.

Otra característica de Nagios es que permite enviar notificaciones cuando hay errores o se ha restablecido el servicio. Este dispone de una interfaz web que muestra el estado de los diferentes servicios en tiempo real.

Requerimientos

En esta disciplina no se registró ningún nuevo requisito, por lo cual, los existentes hasta ahora se tomaron como definitivos.

En esta iteración se cumplió con los requisitos planteados en las fases de inicio.

Análisis y diseño

Diseño del firmware del Arduino para el sistema de monitorización

A continuación en la figura 26, se presenta un diagrama de flujo del programa principal de Arduino que se utilizó para el desarrollo del proyecto, primero se dio inicio al programa, luego se definió la dirección MAC del dispositivo, esta dirección ya viene estandarizada en la placa Arduino Ethernet Shield, después se pasó a definir la dirección IP del mismo, que este caso es fija, con esto conseguimos que la conexión entre el Router e Internet no cambie, así se podrá acceder siempre desde la misma dirección IP, luego se pasa a definir el número del puerto TCP, se definen las variables a utilizar en el programa y por ultimo pasa a ejecutar la función setup que es la encargada de recoger la configuración.

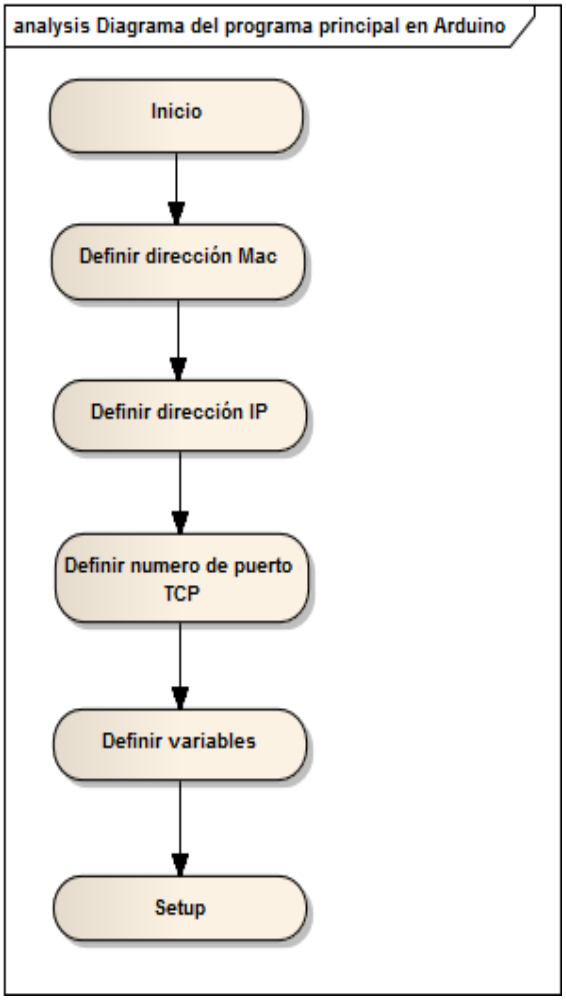


Figura 26. Diagrama de flujo del programa principal en Arduino

La rutina setup se encarga de definir la configuración inicial de las variables y procesos que se requieren para la ejecución del software. El diagrama de flujo de esta rutina se muestra en la figura 27.

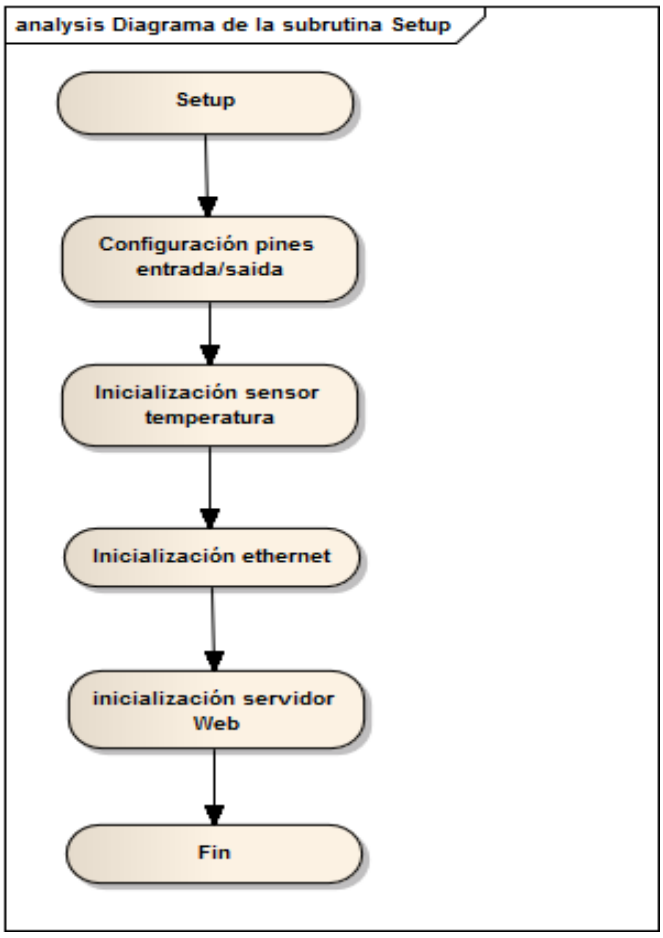


Figura 27. Diagrama de flujo de la subrutina Setup.

Cuando se termina el proceso de la subrutina Setup el Arduino, revisa el sensor, y envía una respuesta al cliente con el valor de temperatura y humedad actual. La transferencia de datos se hace por medio de solicitudes GET

Para la comunicación con el sensor de temperatura y humedad, Arduino cuenta con la librería DHT11.h que contiene el protocolo 1-wire. Esta debe ser importada al programa principal.

El sistema de monitoreo se comunica con el plugin del software Nagios, el cual evalúa las variables y de acuerdo al valor de temperatura y humedad mostrara el estado en que se encuentra el centro de datos UDO, estos se subdividen en cuatro niveles que son: UNKNOWN (desconocido), OK (bueno), WARNING (peligro) y CRITICAL (critico). El estado UNKNOWN se da en caso de que no exista comunicación con el dispositivo. OK significa que la temperatura y humedad se encuentran en un rango estable; WARNING se da cuando la temperatura está llegando a un nivel crítico y el estado CRITICAL significa que supero los límites de seguridad estándar de temperatura y humedad para el centro de datos.

El Arduino está programado para realizar funciones de servidor WEB; después de abrir un navegador web e ingresar la dirección IP asignada al Arduino Ethernet Shield, este responderá mostrando la página web que esta almacenada en la tarjeta de memoria micro SD. Para esto se agregó la librería SD.h al programa principal del arduino, esta contiene el driver necesario para la lectura de datos de la página.

Diseño del plugin de Nagios para el sistema de monitorización

Los Plugins son códigos a los que Nagios hace referencia para monitorizar un host, dispositivos, servicios, protocolos. El desarrollo de estos se pueden utilizar diferentes lenguajes de programación como por ejemplo: C, Python, Perl, PHP, Ruby, etc.

El plugin para la comunicación entre Nagios y Arduino fue escrito en lenguaje Python. En la figura 28, se observa el diagrama de flujo en el cual se basa el código del plugin desarrollado

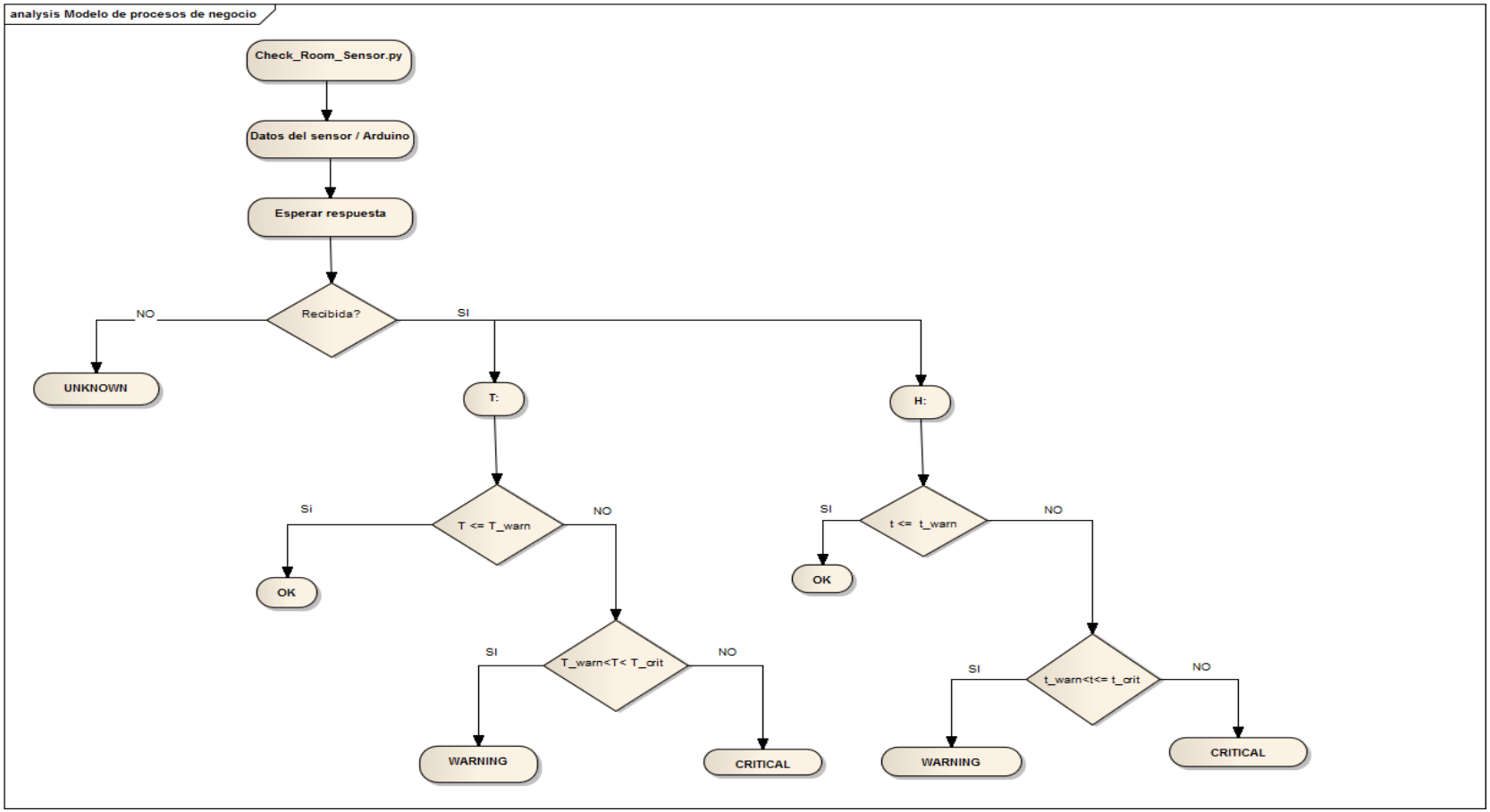


Figura 28. Diagrama de flujo del plugin de Nagios.

Nagios ejecuta este plugin check_room_sensor.py y se obtiene los datos del sensor/Arduino por medio de la URL /IP, seguidamente espera la respuesta. En el código del plugin se escribe un algoritmo que permite identificar cada uno de los servicios (T: Temperature, H: humedity). Los valores de T_warn, T_crit, t_warn y t_crit, son parámetros que se definen en los archivos de configuración de Nagios, la T (mayúscula) es de temperatura en °C y la H es humedad.

Configuración servicios de Nagios

Nagios es un sistema de monitorización de redes de código abierto bajo la licencia GPL, y su principal función es observar el comportamiento de host (servidores, switch, router, impresoras, etc.) y servicios de red (software) que se especifiquen, alertando vía correo electrónico cuando el comportamiento de los mismos no sea el deseado. Entre sus características principales están:

Monitorización de servicios de red SMTP, HTTP, SNMP, SSH, DNS, etc.

Monitorización de recursos de hardware como por ejemplo: carga del procesador, uso de los discos, memoria, estado de los puertos

Programación de nuevos *plugins* desarrolladas en diferentes lenguajes de programación (Bash, C, C++, Perl, Ruby, Python, PHP, C#, Java, etc.).

Envío de notificaciones

Interfaz web que permite la visualización de los servicios, genera estadísticas, historial de alarmas, notificaciones, entre otros

Interfaz web que permite la visualización de los servicios, genera estadísticas, historial de alarmas, notificaciones, etc.

La instalación de Nagios no se abarca en este documento, sin embargo en la página principal http://www.Nagios.org, se encuentra un tutorial de instalación

Nagiosgraph

Es un agregado para el sistema de monitoreo Nagios, que permite la visualización de gráficas sobre los servicios monitoreados

Postfix

Es un servidor de correo de código abierto, para el enrutamiento y envío de correo electrónico, creado con la intención de que sea una alternativa más rápida, fácil de administrar y segura.

Prueba

Las actividades de la disciplina de pruebas serán ejecutadas en la fase de construcción.

