



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE RED DE SENSORES INALÁMBRICOS PARA EL MONITOREO DE LOS PACIENTES. CASO PRÁCTICO: CENTRO DE ATENCIÓN INTEGRAL EN SALUD DE LA ESPOCH”

TESIS DE GRADO

Previa a la obtención del título de

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

Presentado por:

MARÍA ALEXANDRA PILCO LLUMITAXI

VERÓNICA ALEXANDRA ZAVALA ANGAMARCA

RIOBAMBA – ECUADOR

-2015-

AGRADECIMIENTO

Primeramente agradecemos a Dios y a la Virgen por habernos guiado en esta etapa de nuestra vida dándonos fuerza, sabiduría e inteligencia.

Agradecemos infinitamente a nuestros queridos padres y hermanos por siempre apoyarnos, confiar en nosotros y estar siempre a nuestro lado incentivándonos para que culminemos con éxito esta meta.

A nuestro director Ingeniero Tony Flores por su colaboración, al Ingeniero Oswaldo Martínez por su ayuda, su orientación, de igual manera al Ingeniero Alberto Arellano por haber compartido sus conocimientos cuando necesitamos de ellos para culminar este trabajo investigativo.

A la licenciada Lupe Zabala O. que nos brindó su tiempo y conocimiento para la validación de nuestro prototipo.

A nuestros amigos por todo su apoyo y ánimo brindado en el transcurso de los años en la Politécnica.

Ma. Alexandra y Verónica

DEDICATORIA

Al culminar una etapa más en mi vida dedico este logro a mis queridos padres Blanca y Segundo por todo su apoyo, cariño, amor y comprensión brindados en el transcurso de mi carrera, gracias a sus sabios consejos y por ser la fuente donde siempre encontré las fuerzas necesarias para continuar.

A mis queridos hermanos David y Anthony por confiar en mí y estar pendientes de que continúe pujante en lograr cumplir mis sueños siendo ejemplo de perseverancia.

También a mis sobrinitos queridos Karen y Josua que con su alegría me han motivado a seguir adelante, a mi primo Darwin por apoyarme en todo momento y a mi cuñada Jackeline gracias por las palabras de ánimo que me ha brindado.

A mi querida abuelita Rosaura por haber puesto en mí sus sabios consejos para poder culminar con éxito y a toda mi familia por haber confiado en mí y por considerarme ejemplo de superación y perseverancia.

María Alexandra

Este proyecto de finalización de mi carrera lo dedico a mi mami que siempre me apoyo incondicionalmente en todos estos años de estudio, a mis hermanos y mi papi.

A mis profesores que me han llenado de conocimiento y a todos mis amigos y compañeros que han estado presentes durante toda mi carrera.

Verónica

FIRMAS RESPONSABLES Y NOTA

NOMBRE	FIRMA	FECHA
Ing. Nicolay Samaniego, Ph.D DECANO FACULTAD INFORMÁTICA Y ELECTRÓNICA
Ing. Franklin Moreno DIRECTOR DE LA ESCUELA ING. ELECTRÓNICA, TELECOMUNICACIONES Y REDES
Ing. Tony Flores, MsC. DIRECTOR DE TESIS
Ing. Oswaldo Martínez, MsC. MIEMBRO DEL TRIBUNAL
DIRECTOR CENTRO DE DOCUMENTACIÓN

NOTA DE TESIS ESCRITA:

RESPONSABILIDAD DEL AUTOR

“Nosotras, María Alexandra Pilco Llunitaxi y Verónica Alexandra Zavala Angamarca, somos las responsables de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”.

María Alexandra Pilco Llunitaxi

Verónica Alexandra Zavala Angamarca

ÍNDICE DE ABREVIATURAS

ACRÓNIMO	DESCRIPCIÓN
AES	Advanced Encryption Standard (Estándar de Encriptación Avanzada)
ADC	Analog Digital Conversion (Conversión Analógica Digital)
AODV	Ad hoc On Demand Vector
AVR	Augment Voltage Right (Aumentar Voltaje Derecho)
CH	Channel (Canal)
COM	Communications (Comunicaciones)
CPU	Central Processing Unit (Unidad Central de Proceso)
DARPA	Defense Advanced Research Projects (Defensa de Proyectos de Investigación Avanzada)
DCA	Diseño Completamente al Azar
DSSS	Direct Sequence Spread Spectrum (Espectro Ensanchado por Secuencia Directa)
FFD	Full Function Devices (Dispositivos de Función Completa)
Hb	Deoxygenated hemoglobin (Hemoglobina desoxigenada)
IDE	Integrated Development Environment (Entorno de Desarrollo Integrado)

IEEE	Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos)
ISM	Industrial Scientific and Medical
LED	Light Emitting Diode (Diodo Emisor de Luz)
LTP	Line Print Terminal (Impresión de Línea Terminal)
MAC	Medium Access Control (Control de Acceso al Medio)
MySQL	My Structured Query Language
NAM	Network Animator (Animador de Red)
NS2	Network Simulator 2 (Simulador de Red)
NSF	National Science Foundation (Fundación Nacional para la Ciencia)
OSI	Open Systems Interconnection (Interconexión de Sistemas Abiertos)
PAN	Personal Area Network (Red de Área Personal)
PC	Personal Computer (Computadora Personal)
PHY	Physical Layer (Capa Física)
RFD	Device reduced functionality (Dispositivos de Función Reducida)
Rx	Receiving Data (Recepción de Datos)
SUPERTEL	Superintendencia de Telecomunicaciones

SpO2	Blood oxygenation (Oxigenación de la sangre)
TCL	Tool Command Language
TCP/IP	Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo de Internet)
TIC	Information Technology and Communication (Tecnologías de la Información y la Comunicación)
Tx	Data transmission (Transmisión de Datos)
UDP	User Datagram Protocol (Protocolo de Datagrama de Usuario)
USB	Universal Serial Bus
WIFI	Wireless Fidelity (Fidelidad Inalámbrica)
WIMAX	Worldwide Interoperability for Microwave Access (Interoperabilidad Mundial para Acceso por Microondas)
WPAN	Wireless Personal Area Network (Red Inalámbrica de Área Personal)
WSN	Wireless Sensor Network (Red Inalámbrica de Sensores)
ZED	ZigBee End Device (Dispositivo ZigBee Final)
ZC	ZigBee Coordinator (Coordinador ZigBee)
ZR	ZigBee Router (Router ZigBee)

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS RESPONSABLES Y NOTA

RESPONSABILIDAD DEL AUTOR

ÍNDICE DE ABREVIATURAS

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

CAPÍTULO I

MARCO REFERENCIAL	21
1.1 INTRODUCCIÓN	21
1.2 ANTECEDENTES	22
1.3 JUSTIFICACIÓN.....	23
1.4 OBJETIVOS.....	25
1.4.1 Objetivo General:.....	25
1.4.2 Objetivos Específicos:.....	25
1.5 HIPÓTESIS.....	26

CAPÍTULO II

MARCO TEÓRICO	27
2.1 INTRODUCCIÓN	27
2.2 ESTÁNDAR IEEE 802.11 WIFI	28
2.3 ESTÁNDAR IEEE 802.16 WIMAX	29
2.4 ESTÁNDAR IEEE 802.15.1 BLUETOOTH	30
2.5 ESTANDAR IEEE 802.15.4 (ZIGBEE)	32
2.5.1 Zigbee.....	33

CAPÍTULO III	
MARCO TEÓRICO	38
ARDUINO UNO.....	38
3.1 INTRODUCCIÓN	38
3.2 CONCEPTOS BÁSICOS DE ARDUINO.....	39
3.1.1 ¿Qué es Arduino?.....	39
3.1.2 Instalación de Arduino	39
3.1.3 Probar el Arduino.....	40
3.1.4 Hardware Arduino Uno	43
3.1.5 Pines de entrada y salida de Arduino Uno	44
3.1.6 Pines Digitales.....	44
3.1.7 Pines de entrada analógica.....	45
3.1.8 Arduino IDE.....	45
3.1.9 Serial Monitor	47
3.1.10 Lenguaje Arduino.....	47
3.3 SHIELDS XBEE	47
3.4 MÓDULOS INALÁMBRICOS XBEE S1	48
3.4.1 XBee vs XBee-PRO	50
3.4.2 Series 1 vs Series 2	50
3.4.3 Configuración del módulo XBee	50
3.4.4 XBee USB Adapter.....	51
CAPÍTULO IV	
MARCO TEÓRICO	54
RED INALÁMBRICA DE SENSORES	54
4.1 INTRODUCCIÓN	54
4.2 REDES DE SENSORES INALÁMBRICOS.....	55
4.2.1 Arquitectura	55
4.2.2 Aplicaciones.....	58
4.2.3 Características.....	60
CAPÍTULO V	

SIMULACIÓN DE LA RED DE SENSORES EN NS2.....	63
5.1 INTRODUCCIÓN.....	63
5.2 SIMULADOR NS2.....	64
5.2.1 Arquitectura Básica.....	64
5.2.2 Traza Network AniMation (NAM).....	65
5.2.3 Pasos Principales de la Simulación NS2.....	66
5.2.4 Ad hoc On Demand Vector (AODV).....	68
5.2.5 Awk.....	68
5.3 ESTRUCTURA DE LA SIMULACIÓN.....	69
5.3.1 Parámetros de la Simulación.....	70
5.3.2 Variables de estudio.....	72
5.4 CREACIÓN DE LOS ESCENARIOS.....	74
5.4.1 Configuración del simulador NS2.....	74
5.4.2 Configuración del escenario.....	76
5.4.3 Configuración y creación de los nodos inalámbricos.....	77
5.4.4 Configuración de los agentes de tráfico.....	80
5.4.5 Generación de los archivos de traza.....	82
5.5 ESCENARIOS PROPUESTOS.....	85
5.5.1 Escenario 1.....	86
5.5.2 Escenario 2.....	87
5.5.3 Escenario 3.....	88
CAPÍTULO VI	
IMPLEMENTACIÓN DEL PROTOTIPO DE WSN.....	90
6.1 INTRODUCCIÓN.....	90
6.2 DISEÑO DEL PROTOTIPO WSN.....	91
6.3 IMPLEMENTACIÓN DEL PROTOTIPO WSN.....	92
6.3.1 Sensor de Flujo de Aire.....	92
6.3.2 Sensor de Temperatura Corporal.....	93
6.3.3 Sensor de Pulso y Oxígeno en la Sangre (SpO2).....	94
6.4 CONFIGURACIÓN DE LOS MODULOS XBEE S1.....	95

6.5	CONFIGURACIÓN DEL ARDUINO UNO	101
6.6	ELABORACIÓN DE LA BASE DE DATOS	104
6.7	DISEÑO DE LA INTERFAZ	104
6.7.1	Pasos para la creación de la interfaz.....	105
6.7.2	Activación de Alarmas.....	109
6.7.3	Guardar la información	110
6.7.4	Comunicación Puerto Serial	110
6.7.5	Conexión con la Base de Datos	111
6.8	FUNCIONAMIENTO DEL PROTOTIPO DE RED	112
6.8.1	Escenario 1	113
6.8.2	Escenario2	114
6.8.3	Escenario 3	114
CAPÍTULO VII		
ANÁLISIS Y RESULTADOS		116
7.1	INTRODUCCIÓN	116
7.2	COMPROBACIÓN DE LA SIMULACIÓN DEL PROTOTIPO DE RED EN NS-2.....	117
7.2.1	Métodos a utilizar	117
7.2.2	Análisis Comparativo.....	120
7.3	COMPROBACIÓN DE LA IMPLEMENTACIÓN DEL PROTOTIPO DE RED WSN.....	139
7.3.1	Métodos a utilizar	140
7.3.2	Análisis Comparativo.....	140
7.3.3	Comprobación de la Hipótesis.....	152
CONCLUSIONES		
RECOMENDACIONES		
TRABAJO FUTURO		
BIBLIOGRAFÍA		
RESUMEN		
ABSTRACT		
ANEXOS		
Anexo 1 datasheet Arduino uno		

Anexo 2 Datasheet Xbee s1

Anexo 3 Datasheet Shield XBee

ANEXO 4 Instalación de NS2

Anexo 7 Pulse Oximetry Sensor

Anexo 8 e-health sensor shield

Anexo 9 Antena whip

ÍNDICE DE TABLAS

Tabla II.I: Tabla del Estándar IEEE 802.11 con sus velocidades y bandas de frecuencia.....	28
Tabla II.II: Tabla del Comparación WIMAX VS WIFI	30
Tabla II.III. Características Bluetooth vs ZigBee	32
Tabla III.IV: Algunos parámetros de XBee.	51
Tabla V.V Temperaturas Corporales	93
Tabla VI.VI. Porcentaje de la oxigenación en la sangre.....	94
Tabla VI.VII. Pulsaciones por minuto	94
Tabla VI.VIII. Parámetros de configuración XBEE s1	98
Tabla VI.IX. Tabla de la base de datos	104
Tabla VII.X: Datos del método al azar.	118
Tabla VII.XI: Tabla Anova.	119
Tabla VII.XII: Muestra de los datos Escenario 1	120
Tabla VII.XIII: Muestra de los datos Escenario 2	121
Tabla VII.XIV: Muestra de los datos Escenario 3.....	122
Tabla VII.XV: Promedios de las muestras de los tres escenarios con 100s.	123
Tabla VII.XVI: Promedios de las muestras de los tres escenarios con 3600s.	123
Tabla VII.XVII. Varianza del rendimiento (100s)	124
Tabla VII.XVIII. Anova del Rendimiento (100s)	124
Tabla VII.XIX. Varianza del retardo (100s)	126
Tabla VII.XX. Anova del Retardo (100s).....	126
Tabla VII.XXI. Varianza del throughput (100s)	128
Tabla VII.XXII. Anova del Throughput (100s).....	128
Tabla VII.XXIII. Varianza del consumo de energía (100s)	130
Tabla VII.XXIV. Anova del Consumo de Energía (100s)	130
Tabla VII.XXV. Varianza del rendimiento (3600s)	132
Tabla VII.XXVI. Anova del Rendimiento (3600s)	132
Tabla VII.XXVII. Varianza del retardo (3600s).....	134
Tabla VII.XXVIII. Anova del Retardo (3600s).....	134
Tabla VII.XXIX. Varianza del throughput (3600s)	136
Tabla VII.XXX. Anova del Throughput (3600s).....	136
Tabla VII.XXXI. Varianza del consumo de energía (3600s)	138
Tabla VII.XXXII. Anova del Consumo de Energía (3600 s).....	138
Tabla VII.XXXIII. Captura de datos fisiológicos de la manera manual	141
Tabla VII.XXXIV. Captura de datos fisiológicos con el prototipo	143
Tabla VII.XXXV. Varianza del tiempo de respuesta (2m prototipo y 1m manual)	146
Tabla VII.XXXVI. Anova del tiempo de respuesta (2m prototipo y 1m manual).....	146
Tabla VII.XXXVII. Varianza del tiempo de respuesta (15m prototipo y 1m manual)	148

Tabla VII.XXXVIII. Anova del tiempo de respuesta (15m prototipo y 1m manual)	148
Tabla VII.XXXIX. Varianza del tiempo de respuesta (20m prototipo y 1m manual)	150
Tabla VII.XXXX. Anova del tiempo de respuesta (20m prototipo y 1m manual)	150
Tabla VII.XXXXI. Valoración	152
Tabla VII.XXXXII. Resultados	153
Tabla VII.XXXXIII. Promedios	154

ÍNDICE DE FIGURAS

Figura 1: Centro Médico monitoreando.....	24
Figura 2: Red de sensores inalámbricos.....	25
Figura 3: Estructura de la trama 802.11.....	29
Figura 4: Estructura de la trama 802.16.....	29
Figura 5: Espectro de WIMAX.....	30
Figura 6. Formato de la Trama Bluetooth en la Capa Banda Base	31
Figura 7. Bandas de frecuencia ZigBee	33
Figura 8: Capas del Protocolo de Red Inalámbrica ZigBee.	35
Figura 9: Aplicaciones Zigbee	37
Figura 10: Arduino uno	39
Figura 11: Selección de la placa.	41
Figura 12: Selección del puerto serial.	42
Figura 13: Abriendo el programa.....	43
Figura 14: Placas Oficiales Arduino.....	43
Figura 15: Pines de Arduino digitales (parte superior) y pines analógicos (abajo).....	44
Figura 16: Arduino IDE.	46
Figura 17: Arduino entorno de desarrollo.....	46
Figura 18: Monitor Serial de Arduino.....	47
Figura 19: Shields oficiales Arduino.	48
Figure 20. Dos módulos XBee series 1.	49
Figura 21: Configuraciones PAN.	49
Figura 24: Estructura del nodo sensor.	57
Figura 25. Arquitectura básica de NS.....	65
Figura 26: Una comparación de los pasos de simulación generales y pasos de simulación NS2.	66
Figura 28: Estructura de la Simulación.....	69
Figura 29: Parámetros de la simulación.....	70
Figura 30: Parámetros de estudio.....	73
Figura 31: Estructura de la traza inalámbrica.....	83
Figura 32: Estructura del nuevo formato de trazas inalámbricas.....	84
Figura 33: Ejecución de filtros .awk	85
Figura 33: Escenario 1	87
Figura 34: Escenario 2	88
Figura 35: Escenario 3	89
Figura 36: Prototipo de la WSN	91
Figura 37: Conexión del sensor flujo de aire	92
Figura 38: Conexión del sensor de temperatura corporal	93
Figura 39: Conexión del sensor de pulso y oxígeno en la sangre	95

Figura 40: Pestaña PC settings del software X-CTU	96
Figura 41: Pestaña Modem configuration del Software X-CTU	97
Figura 42: Direcciones de los XBee S1 usados.	99
Figura 42: Configuración del CH, PAN ID, DH, DL, MY y AES en el software X-CTU.....	100
Figura 43: Guardando la configuración en el XBee S1.	101
Figura 44. Verificación del programa en arduino.	103
Figura 45. Cargando programa en arduino.	103
Figura 46: Abriendo Delphi XE7	105
Figura 47: Pantalla principal de la interfaz gráfica.....	106
Figura 48: Estructura en bloques de la configuración de la interfaz gráfica.	106
Figura 49: Ventana para el ingresos de pacientes en la interfaz gráfica.....	107
Figura 50: Estructura en bloques de la ventana para el ingresos de pacientes.	107
Figura 51: Ventana de los signos vitales.	108
Figura 52: Estructura de la ventana de la visualización de los signos vitales.....	109
Figura 53: Visualización de los controles.	110
Figura 54: Puerto ComPort en Delphi EX7.....	111
Figura 55: Conexión de la interfaz con la base de datos.	111
Figura 56: Medición de signos vitales con el prototipo.	112
Figura 57: Medición de signos vitales manualmente.....	113
Figura 58: Escenario 1 del prototipo.....	114
Figura 59: Escenario 2 del prototipo.....	114
Figura 60: Escenario 3 del prototipo.....	115
Figura 61: Promedio del Rendimiento (100s).....	125
Figura 62. Promedio del Retardo (100s).	127
Figura 63. Promedio de THROUGHPUT (100s).	129
Figura 64. Promedio de consumo de energía (100s).	131
Figura 65. Promedio de rendimiento (3600s).	133
Figura 66. Promedio de retardo (3600s).	135
Figura 67: Promedio de throughput (3600s).	137
Figura 68: Promedio del consumo de energía (3600s).	139
Figura 69: Tiempo de respuesta (2m prototipo y 1m manual).....	147
Figura 70: Tiempo de respuesta (15m prototipo y 1m manual).....	149
Figura 71: Tiempo de respuesta (20m prototipo y 1m manual).....	151
Figura 72: Temperatura (2m prototipo y 1m manual).....	155
Figura 73: Pulso (15m prototipo y 1m manual).....	155

INTRODUCCIÓN

Este trabajo de tesis busca mejorar el tiempo de respuesta en la medición de los signos vitales en el Centro de atención integral en salud de la ESPOCH por medio de un prototipo de Red de Sensores Inalámbricos.

En la actualidad las instituciones de salud para medir los datos fisiológicos del paciente, en la consulta externa se realiza de manera manual en el caso del pulso se coloca los dedos índice y medio en la parte anterior de la muñeca o en el cuello a lado de la manzana de Adan se cuenta el número de pulsaciones en un minuto, la respiración se coloca la mano sobre la estómago se cuenta el número de respiraciones durante un minuto y para la temperatura se usa un termómetro. Tardando un tiempo promedio de 2 a 5 minutos. Este método no le brinda movilidad al paciente.

Las Redes de sensores Inalámbricos (WSN) hoy en día disminuyen el costo en la implementación debido a la ausencia de cables, conectores y demás implementos para armar una red cableada; ganando espacio por la ausencia de estos. Además permite el desplazamiento de los sensores de la red. La tecnología más usada para WSN es ZigBee debido a que proporciona menor consumo de energía y trabaja en bandas de frecuencia libres dedicadas para investigación.

Para determinar si es factible la implementación de este prototipo, se realizó una simulación con varios escenarios en el Software NS2. En donde se analizan varios parámetros para comprobar la fiabilidad de la red. Después procedemos al diseño e implementación del prototipo; en donde se configura la placa electrónica [Arduino uno] para que tome los signos vitales a través de los sensores de temperatura corporal, flujo de aire, pulso y oxígeno en la sangre. Para la transmisión

inalámbrica se utilizan los módulos XBee S1. En la recepción de datos se configura una interfaz gráfica en Delphi XE7.

Luego del análisis de datos determinamos que el prototipo brinda un tiempo de respuesta mucho menor que el método manual, obteniendo mayores ventajas en la toma de muestras y estableciendo que si es factible su uso en casos de emergencia.

CAPÍTULO I

MARCO REFERENCIAL

1.1 INTRODUCCIÓN

En el presente capítulo se realizará el estudio del arte de las redes de sensores inalámbricos, su importancia en la actualidad y sus diversas aplicaciones tanto en el mundo como en el país, como por ejemplo en la agricultura para monitorear cultivos, en la industria, entre otras múltiples aplicaciones; también se va a especificar el estándar inalámbrico que se usará en este proyecto.

