



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

TEMA:

**CONTROL Y MONITOREO DE LA RED DE DATOS, INTERNET Y
CLIMATIZACIÓN DE LA CENTRAL DE DATOS DEL HOSPITAL MUNICIPAL
“NUESTRA SEÑORA DE LA MERCED”.**

Trabajo de Graduación. Modalidad: Proyecto de investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

SUBLINEA DE INVESTIGACIÓN: Administración de redes

AUTOR: Alex Francisco Rodríguez Ortiz

PROFESOR REVISOR: Ing. Patricio Córdova, Mg.

Ambato – Ecuador

Julio de 2015

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

“CONTROL Y MONITOREO DE LA RED DE DATOS, INTERNET Y CLIMATIZACIÓN DE LA CENTRAL DE DATOS DEL HOSPITAL MUNICIPAL NUESTRA SEÑORA DE LA MERCED”, del señor Alex Francisco Rodríguez Ortiz, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, Julio de 2015

EL TUTOR

Ing. Patricio Córdova, Mg.

AUTORIA

El presente proyecto de investigación titulado: **“CONTROL Y MONITOREO DE LA RED DE DATOS, INTERNET Y CLIMATIZACIÓN DE LA CENTRAL DE DATOS DEL HOSPITAL MUNICIPAL NUESTRA SEÑORA DE LA MERCED”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Julio de 2015

Alex Francisco Rodríguez Ortiz
C.C.: 180479050-7

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la universidad.

Ambato, Julio de 2015

Alex Francisco Rodríguez Ortiz
C.C.: 180479050-7

APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión Calificadora del presente trabajo conformada por los señores docentes, Ing. Mg. Santiago Altamirano e Ing. Mg. Edwin Morales revisó y aprobó el Informe Final del trabajo de graduación titulado: **“CONTROL Y MONITOREO DE LA RED DE DATOS, INTERNET Y CLIMATIZACIÓN DE LA CENTRAL DE DATOS DEL HOSPITAL MUNICIPAL NUESTRA SEÑORA DE LA MERCED”**, presentado por el señor Alex Francisco Rodríguez Ortiz de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, Julio de 2015

Ing. Vicente Morales L., Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Mg. Santiago Altamirano
DOCENTE CALIFICADOR

Ing. Mg. Edwin Morales
DOCENTE CALIFICADOR

DEDICATORIA

Dedico este Proyecto de Titulación a mis padres Esthela del Roció Ortiz Vélez y Julio Heriberto Rodríguez Benítez quienes me han entregado su apoyo, cariño y comprensión en todo momento a lo largo de mi vida en especial para el cumplimiento de este objetivo ayudándome con sus consejos a levantarme tras cada caída y a luchar con más fuerza.

Alex Francisco Rodríguez Ortiz

AGRADECIMIENTO

Agradezco, a Dios por guiar mi camino y por darme fortaleza a lo largo de todos estos años, a mis padres y mi hermano por brindarme el apoyo incondicional en todo momento, a la Universidad Técnica de Ambato que aportó a mi formación profesional y personal, a mi Tutor Ing. Patricio Córdova por su valiosa colaboración para la culminación del presente Proyecto, al Hospital Municipal Nuestra Señora de la Merced en especial al Ing. Diego Rubio por su generosa ayuda y finalmente gracias a los docentes y amigos que contribuyeron con su asesoría durante el desarrollo del presente Proyecto.

Alex Francisco Rodríguez Ortiz

ÍNDICE DE CONTENIDOS

PÁGINAS PRELIMINARES

PORTADA	I
APROBACIÓN DEL TUTOR	II
AUTORÍA.....	III
DERECHOS DE AUTOR	IV
APROBACIÓN COMISIÓN CALIFICADORA.....	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
INDICE DE CONTENIDOS	VIII
INDICE DE TABLAS	XI
INDICE DE GRÁFICOS.....	XII
RESUMEN	XIV
SUMMARY.....	XV
LISTADO DE TERMINOS	XVI
INTRODUCCIÓN.....	XVII

PROYECTO

CAPÍTULO 1	1
El problema	1
1.1 Tema de investigación:	1
1.2 Planteamiento del problema.....	1
1.3 Delimitación.....	2
1.4 Justificación.....	2
1.5 Objetivos	3
1.5.1 General	3
1.5.2 Específicos.....	3
CAPÍTULO 2	4
Marco Teórico.....	4
2.1 Antecedentes Investigativos.....	4
2.2 Fundamentación teórica	5

2.2.1 Data Center (Central de Datos).....	5
2.2.2 Técnicas de monitoreo:	7
2.2.3 Topología del sistema de monitoreo	10
2.2.4 Sistema Operativo Movil	12
2.2.5 Sistema operativo Android.....	14
2.3 Propuesta de Solución	16
CAPÍTULO 3	17
3.1 Modalidad de la Investigación	17
3.2 Población y Muestra.....	17
3.3 Recolección de Información	17
3.4 Procesamiento y Análisis de Datos	18
CAPÍTULO 4	19
4.1 Levantamiento de la información.....	19
4.2 Identificación de los dispositivos a ser controlados y monitoreados.	20
4.3 Determinación de las formas de configuración, rendimiento, seguridad y manejo de fallas, mediante un uso adecuado de los protocolos y recursos que presentan la red interna.	23
4.4 Determinar el método y la alternativa de solución encontrada mediante la investigación bibliográfica.	25
4.4.1 Administración y Gestión de Red	25
4.4.2 Protocolo SNMP	31
4.4.3 Herramientas de Desarrollo para Android	33
4.4.4 Eclipse	36
4.4.5 Android	39
4.4.6 Micro Controlador.....	45
4.4.7 Elección de Arduino y su modelo	47
4.4.8 Modulo Ethernet.....	50
4.4.9 Pantalla de Cristal Líquido (LCD)	53
4.4.10 Sensor LM35	54
4.5 Diseño e implementación de un circuito para el monitoreo y control de la climatización interna de la Central de Datos.....	56
4.6 Diseño e implementación del software sobre el sistema operativo Android para conocer el estado operativo y funcional del Data Center a través del protocolo SNMP	70

4.7 Realización de pruebas del sistema de control y monitoreo de la Central de Datos diseñado.....	84
4.8 Evaluación de la propuesta por el método de experto.....	88
4.9 PRESUPUESTO	89
CAPÍTULO 5	91
5.1 Conclusiones	91
5.2 Recomendaciones:.....	92
Referencias	93
Anexos.....	95

INDICE DE TABLAS

Tabla 2.1 Tipos de Tiers	6
Tabla 2.2 Detalles Básicos de los Sistemas Operativos Móviles	14
Tabla 4.1 Tabla comparativa microcontroladores.	47
Tabla 4.2 Tabla comparativa arduino.	48
Tabla 4.3 Resumen Características Arduino Mega 2560	48
Tabla 4.4 Resumen Arduino Ethernet.....	51
Tabla 4.5 Presupuesto general.	89
Tabla 4.6 Presupuesto hardware para control y monitoreo de la climatización.	90

INDICE DE GRÁFICOS

Figura 2.1: División de los elementos que pueden ser monitoreados.....	9
Figura 2.2: Solicitud mediante SNMP	10
Figura 2.3: Envío de traps	10
Figura 2.4: Monitoreo basado en el protocolo SNMP-CACTI.....	11
Figura 2.5: Gráficos en modo árbol (Router GW-UCPETP – RIMED).....	11
Figura 2.6: Aceptación en el mercado-Android.....	15
Figura 2.7: Aspecto fichero descargado.....	15
Figura 2.8: Instalación en computador personal (2014)-1	16
Figura 4.1: Router de Balanceo TP Link TL-R480T	21
Figura 4.2: Switch Cisco Catalyst 3560-C.....	21
Figura 4.3: Servidor Microtower HP	22
Figura 4.4: Diagrama Simplificado de red.....	23
Figura 4.5 Topología de Red	24
Figura 4.6: Modelos de gestión de red.....	25
Figura 4.7: Arquitectura de un Sistema de Administración.....	26
Figura 4.8: Parte de árbol de objetos según SMiv1	28
Figura 4.9: Parte de árbol de objetos según SMiv2	29
Figura 4.10: Subárbol MIB II	31
Figura 4.11: Arquitectura TCP/IP y SNMP	32
Figura 4.12: Creación del Workspace.....	37
Figura 4.13: Pluggin ADT	38
Figura 4.14: Opciones de pluggin ADT.....	38
Figura 4.15: Árbol de prioridad de finalización de los procesos	43
Figura 4.16: Vista de LogCat.....	44
Figura 4.17: Arduino Ethernet Placa vista frontal - visión trasera	50
Figura 4.18: Pantalla de Cristal Líquido – LCD 20x4[12]	53
Figura 4.19: Descripción de pines LCD	54
Figura 4.20: Diagrama de conexión sensor LM35	55
Figura 4.21 Diagrama esquemático hardware climatización.....	57
Figura 4.22: Diagrama pictórico hardware climatización	58
Figura 4.23: Diseño del sitio web basado en HTML5, CSS y Ajax	64
Figura 4.24: Medidor de temperatura operando	65
Figura 4.25: Login al sistema de control y monitoreo	65
Figura 4.26: Medidor de temperatura funcionando con swtich de apagado / encendido	65
Figura 4.27: Login Erroneo (Mensaje)	65
Figura 4.28: Creación de la cuenta en Twitter	66
Figura 4.29: Autorización del uso de la cuenta en Twitter	66
Figura 4.30: Autorización del uso de la cuenta en Twitter	67
Figura 4.31: Servicio de eliminación de Tweets.....	68
Figura 4.32: Confirmación del Servicio de eliminación de Tweets	68
Figura 4.33: Vista Conexión de cableado interno superior	69
Figura 4.34: Ensamblaje del chasis culminado.....	69
Figura 4.35 Topología de red.....	71

Figura 4.36: Diagrama de flujo software Android (Parte1).....	72
Figura 4.37: Diagrama de flujo software Android (Parte2).....	73
Figura 4.38: Conexión a través de Putty por el protocolo SSH.....	73
Figura 4.39: Login del usuario.....	74
Figura 4.40: Ingreso de contraseña y mensajes de inicio	74
Figura 4.41: verificación de versión instalada	74
Figura 4.42: Instalación del paquete SNMP	75
Figura 4.43: Configuración de la comunidad para el servicio SNMP	75
Figura 4.44: iniciar el servicio SNMPD	76
Figura 4.45: Verificación servicio SNMP	76
Figura 4.46: Verificación el servicio SNMPD.....	76
Figura 4.47: Configuración protocolo SNMP.....	77
Figura 4.48: Login App Sistema de Control y Monitoreo	78
Figura 4.49: Menú de administración	78
Figura 4.50: Gestor SNMP	79
Figura 4.51: Interfaz de Consulta Interfaz MIB	80
Figura 4.52: Consulta Sistema MIB	81
Figura 4.53: Conectividad Dispositivos (Ping).....	82
Figura 4.54: Consulta Estado de la Red WIFI	83
Figura 4.55: Activación notificaciones de alerta	83
Figura 4.56 Establecer Parametros	84
Figura 4.57: Conexión al servidor	85
Figura 4.58: Interfaz MIB total de interfaces	85
Figura 4.59: Interfaz MIB - Descripción general	86
Figura 4.60: Verificación Descripción general Interfaz MIB.....	87

RESUMEN

El Data Center del Hospital Municipal Nuestra Señora de la Merced alberga toda la estructura neurálgica y tecnológica para el funcionamiento administrativo, técnico y operativo del Hospital, por lo cual es necesario velar por su marcha adecuada, siendo necesario implementar mecanismos de monitoreo y control.

En base a los requerimientos del administrador del área de Sistemas y Comunicaciones, se pudo determinar la necesidad de monitorear y controlar los equipos Core que son la base de toda la red informática del Hospital, en tal virtud previo un análisis y sustentación teórico – científica se determinó que el protocolo SNMP y el Control de la climatización son los mecanismos necesarios para proporcionar información del estado del Data Center para la toma de decisiones en el caso de presentarse una falla lógica o física a través del sistema operativo android, al obtener información como la temperatura al interior del Data Center, caída o falencias en uno de los dispositivos físicos de la red, alertando inmediatamente al usuario sobre el suceso ocurrido y permitiéndole resolver el inconveniente generado.

SUMMARY

The “Municipal Hospital Nuestra Señora de la Merced” Data Center have all the neuralgic and technological structure for administrative , technical and operational functioning of the Hospital , so it is necessary to ensure proper running , being necessary to implement mechanisms for monitoring and control.

Based on the requirements of the administrator area of Systems and Communications, it was determined the need to monitor and control the Core teams that are the source of all computer network Hospital, in such prior under analysis and theoretical support - Scientific is determined that the SNMP protocol and climate control are the necessary mechanisms to provide status information data Center for decision making in the event of a logical or physical failure through the operating system android, the information as temperature inside the Data Center, fall or shortcomings in one physical network devices , immediately alerting the user about the event occurred and allowing solve the problem generated.

LISTADO DE TÉRMINOS

ADT Android Development Tools

API Application programming interface

APK Android Package

ASN.1 Abstract Syntax Notation One

DDL Data definition language

DDMS Dalvik Debug Monitor Server

ICMP Internet Control Message Protocol

IDE Integrated Development Environment

ISO/OSI The International Standard Organization's /Open System Interconnect

JDK Java Development Kit

JRE Java Runtime Environment

MAC Media Access Control address

MD5 Message-Digest Algorithm

MIB Management Information Base

MPv1 Message processing model for SNMPv1

MPv2c Message processing model for SNMPv2c

NMA Network Management Application

OID Object Identifier

PDU Protocol data unit

RFC Request for Comments

SDK Software development kit

SGMP Simple Gateway Monitoring Protocol

SMI Structure of Management Information

SNMP Simple Network Management Protocol

SNMP4J SNMP for Java

TCP/IP Transmission Control Protocol/Internet Protocol

TMN Telecommunications Management Network

UDP User Datagram Protocol

VM Virtual Machine

XML eXtensible Markup Language

INTRODUCCIÓN

El presente trabajo estructurado de manera independiente denominado: “CONTROL Y MONITOREO DE LA RED DE DATOS, INTERNET Y CLIMATIZACIÓN DE LA CENTRAL DE DATOS DEL HOSPITAL MUNICIPAL “NUESTRA SEÑORA DE LA MERCED” para un mejor entendimiento, se lo ha dividido en los siguientes capítulos:

CAPÍTULO I denominado “**EL PROBLEMA**”, identifica el problema de la Central de Datos del Hospital Municipal Nuestra Señora de la Merced, una justificación de la implementación de un sistema de control, además del establecimiento de los objetivos que plantean opciones para mejorar la gestión y control del administrador de la red.

CAPÍTULO II denominado “**MARCO TEÓRICO**”, muestra las investigaciones previas que sirven de soporte a la investigación, donde se desglosa información técnica y relevante sobre lo que es un Data Center, Control y Monitoreo de una red, Protocolo SNMP, Climatización y el Sistema Operativo Android, apoyado además de la información de estudios similares anteriormente realizados, además de los aspectos conceptuales que sustentan el tema en general y el conjunto de conceptos y fundamentos teóricos que han sido analizados en base al problema establecido.

CAPÍTULO III denominado “**METODOLOGÍA**”, define el tipo de investigación que ha sido desarrollada, el tratamiento de los procesos que señala la modalidad de investigación, asimismo se presenta el tipo de análisis de los datos según el tipo de investigación para el desarrollo del proyecto.

CAPÍTULO IV denominado “**DESARROLLO DE LA PROPUESTA**”, establece los requerimientos presentados según el análisis de las áreas críticas definidas, el desarrollo de un modelo de control y monitoreo basado en estándares internacionales, donde se concluye que el protocolo SNMP es el indicado para realizar las labores de control y monitoreo basados en el sistema operativo móvil Android.

CAPÍTULO V denominado “**CONCLUSIONES Y RECOMENDACIONES**”, expone de forma clara y concisa las consideraciones más relevantes que se han obtenido al finalizar el proyecto, además se indican recomendaciones que servirán de apoyo para el desarrollo del mismo.

CAPÍTULO 1

El problema

1.1 Tema de investigación:

CONTROL Y MONITOREO DE LA RED DE DATOS, INTERNET Y CLIMATIZACIÓN DE LA CENTRAL DE DATOS DEL HOSPITAL MUNICIPAL “NUESTRA SEÑORA DE LA MERCED”.

1.2 Planteamiento del problema

El control remoto de los sistemas eléctricos y redes de datos comenzó en la década de los años 60, y durante la década de los años 70 el reemplazo de los dispositivos electromecánicos por equipos de estado sólido que en su momento se encontraba bien avanzado.

Aún hoy, existen muchas redes que todavía no se han integrado totalmente a la tecnología de los dispositivos de control y monitoreo, en parte debido a que la interoperabilidad entre los dispositivos está obstaculizada por el exceso de protocolos e interfaces incompatibles [1].

“La detección oportuna de fallas y el monitoreo de los elementos que conforman una red de computadoras son actividades de gran relevancia para brindar un buen servicio a los usuarios. De esto se deriva la importancia de contar con un esquema capaz de notificarnos las fallas en la red y de mostrarnos su comportamiento mediante el análisis y recolección de tráfico”.

Hasta hace poco hablar de sistemas de monitoreo de servicios de red en sistemas operativos de red o desktop resultaba casi imposible, ya que no se contaba con las herramientas necesarias para hacerlo. Bastaba con saber que el servidor estaba operativo; y se realizaba solo con ejecutar el comando ping y la dirección IP (Protocolo de Internet) del servidor [1].

Según publicaciones de la empresa de seguridad informática GMS (Sistema de Gestión de Gráfica) (2013), A nivel de nuestro país el monitoreo de redes se encuentra en

constante crecimiento debido a la importancia que tiene el buen desempeño y desarrollo de una red para la empresa.

La gran mayoría de empresas que manejan procesos informáticos cuentan en su departamento de sistemas o comunicaciones con sistemas de detección de fallas y monitoreo 24/7 para su red interna, para prevenir y conocer posibles caídas en la red interna o en su red de acceso a internet, debido a la importancia que supone las comunicaciones, pudiendo ejemplificar las redes de bancos, instituciones financieras, empresas de telecomunicaciones, etc., las cuales envían informes del estado general de la red cada cierto tiempo, permitiendo a los administradores tomar las decisiones pertinentes ante fallas inminentes.

El Hospital Municipal “Nuestra Señora de la Merced” no posee un sistema de control ni monitoreo del estado general de su Data Center, por lo cual se han presentado cortes e interrupciones en el servicio de forma abrupta al no conocerse o tener herramientas que permitan conocer el estado en tiempo real de la red.

1.3 Delimitación

Área académica: Programación y Redes

Línea de investigación: Programación y Redes

Sublíneas de investigación: Administración de Redes.

Delimitación espacial: La investigación se realizará en la Ciudad de Ambato en el Hospital Municipal “Nuestra Señora de la Merced” ubicado en el barrio Letamendi de la ciudad de Ambato.

Delimitación temporal: El proyecto de investigación se desarrollará en un plazo de seis meses a partir de la aprobación del Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.4 Justificación

La importancia de la investigación se fundamenta en permitir al personal de Sistemas del Hospital Municipal “Nuestra Señora de la Merced” mantener un monitoreo permanente del funcionamiento de la red de datos, internet, back up eléctrico y un control de la climatización, servicios de red de datos en el Data Center del hospital, manteniendo una conexión a través de dispositivos móviles conectados a través de IP públicas y los respectivos API's (Interfaz de Programación de Aplicaciones) al hardware existente, proporcionando un conocimiento en tiempo real de lo que sucede, permitiendo tomar decisiones oportunas y rápidas en caso de presentarse alguna anomalía y tomar las medidas correspondientes para solventar la eventualidad.

El protocolo SNMP (Protocolo Simple de Administración de Red) permitirá conocer en rasgos generales el funcionamiento y operatividad de una red, en este caso gracias a este protocolo se podrá conocer si los dispositivos principales de la red, entre los que se puede mencionar servidor de correo, firewall, servidor de base de datos, servidor de aplicaciones, Router Core y Switch Core se encuentran operando de forma eficiente o si presentan fallas, con lo cual se puede tomar decisiones de manera inmediata o iniciar una planificación para el mantenimiento, mejora o cambio de la red.

El Monitoreo de red describe el uso de un sistema que constantemente monitoriza una red de datos en busca de componentes defectuosos o lentos, para luego informar a los administradores de redes mediante correo electrónico, pager u otras alarmas.

Para monitorear continuamente se necesita un software de monitoreo. En el mercado hay diversas aplicaciones para monitorear una red. Todos tienen en común que son muy costosos y difíciles de implementar.

1.5 Objetivos

1.5.1 General:

- Implementar un sistema de control y monitoreo de la red de datos, internet y climatización de la central de datos del Hospital Municipal “Nuestra Señora de la Merced”.

1.5.2 Específicos:

- Analizar el estado de funcionamiento y características de la Central de Datos actual.
- Diseñar una aplicación para dispositivos móviles bajo la plataforma Android para el control y monitoreo de la red de datos, internet y climatización de la Central de Datos usando el protocolo SNMP.
- Implementar un circuito para el control y monitoreo de la climatización interna de la Central de Datos del Hospital Municipal “Nuestra Señora de la Merced”.

CAPÍTULO 2

Marco Teórico

2.1 Antecedentes Investigativos

Dentro del repositorio de la Universidad Técnica de Ambato “U.T.A” se halló una tesis presentada por el Sr. Yu Lung Li bajo el título: “Estudio y evaluación de aplicaciones para el análisis forense de dispositivos móviles bajo Android en la Ciudad de Ambato”, en el año 2013 quien concluye que: “Actualmente la tecnología está avanzando a pasos agigantados, Ahora toda la información es almacenada en los ordenadores de manera automática, a diferencia de épocas anteriores en donde la información se almacenaba de manera manual y en papel. Las ventajas son evidentes, mayor facilidad en el manejo de la información, rapidez en la recolección y análisis de la misma, alta disponibilidad tanto en tiempo como en localidad por lo cual plantea una propuesta que mejore el análisis forense de los dispositivos móviles con Android, mediante la aplicación del software más adecuado”. [2]

Dentro del repositorio de la Universidad Técnica de Ambato “U.T.A” se halló una tesis presentada por el Sr. Carlos Vinicio Ailaca Ramírez en el año 2011 bajo el título: “Sistema de monitoreo y control de redes inalámbricas para optimización del servicio de internet en la empresa INTERCOMPU”, quien concluye que: “La investigación de nuevas aplicaciones en el control y monitoreo de la tecnología de redes de comunicación inalámbrica, brinda un orden a un mundo congestionado presente en el internet y la transferencia de información, con el fin de mejorar las asignaciones de ancho de banda y protocolos de transmisión”. [3]

Dentro del repositorio de la Universidad Técnica de Ambato “U.T.A” se halló una tesis presentada por el Sr. Mario Villalba en el año 2011 bajo el título: “Sistema de control de temperatura para la climatización del quirófano y la sala de recuperación del Hospital Municipal Nuestra Señora de la Merced”, quien concluye que: “Un sistema de control de temperatura para la climatización altamente efectivo mediante la electrónica avanzada, ha logrado tener la capacidad de ofrecer un nivel de ambiente adecuado a la industria, viviendas, oficinas, aulas y principalmente a los hospitales con la finalidad de proveer un ambiente confortable mediante sistemas de control estables y robustos frente a perturbaciones y errores, evitando así comportamientos bruscos e irreales”. [5]

Dentro de las publicaciones Internacionales de la IEEE se tiene un artículo técnico presentado en abril de 2013 por el Ing. Thomas Timmermann bajo el título “Criterios para la selección adecuada de una solución de monitoreo de red” quien de manera clara concluye que: “El mal funcionamiento y las fallas en los procesos empresariales fácilmente se traducen en pérdidas de tiempo, así como pérdidas financieras. Un sistema de monitoreo de red el cual constantemente monitoree los procesos dentro de la red informática, y efectúe un análisis y alerte al personal de TI (Tecnología Informática) tan pronto como se produzca un error o se supere alguno de los valores críticos definidos, es altamente recomendado a fin de mantener un control constante de la disponibilidad, el rendimiento y el uso del ancho de banda dentro de la red de TI. El monitoreo de red le permite a los administradores intervenir rápidamente, ya sea local o de forma remota, en caso de no encontrarse presente”. [4]

2.2 Fundamentación teórica

Este capítulo trata de la sustentación teórica del proyecto por lo que se empezó con una introducción teórica a la gestión de red, luego se realizó un estudio del protocolo SNMP y de sus características enfocadas a la versión 3, además se llevó a cabo un análisis de las herramientas de desarrollo que se van a utilizar.

2.2.1 Data Center (Central de Datos)

Un Data Center es un sitio de procesamiento de datos donde se concentran los recursos tecnológicos de una red en particular, permitiendo que todo el tráfico de la red sea procesado en este lugar. Generalmente un Data Center debe cumplir con estándares internacionales en cuanto a seguridad y disponibilidad del servicio, la infraestructura se detalla a continuación:

- Espacio físico para instalación de dispositivos de redes, telecomunicaciones, servidores, etc., con respaldos de energía, aire acondicionado y seguridad.
- Conectividad a una salida internacional (internet) y privada (LAN).
- Personal de operación y supervisión de los dispositivos existentes.

TIA (Telecommunications Industry Association's) 942

La “Asociación de la Industria de Telecomunicaciones” (Telecommunications Industry Association, TIA) es una asociación de comercio en los Estados Unidos que representa a cientos de compañías involucradas en el área de telecomunicaciones, la cual se encarga de gestionar normativas y procesos en lo referente a equipos y centros de telecomunicaciones. [6]

Tier

La definición de Tier lleva implícito el nivel de fiabilidad de un centro de datos asociados a cuatro niveles de disponibilidad definidos. A mayor número en el Tier, mayor disponibilidad, y por lo tanto mayores costes asociados en su construcción y más tiempo para hacerlo. [6]

Establece cuatro niveles de (TIERS), en función de la redundancia necesaria para alcanzar niveles de disponibilidad de hasta el 99.995% [6]

Tabla 2.1 Tipos de Tiers

Tier	% de Disponibilidad	% de Parada	Tiempo de parada (Anual)
Tier I	99.671%	0.329%	28.82 horas
Tier II	99.741%	0.251%	22.68 horas
Tier III	99.982%	0.018%	1.57 horas
Tier IV	99.995%	0.005%	52.56 minutos

Tiers Características

Tier I

- Componentes no redundantes (Aire, UPS Generador)
- Salida internacional y local no redundante

Tier II

- Componentes redundantes (Aire, UPS, Generador)
- Salida internacional y local no redundante

Tier III

- Componentes redundantes
- Vías de distribución redundantes (Activa – Pasiva) a través de fibra óptica y enlaces inalámbricos en frecuencias licenciadas
- El Data Center dispone de dispositivos hot swap, es decir, dispositivos que pueden ser removidos mientras operan, sin causar efectos negativos en la red.

Tier IV

- Componentes redundantes

- Múltiples vías de distribución activas y redundantes, además de varias salidas internacionales
- Los componentes pueden ser removidos durante un evento planeado sin generar interrupciones en el sistema (concurrentemente mantenible)
- No susceptible a interrupciones por un evento no planeado

Monitoreo de Infraestructura TI (Tecnologías de la Información)

Las herramientas de monitoreo tienen como objetivo facilitar el trabajo para los administradores de TI (Tecnologías de la información), debido a que proporcionan un conocimiento en tiempo real del estado y servicios que la red dispone, entre los objetivos de las herramientas de monitoreo se disponen las siguientes: [6]

- Alertar sobre problemas que puedan presentarse en la red actualmente o a posterior
- Conocer de forma centralizada todo el equipamiento existente, tanto a nivel de hardware y software
- Mantener información histórica del comportamiento de la red, así como gráficas de sucesos, rendimiento, comportamiento, etc

2.2.2 Técnicas de monitoreo:

Monitoreo activo:

Se realiza a través de pruebas en la red operativa, esto se realiza enviando o generando tráfico, lo cual permite medir tiempos de respuesta y rendimiento de los equipos activos en toda la red, esta prueba permite medir la eficiencia general de la red. [7]

Técnicas de monitoreo activo:

Basado en ICMP con protocolo UDP

- Diagnosticar problemas en la red.
- Detectar retardo, pérdida de paquetes.
- RTT (Traceroute)
- Disponibilidad de host y redes.

Basado en TCP (Protocolo de Control de Transmisión)

- Tasa de transferencia.
- Diagnosticar problemas a nivel de aplicación

Monitoreo pasivo:

Se basa en la recolección y análisis del tráfico que se genera en la red, empleando dispositivos como routers, switches o servidores que tengan instalado un software para el monitoreo, empleando protocolos tales como SNMP, RMON y NETFLOW. [7]

Técnicas de monitoreo pasivo:

- **Solicitudes remotas:**

SNMP (Protocolo Simple de Administración de Red):

Esta técnica permite obtener información sobre el uso del ancho de banda en los diferentes dispositivos de red que se encuentre configurado y tengan el servicio activo de SNMP, y además con el uso de paquetes denominados traps es posible conocer información cuando un evento inusual se ha generado. [7]

Análisis del tráfico:

Es una técnica que se puede utilizar a través de software en un sistema operativo o de un hardware específico, el cual se encarga de detectar el puerto y protocolo que la trama de red tiene, con lo cual el administrador puede detectar que páginas o aplicaciones utiliza un usuario con una ip de origen, pudiendo de esta manera crear reglas o restricciones para tal o cual usuario y servicio. [7]

Consideraciones para el monitoreo

El monitoreo debe enfocarse en ciertos recursos o actividades que realiza un dispositivo de red, donde los puntos más comunes a monitorearse son los siguientes: siguientes: [7]

- uso de ancho de banda
- Consumo de CPU y Memoria Ram.
- Estado operativo y lógico de las conexiones.
- Tipo de tráfico que atraviesa o genera la red.
- Alarmas

Es necesario definir el alcance de los dispositivos que se va a monitorear, debido a la amplitud que la red pueda tener, para lo cual es necesario segmentar de la siguiente manera: [7]

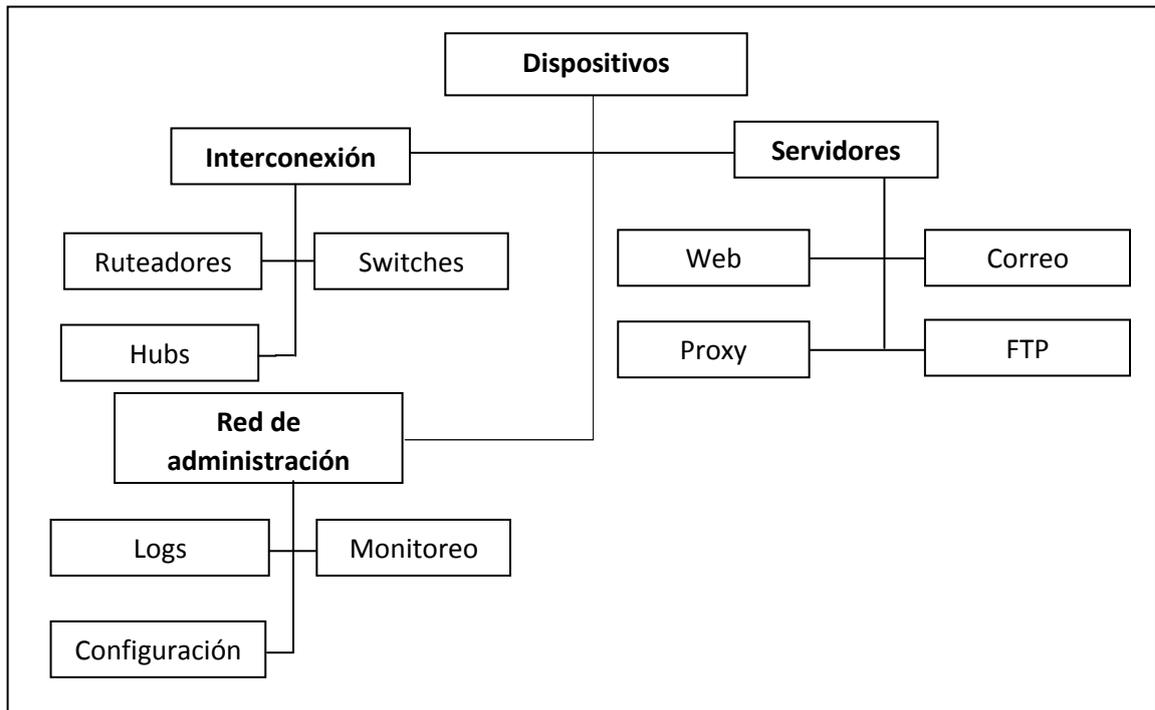


Figura 2.1: División de los elementos que pueden ser monitoreados
Fuente: Tecnología – Los recursos de la red y su monitoreo [7].