3.3.2.2 Software del sistema web.

Modelado de negocio

De acuerdo a los requerimientos funcionales de la fase inicio se modelaron los requerimientos desde la perspectiva de los usuarios. Tenemos el caso de uso del sistema figura 26. La descripción del caso de uso, se encuentran en el apéndice C

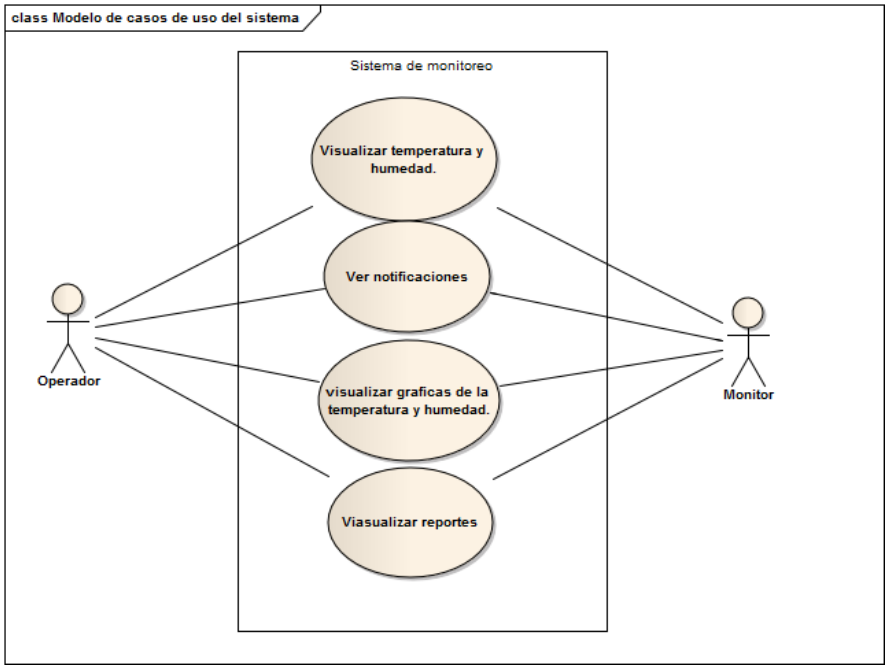


Figura 29. Caso de uso general del sistema de monitoreo

Requisitos

Especificación de Requisitos

Plantillas de Volére

Con el propósito de documentar los requisitos recolectados se procedió a la utilización de la plantilla de Volére. La plantilla de especificación de requisitos Volére está creada para ser utilizada como una base para las especificaciones de requisitos, estas se muestran en el apéndice

Análisis y diseño.

Vista lógica.

La vista lógica del sistema de monitoreo se encuentra representada por la realización de los casos de uso mediante los Diagramas de secuencia, artefacto que buscan establecer una asociación entre las tareas que podrá desarrollar el usuario dentro del sistema y la forma en que debe ser capaz de percibir lo que está haciendo, definen el principio mediante el cual se diseña la interacción. Los otros diagramas de secuencia se encuentran en el apéndice F

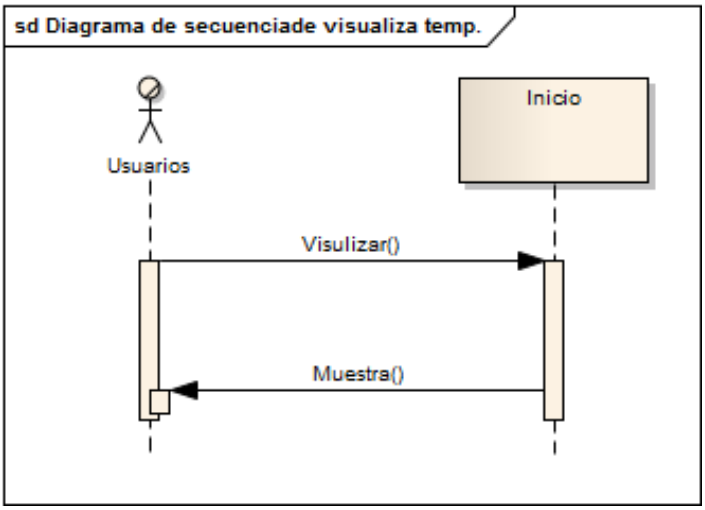


Figura 30. Diagrama de secuencia para visualizar la temperatura y humedad para los usuarios visiten el sistema

También se realizó en esta vista el Diagrama de clase que se muestra en la figura 28.

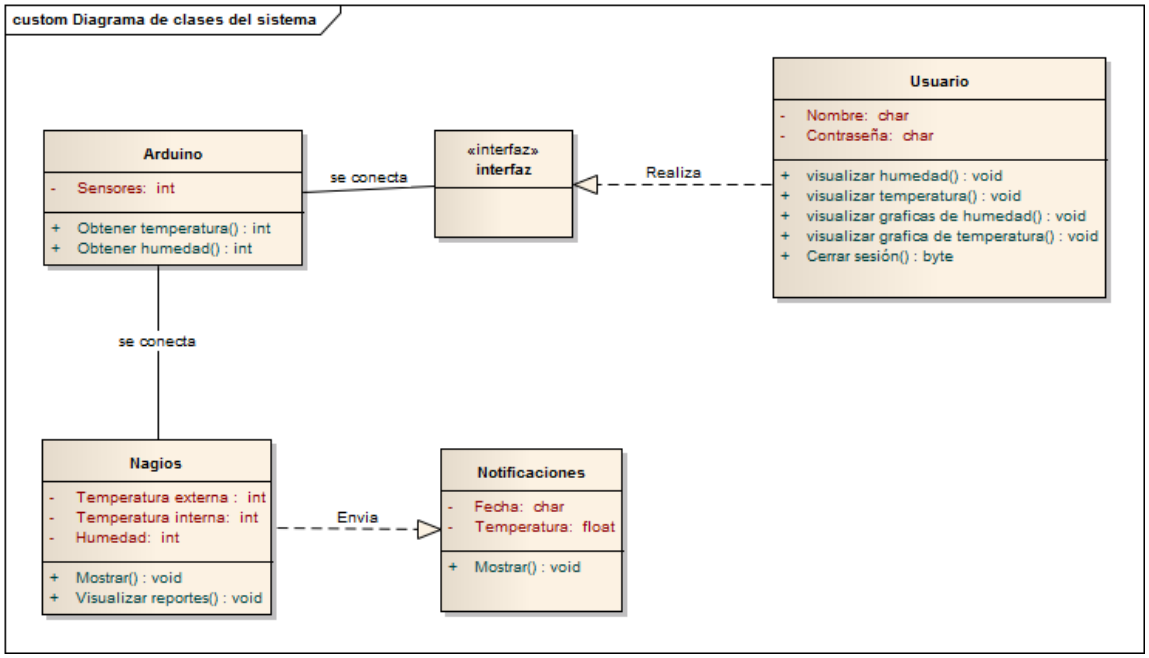


Figura 31.Diagrama de clases del sistema

Vista de implementación

En esta etapa del desarrollo del sistema, mediante un acoplamiento del diseño conceptual a requerimientos como lenguaje, plataforma de desarrollo, herramientas de desarrollo y otros, fueron definidos los detalles de la implementación de la aplicación. Esta vista cuenta con un diagrama de componentes que explica la relación entre los distintos artefactos que conforman la aplicación. Para ello se identificaron aquellos componentes que formarían parte de la aplicación y posteriormente se determinó la asociación entre ellos.

Vista de despliegue

Esta vista especifica los detalles del despliegue, instalación y operación del sistema. Para describir en que nodos de hardware se instalarían los diferentes componentes de la aplicación, se elaboró el diagrama de despliegue

Diseño de interfaz

El diseño de interfaz permitió establecer el conjunto de pantallas que formarán parte del aspecto visual del sistema de monitoreo, así como también el modelo de la navegación y el contenido de la misma.

Para el diseño de interfaz se consideraron algunos de los principios planteados por Tognozzi, citado por Pressman (2005), con la finalidad de que la interfaz sea fácil de utilizar, fácil de aprender, intuitiva, consistente, libre de errores y eficiente. Los principios considerados son los siguientes:

Comunicación: la interfaz comunica el estado de cualquier actividad iniciada por el usuario.

Eficiencia: el diseño de la interfaz optimiza el trabajo del usuario.

Consistencia: el uso de los controles de navegación, menús, iconos y aspectos estéticos son consistentes en toda la interfaz.

Flexibilidad: la interfaz es flexible pues permite que los usuarios puedan realizar sus tareas directamente y para que puedan explorar la aplicación más a fondo.

Centrada en el usuario: la interfaz se centra en las tareas que los usuarios deben realizar.

Autonomía controlada: la aplicación web está diseñada para que el contenido al que accede el usuario esté acorde con su perfil, y la navegación hacia áreas fuera de su alcance se controlen a través de la identificación de los usuarios.

Legibilidad: la información que se presenta a través de la interfaz es legible por cualquier usuario.

Establecidos los principios para el diseño de la interfaz, se eligieron el tipo de letras, tamaños y colores, fondos, entre otros. Para que la interfaz fuese estéticamente agradable se escogieron tonos azules para los fondos, bordes, etc. Se utilizó el framework Bootstrap el cual permitió adaptar la página en torno a cualquier dispositivo proveyendo al usuario un gran abanico de opciones con respecto a adaptabilidad en los dispositivos

Diseño de contenido.

El contenido mostrado en este sistema de monitoreo se encuentra distribuido de manera uniforme, para ello se definió y estructuro el mismo. Se hizo un bosquejo de todo el contenido que se presenta en el sistema.

El contenido se distribuyó de la siguiente manera: en la parte superior un banner que funciona como encabezado de la página, en la parte izquierda presenta los bloques de navegación que emplean funciones primordiales dentro del sistema de información web (ver figura 32).

Figura 32. Estructura de presentación del contenido.

Diseño de navegación

El flujo de navegación entre los objetos de contenido se plasmó a través de los bloques de navegación que permiten los enlaces con los diversos componentes que comprenden el sistema de monitoreo. Estos bloques contienen toda la información relacionada a cada enlace o botón ya que la interfaz se diseñó de tal manera que fuera lo suficientemente intuitiva para los usuarios y se mantuviera una navegación uniforme luego que se haya iniciado sesión con un nombre de usuario y una contraseña asignada

Tabla 11.Descripción del mapa de navegación

Rutas de Navegación.	Usuario.
Iniciar Sesión.	El usuario puede iniciar sesión con su nombre de usuario y contraseña
Inicio.	El sistema presenta una interfaz de inicio con una descripción de las funcionalidades de la herramienta y una breve descripción de la misma. El usuario cuenta con un menú superior con las opciones de: **Inicio:** Muestra la temperatura y humedad en tiempo real. **Graficas:** Muestra por separado las gráficas de temperatura y humedad que determina las variaciones detectadas. **Mapa:** Muestra donde está ubicado el centro de datos (U.D.O). **Cerrar Sesión:** Para salir y volver a la ventana de iniciar sesión.

Pruebas

En esta iteración se mantienen las pruebas establecidas en la fase pasada, por lo tanto no existen cambios que destacar.

3.4 FASE III: CONSTRUCCIÓN

El propósito de esta fase es completar la funcionalidad del sistema.

Primera iteración

3.4.1 Modelado de negocio

En esta fase el modelado de negocio no sufrió ninguna modificación por lo cual se tomó como definitiva para el proceso

3.4.2 Requerimientos

En esta disciplina no se registró ningún nuevo requisito, por lo cual, los existentes hasta ahora se tomaron como definitivos.

En esta iteración se cumplió con los requisitos planteados en las fases de inicio y de elaboración.

3.4.3 Análisis y diseño

A continuación se muestra el código final que se utilizó para el programa de Arduino