Se realizará un análisis para demostrar porque se eligió trabajar con este proyecto de grado, que se aplicará en el Centro de Atención Integral en Salud en la ESPOCH. Además se definirá los objetivos que se han planteado alcanzar con este proyecto.

1.2 ANTECEDENTES

La evolución del mundo de las redes ha permitido que la sociedad actual pueda compartir información. La tecnología ha logrado crear sistemas capaces de almacenar y procesar una gran cantidad de información y transmitirla inalámbricamente a diferentes lugares alcanzando mayores distancias.

Las Wireless Sensor Networks (WSN) permiten realizar funciones de control, administración, gestión y monitoreo de los datos en tiempo real, actividades que se pueden ejecutar desde cualquier ubicación a través del internet; brindando una infinidad de aplicaciones en todas las áreas. En la agricultura se ha implementado el uso de sensores para controlar remotamente la temperatura, humedad, etc., en los cultivos. Además las redes de sensores inalámbricas en el campo de la salud han dado movilidad a los pacientes. Incluso existen aplicaciones militares donde se usan estas redes para ubicar a personas y monitorear a las mismas.

La necesidad de monitoreo constante de un paciente han permitido la aplicación de WSN que son capaces de medir todo tipo de datos fisiológicos y a la vez transmitir la información a lugares locales o remotos, en donde el médico va a visualizar estos datos y dar un diagnóstico o mantener un control del estado del paciente como puede darse el caso de un paciente hipertenso, diabético, etc.

El estándar IEEE 802.15.4 es un sistema de comunicación de corto alcance, usado en redes de área personal inalámbricas (WPAN) con dispositivos de bajo costo, bajo consumo de energía y su principal área de aplicación es en las WSNs. El protocolo ZigBee es una tecnología basado en el estándar IEEE 802.15.4 el cual especifica la capa física y la capa de control de acceso al medio (MAC), para transmisiones de baja potencia se puede utilizar para redes inalámbricas PAN, específicamente para WSNs como es domótica, control de invernaderos, etc.

1.3 JUSTIFICACIÓN

Gracias al avance de la tecnología hoy en día es posible utilizar las redes de sensores inalámbricas (WSN) aplicadas al campo de la medicina y cuidado de la salud, utilizando los datos enviados por estos dispositivos para llevar un control del estado del paciente, además se va a analizar el tipo de transmisión, así como también los tipos de sensores que se utilizan para estas aplicaciones para optimizar los resultados a los pacientes y personas en general, brindando así un apoyo a la medicina mediante el uso de esta tecnología lo que se le conoce con el nombre de telemedicina.

En los hospitales es evidente el aumento de su espacio físico y por ende el tiempo que involucra monitorear a todos ellos, esto hace necesario el uso de un dispositivo que se encargue de vigilar el estado del paciente y envíe la información precisa y en tiempo real a la persona encargada de monitorear los datos fisiológicos de cada paciente. Figura 1.

Gracias al desarrollo de las TICs en la actualidad se ha logrado contribuir al desarrollo de nuevos servicios para la medicina proporcionando herramientas para hacer posible la monitorización de enfermedades, detectar situaciones extrañas en la evolución o en los signos vitales del paciente como es la temperatura, la presión sanguínea, la frecuencia cardíaca entre otras. Dar un aviso de alarma, de esta manera actuar de forma inmediata y brindar la ayuda necesaria al paciente; de aquí surge este proyecto para ayudar de manera más precisa a monitorear a los pacientes en un hospital.

Los sensores inalámbricos en comparación con los sensores tradicionales son inteligentes brindando una respuesta inmediata es decir en tiempo real, permitiendo tener almacenados los

datos con información actualizada y además no se encuentran limitados en distancias por el uso de cables. Las redes de sensores inalámbricas en comparación con las redes tradicionales ayudan a reducir los costos de implementación, también el espacio físico utilizado es mucho menor. Figura 2.

Se va a desarrollar un proyecto que se basa en el diseño e implementación de un prototipo de control y monitorización de los signos vitales de una persona; basado en tecnología WSN en el cual se medirán varios parámetros y ver qué tan eficiente es usando la tecnología ZigBee. Esto va a ser posible debido a la existencia de un Centro de Atención Integral en Salud en la ESPOCH, para la finalización de una carrera de ingeniería.

DIAGRAMA DE LA RED DE SENSORES INALAMBRIOS.

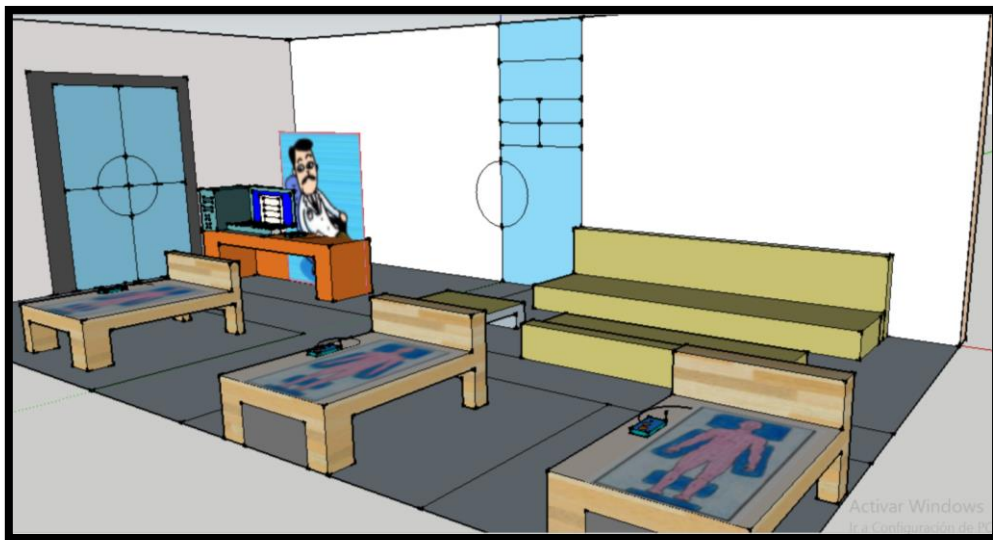


Figura 1: Centro Médico monitoreando.

Fuente: Elaboración propia.

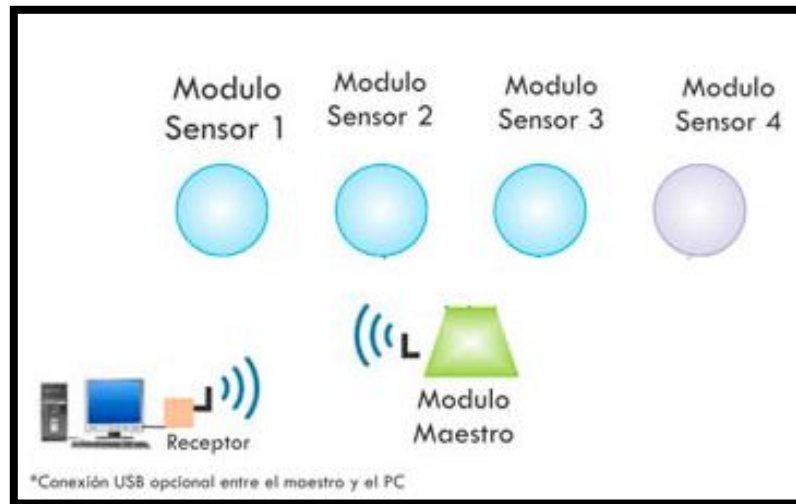


Figura 2: Red de sensores inalámbricos.

Fuente: <http://upcommons.upc.edu/pfc/bitstream/2099.1/9511/1/memoria.pdf>

1.4 OBJETIVOS

1.4.1 Objetivo General:

Diseñar e implementar un prototipo de red de sensores inalámbricos para el monitoreo de los pacientes, caso práctico en el Centro de atención integral en salud de la ESPOCH.

1.4.2 Objetivos Específicos:

- Recopilar información sobre los estándares inalámbricos, Arduino, módulo XBee S1 y las redes de sensores inalámbricos.
- Programar el Arduino uno para que procese la información sobre los signos vitales del paciente por medio de los sensores y configurar los módulos XBee S1 como transmisor y receptor en el prototipo WSN.
- Construir diferentes escenarios de la red WSN en el Simulador de Redes NS2 para determinar diferencias en los factores.

- Construir el prototipo de red WSN para la toma de datos y aplicar el prototipo de red en el Centro de Atención Integral en Salud de la ESPOCH.

1.5 HIPÓTESIS

Es posible mediante el diseño e implementación de un prototipo de red de sensores inalámbricos para el monitoreo de los pacientes en el Centro de atención integral en salud de la ESPOCH mejorar el tiempo de respuesta y brindar un mejor servicio en casos de emergencia.

CAPÍTULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN

La tecnología inalámbrica ha permitido transmitir información a mayores distancias, brindando a los usuarios movilidad. En este capítulo se analizarán algunos de los tipos de tecnologías inalámbricas más usadas, analizaremos algunas de sus características como: su alcance, bandas de frecuencia, velocidades de transmisión, etc.

Primero se investiga a las tecnologías que tienen mayor alcance como WIFI y WIMAX. Después seguiremos con el estudio de la tecnología BLUETOOTH que maneja distancias de cobertura mucho menores, usada en la conexión de dispositivos móviles como teléfonos; para finalmente

centrarnos en la tecnología que vamos a utilizar en nuestro proyecto que es ZIGBEE, esta tecnología es más usada en aplicaciones con redes de sensores inalámbricos.

2.2 ESTÁNDAR IEEE 802.11 WIFI

El estándar inalámbrico IEEE 802.11 tiene un alcance de 200m en zonas que no existan obstáculos, por esta razón se usa en campus educativos e instituciones etc., que necesiten un punto de acceso no extenso. Para que esta tecnología tenga mayor alcance se deben usar repetidoras. El estándar IEEE 802.11 se usa en redes inalámbricas de área local (WLAN) basadas en puntos de acceso. En la tabla II.I se muestra los estándares IEEE 802.11 con sus velocidades y bandas de frecuencia.

Tabla II.I: Tabla del Estándar IEEE 802.11 con sus velocidades y bandas de frecuencia

Estándar IEEE	Velocidad	Banda de frecuencia
802.11	1 Mbps – 2 Mbps	2.4 GHz
802.11 ^a	Hasta 54 Mbps	5 GHz
802.11b	5.5 Mbps – 11 Mbps	2.4 GHz
802.11g	Hasta 54 Mbps	2.4 GHz
802.11 n	100 Mbps	2.4 GHz and 5 GHz

Fuente: Elaboración propia

Para la seguridad en WIFI se utilizan tres encriptaciones que son WEP, WAP, WPA2. En la tabla II.II se muestra la estructura de la trama del estándar 802.11.

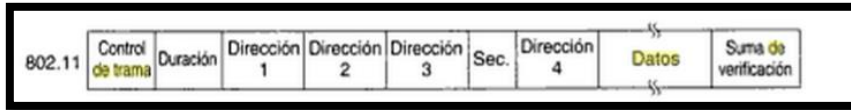


Figura 3: Estructura de la trama 802.11

Fuente:<https://books.google.com.ec/books?id=WWD-4oF9hjEC&pg=PA321&dq=trama++de+datos+802.11&hl=es&sa=X&ei=TWLGVP6wCaPfsATV6YGgCg&ved=0CDcQ6AEwBQ#v=onepage&q=trama%20%20de%20datos%20802.11&f=false>

2.3 ESTÁNDAR IEEE 802.16 WIMAX

El estándar IEEE 802.16 se ha creado con el propósito de reemplazar múltiples tecnologías por las altas velocidades que ofrece y amplio alcance, además por ser una solución de bajo costo. WIMAX llega a distancias de hasta de 50km, transmitiendo hasta una velocidad de 280 Mbps. La estructura de la trama del estándar 802.16 se muestra en la Figura 4.



Figura 4: Estructura de la trama 802.16

Fuente:<https://books.google.com.ec/books?id=WWD-4oF9hjEC&pg=PA321&dq=trama++de+datos+802.11&hl=es&sa=X&ei=TWLGVP6wCaPfsATV6YGgCg&ved=0CDcQ6AEwBQ#v=onepage&q=trama%20%20de%20datos%20802.11&f=false>

Este estándar se usa en redes inalámbricas de área metropolitana. Este estándar trabaja en bandas de frecuencias licenciadas o no licenciadas, las mismas que muestran en la figura 4 según el país en el que aplique. La ventaja de utilizar una banda licenciada es que no va a existir interferencia; mientras si se usa una banda no licenciada se puede implementar de inmediato, pero con el riesgo de interferencia. En la Figura 5 se visualiza las bandas de frecuencia de WIMAX.

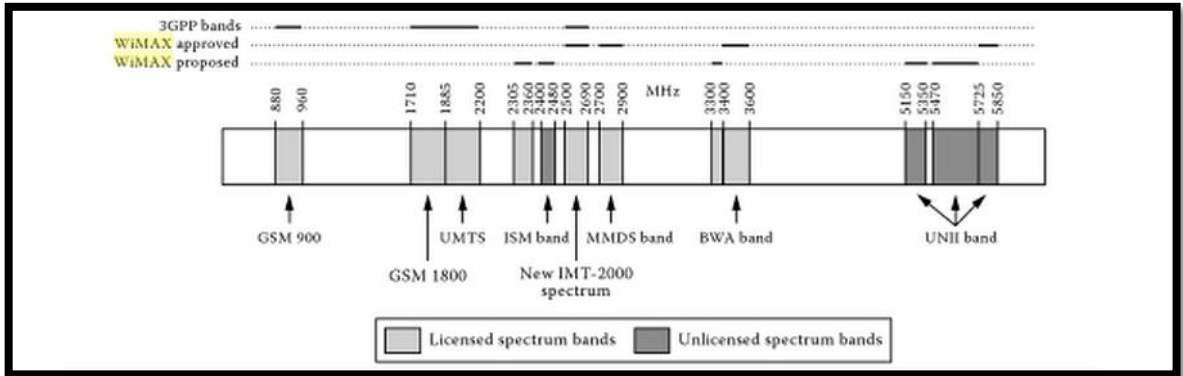


Figura 5: Espectro de WIMAX.
Fuente: www.wimaxforum.org.

En la tabla II.II se observa las características de WIMAX VS WIFI.

Tabla II.II: Tabla del Comparación WIMAX VS WIFI

Característica	WiMax (802.16a)	Wifi (802.11b)	Wi-Fi (802.11a / g)
Redes de aplicación	Red inalámbrica de acceso de banda ancha	Redes Inalámbricas de área local	Redes Inalámbricas de área local
Banda de frecuencia	Licenciado/sin licencia 2 G a 11 GHz	2,4 GHz ISM	2,4 GHz ISM (g) 5 GHz U-NII (a)
Ancho de banda del canal	Ajustable 1,25 M a 20 MHz	25 MHz	20 MHz
Modo de transmisión Half/Full duplex	Full Duplex	Half Duplex	Half Duplex

2.4 ESTÁNDAR IEEE 802.15.1 BLUETOOTH

Bluetooth es una tecnología inalámbrica para distancias pequeñas, diseñado para comunicarse entre dos equipos, permitiendo la transmisión inalámbrica de información de tipo multimedia. Bluetooth proviene del estándar IEEE 802.15.1 que es un estándar orientado a dispositivos de bajo costo y de bajo consumo de energía para distancias entre 1m a 10m.

Los protocolos que se utilizan en una comunicación Bluetooth son similares a los que se emplean con tecnologías de infrarrojos, por lo que no hace falta desarrollar otros nuevos, pero mientras en una comunicación por infrarrojos se requiere un enlace visual entre dispositivos, con Bluetooth no es necesario, ya que emite en todas direcciones y no tiene problemas con los obstáculos.

Bluetooth trabaja en la banda de frecuencia de los 2.4 GHz, al igual que el estándar WIFI pero a diferencia de este trabaja en dos modos, tiene varias velocidades y una cobertura de 10m. Esta tecnología es utilizada para redes de área personal (PAN). Es la tecnología inalámbrica con menor velocidad de transmisión en relación con las otras tecnologías por lo que es recomendado solo la transferencia de datos de pequeño tamaño.

La estructura de la trama Bluetooth como se puede observar en la Figura 6, comienza con un código de acceso que se encarga de identificar al maestro, con el objetivo de que los esclavos que se encuentran en el área de cobertura de dos maestros identifiquen cual tráfico es para ellos. Tiene una cabecera de 54 bits que contiene campos comunes de la subcapa (MAC) de 18 bits que se repite tres veces para el total de los bits. El campo de Datos acepta hasta 2744 bits para transmisiones de cinco ranuras y de 240 bits para transmisiones de una sola ranura de tiempo.



Figura 6. Formato de la Trama Bluetooth en la Capa Banda Base

Fuente: http://ldc.usb.ve/~poc/RedesII/Grupos/G1/como_funciona.html

En la Tabla II.III. Se puede observar las principales características de Bluetooth vs ZigBee:

Tabla II.III. Características Bluetooth vs ZigBee

Características	Bluetooth	ZigBee
Duración de la Batería	Días	Años
Seguridad	64 bit – 128 bit	128 bit AES
Tiempo de acceso a la red	3s	30s
Frecuencia de operación	2,4 GHz ISM	868 MHz, 902-928 MHz, 2,4 GHz ISM
Tiempo de acceso al canal	2ms	15ms
Aplicaciones	Sincronización de Pcs, teléfonos celulares es decir transferencia media de datos y servicio ininterrumpido.	De baja transferencia de datos y ciclos de servicio bajos como es: control, red de sensores, redes con muchos dispositivos.
Tasa de transferencia	750 Kbps	250 Kbps
Número de nodos	8 nodos en una subred	65536 por red y 255 nodos por subred.

Fuente: Elaboración propia.

2.5 ESTANDAR IEEE 802.15.4 (ZIGBEE)

ZigBee proviene del estándar IEEE 802.15.4 que forma parte de un grupo de estándares destinados a reglamentar la realización de redes personales inalámbricas, o WPANs, de baja demanda de rendimiento y baja latencia en la red, empleando dispositivos económicos que consumen poca energía. Su principal campo de aplicación del estándar IEEE 802.15.4 son las redes de sensores inalámbricas, la misma que se enfoca en la estandarización de las dos capas inferiores que son la capa física (PHY) y la capa de control de acceso al medio (MAC).

La capa física del estándar IEEE 802.15.4 trabaja en tres bandas diferentes dependiendo de la zona geográfica donde esté operando el sistema como se puede observar en la Figura 7: la primera es la

banda de los 868/915 MHz que soporta un solo canal entre 868 y 868.6 MHz; la segunda banda está comprendida entre los 902.0 y 928.0 MHz soportando 10 canales y la tercera banda es de 2.4 GHz que soporta 16 canales comprendidos entre 2.4 y los 2.4835 GHz.

Frequency Band	License Required?	Geographic Region	Data Rate	Channel Number(s)
868.3 MHz	No	Europe	20kbps	0
902-928 MHz	No	Americas	40kbps	1-10
2405-2480 MHz	No	Worldwide	250kbps	11-26

Figura 7. Bandas de frecuencia ZigBee

Fuente: <http://rua.ua.es/dspace/bitstream/10045/1109/1/InformeTecZB.pdf>

Esta capa se encarga de la activación y desactivación del radio transceptor de la comunicación, liberación del canal, selección del canal, la modulación y demodulación de las tramas, la transmisión y recepción de la información y la sincronización del Tx y Rx en el momento de la transmisión, así como también Direct Sequence Spread Spectrum (DS-SS) ayuda a reducir el nivel de interferencias en las bandas sin licencia que son compartidas.

2.5.1 Zigbee

ZigBee es una tecnología de comunicación inalámbrica usada para la transmisión de datos a baja velocidad y para redes inalámbricas de corto alcance. La tasa mínima de transferencia de datos es de 20Kbps y su tasa máxima de transferencia de datos es de 250 Kbps, así como también sus aplicaciones están orientadas a bajo costo y las baterías tienen tiempo de duración. ZigBee puede operar en las bandas de frecuencia de 868 MHz o 915 MHz pero preferentemente opera en la banda de los 2.4 GHz, las redes ZigBee pueden estar conformadas de máximo 65536 nodos

distribuidos en subredes de 255 nodos, así como también ofrece diferentes topologías de red como es punto a punto, estrella, malla y árbol.

Los dispositivos ZigBee pasan la mayor parte de su tiempo en el modo de suspensión o en un modo de ahorro de energía lo que le permite el menor consumo de energía al dispositivo y así aumentar el tiempo de vida de las baterías.

2.5.1.1 Estructura del Estándar IEEE 802.15.4

La estructura de las capas de ZigBee se referencia en base del modelo OSI, que divide al protocolo de red en capas lo que hace que tenga algunas ventajas. Como se puede ver en la Figura 8 las dos primeras capas están dadas por la tecnología IEEE 802.15.4 que da las especificaciones para la capa Física y la capa de Acceso al Medio en la red inalámbrica.

Las siguientes dos capas superiores como son la capa de Red que es la que se encarga del ruteo de los paquetes y la capa de Aplicación que es la capa del ultimo nivel y es la que permite la interacción con el usuario, estas capas están definidas por el estándar IEEE 802.15.4 y al unir las cuatro capas se tiene la Red Inalámbrica ZigBee.

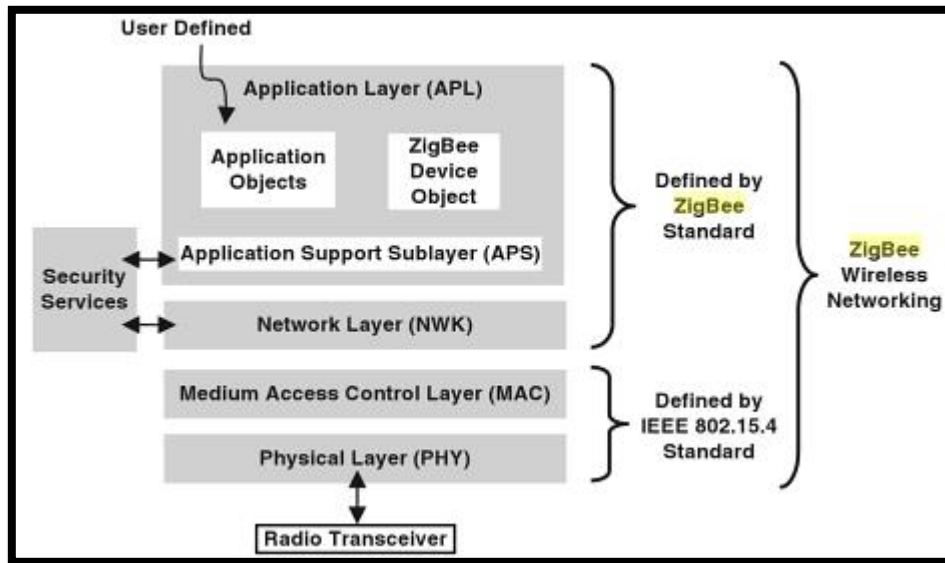


Figura 8: Capas del Protocolo de Red Inalámbrica ZigBee.