Elección de herramientas de monitoreo:

Existe un gran número de herramientas para resolver el problema del monitoreo de una red, a nivel de software libre como propietario, su elección depende de varios factores, los cuales se citan a continuación: [8]

- El perfil del administrador de la red, así como sus conocimientos en el uso de determinadas herramientas
- Recursos económicos disponibles.
- Hardware disponible

Se hará énfasis en dos herramientas:

Cacti:

Es un software para el monitoreo completo de una red, utilizada Herramientas de Bases de Datos Circulares donde almacena información de los dispositivos seleccionados y sus gráficos. Dispone de múltiples métodos de obtención de datos como SNMP y scripts, donde al recibir alguna notificación señalada como error o daño enviará una alarma mediante el manejo de umbrales. [9]

Net-SNMP:

Es un software que puede obtener información vía SNMP de los equipos de interconexión. Soporta la versión 3 del protocolo SNMP, con mecanismos de seguridad a nivel de confidencialidad y de autenticación, al utilizar SNMP utiliza traps para enviar y recibir notificación de eventos la cual ofrece mecanismos de seguridad tanto de confidencialidad como de autenticación. Provee de manejo de traps para la notificación de eventos. [9]

2.2.3 Topología del sistema de monitoreo

Un sistema de monitoreo se instala en un servicios que hace las solicitudes mediante SNMP a los dispositivos que poseen capacidad para ejecutar este protocolo, y mediante un agente SNMP envía la información requerida, lo cual se describe a continuación: [9]

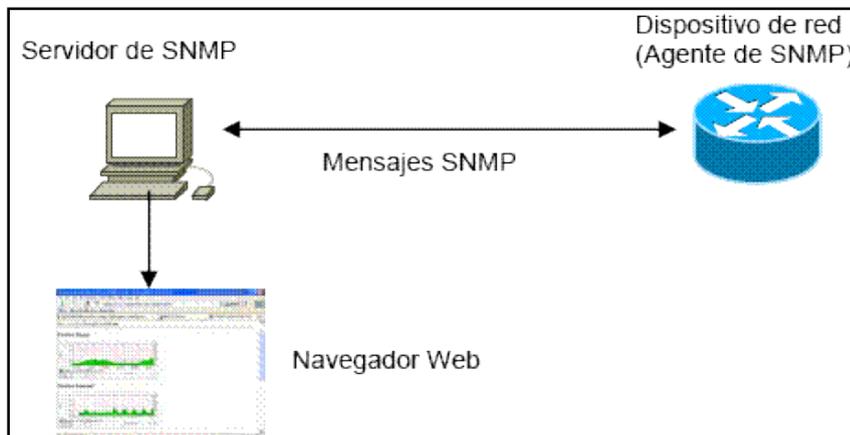


Figura 2.2: Solicitud mediante SNMP

Fuente: Technologic – simple network management protocol [9]

A continuación se detalla el envío de un trap generados por el dispositivo de red hacia el servidor SNMP: [9]

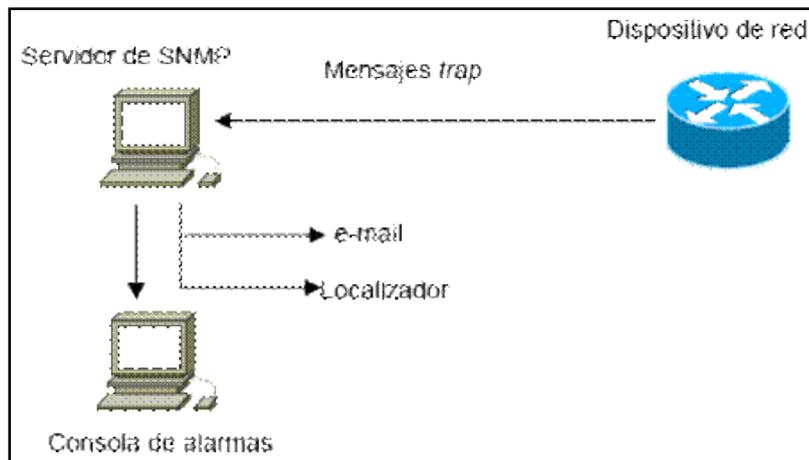


Figura 2.3: Envío de traps

Fuente: Technologic – simple network management protocol [9]

Ejemplo de monitoreo basado en el protocolo SNMP con la herramienta CACTI.



Figura 2.4: Monitoreo basado en el protocolo SNMP-CACTI
Fuente: Technologic – simple network management protocol [9].

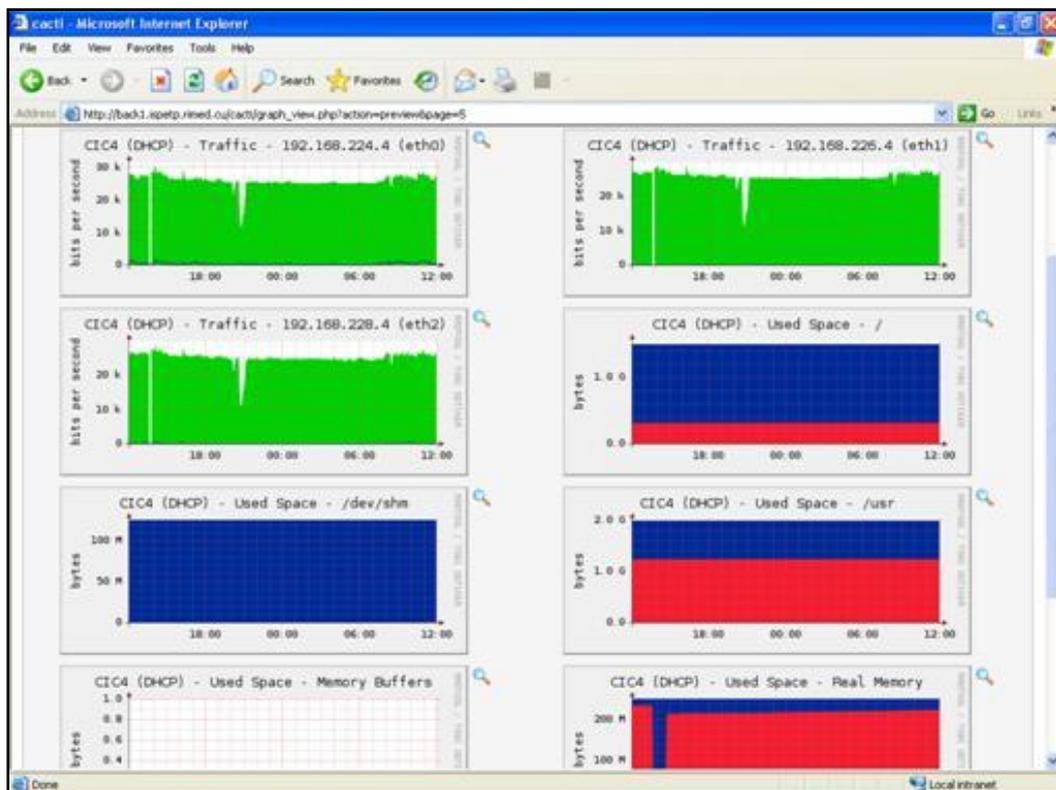


Figura 2.5: Gráficos en modo árbol (Router GW-UCPETP – RIMED)
Fuente: Technologic – simple network management protocol [9].

2.2.4 Sistema Operativo Movil

Un sistema operativo móvil, es un software que se encarga de controlar todo el hardware existente en el dispositivo, de la misma manera como Microsoft Windows 7 realiza el control y uso del hardware de una PC de escritorio o portátil. [10]

Sistemas Operativos para Dispositivos Móviles

Existen varios fabricantes de sistemas operativos móviles pero entre las más reconocidas están Symbian, BlackBerry OS, Windows Mobile, iPhone OS, y Android, pero el uso de uno u otro de los sistemas operativos es el que determina las capacidades multimedia de los dispositivos y la forma en cómo estas van a interactuar con el usuario. [10]

Symbian

Symbian es un sistema operativo multitarea, que posee una arquitectura diseñada para que la batería tenga una mayor durabilidad y rendimiento, con alta capacidad de gestión gracias a su memoria y la gestión limitada de recursos que administra, lo cual la hace versátil y eficiente.

Android

Android es un sistema operativo basado en Linux, donde dispone de un kernel que puede ser comprendido y ejecutado a través de código java. Las aplicaciones para Android se escriben y desarrollan en el lenguaje de programación JAVA

Características de Android

Una de las características importantes de la plataforma de Android es que no hay diferencia entre las aplicaciones incorporadas en el sistema operativo y las aplicaciones que se crean con el SDK, esto significa que el desarrollador puede escribir aplicaciones de gran alcance para aprovechar los recursos disponibles en el dispositivo.

Android es únicamente un software que aprovecha las bondades y características del kernel de Linux para realizar la interacción con el hardware que posee el dispositivo, siendo compatible con casi todas las marcas de dispositivos móviles a nivel mundial, lo cual lo vuelve más atractivo y requerido. [10]

Arquitectura de Android

La arquitectura de Android define estos cuatro elementos de los que podemos heredar:

- **Activities:** es la parte neurálgica del entorno gráfico de la aplicación, allí se interactúa entre los elementos gráficos y el código fuente para realizar acciones o procesos

- Services: es un proceso que se ejecuta sin necesidad de disponer de una interfaz gráfica.
- Intent: se encarga de interpretar de forma abstracta una acción, funciona de forma similar a un evento o a una interrupción.

Seguridad en Android

Android es un sistema operativo móvil basado en la tecnología Linux, por lo cual podemos decir que este funciona de la misma manera que un Desktop corriendo un sistema operativo moderno y con accesos a internet, pero con la diferencia de un hardware adicional como una cámara fotográfica o un GPS, lo cual significa que al igual que cualquier computador Android tiene los mismos riesgos con los Smartphones, hay una diferencia importante que también puede tener un alto impacto y es la Movilidad. [10]

Blackberry

Es un sistema operativo creado por RIM para dispositivos móviles y actualmente ocupa el cuarto puesto de ventas a nivel mundial con un 11% del mercado.

La familia Blackberry incluye dentro de sus componentes Smartphones y software, además de su Playbook que fue la primera Tablet a nivel profesional, pero que desafortunadamente no ha dado los resultados esperados a nivel de ventas, superada ampliamente por Tablets de otras marcas como Apple con su iPad y Samsung. [10]

IOS

IOS (anteriormente era mencionado como iPhone OS) es un sistema operativo móvil de Apple desarrollado originalmente por iPhone, siendo utilizado por los dispositivos iPhone, iPod Touch y iPad. Es un derivado de Mac OS X, que a su vez está basado en Darwin BSD.

El sistema operativo IOS tiene cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de servicios principales, la capa de medios de comunicación y la capa de Cocoa Touch.

Todo el sistema se encuentra en la partición “root” del dispositivo, ocupa bastante menos del medio en megabytes, del total del dispositivo de 8 GB o de 16 GB.

Esto se realizó para poder soportar futuras aplicaciones de Apple, también como aplicaciones de terceros publicadas como iTunes Store o App Store. [10]

Características Generales de los S.O. Móviles

A continuación se explicaran por medio de algunos cuadros, las características más comunes y necesarias de los sistemas operativos móviles desde el punto de vista de detalles básicos, interfaz de usuario, funcionamiento y desarrollo de terceros. [11]

Tabla 2.2 Detalles Básicos de los Sistemas Operativos Móviles
Fuente: La gran comparación de los sistemas operativos móviles [11].

Características	IOS 5	Android 4.0
Kernel	OSX	Linux
Multitarea	Pseudo	Si
Hardware soportado	iPhone, iPod touch y iPad solamente	Soporte para una gran variedad de hardware
Seguridad	Si (no necesario para security suit)	Susceptible a malware
Soporte flash	No	Si
Aplicaciones	Más de 500.000	Más de 250.000

Como podemos observar en la Tabla 2.4, el estudio de basó en ítems como el tipo de núcleo, adaptabilidad, edad de la plataforma, soporte para empresas y tecnologías inalámbrica y lógicamente la mayoría de los sistemas operativos tienen sus pros y sus contras. Lo más importante en un sistema operativo es el núcleo (Kernel). Por ejemplo, Android usa un kernel Linux, con una mezcla especial de Java. [11]

El iPhone se basa en OS X, que a su vez es una variante de Unix, uno de los sistemas operativos más poderosos en el mundo de la informática. S60 y Windows Mobile son SO muy maduros y estables, aunque la edad no siempre es una ventaja. Por último, RIM usa un kernel propio, que al igual que Android, tiene un motor Java y aunque han mejorado la interfaz notablemente, suele mostrar algunas limitaciones propias de su edad. [11]

2.2.5 Sistema operativo Android

Es un sistema operativo diseñado para dispositivos móviles, tiene 7 años de vida, el mismo que se encuentra en constante evolución, adaptación y buscando cubrir las necesidades de usuarios móviles como fijos en la actualidad, con aplicaciones a nivel de domótica, televisión, redes, etc. [12]

Características

Android es una plataforma para el desarrollo basada en software libre y construido en el

lenguaje JAVA, para dispositivos móviles, creada por una pequeña empresa llamada AndroidInc. Android ha tenido gran aceptación en el mercado tanto por usuarios como por la industria.

A continuación se muestra el predominio que tiene actualmente Android: [12]

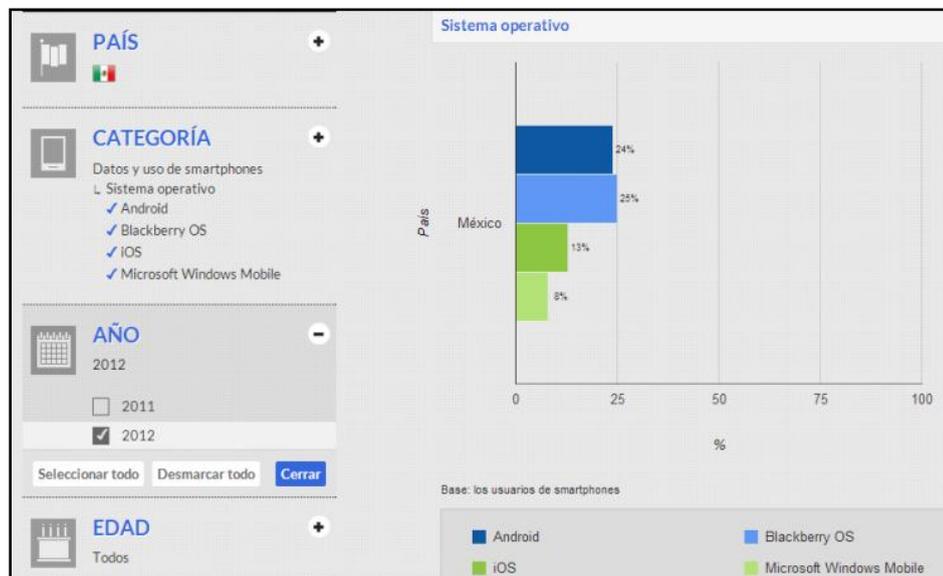


Figura 2.6: Aceptación en el mercado-Android
Fuente: Desarrollo de aplicaciones para Android [12].

Desarrollo de aplicaciones

Existen diferentes herramientas de desarrollo para realizar aplicaciones en Android, tanto a nivel netamente intuitivo y a nivel de programación intensa. [12]

El entorno de desarrollo de Eclipse es el software más utilizado para crear aplicaciones Android, en éste se programa en Java, pero también en conjunto con el SDK (Software Development Kit) creado por Google.

El software se puede descargar desde la página oficial de Android [Android Developers, 2014]

Una vez que se descomprima el fichero descargado, se tendrá el aspecto de la figura:



Figura 2.7: Aspecto fichero descargado
Fuente: Desarrollo de aplicaciones para Android [12].

El siguiente paso será descargar las diferentes API, para ello se ejecuta el archivo SDK Manager que se muestra en la figura. La pantalla de la aplicación ejecutada se muestra en la figura:

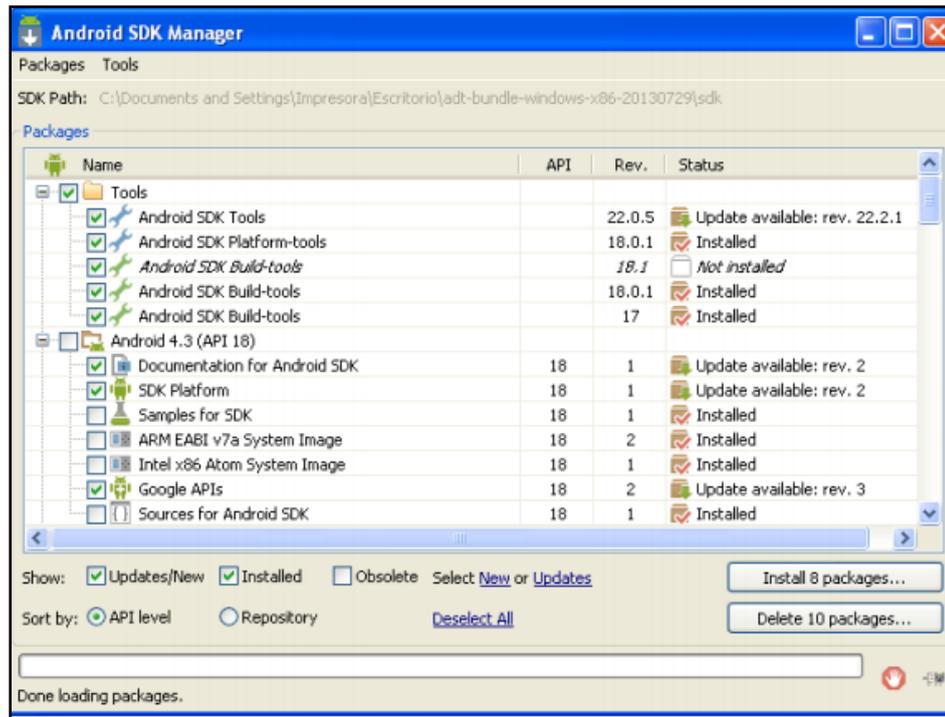


Figura 2.8: Instalación en computador personal (2014)-1

Fuente: Desarrollo de aplicaciones para Android [12].

Para crear un nuevo proyecto se tiene la opción del menú archivo/nuevo/proyecto/android. Se indican los parámetros iniciales como nombre del proyecto, entre otros. Hecho esto, se obtiene como resultado un esqueleto de programa como el indicado en la figura 2.8. [12]

2.3 Propuesta de Solución

En este proyecto de Investigación se implementa un sistema de control y monitoreo de la Central de Datos del Hospital Municipal “Nuestra Señora de la Merced”, aplicando tecnología para dispositivos móviles en la plataforma android; para optimizar los recursos de la red de datos y mejorar la climatización existente en el interior de la Central de Datos.

CAPÍTULO 3

Metodología

3.1 Modalidad de la Investigación

Se realizó una investigación de campo, ya que es necesario realizar un estudio sistemático de los hechos donde se generó el problema, con el fin de obtener información útil en el desarrollo de la propuesta y apoyar a la fundamentación de los objetivos planteados.

De igual manera se realizó una investigación bibliográfica-documental mediante libros, revistas científicas y publicaciones electrónicas para profundizar diferentes enfoques con respecto al tema de investigación, de esta manera se recopiló información valiosa que servirá como sustento científico del proyecto, ampliando teorías, conceptualizaciones y criterios de diversos autores, acorde a los requerimientos del proyecto.

3.2 Población y Muestra

En este proyecto de investigación no es necesaria la utilización de población y muestra considerando que la información necesaria se encuentra disponible y será recolectada en el departamento de tecnologías de comunicación del Hospital Municipal de Ambato.

3.3 Recolección de Información

Para la recolección de información se tomó en cuenta fuentes bibliográficas, repositorios de publicaciones, además se acudirá a diversos lugares de información como archivos, bibliotecas, librerías, videotecas, institutos de investigación, Internet y guía del tutor para el desarrollo de la parte técnica para registrar una descripción concreta y concisa de los avances que se obtuvieron durante el presente desarrollo del proyecto de investigación. Para ello es importante tener presente las diversas fuentes que nos pueden ser útiles en la tarea de conseguir información para nuestra investigación.

Entrevista

En la presente investigación se empleó ésta técnica debido a que el beneficiario directo es la persona encargada de controlar y gestionar el Data Center del Hospital Municipal Nuestra Señora de la Merced, la encuesta se encuentra en el Anexo 1.

3.4 Procesamiento y Análisis de Datos

Para el desarrollo del proyecto se llevarán a cabo las siguientes actividades.

- Levantamiento de la información necesaria en el departamento de tecnologías de comunicación del Hospital Municipal “Nuestra Señora de la Merced”.
- Identificación de los dispositivos a ser controlados y monitoreados que se encuentran en la Central de Datos.
- Determinación de las formas de configuración, rendimiento, seguridad y manejo de fallas, mediante un uso adecuado de los protocolos y recursos que presentan la red interna.
- Determinar el método y la alternativa de solución encontrada mediante la investigación bibliográfica.
- Diseño e implementación de un circuito para el monitoreo y control de la climatización interna de la Central de Datos.
- Diseño e implementación del software sobre el sistema operativo Android para conocer el estado operativo y funcional del Data Center a través del protocolo SNMP
- Realización de pruebas del sistema de control y monitoreo de la Central de Datos diseñado.
- Evaluación de la propuesta por el método de experto.

CAPÍTULO 4

Desarrollo de la propuesta

4.1 Levantamiento de la información

Se realizó una encuesta a la área de sistemas y comunicaciones del Hospital Municipal “Nuestra Señora de la Merced” encargada por el Ing. Diego Rubio / Administrador de TIC logrando así determinar los problemas más representativos que presenta la red de datos internet y climatización del data center, para su respectiva administración, haciendo énfasis en modelos funcionales, proyectos de mejoras, compra de equipos, servicios de red, sistemas de monitoreo, determinando cuales son los problemas más comunes para establecer una propuesta de solución

Desarrollo de la Entrevista

1. ¿El Data Center del Hospital cumple con todos los requerimientos técnicos para su funcionamiento y operatividad?

El Data Center se encuentra organizado de una manera muy adecuada, pero no cumple reglamentos internacionales, es decir, los criterios para evaluar a un Data Center como son los Tiers no se encuentran aplicados, pero se dispone de un rack y un cuarto frío donde se albergan todos los dispositivos y existe un respaldo de energía suficiente para evitar daños en los equipos.

2. ¿Cuenta actualmente con un sistema de control y monitoreo de los dispositivos del Data Center?

No, el control únicamente lo puedo realizar encontrándome en el Data Center físicamente.

3. ¿Puede verificar en tiempo real el estado de la climatización al interior del Data Center?

Sí, pero no se dispone de un dispositivo de monitoreo constante, solo es posible verificar si el aire acondicionado se encuentra activo.

4. ¿Necesita un conocimiento general del estado de su red, sin la necesidad de encontrarse presente en el Hospital?

Sí, es muy necesario debido a que se han suscitado problemas de calentamiento en los equipos debido a fallas en el aire acondicionado y caída de ciertos dispositivos de la red varias horas sin tener un conocimiento inmediato del suceso, lo cual genera problemas en el sistema informático del Hospital, pues no tengo un conocimiento constante del funcionamiento del Data Center.

5. ¿Cree necesario disponer de una herramienta para el control de sus equipos Core y Climatización del Data Center basada en una aplicación para dispositivos móviles?

Sí, es muy necesario y el tener un control y monitoreo del Data Center en un dispositivo celular sería un apoyo significativo para mi labor.

6. ¿Dispone de un Smartphone con tecnología Android o iOS para utilizar la aplicación antes mencionada?

Sí, dispongo de un teléfono celular Samsung con tecnología Android y disponibilidad de datos y de una Tablet con el mismo sistema operativo apto para instalar y utilizar la aplicación.

Una vez realizada la entrevista las fue posible determinar que el administrador del Data Center necesita disponer de una herramienta tecnológica que en tiempo real le permita conocer el estado general del Data Center, para evitar daños o falencias en el sistema interno; todo esto respaldado por el aval del Hospital, específicamente del administrador para proporcionar la información requerida para realizar de forma efectiva y eficiente el actual proyecto; el sistema operativo sobre el cual se realizará la aplicación será Android, justificado en la disponibilidad de equipos por parte del administrador y lo basta de la información disponible.

4.2 Identificación de los dispositivos a ser controlados y monitoreados.

El Data Center del Hospital cuenta con una red centralizada, la cual tiene los siguientes dispositivos:

Router de Balanceo

Recibe las dos conexiones WAN existentes por parte de los proveedores de internet, en este caso una conexión con CNT y TELCONET, las cuales salen a través de otra interface que tiene conexión con la red interna, como se visualiza en la figura 4.1



Figura 4.1: Router de Balanceo TP Link TL-R480T
Elaborado por: Alex Francisco Rodríguez

Características:

- Hasta 4 puertos WAN equipados con herramientas avanzadas de carga para garantizar el máximo ancho de banda y respaldos disponibles
- Provee cuentas extensivas al cliente y a la administración de la red con soporte a un servidor PPPoE
- Recursos de ancho de banda específicos a clients basados en un único ambiente de aplicación

Switch de Core

Realiza un enrutamiento básico lógico de los paquetes que circulan por la red, a través de vlans y rutas estáticas, con lo cual se evita que la red se inunde de broadcast y que exista comunicación entre todos los usuarios, para lo cual se realiza la segmentación, el switch en mención se puede visualizar en la figura 4.2:



Figura 4.2: Switch Cisco Catalyst 3560-C
Elaborado por: Alex Francisco Rodríguez

Funciones:

La familia Catalyst 3560 (WS-C3560G-24TS-S) de Cisco es una completísima línea de switches de alto rendimiento diseñados para ayudar a los usuarios a que pasen de forma sencilla de las redes LAN compartidas tradicionales a redes completamente conmutadas.

Los switches Catalyst de Cisco ofrecen un amplio espectro para aplicaciones de usuarios, desde switches para pequeños grupos de trabajo hasta switches multicapa para aplicaciones empresariales escalables en el centro de datos o en el backbone.

Características

- Tipo de dispositivo Conmutador
- Memoria RAM 128 MB
- Memoria Flash 32 MB
- Cantidad de puertos 24 x Ethernet 10Base-T, Ethernet 100Base-TX, Ethernet 1000Base-T
- Velocidad de transferencia de datos 1 Gbps
- Protocolo de interconexión de datos Ethernet, Fast Ethernet, Gigabit Ethernet
- Protocolo de gestión remota SNMP 1, RMON 1, RMON 2, RMON 3, RMON 9, Telnet, SNMP 3, SNMP 2c, HTTP, SSH-2
- Modo comunicación Semidúplex, dúplex pleno
- Características Capacidad duplex, conmutación Layer 3, conmutación Layer 2, auto-sensor por dispositivo, Encaminamiento IP, soporte de DHCP, alimentación mediante Ethernet (PoE), negociación automática, concentración de enlaces, soporte de MPLS, soporte VLAN.

Servidores de base de datos y correo

Existen en la Data Center dos Servidores Microtower HP, el primero es utilizado como Back Up de Base de Datos, es decir, es el respaldo que existe del servidor de base de datos y otro servidor de correo el cual se encuentra configurado para el almacenamiento y gestión del correo interno, como se puede observar en la figura 4.3:



Figura 4.3: Servidor Microtower HP
Elaborado por: Alex Francisco Rodríguez

Características

La Computadora de Escritorio Empresarial Microtorre HP Compaq dx2000 ha sido específicamente proyectada para ofrecer valor de primera línea al más bajo precio posible. Construida en torno a la tecnología para computadora de Intel y ofreciendo componentes estándares de la industria, estas plataformas de microtorre compactas pero expansibles satisfacen las necesidades básicas de la computación empresarial a un precio accesible. La HP Compaq Microtorre dx2000 proporciona una excelente relación de precio/rendimiento mientras ofrece simplemente la calidad, confiabilidad y asistencia técnica que los clientes de HP esperan y reciben.

Tecnología de Intel

Intel suministra los componentes principales a la plataforma de la Microtorre HP Compaq dx2000, incluyendo el chipset, microprocesador y solución de interfaz de red.

4.3 Determinación de las formas de configuración, rendimiento, seguridad y manejo de fallas, mediante un uso adecuado de los protocolos y recursos que presentan la red interna.

Diagrama actual simplificado de la red

Identificación de los dispositivos a ser controlados y monitoreados que se encuentran en el Data Center:

- Router de Balanceo TP Link: Comprobación de conectividad mediante respuesta por Ping, debido a que el router no soporta protocolos de monitoreo como SNMP, al ser un router muy básico.
- Switch de Core: Control y Monitoreo mediante el protocolo SNMP para conocer el estado de la interfaces del switch, su tráfico, estado de operatividad.
- Servidor de NAT: Equipo con plataforma Linux Centos sobre el cual se montó el servicio SNMP, el mismo que se encarga de ser la puerta de enlace para toda la red.
- Servidor de Correo: Equipo con plataforma Linux Centos sobre el cual se montó el servicio SNMP, el mismo que se encarga de gestionar los correos internos del Hospital.

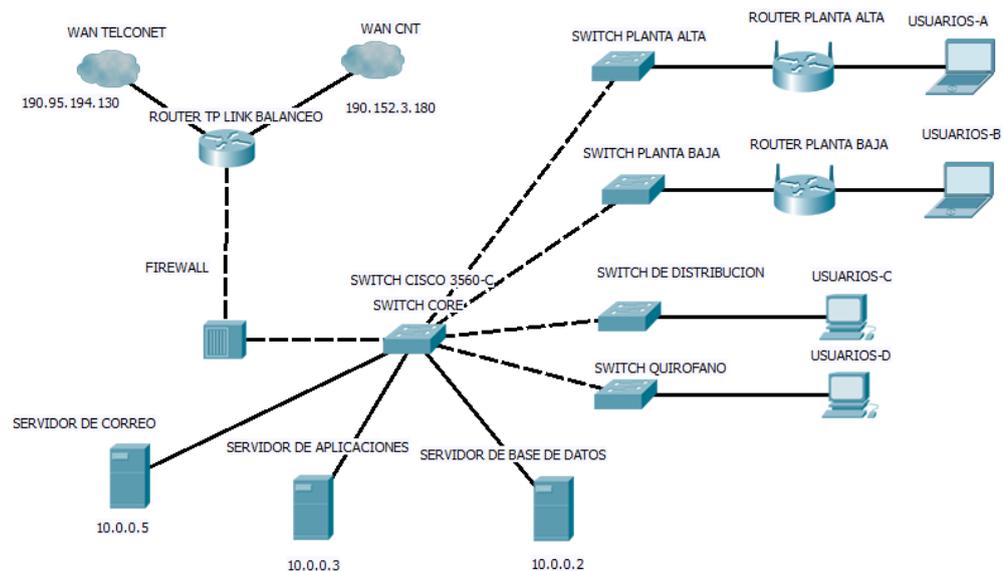


Figura 4.4: Diagrama Simplificado de red
Elaborado por: Alex Francisco Rodríguez

Configuración, rendimiento, seguridad y manejo de fallas

La red posee una topología estrella extendida la cual se muestra en la figura 4.5 ya que todos los dispositivos tienen como punto central el Switch de Core, el cual gestiona y encamina todo el tráfico, dirigiéndolo a su destino a través de Vlans, con lo cual se segmenta el tráfico y se reduce el broadcast, permitiendo que la red funcione de una manera adecuada, potenciando su rendimiento, evitando congestión de la red física y proporcionando seguridad a través de las Vlans para que según el tipo de usuario o vlan asignada pueda ver solo a los usuarios de ese segmento de red.

En cuanto al manejo de fallas no existe mecanismos de control automatizados ni lógicos en el Data Center, lo único que existe es un sistema de control de la climatización a través de un aire acondicionado encendido permanentemente para prevenir daños por sobrecalentamiento, pero a nivel lógico no existe mecanismos de detección, por lo cual es necesario implementar el protocolo SNMP en la red.