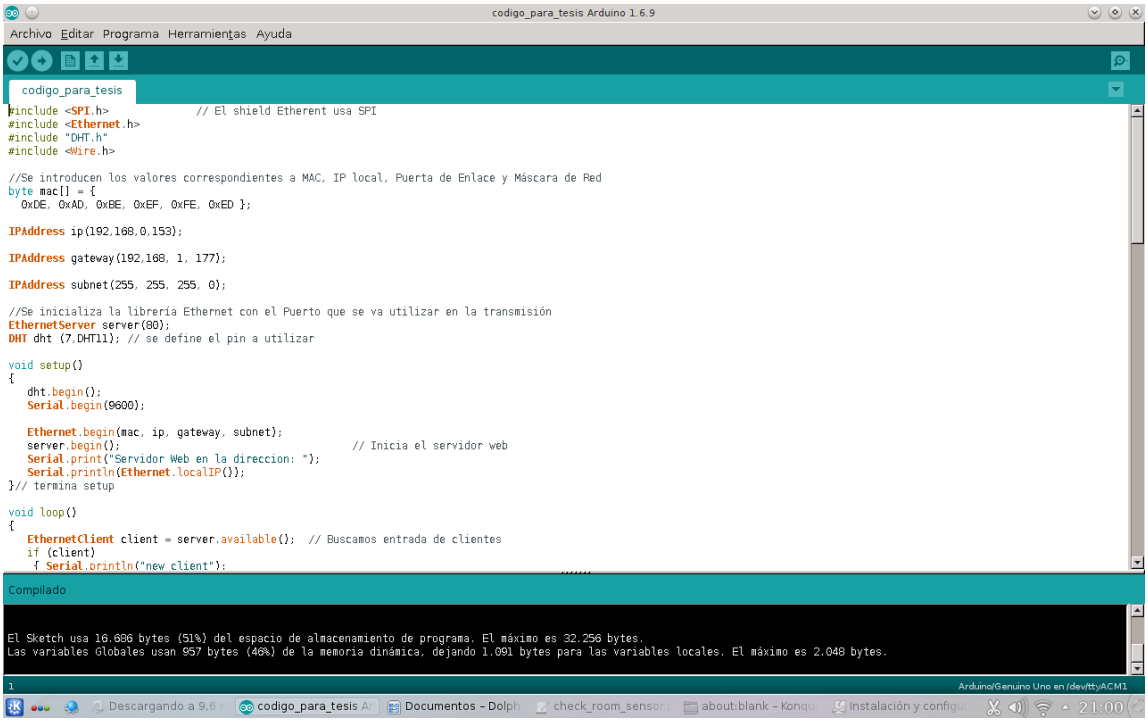


Figura 33. Código utilizado para el sistema de monitoreo parte 1

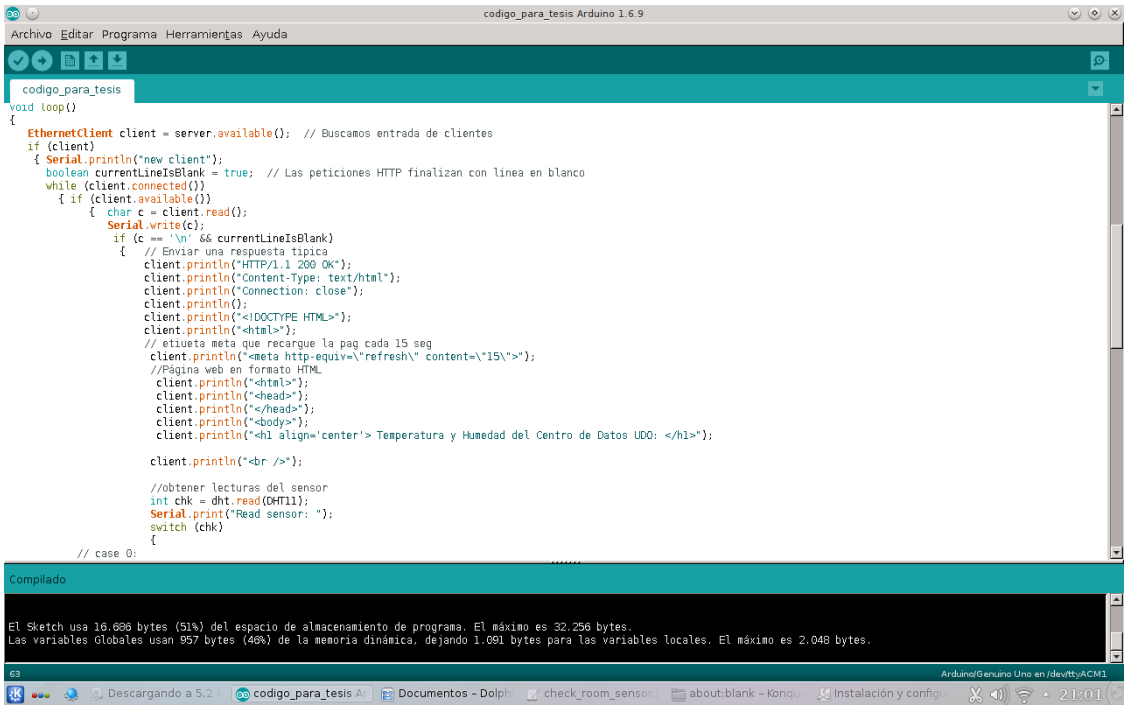


Figura 34. Código utilizado para el sistema de monitoreo parte 2

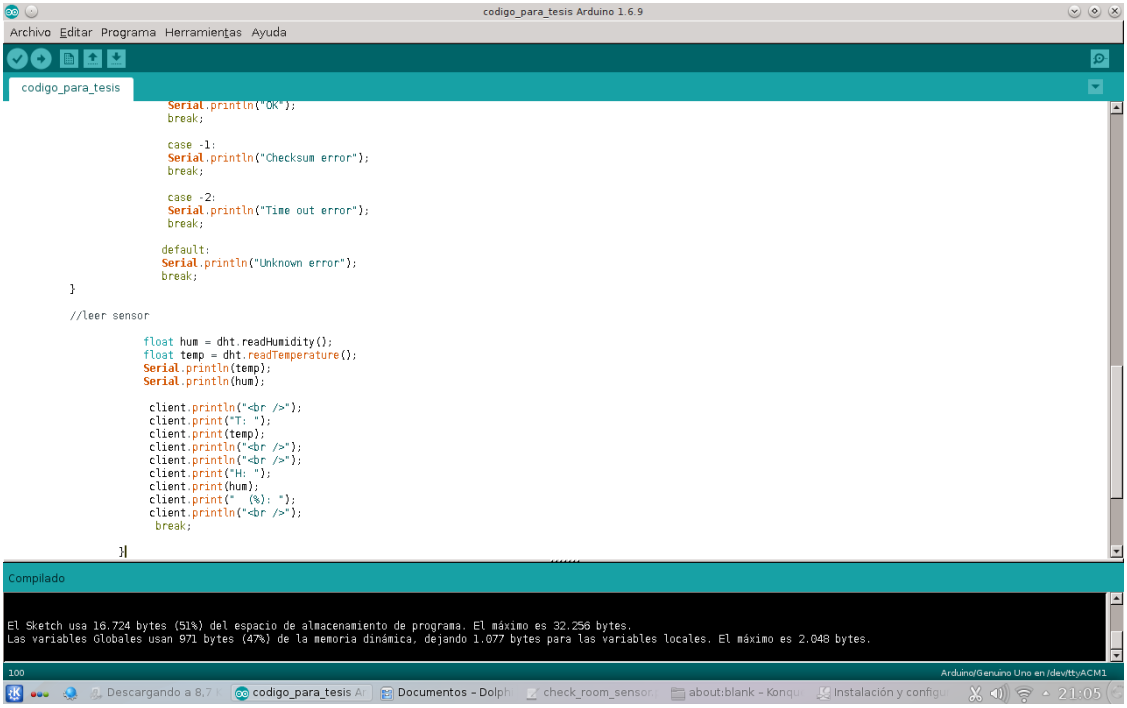


Figura 35. Código utilizado para el sistema de monitoreo parte 3

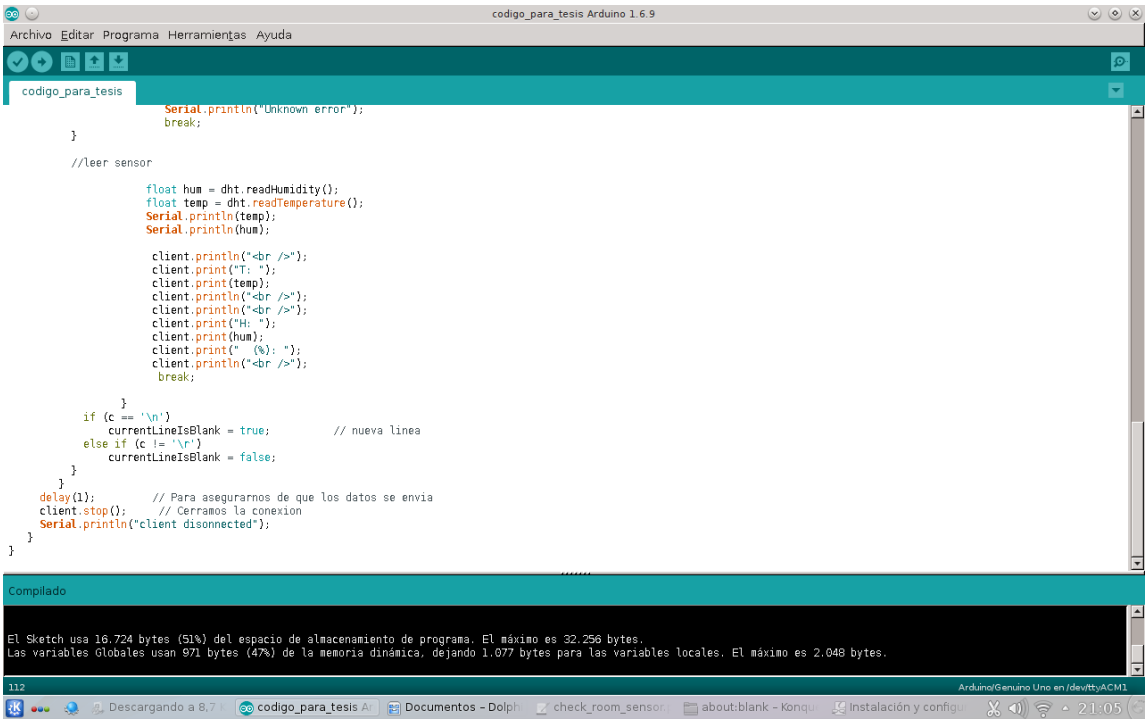
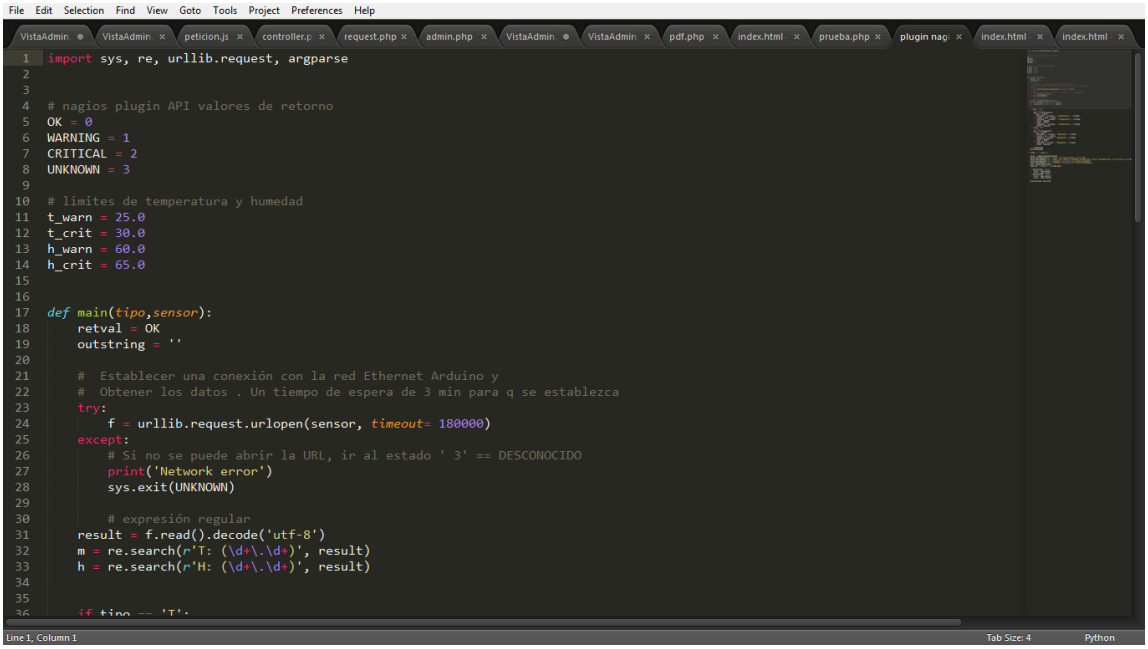


Figura 36. Código utilizado para el sistema de monitoreo parte 4

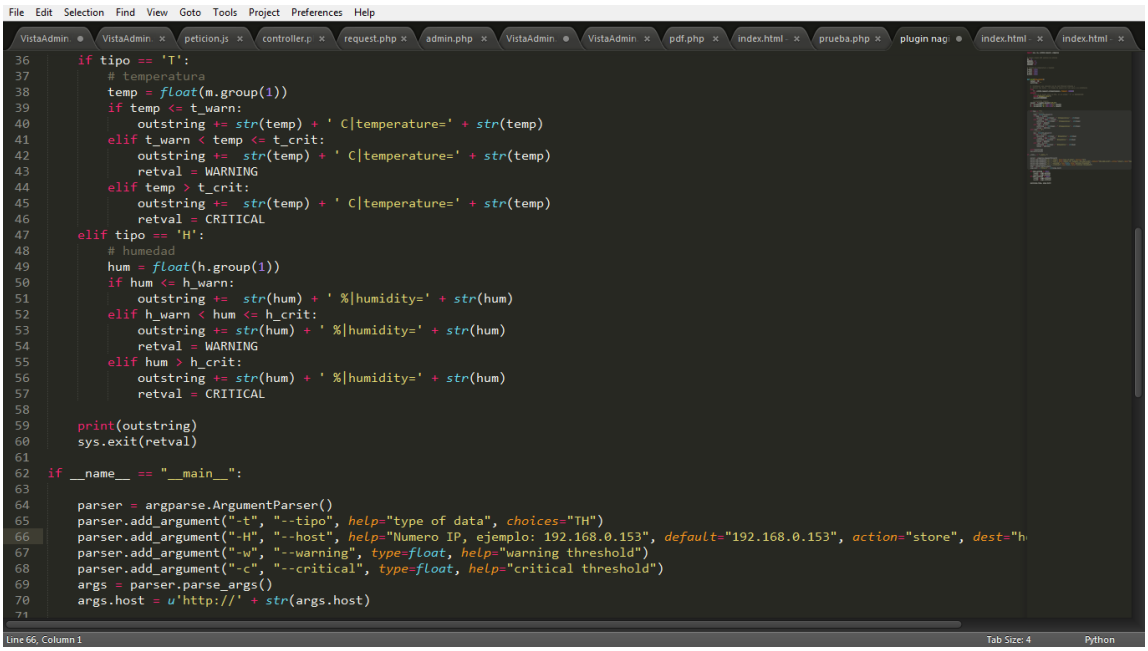
Luego de tener el programa en Arduino pasamos a construir el plugin que interpretó Nagios para la comunicación entre las dos herramientas.



```python
import sys, re, urllib.request, argparse


# nagios plugin API valores de retorno
OK = 0
WARNING = 1
CRITICAL = 2
UNKNOWN = 3

# limites de temperatura y humedad
t_warn = 25.0
t_crit = 30.0
h_warn = 60.0
h_crit = 65.0


def main(tipo,sensor):
    retval = OK
    outstring = ''

    #  Establecer una conexión con la red Ethernet Arduino y
    #  Obtener los datos . Un tiempo de espera de 3 min para q se establezca
    try:
        f = urllib.request.urlopen(sensor, timeout= 180000)
    except:
        # Si no se puede abrir la URL, ir al estado ' 3' == DESCONOCIDO
        print('Network error')
        sys.exit(UNKNOWN)

        # expresión regular
    result = f.read().decode('utf-8')
    m = re.search(r'T: (\d+\.\d+)', result)
    h = re.search(r'H: (\d+\.\d+)', result)


    if tipo == 'T':
```

Figura 37. Código del plugin para Nagios parte 1

.



```python
    if tipo == 'T':
        # temperatura
        temp = float(m.group(1))
        if temp <= t_warn:
            outstring += str(temp) + ' C|temperature=' + str(temp)
        elif t_warn < temp <= t_crit:
            outstring +=  str(temp) + ' C|temperature=' + str(temp)
            retval = WARNING
        elif temp > t_crit:
            outstring +=  str(temp) + ' C|temperature=' + str(temp)
            retval = CRITICAL
    elif tipo == 'H':
        # humedad
        hum = float(h.group(1))
        if hum <= h_warn:
            outstring +=  str(hum) + ' %|humidity=' + str(hum)
        elif h_warn < hum <= h_crit:
            outstring += str(hum) + ' %|humidity=' + str(hum)
            retval = WARNING
        elif hum > h_crit:
            outstring += str(hum) + ' %|humidity=' + str(hum)
            retval = CRITICAL

    print(outstring)
    sys.exit(retval)

if __name__ == "__main__":

    parser = argparse.ArgumentParser()
    parser.add_argument("-t", "--tipo", help="type of data", choices="TH")
    parser.add_argument("-H", "--host", help="Numero IP, ejemplo: 192.168.0.153", default="192.168.0.153", action="store", dest="h
    parser.add_argument("-w", "--warning", type=float, help="warning threshold")
    parser.add_argument("-c", "--critical", type=float, help="critical threshold")
    args = parser.parse_args()
    args.host = u'http://' + str(args.host)
```