Fuente:

<https://books.google.com.ec/books?id=m5NYbUpqXY0C&pg=PA9&dq=ieee+802.15.4+Zigbee&hl=es&sa=X&ei=z2mYVPzZHsmTyAT4oYKYAw&ved=0CEMQ6AEwAw#v=onepage&q=ieee%20802.15.4%20Zigbee&f=false>

2.5.1.2 Dispositivos Zigbee

Existen dos tipos de dispositivos ZigBee que son:

- Dispositivos de Función Completa (FFD) que son capaces de realizar todas las funciones que necesita la tecnología ZigBee, es decir puede transmitir información con cualquier otro dispositivo dentro de la red.
- Dispositivos de Función Reducida (RFD) tienen capacidades limitadas dentro de la red, así como también la potencia de procesamiento y el tamaño de memoria de los dispositivos RFD es menor que de los dispositivos FFD.

2.5.1.3 Roles de los Dispositivos Zigbee

Los roles de los dispositivos son:

- Coordinador ZigBee (ZC) se requiere uno por red, es el encargado de iniciar la formación de la red, asignación de direcciones y es el coordinador en la red.
- Router ZigBee (ZR) es un componente opcional en la red ya que se asocia con el coordinador ZigBe.
- Dispositivo Final (ZED) es esencial en las redes reales porque realiza funciones de detección y control dentro de la red, con la característica de que no puede transmitir información a otros equipos por lo que la mayoría del tiempo pasa en estado de sleep (dormido).

2.5.1.4 Tráfico de datos ZigBee/IEEE 802.15.4.

Dentro del tráfico de datos que existe tenemos:

- Tráfico de Datos Periódicos aquí la aplicación determina la frecuencia del tráfico del dato, es decir el sensor entra en estado activo para verificar los datos, después de esta acción el sensor vuelve a su estado desactivado.
- Tráfico de Datos Intermitentes los datos se interrumpen periódicamente debido a alguna acción producida en la red, de esta manera se mantiene el ahorro de energía en la red.
- Tráfico de Dato Repetitivo se establece periodos de tiempo fijo para la conexión.

2.5.1.5 Seguridad de la tecnología ZigBee

La tecnología ZigBee utiliza la seguridad de la capa MAC del estándar IEEE 802.15.4 en la que mantiene un control de acceso a la red, la transmisión de los datos es de manera encriptado utilizando código de 128 bits, protección y verificación de las tramas que sean igual que las enviadas al inicio del mensaje.

2.5.1.6 Aplicaciones de la Tecnología ZigBee

Dentro de las aplicaciones más comunes son las aplicadas a la domótica, otros ejemplos es el uso en la industria para control de procesos de alto riesgo, en la medicina para monitorear remotamente a pacientes, en la agricultura para control de temperatura de invernaderos o cultivos en general, para sistemas de seguridad, etc. Como se puede ver en la Figura 9.



Figura 9: Aplicaciones Zigbee

Fuente: <http://sx-de-tx.wikispaces.com/ZIGBEE>

CAPÍTULO III

MARCO TEÓRICO

ARDUINO UNO

3.1 INTRODUCCIÓN

En este capítulo se va a conocer más de cerca lo que es la placa Arduino uno, cuáles son sus componentes, formas de configuración y su manera de comunicación. Además se encontrara características de Shields XBee, XBee USB Adapter y del módulo XBee y la forma de funcionamiento del módulo XBee.

3.2 CONCEPTOS BÁSICOS DE ARDUINO

3.1.1 ¿Qué es Arduino?

Arduino es una plataforma de código abierto compuesto por un microcontrolador, un lenguaje de programación Arduino, y un IDE (entorno de desarrollo integrado).



Figura 10: Arduino uno

Fuente: Elaboración propia

3.1.2 Instalación de Arduino

Para trabajar con Arduino primero se debe comprar una placa Arduino y un cable USB standart (para conectar al computador).

Arduino se ejecuta en Windows, Mac OS X y Linux, por lo que hay una versión de Arduino para cualquiera de estos sistemas operativos. En el sitio web del software de Arduino en <http://arduino.cc/en/Main/Software> se descarga la versión del software compatible con nuestro sistema operativo.

3.1.2.1 Instalación en Windows

Se descarga el archivo de Windows .zip y se descomprime. Esta carpeta de Arduino se guarda en un lugar específico de la computadora para después ejecutar el programa. La carpeta Archivos de programa podría ser un lugar adecuado para el software de Arduino. Es necesario instalar los controladores Arduino antes de poder empezar a trabajar con la placa. A continuación se describen estos pasos:

- Se conecta la placa al equipo y se espera a que Windows inicie el proceso de instalación del controlador.
- Hacer clic en el menú Inicio y abrir el Panel de control.
- Ir a la red y la seguridad, luego Sistema y luego en Administrador de dispositivos.
- Buscar el puerto Arduino Uno que aparece en Puertos (COM y LPT).
- Hacer clic derecho sobre él y seleccionar "Actualizar software de controlador", seleccionando "Buscar en mi sistema software de controlador."
- Por último, navegar y seleccionar el archivo controlador de Arduino Uno llamado ArduioUNO.inf, que se encuentra en la carpeta de controladores dentro de la carpeta del software de Arduino que se descargó. Windows instalará con éxito la placa ahora.

3.1.3 Probar el Arduino

Una vez instalado el software y los controladores de Arduino, es necesario realizar una prueba para verificar su funcionamiento.

Entre los numerosos ejemplos de Arduino que se incluyen con el Arduino IDE, hay un ejemplo llamado Blink que hace que este LED incorporado comience a parpadear cada segundo. Se va a utilizar este ejemplo como una manera de averiguar si el Arduino puede comunicarse

correctamente con el ordenador y si los ejemplos que vienen incluidos pueden ser cargados a la placa.

Se conecta la placa Arduino al ordenador y se ejecuta el software de Arduino. En el menú Herramientas, seleccionamos la placa Arduino Uno como se muestra en la Figura 11.

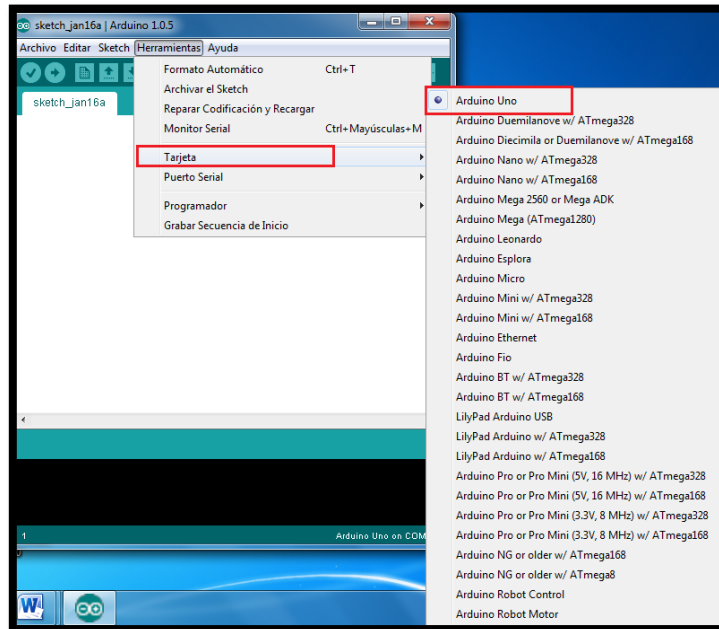


Figura 11: Selección de la placa.

Fuente: Elaboración propia

Después se selecciona el dispositivo serie en la opción de puerto serie en el menú Herramientas.

En Windows el puerto es probable que sea COM4 o superior como se observa en la Figura 12.

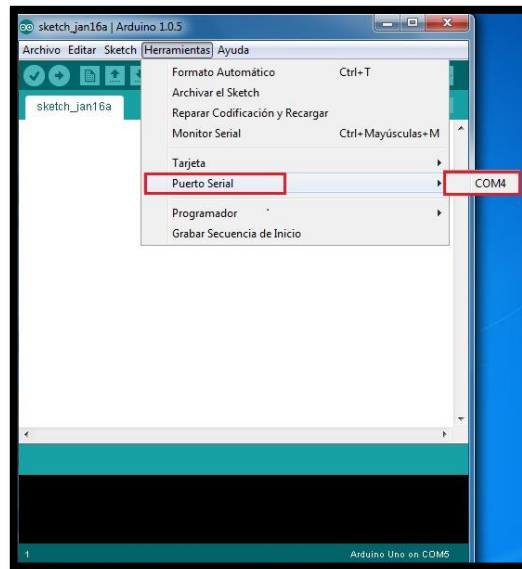


Figura 12: Selección del puerto serial.

Fuente: Elaboración propia.

Ahora se carga el programa a la placa Arduino como se puede observar en la Figura 13. Para ello se da clic en el botón Abrir de la barra de herramientas y se busca el programa que se desea cargar en este caso "arduinoprograma". A continuación, subir el programa haciendo clic en el botón Upload. Después de algunos segundos, se visualiza un mensaje que dice "Carga terminada".

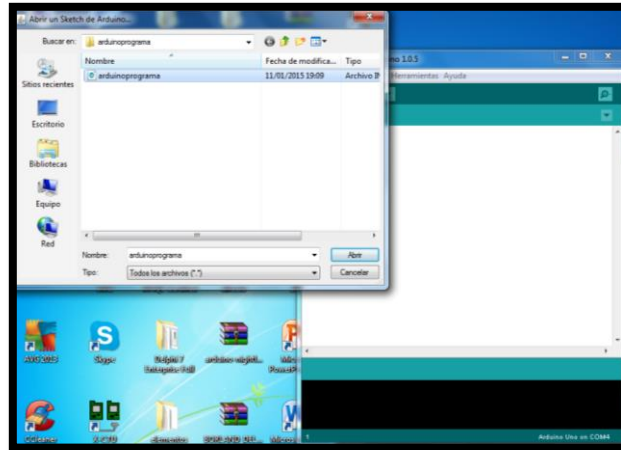


Figura 13: Abriendo el programa

Fuente: Elaboración propia.

3.1.4 Hardware Arduino Uno

Arduino es como un CPU pequeño que permite conectar entre sí a varios dispositivos electrónicos como sensores, alarmas, actuadores, etc. Para diferentes aplicaciones.

La placa Arduino uno consta de un microcontrolador Atmel AVR de 8-bit. Existen diferentes tipos de placas arduino que se pueden ver la Figura 14. El Arduino Uno en el momento de la escritura es la placa Arduino más simple de usar.

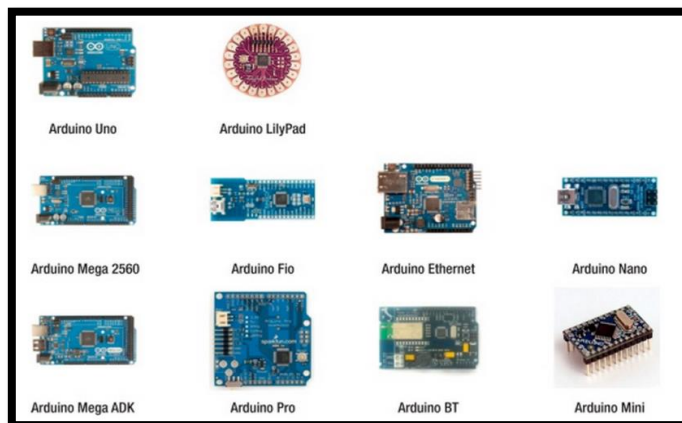


Figura 14: Placas Oficiales Arduino

Fuente: <http://arduino.cc/en/Main/Hardware>

3.1.5 Pines de entrada y salida de Arduino Uno

El Arduino Uno tiene 14 pines digitales de entrada / salida y seis pines de entrada analógicos, como se muestra en la Figura 15.

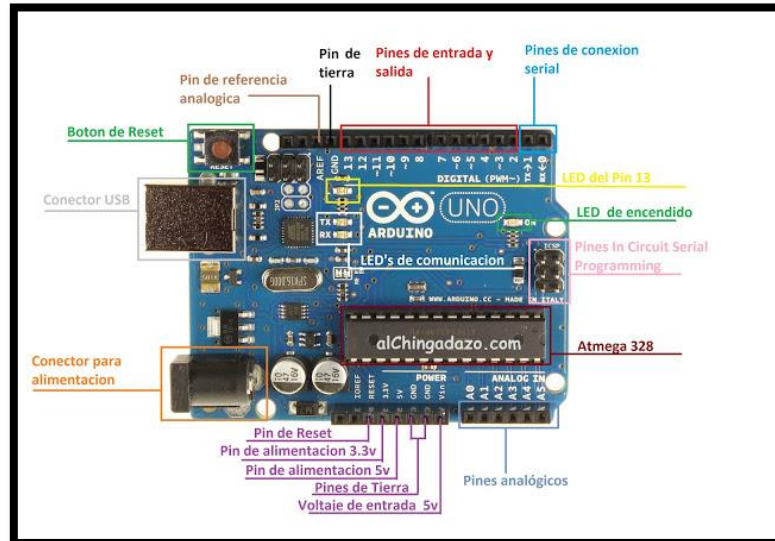


Figura 15: Pines de Arduino digitales (parte superior) y pines analógicos (abajo)

Fuente: <http://electronicabyenel9.blogspot.com/>

3.1.6 Pines Digitales

Arduino uno tiene 14 pines digitales, numerados de 0 a 13. Los pines digitales se pueden configurar como entrada o salida, utilizando la función `pinMode ()`. En ambos modos, los pines digitales sólo pueden enviar o recibir señales digitales, que consisten en dos estados diferentes: ON (ALTAS, o 5V) y OFF (bajo, o de 0 V).

Los pines se pueden establecer como OUTPUT (salida) y pueden proporcionar corriente a dispositivos externos. Los pines de INPUT (entrada) están listos para leer las corrientes de los dispositivos conectados a ellos.

3.1.7 Pines de entrada analógica

Los microcontroladores ATmega utilizados en la placa Arduino contienen seis canales de analógico a digital (ADC). La función de este dispositivo es para convertir una tensión de entrada analógica en un número digital proporcional a la magnitud de la tensión de entrada en relación con la tensión de referencia (5V). Ver Anexo 1

El microcontrolador de ATmega en su placa Arduino tiene una resolución de 10 bits, lo que significa que devolverá números enteros de 0 a 1023 de forma proporcional basado en el potencial que presenta la solicitud en comparación con la referencia de 5V.

Estos pines se establecen como pines de entrada o de salida exactamente como sus pines digitales, llamándolos A0, A1, etc.

3.1.8 Arduino IDE

El IDE de Arduino incluye un editor de código fuente que es un editor de texto con algunas características útiles para la programación y un compilador que convierte el código que ha sido escrito en las instrucciones legibles por código máquina. La Figura 16 muestra el Arduino IDE.

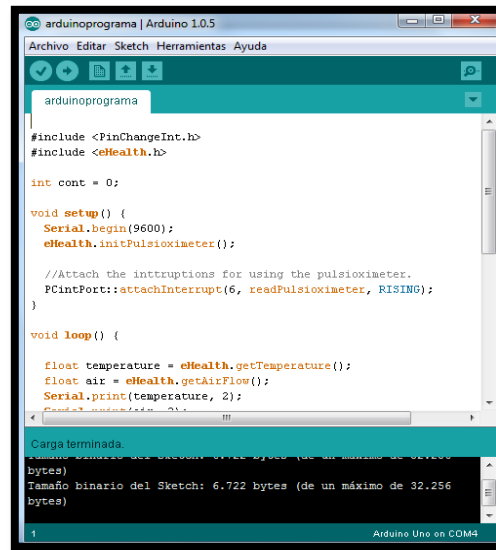


Figura 16: Arduino IDE.

Fuente: Elaboración propia

La Figura 17 ofrece una descripción de todos los botones que se encontrará en la barra de herramientas.

Symbol	Description
	Verify Checks your code for errors.
	Upload Compiles your code and uploads it to the Arduino I/O board.
	New Creates a new sketch.
	Open Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window.
	Save Saves your sketch.
	Serial Monitor Opens the serial monitor.

Figura 17: Arduino entorno de desarrollo

Fuente: <http://arduino.cc>

3.1.9 Serial Monitor

El Monitor de serie (en la parte superior derecha del Arduino IDE) es una herramienta para visualizar los datos en serie que se envían desde la placa Arduino al ordenador.

El monitor serie es muy útil en la depuración de los programas de Arduino, permitiendo imprimir mensajes en el monitor serie para ver el estado del programa en tiempo real como se observa en la Figura 18.

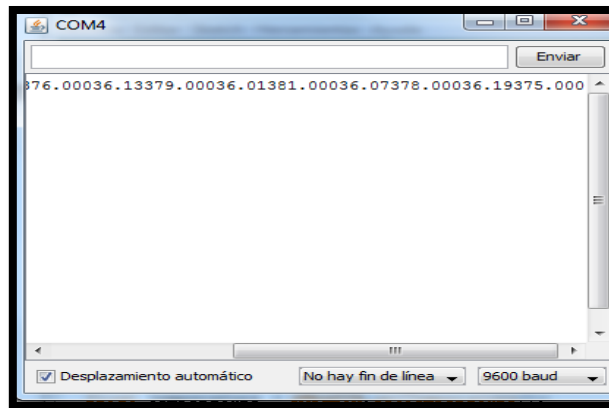


Figura 18: Monitor Serial de Arduino

Fuente: Elaboración propia.

3.1.10 Lenguaje Arduino

El lenguaje de Arduino está implementado en C / C ++ y basado en Wiring. Cuando se escribe un programa Arduino, se está haciendo implícitamente el uso de la biblioteca de Wiring, que se incluye con el Arduino IDE. Esto le permite hacer programas ejecutables mediante el uso de sólo dos funciones: `setup ()` y `loop ()`.

3.3 SHIELDS XBEE

Los Shields de Arduino son placas compatibles con Arduino. Se conectan al módulo XBee S1 y dan una comunicación inalámbrica a la placa Arduino Uno, siendo aplicada para redes de sensores

inalámbricas. Estas placas se colocan en la parte superior del Arduino. En la Figura 19 se puede visualizar algunos ejemplos de Shields Arduino.

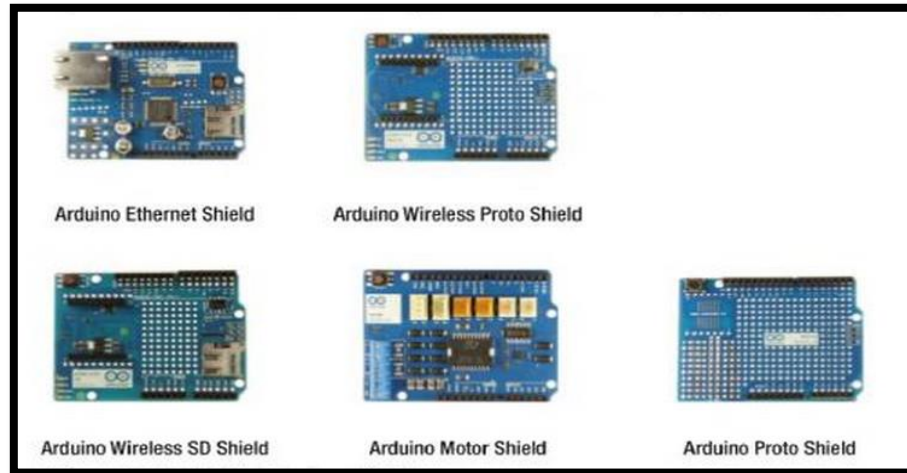


Figura 19: Shields oficiales Arduino.

Fuente: <http://arduinodesdecero.blogspot.com/2013/10/arduino-shields-escudos.html>

3.4 MÓDULOS INALÁMBRICOS XBEE S1

Los transmisores inalámbricos transmiten datos en forma de ondas de radio a través del espacio libre por medio de la radiación electromagnética a una frecuencia particular.

En la república del Ecuador la Superintendencia de Telecomunicaciones (SUPERTEL) es la organización encargada de regular el espectro y publicar reglas que especifican las frecuencias que se pueden utilizar y para qué. El módulo XBee S1 transmite a una frecuencia de 2.4 GHz que es una banda libre, por lo tanto muchos dispositivos lo utilizan.

Los módulos XBee S1 en la Figura 20 se basan en el estándar ZigBee, que trabaja con redes de múltiples nodos inalámbricos.



Figure 20. Dos módulos XBee series 1.

Fuente: Elaboración propia.

Generalmente los módulos XBee se utilizan en una WPAN de punto a punto o la configuración punto a multipunto. La Figura 21 muestra ejemplos de ambos. Punto a punto es útil cuando se desea reemplazar la comunicación serie con cable entre dos unidades remotas. Se utiliza punto a multipunto cuando las frecuencias de las redes de sensores se distribuyen.

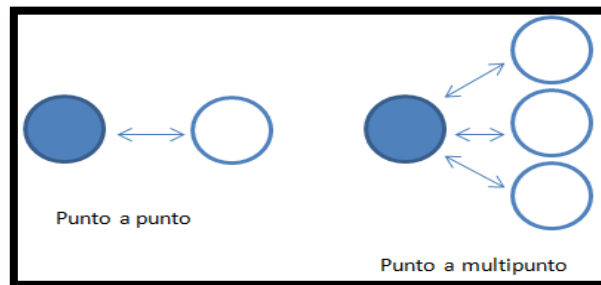


Figura 21: Configuraciones PAN.