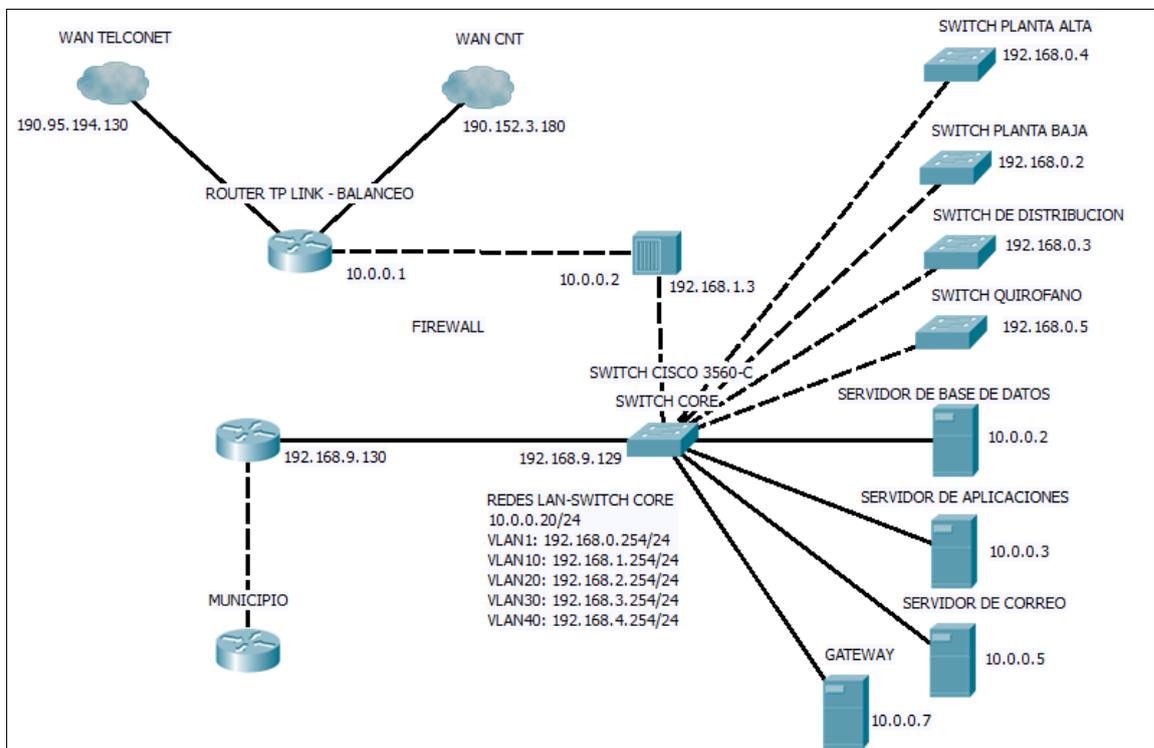


Figura 4.5 Topología de Red
Elaborado por: Alex Francisco Rodríguez

4.4 Determinar el método y la alternativa de solución encontrada mediante la investigación bibliográfica.

Fundamentación Técnico Científica

4.4.1 Administración y Gestión de Red

La administración y gestión de red comprende todo un conjunto de actividades que se encargan de la vigilancia y control de los recursos de una red, para garantizar su eficiencia y rendimiento con un mínimo costo. [13]

Sistema de Gestión de Red

Es un conjunto de elementos que permiten realizar el proceso de administración de la red, a través de una interfaz física donde se envían instrucciones y muestra información para monitorear y gestionar cualquier elemento de la red. [13]

Modelos de Gestión de Red

El modelo de gestión implementado se fundamentará en los siguientes modelos:

- Modelo de gestión de Internet.
- Modelo de gestión ISO/OSI.
- Modelo de gestión TMN.

El modelo de gestión más utilizado es el de Internet. En la Figura 2.6 se muestra los modelos de gestión de red.

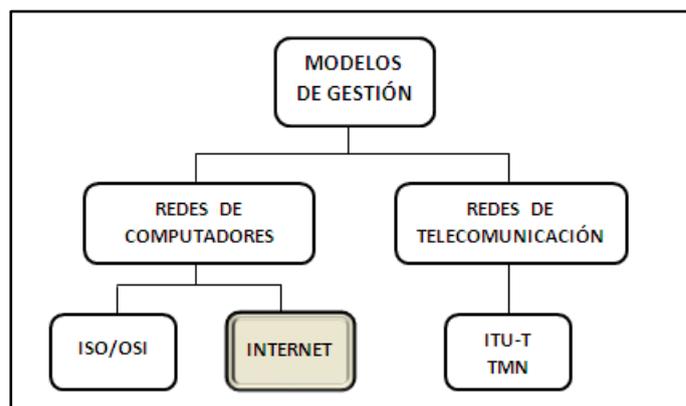


Figura 4.6: Modelos de gestión de red
Fuente: Fundamentos de administración de red [13]

Modelo de gestión de Internet

Este modelo gestiona la arquitectura para gestionar la arquitectura TCP/IP.

Los componentes que trabajan en forma conjunta en la gestión de red en Internet son los siguientes: [13]

- Estructura de la información de gestión (SMI).
- Base de información de gestión (MIB).
- Protocolo simple de gestión de red (SNMP).

Base de Información de Gestión (MIB)

La base de información de gestión (MIB) está compuesta por un conjunto de objetos administrados con sus atributos, esta información se encuentra organizada jerárquicamente. [13]

La Figura 4.9 muestra los elementos que conforman la arquitectura de un sistema de administración.

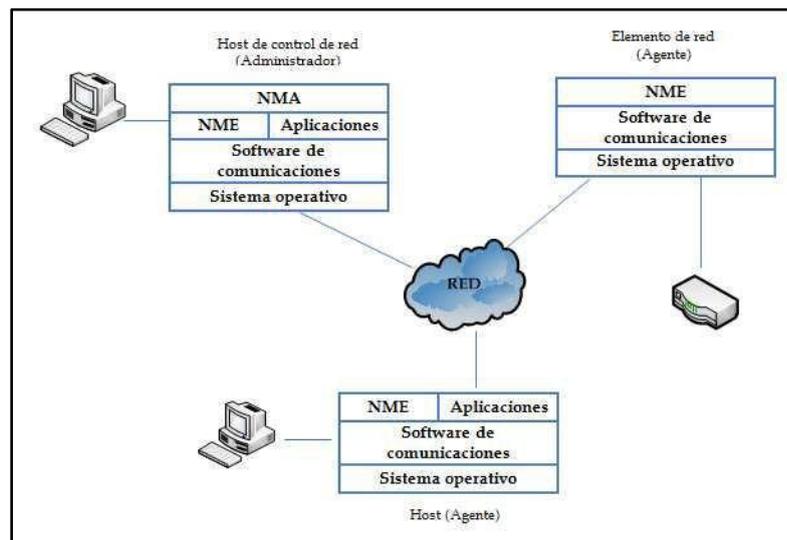


Figura 4.7: Arquitectura de un Sistema de Administración

Fuente: Fundamentos de administración de red [13]

Estructura de la Información de Gestión (SMI)

La SMI (Structure of Management Information - Estructura de la Información de Gestión) es un Lenguaje de Definición de Datos (DDL) que proporciona una manera de definir objetos administrados y su comportamiento, identifica los tipos de datos que pueden utilizarse en la MIB y especifica cómo se debe representar y nombrar los objetos dentro de la MIB, elimina la ambigüedad en la sintaxis y semántica de los datos.

La SMI está definida en el RFC 1155 para la versión 1 (SMIv1) y en el RFC 2578 para la versión 2 (SMIv2) que tiene mejoras para SNMPv2.

El objetivo principal de la SMI es mantener a la MIB simple y extensible, por lo cual solo se pueden utilizar los tipos de datos simples como escalares y arreglos bidimensionales de escalares.

La descripción de los objetos gestionados se lo hace mediante el ASN.1 (Abstract Syntax Notation - Notación Sintáctica Abstracta). [13]

La SMI proporciona formas para definir:

- La estructura de la MIB.
- Sintaxis y tipos de valores para objetos individuales.
- Codificación de los valores de los objetos.

Estructura de la MIB

La MIB es una base de datos que contiene información jerárquica, estructurada en forma de árbol, donde sus ramificaciones son los objetos a ser gestionados en una red de comunicaciones. [14]

Asociado con cada objeto en una MIB existe un identificador único conocido como OID (Object Identifier - Identificador de Objeto); este identificador de objeto es una secuencia de números enteros positivos, separados por un punto. Existe otra forma para identificar a los objetos gestionados y se lo hace mediante una secuencia de nombres que representan los números, lo que facilita al usuario al momento de ubicar cualquier objeto dentro del árbol.

En la Figura 4.10 se muestra los niveles superiores del árbol de objetos organizados según SMIv1. En este árbol de objetos tenemos en la parte superior el nodo denominado como Raíz (Root Node) el cual no tiene ningún número de identificación, solo se lo identifica con un punto (.) o con su nombre; se denomina nodos hojas a los objetos que no poseen ninguna ramificación y subárbol a los objetos que poseen ramificaciones por ejemplo el nodo iso (1). Cabe mencionar que los nodos ccitt y joint no se relacionan con SNMP.

Para el modelo de gestión de Internet el subárbol iso.org.dod.internet que se representa como 1.3.6.1 es el más importante, a partir de este existen más ramificaciones o estructuras que se derivan del subárbol mencionado.

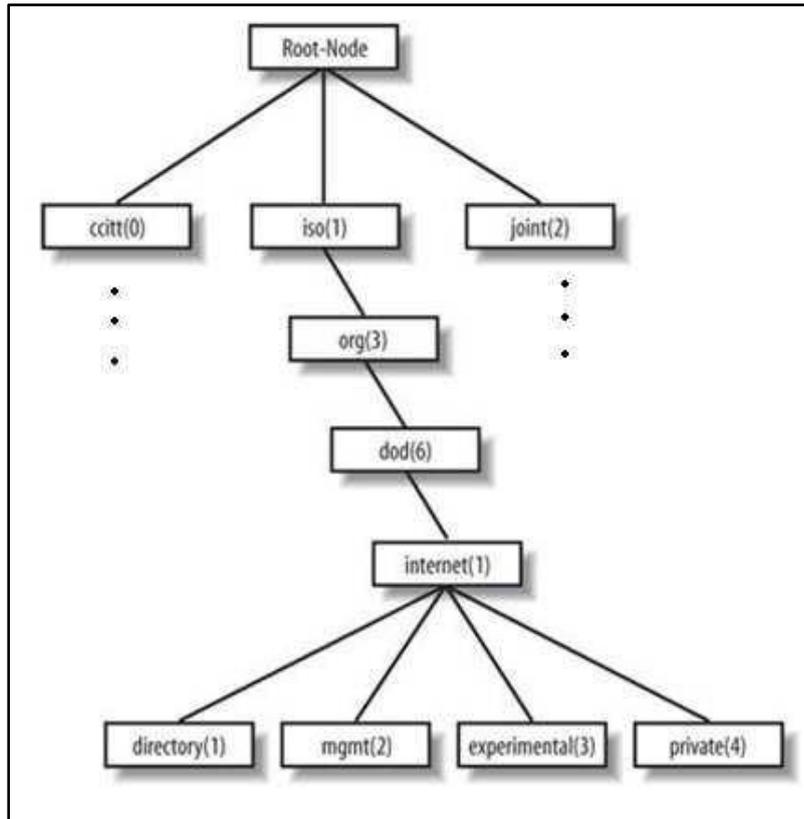


Figura 4.8: Parte de árbol de objetos según SMIV1
Fuente: Utilidades de la gestión de Redes [14]

SMIV2 extiende el árbol de objetos SMI mediante la adición de la rama SNMPv2 al subárbol Internet, además de aumentar nuevos tipos de datos e incluir una serie de cambios.

En la Figura 4.11 se muestra parte del árbol de objetos MIB según SMIV2. El OID de esta nueva rama es 1.3.6.1.6 o iso.org.dod.internet.snmpV2. [14]

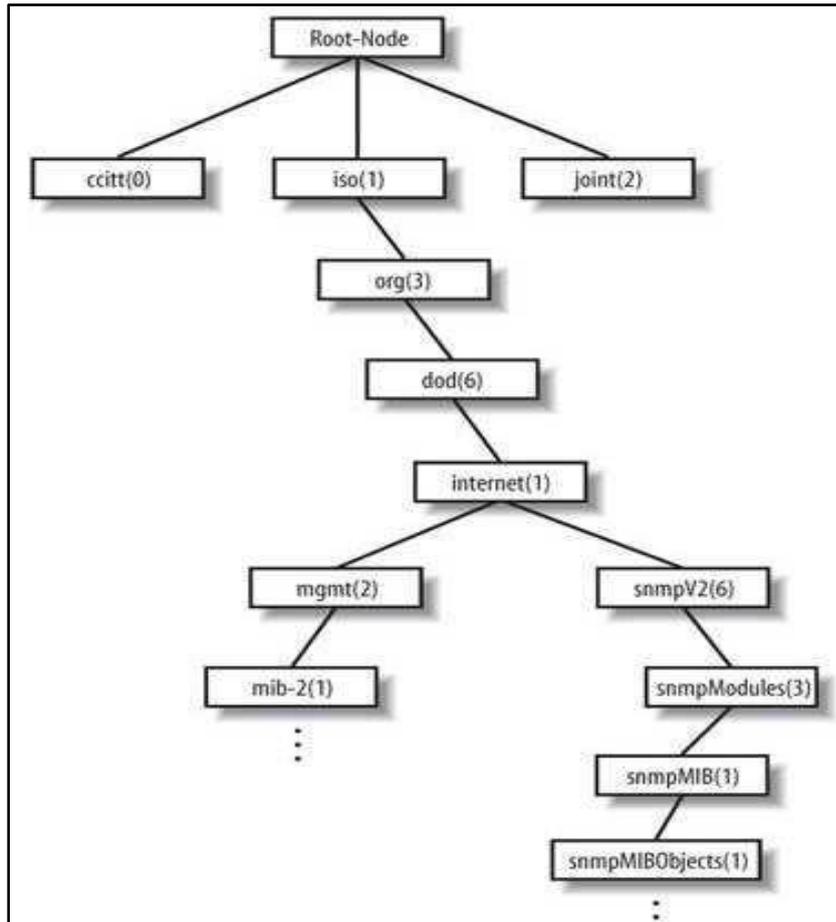


Figura 4.9: Parte de árbol de objetos según SMIv2
Fuente: Utilidades de la gestión de Redes [14]

Sintaxis de objetos

Los objetos dentro de la MIB se definen de manera específica mediante la sintaxis ASN.1 (Abstract Syntax Notation One). La definición de un objeto MIB comprende el tipo de dato, los estados permitidos, rango de valores y la relación con otros objetos dentro de la MIB. [14]

Tipos de datos

SMI define los diversos tipos de datos que son permitidos, utilizando un subconjunto de elementos y características de ASN.1. [14]

Los tipos de datos se los clasifica en los siguientes grupos:

- Primitivos o simples.
- Estructurados o compuestos.
- Definidos o etiquetados.

Base de Información de Gestión (MIB)

La MIB como ya se indicó en la sección 1.3.1, es una colección de datos que almacena valores de los objetos gestionados y se organiza de manera jerárquica. Existen dos tipos de objetos MIB:

- Escalares: Objetos que tienen una sola instancia de la variable que almacenan.
- Tabulares: Objetos que especifican varias instancias relacionadas a objetos MIB.

Dentro de la MIB existen varios grupos definidos por organismos de estandarización o empresas privadas, sin embargo el grupo MIB II debe ser implementado en cualquier elemento de red sin importar su fabricante. [14]

MIB II

La MIB II es un grupo importante dentro de la administración de red ya que todo dispositivo que soporte SNMP debe tener este grupo. La MIB II es parte del grupo management identificándose a este grupo de la siguiente manera root.iso.org.dod.internet.management.mibII o .1.3.6.1.2.1. [14]

La MIB II se compone de los siguientes grupos:

System: Los objetos de este grupo proporcionan información del sistema gestionado, como el nombre, descripción, contacto del sistema, disponibilidad.

OID: .1.3.6.1.2.1.1: Interfaces: Registra información de las interfaces de red presentes en el sistema y de los eventos ocurridos en las mismas.

OID: .1.3.6.1.2.1.2: At: Grupo de traducción de direcciones (address translation), relacionado con las traducciones de direcciones de red a direcciones físicas. Obsoleto pero se mantiene por compatibilidad con MIB-I.

OID: .1.3.6.1.2.1.3: Ip: Almacena información correspondiente al protocolo IP, tanto de configuración como de estadísticas.

OID: .1.3.6.1.2.1.4: Icmp: Recopila información sobre el protocolo ICMP, como paquetes perdidos, descartados o con errores.

OID: .1.3.6.1.2.1.5: Tcp: Almacena información relacionada al protocolo TCP, como el estado de una conexión TCP.

OID: .1.3.6.1.2.1.6: Udp: Mantiene información correspondiente al protocolo UDP.

OID: .1.3.6.1.2.1.7: Egp: Recopila información relativa al protocolo EGP.

OID: .1.3.6.1.2.1.8: Transmission: Guarda información sobre los medios de transmisión.

OID: .1.3.6.1.2.1.10: Snmp: Contiene información para implementación y operación del protocolo SNMP.

OID: .1.3.6.1.2.1.11 [25]: Se muestra en la Figura 12 los grupos que componen la MIB II.

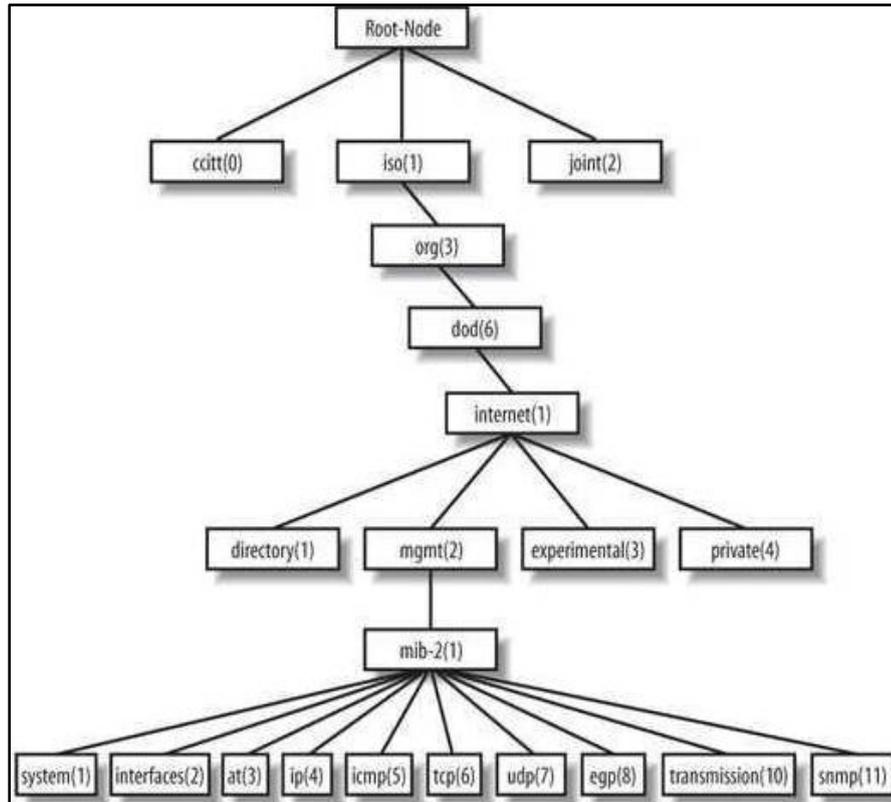


Figura 4.10: Subárbol MIB II

Fuente: Utilidades de la gestión de Redes [14]

4.4.2 Protocolo SNMP

El protocolo SNMP basado en la arquitectura TCP/IP, trabaja con una estructura cliente-servidor y usa servicios no orientados a conexión a través del protocolo de transporte UDP (User Datagram Protocol - Protocolo de Datagrama de Usuario). [15]

Los agentes SNMP utilizan el puerto UDP 161 para escuchar peticiones por parte del gestor SNMP. El gestor a través del puerto UDP 162 recibe notificaciones que genera el agente y donde debe existir un proceso gestor de interrupciones que las procesa.

La Figura 19 muestra la arquitectura TCP/IP y SNMP.

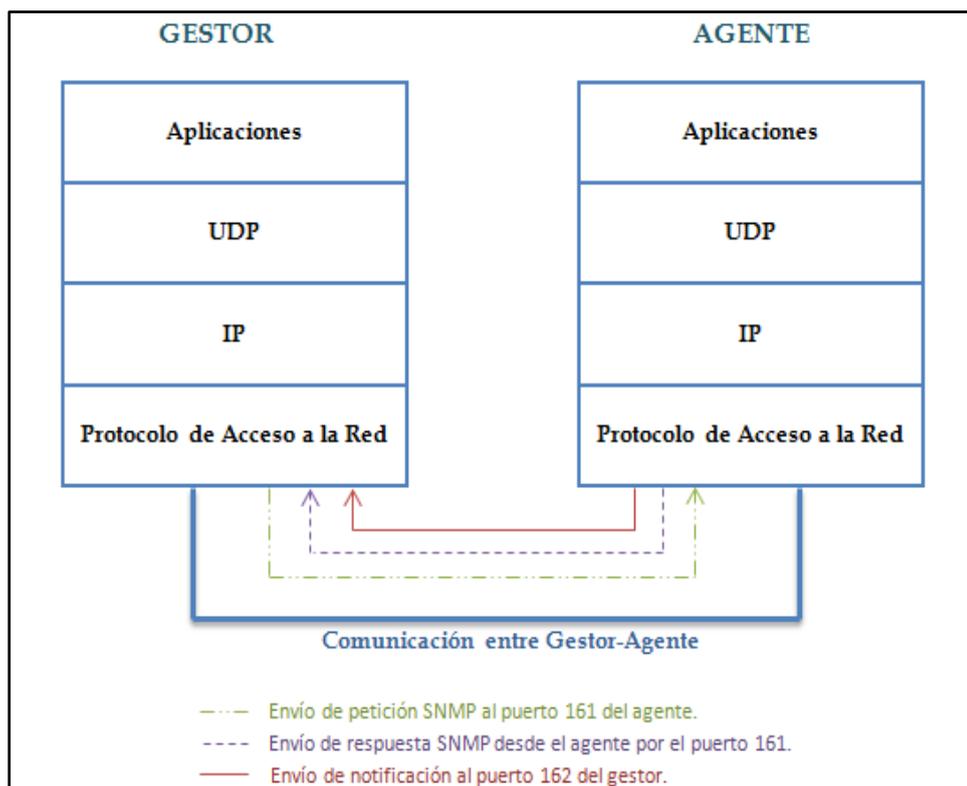


Figura 4.11: Arquitectura TCP/IP y SNMP
Fuente: Essential SNMP [15].

El protocolo SNMP está definido en varios RFC (Request For Comments - Petición de Comentarios) propuestos por el IETF (Internet Engineering Task Force - Fuerza de Tareas de Ingeniería de Internet), existen tres versiones: SNMPv1, SNMPv2c y SNMPv3.

SNMP Versión 2c

SNMPv2c como tal no incluye mecanismos de seguridad, pero sí presenta mejoras en el intercambio de información de gestión como lo es la eficiencia de operación, la funcionalidad y el rendimiento. [15]

SNMPv2c además permite la comunicación entre NMA (gestor-gestor), aumenta el número de operaciones mediante el manejo de nuevas PDU, permite la lectura de tablas completas en una sola operación, agrega características a la SMI y desarrolla una MIB para esta versión de SNMP.

El formato de mensajes para SNMPv2c es el mismo que se indicó para SNMPv1, formado por los campos versión, nombre de comunidad y PDU. Ver la Figura 1.14 que muestra el formato del mensaje.

SNMPv2 agrega las siguientes PDU:

- GetBulkRequest
- InformRequest
- Trap v2
- Report
- GetBulkRequest

Esta PDU permite reducir el número de intercambios de información necesarios para recuperar gran cantidad de información, es decir que se puede obtener mucha información con una sola petición. La limitante en la cantidad de información a recuperar es la longitud del mensaje. [15]

Es similar a getNextRequest en la manera de solicitar la información, debido a que solicita el siguiente valor en orden lexicográfico. Además esta PDU permite indicar el número de sucesores lexicográficos a seleccionar.

La PDU GetBulkRequest tiene un formato diferente, puesto que existen dos campos que no se encuentran en las demás PDU; los demás campos que conforman la PDU son Tipo de PDU (con un valor de 5), solicitud del identificador y variables asociadas que ya se explicaron anteriormente; a continuación se indican los campos adicionales que esta PDU contiene.

No repetidores: Especifica el número de variables en el campo variables asociadas, para las cuales debe ser retornado un solo sucesor lexicográfico.

Número máximo de repetidores: Determina la cantidad de sucesores lexicográficos que deben retornarse para el resto de variables del campo variables asociadas.

El número total de variables asociadas que se solicita, está dado por la fórmula, siendo N el número de no repetidores (objetos escalares en la solicitud), M el número máximo de repetidores y R el número de objetos no escalares en la solicitud.

4.4.3 Herramientas de Desarrollo para Android

Para el desarrollo de aplicaciones en Android es necesario utilizar algunas herramientas como: el lenguaje de programación Java (IDE - Eclipse), el SDK de Android y el plugin (ADT- Android Development Tools). [16]

JAVA

Java es un lenguaje de programación orientado a objetos, compilado e interpretado, que trabaja con la máquina virtual JVM. Tiene algunas herramientas que permiten desarrollar código en este lenguaje de programación: JDK e IDE. [16]

Java Development Kit (JDK): La herramienta de desarrollo Java es un conjunto de programas y librerías, que permiten que se desarrolle, compile e interprete programas en Java. La versión simplificada del JDK, la cual solo permite que se ejecute código Java, es conocida como JRE (Java Runtime Environment - Entorno en Tiempo de Ejecución de Java).

Integrated Development Environment (IDE): Los ambientes de desarrollo integrado son entornos que facilitan el desarrollo de aplicaciones de manera más ágil, debido a que se puede integrar librerías con componentes ya desarrollados. Los ambientes de desarrollo permiten desarrollar, compilar e interpretar códigos en Java en un solo programa, por lo general hacen uso de las funciones de los JDK.

Eclipse, NetBeans, IntelliJ IDEA, JCreator, etc. son algunos IDE utilizados para la creación de aplicaciones en Java.

Eclipse

Eclipse fue desarrollado inicialmente por IBM y actualmente por la Fundación Eclipse. Es un entorno de desarrollo integrado, de código abierto y multiplataforma.

SDK de Android

El SDK de Android facilita las herramientas y las API necesarias para desarrollar aplicaciones para Android, empleando el lenguaje de programación Java. [16]

Utiliza una estructura modular que separa las partes principales de las versiones de la plataforma Android, extensiones, herramientas, muestras y documentación en componentes que pueden ser instalados por separado.

El SDK incluye herramientas para depuración, empaquetado e instalación de aplicaciones, etc.; pero las más importantes son el emulador Android y el plugin para el IDE Eclipse, que se explican más adelante.

SNMP4J

Previamente se dará una definición de un término fundamental para conocer la clase SNMP4J: API: interfaz de programación de aplicaciones (IPA), abreviada como API (del inglés: Application Programming Interface), es el conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas. [17]

SNMP4J es un API de SNMP para Java, consiste de una clase de código abierto que permite el desarrollo de agentes y gestores. Existen versiones gratuitas y comerciales.

Las características que brinda el SNMP4J son las siguientes:

- Puede trabajar con UDP o TCP.
- Maneja todos los tipos de PDU.
- Soporta peticiones síncronas y asíncronas.
- SNMPv3 soporta autenticación MD544 y SHA45.
- SNMPv3 soporta privacidad DES46 y AES47.
- Maneja el módulo generador de comandos y contestador de comandos.
- Soporta modelos de procesamiento de mensajes MPv148, MPv2c49 y MPv350.
- Maneja multihilos.
- Licencia Apache.
- Trabaja con Java SE 1.6 o posteriores o con Java EE.
- SNMP4J tiene un API para los clientes SNMP y otro para los agentes.

API SNMP4J 2.0.0 51

Maneja algunos paquetes importantes, los cuales se indican a continuación:

org.snmp4j: Provee de clases e interfaces utilizadas para la creación, envío y recepción de mensajes SNMP (v1, v2c, v3). [17]

Las interfaces que forman parte del paquete org.snmp4j son:

- Certified Identity,
- Command Responder,
- Message Dispatcher,
- Snmp. ReportHandler,
- Timeout Model,
- Transport Mapping,
- Target, User y Session.

Las clases que forman parte del paquete org.snmp4j son:

- Snmp,
- Abstract Target,
- Certified Target,
- Command Responder Event,
- Community Target,
- Mutable PDU,
- Default Time out Model,
- Message Dispatcher Impl,
- PDU, PDUv1,
- User Target,

- Scoped PDU,
- Secure Target,
- SNMP4JSettings y Transport State Reference.

Clase Snmp: Esta clase es el núcleo del API SNMP4J, provee de funciones que se encargan del envío de todo tipo de PDU y recepción de las mismas; permite envíos de forma sincrónica o asincrónica. Es una clase independiente del protocolo de transporte; se puede configurar por defecto al protocolo UDP para el transporte de SNMP. [17]

Proporciona varios constructores y algunos métodos utilizados para el envío y recepción de todas las PDU.

- **org.snmp4j.asn1:** Proporciona algunas clases e interfaces para la asignación entre valores con formato y la sintaxis de transferencia, siguiendo las reglas básicas de codificación.
- **org.snmp4j.event:** Contiene varias clases e interfaces utilizadas para el procesamiento de eventos.
- **org.snmp4j.log:** Proporciona clases e interfaces para el manejo de registros.
- **org.snmp4j.mp:** Contiene clases e interfaces utilizadas para el procesamiento de mensajes.
- **org.snmp4j.security:** Provee de varias clases e interfaces para la autenticación y privacidad de mensajes SNMPv3.
- **org.snmp4j.smi:** Contiene algunas clases utilizadas para representar los tipos de datos SMIV1, SMIV2 y algunos tipos de datos primitivos ASN.1.
- **org.snmp4j.tools.console:** Provee la clase LogControl y SnmpRequest.
- **org.snmp4j.transport:** Facilita el protocolo de transporte para SNMP.
- **org.snmp4j.transport.ssh:** Provee dos interfaces que facilitan el acceso a una sesión SSH.
- **org.snmp4j.transport.tls:** Provee de clases y una interfaz que utilizan el protocolo TLS.
- **org.snmp4j.util:** Contiene clases para la recuperación de tablas SNMP y para soporte de mensajes de envío multihilo.
- **org.snmp4j.version:** Devuelve información sobre la versión de SNMP4J.

4.4.4 Eclipse

El elemento necesario para desarrollar el software en Android es Eclipse. Este programa es un IDE (Integrated Development Environment - Entorno de desarrollo integrado) que facilita el desarrollo y la creación de aplicaciones. [18]

Instalación

Para instalar Eclipse, diríjase a: www.eclipse.org/downloads/ y descargue la versión de Eclipse para Java (a lo largo del libro se utilizará la versión Indigo). Cuando se haya descargado el archivo, extraiga la carpeta del archivo comprimido zip en una ubicación de su elección.

Configuración

Inicie Eclipse mediante el acceso que se encuentra en la raíz de la carpeta extraída.

En la primera ejecución de Eclipse, deberá configurar su workspace. Se trata de su espacio de trabajo, todos los proyectos que desarrollará se crearán y se guardarán en esta carpeta.

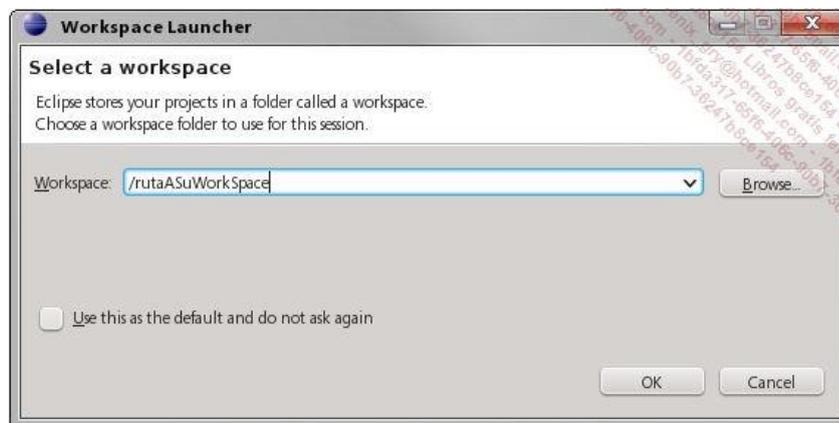


Figura 4.12: Creación del Workspace
Fuente: Android Developer - Eclipse IDE [18].

Plugin ADT

El siguiente paso consiste en la integración del plugin ADT (Android Development Tools) en Eclipse. Este plugin le da acceso a un entorno de desarrollo integrado para Android en Eclipse.

Utiliza las características de Eclipse para ayudarle en varias tareas:

- Crear fácilmente proyectos Android.
- Crear interfaces y componentes basados en el framework Android.
- Gestionar emuladores.
- Gestionar y visualizar logs.
- Depurar su aplicación utilizando el SDK Android.
- Exportar sus aplicaciones.

Para comenzar, haga clic en Help - Install new software. A continuación, haga clic en el botón Add. Introduzca un nombre, la siguiente URL: <https://dl-ssl.google.com/android/eclipse/> y haga clic en OK.