Figura 38. Código del plugin para Nagios parte 2

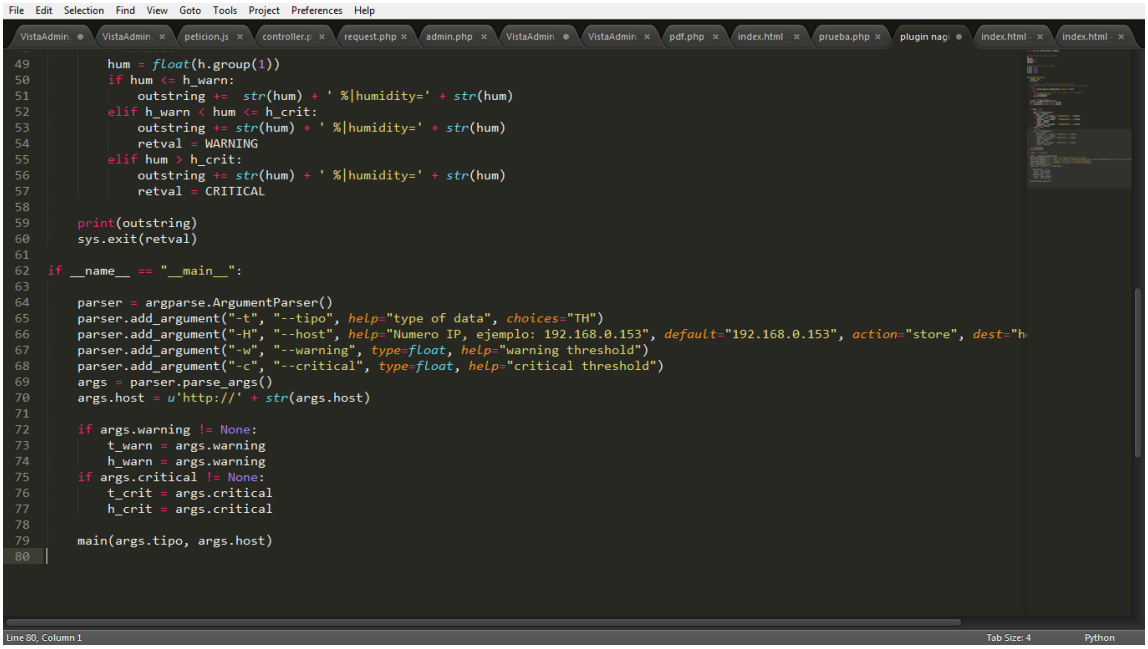


Figura 39. Código del plugin para Nagios parte 3

Configuración de Nagios para el monitoreo de los sensores de humedad y temperatura.

Una vez instalado Nagios, se procedió a la creación de un Host y la configuración de los comandos y servicios para el monitoreo de los sensores de humedad y temperatura.

Primeramente se define un Host, como se muestra en la figura 40.Lo principal es definir el nombre del host, y la dirección IP del dispositivo.

```
Define host
{

use                     linux-server,host-pnp
host_name               Arduino
alias                   monitoreo
address                 192.168.1.177
max_check_attempts      5
check_period            24x7
check_interval          5
notification_interval   15
icon_image              A.png

}
```

Figura 40. Configuración del Arduino como un host en Nagios

Una vez definido el host, se establecen los servicios a este dispositivo, en la figura 40 se muestra la configuración.

```
Define service
{

    use                     generic-service,srv-pnp
    host_name               Arduino
    service_description     Temperature
    check_command           check_room_sensor!'T'!25!30
    check_interval          1
}


Define service
{
      use                   generic-service,srv-pnp
      host_name             Arduino
      service_description   Humidity
      check_command         check_room_sensor!'H'!60!65
      check_interval        1
}
```

Figura 41. Configuración de los servicios a monitorizar por Nagios en el Arduino

Por último se define el comando concreto que Nagios utiliza para ejecutar el *plugin*. Este se muestra en la figura 42

```
Define command
{
      command_name check_room_sensor
      command_line /usr/bin/python3 /usr/local/nagios/libexec/check_room_sensor.py --tipo $ARG1$ --host $HOSTADDRESS$  --warning $ARG2$ --critical $ARG3$

}
```

Figura 42. Comando utilizado para ejecutar el plugin

3.4.4 Pruebas

3.4.4.1 Prueba de adquisición de datos

La integración de Nagios y Arduino para la monitorización de diversos servicios de red, y equipos permiten la administración y prevención de errores.

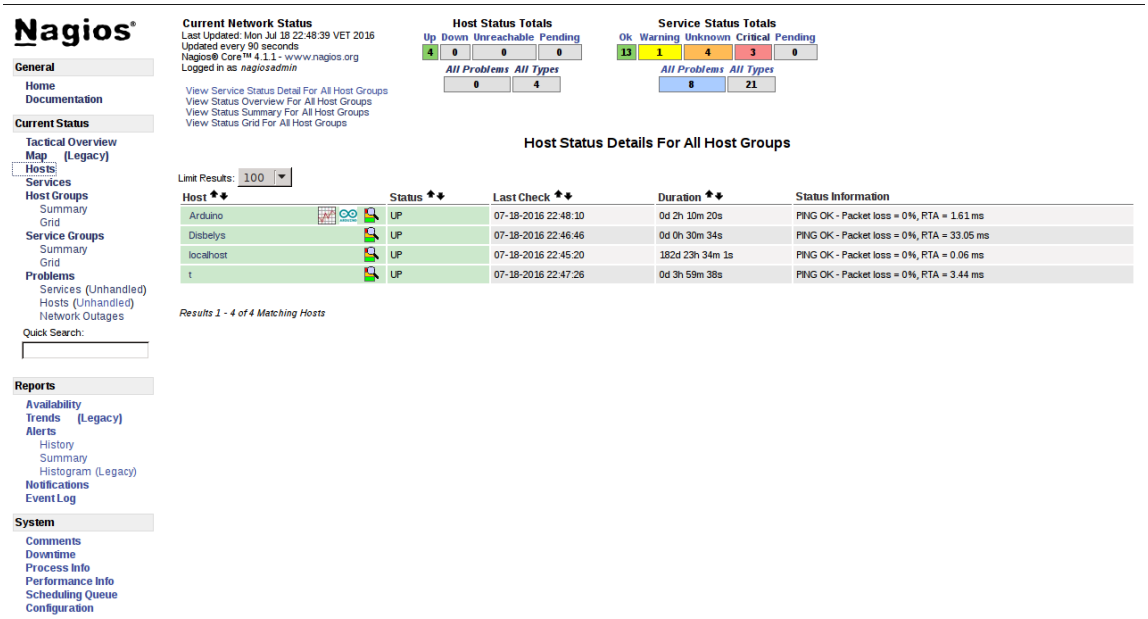


Figura 43. Captura de los host monitoreados

La construcción de la infraestructura de monitoreo se basó en el software Nagios. En este, inicialmente se configuró para administrar 4 hosts de prueba los cuales son: Arduino, Servidor Windows, servidor local Nagios, y el router/switch. En la figura 43 se

muestra la estructura gráfica de cómo está definido el Nagios, en este se pueden observar los host monitorizados, y en verde indican que están en estado “up”.

Figura 44. Detalles del host Arduino.

El estado de cada host puede ser analizado de manera independiente. Algunas de las opciones son: Reporte de disponibilidad para un host, detalle del estado del host e Historial de alerta.

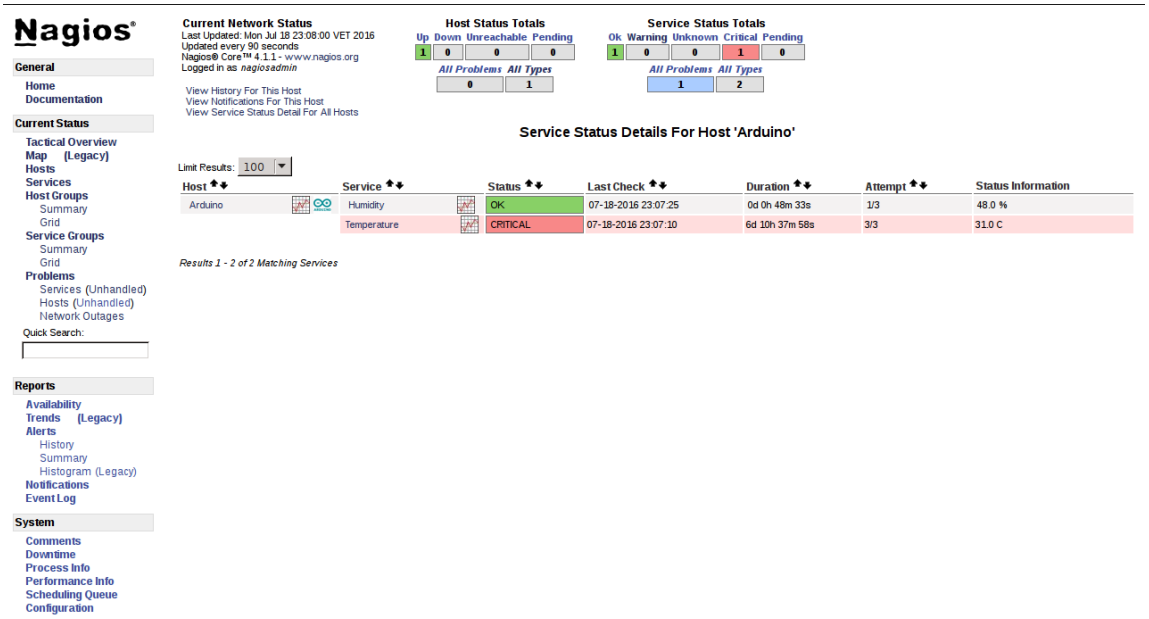


Figura 45 Servicios a monitorear

Para observar los servicios monitorizados por cada host y el estado actual de estos, se puede entrar por medio de la pestaña servicios al lado izquierdo de la interfaz. Para el sistema desarrollado, en la figura 45, se observa el host Arduino con sus correspondientes servicios, donde describe el estado (OK, WARNING, CRITICAL, UNKNOWN), el último chequeo de cada servicio, duración, intento y por último la información, que en este caso arroja la temperatura y humedad..

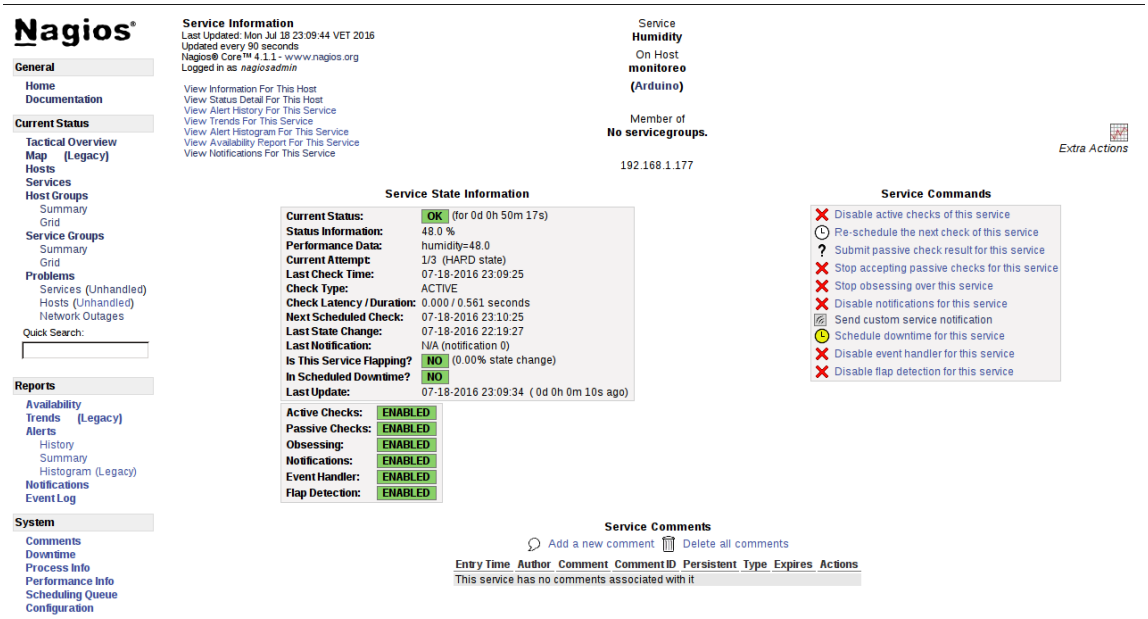


Figura 46.Detalles del servicio humedad

En la figura 46 se muestra en detalle el servicio de humedad, donde se observa propiedades asociadas a él, tales como ultimas notificaciones, probabilidades de cambio del servicio, historial, última actualización.