Fuente: Elaboración propia

Existe una gran variedad de XBee que se lista a continuación:

- XBee
- XBee-Pro
- XBee Series 1

- XBee Series 2

3.4.1 XBee vs XBee-PRO

Los XBee ofrecen una cobertura de hasta 30 m y consumen muy poca energía, al contrario del XBee-PRO que alcanza distancias mucho mayores (hasta de 90 m) pero realiza así mismo un consumo de energía muy superior; esto debido a que la potencia de transmisión del XBee-PRO es mayor.

3.4.2 Series 1 vs Series 2

Los módulos XBee series 1 generalmente se usan en redes pequeñas, en conexiones remotas entre dos dispositivos; mientras que los series 2 se aplican en redes grandes. Además los series 2 requieren ser configurados antes de ser conectados ya que tienen la capacidad de crear routers en sus nodos, a diferencia de los series 1 que se pueden conectar directamente; por lo que dentro de una misma red no son compatibles. Ver Anexo 2.

3.4.3 Configuración del módulo XBee

Hay muchos parámetros que se deben configurar para comunicar dos módulos. Los módulos deben encontrarse en la misma red (parámetro ID), también en el mismo canal (parámetro CH). Además se debe determinar el módulo destino y el canal por el que se va transmitir (parámetros DH y DL).

- Algunos de los parámetros más usados se describen en la siguiente Tabla III.IV.

Tabla III.IV: Algunos parámetros de XBee.

Comando	Descripción	Valores válidos	Valor por defecto
ID	El ID de red del módulo Xbee	0 - 0xFFFF	3332
CH	El canal del módulo Xbee	0x0B - 0x1A	0x0C
SH y SL	El número de serie del módulo Xbee (SH da los altos 32 bits, SL los bajos 32 bits). Sólo lectura.	0 - 0xFFFFFFFF (tanto para SH y SL)	diferente para cada módulo
MI	La dirección de 16 bits del módulo	0 - 0xFFFF	0
DH y DL	La dirección de destino para la comunicación inalámbrica (DH es el alto de 32 bits, la baja DL 32)	0 - 0xFFFFFFFF (tanto para DH y DL)	0 (tanto para DH y DL)
BD	La velocidad de transmisión utilizada para la comunicación serie con la placa Arduino o computadora	0 (1200 bps) 1 (2400 bps) 2 (4800 bps) 3 (9600) 4 (19200 bps) 5 (38400 bps) 6 (57600 bps) 7 (115200 bps)	3 (9600 baud)

Fuente:<http://arduino.cc/en/pmwiki.php?n=Main/ArduinoXbeeShield>

3.4.4 XBee USB Adapter

Características principales del XBee USB Adapter:

- Funciona con todos los módulos XBee incluyendo la Serie 1, Serie 2 y Serie 2.5; estándar y Pro.
- No necesita alimentación externa
- La compatibilidad con USB 1.0 y 2.0
- El consumo de ancho de banda bajo USB

- 256/128 bytes reciben búfer de recepción / transmisión
- 0.1 "de espaciamiento (protoboard amistoso)
- Regulador de voltaje 250mA 3.3V (se puede utilizar para alimentar dispositivos externos)
- LEDs de comunicación TX y RX
- PWR, RSSI, ON, y LEDs ASOCIADOS
- Accesibilidad a cada pin Xbee
- Funciona con el software X-CTU desde Digi Int.
- Soporte para Windows 98, 98SE, ME, 2000, Server 2003, XP, Vista. Mac OS8 / 9, OS X y Linux 2.4 y superior.

(Para configurar el módulo XBee se conecta a la PC a través del XBee USB Adapter donde se debe montar el módulo XBee S1, como se muestra en la figura 22 y 23. Ver anexo 3).

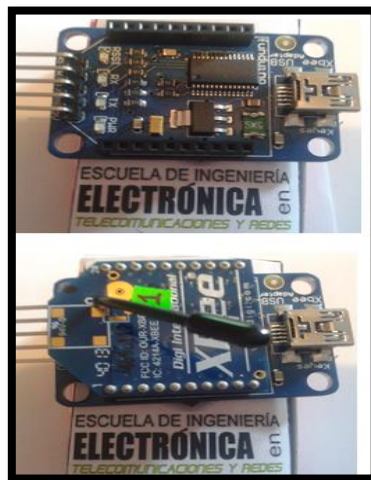


Figura 22: Xbee USB Adapter y Xbee s1 colocado en Xbee USB Adapter.

Fuente: Elaboración propia

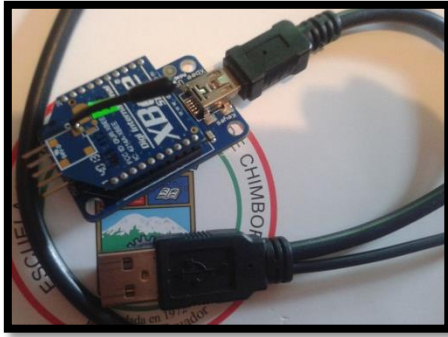


Figura 23: Cable USB tipo B conectado al XBee USB Adapter.

Fuente: Elaboración propia

CAPÍTULO IV

MARCO TEÓRICO

RED INALÁMBRICA DE SENSORES

4.1 INTRODUCCIÓN

En este capítulo vamos a investigar todo acerca de las Redes de Sensores Inalámbricas (WSN) su concepto para conocer mejor su funcionamiento, la arquitectura, la forma de comunicación, la estructura del nodo sensor, sus características más relevantes y algunas de las aplicaciones como es en la industria, la medicina, entre otros. Además determinar los campos de uso de estas redes.

4.2 REDES DE SENSORES INALÁMBRICOS

Una red de sensores inalámbricos consiste típicamente en una gran cantidad de nodos sensores que se desarrolla en una región de interés, proporcionando flexibilidad y facilidad de uso. Los nodos sensores son dispositivos de tamaño pequeño que tienen un microprocesador y unos radios transceptores. Los nodos sensores forman redes Ad-hoc que se pueden ir desplazando en una región determinada, además tienen capacidades de procesamiento y de comunicación de datos, estos datos son transmitidos a otros dispositivos de forma inalámbrica.

Los nodos sensores pueden cumplir diferentes funciones en la red pueden ser receptores, transmisores dependiendo de la configuración de cada dispositivo, estos sensores son alimentados con baterías que tienen una duración de varios años y pueden ser recargables, esto dependiendo de sus características químicas con las que están construidas.

4.2.1 Arquitectura

Toda red inalámbrica de sensores en su arquitectura está compuesta de dispositivos de diminuto tamaño llamados: nodo sensor, puerta de enlace y estación base.

4.2.1.1 *Nodo Sensor*

Comúnmente llamados motas, son dispositivos de diminuto tamaño y de bajo consumo de energía, que son capaces de recibir información del medio en donde se encuentren programados y va a ser procesada y posteriormente transmitida inalámbricamente a otro dispositivo. El nodo sensor está compuesto por las siguientes unidades:

4.2.1.1.1 Unidad de detección

En esta unidad se puede tener uno o más sensores, la función de estos sensores consiste en observar el fenómeno físico y generar señales analógicas de dicho fenómeno y convertir las

señales analógicas en digitales que posteriormente servirán para alimentar a la unidad de procesamiento. Los principales tipos de sensores son:

- **Sensores pasivos omnidireccionales:** Pueden medir magnitudes físicas sin tener que tocar el medio ambiente y algunos de estos sensores son autoalimentados por la energía ambiente, es decir la energía residual que se encuentra en él medio, lo cual se necesita para la amplificación de la señal analógica.
- **Sensores pasivos de haz estrecho:** Son sensores que tienen bien definida la dirección de medición, desde donde deben captar la información.
- **Sensores activos:** Estos sensores monitorean el ambiente activamente.

4.2.1.1.2 Unidad de Procesamiento

Generalmente tiene un microcontrolador o microprocesador que brinda control inteligente para el nodo sensor, que interpreta y procesa los datos para transmitirlos a otra estación y gestiona el almacenamiento de datos en la memoria. La memoria viene integrada en el chip del microcontrolador, las cuales son de dos tipos: la memoria del programa (ROM) para almacenar las rutinas de funcionamiento así como de memoria de datos (RAM) para las variables de programa y para almacenar los datos de los sensores instalados en el nodo.

4.2.1.1.3 Unidad de Comunicación

Esta unidad se compone de un radio de corto alcance para la transmisión y recepción de datos por un canal de radio dentro de su rango de transmisión, las motas usan la banda de frecuencia ISM que son bandas reservadas para propósitos industriales, científicos y médicos.

4.2.1.1.4 Unidad de Potencia

Esta unidad tiene una batería para brindar energía para todos los demás componentes en el sistema y que puedan funcionar correctamente, el consumo de energía es por parte de los sensores, la comunicación y el proceso de la información; estas baterías pueden ser recargables o no recargables dependiendo del material en que estén fabricadas.

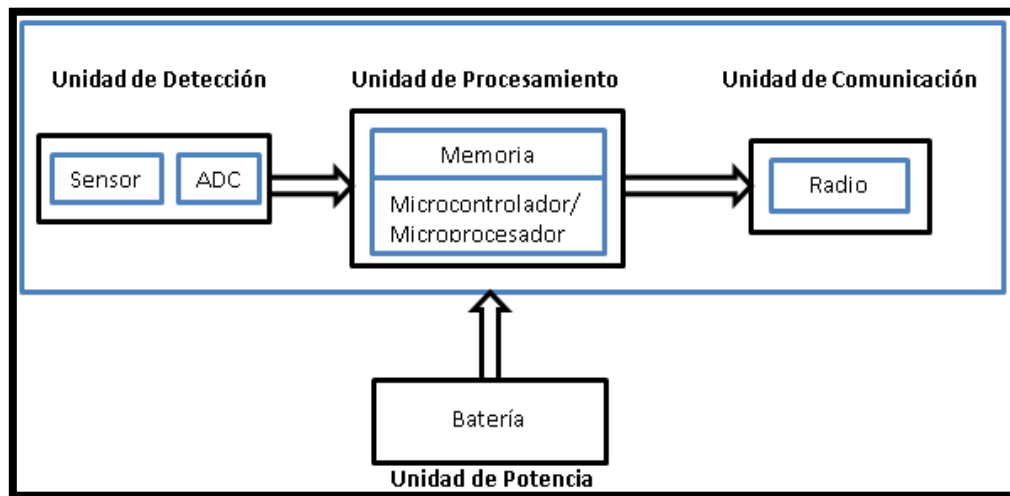


Figura 24: Estructura del nodo sensor.

Fuente: Elaboración propia

4.2.1.2 Puertas de Enlace (Gateway)

Es un dispositivo que nos permite la conexión entre redes de diferente tipo como es las redes de sensores y las redes de datos TCP/IP, los Gateway son nodos que no tienen sensores en su interior.

4.2.1.3 Estación Base

Es una estación donde se almacena toda la información que es recolectada por los sensores, como puede ser un servidor en la cual está contenida una base de datos que va a estar disponible para los usuarios.

4.2.2 Aplicaciones

En la actualidad las Redes de Sensores Inalámbricos realizan varias actividades como medidores de temperatura, humedad, detectores de movimiento, etc., de acuerdo al lugar o área de análisis. Dependiendo de la función que cumplen los sensores se analiza los modos de aplicación comerciales y académicas existentes, desarrollado para redes inalámbricas de sensores. Estas aplicaciones de redes inalámbricas de sensores van en aumento y se pueden clasificar principalmente en las siguientes:

4.2.2.1 Detección de eventos y presentación de informes.

Las WSN se pueden aplicar en actividades relacionadas con el control de un determinado evento por ejemplo se puede usar para vigilancia militar o en sistemas de seguridad si se detecta movimientos, en procesos industriales como comportamientos incorrectos en un proceso, detección de humo para incendios forestales y domésticos.

Esta función se usa para eventos que ocurren esporádicamente. Entonces se espera que la WSN se encuentre pasiva, iniciando una actividad cuando se detecta un evento determinado. La WSN tiene que transmitir inmediatamente un informe de evento a la base. Este informe de eventos tiene que contener datos de ubicación del evento, y una descripción del mismo. Un problema importante para este tipo de redes, a nivel de aplicación, es reducir al mínimo las falsas alarmas. Una posible solución para este problema es usar una mayor cantidad de sensores.

4.2.2.2 Recopilación de datos y presentación de informes periódicos.

Las WSN en esta función se aplican para realizar monitorización de condiciones ambientales como temperatura, humedad e iluminación tanto para entornos naturales (invernadero) o en lugares

artificiales (oficinas e industrias). En estas aplicaciones los sensores deben encontrarse siempre activos para recopilar los datos y transmitir a la base de la red. Además se pueden realizar cualquier tipo de cálculo de los datos si lo que se necesita son estadísticas de estos y visualizar en la base los resultados.

4.2.2.3 Consultas iniciadas de la base

En este caso se sirve del anterior ya que hace uso de los datos captados periódicamente por parte de los sensores para realizar consultas a la base. Estas redes se pueden aplicar en proceso que necesiten un monitoreo constante como lo es los procesos industriales o incluso para medicina en proceso de recuperación de los pacientes. Si existieran errores en los procesos o mediciones altas o bajas según sea el caso se puede enviar un informe por parte del sensor a las bases.

Con esta información la base puede hacer consultas específicas a los sensores que envíen el informe y analizar los datos. Según el análisis que realice la base puede desplegar alarmas o procesos correctivos.

4.2.2.4 Las aplicaciones basadas en seguimientos.

Las WSNs en estas aplicaciones disponen de una gran variedad de usos como en sistemas de seguridad militares o privados para el descubrimiento de intrusos. En estudio de animales cuando hacen el seguimiento de un animal cautivo insertado en el mundo animal o incluso para estudiar su hábitat y comportamientos.

Además se utilizan también en los sistemas de transporte inteligente cuando hay casos de robo del vehículo para la localización o también para trayectorias seguidas por el vehículo. Esta aplicación puede agruparse con las anteriores y enviar alarmas a la base o iniciar procesos.

4.2.3 Características

Después de conocer su amplia gama de aplicaciones en varios campos, vamos en esta sección a conocer sus características más importantes de las WSNs.

4.2.3.1 *Objetivo de colaboración*

El principal objetivo de una WSN es el monitoreo de un determinado evento en donde cada sensor que forma parte de la red envía información a la base y mientras más sensores use dispondrá de una mayor área de análisis. Donde todos los sensores dentro de la red trabajan en conjunto a diferencia de una red de área local donde cada dispositivo trabaja de manera independiente.

4.2.3.2 *Red de escala*

Las redes más sencillas son las pequeñas y constan de un área delimitada de estudio mientras si se requiere realizar un análisis más grande se usa una mayor cantidad de sensores disminuyendo la probabilidad de errores en la red, porque se cuenta con más datos.

En el futuro es posible que gracias a los avances en la tecnología se hagan sensores más baratos y las redes de sensores crezcan mucho más. Además se desea que exista una alta densidad por nodo, es decir, que el nodo tenga más capacidad de comunicación con otros nodos. Creando sistemas más robustos por el incremento de cantidad de rutas alternativas a la base.

4.2.3.3 *Comunicación muchos-a-uno.*

Este es el modo de comunicación de las WSNs porque se dispone de una base a la cual se comunican todos los sensores (muchos-a-uno) y cuando la base pide información a los sensores para actualizar datos (uno-a-muchos). Esta forma de comunicación limita al sensor ya que solo existe una ruta para llegar al destino o base.

4.2.3.4 Capacidades limitadas de los nodos.

La limitación más grande de un nodo sensor es el hardware a diferencia de otros dispositivos como un celular o una PC portátil. Los nodos sensores tienen poca batería (dada por unos pocos cientos de mAh), memorias pequeñas, las bajas tasas de datos (ya que transmiten bit a bit) y un alcance corto (10-100m) por esto las WSN se usan en áreas pequeñas.

4.2.3.5 Escalabilidad para Clustering

Las redes de sensores inalámbricas contienen varios nodos, por esto se usan protocolos para la recopilación de datos y controlar el acceso al canal inalámbrico. Para escalar los protocolos y nodos que aumente en la red se diseñan subredes llamadas clusters, donde se coordina la MAC, el enrutamiento y la agregación de datos por uno de sus nodos.

4.2.3.6 Colocación frente despliegue del nodo.

La ubicación del nodo se determina de acuerdo a la aplicación si se trata de mediciones de temperatura, humedad, en el caso de iluminación o vigilancia se colocan en lugares fijos por lo que la comunicación no es un problema porque van estar en el mismo lugar y se puede determinar el área de cobertura. Cuando se trata de aplicaciones donde se ubica a los nodos de manera aleatoria como en la detección de incendios forestales, vigilancia en las guerras, etc., entonces la comunicación se complica porque se debe tener muy en cuenta el alcance para la distancia máxima de los nodos para garantizar la transmisión.

4.2.3.7 Movilidad del nodo y topología dinámica

Cuando se usa nodos sensores que se desplazan, por ejemplo para monitorear animales en su hábitat, vehículos, o incluso personas; la topología cambia constantemente por lo que se debe ir actualizando la ubicación de los nodos. En estas redes las actualizaciones no son de manera

constante, los nodos hasta se apagan para realizar un ahorro de energía. Todos estos cambios de ubicación, de estado (apagado/encendido) y fallos del nodo provocan cambios en la topología de la red; pero es debido a la movilidad del nodo que la topología se transforma en una topología dinámica.

CAPÍTULO V

SIMULACIÓN DE LA RED DE SENSORES EN NS2

5.1 INTRODUCCIÓN

En este capítulo se realizará la simulación de la Red de Sensores Inalámbricos (WSN) en el simulador NS2 (Network Simulator 2), este es un simulador desarrollado para comunicaciones inalámbricas que soportan multicast y es software libre.

En este simulador se van a crear tres escenarios propuestos, el escenario 1 está formado por 10 nodos y 1 gateway, el escenario 2 está formado por 20 nodos y 1 gateway y el escenario 3 está formado por 24 nodos y 1 gateway que es el número máximo de nodos que soporta Arduino uno. En los 3 escenarios se realiza una comunicación de muchos a uno, porque cada nodo se conecta

con el Gateway para enviar los valores receptados. En cada escenario se va a estudiar las siguientes variables de la red como el Rendimiento (Performance), Throughput, Retardo (Delay) y Consumo de energía (Energy consumption). Estas variables se van a captar por medio de cuatro filtros creados por medio de AWK para cada una de ellas.

Para la comprobación de la hipótesis se va a utilizar el método estadístico de diseño completamente al azar (DCA), mediante este método se determinará qué escenario dispone de un mayor rendimiento y Throughput; así como determinar el que tenga menor consumo de energía y retardo, para encontrar el mejor y luego comparar con las variables del prototipo físico.

5.2 SIMULADOR NS2

NS2 está basado en eventos. Este simulador de redes se puede usar para simular redes cableadas, redes inalámbricas y protocolos (algoritmos de enrutamiento, CBR, TCP, UDP).

Como NS2 es de software libre ha ido evolucionando gracias a los aportes de varias instituciones como la Agencia de Defensa de Proyectos de Investigación Avanzada (DARPA), la Fundación Nacional de Ciencia (NSF) y grupos de investigadores, logrando que esta herramienta se convierta en un software muy conocido.

5.2.1 Arquitectura Básica

Para la programación en NS2 se realiza en un script tcl por medio de secuencia de comandos y para crear una animación gráfica de los resultados, es a través de XGRAPH o NAM (Network AniMator), aunque para análisis es mejor obtener los resultados en texto. Este script se ejecuta por medio del comando "ns" seguido del nombre del script con extensión tcl.

NS2 está compuesto por dos lenguajes de programación: C++ y orientado a objetos TCL Tool Command Language (OTcl). C++ define el mecanismo interno de la simulación y OTcl realiza la configuración de los objetos y también la programación de eventos discretos. Estos dos lenguajes se encuentran unidos por TclCL.

Las variables en los dominios OTcl se refieren como handles, En el dominio OTcl un handle interactúa con el usuario y otros objetos OTcl. Se puede definir sus propios procedimientos y variables, los mismos que en el dominio OTcl se llaman procedimientos de instancia y variables de instancia.

En la Figura 25 se muestra la arquitectura básica del simulador NS.

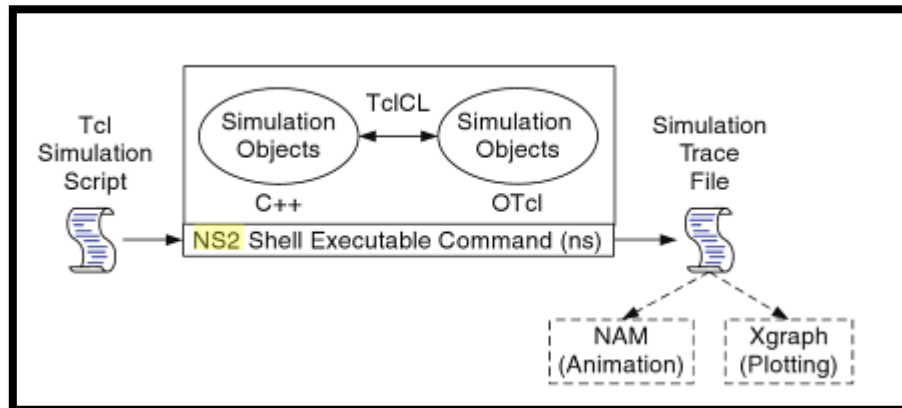


Figura 25. Arquitectura básica de NS.

Fuente: <http://forum.wsnlab.org/threads/how-to-install-cygwin-ns2-in-windows.14/>

5.2.2 Traza Network AniMation (NAM)

Traza NAM está escrita en lenguaje Tcl/Tk, es activado por el comando “\$ns namtrace-all \$file”, donde ns es el handle del simulador y el archivo es un handle asociado (por ejemplo, escenario1.nam) que almacena la información de rastreo de NAM. Después de obtener un archivo

de rastreo NAM, la animación puede ser iniciada directamente en el símbolo del sistema mediante el comando siguiente:

```
>>nam escenario1.nam
```

Muchas de las funciones de visualización están disponibles en NAM. Estas características son, por ejemplo, la animación de los flujos de paquetes de colores, arrastrar y soltar los nodos (posicionamiento), etiquetado de los nodos en un instante determinado, la conformación de los nodos, la coloración de un enlace específico, y el seguimiento de una cola.