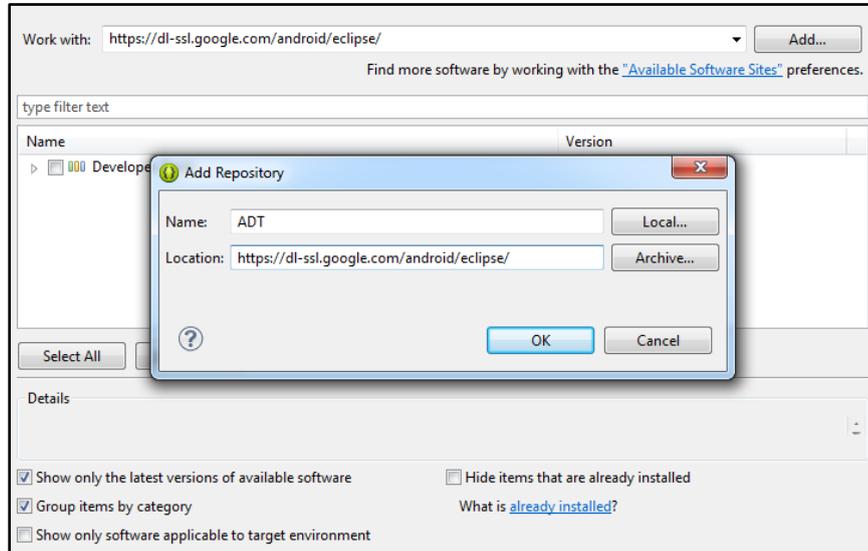


Figura 4.13: Pluggin ADT

Fuente: Android Developer - Eclipse IDE [18].

Pasados unos segundos, aparecerá la siguiente ventana:

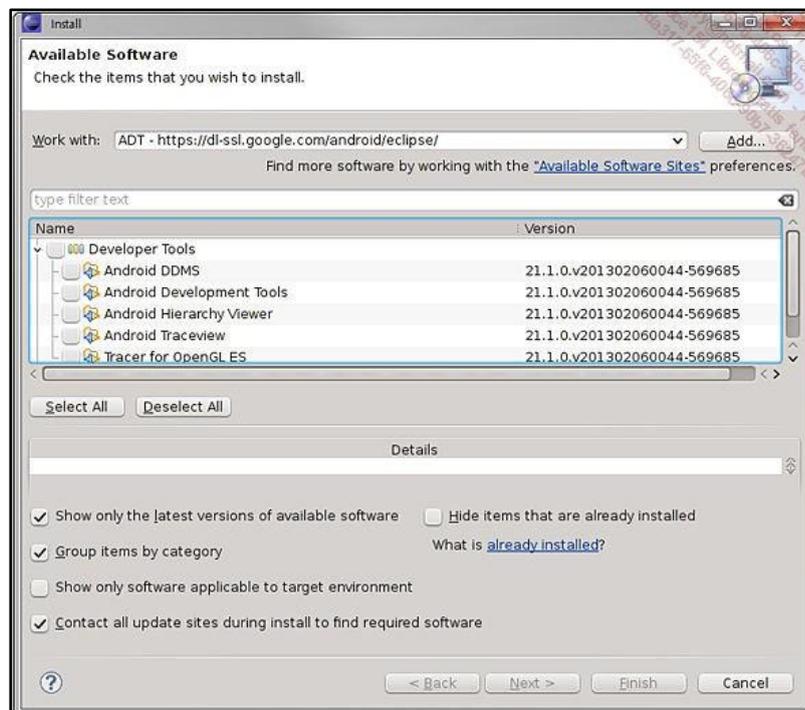


Figura 4.14: Opciones de pluggin ADT

Fuente: Android Developer - Eclipse IDE [18].

A continuación se enumeran los componentes que se instalarán:

- Android DDMS (Android Dalvik Debug Monitor Server): herramienta muy útil que integra la depuración y el control del teléfono o del emulador Android desde Eclipse.
- Android Development Tools: herramienta útil para el desarrollo de aplicaciones en Android.
- Android Hierarchy Viewer: herramienta que permite visualizar la arquitectura de sus vistas mediante un gráfico. Útil para la optimización de las vistas (véase el capítulo Creación de interfaces avanzadas - Optimizar sus interfaces).
- Android Traceview: permite mostrar el registro y los mensajes provenientes de su teléfono o emulador mediante un gráfico.
- Seleccione todos los elementos y, a continuación, haga clic en Next. Compruebe su selección mediante el botón Next. Acepte las condiciones y los términos de uso y, finalmente, haga clic en Finish.
- Cuando haya finalizado la instalación, es necesario reiniciar Eclipse.

4.4.5 Android

Android es una pila de software para dispositivos móviles que incluye un sistema operativo, middleware y aplicaciones esenciales. [19]

Android está basado en el kernel (núcleo) de Linux, el sistema operativo es open source (código abierto) y es utilizado en dispositivos móviles (teléfonos inteligentes y tablets); mediante las bibliotecas de Google creadas en lenguaje de programación Java, se puede controlar a los dispositivos.

Android Inc. una pequeña compañía de Palo Alto (California) la cual fue fundada en el año 2003, fue la encargada del desarrollo de Android en sus inicios; en el año 2005 Google compra esta firma; el 5 de noviembre de 2007 se crea el grupo Open Handset Alliance y se hace la presentación oficial de Android, aunque el funcionamiento de este se lo ve un año después, el 22 de Octubre de 2008 en el HTC Dream (G1), el primer smartphone (teléfono inteligente) con Android.

Los dispositivos móviles constituyen cada vez más una realidad que ofrece al usuario, en un mismo y reducido aparato, funciones de comunicación y procesamiento de datos que van mucho más allá de las simples llamadas telefónicas o la ejecución de aplicaciones básicas.

Aplicaciones

Android posee un conjunto de aplicaciones básicas, como un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos, y otros³⁰. Estas aplicaciones están escritas en el lenguaje de programación Java y sus archivos ejecutables tienen la extensión .apk (Android Package). [19]

Las aplicaciones son procesos Linux que se ejecutan sobre la máquina virtual VM (Virtual Machine) y se les asigna un ID de usuario único. Tienen la capacidad de cumplir con ciertas funcionalidades y además pueden utilizar las funcionalidades de otras aplicaciones.

Las aplicaciones hacen uso de las capas inferiores de la arquitectura Android (servicios, API, librerías).

Framework de Aplicaciones

Permite la reutilización de componentes facilitando el desarrollo para no empezar desde cero la creación de aplicaciones. Las principales API que esta capa posee son las siguientes: [19]

- **Administrador de Actividad: (Activity Manager)** Gestiona el ciclo de vida de las aplicaciones.
- **Administrador de Ventanas: (Window Manager)** Encargado de administrar las ventanas de las aplicaciones y utiliza la librería Surface Manager.
- **Administrador de Telefonía: (Telephone Manager)** Incluye las API relacionadas a las funcionalidades propias del teléfono.
- **Proveedor de Contenido: (Content Provider)** Permite que cualquier aplicación puede compartir sus datos con el resto de aplicaciones Android.
- **Sistema de vistas: (View System)** Proporciona algunos elementos (vistas, listas, botones, mosaicos, tamaños de ventana, etc.) necesarios para construir interfaces de usuario (GUI).
- **Administrador de Localización: (Location Manager)** Permite a las aplicaciones obtener información de localización y posicionamiento.
- **Administrador de Notificación: (Notification Manager)** Las aplicaciones comunican al usuario eventos (llamada entrante, mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc.) que ocurran durante su ejecución. Si ocurre una acción (Intent), como el contestar una llamada recibida esta se activa mediante un clic.
- **Servicio XMPP: (Extensible Messaging and Presence Protocol - Protocolo de Mensajería y Comunicación de Presencia)** Conjunto de funciones que utilizan este protocolo de intercambio de mensajes basado en XML.

Librerías

Android contiene un conjunto de librerías escritas en C/C++ utilizadas por los diversos componentes. [19]

Las librerías más importantes que existen son las siguientes:

- Administrador de Superficies: (Surface Manager) Encargada de la gestión de la pantalla y de las ventanas de aplicaciones activas.
- Framework de Medios: (Media framework) Proporciona los códecs necesarios que permitan la reproducción y grabación de varios formatos (vídeo, audio, imágenes estáticas y animadas, etc.).
- Lenguaje de consulta estructurado - Ligerito: (SQLite) Motor de base de datos relacional.
- Librería Abierta de Gráficos: (OpenGL/SL) Librería que permite el manejo de gráficos en 3D.
- Librería Simple de Gráficos: (SGL) Librería que permite el manejo de gráficos en 2D.
- FreeType: Permite la renderización de bitmaps y fuentes vectoriales.
- Herramientas Web: (WebKit) Incorpora un motor para el navegador web.
- Capa de Conexión Segura: (SSL) Se utiliza este protocolo Secure Sockets Layer, para comunicaciones seguras.
- Librerías C: (Libc) Incorpora las cabeceras y funciones según el lenguaje C.

Núcleo de Linux

Android se basa en la versión 2.6 de Linux para los servicios del núcleo del sistema como la seguridad, la gestión de memoria, gestión de procesos, la pila de red, y el modelo de controlador. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software. [19]

Archivo de Manifiesto

Toda aplicación contiene un archivo base llamado AndroidManifest.xml o “Archivo de Manifiesto”, ubicado en el directorio raíz del proyecto.

Este archivo proporciona información importante de la aplicación al sistema Android, esta información debe ser provista al sistema antes de poder ejecutar la aplicación antes mencionada; a continuación se lista algunas de las secciones más importantes que contiene el archivo de manifiesto: [19]

- El nombre del paquete.
- El mínimo nivel de la API para ejecutar la aplicación.
- Los permisos de usuario para la aplicación.

- Los componentes existentes en la aplicación.

Componentes de una Aplicación

Como ya se dijo una aplicación Android corre su propio proceso Linux, sin embargo el tiempo y el ciclo de vida de la misma lo determina el sistema en base a una combinación de parámetros como la disponibilidad de memoria, la prioridad del usuario o de acuerdo a las aplicaciones que se encuentren ejecutándose en ese momento. [19]

Cada componente de una aplicación cumple con una función específica y tiene ciclos de vida diferentes, los principales componentes son:

- Actividades
- Servicios
- Proveedores de contenido
- Receptores de difusión
- Intención
- Vistas

Estados de los Procesos

Cada proceso es creado por una aplicación cuando esta se ejecute. Los estados de los procesos definidos en Android son: [19]

- Procesos en primer plano: (Active process) Proceso que contiene una actividad en pantalla. Estos procesos solo serán eliminados como último recurso si el sistema lo requiere por necesidad de memoria.
- Procesos visibles: (Visible process) Contiene una actividad que está en segundo plano, es decir son procesos visibles pero que se encuentran inactivos. Estos procesos solo serán eliminados en circunstancias extremas, para permitir que sigan corriendo los procesos en primer plano.
- Procesos de servicio: (Started service process) Contiene un servicio que fue iniciado por el método `startService()`. Estos procesos no interactúan directamente con el usuario pero son importantes porque realizan tareas necesarias para el correcto funcionamiento de la aplicación, razón por la cual el sistema por lo general no va a eliminar este tipo de procesos a menos que sea totalmente necesario para mantener vivos los dos tipos de procesos antes mencionados.
- Procesos en segundo plano: (Background process) Proceso que contiene una actividad que no es visible para el usuario. Estos procesos se eliminan para poder liberar memoria en casos de falta de recursos.
- Procesos vacíos: (Empty process) Proceso que no contiene ningún componente, pero es utilizado para mejorar el tiempo de activación de la aplicación cuando es

iniciada nuevamente puesto que usa la memoria caché. Este tipo de procesos se eliminan para poder liberar memoria en casos de falta de recursos.

La Figura 4.27 muestra el árbol de prioridad con que se determina el orden de finalización de la aplicación.

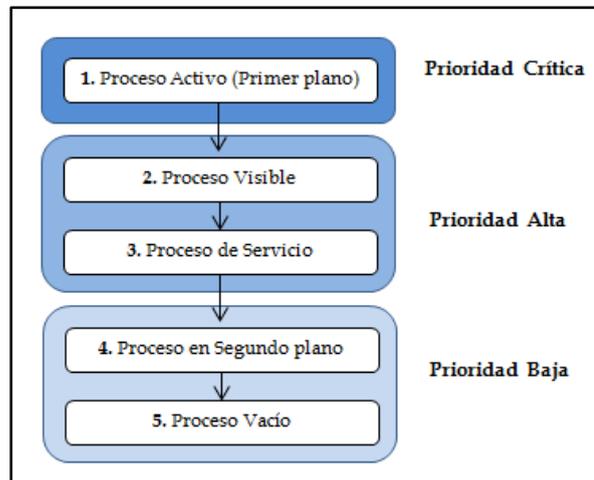


Figura 4.15: Árbol de prioridad de finalización de los procesos
Fuente: Plataformas orientadas al desarrollo móvil [19]

Seguridad y Permisos

Android proporciona un entorno seguro, es por eso que todas las aplicaciones vienen por defecto sin permisos para realizar ciertas tareas. [19]

Para establecer permisos en una aplicación, se necesita declarar uno o más elementos `<uses-permission>` en el archivo de manifiesto, los cuales especifican el tipo de permiso que se desea autorizar.

- Algunos de los atributos que se definen en el elemento `<uses-permission>` son los siguientes:
- `android: name`
- `android: label`
- `android: permissionGroup`
- `android: protectionLevel`
- `android: description`
- `android: icon`

Algunos de los permisos utilizados en las aplicaciones, se especifican a continuación:

- `INTERNET`: Permite a las aplicaciones abrir conexiones de red.

- Filtrar los mensajes por nivel (verbose, debug, error, etc.).
- Guardar los logs seleccionados en un archivo.

Puede acceder a los logs por línea de comandos mediante la herramienta adb:

```
./adblogcat
```

4.4.6 Micro Controlador

Un microcontrolador (abreviado μC , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida. [20]

Opciones para microcontroladores

Launchpad MSP430

El Texas Instruments MSP430 es muy similar al chip microcontrolador ATmega de Atmel. Las diferencias notables incluyen un precio muy bajo, así como algunas mejoras interesantes para un bajo consumo de energía. La forma más fácil de familiarizarse es con una tarjeta de desarrollo TI LaunchPad. [20]

La principal diferencia entre LaunchPad y Arduino es el costo. Mientras que un Arduino Uno te costará unos \$30 dólares, y un Leonardo \$25, el LaunchPad MSP430 sólo \$10 directamente de TI o de un distribuidor importante (e incluye un cable USB).

Aunque el chip MSP430G2553, mismo que utiliza la LaunchPad, sólo tiene 14 pines I/O y 16 K de memoria de programa, en comparación con el ATmega328 del Uno (con 32 K y 20 pines I/O), quizás sea todo lo que necesitas para un proyecto en particular.

Para una generación de makers acostumbrados al fácil uso del IDE de Arduino, el viejo entorno de desarrollo basado en Eclipse del MSP430 parecía demasiado complicado y difícil de usar. Sin embargo, la nueva plataforma de prototipado de código abierto Energia ha cambiado todo eso.

PICAXE

La casi omnipresencia de la plataforma Arduino ha llevado a muchos sistemas, que no son compatibles con el software, a ser por lo menos compatibles físicamente. El Shield Base PICAXE-28 \times 2 replica la forma de Arduino, estableciendo compatibilidad con los cientos de shields Arduino existentes en el mercado. [20]

Wiring

El gran éxito de Arduino le ha robado un poco de amor y atención a la tarjeta Wiring y su entorno de programación. Probablemente se merece más. La tarjeta está basada en el mismo entorno de desarrollo derivado de Processing del que proviene Arduino — aunque por ahora es una rama un tanto diferente del árbol de la familia — y debe sentirse familiar para cualquiera que esté acostumbrado al IDE de Arduino. [20]

El entorno de programación de Wiring es compatible con cualquier hardware basado en la serie de procesadores AVR de Atmel, no sólo las propias tarjetas Wiring. Y esto incluye la línea de Arduino.

La última placa, la Wiring S, es similar al viejo Arduino Diecimila, pero con un procesador más grande. Al igual que el Shield Base Picaxe, emparejando la tarjeta con un Shield Play Wiring S hace compatible a sus pines con la forma de Arduino, para que puedas volver a utilizar tus shields Arduino con tus tarjetas Wiring.

Netduino

La Netduino también adopta la forma de Arduino — puedes conectar la mayoría de los shields de Arduino existentes. Pero ahí es donde termina la similitud. Hay muchas tarjetas Netduino disponibles, y a diferencia de las tarjetas que hemos conocido hasta ahora, que han sido en torno a microcontroladores de 8 o 16 bits de un tipo u otro, la Netduino es una placa basada en un ARM Cortex construida alrededor del microcontrolador STMicro STM32Fx de 32 bits. [20]

El sistema operativo de la tarjeta es el NET Micro Framework. Estas tarjetas son programables en C#, directamente desde Microsoft Visual C# Express 2010, y son extremadamente potentes y flexibles. Los desarrolladores de C# en plataformas que no sean Windows no están del todo abandonados, ya que existe cierto apoyo para OS X y Linux.

Arduino

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa de circuito impreso que contiene un microcontrolador 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 mundo analógico con el digital controlando, sensores, alarmas, sistemas de luces, motores, sistemas comunicaciones y actuadores físicos. [20]

Tabla comparativa microcontroladores

Tabla 4.1 Tabla comparativa microcontroladores.
Fuente: Guía de aprendizaje para microcontroladores [20]

	LAUNCHPAD MSP430	PICAXE	WIRING	NETDUINO	ARDUINO
Voltaje operativo	5 V	5 V	5 V	5 V	5 V
Voltaje de entrada recomendado	7-10 V	7-9 V	7-10 V	7-12 V	7-12 V
Voltaje de entrada límite	6-20 V	6-20 V	6-20 V	6-20 V	6-20 V
Contactos de entrada y salida digital	SI	SI	SI	SI	SI
Contactos de entrada analógica	SI	SI	SI	SI	SI
Intensidad de corriente	30 mA	40 mA	20 mA	30 mA	40 mA
Memoria Flash	16KB	32KB	128KB	32KB (4KB reservados para el bootloader)	128KB (4KB reservados para el bootloader)
SRAM	1 KB	2 KB	8 KB	2 KB	8 KB
EEPROM	512 bytes	1 KB	4 KB	1 KB	4 KB
Frecuencia de reloj	16 MHz	16 MHz	16 MHz	16 MHz	16 MHz

4.4.7 Elección de Arduino y su modelo

Se eligió trabajar con Arduino ya que se basa en una plataforma denominada open hardware que reúne en una pequeña placa de circuito impreso (PCB) los componentes necesarios para conectar con el mundo exterior y hacer funcionar un microcontrolador Atmega. Al ser Open-Hardware, tanto su diseño como su distribución son libres. Es decir, puede utilizarse sin inconvenientes para desarrollar cualquier tipo de proyecto sin tener que adquirir ningún tipo de licencia. La placa es de muy fácil montaje, con pocos componentes periféricos al microcontrolador, a continuación se muestra una tabla comparativa de arduino.

Tabla comparativa

La siguiente tabla compara las características básicas de las tarjetas Arduino mencionadas anteriormente en ésta guía:

Tabla 4.2 Tabla comparativa arduino.
Elaborado por: Guía de aprendizaje para microcontroladores [20]

Modelo	Voltaje De Operación	Voltaje de Alimentación	Flash [KB]	SRAM [KB]	I/O digitales /PWM	Pines Análogos (I/O)	UART	Compatibilidad Con Shields
Uno	5v	7-12v	32	2	14/6	6/0	1	Excelente
Pro	5v	5-12v	32	2	14/6	6/0	1	Excelente
Leonardo	5v	7-12v	32	2.5	20/7	12/0	1	Moderada
Micro	5v	5v	32	2.5	20/7	12/0	1	N/A
Mega	5v	7-12v	256	8	54/15	16/0	4	Moderada

En cuanto al modelo a utilizarse será Arduino Mega 2560 R3, debido a que tiene mayor cantidad de pines, y un mayor espacio en memoria, requerimientos necesarios para la programación, uso que se va a dar al hardware y la compatibilidad con Arduino Ethernet Shield R3

Arduino Mega 2560

El Arduino Mega 2560 es una placa electrónica basada en el chipset Atmega 2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertas seriales), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB, adaptador de CA o la batería a CC para empezar. [21]

Resumen

Tabla 4.3 Resumen Características Arduino Mega 2560
Fuente: Arduino Mega 2560 [21]

Microcontroladores	Atmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Pines digitales I / O	54 (de las cuales 15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente DC por Pin I / O	40 mA
Corriente DC de 3.3V Pin	50 mA
Memoria Flash	256 KB de los cuales 8 KB utilizado por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz

Energía

Arduino Mega puede ser alimentado a través de conexión USB o con una fuente de alimentación externa. [21]

La alimentación energía (no USB) externa puede venir con un adaptador de CA a CC (pared) o la batería. El adaptador se puede conectar a un enchufe de 2.1mm centro-positivo en el conector de alimentación de la placa. Los cables desde una batería se pueden insertar en los cabezales de pin GND y Vin del conector de alimentación de la placa arduino.

Memoria

Atmega 2560 tiene 256 KB de memoria flash para el almacenamiento de código (de los cuales 8 KB se utiliza para el gestor de arranque), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la librería EEPROM). [21]

Entrada y Salida

Cada uno de los 54 pines digitales en Aruino Mega se puede utilizar como una entrada o salida, funcionan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA. Además, algunos pines tienen funciones especializadas: [21]

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Estos pines admiten la comunicación SPI utilizando la librería SPI . Los pines SPI también se desglosan en la cabecera ICSP, que es físicamente compatible con el Uno.

La placa Mega 2560 tiene 16 entradas analógicas. Por defecto a 5 voltios con respecto a una conexión a tierra.

Comunicación

El Arduino Mega2560 tiene una serie de pines para comunicarse con un ordenador, otro Arduino u otros microcontroladores. El Atmega 2560 ofrece cuatro hardware UART para TTL (5V) de comunicación serie. [21]

Una biblioteca Software Serial permite la comunicación en serie en cualquiera de los pines digitales del Mega 2560.

El Atmega 2560 también soporta la comunicación SPI.

Programación

El Arduino Mega se puede programar con el software de Arduino.

4.4.8 Modulo Ethernet

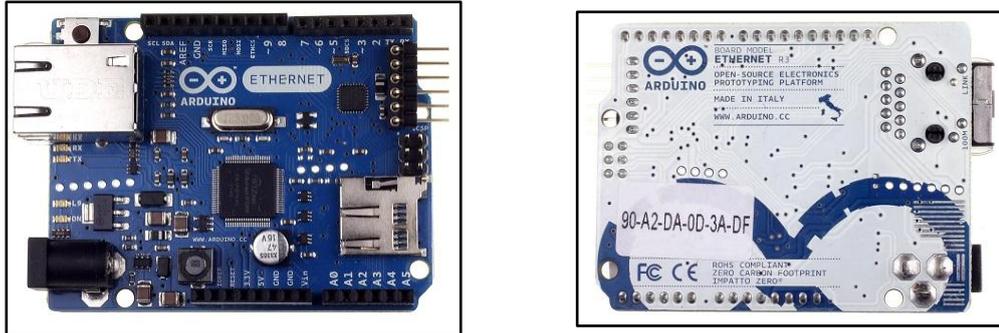


Figura 4.17: Arduino Ethernet Placa vista frontal - visión trasera
Fuente: Arduino Ethernet [22]

Visión de conjunto

Arduino Ethernet es una placa electrónica basada en el chipset ATmega328. Cuenta con 14 pines digitales de entrada / salida, 6 entradas analógicas, un oscilador de cristal de 16 MHz, un RJ45 de conexión, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. [22]

Nota: Los pines 10, 11, 12 y 13 están reservados para la conexión con el módulo Ethernet y no debe ser utilizado de otra manera. Esto reduce el número de pines disponibles a 9, con 4 disponibles como salidas PWM.

Una alimentación opcional sobre el módulo Ethernet se puede agregar a la tarjeta también.

El módulo Ethernet se diferencia de otras placas en que no tiene un chip integrado controlador de USB a serie, pero tiene una interfaz Wiznet Ethernet. Esta es la misma interfaz que se encuentra en la placa Ethernet.

Un lector de tarjetas microSD integrada, que se puede utilizar para almacenar archivos para servir a través de la red, es accesible a través de la Biblioteca SD. El Pin 10 está reservado para la interfaz Wiznet, SS para la tarjeta SD está en el pin 4.

Cuenta con soporte para rearme automático, permitiendo bocetos que se cargan sin necesidad de pulsar el botón de reinicio en la placa. Cuando se conecta a un adaptador USB a Serial, el Arduino Ethernet es alimentado desde el adaptador.

Resumen

Tabla 4.4 Resumen Arduino Ethernet
Fuente: Arduino Ethernet [22]

Microcontroladores	ATmega328
Tensión de funcionamiento	5V
Plug Voltaje de entrada (recomendado)	7-12V
Tensión enchufe de entrada (límites)	6-20V
Voltaje de entrada PoE (límites)	36-57V
Digital pines I / O	14 (de las cuales 4 proporcionan salida PWM)
Arduino prendedores reservados:	
	10 a 13, utilizado para SPI
	4 utilizado para la tarjeta SD
	2 W5100 de interrupción (cuando puente)
Pines de entrada analógica	6
Corriente DC por Pin I / O	40 mA
Corriente DC de 3.3V Pin	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz
Ethernet Controller Embedded W5100 TCP / IP	
Power Over Ethernet listo magnética Jack	
Tarjeta Micro SD, con traductores de voltaje activos	

Energía

La placa también puede ser alimentado a través de una fuente de alimentación externa, una potencia opcional a través de Ethernet (PoE módulo), o mediante el uso de un conector Serial / cable USB FTDI. [22]

Memoria

El ATmega328 tiene 32 KB (con 0,5 KB utilizan para el gestor de arranque). También cuenta con 2 KB de SRAM y 1 KB de EEPROM (que puede ser leído y escrito con la librería EEPROM). [22]

Entrada y Salida

Cada uno de los 14 pines digitales en el tablero de Ethernet se puede utilizar como una entrada o salida, usando `pinMode ()`, `digitalWrite ()`, y `digitalRead ()` funciones. Funcionan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 kOhm. Además, algunos pines tienen funciones especializadas: [22]

- Serial: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir datos en serie (TX) TTL.
- Interrupciones externas: 2 y 3 Estos pines pueden configurarse para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Ver el `attachInterrupt ()` función para más detalles.
- PWM: 3, 5, 6, 9, 10 y proporcionar una salida PWM de 8 bits con el `analogWrite ()` función.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) Estos pines admite la comunicación SPI utilizando la librería SPI .
- LED: 9. Hay un LED incorporado conectado al pin digital 9. Cuando el pasador es de alto valor, el LED está encendido, cuando el pasador es bajo, es apagado. En la mayoría de otras placas Arduino, este LED se encuentra en el pin 13. Es en el pin 9 en el tablero de Ethernet porque pin 13 se utiliza como parte de la conexión SPI.

Comunicación

Arduino Ethernet tiene una serie de instalaciones para comunicarse con un ordenador, otro Arduino u otros microcontroladores. [22]

Una biblioteca Software Serial permite la comunicación en serie en cualquiera de los pines digitales del Uno.

El ATmega328 también soporta la comunicación TWI y SPI. El software de Arduino incluye una biblioteca de alambre para simplificar el uso del bus TWI; consulte la documentación para obtener más información. Para la comunicación SPI, utilice la librería SPI.

La junta también se puede conectar a una red cableada a través de Ethernet. Cuando se conecta a una red, tendrá que proporcionar una dirección IP y una dirección MAC. La Biblioteca Ethernet es totalmente compatible.

El lector de tarjetas microSD a bordo es accesible a través de la Biblioteca SD. Cuando se trabaja con esta biblioteca, SS está en el pin 4.

Características Físicas

La longitud máxima y la anchura de la placa Ethernet son 2,7 y 2,1 pulgadas, respectivamente, con el RJ45 jack conector y el poder que se extiende más allá de la dimensión anterior. Cuatro orificios de los tornillos que la Junta pudiera estar unido a una superficie o caso. Tenga en cuenta que la distancia entre los pines digitales 7 y 8 es de 160 milésimas de pulgada (0,16 "), no un múltiplo par de la separación de 100 milésimas de pulgada de los otros pasadores. [22]

4.4.9 Pantalla de Cristal Líquido (LCD)

Una pantalla de cristal líquido permite visualizar ciertos caracteres. Para funcionar correctamente una pantalla LCD debe de estar conectada a un circuito impreso en el que estén integrados los controladores del display y los pines para la conexión del display. [23]

Tiene un consumo de energía menor a 5mA y son ideales para dispositivos que requieran una visualización pequeña o media. En la tabla se muestra la asignación de pines en una pantalla LCD 16x2.

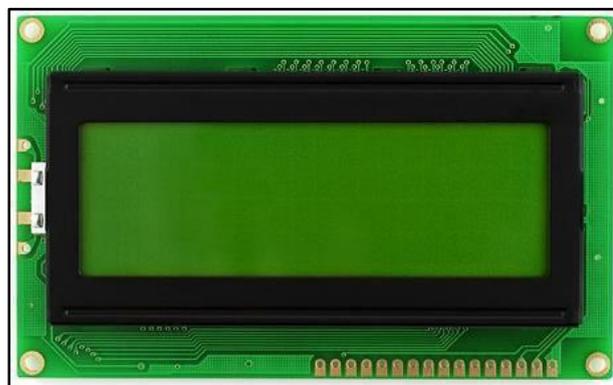


Figura 4.18: Pantalla de Cristal Líquido – LCD 20x4[12]

Fuente: Construcción de un prototipo de control de un juego de luces y la ventilación de una discoteca [23].

Descripción de pines LCD

Los pines 1 y 2, son utilizados para la alimentación del módulo LCD. La tensión utilizada es de 5 voltios. [23]

El pin 3 se utiliza para ajustar el contraste de la pantalla LCD. Por medio de un potenciómetro se regula la intensidad de los caracteres, a mayor tensión mayor intensidad. Se suele utilizar un potenciómetro de unos 10 o 20 k Ω , que regulara la misma tensión que se utiliza para la alimentación.

El pin 4 se utiliza para indicar al bus de datos si la información que le llega es una instrucción o por el contrario es un carácter. Si RS=0 indicara que en el bus de datos hay presente una instrucción, y si RS=1, indicará que tiene un carácter alfanumérico.

El pin 5 es el de escritura o lectura. Si está a 0 el módulo escribe en la pantalla el dato que haya en el bus de datos, y si está a 1 leeremos lo que hay en el bus de datos.

El pin 6 es el encargado para que el módulo LCD funcione correctamente, o por el contrario no acepte órdenes de funcionamiento. Cuando E=0 no se podrá utilizar el display y cuando E=1 se podrá transferir datos y realizar las demás operaciones.

Los pines del 7 al 14 son los del bus de datos.

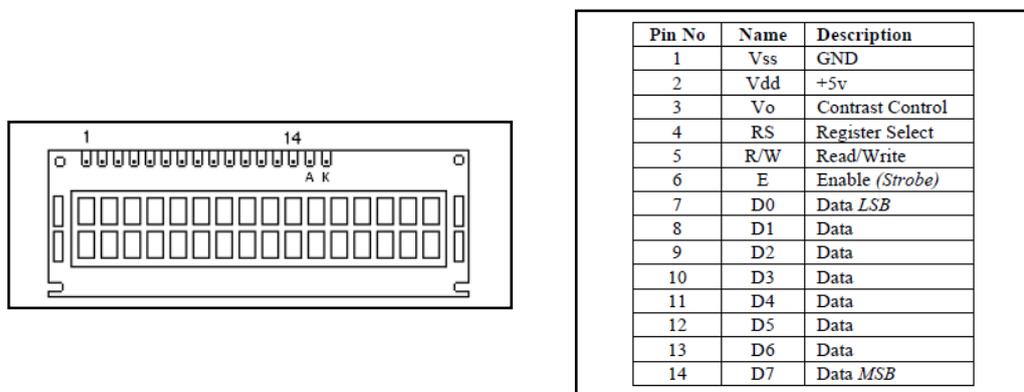


Figura 4.19: Descripción de pines LCD

Fuente: Construcción de un prototipo de control de un juego de luces y la ventilación de una discoteca [23].

4.4.10 Sensor LM35

El LM35 es un sensor de temperatura con una precisión calibrada de 1°C. Su rango de medición abarca desde -55°C hasta 150°C. La salida es lineal y cada grado centígrado equivale a 10mV, por lo tanto: [24]

- $150^{\circ}\text{C} = 1500\text{mV}$
- $-55^{\circ}\text{C} = -550\text{mV}$

Características

Sus características más relevantes son: [24]

- Está calibrado directamente en grados Celsius.
- La tensión de salida es proporcional a la temperatura.
- Tiene una precisión garantizada de 0.5°C a 25°C .
- Opera entre 4 y 30 voltios de alimentación.
- Baja impedancia de salida.
- Baja corriente de alimentación ($60\mu\text{A}$).
- Bajo coste.