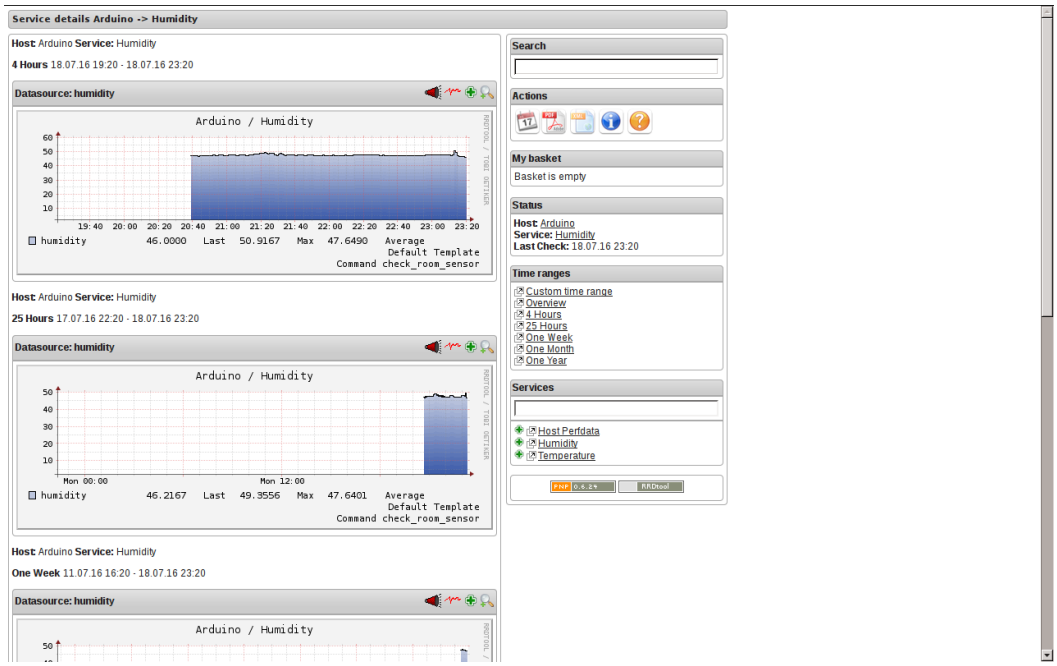


Figura 47. Gráfica del servicio humedad

En la figura 47 visualizamos las gráficas sobre el servicio de humedad, las cuales están organizadas según calendario e imprime los reportes en PDF.

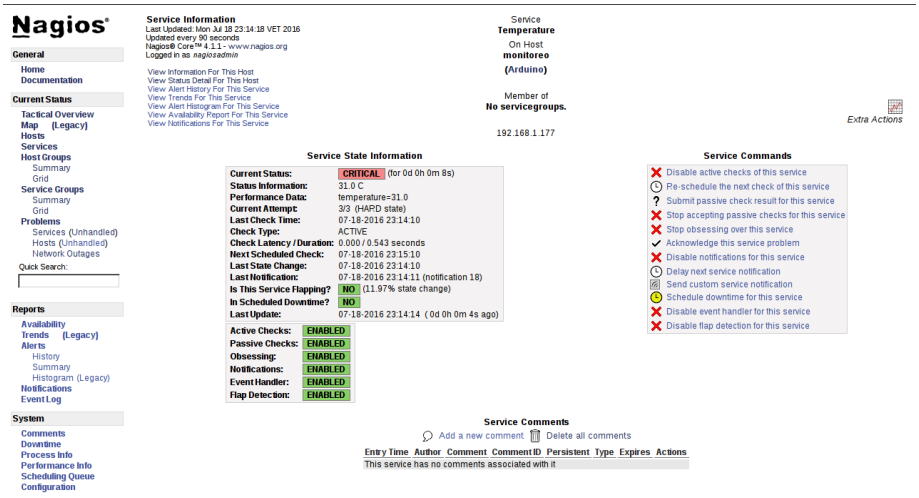


Figura 48.Detalles del servicio temperatura

Figura 49. Gráfica del servicio temperatura

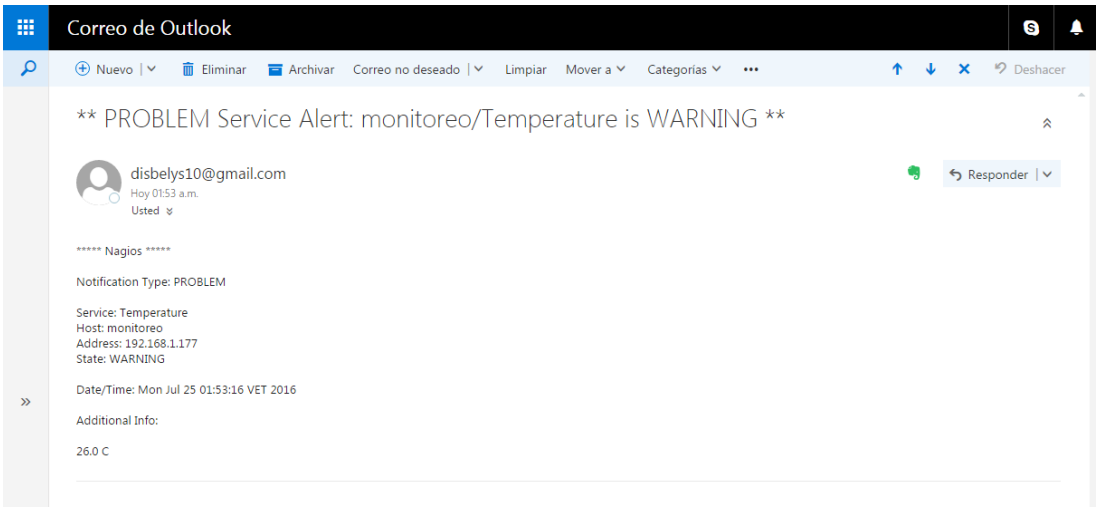


Figura 50. Notificación de cambio de estado vía correo.

3.4.4.2 Prueba de interfaz

Figura 51. Pantalla principal.

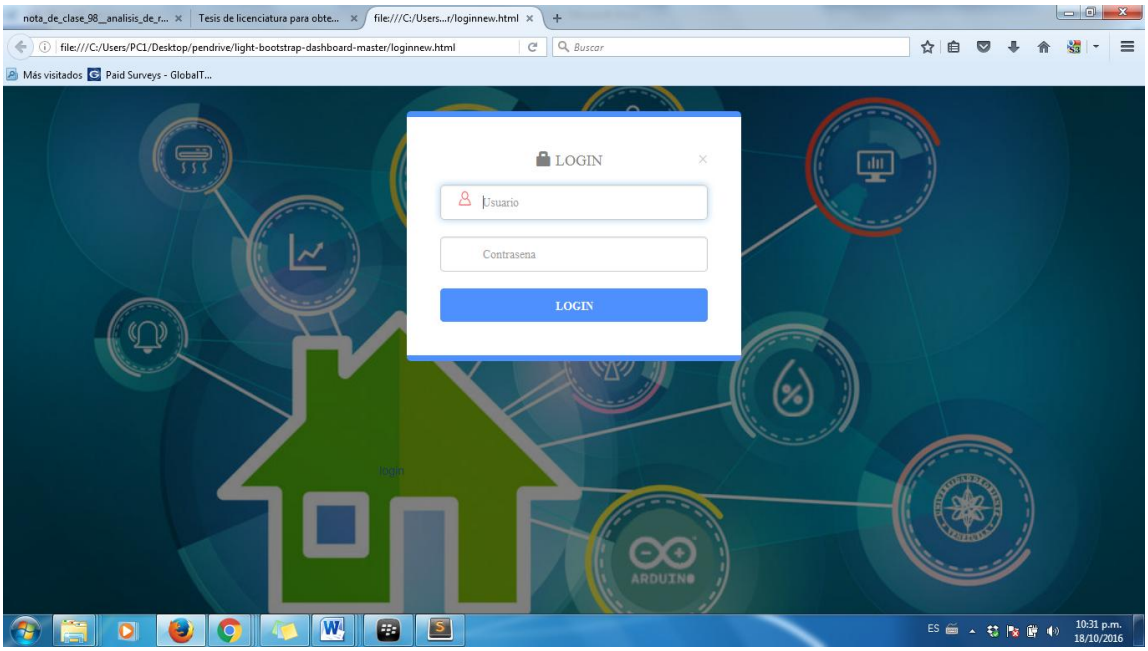


Figura 52. Iniciar sesión.

Figura 53. Pantalla de inicio.

.

CONCLUSIONES

El Proceso Unificado Racional (RUP), brindó una herramienta de trabajo buena para la construcción del sistema paso a paso. Se logró adaptarla a las necesidades y requisitos del proyecto

Las distintas disciplinas presentes en la metodología RUP ayudaron a la recolección y análisis de información, permitiendo una comprensión profunda del negocio y los requisitos que debían ser satisfechos. Esto facilitó la creación de un plan de trabajo y ayudó a definir los objetivos principales que garantizan la funcionalidad del producto final.

Se diseñó el *plugin* para la comunicación del sistema Nagios y la plataforma Arduino, para la adquisición de datos de las variables ambientales, analizando su comportamiento para tomar las respectivas medidas ante un evento inesperado.

El sistema de monitoreo se puede adaptar a otros Centros de Datos y en general se puede emplear en cualquier sitio donde se quiera monitorear variables ambientales.

Se puede realizar un análisis de datos para inspeccionar, y transformar datos con el objetivo de resaltar información útil, lo que sugiere conclusiones, y apoyo a la toma de decisiones en cualquier situación de riesgo.

RECOMENDACIONES

Con respecto a posibles desarrollos futuros, se podrían añadir nuevos módulos para agregar diferentes sensores, con el fin de elaborar redes más complejas y así maximizar los beneficios para el Centro de Datos UDO.

Implementar un sistema de control supervisorio automatizado.

BIBLIOGRAFIA

Arias, F. 2006. El proyecto de investigación introducción a la metodología científica. (5ª ed.). Episteme, Caracas.

Barban A. 1999. Gestión de Red. Edición UPC. Barcelona, España.

Balcelis J. y Romeral, J. 1997. Autómatas programables. Barcelona, España

Banzi, M. 2012. Arduino. Obtenido de What is arduino: http://www.arduino.cc/

Barrientos, I, y Beites, J. (2006, Marzo). Nagios un Sistema de Monitorización de Servicios de Red. Ponencia presentada en la VI Jornada de software libre. Oviedo (Asturias)

Becerra, J. 2007. Automatización, control y supervisión remota del sistema central de aire acondicionado (agua helada) para un edificio. Trabajo de pregrado. Departamento de Ingeniería Eléctrica, Universidad Central de Venezuela, Caracas – Venezuela.

Betancourt, A. 2008. Diseño e implementación de un sistema para el control de iluminación, temperatura e inundación, en un ambiente de oficina. Trabajo de pregrado. Departamento de Ingeniería Eléctrica, Universidad Central de Venezuela, Caracas - Venezuela.

Boylestad, R y Nashelsky, L. 2003. Electrónica: teoría de circuitos y dispositivos electrónicos. Pearson Educación.

Castells M. 2001. La Galaxia Internet. Reflexiones sobre Internet, empresa y sociedad. Madrid.

Fowler, M. y Cols. 1999. UML Gota a Gota. México: Pearson Educación

Dulhoste, J 2010. Fundamento de medición. Universidad de los Andes. Mérida, Venezuela.

Fraden, J. 2004. Handbook of modern sensors: physics, designs, and applications. (3ª ed.). Springer-Verlag, New York.

Gayo, D. (2000). Diseño gráfico de páginas Web. Microsiervos. Obtenido el 08 de marzo de 2015 de http://www.microsiervos.com/archivo/diseno/estilos-css-tablas.html

Gómez, C. 2012. Desarrollo de un sistema de control de climatización de un edificio, basado en un sistema de inteligencia ambiental y dispositivos móviles. Trabajo de fin de

carrera. Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, España.

Gómez, J. 2007. Fundamentos de la Metodología RUP (RationalUnifiedProcess). Ing Jorge Galves. Universidad Tecnológica de Pereira.

Gonzáles, E. 2005. Nagios: "herramienta libre para la monitorización de sistemas".<http://www.eghost.deusto.es/docs/2005/cursos/charlaNagios_20050224.pdf> (28/03/2016)

Knowlton, J. (2009). Python. Traducido por: Fernández Vélez, María Jesús (1 edición). Anaya Multimedia-Anaya Interactiva

Lopez, A. 2010. Seguridad Informatica. Editorial Editex.

Montilva, J. (2007). Desarrollo de software empresarial. Mérida, Venezuela.

Ortega, E. 2014. Slideshare. Obtenido de es.slideshare.net/clasificacion-de-sensores-36167798

Pressman, R. (2005). Ingeniería del software. Un enfoque práctico. Sexta edición. McGraw-Hill/Interamericana Editores S.A. de C.V., México.

Mendoza, J. 2012, Desarrollo de sensores basado en entropía y la entropía aproximada para la implementación de control supervisor en cierta clase de sistema dinámico hibrido.

Real Academia Española (RAE) 2001. "Diccionario de la lengua española". <http://www.rae.es/obras-academicas/diccionarios/diccionario-de-la-lengua-espanola> (01/06/2014)

Sabino C. 2007 "El Proceso de Investigación". Editorial Panapo de Venezuela, Caracas.

Rivera, O (2005). "Monitoreo e indicadores". <http://www.oei.es/idie/mONITOREOEINDICADORES.pdf> (30/05/2014)

Roman, A. Automatización industrial. Universidad Nacional de San Antonio Abad del Cusco, Perú.

Sánchez. E. 2012. Diseño de un sistema de control basado en la plataforma Arduino. Departamento de ingeniería técnica en informática de sistemas. Universidad Politécnica de Valencia, España.

Salvador, J y cols. 2003. Ingeniería de proyectos informáticos: actividades y procedimientos. Editorial Universitat Jaume I

Schmuller, J. 2002. Aprenda UML en 24 horas. Prentince/Hall.

Tanenbaum, 2003 “Redes de Computadoras”. 4º Edición. Pearson Education, México.