5.2.3 Pasos Principales de la Simulación NS2

Como se muestra en la Figura 26, los pasos de simulación en general se pueden adaptar para ajustarse al marco NS2.

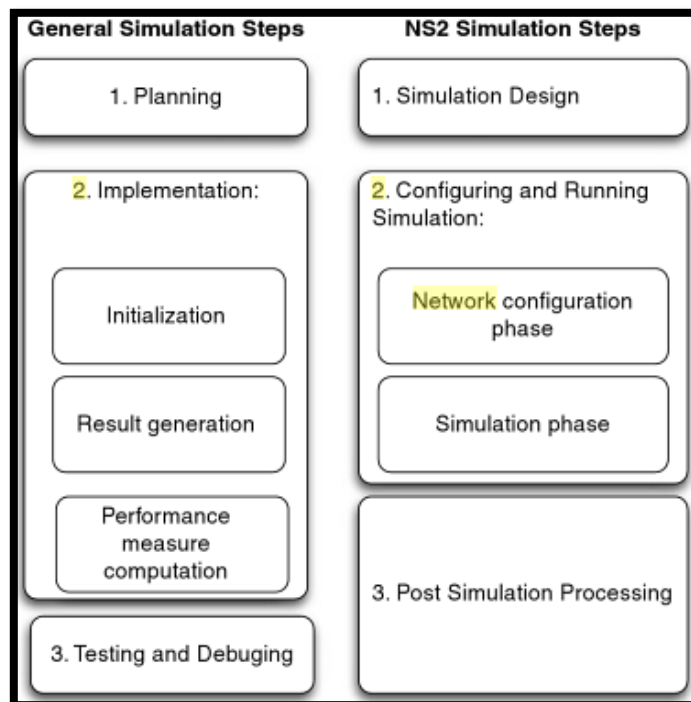


Figura 26: Una comparación de los pasos de simulación generales y pasos de simulación NS2.

Fuente: http://www.ehowenespanol.com/usuarios-zigbee-info_226594/

Los pasos clave de simulación NS2 incluyen lo siguiente:

1. Diseño Simulación

El primer paso en la simulación de una red es diseñar la simulación. En este paso, los usuarios deben determinar los efectos de simulación, configuración de red, los supuestos, las medidas de rendimiento, y el tipo de resultados que se esperan.

2. Configuración y Ejecución de la Simulación.

Este paso implementa el diseño en la primera etapa. Se compone de dos fases:

- Fase de configuración de red: En esta fase, los componentes de red (por ejemplo, nodo, TCP y UDP) se crean y se configuran de acuerdo con el diseño de la simulación. Además, los eventos tales como la transferencia de datos están programadas para comenzar en un momento determinado.
- Fase de Simulación: En esta fase se inicia la simulación que fue configurado en la fase de configuración de red. Esta fase normalmente se ejecuta hasta que el reloj de simulación alcanza un valor umbral especificado en la fase de configuración de red.

En la mayoría de los casos, es conveniente definir un escenario de simulación en un archivo de secuencias de comandos Tcl (por ejemplo, <file>) y alimentar el archivo como un argumento de entrada de una invocación NS2 (por ejemplo, la ejecución de "ns <file>").

3. Procesamiento Postsimulación

Las tareas principales en estos pasos incluyen la verificación de la integridad del programa y evaluar el rendimiento de la red simulada. Mientras que la primera tarea se conoce como la

depuración, el segundo se logra mediante la recopilación correctamente y la compilación de resultados de la simulación.

5.2.4 Ad hoc On Demand Vector (AODV)

Es un protocolo de enrutamiento reactivo, que construye rutas desde el origen al destino, cada vez que sea necesario, permitiendo que los nodos móviles, pasen mensajes a través de sus vecinos a los nodos con los que no se pueden comunicar directamente, a través del descubrimiento de rutas mediante el uso de sus vecinos inmediatos; tratando de encontrar rutas que no contengan bucles y así hallar la ruta más corta posible.

Este protocolo es capaz de manejar cambios en las rutas y puede crear nuevas rutas si hay un error, para esto cada nodo difunde un mensaje HELLO a intervalos regulares para mantener un registro de la lista de vecinos, en el caso de que un nodo envía un mensaje a otro nodo que no es su vecino este difunde un mensaje de solicitud de ruta (RREQ). Debido a que RREQ viaja de un nodo a otro se configura el camino inverso automáticamente. La Figura 23 muestra el mensaje RREQ.

Origen	Destino	Identificación de difusión	Número de saltos	Secuencia de destino	Secuencia de origen	Lifespan
--------	---------	----------------------------	------------------	----------------------	---------------------	----------

Figura 27. Mensaje RRQ
Fuente: Elaboración propia.

5.2.5 Awk

AWK está diseñado para el procesamiento de datos de texto, como puede ser en archivo o flujos de datos. Mediante el uso de estos filtros como comúnmente se les conoce se puede calcular el rendimiento, la energía consumida por la red, el throughtput, entre otros parámetros. Los valores así calculados se utilizan para visualizar los resultados de forma gráfica utilizando Gnuplot, un

programa de datos de trazado interactivo destinado principalmente para representar los datos científicos.

Es un excelente filtro y escritor de informe, también tiene funciones de manipulación de cadena, por lo que puede buscar cadenas particulares y modificar la salida.

5.3 ESTRUCTURA DE LA SIMULACIÓN

En este estudio de investigación se ha creado un archivo por cada escenario propuesto en este caso se tiene 3 archivos cada uno con sus propias características, en cada archivo se va variando el tiempo de ejecución del programa para 100s y para 3600s. De cada escenario propuesto se ha generado 200 muestras pero de las cuales solo se han tomado 30 muestras aleatorias basándonos en el método estadístico completamente al azar, para con estos datos generar un promedio y ver cual escenario es más eficiente.

En la Figura 28 se indica la estructura de la simulación que se ha planteado:

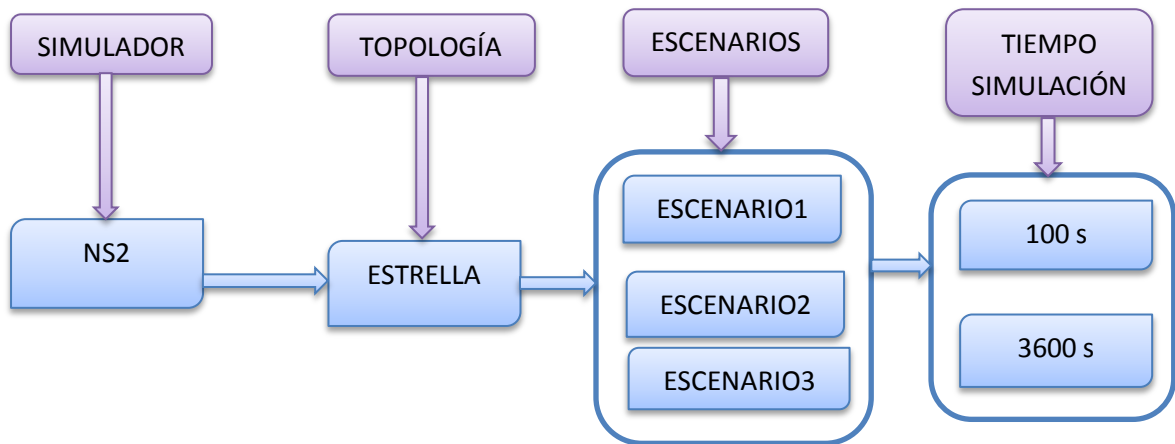


Figura 28: Estructura de la Simulación.

Fuente: Elaboración propia

5.3.1 Parámetros de la Simulación

En este trabajo de investigación en cada escenario planteado se han utilizado parámetros variables y constantes, dentro de los parámetros variables se tiene el tiempo de duración de la simulación que ha utilizado un tiempo de 100 segundos y de 3600 segundos en cada escenario, el número de nodos teniendo un escenario de 10 nodos, 20 nodos y 24 nodos, Retardo, Throughput, Rendimiento y la Energía consumida por la red. Dentro de los parámetros constantes tenemos el protocolo de enrutamiento, área de simulación y la potencia de la antena.

En la Figura 29 se puede observar los parámetros de las constantes y las variables utilizadas en cada escenario.

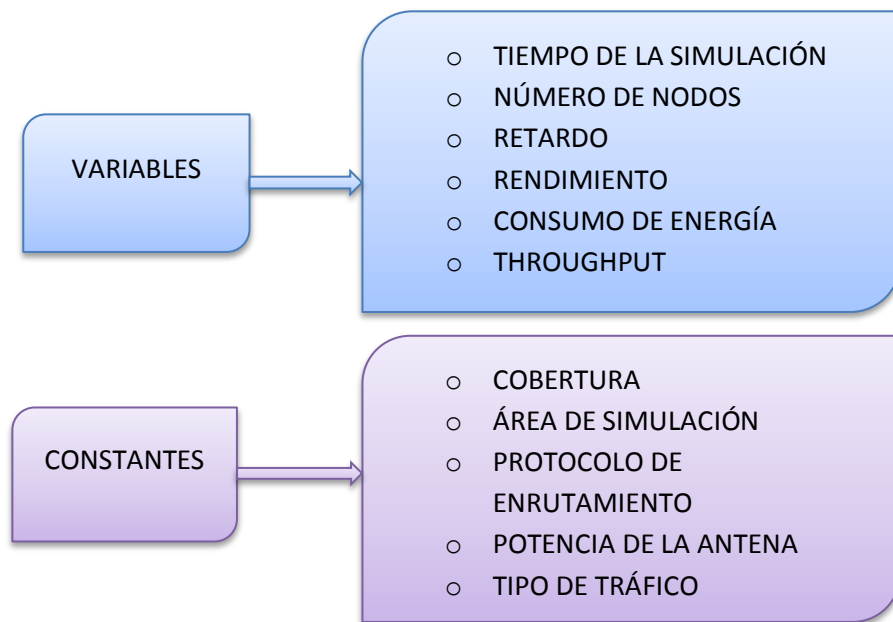


Figura 29: Parámetros de la simulación.

Fuente: Elaboración propia.

Para este estudio se han tomado en cuenta los parámetros más relevantes de la red como son los siguientes:

- **Cobertura**

Para que todos los nodos alcancen la cobertura deseada, se tomó en cuenta la cobertura que alcanza los módulos XBee S1 y en base a esto se diseñó la WSN en los escenarios.

- **Área de simulación**

El área de simulación que se ha designado para los escenarios es de cuarenta metros por cuarenta metros; tomando en cuenta la distancia de cobertura de los módulos XBee S1 y también el área de trabajo que se tiene en el Centro de Atención Integral en Salud de la ESPOCH.

- **Protocolo de enrutamiento**

El protocolo de enrutamiento que se ha utilizado es el AODV (Ad hoc On Demand Distance Vector), porque es un protocolo que utiliza pocos recursos y es más usado para estudios de tiempo real.

- **Tipo de Tráfico**

El tipo de tráfico que se utiliza para realizar la simulación de la red es el tráfico CBR que envía una tasa de bits de manera constante, por la razón de que es el tráfico que más se parece al que se va a tener en el prototipo de la Red que se va a implementar, como es tráfico UDP que no necesita confirmación de entrega.

- **Tiempo de duración de la Simulación**

Este parámetro en la red es variable, ya que se ha realizado simulaciones con los tres escenarios con diferente tiempo como es 100 segundos y 3600 segundos permitiendo observar la variación de la red desde el establecimiento de la conexión con todos los nodos, así como también en el transcurso de la transmisión de la información. Se ha variado el tiempo de simulación para ver que tanto evoluciona la red en el aumento del tiempo de la simulación y como es su comportamiento.

- **Número de Nodos**

El número de nodos ha variado en cada escenario como ya se ha explicado, se tiene tres escenarios cada uno con 10 nodos, 20 nodos y 24 nodos, se ha realizado esta variación para ver cómo van cambiando los resultados en la red.

- **Consumo de energía**

Este parámetro va a ir variando dependiendo del número de nodos en la red y la cobertura de cada uno de ellos.

5.3.2 Variables de estudio.

Para realizar el estudio se ha tomado en cuenta las variables más significativas en los escenarios que se ha planteado en la WSN como es Throughput, Retardo, Consumo de Energía y Rendimiento.

En la Figura 30 se muestra los parámetros que se van analizar para realizar el estudio.

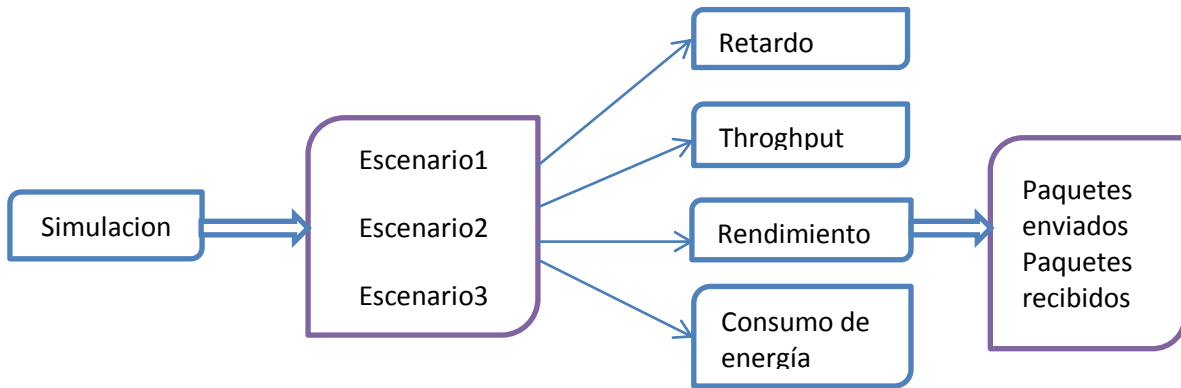


Figura 30: Parámetros de estudio.

Fuente: Elaboración propia.

5.3.2.1 Retardo

Para obtener información sobre el retardo que sufren los paquetes en la red se creó un filtro con el cual se va analizar en cada escenario cuanto tardan los paquetes en llegar a su destino. El retardo viene dado en milisegundos (ms) por el tiempo que los paquetes se demoran en llegar al destino y el número de conexiones de la red.

5.3.2.2 Throughput

El throughput es la cantidad máxima de información que puede ser transmitida en la red por unidad de tiempo. Estos valores se obtuvieron de la simulación utilizando un filtro awk para el análisis de estos valores.

5.3.2.3 Rendimiento

Para el cálculo del rendimiento se divide el número de paquetes recibidos para el número de paquetes enviados durante toda la simulación.

5.3.2.4 Consumo de Energía

Para obtener los valores del consumo de energía en la WSN se utilizó un filtro awk que permite medir el consumo que tiene la red en los diferentes escenarios planteados. Para medir esto se toma en cuenta la cobertura de la red de cada nodo sensor y los paquetes que son transmitidos en la WSN.

5.4 CREACIÓN DE LOS ESCENARIOS

Para la creación de los escenarios propuestos se consideró las variables que fueron mencionadas anteriormente y en cada escenario se va a ir variando los parámetros para ir obteniendo los resultados en cada uno de ellos.

5.4.1 Configuración del simulador NS2.

- Inicializando las variables globales y generación de los archivos traza y NAM.

Para comenzar a configurar los requerimientos de la red planteada se empieza creando una nueva simulación que se puede realizar de dos maneras la primera es entrar en el terminal de Ubuntu y poner \$ns al escribir esto se está creando un script en modo interactivo y también se puede crear un script en modo batch que es lo que se hizo y a continuación se detalla entre las partes que tiene la configuración de la red, para definir variables se debe poner la palabra set y seguido el nombre de la variable de igual manera si se desea definir un objeto se usa el método new.

A continuación se va a crear la parte principal del programa que es crear una nueva instancia de simulación, e indicar al simulador donde va a guardar las trazas de los paquetes que se generan de la simulación y que van a contener los resultados que luego van hacer analizados. De la misma manera se indica que se guarde el archivo .nam que es el que genera la imagen del escenario.

```
set ns_ [new Simulator] ; # crea la nueva instancia de simulación
set tracefd [open ./escenario1.tr w] ; # guarda la traza de la simulación
set namtrace [open escenario1-out.nam w] ; # visualización del escenario
$ns_ trace-all $tracefd
```

Seguidamente se va a crear un objeto de topología que se encarga de realizar un seguimiento de los movimientos de los nodos móviles dentro de la frontera topológica, es decir es el área donde van a trabajar los nodos inalámbricos:

```
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
```

Luego se crea el objeto God que va a simular todos los eventos de los nodos y es el encargado de los enlaces entre los nodos:

```
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]
```

Se procede a la configuración de un punto muy importante en el programa, en este punto se utiliza la traza nueva para la comunicación inalámbrica y de esto depende la utilización de los filtros para el análisis de los resultados:

```
$ns_ set WirelessNewTrace_ON ; # nuevo formato de traza
```

- **Finalización de la simulación**

Para terminar la simulación se procede a configurar la siguiente línea de comando en el cual se va a tener un tiempo de stop que en las líneas de comando anterior se puso, en el cual se va a finalizar los procesos que se están ejecutando como es la generación de las trazas de datos, al realizar este proceso se mostrara un mensaje que dice “saliendo de NS” y para finalmente indicar que se “termino la simulación”:

```
$ns_ at $stopTime "stop"  
$ns_ at $stopTime "puts \"NS EXITING...\n\""  
$ns_ at $stopTime "$ns_ halt"  
puts "\nStarting Simulation..."  
$ns_ run
```

5.4.2 Configuración del escenario

En este punto se define las características más relevantes que se tiene que tomar en cuenta en el momento de la configuración del programa para la red inalámbrica, seleccionando los que son más eficientes para la red planteada como es: el tipo de canal, el modelo de propagación, tipo de estándar a usar, tipo de cola de la interfaz, tipo de capa de enlace, modelo de la antena, paquetes máximos en ifq, número de nodos móviles, protocolo de ruteo, tipo de MAC, las coordenadas del escenario tanto en “x” como en “y” aquí se pone el tamaño del escenario que se plantea, nombre del archivo nam que se crea, en el cual se van a guardar las trazas de la simulación y el tipo de tráfico que va a generar, :

```
# =====  
# Define options  
# =====  
set val(chan) Channel/WirelessChannel ; # tipo de canal  
set val(prop) Propagation/TwoRayGround ; # modelo de radio-propagación  
set val(netif) Phy/WirelessPhy/802_15_4 ;# interfaz inalámbrica de comunicación  
set val(mac) Mac/802_15_4 ; # tipo de MAC  
set val(ifq) Queue/DropTail/PriQueue ; # tipo de cola de la interfaz  
set val(ll) LL ; # tipo de capa de enlace  
set val(ant) Antenna/OmniAntenna ; # modelo de antena  
set val(ifqlen) 150 ; # paquetes maximos en ifq  
set val(nn) 11 ; # número de nodos moviles  
set val(rp) AODV ; # protocol de ruteo ad-hoc  
set val(x) 40 ; #coordenada en x  
set val(y) 40 ; #coordenada en y  
set val(nam) escenario1.nam ; # nombre archivo .nam  
set val(traffic) cbr ; # tipo de tráfico
```

5.4.3 Configuración y creación de los nodos inalámbricos

El proceso de creación de los nodos móviles se explica en los siguientes pasos a continuación detallado:

- Primeramente se procede a configurar los nodos con la siguiente línea de comando en donde se pone obligatoriamente `adhocRouting` siendo este el encargado de controlar como los nodos establecen el camino para enrutar los paquetes:

```
$ns_ node-config -adhocRouting $val(rp) \
```

- A continuación se realiza la creación de los nodos móviles donde se activa los servicios que hacen que la red sea disciplinada en la entrega de paquetes, esto es porque da prioridad a la cola de paquetes de enrutamiento:

```
-llType $val(ll) \  
  -macType $val(mac) \  
  -ifqType $val(ifq) \  
  -ifqLen $val(ifqlen) \  
  -antType $val(ant) \  
  -propType $val(prop) \  
  -phyType $val(netif) \  
  -topoInstance $topo \
```

Se añade las siguientes líneas de comando para activar los agentes necesarios para la simulación como es: la capa de acceso al medio, protocolo de ruteo y las trazas para los paquetes, para este estudio no se activó la línea de comando de nodos móviles por que el alcance de la cobertura de los XBee es suficiente para el centro de salud donde van a estar ubicados los sensores y así haya movimiento de los nodos si va a ver transmisión entre ellos.

```
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \
```

```
-movementTrace OFF \
```

Para determinar la energía consumida por la WSN se configura en las siguientes líneas de comando en donde se va a poner las características de la energía inicial de los nodos:

```
-energyModel "EnergyModel" \  
-initialEnergy 9720 \
```

Para realizar el análisis del consumo de energía se lo va a realizar en las unidades de Amperio Hora [Ah] debido a que se va a usar unas baterías para dar alimentación al prototipo WSN en este caso se van usar baterías de 9V a 300mA, este valor de Ah lo transformamos a Julios que es la unidad en que trabaja el simulador NS2.

$$J = A * s * V$$

$$J = 0.3 * 3600 * 9$$

$$J = 9720J$$

Para calcular la potencia de transmisión de los nodos se tome en cuenta la potencia con la que trabaja el Xbee S1 que es este caso es de 1mW que se va a usar para este estudio y va hacer configurada en la siguiente línea de comandos, también se selecciona el canal que va a usar para la transmisión.

```
-rxPower 0.5319 \  
-txPower 0.2 \  
-channel $chan_1_
```

Se aplica la siguiente ecuación para saber cuál es la potencia de recepción de una antena que va a tener el nodo coordinador y luego poner este valor en la línea de comando anteriormente escrita:

$$Prx = Ptx * Gtx * Grx * Lo$$

$$Prx = 1 * -2dBi * -2dBi * 129.99$$

$$Prx = 0.5319$$

Donde cada parámetro significa lo siguiente:

- Prx: Es la potencia de recepción de la antena receptora
- Ptx: Potencia de transmisión de la antena transmisora.
- Gtx: Ganancia de la antena transmisora.
- Grx: Ganancia de la antena receptora.
- Lo: Perdidas del espacio libre

La fórmula para calcular la pérdida del espacio libre es la siguiente en donde debe estar la frecuencia en MHz y la distancia en Km:

$$Lo(db) = 32,45 + 20 \log f + 20 \log d$$

$$Lo(db) = 32,45 + 20 \log(2400) + 20 \log(30)$$

$$Lo(db) = 129.29$$

Donde cada variable significa lo siguiente:

- Lo: Perdida del espacio libre en db
- F: Frecuencia en la que va a transmitir en MHz
- D: Distancia de las antenas en Km

A continuación se indica el código que fue configurado en el simulador para la configuración de las antenas en donde se tiene una distancia de 15m y una perdida por trayectoria de 1db:

```
Phy/WirelessPhy set CStresh_ $dist(15m)
Phy/WirelessPhy set RXThresh_ $dist(15m)
Phy/WirelessPhy set CPTresh_ 10
Phy/WirelessPhy set L_ 1.0 ;# Perdida por Trayectoria = 1.0
```

Después de haber definido los parámetros anteriores se procede a la creación de los nodos inalámbricos con las siguientes líneas de código en donde se va a deshabilitar el movimiento aleatorio y se va creando uno a uno los nodos dependiendo del número de nodos que se tenga en los escenarios planteados.

```
for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0           ;# disable random motion
}
$node_(0) set X_ 15
$node_(0) set Y_ 15
$node_(0) set Z_ 0
$node_(1) set X_ 15
$node_(1) set Y_ 4
$node_(1) set Z_ 0
```

5.4.4 Configuración de los agentes de tráfico

Para que los nodos transmitan y reciban tráfico necesitan ser configurados los agentes que van a realizar esta función, en el cual se va asociar cada uno de los nodos con el nodo coordinador que es el que va a recibir el tráfico CBR/UDP que la red está generando.