Destacables

El LM35 no requiere de circuitos adicionales para calibrarlo externamente. La baja impedancia de salida, su salida lineal y su precisa calibración hace posible que este integrado sea instalado fácilmente en un circuito de control. Debido a su baja corriente de alimentación se produce un efecto de auto calentamiento muy reducido. Se encuentra en diferentes tipos de encapsulado, el más común es el TO-92, utilizada por transistores de baja potencia. [24]

Diagrama de conexión

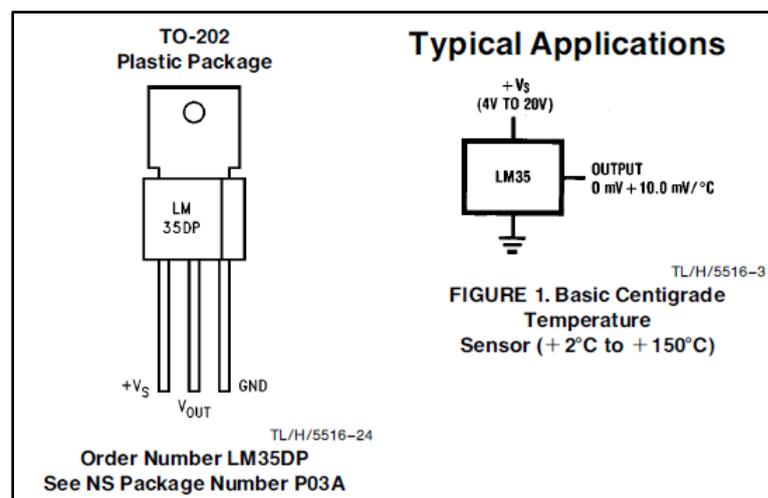


Figura 4.20: Diagrama de conexión sensor LM35

Fuente: Sistema de control de temperatura mediante un sistema telemétrico [24]

4.5 Diseño e implementación de un circuito para el monitoreo y control de la climatización interna de la Central de Datos.

En base a la investigación bibliográfica se determinó que los métodos de solución a la falta de medios de detección de fallas en la red son el Protocolo SNMP para conocer el estado lógico y operativo de la red, además de un Hardware para la detectar temperatura al interior del Data Center y con esto evitar que por causas de sobrecalentamiento los equipos sufran daños físicos.

En cuanto al protocolo SNMP se determinó que la plataforma operativa sobre la cual debe desarrollarse la aplicación de monitoreo sea sobre dispositivos móviles con Android, debido a la gran cantidad de dispositivos existentes y que éste dispositivo dispone el administrador del área de Sistemas y Comunicaciones del Hospital, con lo cual podrá realizar consultas y conocer el estado operativo y funcional de toda su red con la aplicación, independientemente si se encuentra en la red interna del hospital o fuera de ella, con una conexión a través de sus IP públicas.

Para la parte de la detección de la climatización al interior del Data Center será necesario elaborar un hardware utilizando la placa Arduino Mega en conjunto con los módulos Ethernet, relé y un sensor de temperatura interactuando entre sí de forma lógica para que el Hardware sea capaz de obtener la temperatura, compararla con un valor fijado y en el caso de superar dicho valor encender el aire acondicionado con lo cual se regulará la temperatura a valores normales y una vez obtenida la temperatura deseada apagarlos, de esta manera existirá un control automático y además estos eventos podrán ser conocidos por el administrador a través de publicación de mensajes en una cuenta de twitter que estará en el dispositivo móvil que pertenece al administrador.

Diagrama pictórico control y monitoreo climatización

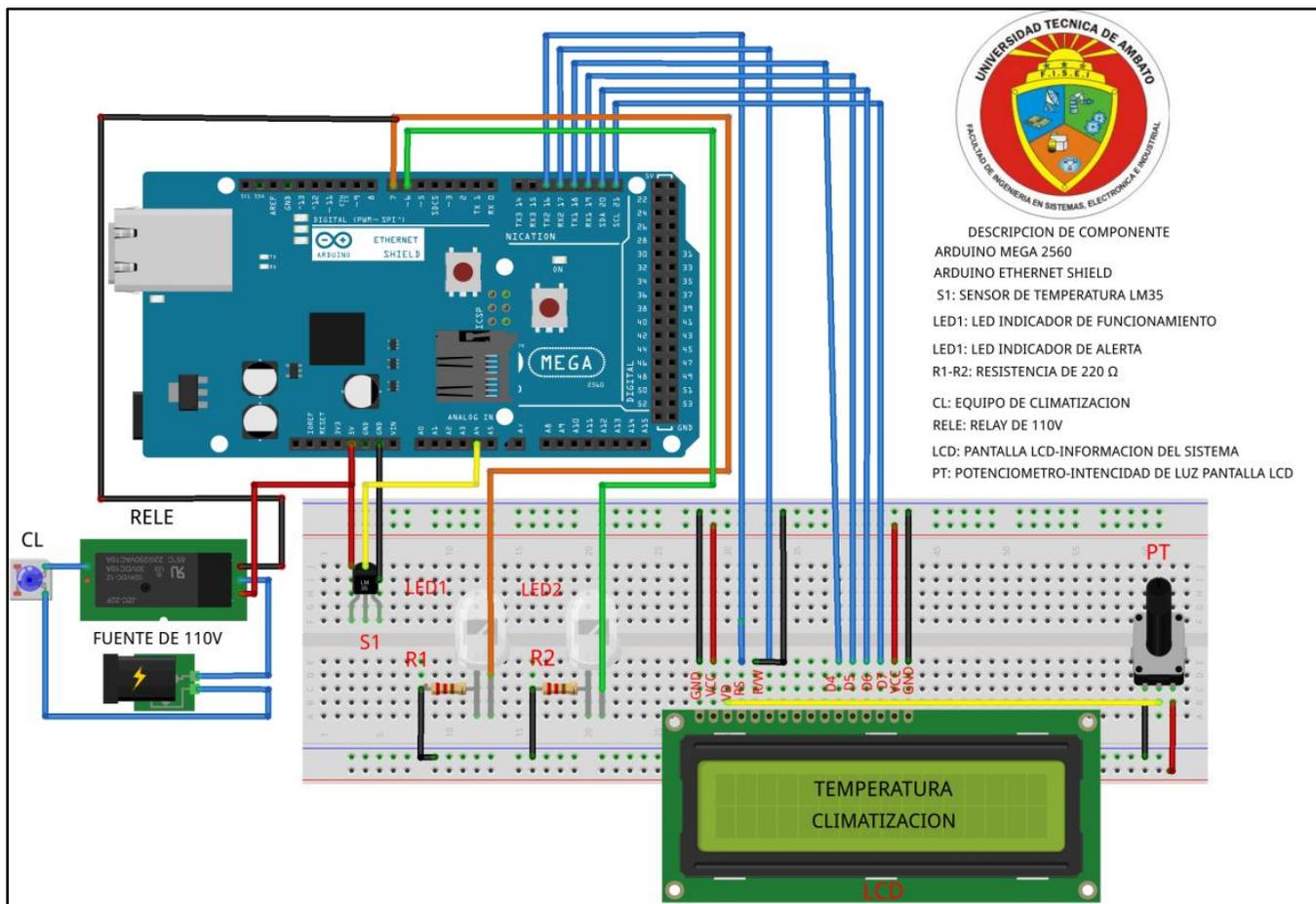


Figura 4.22: Diagrama pictórico hardware climatización
 Elaborado por: Alex Francisco Rodríguez

Diagrama de flujo control y monitoreo climatización

El diagrama de flujo representado a continuación describe cada una de las etapas de funcionamiento del sistema de control y monitoreo implementado como se puede observar en la figura 4.24.

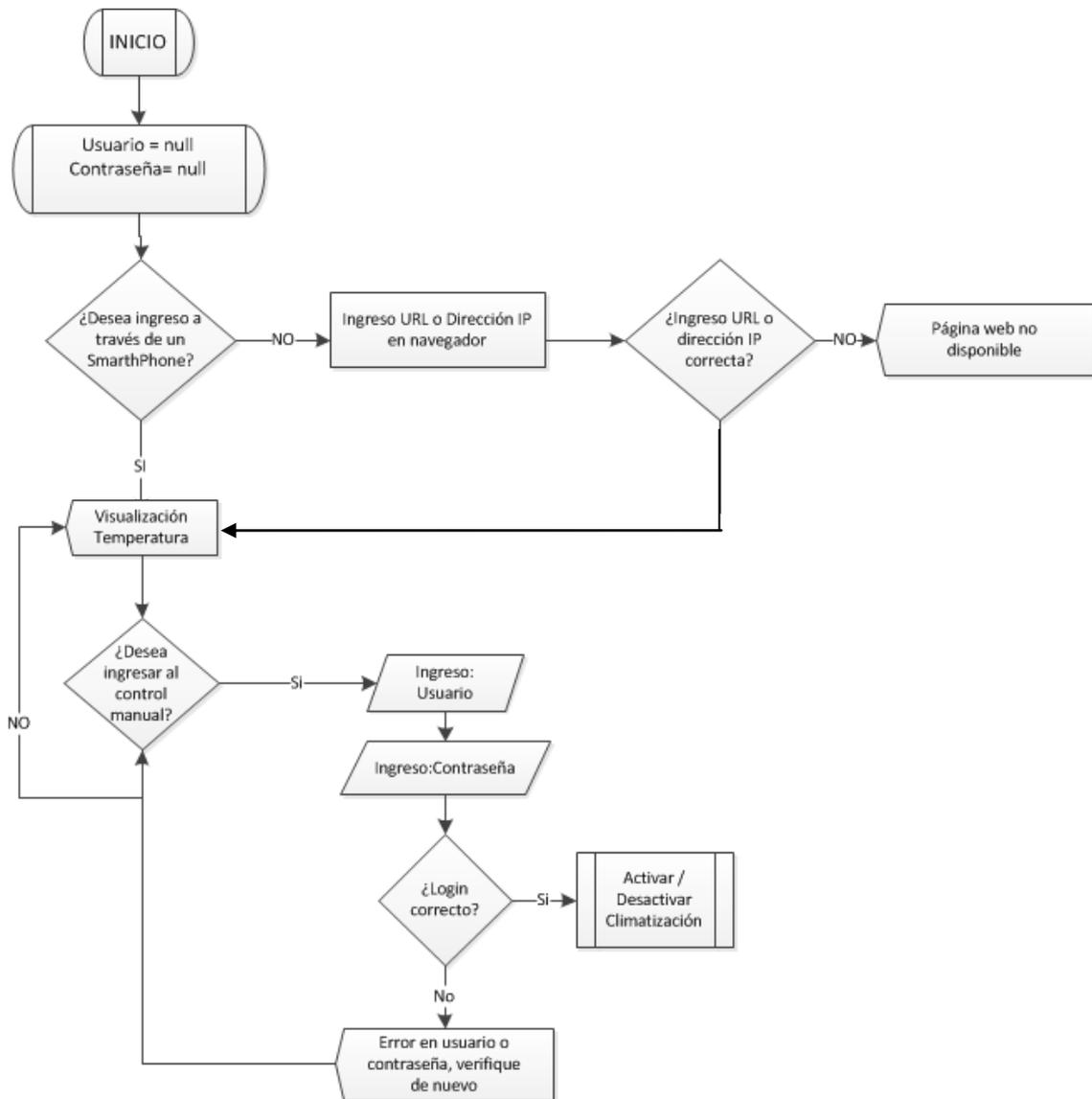


Figura 4.24: Diagrama de flujo Climatización
Elaborado por: Alex Francisco Rodríguez

Elemento necesarios para la implementación son los siguientes:

Equipos:

- Multímetro
- Taladro

Listado de dispositivos:

- Módulo Ethernet Shield
- Microcontrolador Arduino Mega 2560 R3
- Pantalla LCD
- Módulo de Relés

Listado de elementos:

- | | |
|-------------------------------|------------------------------|
| • Chasis | • Diodos |
| • Interruptor | • Potenciómetro |
| • Conector hembra en ocho | • Cautín |
| • Conector hembra plug dc | • Luz piloto |
| • Porta fusible | • Broca 1/8 |
| • Fusible – 15A | • Sensor de temperatura LM35 |
| • Jack bananas hembra y macho | • Tornillos y tuercas |
| • Cable USB A-B | • Cable de poder en ocho |
| • Cargador AC/DC plug macho | • Cable de Red ETHERNET |

Control y Monitoreo de la Climatización Data Center

Programación

Arduino es una plataforma de código y hardware abierto, es decir, se puede acceder a todo aspecto del funcionamiento circuital y algorítmico de las placas. Arduino también ofrece un entorno de desarrollo integrado (IDE) con funciones preestablecidas que reducen la lógica a lectura de entradas, control de tiempos y salidas de una manera semántica e intuitiva, es por eso que lo convierte en una herramienta perfecta para el desarrollo de aplicaciones integradas.

Previo el desarrollo del sistema se realizó un conjunto de pruebas para determinar la eficiencia y veracidad de los datos a obtenerse por parte de todos los dispositivos que integran el Sistema de Control y Monitoreo del Data Center, iniciando por la Climatización, la cual se describe a continuación:

Inicialmente se realizó una prueba de comprobación entre la placa Arduino Mega y el sensor de temperatura, para verificar la fiabilidad de la información obtenida (Anexo A).

En segunda instancia se realizó una prueba entre la placa, el sensor y la pantalla LCD, basándonos en ejemplos existentes en el software, donde se realizó una conexión entre los elementos a través de una protoboard, lo cual permitió conocer el funcionamiento y la forma de envío de los datos obtenidos por el sensor e imprimirlos en la pantalla LCD. (Anexo B)

Posteriormente se realizaron pruebas con el módulo Ethernet, el módulo Arduino, el sensor de temperatura y la pantalla LCD; donde previamente se configuró una cuenta y una autenticación para que el módulo envíe un mensaje de saludo (“Hello world I’m arduino”) hacia Tweeter, la cual es posteada en la cuenta de: “climatizaciónHMNSM” con lo cual verificamos la conexión entre el módulo y tweeter, para posteriormente desarrollar la aplicación donde se compare un valor X de la temperatura y en caso de ser superado enviar un mensaje de alerta.

Inicialmente se realiza un diagrama de conexión del Hardware de Control y Monitoreo de la climatización al interior de la red Lan del Hospital, el cual se detalla a continuación en la figura 4.22

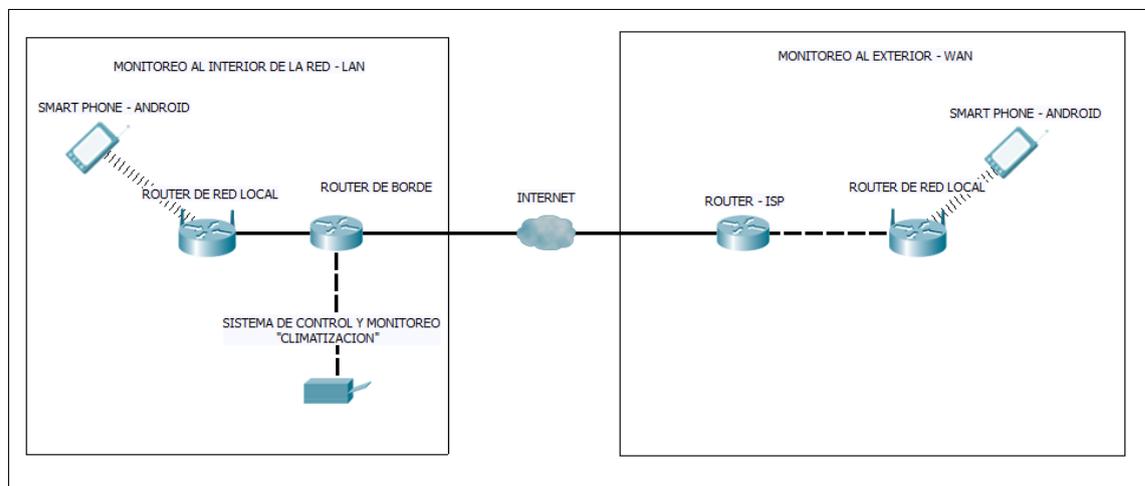
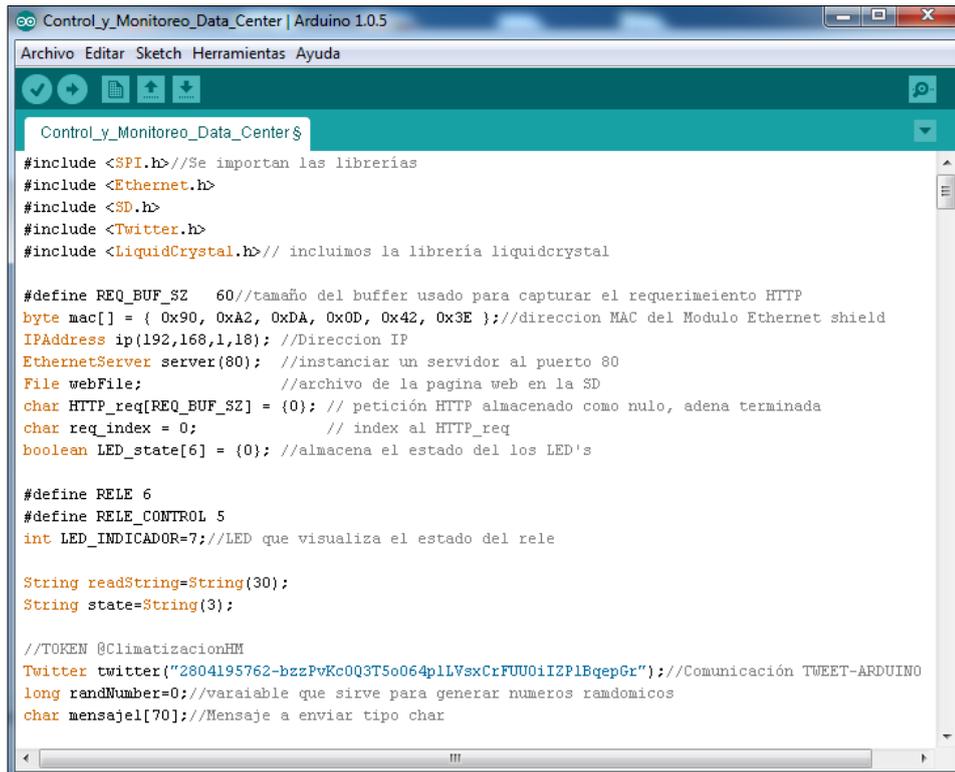


Figura 4.23: Diagrama de bloques del sistema de climatización Data Center
Elaborado por: Alex Francisco Rodríguez



```
Control_y_Monitoreo_Data_Center$
#include <SPI.h> //Se importan las librerías
#include <Ethernet.h>
#include <SD.h>
#include <Twitter.h>
#include <LiquidCrystal.h> // incluimos la libreria liquidcrystal

#define REQ_BUF_SZ 60 //tamaño del buffer usado para capturar el requerimeiento HTTP
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x42, 0x3E }; //direccion MAC del Modulo Ethernet shield
IPAddress ip(192,168,1,18); //Direccion IP
EthernetServer server(80); //instanciar un servidor al puerto 80
File webFile; //archivo de la pagina web en la SD
char HTTP_req[REQ_BUF_SZ] = {0}; // petición HTTP almacenado como nulo, adena terminada
char req_index = 0; // index al HTTP_req
boolean LED_state[6] = {0}; //almacena el estado del los LED's

#define RELE 6
#define RELE_CONTROL 5
int LED_INDICADOR=7; //LED que visualiza el estado del rele

String readString=String(30);
String state=String(3);

//TOKEN @ClimatizacionHM
Twitter twitter("2804195762-bzzPvKc0Q3T5o064p1LVsxCrFUU0iI2P1BqepGr"); //Comunicación TWEET-ARDUINO
long randomNumber=0; //varaiaable que sirve para generar numeros randomicos
char mensaje[70]; //Mensaje a enviar tipo char
```

Figura 4.5: Programación Control y Monitoreo de la climatización a través de Arduino
Elaborado por: Alex Francisco Rodríguez

Una vez concluido estos pasos y de haber verificado su correcto funcionamiento se pudo implementar el “Sistema de Control y Monitoreo de la Climatización”, en la cual se utilizaron distintas librerías de Arduino que sirve para proporcionar funcionalidad extra en nuestros sketches (ofreciendo la posibilidad de manipular datos en diferentes formatos, de gestionar protocolos de comunicación diversos, de interactuar con hardware variado, etc.).

Hay que tener presente que para poder utilizar las funciones que proveen, la librería en cuestión primero se ha de importar, o bien mediante la opción “Sketch”->”Import library” del menú del IDE, o bien incluyendo al principio del código del sketch la sentencia #include pertinente.

A continuación, se hará un breve resumen de las librerías oficiales que se instalaron junto con el IDE de Arduino.

Librería Liquid Crystal

Permite controlar pantallas de cristal líquido (LCDs) basadas en el chip HD44780. Este modelo de chip se encuentra en la mayoría de LCDs. La librería puede trabajar tanto en el modo 4-bit como en 8-bit (es decir: puede utilizar tanto 4 como 8 líneas de datos).

Librería SD

Permite leer y escribir datos en una tarjeta SD (o microSD) acoplada al zócalo Arduino Ethernet Shield. Las tarjetas SD son muy útiles para almacenar ficheros tales como audio, vídeo o imágenes, o bien datos textuales obtenidos de diferentes sensores, ya que ofrecen un sistema de grabación mucho mayor que la memoria EEPROM.

La librería SD puede utilizar tarjetas de tipo SDSC o SDHC. La comunicación entre la tarjeta SD y el microcontrolador se establece mediante SPI a través de los pines digitales 11, 12, 13 de la placa Arduino MEGA, además de otro pin usado como SS (pin de activación), el cual suele ser el 10.

Librería Ethernet

Permite conectar el Arduino Ethernet Shield a una red Ethernet (TCP/IP). Se puede configurar para que la placa actúe como servidor (es decir, que de forma permanente e ininterrumpida escuche y acepte peticiones de otros dispositivos de la red que soliciten algún tipo de servicio o dato ofrecido por ella) o bien como cliente (es decir, que sea la placa la que solicite puntualmente esos servicios o datos a otro dispositivo de red). Esta librería soporta hasta un total cuatro conexiones concurrentes.

Librería SPI

Permite comunicar mediante el protocolo SPI la placa Arduino (que actúa siempre como “maestro”) con dispositivos externos (que actuarían como “esclavos”). Tal como ya se ha comentado, la comunicación se establece mediante los pines MOSI (línea de envío de datos del maestro al esclavo, ubicada en la placa Arduino en el pin digital 11), MISO (línea de envío de datos del esclavo al maestro, ubicada en el pin digital 12), SCK (línea de reloj, ubicada en el pin digital 13) y el pin SS (línea de selección del esclavo, ubicada en el pin digital 10).

Diseño del sitio web basado en CSS y Ajax

Pantalla de Login

Para el diseño de la página web se ha utilizado el software “SUBLIME TEXT” el cual es un editor de texto y editor de código fuente con una interfaz muy amigable que facilita el diseño del sitio. El diseño de esta página consta de tres partes fundamentales HTML, HEAD Y BODY, de los cuales se realiza una breve descripción a continuación:

<HTML>: Limitan el documento e indica que se encuentra escrito en este lenguaje. En este caso la versión utilizada es la más avanzada, la cual es HTML5 y CSS3.

<HEAD>: Especifica el prólogo del resto del archivo. Son pocas las tags que van dentro de ella, destacando la del título <TITLE> que será utilizado por los marcadores del

navegador e identificará el contenido de la página. En esta sección se encuentra el estilo y diseño de la página y el título de la misma

<BODY>: Encierra el resto del documento, el contenido. En la página se realizó la construcción de los elementos a ser visualizados mediante código html y el uso de Ajax para el control de temperatura analógico, funciones para tener un login con autenticación, recepción de parámetros del módulo arduino para graficar la temperatura obtenida y una herramienta de control para activar / desactivar manualmente la climatización.

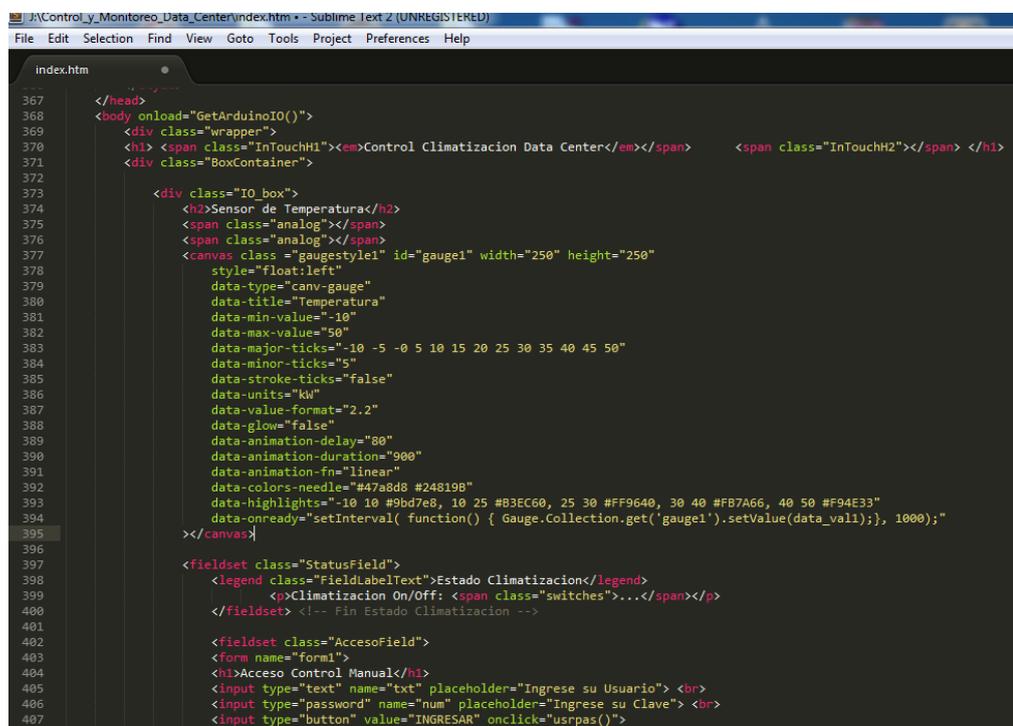
Ejemplo:

```
<HTML>
<HEAD>
<TITLE>Ejemplo 1</TITLE>

</HEAD>
<BODY>
Hola mundo

</BODY>
</HTML>
```

A continuación se muestra la captura parcial del desarrollo de la página web elaborada (Figura 4.43) en el software “Sublime Text”.



```
index.htm
367 </head>
368 <body onload="GetArduinoIO()">
369 <div class="wrapper">
370 <h1> <span class="InTouchH1"><em>Control Climatizacion Data Center</em></span> <span class="InTouchH2"></span> </h1>
371 <div class="BoxContainer">
372
373 <div class="IO_box">
374 <h2>Sensor de Temperatura</h2>
375 <span class="analog"></span>
376 <span class="analog"></span>
377 <canvas class="gaugestyle1" id="gauge1" width="250" height="250"
378 style="float:left"
379 data-type="canv-gauge"
380 data-title="Temperatura"
381 data-min-value="-10"
382 data-max-value="50"
383 data-major-ticks="-10 -5 -0 5 10 15 20 25 30 35 40 45 50"
384 data-minor-ticks="5"
385 data-stroke-ticks="false"
386 data-units="kw"
387 data-value-format="2.2"
388 data-glow="false"
389 data-animation-delay="80"
390 data-animation-duration="900"
391 data-animation-fn="linear"
392 data-colors-needle="#47a8d8 #24819B"
393 data-highlights="-10 10 #9bd7e8, 10 25 #B3EC60, 25 30 #FF9640, 30 40 #F87A66, 40 50 #F94E33"
394 data-onready="setInterval( function() { Gauge.Collection.get('gauge1').setValue(data_val1)}; 1000);"
395 ></canvas>
396
397 <fieldset class="StatusField">
398 <legend class="FieldLabelText">Estado Climatizacion</legend>
399 <p>Climatizacion On/Off: <span class="switches">...</span></p>
400 </fieldset> <!-- Fin Estado Climatizacion -->
401
402 <fieldset class="AccesoField">
403 <form name="form1">
404 <h3>Acceso Control Manual</h3>
405 <input type="text" name="txt" placeholder="Ingrese su Usuario"> <br>
406 <input type="password" name="num" placeholder="Ingrese su Clave"> <br>
407 <input type="button" value="INGRESAR" onclick="usrpas()">
```

Figura 4.23: Diseño del sitio web basado en HTML5, CSS y Ajax

Elaborado por: Alex Francisco Rodríguez

La visualización de la página web por parte del usuario es la siguiente (Figura 4.24) el cual comprende del módulo para control de la temperatura, estado de la climatización.

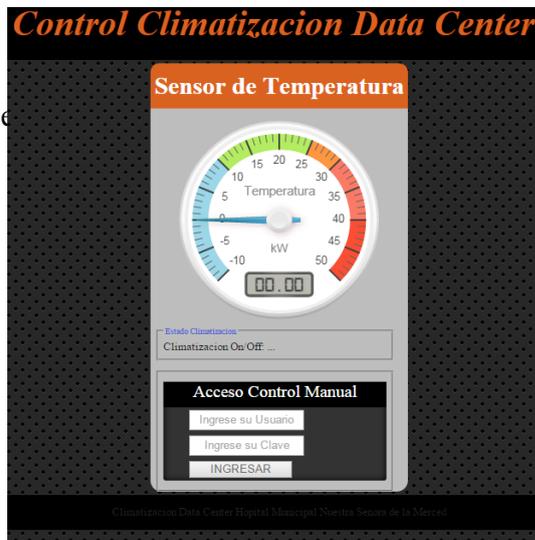


Figura 4.25: Login al sistema de control y monitoreo
Elaborado por: Alex Francisco Rodríguez

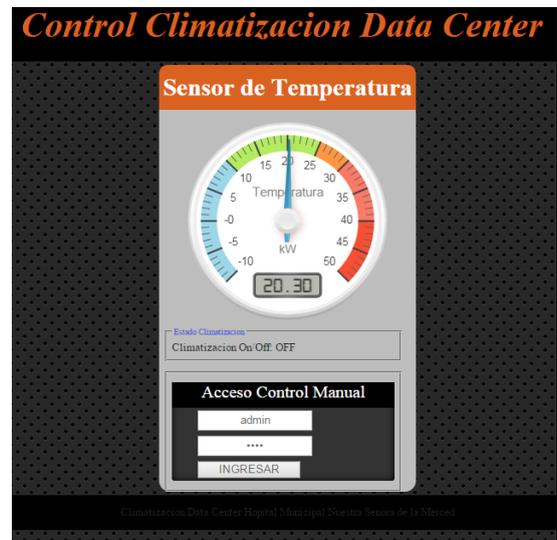


Figura 4.24: Medidor de temperatura operando
Elaborado por: Alex Francisco Rodríguez

En la figura 4.27 es posible determinar que el ingreso es satisfactorio y logra desplegar la herramienta para activar / desactivar manualmente la climatización, caso contrario aparecerá un mensaje de error indicando que las credenciales proporcionadas son erróneas y el control no se activará, como se indica en la figura 4.26.

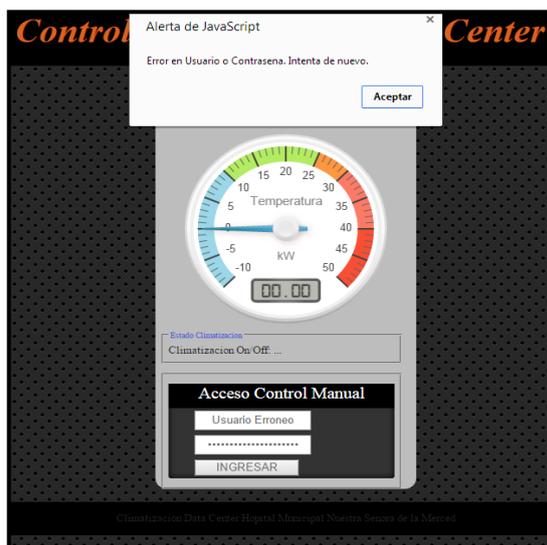


Figura 4.27: Login Erroneo (Mensaje)
Elaborado por: Alex Francisco Rodríguez

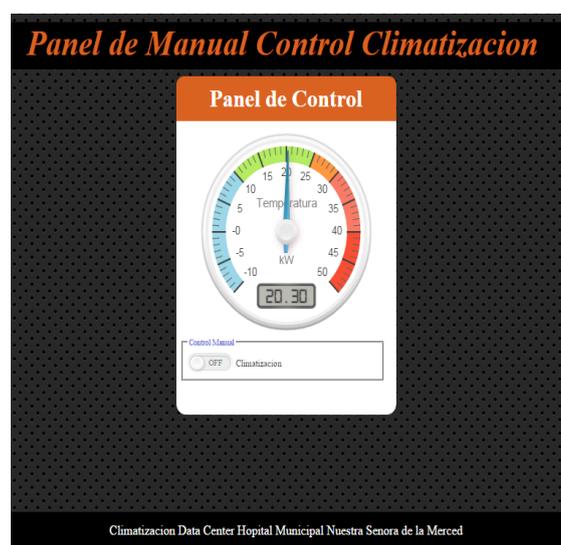


Figura 4.26: Medidor de temperatura funcionando con switch de apagado / encendido
Elaborado por: Alex Francisco Rodríguez

Tweet de alerta para conocer el estado de la temperatura del Data Center

Para el envío del tweet es necesario crear una cuenta en twitter, la cual recibir los mensajes que el módulo arduino envíe a esta cuenta, la cuenta creada será proporcionada al administrador del Data Center para que pueda recibir los respectivos mensajes de alerta sobre el estado de la climatización, como se indica en la figura 4.28.