Torres y col. (2012). “Red de monitoreo ambiental bajo un contexto urbano“. <http://www.revistatlaloc.org.mx/amh_congreso/articulos/CambioClimaticoyEventosExtremos/094art_tma1.pdf> (25/05/2014)

Arduino. (Septiembre, 2010). Página oficial de Arduino. <http://www.arduino.cc>. (5/05/2014)

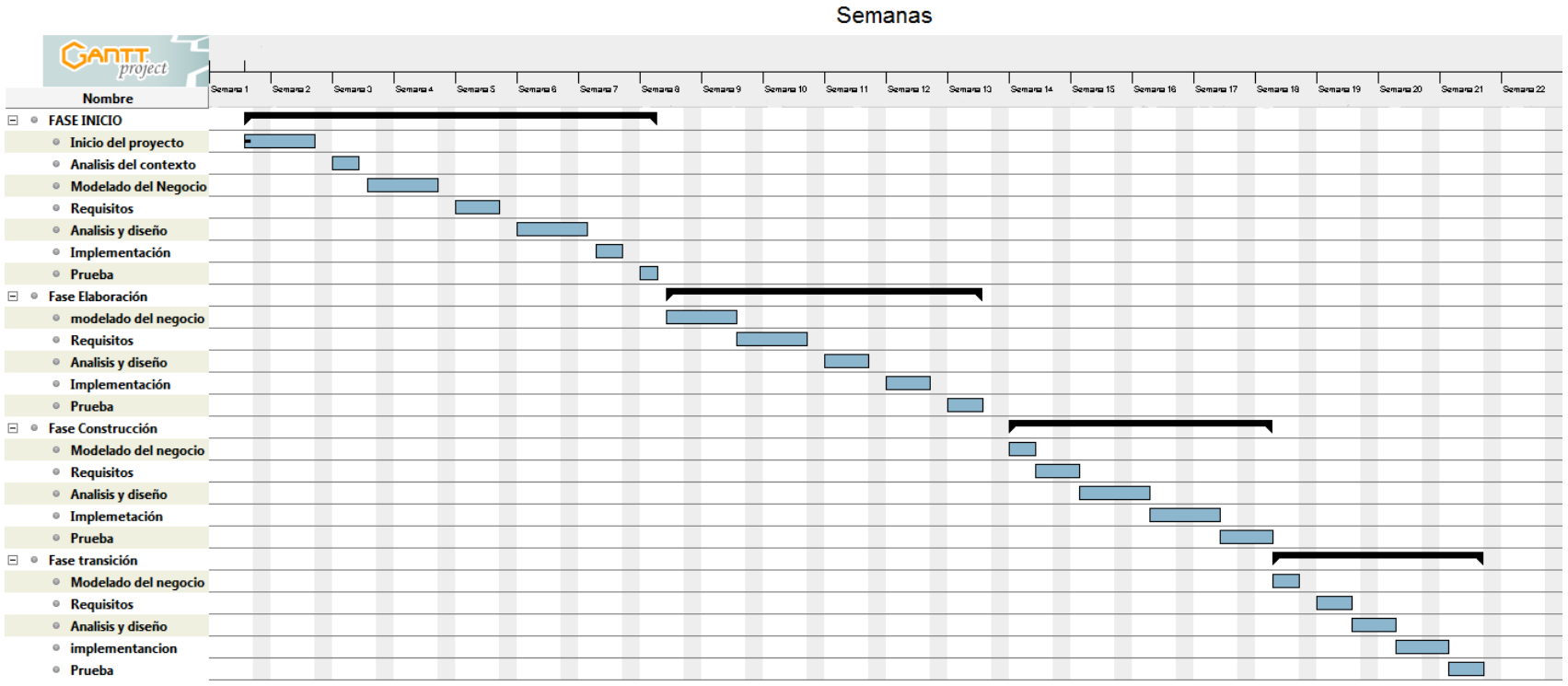
Nagios (Julio, 2009). Página oficial de Nagios. <http://www.Nagios.org>. (10/05/2014)
- http://www.Nagios.org/documentation

Nagios Plugins Development Team. (2009). Nagios plug-in development guidelines. Recuperado en febrero de 2016, de http://Nagiosplug.sourceforge.net/developer-guidelines.html

APÉNDICE

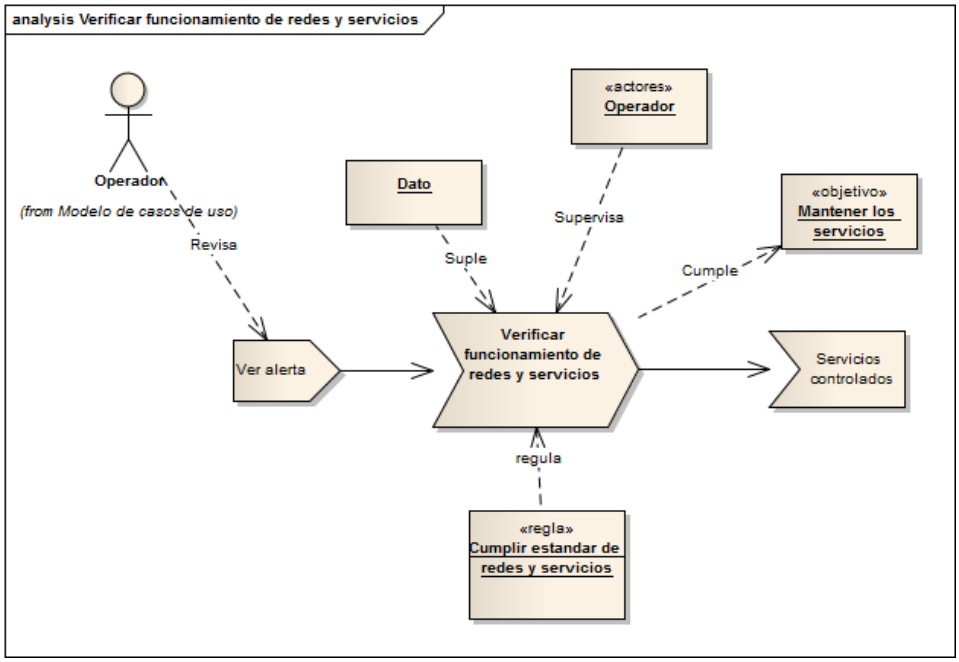
ÍNDICE

Apéndice A:

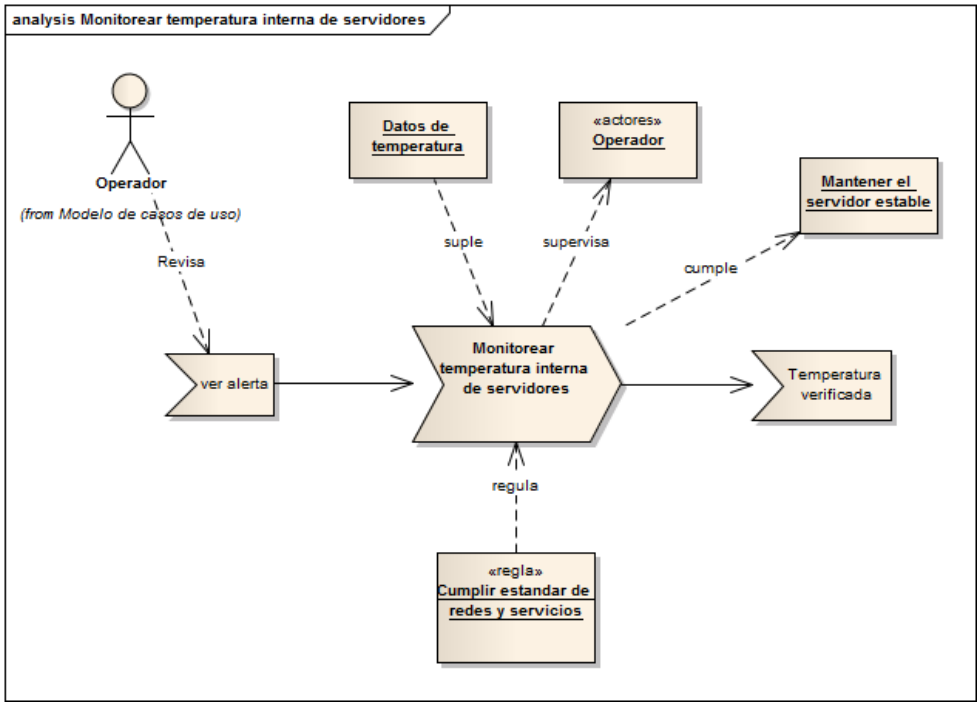


Apéndice A1. Cronograma de Actividades

Apéndice B:



Apéndice B1. Diagrama de proceso del subproceso Verificar funcionamiento de redes y servicios



Apéndice B2. Diagrama de proceso del subproceso monitorear temperatura interna de servidores

Apéndice C**:**

Caso de Uso ID:	01
Nombre:	Recibir notificaciones
Creado Por:	Disbelys Díaz
Fecha de Creación:	09/04/2016
Actores:	Operador y monitor
Descripción:	El operador y monitor reciben notificaciones de la temperatura actual del centro de datos UDO
Pre-condiciones:	Estar registrados
Pos-condiciones:	
Flujo Normal:	El administrador selecciona la opción “Registrar usuario”. Selecciona una de las opciones del menú “enviar notificación” El flujo continúa dependiendo de la opción seleccionada.
Flujos Alternativos	
Reglas de Negocio:	

Apéndice C1. Descripción de casos de uso.

Caso de Uso ID:	02
Nombre:	Visualizar temperatura
Creado Por:	Disbelys Díaz
Fecha de Creación:	09/04/2016
Actores:	Operador y monitor
Descripción:	Permite al operador y monitor visualizar la temperatura en tiempo real del centro de datos UDO
Pre-condiciones:	Estar en la interfaz inicial de la aplicación
Pos-condiciones:	
Flujo Normal:	Estar en la interfaz inicial Visualizar en la temperatura real.
Flujos Alternativos	
Reglas de Negocio:	

Apéndice C2. Descripción de casos de uso.

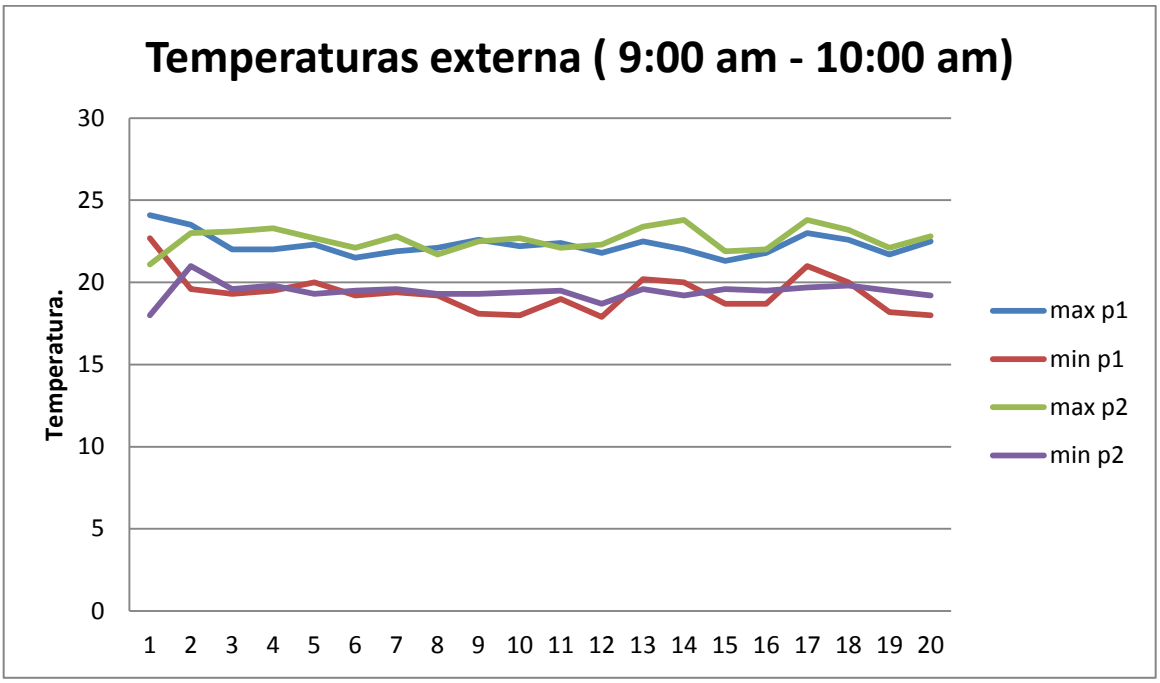
Caso de Uso ID:	03
Nombre:	Visualizar reportes
Creado Por:	Disbelys Díaz
Fecha de Creación:	09/04/2016
Actores:	Operador y monitor
Descripción:	El sistema deberá permitir generar reportes relacionados a las temperaturas obtenidas del sensor de temperatura.
Pre-condiciones:	Estar en la interfaz inicial de la aplicación e iniciar sesión.
Pos-condiciones:	Se genera el reporte.
Flujo Normal:	El caso de uso inicia cuando el usuario selecciona la opción Reportes en el módulo correspondiente. El sistema genera el tipo reporte.
Flujos Alternativos	
Reglas de Negocio:	

Apéndice C3. Descripción de casos de uso.