Para tráfico CBR/UDP el tamaño máximo de trama que soporta es de 100 Bits pasado de este límite de paquetes el simulador ya no transmite, con intervalos de tiempos, para este estudio se ha ido variando el tamaño de la trama hasta 100 Bits.

```
# Configuración del flujo de tráfico entre nodos
```



```
proc cbrtraffic { src dst interval starttime } {
  global ns_ node_

  set udp_($src) [new Agent/UDP]
  eval $ns_ attach-agent \ $node_($src) \ $udp_($src)

  set null_($dst) [new Agent/Null]
  eval $ns_ attach-agent \ $node_($dst) \ $null_($dst)

  set cbr_($src) [new Application/Traffic/CBR]
  eval \ $cbr_($src) set packetSize_ 32
  eval \ $cbr_($src) set interval_ $interval
  eval \ $cbr_($src) set random_ 0
  eval \ $cbr_($src) set maxpkts_ 10000
  eval \ $cbr_($src) attach-agent \ $udp_($src)
  eval $ns_ connect \ $udp_($src) \ $null_($dst)
  $ns_ at $starttime "$cbr_($src) start"
}
```

También se configuró desde que instante se va a empezar con la transmisión del tráfico hasta que tiempo debe parar en las siguientes líneas de comando:

En estas líneas de código se está creando las variables para llamarlas y empezar la transmisión desde ese instante:

```
getCmdArgv $argc $argv

set appTime1      7.0 ;# in seconds
set appTime2      7.1 ;# in seconds
set appTime3      7.2 ;# in seconds
set appTime4      7.3 ;# in seconds
```

En las siguientes líneas de código se está llamando a la variable que se creó en las líneas anteriores para empezar con la transmisión.

```
puts [format "Acknowledgement for data: %s" [Mac/802_15_4 wpanCmd ack4data]]
$ns_ at $appTime1 "Mac/802_15_4 wpanNam PlaybackRate 0.5ms"
$ns_ at [expr $appTime1 + 0.5] "Mac/802_15_4 wpanNam PlaybackRate 1.5ms"
$val(traffic)traffic 1 0 0.6 $appTime1
$val(traffic)traffic 2 0 0.6 $appTime2
$val(traffic)traffic 3 0 0.6 $appTime3
```

Para crear etiquetas a los nodos se configura las siguientes líneas de código en donde se escribe el nombre que va a tener cada uno de los nodos y desde que instante se va a ver la etiqueta:

```
$ns_ at 0.0 "$node_(0) NodeLabel PAN Coord"
$ns_ at 0.0 "$node_(0) sscs startPANCoord"           ;# startPANCoord
<txBeacon=1> <BO=3> <SO=3>
$ns_ at 0.5 "$node_(1) sscs startDevice 1 0"         ;# startDevice <isFFD=1> <assoPermit=1>
<txBeacon=0> <BO=3> <SO=3>
$ns_ at 1.5 "$node_(2) sscs startDevice 1 0"
```

Las líneas de código siguientes indican desde que nodo va a empezar a generar el flujo del tráfico:

```
$ns_ at $appTime1 "$ns_ trace-annotate \"(at $appTime1) $val(traffic) tráfico desde nodo 1 al
nodo 0\""
$ns_ at $appTime2 "$ns_ trace-annotate \"(at $appTime2) $val(traffic) tráfico desde nodo 2 al
nodo 0\""
$ns_ at $appTime3 "$ns_ trace-annotate \"(at $appTime3) $val(traffic) tráfico desde nodo 3 al
nodo 0\""
```

5.4.5 Generación de los archivos de traza

Los archivos de traza son los resultados generados al final de la simulación estos archivos nos ayudan a obtener información de lo que sucedió durante la ejecución de la simulación, para luego ser analizados de manera gráfica o estadística. Para realizar el análisis de estos archivos se debe crear unos filtros en formato .awk en donde se debe configurar lo que se desea analizar de las trazas obtenidas.

- Estructura del nuevo formato de trazas inalámbricas

En la siguiente Figura 31 y 32. Se indica detalladamente la estructura de las trazas que se obtienen de una red inalámbrica que son generadas en texto plano y que tienen los siguientes elementos:

```
escenario1.tr x
s -t 0.000640000 -Hs 0 -Hd -1 -Nl 0 -Nx 15.00 -Ny 15.00 -Nz 0.00 -Ne
5759.999830 -Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
N -t 0.000640 -n 1 -e 5759.999762
N -t 0.000640 -n 2 -e 5759.999762
N -t 0.000640 -n 9 -e 5759.999762
N -t 0.000640 -n 8 -e 5759.999762
N -t 0.000640 -n 7 -e 5759.999762
N -t 0.000640 -n 10 -e 5759.999762]
N -t 0.000640 -n 6 -e 5759.999762
N -t 0.000640 -n 3 -e 5759.999762
N -t 0.000640 -n 4 -e 5759.999762
N -t 0.000640 -n 5 -e 5759.999762
r -t 0.001088037 -Hs 1 -Hd -1 -Nl 1 -Nx 15.00 -Ny 4.00 -Nz 0.00 -Ne 5759.999762
-Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
```

Figura 31: Estructura de la traza inalámbrica

Fuente: Simulador NS2, escenario1.

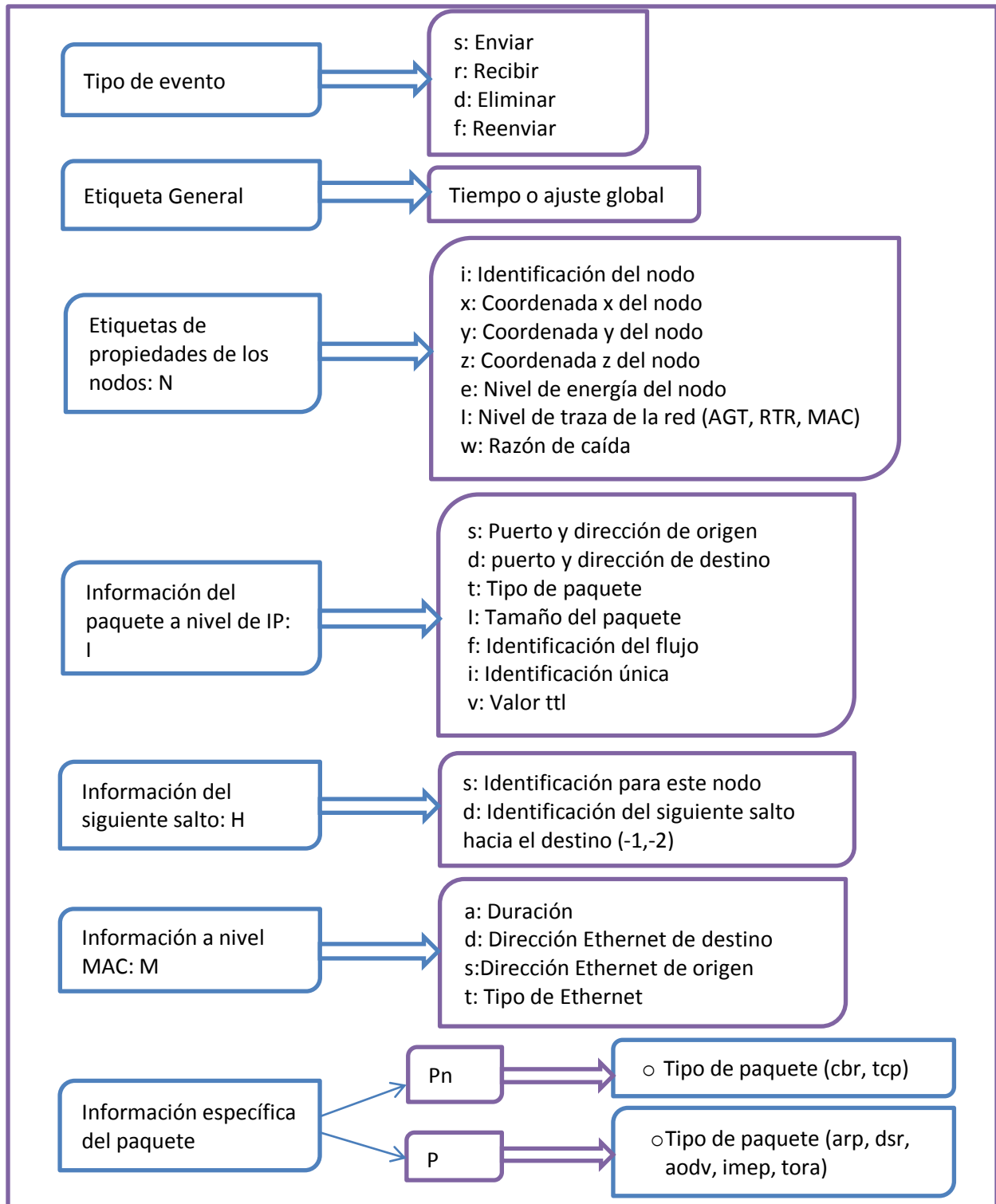


Figura 32: Estructura del nuevo formato de trazas inalámbricas.

Fuente: Elaboración propia

- Creación de los filtros .awk

AWK en un lenguaje de programación que es utilizado para analizar archivos de texto plano que son generados por el simulador NS2. Para este estudio se crearon cinco filtros .awk, el primer filtro es para analizar los paquetes enviados, el segundo filtro es para analizar los paquetes reenviados, el tercer filtro es para analizar los paquetes enrutados de la red, el cuarto filtro es para la energía consumida por la red y el ultimo filtro es para tener el valor total de las variables que se están analizando en este estudio como es el retardo, rendimiento, throughput y la energía consumida por toda la red.

- Ejecución de los filtros .awk

Para la ejecución de los filtros .awk se debe ubicar desde la dirección en donde se encuentre el filtro a continuación se muestra la Figura 33 de la ejecución de los filtros. Al finalizar la generación de todos los filtros se tiene el promedio total de cada parámetro a analizar:

```
mary@mary:~/Escritorio/tesis/escenario1/44$ awk -f 1filtrorecenv.awk escenario1.tr
mary@mary:~/Escritorio/tesis/escenario1/44$ awk -f 2filtrocai_reen.awk escenario1.tr
mary@mary:~/Escritorio/tesis/escenario1/44$ awk -f 3filtropqtenru.awk escenario1.tr
mary@mary:~/Escritorio/tesis/escenario1/44$ awk -f 4filtroenergy.awk escenario1.tr
mary@mary:~/Escritorio/tesis/escenario1/44$ awk -f totalestadist.awk escenario1.tr
```

Figura 33: Ejecución de filtros .awk

Fuente: Simulador NS2, escenario1.

5.5 ESCENARIOS PROPUESTOS

Los escenarios se determinaron según la capacidad del Arduino, así pues se fue variando el número de nodos hasta alcanzar la capacidad máxima del Arduino que son 24 nodos.

5.5.1 Escenario 1

En el escenario 1 se crearon 10 nodos, siendo el escenario más pequeño de todos; donde para el análisis del tráfico se cambió el tiempo de duración primero con un tiempo de simulación de 100s y luego con un tiempo de 3600s para determinar los factores de análisis anteriormente explicados como son el rendimiento, el retardo, Throughput y el consumo de energía de la red. Para las diferentes mediciones realizadas también se cambió el tamaño del paquete.

Rendimiento: Con el análisis del rendimiento podemos saber la cantidad real de paquetes que se reciben. El rendimiento viene definido por la siguiente formula:

$$\text{Rendimiento} = \frac{\sum \text{Paquetes recibidos}}{\sum \text{Paquetes enviados}}$$

Retardo: Es el tiempo que tardan los paquetes en llegar a su destino. Se calcula con la siguiente formula:

$$\text{Retardo} = \frac{\sum (\text{Tiempo de llegada del paquete} - \text{Tiempo de envío})}{\sum \text{número de conexiones}}$$

Throughput: Es la cantidad real de información que se transmite en un periodo de tiempo. Se determina con la siguiente fórmula:

$$\text{Throughput} = \frac{\text{número de bits}}{\text{tiempo de simulación}}$$

Consumo de energía: Se calcula realizando una resta en cada periodo de la red.

Cuando ejecutamos el archivo.tcl anteponiendo ns podemos visualizar la red como se muestra en la Figura 33.

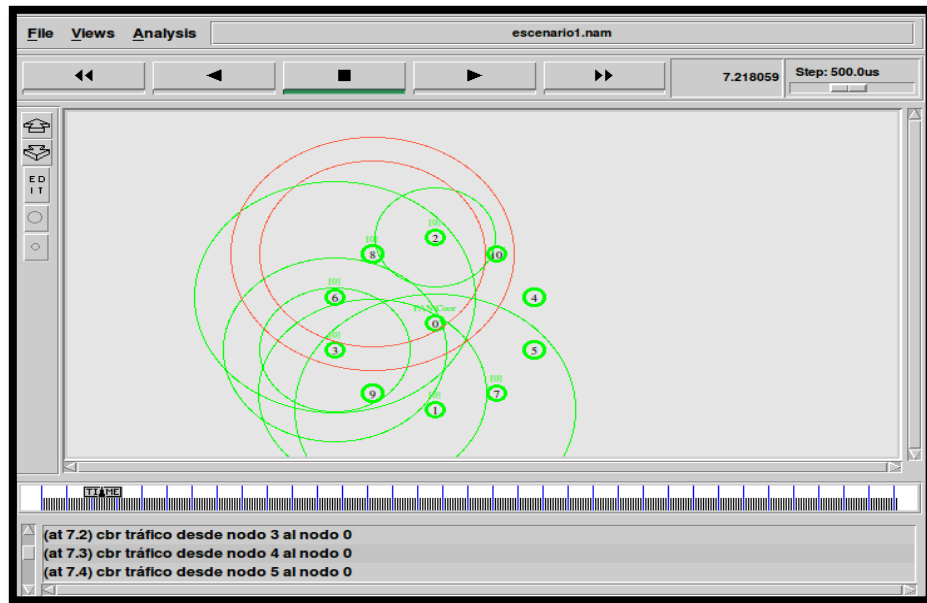


Figura 33: Escenario 1

Fuente: Elaboración propia.

5.5.2 Escenario 2

Para el segundo escenario se incrementó el número de nodos en el doble siendo 20 nodos, pero al igual que en el primero se cambió el tiempo de simulación de 100s a 3600s para el análisis de las variables del rendimiento, retardo, Throughput y el consumo de energía que se obtienen ejecutando los filtros desarrollados en AWK. Manteniendo la distancia de cobertura pero variando además el tamaño de los paquetes para cada medición. Se obtiene la figura 34.

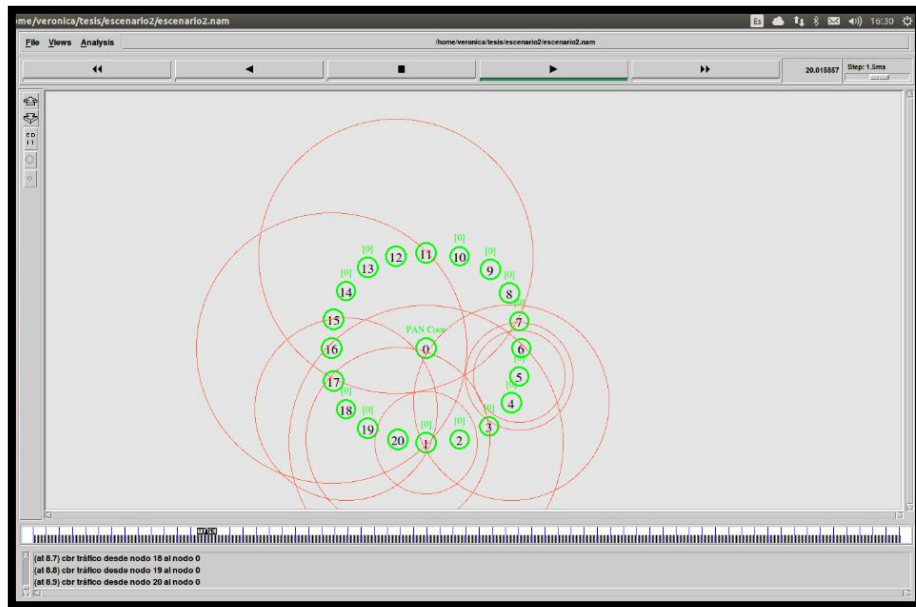


Figura 34: Escenario 2

Fuente: Elaboración propia

5.5.3 Escenario 3

En el último escenario se procedió a crear 24 nodos que es la máxima capacidad del arduino que se usó, cambiando las mismas variables que en los escenarios anteriores las cuales son el tiempo de duración de la simulación, así también como el tamaño del paquete determinó en los agentes de tráfico. Vamos a obtener el rendimiento, retardo, throughput y consumo de energía para comparar con los resultados de los dos escenarios anteriores. La figura 35 muestra en escenario 3.

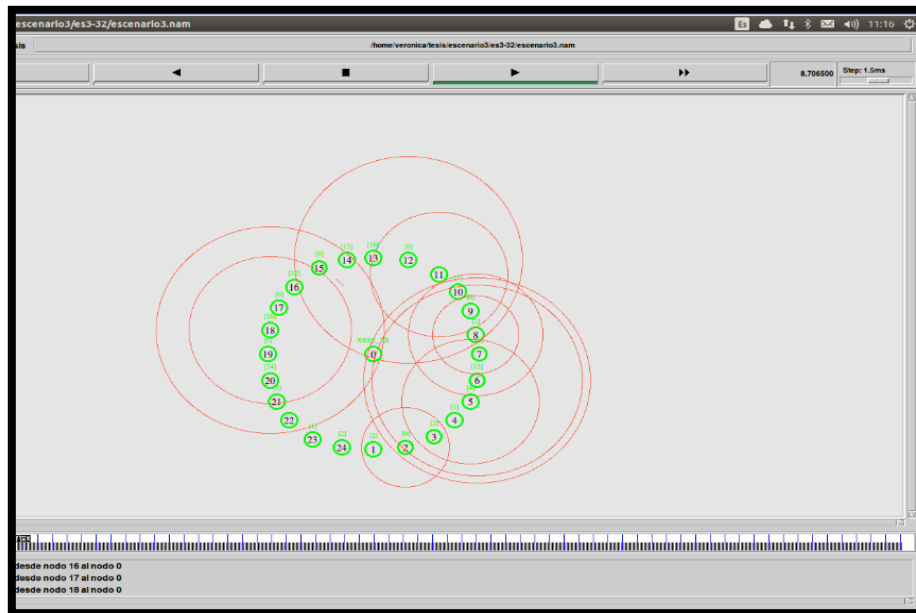


Figura 35: Escenario 3

Fuente: Elaboración propia.

CAPÍTULO VI

IMPLEMENTACIÓN DEL PROTOTIPO DE WSN

6.1 INTRODUCCIÓN

En este capítulo se va a construir el prototipo de la Red de Sensores Inalámbrica para el Monitoreo de los Pacientes en el Centro de Salud Integral de la ESPOCH.

Este prototipo consta de un arduino uno, un transmisor en donde se va programar a los sensores de temperatura corporal, de flujo de aire, además del sensor de Pulso y oxigenación de la sangre (SpO2), los mismos que irán conectados en el e-Health Sensor Shield que va colocada sobre el arduino uno.

Pero además para la transmisión inalámbrica de los datos tomados por medio de los sensores se utiliza dos XBee s1, los cuales cumple la función de un router y permiten la transmisión inalámbrica de los datos. Para conectar los XBee S1 al arduino se utiliza el Arduino XBee shield que se coloca sobre el e-Health Sensor Shield y por último se coloca el XBee S1. Como el arduino transmisor es móvil para darle energía será mediante una batería de 9V a 300mA.

Para el Arduino Uno receptor se necesita el Arduino Xbee shield donde se coloca al XBee S1 que va sobre el mismo y además un cable serial USB tipo B para conectar a la PC donde para la visualización de los datos se utiliza una interfaz gráfica en Delphi XE7, la misma que irá conectada una base de datos para el almacenamiento de estos.