Figura 4.28: Creación de la cuenta en Twitter

Elaborado por: Alex Francisco Rodríguez

Es necesario proporcionar los permisos respectivos para que el módulo Arduino poste en twitter, para lo cual es necesario registrarse, como se muestra en la figura 4.29 y 4.30



Figura 4.29: Autorización del uso de la cuenta en Twitter

Elaborado por: Alex Francisco Rodríguez

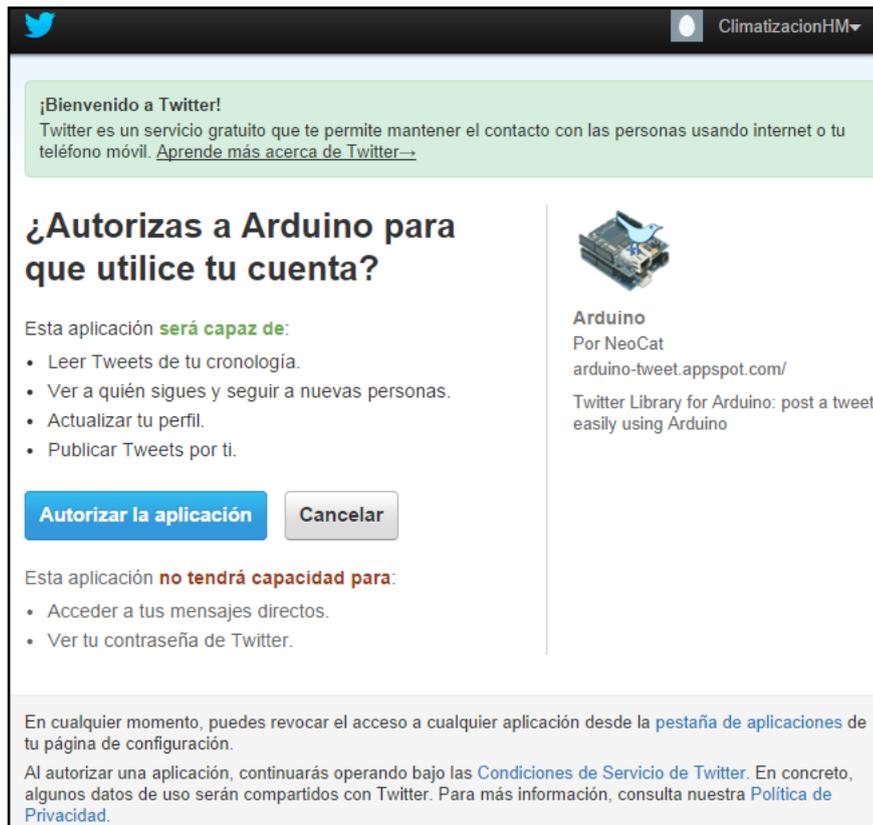


Figura 4.30: Autorización del uso de la cuenta en Twitter
Elaborado por: Alex Francisco Rodríguez

Un vez realizado los pasos de seguridad y de a ver proporcionado los respectivos permisos a la placa Arduino la página automáticamente se redireccionará hacia un código llamado “TOKEN”, el cual será usado en la programación del código fuente del sistema, como se muestra en la figura 4.32.



Figura 4.32: Token de seguridad
Elaborado por: Alex Francisco Rodríguez

También se ha añadido una aplicación de twitter llamada “TWEET DELETE” la cual se encarga de borrar cada cierto tiempo los tweets almacenadas en la cuenta de acuerdo a su configuración, en este caso se ha configurado a la aplicación para que cada mes se borren los tweets posteados en dicha cuenta, como se muestra en la figura 4.31.



Figura 4.31: Servicio de eliminación de Tweets
Elaborado por: Alex Francisco Rodríguez

En el caso de no querer hacer uso de esta aplicación o de configurarla de otra manera, también nos brinda la opción de “Apague TweetDelete (o recoger nuevos ajustes)” la cual se encarga de desactivar la aplicación o de reconfigurarla de acuerdo a la necesidad del usuario.



Figura 4.32: Confirmación del Servicio de eliminación de Tweets
Elaborado por: Alex Francisco Rodríguez

Ensamble del chasis:

A continuación se muestra el ensamble del chasis para el circuito de control y monitoreo de la climatización del data center, la cual se encuentra identificados sus componentes de entrada y salida con sus respectivas etiquetas.



Figura 4.33: Vista Conexión de cableado interno superior

Elaborado por: Alex Francisco Rodríguez



Figura 4.34: Ensamblaje del chasis culminado

Elaborado por: Alex Francisco Rodríguez

4.6 Diseño e implementación del software sobre el sistema operativo Android para conocer el estado operativo y funcional del Data Center a través del protocolo SNMP

Diagrama de red para el software de monitoreo en Android

A continuación se muestra la topología de la red con su respectivo direccionamiento IP y el recorrido para acceder al protocolo SNMP configurado en los servidores de Correo router de distribución planta alta a los cuales el administrador del área de Sistemas y Comunicaciones proporcionó el acceso, como se muestra en la figura 4.35

Cabe destacar que existen dos formas de conexión al protocolo SNMP para el monitoreo del Data Center las cuales son a través de la red interna del hospital y a través de una red externa, la cual a través de un enrutamiento con la IP pública del Hospital y puertos asignados dirige el tráfico UDP que maneja el protocolo SNMP hacia el respectivo dispositivo que se desea monitorear.

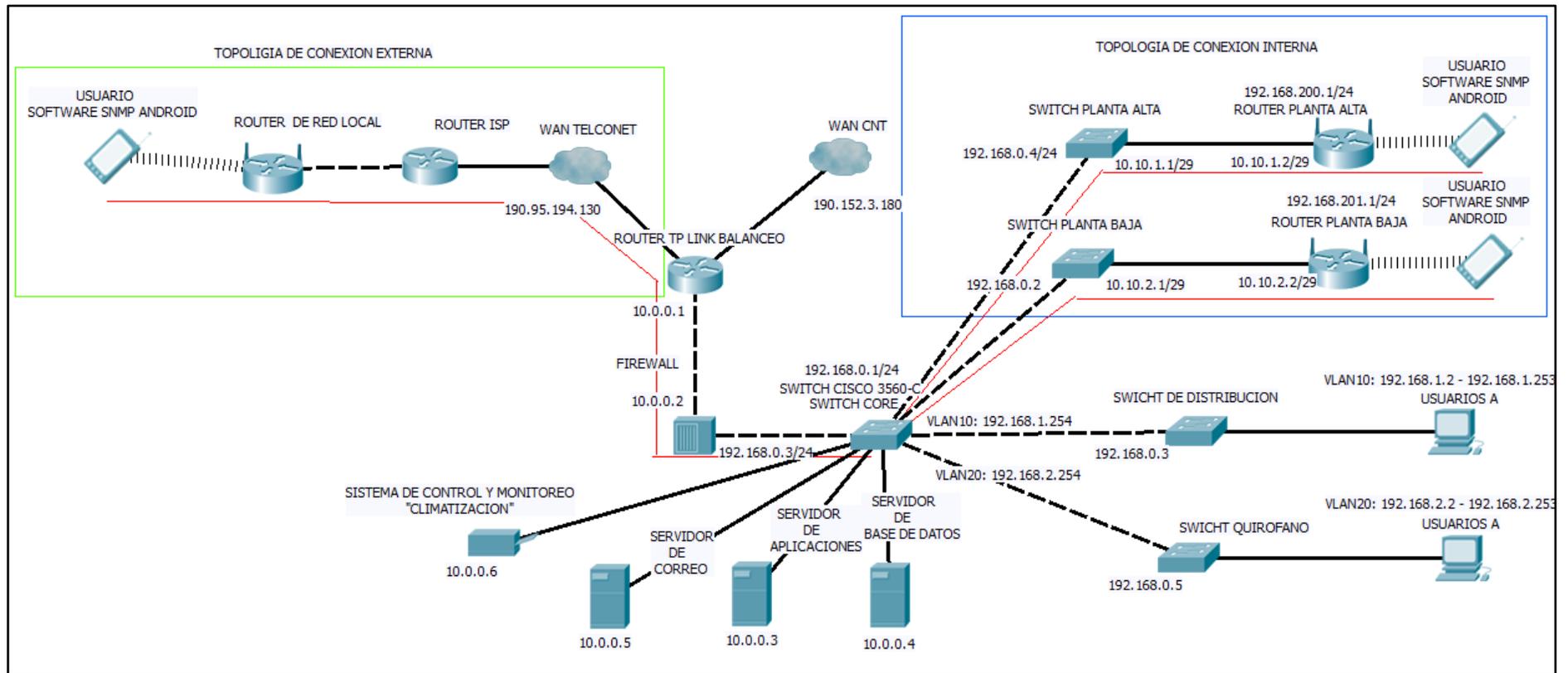


Figura 4.35 Topología de red
Elaborado por: Alex Francisco Rodríguez

Diagrama de funcionamiento del software

El diagrama de funcionamiento o de flujo representado a continuación describe cada una de las etapas de funcionamiento del sistema implementado como se puede observar en la figura 4.36.

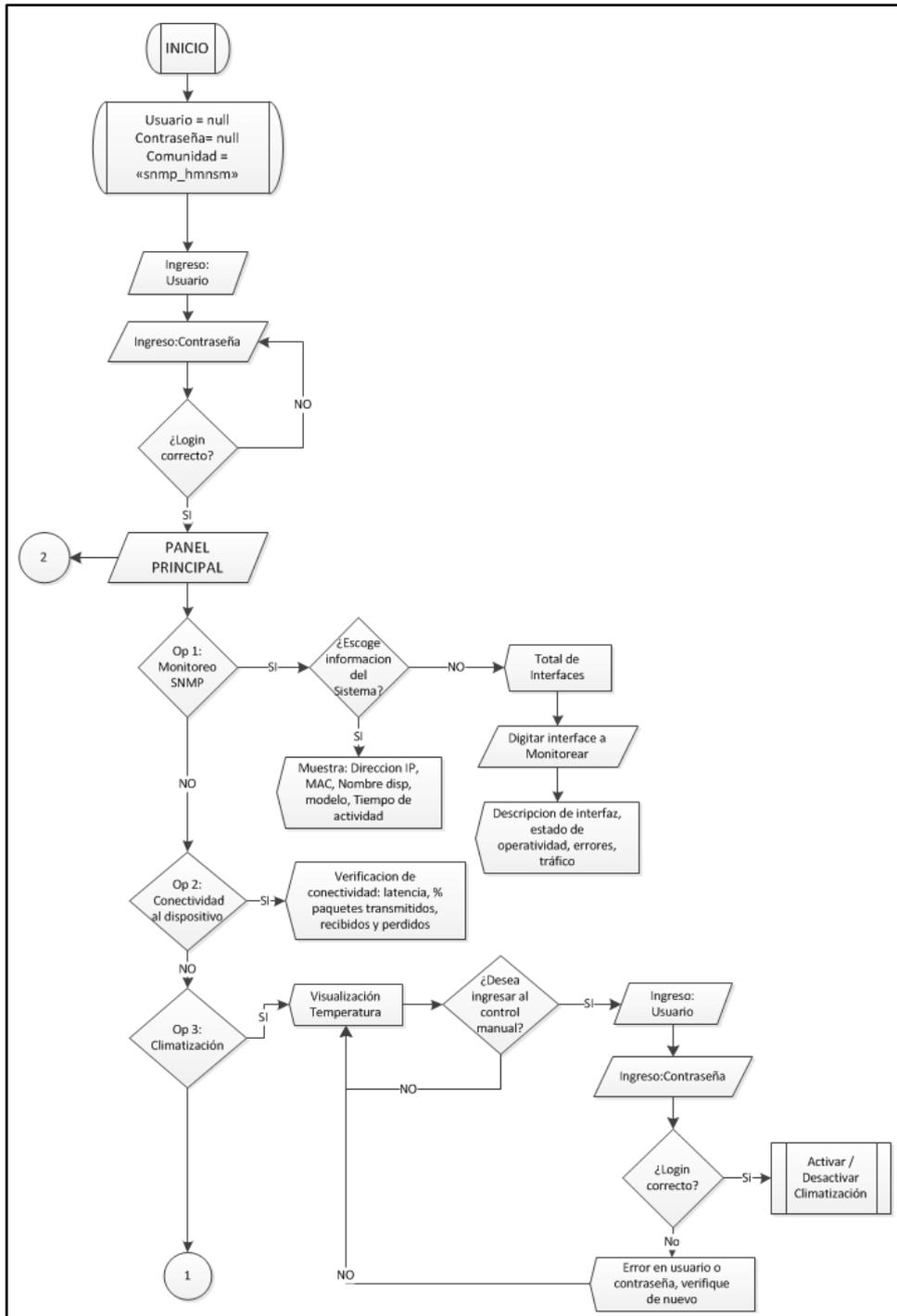


Figura 4.36: Diagrama de flujo software Android (Parte1)
Elaborado por: Alex Francisco Rodríguez

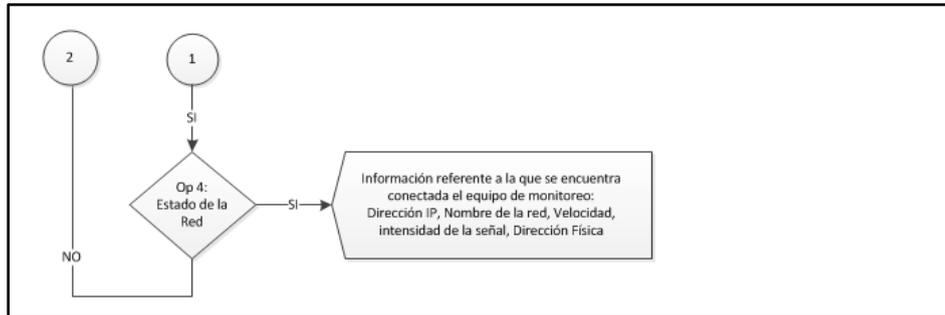


Figura 4.37: Diagrama de flujo software Android (Parte2)

Elaborado por: Alex Francisco Rodríguez

Configuración Protocolo SNMP en Dispositivos de la Red del Hospital

Equipos 1 y 2: Servidor de Correo – Servidor de Base de Datos

PuTTY, es un cliente de acceso remoto a máquinas informáticas de cualquier tipo mediante SSH, Telnet o RLogin, para plataformas Windows y UNIX. En este caso ingresaremos hacia el servidor mediante SSH, el cual es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor que permite a los usuarios conectarse a un host remotamente, en este caso a través del puerto 22.

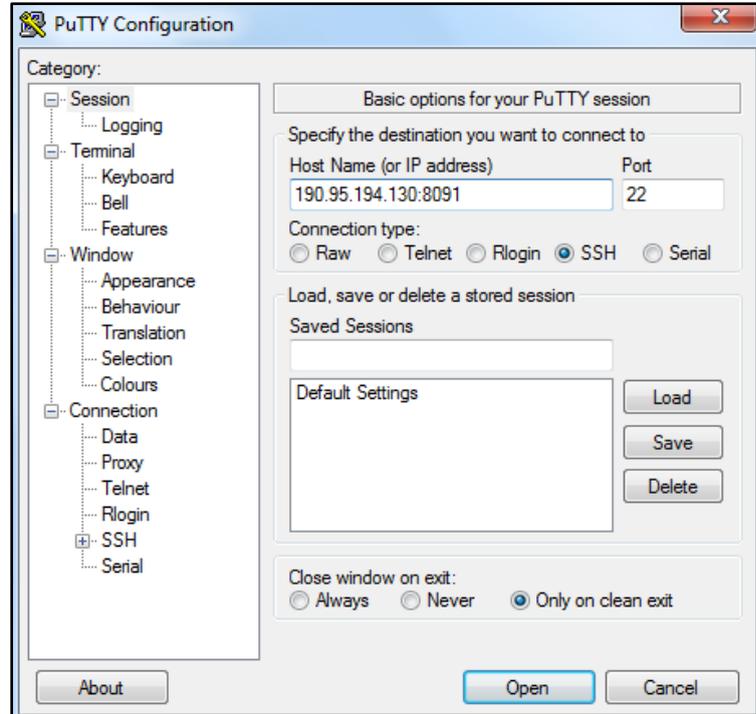


Figura 4.38: Conexión a través de Putty por el protocolo SSH

Elaborado por: Alex Francisco Rodríguez

Ingreso del usuario: root

Ingresamos los datos para realizar la configuraciones pertinentes en este caso ingresamos como root o también llamado superusuario el cual proporciona todos los derechos del sistema.

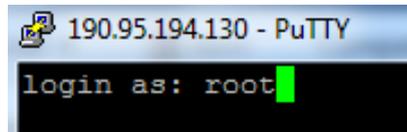


Figura 4.39: Login del usuario
Elaborado por: Alex Francisco Rodríguez

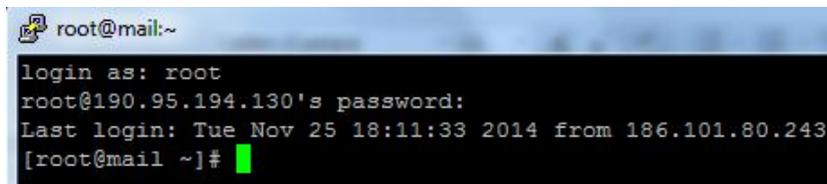


Figura 4.40: Ingreso de contraseña y mensajes de inicio
Elaborado por: Alex Francisco Rodríguez

Como no conocemos la distribución instalada procederemos a buscar que distribución y versión está instalada para según eso configurar o habilitar según sea el caso el servicio SNMP, como es de conocimiento general el demo SNMP viene instalado por defecto en todas las distribuciones, y dependiendo cual sea se encuentra habilitada o no

El comando es: `cat /etc/issue`

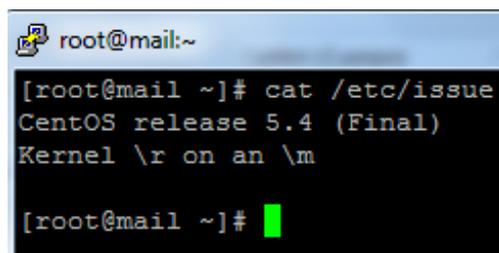


Figura 4.41: verificación de versión instalada
Elaborado por: Alex Francisco Rodríguez

Lo que nos devuelve es: CentOS release 5.4 (Final)

Instalación de SNMP y las utilidades de snmp

```
root@mail~
Loading mirror speeds from cached hostfile
* addons: mirror.esPOCH.edu.ec
* base: mirror.esPOCH.edu.ec
* extras: mirror.esPOCH.edu.ec
* updates: mirror.esPOCH.edu.ec
addons | 1.9 kB 00:00
base | 1.1 kB 00:00
extras | 2.1 kB 00:00
updates | 1.9 kB 00:00
Setting up Install Process
Package 1:net-snmp-5.3.2.2-25.el5_11.x86_64 already installed and latest version
Resolving Dependencies
--> Running transaction check
--> Package net-snmp-utils.x86_64 1:5.3.2.2-25.el5_11 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
net-snmp-utils x86_64 1:5.3.2.2-25.el5_11 updates 193 k
Transaction Summary
-----
Install 1 Package(s)
Update 0 Package(s)
Remove 0 Package(s)

Total download size: 193 k
Is this ok [y/N]: y
Downloading Packages:
net-snmp-utils-5.3.2.2-25.el5_11.x86_64.rpm | 193 kB 00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Installing : net-snmp-utils 1/1

Installed:
net-snmp-utils.x86_64 1:5.3.2.2-25.el5_11

Complete!
[root@mail ~]#
```

Figura 4.42: Instalación del paquete SNMP
Elaborado por: Alex Francisco Rodríguez

SNMP tiene varios métodos de configuración por lo cual veremos los más utilizados y importantes que podremos implementar para nuestro analizador de red. OIDs + snmod

Configuración Public#

Esta configuración es la más sencilla de realizar ya que no se necesita de muchas configuraciones.

```
[root@mail]# vim snmpd.conf
```

```
root@mail~
rocommunity snmp_hmnsm
```

Figura 4.43: Configuración de la comunidad para el servicio SNMP
Elaborado por: Alex Francisco Rodríguez

Con esto solamente tendremos que iniciar el servicio de SNMP.

```
[root@mail]# service snmpd start
```

```
Installed:
  net-snmp-utils.x86_64 1:5.3.2.2-25.el5_11

Complete!
[root@mail ~]# vim snmpd.conf
[root@mail ~]# service snmpd start
Iniciando snmpd:
[root@mail ~]#
```

Figura 4.44: iniciar el servicio SNMPD
Elaborado por: Alex Francisco Rodríguez

Iniciando snmpd: [OK] Comprobaremos que la configuración este bien. Lo probaremos comprobar con el siguiente comando.

Verificamos si los servicios de SNMP se encuentran activos y nos envían respuesta:

```
root@mail:~
[root@mail ~]# snmpwalk -v 1 -c snmp_hmnsn -O e 127.0.0.1 /p
```

Figura 4.45: Verificación servicio SNMP
Elaborado por: Alex Francisco Rodríguez

Verificamos respuesta exitosa:

```
root@mail-
DISMAN-EVENT-MIB::mteEventComment."_snmpd".'linkUp' = STRING:
DISMAN-EVENT-MIB::mteEventComment."_snmpd".'mteTriggerFailure' = STRING:
DISMAN-EVENT-MIB::mteEventComment."_snmpd".'mteTriggerFalling' = STRING:
DISMAN-EVENT-MIB::mteEventComment."_snmpd".'mteTriggerFired' = STRING:
DISMAN-EVENT-MIB::mteEventComment."_snmpd".'mteTriggerRising' = STRING:
DISMAN-EVENT-MIB::mteEventActions."_snmpd".'linkDown' = BITS: 80 0
DISMAN-EVENT-MIB::mteEventActions."_snmpd".'linkUp' = BITS: 80 0
DISMAN-EVENT-MIB::mteEventActions."_snmpd".'mteTriggerFailure' = BITS: 80 0
DISMAN-EVENT-MIB::mteEventActions."_snmpd".'mteTriggerFalling' = BITS: 80 0
DISMAN-EVENT-MIB::mteEventActions."_snmpd".'mteTriggerFired' = BITS: 80 0
DISMAN-EVENT-MIB::mteEventActions."_snmpd".'mteTriggerRising' = BITS: 80 0
DISMAN-EVENT-MIB::mteEventEnabled."_snmpd".'linkDown' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEnabled."_snmpd".'linkUp' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEnabled."_snmpd".'mteTriggerFailure' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEnabled."_snmpd".'mteTriggerFalling' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEnabled."_snmpd".'mteTriggerFired' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEnabled."_snmpd".'mteTriggerRising' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEntryStatus."_snmpd".'linkDown' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEntryStatus."_snmpd".'linkUp' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEntryStatus."_snmpd".'mteTriggerFailure' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEntryStatus."_snmpd".'mteTriggerFalling' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEntryStatus."_snmpd".'mteTriggerFired' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventEntryStatus."_snmpd".'mteTriggerRising' = INTEGER: 1
DISMAN-EVENT-MIB::mteEventNotification."_snmpd".'linkDown' = OID: IF-MIB::linkDown
DISMAN-EVENT-MIB::mteEventNotification."_snmpd".'linkUp' = OID: IF-MIB::linkUp
DISMAN-EVENT-MIB::mteEventNotification."_snmpd".'mteTriggerFailure' = OID: DISMAN-EVENT-MIB::mteTriggerFailure
DISMAN-EVENT-MIB::mteEventNotification."_snmpd".'mteTriggerFalling' = OID: DISMAN-EVENT-MIB::mteTriggerFalling
DISMAN-EVENT-MIB::mteEventNotification."_snmpd".'mteTriggerFired' = OID: DISMAN-EVENT-MIB::mteTriggerFired
DISMAN-EVENT-MIB::mteEventNotification."_snmpd".'mteTriggerRising' = OID: DISMAN-EVENT-MIB::mteTriggerRising
DISMAN-EVENT-MIB::mteEventNotificationObjectsOwner."_snmpd".'linkDown' = STRING: _snmpd
DISMAN-EVENT-MIB::mteEventNotificationObjectsOwner."_snmpd".'linkUp' = STRING: _snmpd
DISMAN-EVENT-MIB::mteEventNotificationObjectsOwner."_snmpd".'mteTriggerFailure' = STRING: _snmpd
DISMAN-EVENT-MIB::mteEventNotificationObjectsOwner."_snmpd".'mteTriggerFalling' = STRING: _snmpd
DISMAN-EVENT-MIB::mteEventNotificationObjectsOwner."_snmpd".'mteTriggerFired' = STRING: _snmpd
DISMAN-EVENT-MIB::mteEventNotificationObjectsOwner."_snmpd".'mteTriggerRising' = STRING: _snmpd
DISMAN-EVENT-MIB::mteEventNotificationObjects."_snmpd".'linkDown' = STRING: linkUpDown
DISMAN-EVENT-MIB::mteEventNotificationObjects."_snmpd".'linkUp' = STRING: linkUpDown
DISMAN-EVENT-MIB::mteEventNotificationObjects."_snmpd".'mteTriggerFailure' = STRING: triggerFail
DISMAN-EVENT-MIB::mteEventNotificationObjects."_snmpd".'mteTriggerFalling' = STRING: triggerFire
DISMAN-EVENT-MIB::mteEventNotificationObjects."_snmpd".'mteTriggerFired' = STRING: triggerFire
DISMAN-EVENT-MIB::mteEventNotificationObjects."_snmpd".'mteTriggerRising' = STRING: triggerFire
DISMAN-EVENT-MIB::nlmConfigGlobalEntryLimit.0 = Gauge32: 1000
DISMAN-EVENT-MIB::nlmConfigGlobalAgeOut.0 = Gauge32: 1440 minutes
DISMAN-EVENT-MIB::nlmStatsGlobalNotificationsLogged.0 = Counter32: 0 notifications
DISMAN-EVENT-MIB::nlmStatsGlobalNotificationsBumped.0 = Counter32: 0 notifications
[root@mail ~]#
```

Figura 4.46: Verificación el servicio SNMPD
Elaborado por: Alex Francisco Rodríguez

Equipos 3 y 4: Router de Balanceo – Switch de Core

En la captura de pantalla de la figura 4.46 se observa la configuración del router de distribución planta alta en la cual se utiliza el software winbox para realizar la configuración del protocolo SNMP, dicho software es utilizado para la realización de configuraciones administrativas. En la barra de herramientas sección IP seleccionamos la opción SNMP en la cual se despliega la ventana SNMP settings en el cual se habilita y se configura el protocolo con la siguiente información: información de contacto, localización, comunidad y versión del protocolo.

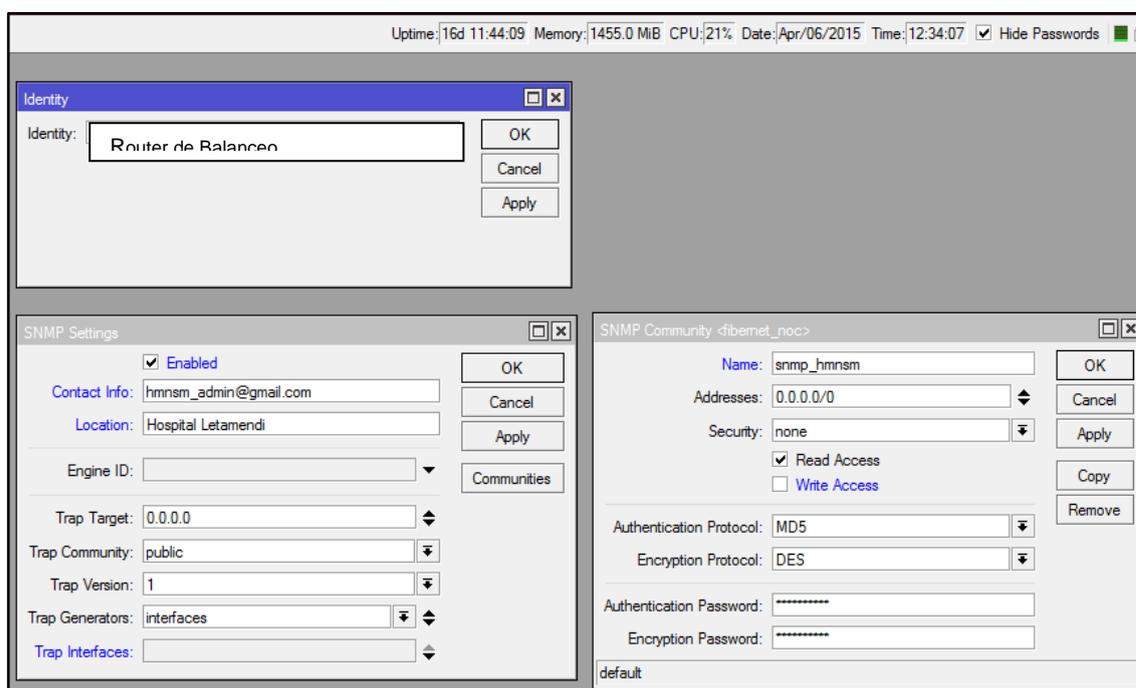


Figura 4.47: Configuración protocolo SNMP
Elaborado por: Alex Francisco Rodríguez

Capturas y Parte del Código del Diseño de la Aplicación

Diseño del Layout (interfaz gráfica), el cual se encuentra enlazado lógicamente a la clase “MainActivity.java”, encargada de procesar los datos de usuario y contraseña ingresados, para el uso del software en el dispositivo móvil.



Figura 4.48: Login App Sistema de Control y Monitoreo

Elaborado por: Alex Francisco Rodríguez

El código empleado en su parte principal se compone de una función de comparación a través de un IF el cual valida la información ingresada por el usuario y en caso de ser satisfactoria dicha evaluación abre el menú siguiente y de ser negativa muestra un mensaje de error. (Ver anexo DI)

Diseño del menú de opciones “Menú de administración” para uso del software, el cual está compuesto por: administrador SNMP, conectividad, estado de la red WIFI, climatización el cual es redirección al sitio web del sistema de control y monitoreo de la climatización Data Center y notificaciones de alerta.



Figura 4.49: Menú de administración

Elaborado por: Alex Francisco Rodríguez

El módulo de “Administración SNMP” consta de dos consultas, de las cuales una se realiza la petición al Sistema (Sistema MIB) y la otra hacia las interfaces físicas (Interfaz MIB) de los dispositivos a monitorear:



Figura 4.50: Gestor SNMP

Elaborado por: Alex Francisco Rodríguez

El módulo de “Administración SNMP” dispone de una sección denominada SNMP INTERFAZ MIB, la librería utilizada para todo lo referente a consultas y transacciones SNMP es SNMP4J-2.2.2.jar correspondiente a la versión 2c del protocolo SNMP; donde se realiza una consulta al equipo monitoreado y devuelve como resultado el número total y nombres de interfaces físicas y virtuales que dispone el dispositivo, en su sección posterior existe una consulta particular hacia una interfaz la cual muestra en su sección Descripción el nombre de la interfaz, tipo, MTU, Dirección MAC, velocidad del puerto, estado de administración, estado operativo y la hora del último cambio realizado, en la segunda sección siguiente se muestran los Errores Producidos a nivel lógico en la interfaz tanto en su lado de transmisión como en recepción, la tercera sección de tráfico proporciona información en relación a la cantidad de tráfico generada en la interfaz de transmisión y recepción representada en porcentajes con relación al tráfico total de dicha interfaz; la sección cuarta de Tráfico de la Interfaz describe la cantidad de tráfico que atravesó la interfaz hasta el momento de la consulta, representada con unidades de almacenamiento de datos en Kbytes y Mbytes según el caso, en la quinta sección se encuentra el módulo de Control de Interfaz MIB el cual posee la opción de Activar / Desactivar la interfaz consultada en el caso de que dicha interfaz presente un bucle, link up / link down persistente o un consumo de tráfico excesivo que pueda colapsar el equipo. (Ver anexo D2)

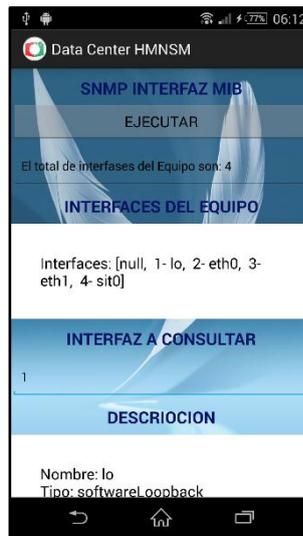


Figura 4.51: Interfaz de Consulta Interfaz MIB
Elaborado por: Alex Francisco Rodríguez

El código empleado realiza la comparación del número total de interfaces que posee el equipo, y en el caso de obtener un resultado diferente de 0 se ejecuta las consultas con el protocolo SNMP en su sección de interfaces, las cuales tienen relación con la descripción de la interfaz, errores producidos, porcentaje de uso y consumo total, las cuales son visualizadas posteriormente en la interfaz gráfica del archivo xml correspondiente, donde se conecta al código a la clase Gestor de Red, de la cual cada resultado obtenido es en base a una consulta a un OID específico de la clase SesiónSNMP. (Ver anexo D3)

El módulo de administración SNMP consta además del módulo de Consulta Sistema MIB, el cual proporciona información general del equipo monitoreado, como es el nombre del dispositivo, modelo, tiempo de actividad y uso de procesador, este último solo se mostrará cuando el equipo disponga de un procesador físico o se encuentren los procesadores activos trabajando en modo paralelo, donde al tener activada esta configuración se muestra el promedio del uso general de todos los procesadores.