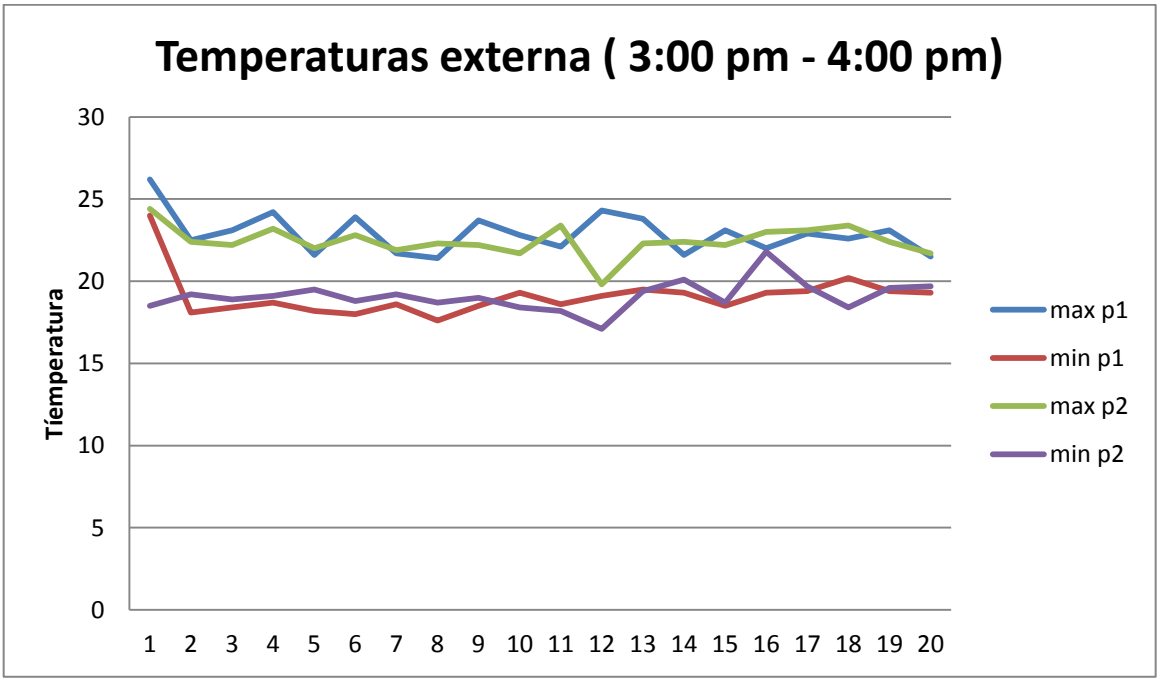
Apéndice D:

TEMPERATURA DE LOS SERVIDORES					
HORARIO	SERVIDOR 1	SERVIDOR 2	SERVIDOR 3	SERVIDOR 4	
9:00 AM - 10:00 AM	59.3	55.5	53.3	54.1	
3:00 PM - 4:00 PM	58.3	55.1	53.5	52.8	DIA 21
9:00 AM - 10:00 AM	58.2	55.3	53.4	54.4	
3:00 PM - 4:00 PM	58.1	55.2	53.6	53	
					DIA 22
9:00 AM - 10:00 AM	58.7	55.6	53.5	54.6	
3:00 PM - 4:00 PM	59.1	56.2	53.5	54.7	
					DIA 23
9:00 AM - 10:00 AM	58	55.2	53.3	55	
3:00 PM - 4:00 PM	59.5	56.6	54.3	54	
					DIA 24
9:00 AM - 10:00 AM	58.5	55.5	53.2	54.2	
3:00 PM - 4:00 PM	58.7	55.4	53.8	54.5	
					DIA 25
9:00 AM - 10:00 AM	58.2	55.8	53.7	53.8	
3:00 PM - 4:00 PM	58.6	55.7	53.9	54.2	
					DIA 26
9:00 AM - 10:00 AM	58.3	57.2	53.8	55.3	
3:00 PM - 4:00 PM	59.1	56.3	53.5	54.6	
					DIA 27
9:00 AM - 10:00 AM	59.5	56.8	55.1	54.3	
3:00 PM - 4:00 PM	60	57	55.2	55.5	
					DIA 28
9:00 AM - 10:00 AM	59.3	56.6	55.2	54.3	
3:00 PM - 4:00 PM	59.2	57.1	55.5	55.9	
					DIA 29
9:00 AM - 10:00 AM	58.4	57.4	53.1	54.2	
3:00 PM - 4:00 PM	58.6	55.3	53.7	54.6	
					DIA 30

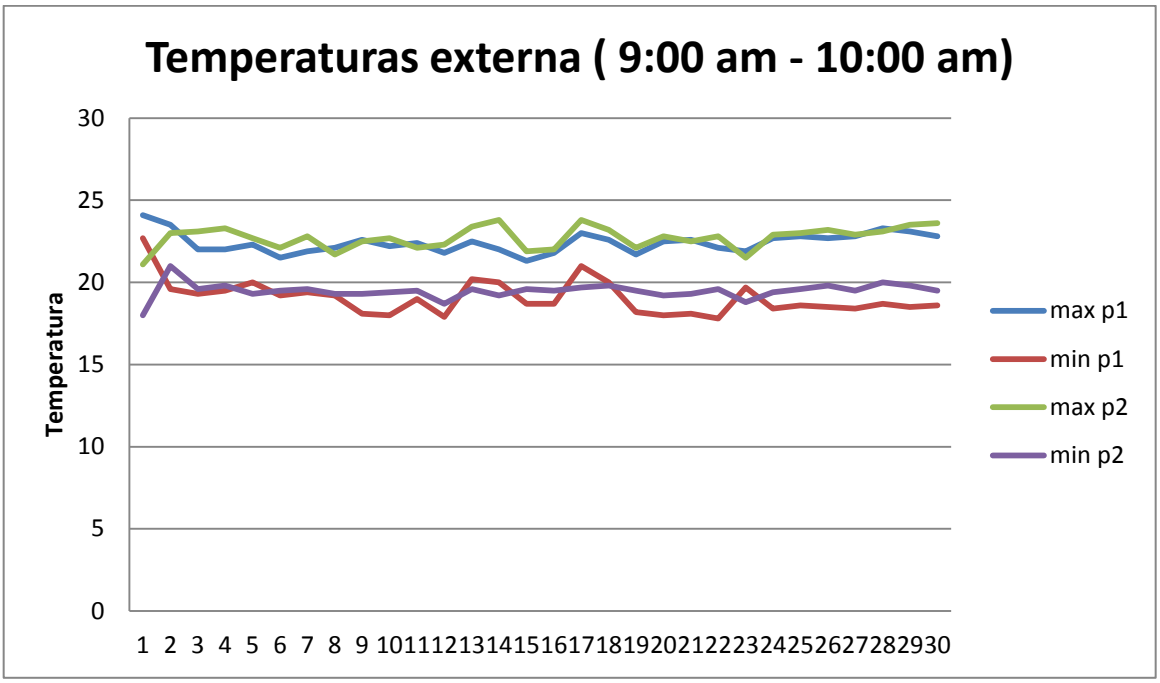
Apéndice D1. Muestreo de temperatura ambiental (externa) y la temperatura de los servidores (interna)



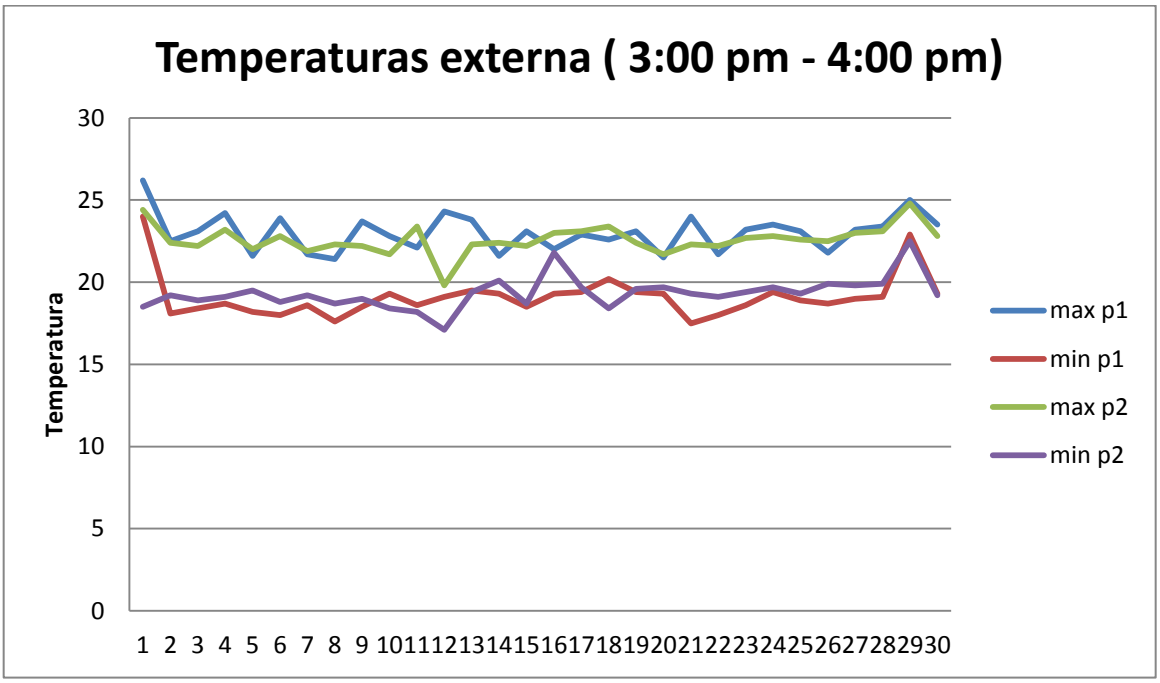
Apéndice D2. Grafica de temperaturas externa del centro de datos (UDO)



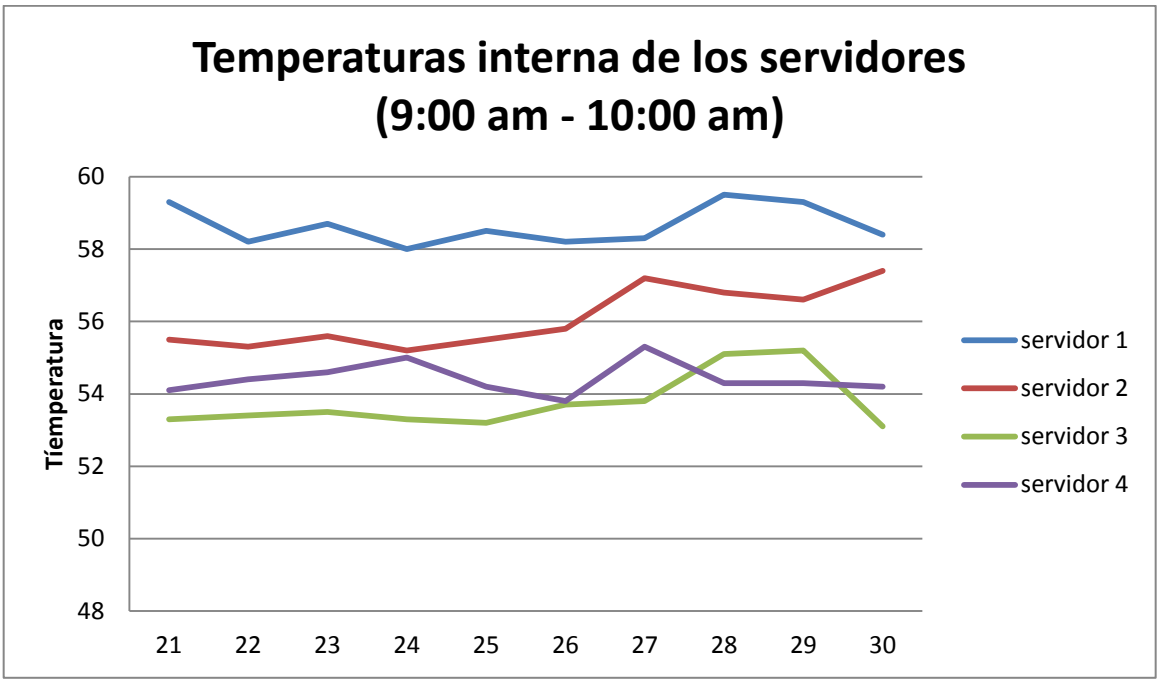
Apéndice D3. Grafica de temperaturas externa del centro de datos (UDO)



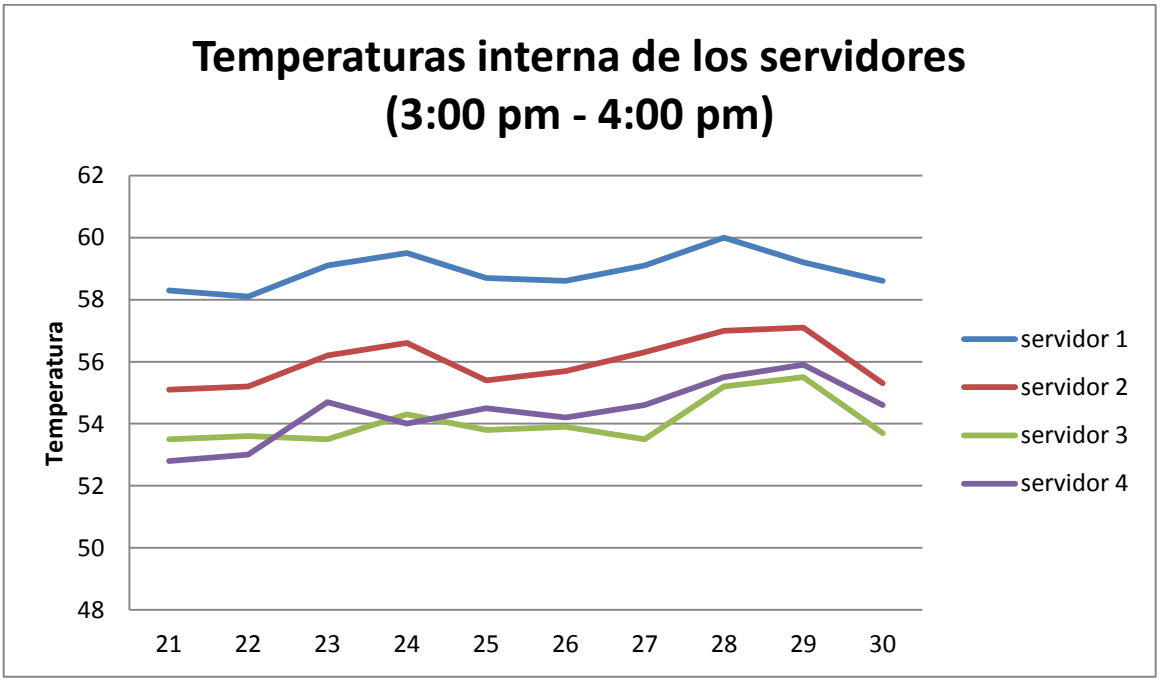
Apéndice D4. Grafica de temperaturas externa del centro de datos (UDO)