6.2 DISEÑO DEL PROTOTIPO WSN

La red consta de cuatro sensores conectados a la placa e-Health Sensor Shield la que transmitirá al nodo receptor que es el XBee S1 que está conectado a la PC para la visualización de los datos. Lo mismo que se puede observar en la Figura 36.

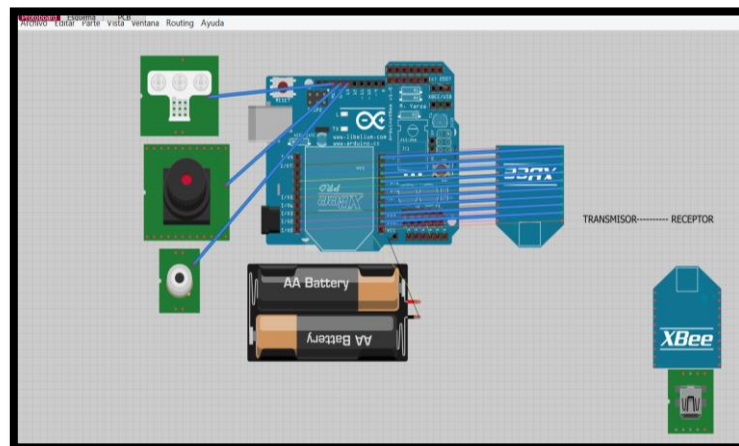


Figura 36: Prototipo de la WSN

Fuente: Elaboración propia.

6.3 IMPLEMENTACIÓN DEL PROTOTIPO WSN

Para la implementación se va a investigar el funcionamiento de los sensores que se van a utilizar, para después configurar en el Arduino uno y posteriormente configurar los XBee S1 para la transmisión de los datos; se debe investigar sobre el programa que da más facilidad en el momento de la creación de la interfaz y que permita establecer conexión con la base de datos y con el puerto serial.

6.3.1 Sensor de Flujo de Aire

El sensor de flujo de aire mide la tasa de respiración de las personas. Este sensor consta de un hilo que se coloca detrás de los oídos y dos puntas que se colocan en la nariz, es a través de estas que se mide la respiración. La forma de conexión es conectar el cable rojo al terminal positivo y el cable negro al terminal negativo como se muestra en la Figura 37.

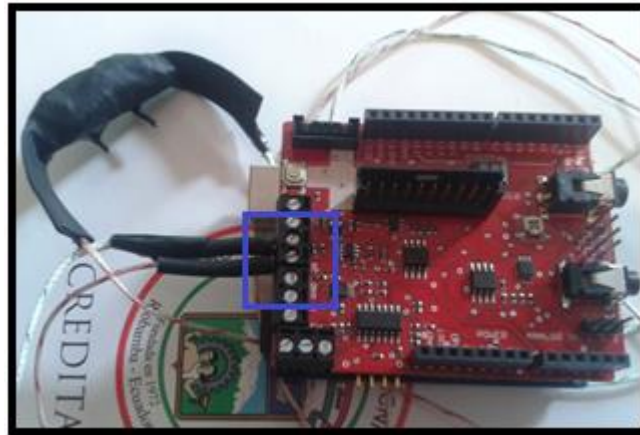


Figura 37: Conexión del sensor flujo de aire

Fuente: Elaboración propia.

6.3.2 Sensor de Temperatura Corporal

La temperatura corporal depende de muchos factores como el momento del día o el lugar del cuerpo en que se mida, esta medida nos puede ayudar a determinar los estados de temperatura corporal como se muestra en la tabla VI.V.

Tabla V.V Temperaturas Corporales

La hipotermia	Menor a 35 ° C
Normal	36,5-37,5 ° C
Fiebre o hipertermia	Mayor a 37,5-38,3 ° C
Hiperpirexia	Mayor a 40,0-41,5 ° C

Fuente: Lcda. Lupe Zabala.

El sensor de temperatura corporal para medir se conecta de una manera muy fácil tan solo con conectarle como se muestra en la figura 38. Y se coloca en el dedo de la mano o pie.



Figura 38: Conexión del sensor de temperatura corporal

Fuente: Elaboración propia.

6.3.3 Sensor de Pulso y Oxígeno en la Sangre (SpO₂)

Este sensor mide la cantidad de oxígeno en la sangre a través de la detección de la hemoglobina y desoxihemoglobina. Hemoglobina desoxigenada (Hb) tiene una absorción mayor a 660 nm y la hemoglobina oxigenada (HbO₂) tiene una mayor absorción en 940 nm. Un fotodetector percibe la luz no absorbida de los LEDs para calcular la saturación arterial de oxígeno. Este sensor es muy útil para cualquier paciente. En la tabla VI.VI se muestra algunos porcentajes. Ver Anexo 6

Tabla VI.VI. Porcentaje de la oxigenación en la sangre

Normal	95 a 99 por ciento
Hipóxico	88 al 94 por ciento
Envenenamiento por monóxido de carbono	100

Fuente: Lcda. Lupe Zabala

En la tabla VI.VII se puede observar los valores normales de las pulsaciones por minuto en las personas adultas y los problemas causados por alguna alteración de estas:

Tabla VI.VII. Pulsaciones por minuto

Normal	60 a 100 pulsaciones por minuto
Bradycardia	Menor a 60 pulsaciones por minuto
Taquicardia	Mayor a 100 pulsaciones por minuto

Fuente: Lcda. Lupe Zabala.

Con el oxímetro también se puede medir el número de pulsos. Y se conecta de la siguiente manera como se indica en la figura 39.

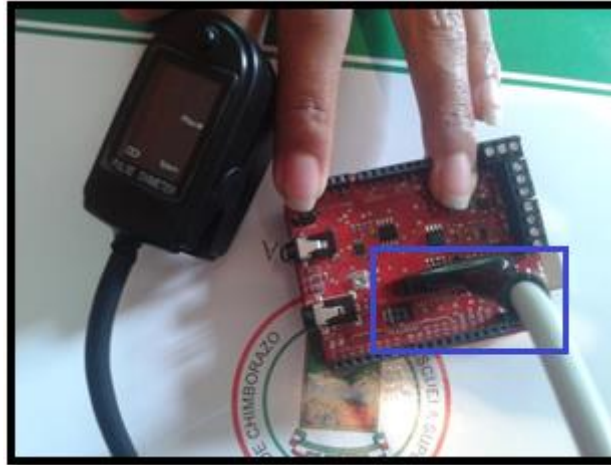


Figura 39: Conexión del sensor de pulso y oxígeno en la sangre

Fuente: Elaboración propia.

6.4 CONFIGURACIÓN DE LOS MODULOS XBEE S1

Para la configuración de los módulos XBee S1 se realiza mediante el software X-CTU en donde se va a configurar todas las características que va a tener la red, como es la dirección, la seguridad, el tipo de dispositivo que es en la red, para lo cual se debe utilizar el XBee USB Adapter que es donde se coloca el XBee S1.

- Primeramente se procede a abrir el programa X-CTU en donde se selecciona el número de puerto que tiene el Xbee USB Adapter que en este caso es el COM 3, además se selecciona los baudios de comunicación, Control de flujo, Bits de datos, Paridad y los bits de parada como se muestra en la Figura 40:

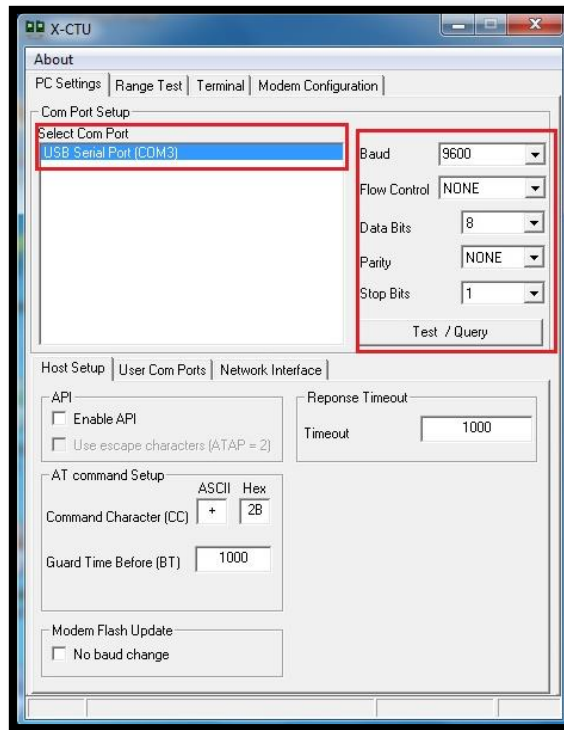


Figura 40: Pestaña PC settings del software X-CTU

Fuente: Elaboración propia.

- Luego se procede a dar clic en Modem Configuration y se da clic en Read para así leer los datos que van hacer configurados en el XBee S1 y verificar el funcionamiento del mismo, como se observa en la Figura 41.

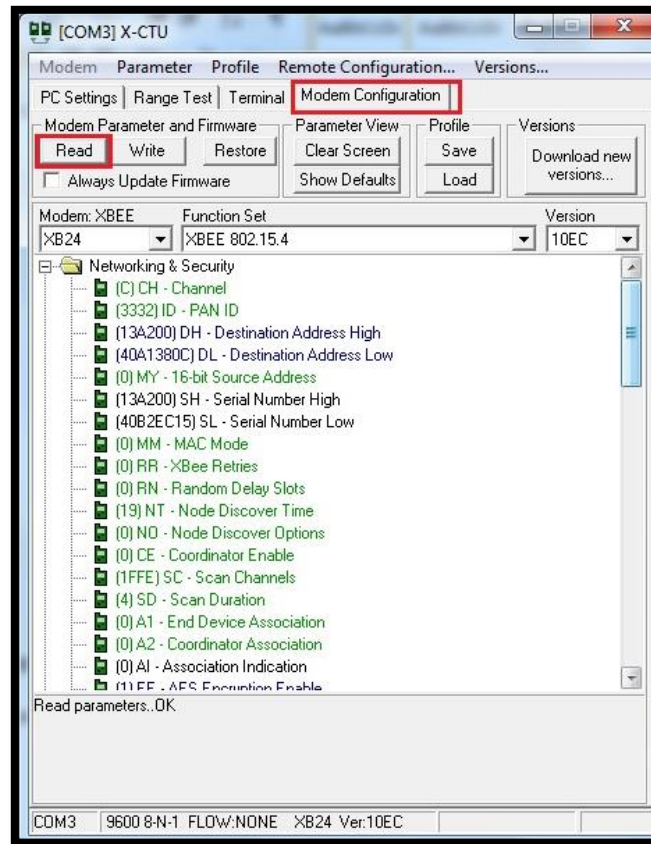


Figura 41: Pestaña Modem configuration del Software X-CTU

Fuente: Elaboración propia.

- Configuración de los parámetros necesarios para la comunicación de la red como son Channel, PAN ID, Destination Address High, Destination Address Low, MY y para la seguridad de la red se va a configurar en el modo AES Encryption Enable en la Tabla VI.VIII se puede ver los valores que van a ser configurados.

Tabla VI.VIII. Parámetros de configuración XBEE s1

Parámetros	Valor XBee S1 (1)	Valor XBee S1 (2)
Channel	C	C
PAN ID	3332	3332
Destination Address High	13A200	13A200
Destination Address Low	40A1380C	40B2EC15
MY	1	0
AES Encryption Enable	1	1

Fuente: Elaboración propia

- ✓ El canal que se está utilizando es el canal C aunque puede ser configurado en otros canales más.
- ✓ La dirección PAN ID con la que se va a configurar es 3332, esta dirección va hacer configurada en los dos XBee de modo que se va a tener un enlace punto a punto.
- ✓ La Destination Address High que está configurada es 13A200 esta dirección corresponde al número que viene en el XBee S1 desde su fabricación, por lo general esta dirección siempre es la misma en todos los XBee S1.
- ✓ La Destination Address Low con la que se configuro el XBee S1 es 40A1380C esta dirección es la dirección del otro dispositivo XBee S1, estas direcciones vienen en los dispositivos desde su fabricación, esta dirección es única en cada dispositivo.



Figura 42: Direcciones de los XBee S1 usados.

Fuente: Elaboración propia.

- ✓ MY es la dirección de origen y es de 16 bits por lo general es "0" o "1", en el primer XBee S1 se va a configurar con el "0" y en el segundo XBee S1 con "1".
- ✓ AES Encryption Enable es la encriptación de los datos de manera que solo el receptor pueda leerlos, este tipo de encriptación es uno de los más seguros y más usados en la actualidad. Para activar este tipo de encriptación se da clic en la pestaña que contiene AES y se pone el número "1", que indica que ya está activado. En la Figura 42 se muestra la configuración realizada en los dos XBee S1:

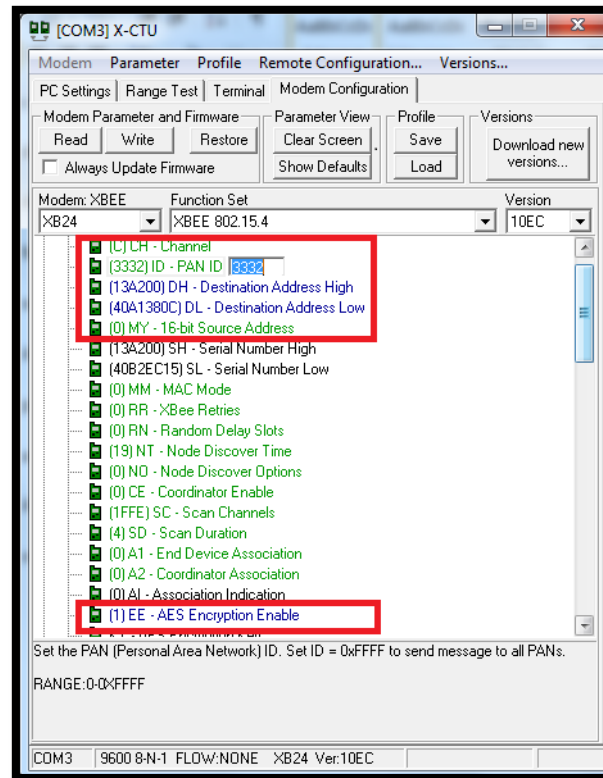


Figura 42: Configuración del CH, PAN ID, DH, DL, MY y AES en el software X-CTU.

Fuente: Elaboración propia.

- Finalmente se da clic en Write mediante esta opción se permite guardar la configuración realizada en el XBee S1 y así se tiene listo para su utilización, al realizar esto al final nos da unos mensajes que indica que todo fue configurado correctamente como se puede observar en la Figura 43:

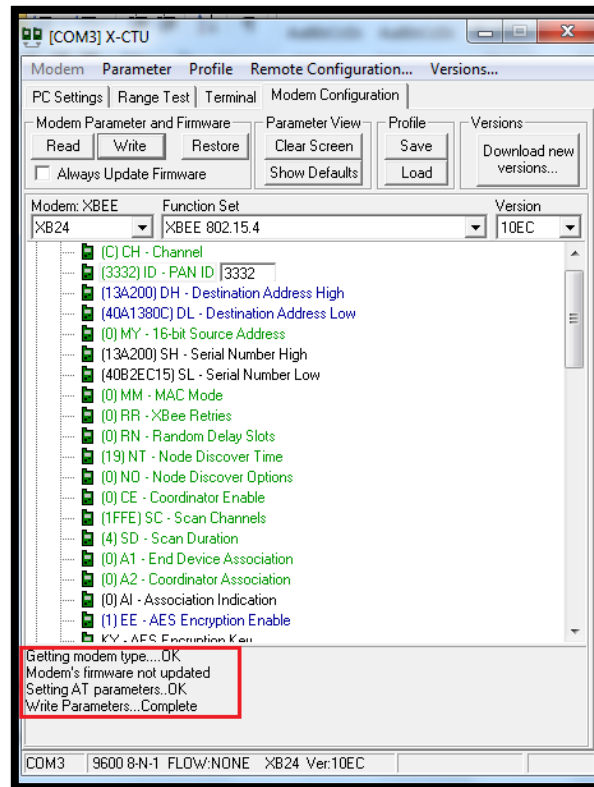


Figura 43: Guardando la configuración en el XBee S1.

Fuente: Elaboración propia.

6.5 CONFIGURACIÓN DEL ARDUINO UNO

Cada uno de los sensores son configurados en el Arduino uno, en la parte de configuración del programa en void setup (); pero antes hay que llamar a las siguientes librerías:

```
#include <PinChangeInt.h>  
#include <eHealth.h>
```

Estas librerías se tienen que descargar desde el internet y copiar en las librerías de arduino.

En el Serial. Begin() se configuró la velocidad de los datos de la placa electrónica arduino uno, en este caso de 9600 baudios. Además se inicializó el sensor de pulso y oxígeno en la sangre. Después se lee el valor que tiene el sensor. .

```
void setup()
{
  Serial.begin(9600);
  eHealth.initPulsioximeter();
  PCintPort::attachInterrupt(6, readPulsioximeter, RISING);
}
```

En la parte de ejecución del programa void loop() se inicia la captura de los datos empezando por el sensor de temperatura corporal; el sensor de flujo de aire y por último se ejecuta al sensor de pulso y oxigenación de la sangre con un tiempo de espera de 1000 ms. Se define el tipo de variable y se llama al sensor para obtener la información. Después se imprime el valor obtenido.

```
void loop() {

  float temperature = eHealth.getTemperature();
  float air = eHealth.getAirFlow();
  Serial.print(temperature, 2);
  Serial.print(air, 2);
  Serial.print(eHealth.getBPM());

  delay(1000);
}
```

Se define un contador para que realice solo 50 medidas el sensor de pulso y oxígeno en la sangre para así disminuir la latencia.

```
  delay(250);
}

void readPulsioximeter(){
  cont ++;
  if (cont == 50) {
    eHealth.readPulsioximeter();
    cont = 0;
  }
}
```

Después de configurar este código en el arduino IDE se debe primero verificar que este correcto como se muestra en la Figura 44.

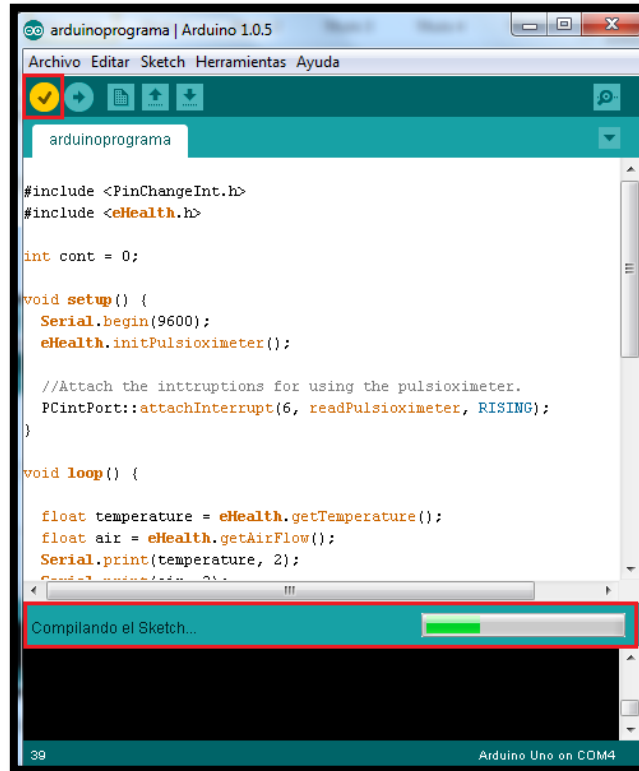


Figura 44. Verificación del programa en arduino.

Fuente: Elaboración propia.

Si nos sale que no hay errores entonces hacemos clic en el siguiente botón para cargar en el arduino como se indica en la Figura 45.

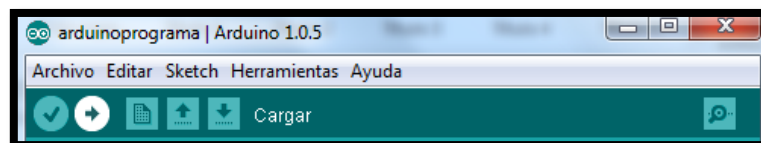


Figura 45. Cargando programa en arduino.

Fuente: Elaboración propia.

6.6 ELABORACIÓN DE LA BASE DE DATOS

Para la elaboración de la base de datos se toma en cuenta los datos que se tiene en el prototipo en este caso se va a tener una tabla en la cual se va a ir almacenando la información tomada por cada uno de los sensores, los datos que se van a tener en la tabla son: Temperatura, Pulso y Oxigenación de la sangre.

La construcción de la base de datos se realizó en MySQL, se utilizó este programa porque es open source además de que es muy rápido y fácil de usar.

La base de datos se llama Salud en la cual se va a tener los datos de todos los sensores como se puede visualizar en la Tabla VI.IX.

Tabla VI.IX. Tabla de la base de datos

Número de Paciente	Nombre y Apellidos	Temperatura	Pulso	SpO2
1	Juan Pico	60	36,1	95
2	Washington Santos	83	34,2	98
3	Jefferson Barragán	84	36,2	96

Fuente: Elaboración propia

6.7 DISEÑO DE LA INTERFAZ

Para poder visualizar de una manera agradable los datos tomados por los sensores se realiza el diseño de una interfaz gráfica para ello se utilizó Delphi XE7, se usó este programa porque se puede conectar con la base de datos que se tiene en MySQL, permitiendo la gestión de los valores almacenados como es consultas, modificar, entre otras opciones, además que facilita la conexión de la interfaz con el puerto serial en este caso el puerto COM3.

6.7.1 Pasos para la creación de la interfaz

- Para la creación de la interfaz se procede a abrir Delphi XE7 cómo se puede visualizar en la Figura 46.



Figura 46: Abriendo Delphi XE7

Fuente: Elaboración propia

- A continuación se configura la interfaz que se va a llamar Sistema de Toma de Signos Vitales, en la pantalla principal se va a tener un menú: Pacientes, Ingreso de Pacientes y Toma de Signos. En el menú Pacientes se tiene la opción de realizar consultas de pacientes como se puede observar en la figura 47:



Figura 47: Pantalla principal de la interfaz gráfica.