Figura 4.52: Consulta Sistema MIB
Elaborado por: Alex Francisco Rodríguez

El código empleado para la consulta al sistema relaciona la información obtenida a través de la consulta SNMP en su sección de Sistema, donde se realiza una comparación con el PDU obtenido (PDU: Contenido de la unidad de datos del protocolo, el que depende de la operación que se ejecute) donde en el caso de ser “null” enviaremos un mensaje indicando que no es posible la conexión, y de ser factible se realizará consulta con los OID’S para recibir el nombre del dispositivo, modelo, tiempo de actividad y uso de procesador. (Ver anexo D4)

El módulo de conectividad, es el encargado de verificar la conectividad de los equipos mediante el comando ping, el cual envía un paquete al host remoto utilizando el protocolo ICMP, los cuales, de ser recibido por dicho host, genera un paquete similar de respuesta para el host local. Cuando el host local los recibe, muestra una línea de información por pantalla.

Al terminar, ping muestra un reporte estadístico, en el que informa la cantidad de paquetes transmitidos, recibidos, y el porcentaje de pérdida de paquetes. También se informa sobre los tiempos mínimo, promedio y máximo que toma el recorrido de ida y vuelta entre ambos hosts, así verificando si el dispositivo se encuentra en actividad o no, como se muestra en la figura 4.49.



Figura 4.53: Conectividad Dispositivos (Ping)
Elaborado por: Alex Francisco Rodríguez

El código fuente realizado para el comando Ping se encuentra configurado para realizarse sobre cuatro dispositivos con direcciones IP distintas, en su parte esencial el código inicialmente detecta la versión del SDK del dispositivo que debe ser menor a 25 debido a que hasta la versión actual la número 24 el código empleado tiene una completa funcionalidad, pero debido a la evolución en versiones superiores el código puede no ejecutarse, de ser positiva la comparación se ingresa a través de la consola del mismo y ejecutar el comando para el Ping el cual se conecta con el kernel de Linux y ejecuta una línea de código en la consola (Runtime) lo cual devuelve una cadena con el resultado, el mismo que será impreso en el layout respectivo.

La aplicación se garantiza en cuanto a su operatividad completa en dispositivos de gama alta y media, que tienen versiones de android / SDK entre la 18 a 24 que es la última lanzada. (Ver anexo D5)

El código además verifica la conexión a la red, y en el caso de no estar activa enviará un mensaje para que el usuario active su wifi o red de datos.

El módulo de "Estado de la red WIFI" nos permite conocer información detallada del estado de la red inalámbrica a la cual nos encontramos conectado, así como la dirección IP, SSID, BSSID, velocidad, RSSI, MAC.



Figura 4.54: Consulta Estado de la Red WIFI
Elaborado por: Alex Francisco Rodríguez

La codificación utilizada para detectar el estado de la red Wifi, se realiza a través de la clase ConnectivityManager, allí se dispone del acceso al controlador del Wifi (WifiManager) y la detección de la información de la conexión, la cual consiste en información de la dirección IP, el SSID de la red, la dirección MAC del router o Access Point, la intensidad de la señal y dirección física del dispositivo. (Ver anexo D6)

Podemos visualizar el layout del módulo de activación de notificaciones de Alerta, el mismo que tiene las opciones para realizar el monitoreo en segundo plano del Servidor de Correo, Router de Distribución de Planta Alta y Router de Distribución de planta baja, sea de forma individual como conjunta; contando con un botón para iniciar el servicio y otro para detener.



Figura 4.55: Activación notificaciones de alerta
Elaborado por: Alex Francisco Rodríguez

El código necesario para ejecutar el monitoreo en segundo plano, necesita como parámetros a una variable de tipo SNMP, una dirección ADDRESS y un MENSAJE DE ERROR, los cuales serán utilizados con un método client.start para realizar una consulta sobre un dispositivo, en este caso se utilizó el OID referente al número total de

interfaces, esta ejecución se realiza con un intervalo de 10 segundos, y de ser negativa la obtención de la información emitirá un mensaje a través de una alerta al dispositivo a través de una alerta de tipo Toast, indicando que no existe conexión con el dispositivo seleccionado para monitoreo, para que el usuario tome las medidas necesarias para visitar físicamente el Data Center y corregir la falla presentada. (Ver anexo D7)

4.7 Realización de pruebas del sistema de control y monitoreo de la Central de Datos diseñado.

En esta sección se ha elaborado pruebas de los dispersivos que se va a controlar y monitorear los cuales son:

Servidor de correo

Los datos que se han obtenido a continuación se han podido corroborar mediante el software de monitoreo denominado “The Dude” propietario del fabricante Mikrotik, donde se observa que en la consulta a Sistema MIB el nombre de dispositivo, modelo y el tiempo en actividad concuerdan con los datos obtenidos por la aplicación, como se muestran en la figura 4.60, esta comprobación se realizó sobre el Servidor de Correo del HMNSM y nos proporcionó resultados similares a los obtenidos con el software desarrollado, demostrando su efectividad y correcta ejecución

Para configurar el dispositivo es necesario establecer la dirección IP, el tipo de agente, el perfil SNMP donde se configura la comunidad y contraseña en caso de tenerla, al igual que la versión del protocolo a utilizar, en este caso es la versión 2c

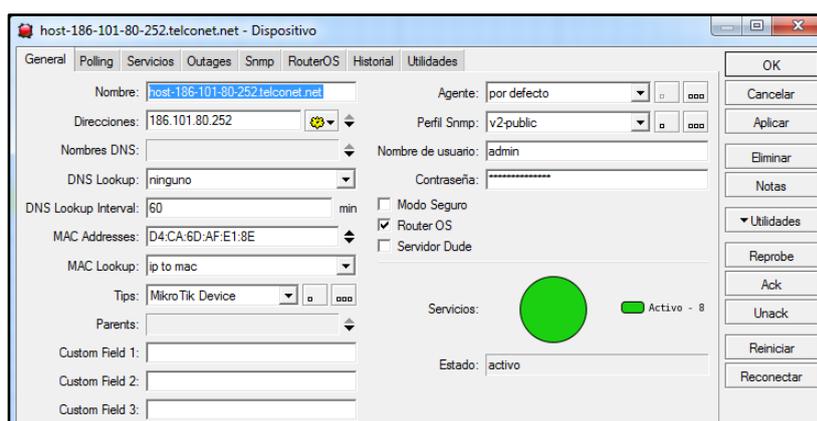


Figura 4.56 Establecer Parametros

Elaborado por: Alex Francisco Rodríguez

Una vez configurado el dispositivo es posible conocer el tráfico real que atraviesa el router en este caso al tener ya levantado el servicio es posible determinar el nombre del host, y la interfaz que tiene conexión con el mundo como se muestra en la figura 4.61.

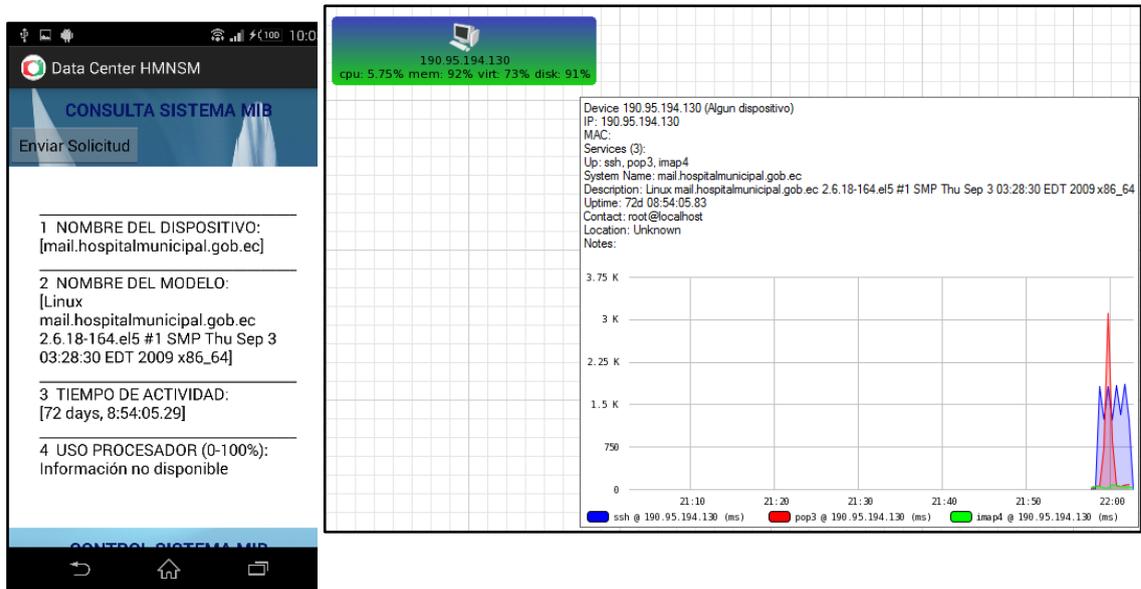


Figura 4.57: Conexión al servidor
Elaborado por: Alex Francisco Rodríguez

En la consulta Interfaz MIB se observa que el número total de interfaces son 4 de las cuales cada interfaz tiene asignado un nombre lo(1), eth0(2), eth1(3) y sit0(4) en la cual la interfaz número 4 que corresponde a sit0 se encuentra desactiva ya que se encuentra resaltada y marcada con una x y la 1,2,3 se encuentran en operatividad, el software The Dude muestra además el tipo de interfaz y MTU de cada interfaz, en contraste del software desarrollado muestra dicha información de forma particular al ejecutar la consulta a la interfaz particular.

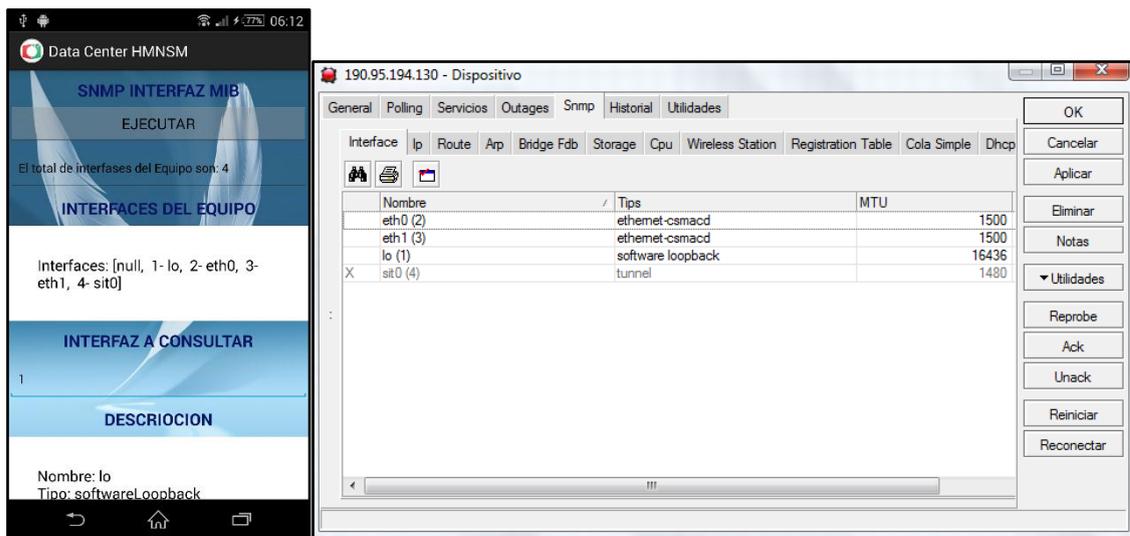


Figura 4.58: Interfaz MIB total de interfaces
Elaborado por: Alex Francisco Rodríguez

A continuación se visualiza la consulta a cada interfaz, la misma que guarda relación de forma directa y exacta con la obtenida en la consulta obtenida por el Software The Dude:

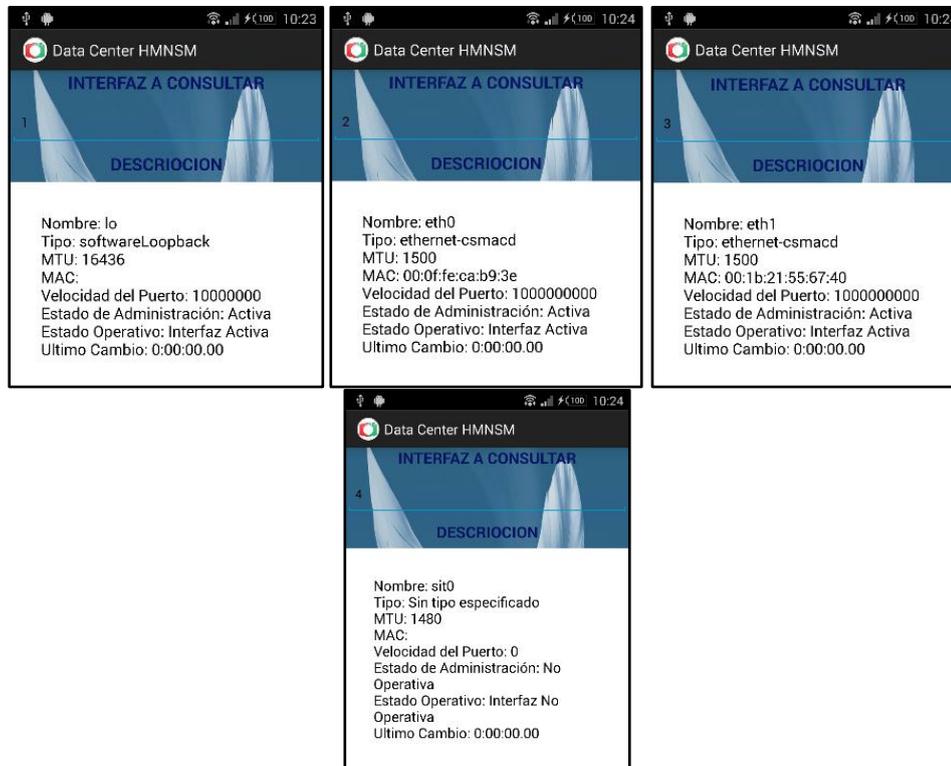
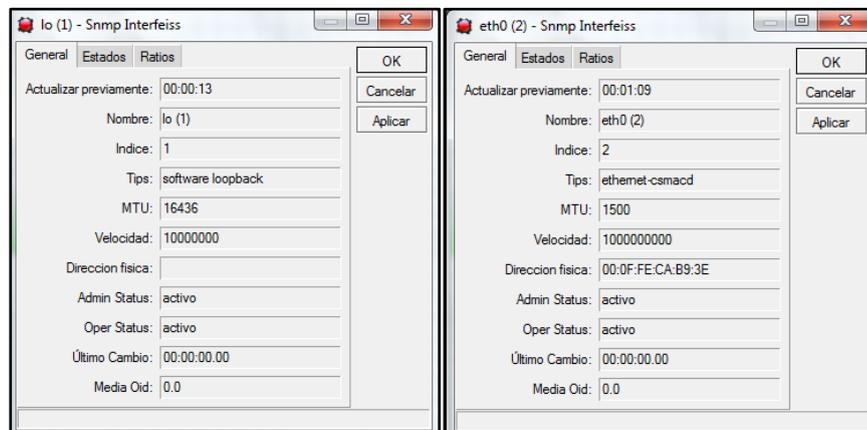


Figura 4.59: Interfaz MIB - Descripción general
Elaborado por: Alex Francisco Rodríguez

Las figuras a continuación muestran los resultados de las consultas en particular a cada interfaz a través del software The Dude, las cuales al relacionarlas una a una con las respuestas obtenidas por el software desarrollado son las mismas, demostrando la eficacia del software motivo de la propuesta de la tesis desarrollada.



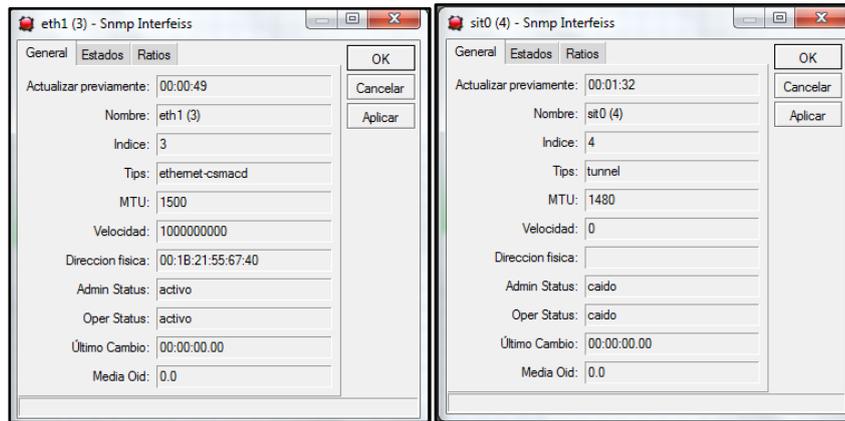


Figura 4.60: Verificación Descripción general Interfaz MIB
Elaborado por: Alex Francisco Rodríguez

Entre la información que nos entrega el protocolo SNMP además de las interfaces disponemos de la información del Sistema, entre las cuales tenemos el nombre del dispositivo, modelo, tiempo de actividad y uso del procesador, lo cual se demuestra a continuación

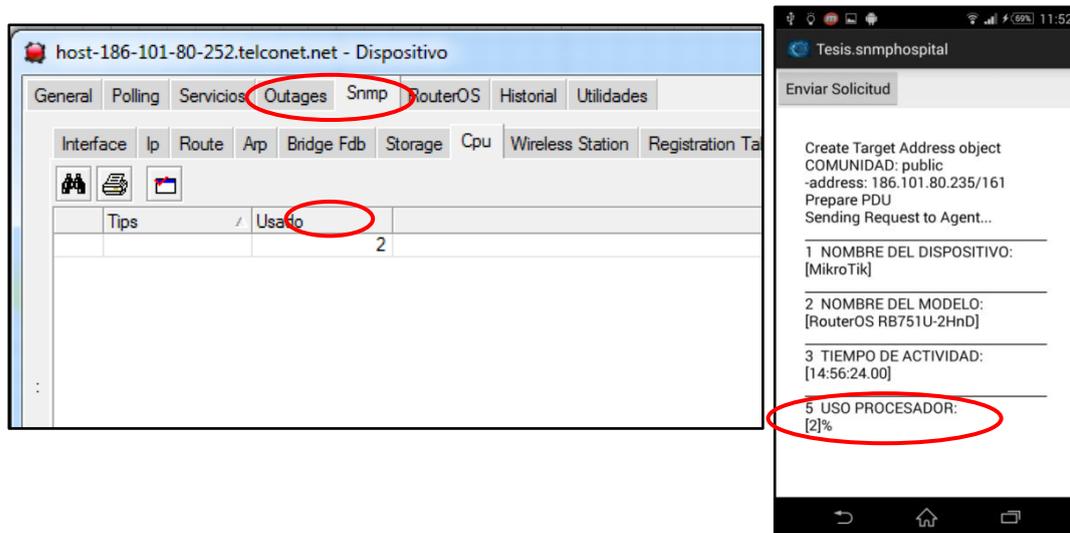


Figura D5: Comprobacion de la Sitema MIB
Elaborado por: Alex Francisco Rodríguez

4.8 Evaluación de la propuesta por el método de experto.

En esta evaluación por el método de experto se ha logrado certificar el sistema que se ha implementado en el “HOSPITAL MUNICIPAL NUESTRA SEÑORA DE LA MERCED”.



Ambato 04-abril-2015

Certificado

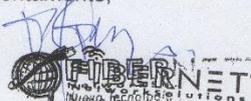
Yo, Luis Fernando Calderón Zambrano, en calidad de Gerente – General, y Administrador de la Red de la Empresa FIBERNET, Certifico lo siguiente:

El Sr. Alex Francisco Rodríguez ha realizado la Implementación de un Sistema de Control y Monitoreo de la Red del Hospital Municipal Nuestra Señora de la Merced, a través de la implementación de un Software para dispositivos móviles basado en la plataforma Android.

Una vez realizada una validación técnica y práctica del código y protocolos empleados, puedo indicar que el sistema cumple con todos los parámetros y requerimientos técnicos en cuanto al monitoreo a través del protocolo SNMP de los dispositivos configurados y proporcionados para el control, reportando las alarmas e información pertinente.

Es todo cuanto puedo Certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento de la manera en que estime conveniente.

Atentamente,



RUC: 1804035820001
Cel: 0958 851342 / 0958 851341

Dirección: Juan Benigno Vela 08-57 entre Castillo yMontalvo, Edificio Villalva Oficinas 2 - 3
Servicio Técnico: 095 885 1341 / 095 885 1342
www.fibernetec.com

5

4.9 PRESUPUESTO

Presupuesto investigación

El proyecto de investigación será financiado por el Investigador.

Tabla 4.5 Presupuesto general.

Ítem	Detalle	Unidad	Cantidad	Valor Unitario	Valor total
1	Internet	horas	500	\$1.00	\$ 500.00
2	Resma de papel A4	c/u	2	\$3.75	\$7.50
3	Perfiles	c/u	5	\$0.60	\$ 3.00
4	Carpetas	c/u	5	\$0.30	\$ 1.50
5	Impresiones	c/u	1000	\$0.10	\$ 100.00
6	Copias	c/u	500	\$0.02	\$ 10.00
7	Esferos	c/u	5	\$0.75	\$3.75
8	Lápices	c/u	2	\$5.00	\$10.00
9	Borrador	c/u	2	\$ 0.50	\$ 1.00
10	CD's	c/u	2	\$0.50	\$1.00
11	Transporte Urbano	c/u	200	\$0.25	\$ 50.00
12	Memory Flash	c/u	1	\$20	\$20
13	Libros	c/u	2	\$50.00	\$100.00
SUBTOTAL					\$ 807.75
IMPREVISTOS (5%)					\$ 40.39
TOTAL					\$ 848.14

Presupuesto para diseño e implementación del hardware para control y monitoreo de la climatización

Tabla 4.6 Presupuesto hardware para control y monitoreo de la climatización.

Ítem	Detalle	Unidad	Cantidad	Valor Unitario	Valor total
1	Multímetro	c/u	1	\$20.00	\$ 20.00
2	Taladro	c/u	1	\$50.00	\$ 50.00
3	Microcontrolador Arduino Mega 2560 R3	c/u	1	\$80.00	\$ 80.00
4	Módulo Ethernet Shield	c/u	1	\$55.00	\$ 55.00
5	Pantalla LCD	c/u	1	\$7.00	\$ 7.00
6	Chasis	c/u	1	\$30.00	\$ 30.00
7	Pistola de silocon	c/u	1	\$5.00	\$ 5.00
8	Cautín	c/u	1	\$8.00	\$ 8.00
9	Sensor de temperatura	c/u	1	\$3.00	\$ 3.00
10	Potenciómetro	c/u	1	\$0.30	\$ 0.30
11	Broca, tornillos y tuercas	c/u		\$3.00	\$ 3.00
12	Interruptor y conectores	c/u		\$2.00	\$ 2.00
13	Cable USB A-B	c/u	1	\$3.50	\$ 3.50
14	Cargador AC/DC plug macho	c/u	1	\$9.00	\$ 9.00
15	Cable de poder en ocho	c/u	1	\$1.50	\$ 1.50
16	Cable de Red ETHERNET	c/u	1	\$3.50	\$ 3.50
17	Jack bananas hembra / macho, fusible 1A, porta fusible, diodos led, luz piloto	c/u		\$ 3.00	\$ 3.00
SUBTOTAL					\$ 233.80
IMPREVISTOS (5%)					\$ 11.69
TOTAL					\$ 245.49

CAPÍTULO 5

5.1 Conclusiones

En base a la investigación realizada fue posible determinar las siguientes conclusiones:

- La falta de un sistema de control y monitoreo activo en la red del Data Center del Hospital, presentaba fallas y caídas en algunos de los dispositivos no era posible determinar al instante el hecho ocurrido, lo cual generaba problemas relacionados con la no prestación de los servicios informáticos, internet, etc., a los usuarios y clientes internos del hospital.
- El protocolo SNMP fue el escogido para suplir las deficiencias de control y monitoreo existentes en el Data Center, debido a que es un protocolo libre y es posible configurarlo en cualquier dispositivo de red o servidor administrable, sin necesidad de manejar una sola marca de equipos.
- Es importante indicar que para el uso del protocolo fue necesario enrutar las conexiones entrantes a través de la Wan de la red y dirigir las a través de puertos hacia el equipo a monitorearse, de esta manera el software puede ser utilizado desde cualquier lugar del mundo y obtener información de los dispositivos del Data Center de forma inmediata.
- En cuanto al circuito para control y monitoreo de la climatización interna se utilizó un microcontrolador Arduino Mega y periféricos relacionados de la misma marca ya que existe la documentación e información necesaria para el uso y programación de los mismos, con el adicional de ser un hardware muy resistente y de altas prestaciones, permitiendo su funcionamiento de manera continua.

5.2 Recomendaciones:

- Para mejorar la conexión a internet a futuro es necesario adquirir un balanceador de carga más adecuado, el cual realice el balanceo en función de calidad de conexión y tráfico disponible, con lo cual se garantizará una navegación y enlace más estable
- Para darle un mayor campo de acción o potencialidad al software y al protocolo SNMP se puede agregar la funcionalidad de agente proxy y utilizar la red celular para realizar las peticiones, de manera que si se pierde la conectividad se pueda retransmitir la petición a otros dispositivos de la red mediante el dispositivo móvil. En etapas posteriores al ciclo de vida del software se debería implementar un nuevo módulo donde se incluyan servicios adicionales para el control de la red, como por ejemplo: traceroute, verificación de puertos abiertos en la red y un análisis del tráfico generado por la red, donde el último punto a tratarse llevaría implícita la complejidad de rootear el dispositivo móvil para que se pueda tener acceso a estos servicios.
- Debido a que el router encargado del balanceo no puede hacer redirecciones entre puertos será necesario adquirir un router que disponga de estas características, tales como un Router Mikrotik RB850G que podría balancear hasta 4 Wan y hacer redirecciones de puertos sin límites, o un Router Cisco Meraki, con lo cual no existiría una limitante en cuanto a los equipos a redireccionar, ya que actualmente solo fue posible realizar el monitoreo sobre cuatro equipos con tres WAN distintas y ese fue un limitante en cuanto al número de equipos a monitorear, debido a que el total de servidores y routers asciende a once dispositivos, por lo cual se decidió aplicar el control y monitoreo sobre los equipos más críticos de la red
- En cuanto al hardware de control y monitoreo de la climatización en un futuro las dimensiones físicas del mismo debería ajustarse a estándares internacionales para poder colocarlo en un rack, de esta manera será posible tener una mejor ubicación física del mismo al interior del Data Center
- Se recomienda proporcionar el requerimiento eléctrico que cada componente necesita en base al datasheet proporcionado por el fabricante, para evitar alimentación a través del puerto USB.

Referencias

- [1] Luis Andreula, “Red de monitoreo y supervisión de las variables eléctricas en sub-estaciones” in Tech. Paper, 2010, pp 3-8.
- [2] YuLung Li, “estudio y evaluación de aplicaciones para el análisis forense de dispositivos móviles bajo android en la ciudad de Ambato” Tesis de ingeniería en sistemas computacionales e informáticos. Ingeniería., UTA., Ambato, Ecuador, 2013.
- [3] Carlos Vinicio Ailaca Ramírez, “Sistema de monitoreo y control de redes inalámbricas para optimización del servicio de internet en la empresa INTERCOMPU” Tesis de ingeniería en electrónica y comunicaciones. Ingeniería., UTA., Ambato, Ecuador, 2011.
- [4] Thomas Timmermann, “Criterio para la selección adecuada de una solución de monitoreo de red” in Tech. Paper, 2013, pp 20-29.
- [5] Manuel Andrés Vargas Evans, "Sistema de monitoreo y control remoto para una central micro-hidráulica", Tesis de Ingeniería Civil Electricista. Ingeniería, Universidad de Chile, Chile, 2008.
- [6] Carlos A. Vicente Altamirano, “Monitoreo de Recursos de Red” in Tech. Paper, Ciudad de México, UNAM – DGSCA, 2005, pp 1-10.
- [7] International Telecommunications Union. (2013). Tecnología – Los recursos de la red y su monitoreo (Primera Edición.) [En línea], Disponible en: <http://clubensayos.com/Tecnolog%C3%ADa/Los-Recursos-De-Red-Y/865411.html>.
- [8] Eugenio, R., (1999) “Telecomunicaciones móviles” in Tech. Paper, Segunda edición, USA, marcombo.
- [9] Asante Networks, Inc. (2005). “Technologic – simple network management protocol”. [En línea], Disponible en: <http://asante.com/downloads/productdocuments/snmp.pdf>
- [10] Parsons, J., (1989) “Mobile communications system” in Tech. Paper, Primera edición, USA, Blackie.
- [11] Sánchez, P., (2007) “Análisis del desempeño de las redes celulares gsmgprs”, Tesis de licenciatura, México, Universidad Nacional Autónoma de México, septiembre 2005.
- [12] Roberto, M., (2012) “Desarrollo de aplicaciones para Android” in Tech. Paper, Primera edición, Madrid, Ra Ma.

- [13] Fundamentos de administración de red. (2002). Funcionalidad de los sistemas de gestión. [En línea], Disponible en: <http://www2.rhernando.net/modules/tutorials/doc/redes/Gredes.html>.
- [14] Gestión de Redes. (2009). Utilidades de la gestión de Redes. [En línea], Disponible en: <http://www.slideshare.net/ing.adolfo/gestion-de-redes>
- [15] M. DOUGLAS, K. SCHMIDT, Essential SNMP, Segunda edición, USA, O'Reilly, 2005.
- [16] R. MEIER, Professional Android Application Development, USA, Wiley Publishing, 2009.
- [17] E. BURNETTE, Hello, Android. Introducing Google's Mobile Development Platform, USA, Julio 2009
- [18] Android Developer. (2014). Eclipse IDE. [En línea], Disponible en: <http://developer.android.com/tools/help/adt.html>
- [19] Programación para equipo móvil. (2010). Plataformas orientadas al desarrollo móvil. [En línea], http://ppem504informatica.blogspot.com/2013_08_01_archive.html
- [20] Microcontroladores. (2010). Guía de aprendizaje para microcontroladores. [En línea], http://www.academia.edu/5931550/17._GUIA_DE_APRENDIZAJE_EJERCICIOS_PARA_MICROCONTROLADOR_INTERMEDIO
- [21] Arduino. (2014). Arduino Mega 2560. [En línea], <http://www.arduino.cc/en/Main/arduinoBoardMega2560>
- [22] Arduino. (2014). Arduino Ethernet. [En línea], <http://www.arduino.cc/en/Main/ArduinoBoardEthernet>
- [23] Edison Javier Melena Heredia., (2007) “Construcción de un prototipo de control de un juego de luces y la ventilación de una discoteca”, Tesis de Ingeniería en electrónica, Quito Ecuador, Escuela politécnica nacional, septiembre 2007.
- [24] Jorge Hernández-Constante., (2014) “SISTEMA DE CONTROL DE TEMPERATURA MEDIANTE UN SISTEMA TELEMÉTRICO” in Tech. Paper, XXIX edición, Puerto valladares, Jalasco, Mexico.