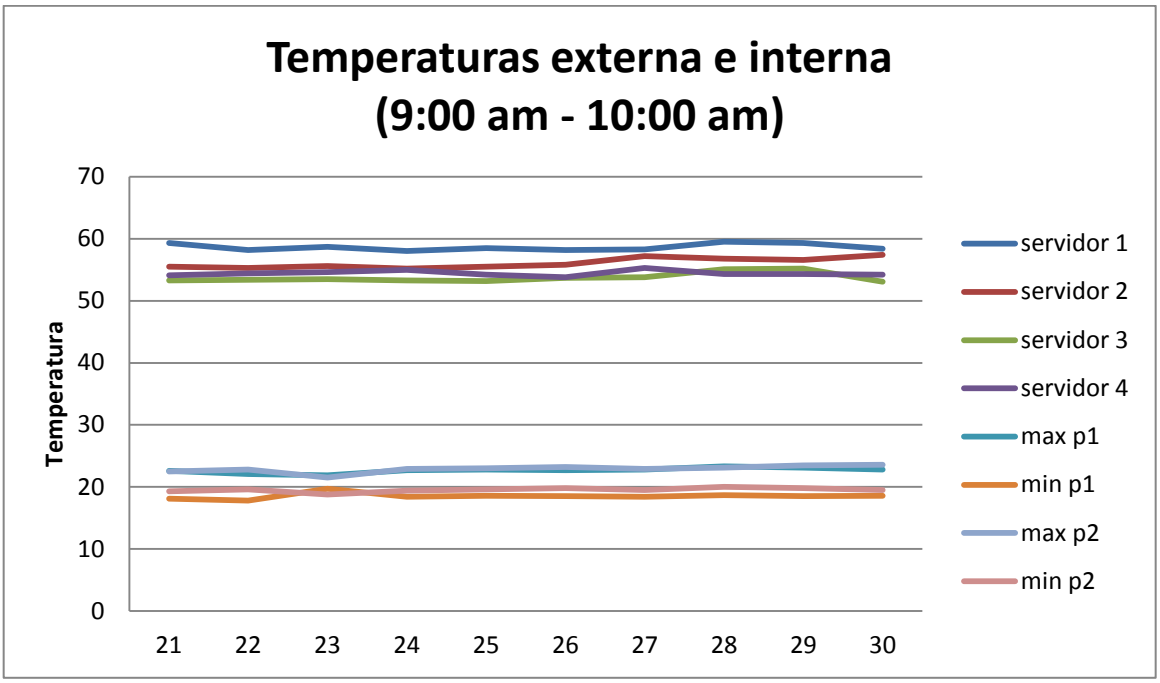
Apéndice D5. Grafica de temperaturas externa del centro de datos (UDO)



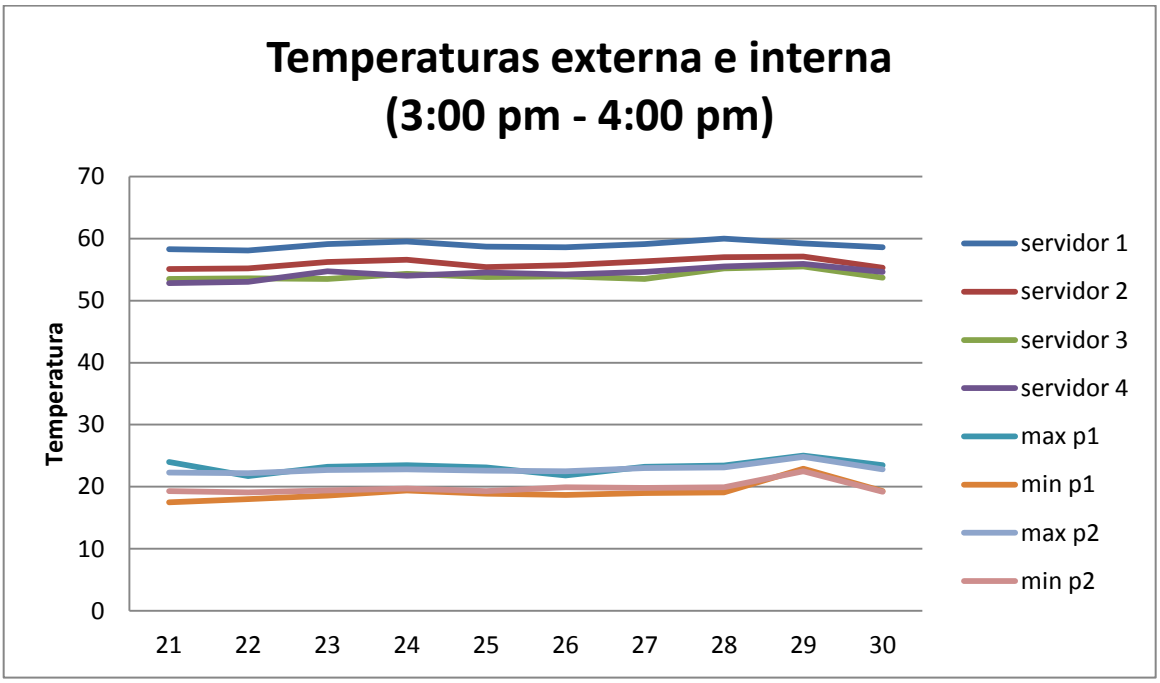
Apéndice D6.Grafica de temperaturas interna de los servidores del Centro de Datos UDO



Apéndice D7. Grafica de temperaturas interna de los servidores del centro de datos (UDO)

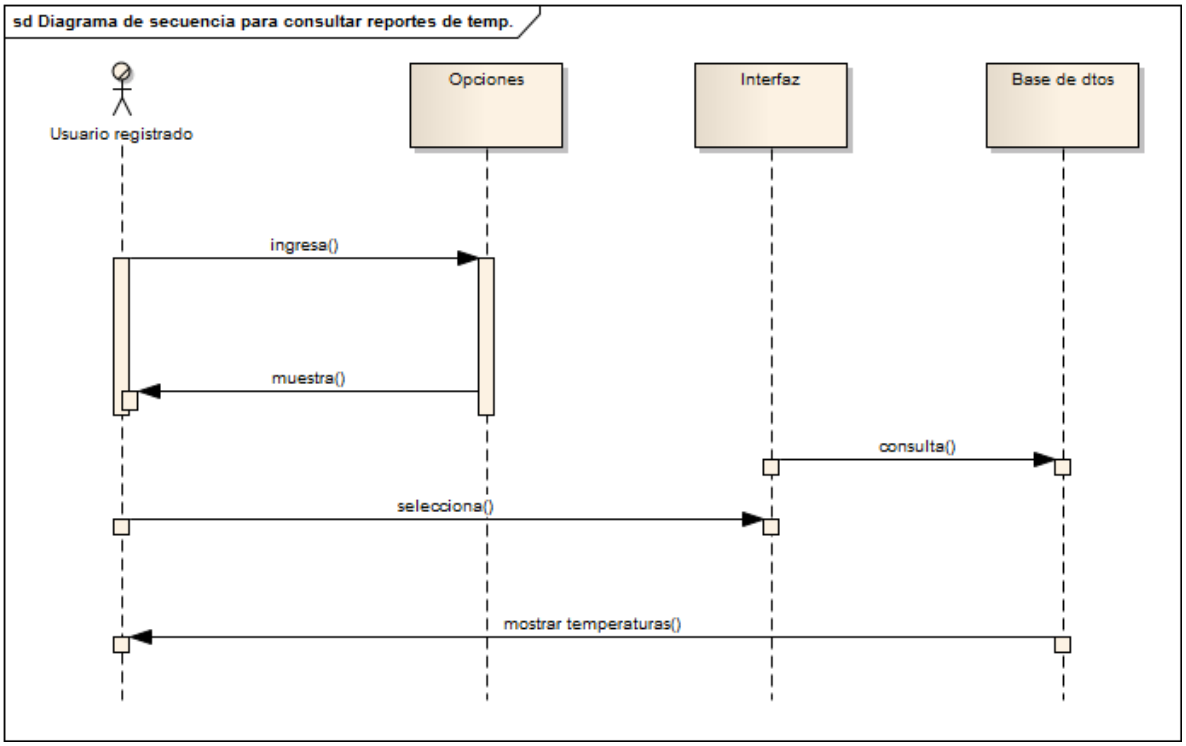


Apéndice D8. Grafica de temperaturas interna e interna del centro de datos (UDO)



Apéndice D9. Grafica de temperaturas interna e interna del centro de datos (UDO)

Apéndice E:



Apéndice E1. Diagrama de secuencia para consultar los reportes de temperatura

HOJAS DE METADATOS

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 1/6

Título	Sistema de monitoreo ambiental basado en arduino y Nagios para el Centro de Datos del rectorado de la Universidad de Oriente
Subtítulo	

Autor(es)

Apellidos y Nombres	**Código CVLAC / e-mail**	
Díaz Millán Disbelys del C.	**CVLAC**	**V_19345371**
	e-mail	**Disbelys10@gmail.com**
	e-mail	
	CVLAC	
	e-mail	
	e-mail	
	CVLAC	
	e-mail	
	e-mail	
	CVLAC	
	e-mail	
	e-mail	

Palabras o frases claves:

monitoreo ambiental, arduino, nagios.

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 2/6

Líneas y sublíneas de investigación:

Área	Subárea
Ciencias	**Departamento de Informática.**

Resumen (abstract):

Se desarrolló un sistema de monitoreo ambiental basado en Arduino (hardware) y Nagios (software) para el Centro de Datos de la Universidad de Oriente (UDO) Venezuela, de bajo costo y adaptable. Primero se analizó las necesidades y requerimientos del Centro de Datos UDO, luego se realizó un estudio de la distribución de temperatura, se determinó cuantos sensores y de qué forma estarían ubicados, después se realizó el script (instrucciones) para la comunicación entre Arduino y Nagios. Para este proyecto se aplicó la metodología Rational Unified Process (RUP, en español Proceso racional unificado), la cual se compone de cuatro (4) fases: inicio, elaboración, construcción y transición. En la fase de inicio se recolectó información para construir la visión del negocio, se recopilaron los requisitos, se plantearon las pruebas, se modelaron los actores y se delimitaron los casos de uso, sobre los cuales se enfocaría el proyecto. En la fase de elaboración se refinaron algunos diagramas, se identificó los equipos necesarios para las actividades de monitoreo, se recolectó información referente a los mismos, se diseñó el script para la comunicación entre Nagios y Arduino, también se diseñó la interfaz web para visualizar los cambios de temperatura y humedad a tiempo real. En la fase de construcción se realizó la integración correcta entre Nagios y Arduino, se configuraron los host (huésped) y servicios para su buen funcionamiento y fue verificada la correcta adquisición de datos.
Al final, se obtuvo el sistema de monitoreo ambiental para el Centro de Datos UDO, el cual capturó datos de temperatura y humedad, para verificar la estabilidad ambiental del mismo, con el fin de corregir problemas y tomar acciones preventivas frente a los diferentes riesgos que se puedan presentar. A este sistema se puede acceder a través de cualquier dispositivo electrónico con acceso a internet.
El sistema de monitoreo ambiental tiene la cualidad de ser configurable y escalable a medida, asegurando como consecuencia la continuidad de los equipos informáticos del Centro de Datos UDO.

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 3/6

Contribuidores:

Apellidos y Nombres	ROL / Código CVLAC / e-mail	
Mendoza José	**ROL**	**CA** [ ] **AS** [x] **TU** [ ] **JU** [ ]
	CVLAC	**V_8639292**
	e-mail	**jamendoza@udo.edu.ve**
Romero José	**ROL**	**CA** [ ] **AS** [ ] **TU** [ ] **JU** [x]
	CVLAC	**V_13631597**
	e-mail	**jfromeropino@gmail.com**
Rodríguez Ramos Carmelys Josefina	**ROL**	**CA** [ ] **AS** [ ] **TU** [ ] **JU** [x]
	CVLAC	**V_13539531**
	e-mail	**carmelysrodriguez@gmail.com**

Fecha de discusión y aprobación:

Año	Mes	Día
2017	**03**	**10**

Lenguaje: **SPA**

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 4/6

Archivo(s):

Nombre de archivo	Tipo MIME
Tesis_DD.doc	**Aplication/word**

Alcance:

Espacial: Universal **(Opcional)**

Temporal: Intemporal **(Opcional)**

Título o Grado asociado con el trabajo: Licenciado en Informática

Nivel Asociado con el Trabajo: Licenciado

Área de Estudio: Ciencias - Informática

Institución(es) que garantiza(n) el Título o grado: Universidad de Oriente

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 5/6

UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

CU Nº0975

Cumaná, 04 AGO 2009

Ciudadano
Prof. JESÚS MARTÍNEZ YÉPEZ
Vicerrector Académico
Universidad de Oriente
Su Despacho

Estimado Profesor Martínez:

Cumplo en notificarle que el Consejo Universitario, en Reunión Ordinaria celebrada en Centro de Convenciones de Cantaura, los días 28 y 29 de julio de 2009, conoció el punto de agenda ***"SOLICITUD DE AUTORIZACIÓN PARA PUBLICAR TODA LA PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE ORIENTE EN EL REPOSITORIO INSTITUCIONAL DE LA UDO, SEGÚN VRAC Nº 696/2009".***

Leído el oficio SIBI – 139/2009 de fecha 09-07-2009, suscrita por el Dr. Abul K. Bashirullah, Director de Bibliotecas, este Cuerpo Colegiado decidió, por unanimidad, autorizar la publicación de toda la producción intelectual de la Universidad de Oriente en el Repositorio en cuestión.

Comunicación que hago a usted a los fines consiguientes.

UNIVERSIDAD DE ORIENTE
SISTEMA DE BIBLIOTECA
RECIBIDO POR
FECHA 5/8/09 HORA 5:20

Cordialmente,

JUAN A. BOLAÑOS CURIEL
Secretario

REPÚBLICA BOLIVARIANA DE VENEZUELA * UNIVERSIDAD DE ORIENTE * SECRETARÍA CONSEJO UNIVERSITARIO

C.C: *Rectora, Vicerrectora Administrativa, Decanos de los Núcleos, Coordinador General de Administración, Director de Personal, Dirección de Finanzas, Dirección de Presupuesto, Contraloría Interna, Consultoría Jurídica, Director de Bibliotecas, Dirección de Publicaciones, Dirección de Computación, Coordinación de Teleinformática, Coordinación General de Postgrado.*

JABC/YGC/maruja

Hoja de Metadatos para Tesis y Trabajos de Ascenso- 6/6

Artículo 41 del REGLAMENTO DE TRABAJO DE PREGRADO (vigente a partir del II Semestre 2009, según comunicación CU-034-2009) : "los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario para su autorización".

Disbelys Diaz

Autor

Prof: José Mendoza

Asesor