Anexos

Anexo 1

Estructura de la entrevista

Saludo.

Cuestionario:

1. ¿El Data Center del Hospital cumple con todos los requerimientos técnicos para su funcionamiento y operatividad?

2. ¿Cuenta actualmente con un sistema de control y monitoreo de los dispositivos del Data Center?

3. ¿Puede verificar en tiempo real del estado de la climatización al interior del Data Center?

4. ¿Necesita tener un conocimiento general del estado de su red, sin tener la necesidad de encontrarse presente en el Hospital?

5. ¿Considera necesario disponer de una herramienta para el control de sus equipos Core y Climatización del Data Center basada en una aplicación para dispositivos móviles?

6. ¿Dispone de un Smartphone con tecnología Android o iOS para utilizar la aplicación antes mencionada?

Prueba N° 1

Pruebas con Arduino

Lectura y Toma de Datos Sensor de Temperatura

Para esta prueba se empleó el sensor de temperatura LM35, la placa Arduino Mega 2560, se inició polarizando el sensor de acuerdo a diagrama de conexión ya mencionado anteriormente en la figura 4.39, para lo cual se realizó una función de acuerdo al data sheet técnico del sensor y de esta manera podemos obtener la temperatura en °C y como se indica en la imagen se la imprimió a través de la consola con una comunicación serial entre Arduino y el Computador con un retardo de 1 segundo.

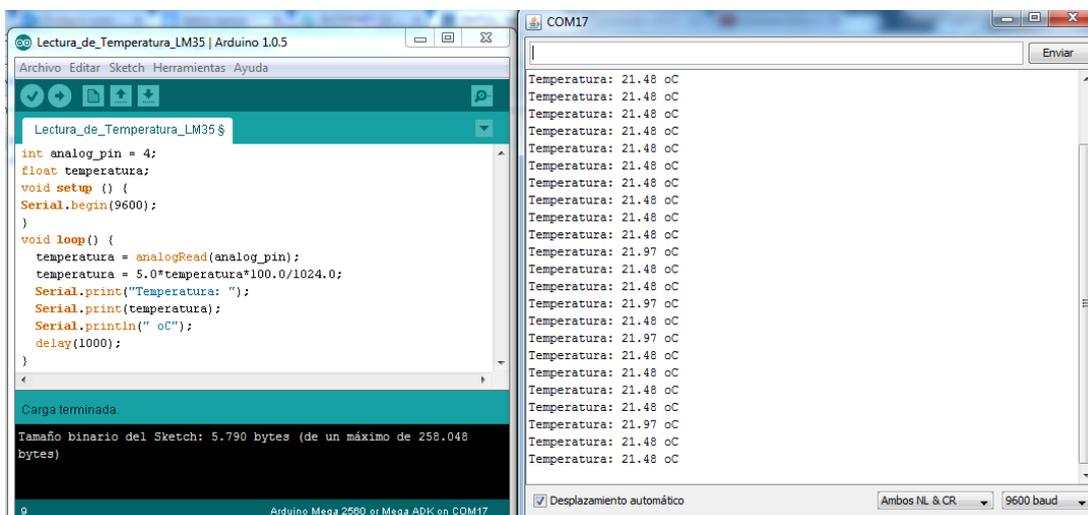


Figura 1: Lectura Y Toma De Datos Sensor De Temperatura

Elaborado por: Alex Francisco Rodríguez

Diagrama de Conexión Arduino y Sensor de Temperatura

El diagrama de conexión representa la placa Arduino Mega, con el sensor LM35, el cual se encuentra conectado a través de un protoboard; la conexión inicia desde la placa con el puerto de polarización de 5v hacia el pin +Vs del sensor, la segunda conexión es desde el pin Vout del sensor hacia el puerto analógico 4 de la placa, y la tercera es desde el pin GND hacia el puerto de polarización GND de la placa, lo cual nos permite interactuar con el sensor.

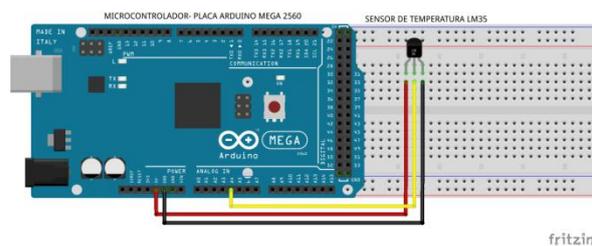


Figura 2: Diagrama De Conexión Arduino Y Sensor De Temperatura

Elaborado por: Alex Francisco Rodríguez

Prueba B

Código de Prueba Pantalla lcd, Sensor de Temperatura y Arduino.

Para la conexión entre el módulo arduino y la pantalla LCD fue necesario emplear la librería LiquidCrystal en la cual nos permite comunicarnos con la pantalla, por motivos de prueba se imprimió una cadena de texto para que se muestre en el LCD

```
Prueba_Pantalla_LCD_Ejemplo_$.  
//Importamos la Biblioteca para poder usar el Display LCD  
#include <LiquidCrystal.h>  
//Inicializamos el display con los pines de interfaz  
LiquidCrystal lcd(16, 17, 18, 19, 20, 21);  
double temp; //Declaramos la variable de temperatura  
/*Creamos un arreglo de 8 bytes  
donde haremos el caracter '°' que necesitamos  
*/  
byte charGrado[8] = {  
  0b01110,  
  0b01010,  
  0b01110,  
  0b00000,  
  0b00000,  
  0b00000,  
  0b00000,  
  0b00000,  
  0b00000  
};  
void setup() {  
  Serial.begin(9600);  
  //Hacemos la inicializacion del display  
  // En la configuracion le decimos que cuenta con 16 columnas y 2 filas  
  lcd.begin(16, 2);  
  
  //Imprimimos un mensaje  
  lcd.setCursor(0,0);  
  lcd.print("Prueba");  
}
```

Figura A3: Código de prueba pantalla LCD,Sensor de Temperatura y Arduino
Elaborado por: Alex Francisco Rodríguez

Diagrama De Conexión Arduino y LCD

Para la conexión física se utilizaron los pines 16, 17, 18, 19, 20, 21, conjuntamente con los pines de polarización 5v y GND para el sensor de temperatura, donde se realiza la lectura del sensor y posteriormente esa información se imprime en la pantalla LCD, además a través del pin de señal del potenciómetro y el pin 3 Vo del LCD se controla el contraste de la pantalla, es decir, la intensidad de los colores de la misma.

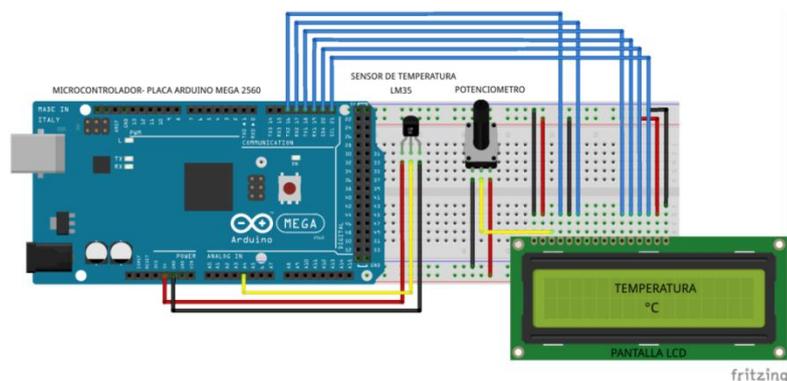


Figura A4: Código de prueba pantalla LCD,Sensor de Temperatura y Arduino
Elaborado por: Alex Francisco Rodríguez

Anexo C

Prueba con el Sketch de Ejemplo Simple Post

Tweet De Arduino

Para publicar un tweet desde arduino es necesario utilizar las librerías SPI (Serial Peripheral Interface) que es básicamente un bus de comunicación a nivel de circuitos integrados. La transmisión de datos se realiza en serie, es decir un bit después de otro, la librería Ethernet encargada de gestionar la comunicación IP desde el arduino hacia el mundo (internet) y por último la librería Twitter para enviar un tweet (mensaje) a través de la placa arduino Ethernet, donde es necesario generar un token de seguridad para la comunicación entre la red social Twitter y Arduino.

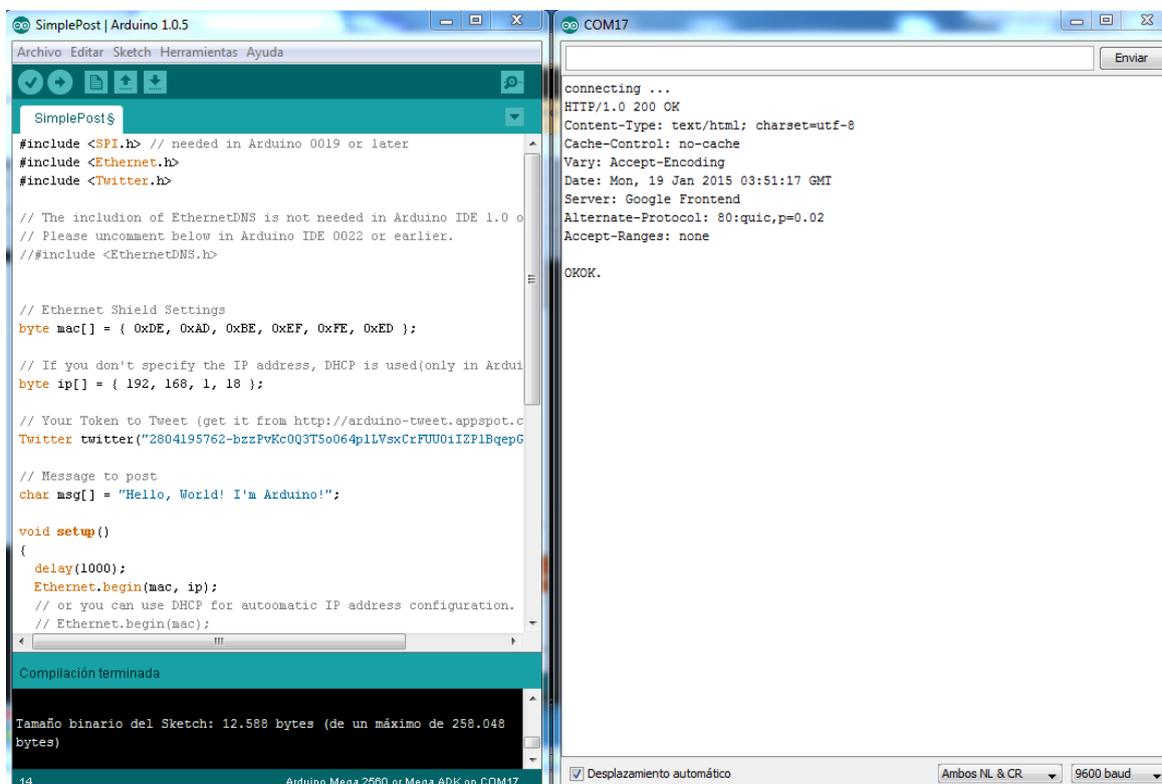
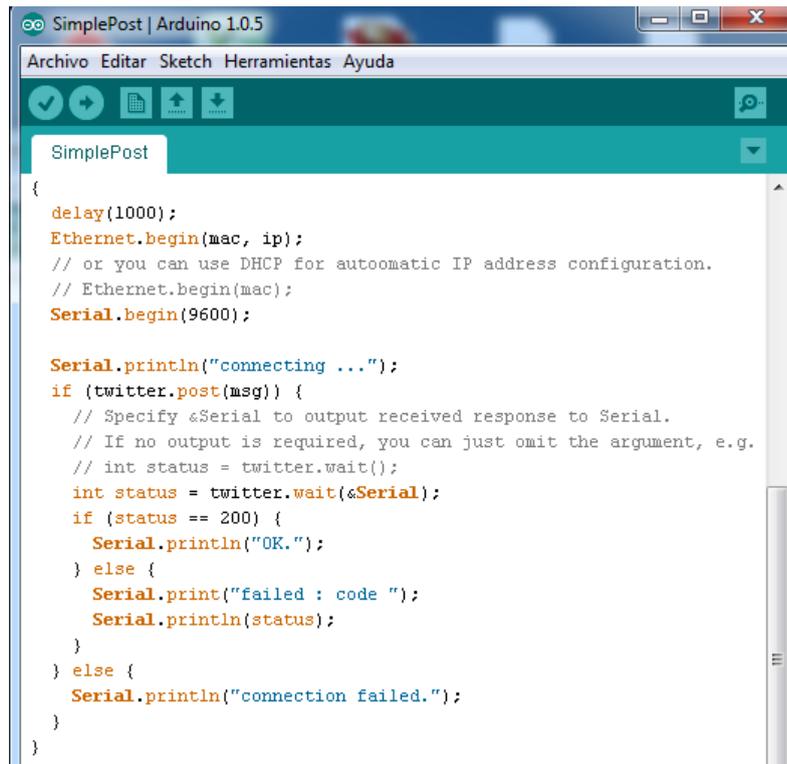


Figura A5: Prueba Con El Sketch De Ejemplo Simple Post Tweet De Arduino (Parte 1)

Elaborado por: Alex Francisco Rodríguez

Para enviar el tweet es necesario elaborar un mensaje, posteriormente inicializar el módulo Ethernet que debe tener configurada una dirección IP que tenga acceso a internet y su dirección MAC definida, posteriormente iniciamos la comunicación serial para verificar que el mensaje sea enviado correctamente, a través del método post de la librería twitter como se muestra en la siguiente figura A6:



```
SimplePost | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
SimplePost
{
  delay(1000);
  Ethernet.begin(mac, ip);
  // or you can use DHCP for automatic IP address configuration.
  // Ethernet.begin(mac);
  Serial.begin(9600);

  Serial.println("connecting ...");
  if (twitter.post(msg) {
    // Specify &Serial to output received response to Serial.
    // If no output is required, you can just omit the argument, e.g.
    // int status = twitter.wait();
    int status = twitter.wait(&Serial);
    if (status == 200) {
      Serial.println("OK.");
    } else {
      Serial.print("failed : code ");
      Serial.println(status);
    }
  } else {
    Serial.println("connection failed.");
  }
}
```

Figura A6: Prueba Con El Sketch De Ejemplo Simple Post
Tweet De Arduino (Parte 2)
Elaborado por: Alex Francisco Rodríguez

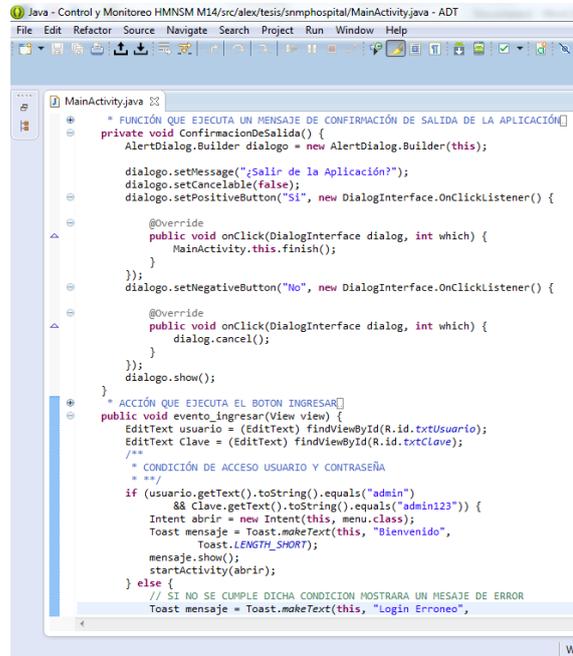
Si la comunicación es satisfactoria podremos recibir el tweet postado por arduino en la cuenta que previamente ha sido creada (Figura A7).



Figura A7: Verificación de Tweet Recibido
Elaborado por: Alex Francisco Rodríguez

Anexo D

Programación



```
Java - Control y Monitoreo HMNSM M14/src/alex/tesis/snmphospital/MainActivity.java - ADT
File Edit Refactor Source Navigate Search Project Run Window Help

MainActivity.java
* FUNCIÓN QUE EJECUTA UN MENSAJE DE CONFIRMACIÓN DE SALIDA DE LA APLICACIÓN
private void ConfirmacionDeSalida() {
    AlertDialog.Builder dialogo = new AlertDialog.Builder(this);

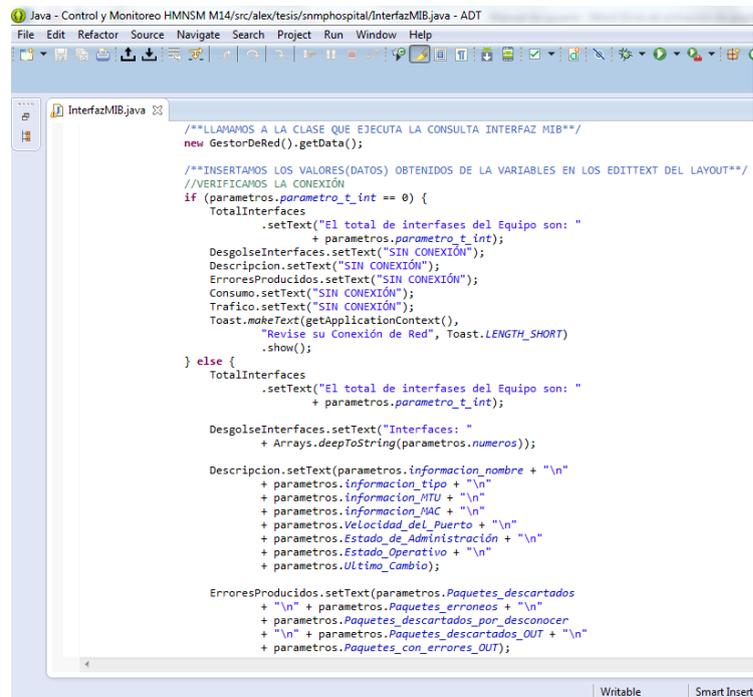
    dialogo.setMessage("¿Salir de la Aplicación?");
    dialogo.setCancelable(false);
    dialogo.setPositiveButton("Si", new DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {
            MainActivity.this.finish();
        }
    });
    dialogo.setNegativeButton("No", new DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialogo.cancel();
        }
    });
    dialogo.show();
}
* ACCIÓN QUE EJECUTA EL BOTON INGRESAR
public void evento_ingresar(View vlew) {
    EditText usuario = (EditText) findViewById(R.id.txtUsuario);
    EditText Clave = (EditText) findViewById(R.id.txtClave);
    /**
     * CONDICIÓN DE ACCESO USUARIO Y CONTRASEÑA
     * **/
    if (usuario.getText().toString().equals("admin")
        && Clave.getText().toString().equals("admin123")) {
        Intent abrir = new Intent(this, menu.class);
        Toast mensaje = Toast.makeText(this, "Bienvenido",
            Toast.LENGTH_SHORT);
        mensaje.show();
        startActivity(abrir);
    } else {
        // SI NO SE CUMPLE DICHA CONDICION MOSTRARA UN MENSAJE DE ERROR
        Toast mensaje = Toast.makeText(this, "Login Erroneo",

```

Figura D1: Clase “MainActivity.java”
Elaborado por: Alex Francisco Rodríguez



```
Java - Control y Monitoreo HMNSM M14/src/alex/tesis/snmphospital/InterfazMIB.java - ADT
File Edit Refactor Source Navigate Search Project Run Window Help

InterfazMIB.java
/**LLAMAMOS A LA CLASE QUE EJECUTA LA CONSULTA INTERFAZ MIB**/
new GestorDeRed().getData();

/**INSERTAMOS LOS VALORES(DATOS) OBTENIDOS DE LA VARIABLES EN LOS EDITTEXT DEL LAYOUT**/
//VERIFICAMOS LA CONEXIÓN
if (parametros.parametro_t_int == 0) {
    TotalInterfaces
        .setText("El total de interfases del Equipo son: "
            + parametros.parametro_t_int);
    DesgolseInterfaces.setText("SIN CONEXIÓN");
    Descripcion.setText("SIN CONEXIÓN");
    ErroresProducidos.setText("SIN CONEXIÓN");
    Consumo.setText("SIN CONEXIÓN");
    Trafico.setText("SIN CONEXIÓN");
    Toast.makeText(getApplicationContext(),
        "Revise su Conexión de Red", Toast.LENGTH_SHORT)
        .show();
} else {
    TotalInterfaces
        .setText("El total de interfases del Equipo son: "
            + parametros.parametro_t_int);

    DesgolseInterfaces.setText("Interfases: "
        + Arrays.deepToString(parametros.numeros));

    Descripcion.setText(parametros.informacion_nombre + "\n"
        + parametros.informacion_tipo + "\n"
        + parametros.informacion_MTU + "\n"
        + parametros.informacion_BAC + "\n"
        + parametros.Velocidad_del_Puerto + "\n"
        + parametros.Estado_de_Administración + "\n"
        + parametros.Estado_Operativo + "\n"
        + parametros.Ultimo_Cambio);

    ErroresProducidos.setText(parametros.Paquetes_descartados
        + "\n" + parametros.Paquetes_erroneos + "\n"
        + parametros.Paquetes_descartados_por_desconocer
        + "\n" + parametros.Paquetes_descartados_OUT + "\n"
        + parametros.Paquetes_con_errores_OUT);

```

Figura D2: Clase InterfazMIB.java
Elaborado por: Alex Francisco Rodríguez

```

SistemaMIB.java
if (errorStatus == PDU.noError) {
    Log.d(TAG,
        "Snmp Get Response = "
        + responsePDU.getVariableBindings());

    logResult.append("_____+\n");

    String var_respondePDU= String.valueOf(responsePDU.getVariableBindings());
    String respuesta_final = var_respondePDU.replaceAll("1.3.6.1.2.1.1.5.0 = ", "");
    System.out.println("NOMBRE DEL DISPOSITIVO: "+respuesta_final);

    logResult.append("1 NOMBRE DEL DISPOSITIVO: "+\n"+respuesta_final +"\n");

    logResult.append("_____+\n");
    String var_respondePDU1= String.valueOf(responsePDU1.getVariableBindings());
    String respuesta_final1 = var_respondePDU1.replaceAll("1.3.6.1.2.1.1.1.0 = ", "");
    System.out.println("NOMBRE DEL MODELO: "+respuesta_final1);

    logResult.append("2 NOMBRE DEL MODELO: "+\n"+respuesta_final1 +"\n");

    logResult.append("_____+\n");
    String var_respondePDU2= String.valueOf(responsePDU2.getVariableBindings());
    String respuesta_final2 = var_respondePDU2.replaceAll("1.3.6.1.2.1.1.3.0 = ", "");
    System.out.println("TIEMPO DE ACTIVIDAD: "+respuesta_final2);

    logResult.append("3 TIEMPO DE ACTIVIDAD: "+\n"+respuesta_final2 +"\n");

    logResult.append("_____+\n");
    String var_Usoprocesador= String.valueOf(responsePDU3.getVariableBindings());
    String respuesta_Usoprocesador = var_Usoprocesador.replaceAll("1.3.6.1.2.1.25.3.3.1.2 = ", "");
    System.out.println("USO PROCESADOR: "+respuesta_Usoprocesador);
    if (respuesta_Usoprocesador.toString().equals("[noSuchInstance]")) {
        logResult.append("4 USO PROCESADOR (0-100%): "+\n"+"Información no disponible"+\n");
    }
    else {
        logResult.append("4 USO PROCESADOR (0-100%): "+\n"+respuesta_Usoprocesador +%"+\n");
    }
}
}

```

FiguraD3: Clase SistemaMIB.java
Elaborado por: Alex Francisco Rodríguez

```

Conectividad.java
/**
 * INICIO BOTON EJECUTAR PING
 */
bt.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // OBTENEMOS LA POSICIÓN DEL VECTOR
        int posicion1 = sp.getSelectedItemId();
        // ACCIÓN SEGUN LA POSICIÓN SELECCIONADA
        // SERVIDOR DE CORREO
        if (posicion1 == 0) {
            resultado_ping.clearFocus();
            ping();
            resultado_ping.setText(respuesta);
            Toast.makeText(getApplicationContext(), "Acción Culminada",
                Toast.LENGTH_LONG).show();
        }

        // ROUTER DE DISTRIBUCIÓN PLANTA ALTA
        if (posicion1 == 1) {
            resultado_ping.clearFocus();
            ping();
            resultado_ping.setText(respuesta);
            Toast.makeText(getApplicationContext(), "Acción Culminada",
                Toast.LENGTH_LONG).show();
        }

        // ROUTER DE DISTRIBUCIÓN PLANTA BAJA
        if (posicion1 == 2) {
            resultado_ping.clearFocus();
            ping();
            resultado_ping.setText(respuesta);
            Toast.makeText(getApplicationContext(), "Acción Culminada",
                Toast.LENGTH_LONG).show();
        }

        // CLIMATIZACION DATA CENTER
        if (posicion1 == 3) {
            resultado_ping.clearFocus();
        }
    }
});

Conectividad.java
* PING DISPOSITIVOS
public void ping() {
    int count = 0;
    String str = null;
    try {
        Process proceso = null;
        if (Build.VERSION.SDK_INT <= 25) {
            proceso = Runtime.getRuntime().exec(
                "system/bin/ping -w 1 -c 1 " + "190.95.194.130");
            System.out.println("Lo que hace proces: " + proceso);
        }
        else {
            Toast mensaje = Toast.makeText(this, "Version no Soportada",
                Toast.LENGTH_LONG);
            mensaje.show();
        }
        BufferedReader reader = new BufferedReader(new InputStreamReader(
            proceso.getInputStream()));

        StringBuffer output = new StringBuffer();
        String temp;
        while ((temp = reader.readLine()) != null) // .read(buffer) > 0
        {
            output.append(temp);
            count++;
        }
        reader.close();
        if (count > 0)
            str = output.toString();
        proceso.destroy();
    } catch (IOException e) {
        e.printStackTrace();
    }
    // =====REVISION DE LA CONEXION=====
    if (str == null) {
        System.out.println("Revise su Conexion de Red");
        Toast.makeText(getApplicationContext(),
            "Revise su Conexion de Red", Toast.LENGTH_LONG).show();
        respuesta = "SIN CONEXION";
    }
}

```

Figura D4: Clase Conectividad.java
Elaborado por: Alex Francisco Rodríguez

```

EstadoDeLaRedWiFi.java
private void EstadoDeLaRedWiFi() {
    ConnectivityManager myConnManager = (ConnectivityManager) getSystemService(CONNECTIVITY_SERVICE);
    NetworkInfo myNetworkInfo = myConnManager
        .getNetworkInfo(ConnectivityManager.TYPE_WIFI);
    WifiManager myWifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
    WifiInfo myWifiInfo = myWifiManager.getConnectionInfo();

    textMac.setText(myWifiInfo.getMacAddress());

    if (myNetworkInfo.isConnected()) {
        int myIp = myWifiInfo.getIpAddress();

        textConnected.setText("--- CONECTADO ---");

        int intMyIp3 = myIp / 0x1000000;
        int intMyIp3mod = myIp % 0x1000000;

        int intMyIp2 = intMyIp3mod / 0x10000;
        int intMyIp2mod = intMyIp3mod % 0x10000;

        int intMyIp1 = intMyIp2mod / 0x100;
        int intMyIp0 = intMyIp2mod % 0x100;

        textIp.setText(String.valueOf(intMyIp0) + "."
            + String.valueOf(intMyIp1) + "." + String.valueOf(intMyIp2)
            + "." + String.valueOf(intMyIp3));

        textSsid.setText(myWifiInfo.getSSID());
        textBssid.setText(myWifiInfo.getBSSID());

        textSpeed.setText(String.valueOf(myWifiInfo.getLinkSpeed()) + " "
            + WifiInfo.LINK_SPEED_UNITS);
        textRssi.setText(String.valueOf(myWifiInfo.getRssi()));
    } else {
        textConnected.setText("--- SIN CONEXIÓN! ---");
        textIp.setText("---");
        textSsid.setText("---");
        textBssid.setText("---");
    }
}

```

Figura D5: Clase EstadoDeLaRedWiFi.java
Elaborado por: Alex Francisco Rodríguez

```

ClimatizacionDataCenter.java
package alex.tesis.snmphospital;

import android.app.Activity;
import android.os.Bundle;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class ClimatizacionDataCenter extends Activity {
    private WebView browser;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.climatizacion_data_denter);

        browser = (WebView) findViewById(R.id.webView);

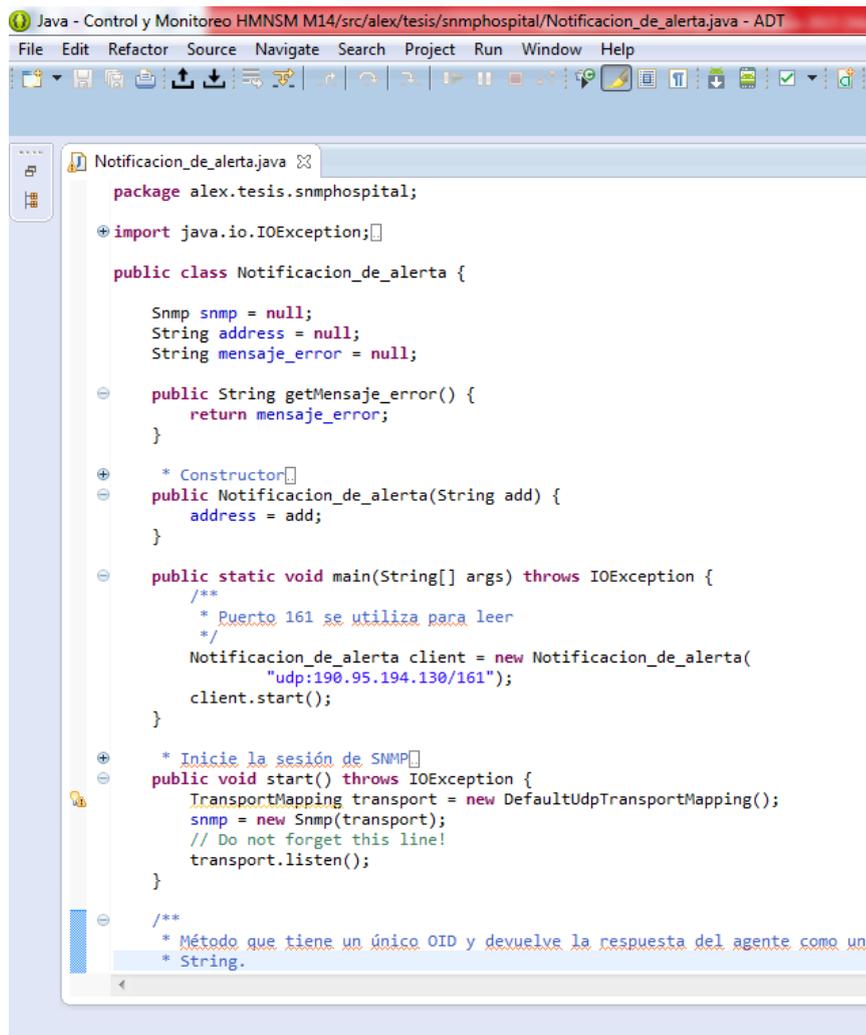
        // HABILITAMOS JAVASCRIPT Y EL ZOOM
        browser.getSettings().setBuiltInZoomControls(true);

        browser.loadUrl("http://190.95.194.130/");

        browser.setWebViewClient(new WebViewClient() {
            // EVITA QUE LOS ENLACES SE ABRA FUERA NUESTRA APP EN EL NAVEGADOR
            // DE ANDROID
            @Override
            public boolean shouldOverrideUrlLoading(WebView view, String url) {
                return false;
            }
        });
    }
}

```

Figura D6: Clase ClimatizacionDataCenter.java
Elaborado por: Alex Francisco Rodríguez



```
Java - Control y Monitoreo HMNSM M14/src/alex/tesis/snmphospital/Notificacion_de_alerta.java - ADT
File Edit Refactor Source Navigate Search Project Run Window Help
Notificacion_de_alerta.java
package alex.tesis.snmphospital;
import java.io.IOException;
public class Notificacion_de_alerta {
    Snmp snmp = null;
    String address = null;
    String mensaje_error = null;
    public String getMensaje_error() {
        return mensaje_error;
    }
    * Constructor
    public Notificacion_de_alerta(String add) {
        address = add;
    }
    public static void main(String[] args) throws IOException {
        /**
         * Puerto 161 se utiliza para leer
         */
        Notificacion_de_alerta client = new Notificacion_de_alerta(
            "udp:190.95.194.130/161");
        client.start();
    }
    * Inicie la sesión de SNMP
    public void start() throws IOException {
        TransportMapping transport = new DefaultUdpTransportMapping();
        snmp = new Snmp(transport);
        // Do not forget this line!
        transport.listen();
    }
    /**
     * Método que tiene un único OID y devuelve la respuesta del agente como un
     * String.
    */
}
```

Figura D7: Clase Notificacion_de_alerta.java
Elaborado por: Alex Francisco Rodríguez