Fuente: Elaboración propia

- Para la configuración de esta pantalla principal se realiza mediante el diseño de formularios visuales haciendo más fácil la programación, aquí se va conectando a cada formulario según el diseño que se desee, en la figura 48 se puede visualizar como está estructurado los formularios utilizados:

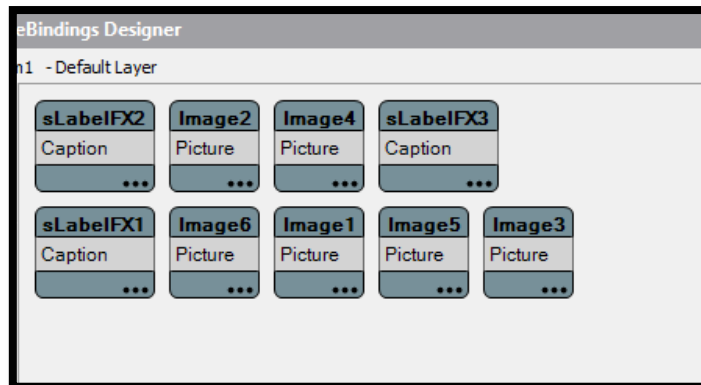


Figura 48: Estructura en bloques de la configuración de la interfaz gráfica.

Fuente: Elaboración propia

- En el menú de ingreso de Paciente se tiene los siguientes datos que se deben llenar de los pacientes, aquí se tiene las opciones de Aceptar, Borrar, Agregar, Refrescar y Cancelar como se observa en la Figura 49:

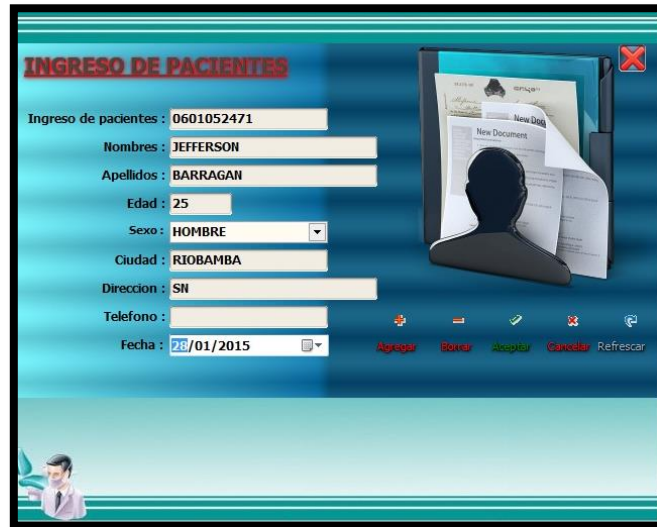


Figura 49: Ventana para el ingresos de pacientes en la interfaz gráfica.

Fuente: Elaboración propia

- La estructura de los formularios para la pantalla de Ingreso de Pacientes es la siguiente, como se puede visualizar estos formularios facilitan la programación mediante estructuras que van conectadas una con otras como se observa en la Figura 50:

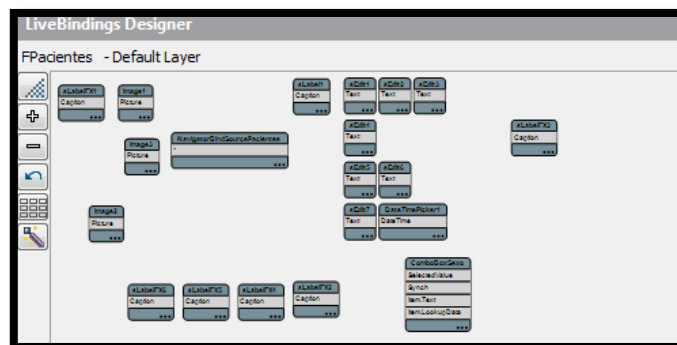


Figura 50: Estructura en bloques de la ventana para el ingresos de pacientes.

Fuente: Elaboración propia

- En la pantalla de Signos Vitales se puede visualizar como está la interfaz que va a mostrar los signos de cada paciente, en esta pantalla se tiene las alarmas que va a presentar en la alteración de cualquier signo vital:

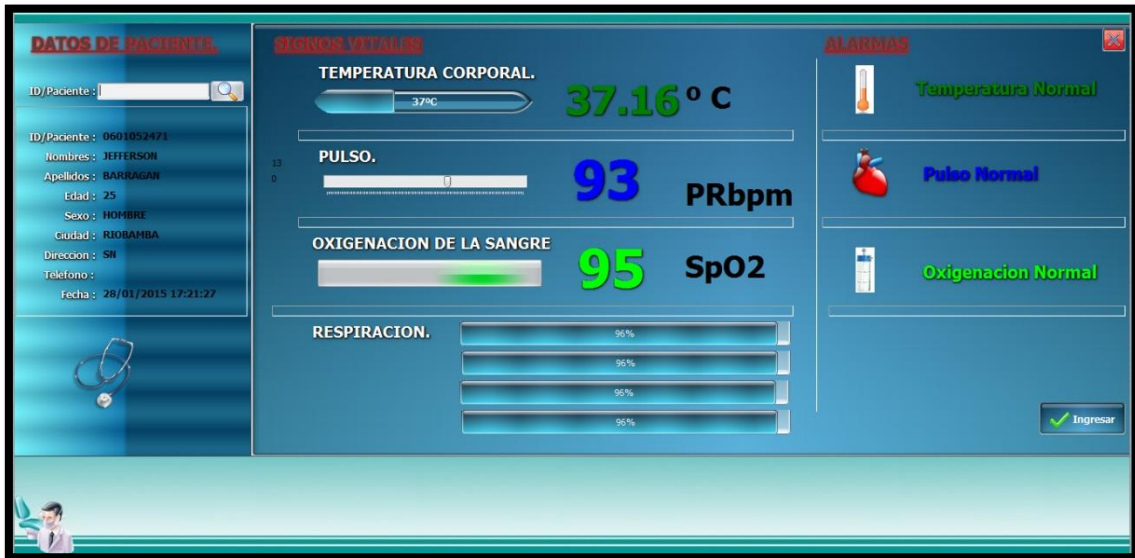


Figura 51: Ventana de los signos vitales.

Fuente: Elaboración propia

- La estructura de los formularios que se utilizó para la construcción de esta interfaz se encuentra en la Figura 52. Como se puede observar en esta parte se tiene muchos más formularios que en las otras pantalla esto es debido a que en esta parte se tiene las alarmas de cada sensor, además de la visualización de los datos del paciente:

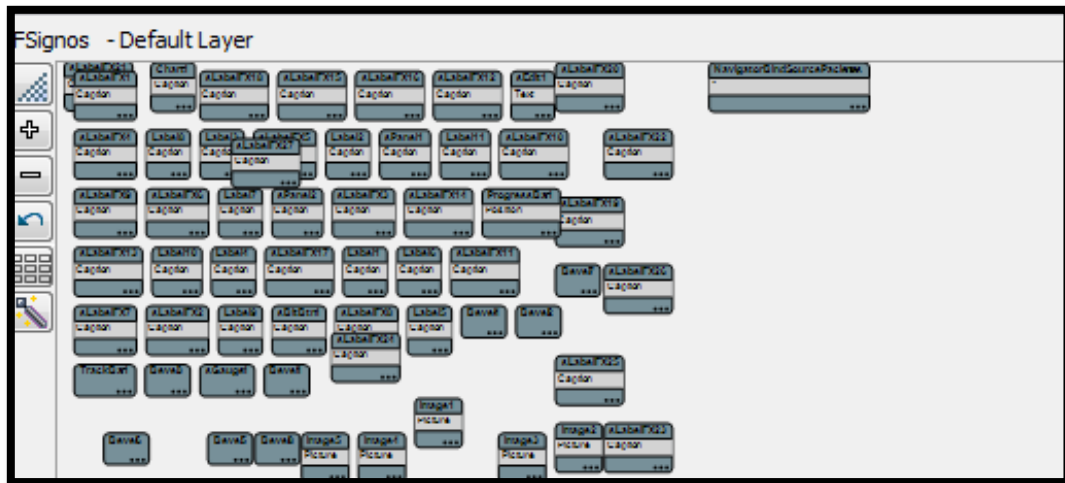


Figura 52: Estructura de la ventana de la visualización de los signos vitales.

Fuente: Elaboración propia

6.7.2 Activación de Alarmas

Para la activación de las alarmas se tomó en cuenta los valores de referencia que fueron facilitados por la Lcda. Lupe Zabala, como se puede observar en la Figura 53. las alarmas se van a activar dependiendo si la temperatura, pulso, oxigenación de la sangre se alteran dándonos un mensaje como en el caso de que la temperatura este alta nos dirá Fiebre Moderada, o en el caso de que el pulso este fuera de los valores normales nos dirá Taquicardia.



Figura 53: Visualización de los controles.

Fuente: Elaboración propia

6.7.3 Guardar la información

Para guardar la información mostrada en la interfaz se lo hizo de dos maneras la primera lo va hacer cuando se active cualquier alarma de algún signo vital, el sistema lo va a guardar; con esto se tiene un sistema automático que no va hacer necesario que una persona esté pendiente de guardar la información en caso de surgir una emergencia. La segunda forma es de manera manual en el caso de que la persona encargada quiera hacerlo.

6.7.4 Comunicación Puerto Serial

Para la comunicación a través del puerto serial en este caso puerto COM3 se utilizó ComPort que es un componente de Delphi XE7 que facilita la comunicación en Windows, en este COM se va a tener la información recibida por el segundo XBee S1, que es el receptor en la red y los datos receptados por este van hacer guardados en la base de datos y visualizados en la interfaz, en la Figura 54 se visualiza el ComPort en Delphi XE7 y la librería en donde se encuentra.

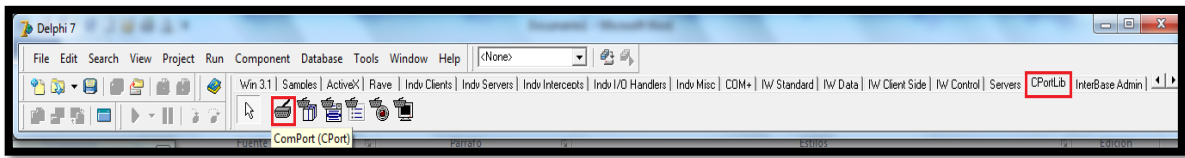


Figura 54: Puerto ComPort en Delphi EX7.

Fuente: Elaboración propia

Las líneas de código que se necesita configurar para que empiece a recibir la información en la base de datos en donde se indica el número de puerto por el cual se va a conectar, los baudios de comunicación, los bits de paridad y los bits de datos es:

```
object ComPort1: TComPort
  BaudRate = br9600
  Port = 'COM3'
  Parity.Bits = prNone
  StopBits = sbOneStopBit
  DataBits = dbEight
```

6.7.5 Conexión con la Base de Datos

Para la conexión con la base de datos se utiliza el icono ADOConnection (ADODB) en donde se podrá configurar la base de datos que se tiene creado, la misma que se llama SALUD y se procede a probar la conexión y luego de esto se da clic en OK en donde ya se tiene una conexión exitosa.

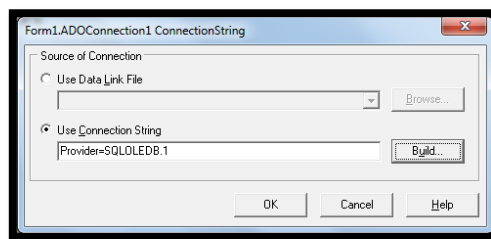


Figura 55: Conexión de la interfaz con la base de datos.

Fuente: Elaboración propia

6.8 FUNCIONAMIENTO DEL PROTOTIPO DE RED

Para determinar el correcto funcionamiento del prototipo se procedió a tomar datos en el Centro de Atención Integral en Salud de la ESPOCH, a las personas que acuden a este centro médico, durante un periodo de una semana los datos fueron tomados de dos maneras: la primera con el Prototipo como se observa en la Imagen 56 y la segunda de manera manual como tradicionalmente se lo hace, esto se hizo con la ayuda de la Lcda. Lupe Zabala como se observa en la Imagen 57.

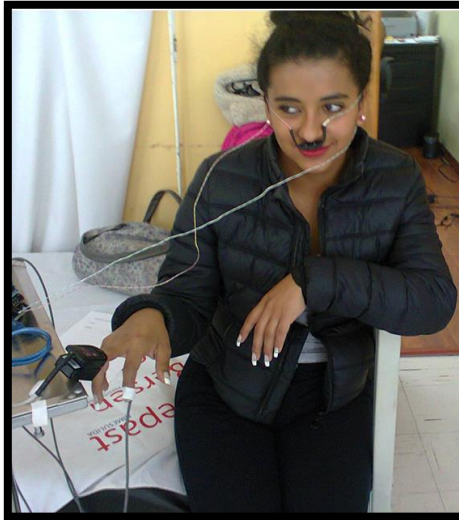


Figura 56: Medición de signos vitales con el prototipo.

Fuente: Elaboración propia



Figura 57: Medición de signos vitales manualmente.

Fuente: Elaboración propia

Para poder ver los cambios que surgen en la transmisión de la información se planteó tres escenarios en donde se ha tomado en cuenta el tiempo de la transmisión con el prototipo y la distancia a la que estaban ubicados tanto el prototipo como el receptor de la información.

6.8.1 Escenario 1

Para este escenario se ubicó el prototipo a una distancia de 2m del dispositivo receptor XBee S1 en el centro médico en donde se va a observar el tiempo que tarde en transmitirse los datos al receptor como se muestra en la Figura 58.

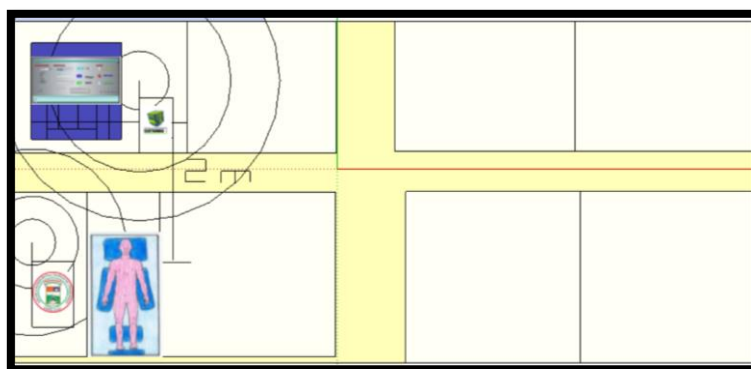


Figura 58: Escenario 1 del prototipo.

Fuente: Elaboración propia

6.8.2 Escenario2

En este escenario se ubicó el prototipo a una distancia de 15m del dispositivo receptor XBee S1 tomando en cuenta los obstáculos presentes en el centro médico, el prototipo se colocó en la estación de enfermería en donde se toma los signos vitales y el dispositivo receptor se ubicó en la sala de espera del centro médico como se puede observar en la Figura 59.

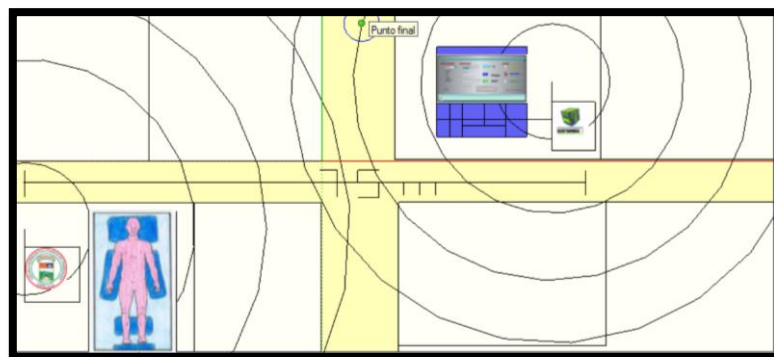


Figura 59: Escenario 2 del prototipo.

Fuente: Elaboración propia

6.8.3 Escenario 3

En este escenario se alcanzó la cobertura máxima del XBee S1 sin línea de vista; en el centro médico esta distancia es de 20 m. El prototipo para la toma de los signos vitales se ubicó en la

estación de enfermería y el dispositivo receptor fue ubicado cerca al departamento de Odontología en la Figura 60 se muestra la ubicación de la red.

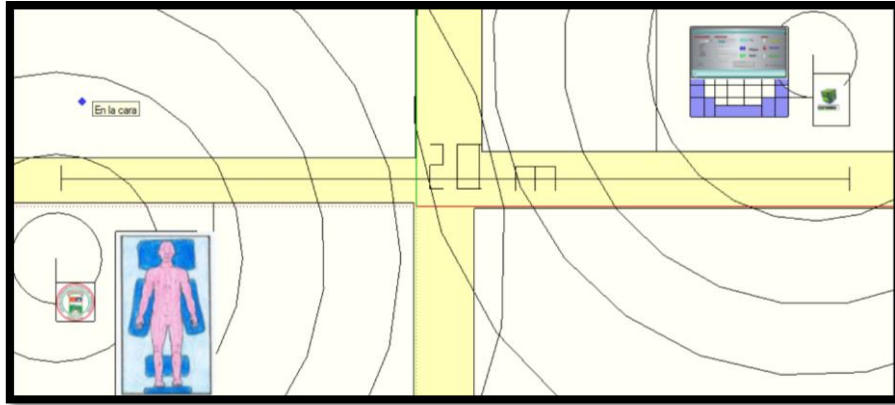


Figura 60: Escenario 3 del prototipo.

Fuente: Elaboración propia

CAPÍTULO VII

ANÁLISIS Y RESULTADOS

7.1 INTRODUCCIÓN

En este capítulo se va determinar cuál es el mejor escenario de los planteados en la simulación para saber si este proyecto se puede implementar. Una vez obtenidos los resultados de este análisis, se procederá a la implementación del prototipo con los datos obtenidos.

Después de la implementación se va realizar un análisis del tiempo de respuesta del prototipo y de la manera manual en la toma de signos vitales, para realizar la comprobación de la hipótesis a través de estos resultados.

7.2 COMPROBACIÓN DE LA SIMULACIÓN DEL PROTOTIPO DE RED EN NS-2

Después de obtener las variables en los tres escenarios con diferentes tiempos de simulación vamos a compararlos para determinar el mejor escenario.

7.2.1 Métodos a utilizar

El método aplicado para analizar las variables obtenidas de los tres escenarios, es el Método Estadístico de Diseño Completamente Aleatorizado también llamado modelo de análisis de la varianza para clasificación simple de las observaciones. Este método es muy fácil de implementar y se lo realiza de la siguiente manera:

Determinar el número de muestras (n)

$$n = (t * r)$$

Dónde:

- t : número de tratamiento
- r : número de repeticiones de n

Modelo estadístico del Diseño Completamente Aleatorizado

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}$$

- Y_{ij} : Variable respuesta de la ij -ésima unidad experimental
- μ : Efecto de la media general
- τ_i : Efecto del i -ésimo tratamiento

- ϵ_{ij} : Efecto del error experimental asociado a la ij-esima unidad experimental

Análisis de Varianza

Hipótesis Nula Ho: $\mu_1 = \mu_2 = \mu_3$ (Los i tratamientos tienen el mismo efecto sobre la variable en estudio)

Hipótesis Alterna Ha: $\mu_1 \neq \mu_2 \neq \mu_3$ (No todos los tratamientos tienen el mismo efecto sobre la variable en estudio)

Tabla de Datos

Tabla VII.X: Datos del método al azar.

Tratamientos	Repeticiones					Y _{i.}
	1	2	3	4	R	
1	Y ₁₁	Y ₁₂	Y ₁₃	...	Y _{1r}	Y _{1.}
2	Y ₂₁	Y ₂₂	Y ₂₃	...	Y _{2r}	Y _{2.}
3	Y ₃₁	Y ₃₂	Y ₃₃	...	Y _{3r}	Y _{3.}
.
.
.
R	Y _{t1}	Y _{t2}	Y _{t3}	...	Y _{tr}	Y _{..}

Fuente: Elaboración propia

Tabla de Análisis de Varianza

Tabla VII.XI: Tabla Anova.

Fuentes de Variación	Grados de Libertad	Suma de Cuadrados	Cuadrados Medios	Fc Calculada	Ft Requerida
Tratamientos	t - 1	$\sum[(Y^2_{..}/r Y^2_{.})/(tr)]$	Sctrat/gltotal	Scmedios/Cmerror	
Error	t (r - 1)	Sctotal - Sctrat	Scerror/glerror		
Total	t r -1	$\sum[(Y_{ij} - Y^2_{..})/(tr)]$			

Fuente: Elaboración propia

Cuando se va a calcular la tabla de la varianza, se puede fácilmente crear por medio de Microsoft Excel.

Interpretación de los datos

Por medio de la tabla de análisis de la varianza o tabla anova se puede rechazar o aceptar la hipótesis H_0 por medio del valor de la probabilidad que es p . Entonces si $p > 0.05$ se acepta pero si $p < 0.05$ se rechaza. Permitiendo así un nivel de confianza del 95%.

7.2.1.1 Población y Muestra

7.2.1.1.1 Población

La población viene a ser los tres escenarios propuestos.

7.2.1.1.2 Muestra

Para la muestra vamos a tomar en cuenta todas las variables se obtuvimos a través de los filtros las cuales son el rendimiento, el retardo, throughput y el consumo de energía de los tres escenarios.

7.2.2 Análisis Comparativo

Se realizaron 30 tomas de cada uno de los tres escenarios para realizar el análisis comparativo de los mismos. En donde el número de muestras es igual:

$$n = (t * r) = 3 * 30 = 90$$

Los datos obtenidos se pueden observar a continuación:

Tabla VII.XII: Muestra de los datos Escenario 1

N° MUESTRA	TAMAÑO PAQUETE CBR	RENDIMIENTO (%)	DELAY (ms)	THROUGHPUT (KBps)	ENERGÍA CONSUMIDA (J)
1	32	99	0,0162136	0,7690	46662,835839
2	34	99	0,013928	0,7720	46661,711578
3	36	92	0,026583	0,7130	46683,359945
4	37	85	0,0574047	0,6600	46704,064239
5	40	82	0,0563183	0,6375	46705,416622
6	42	78	0,0469619	0,6095	46716,343167
7	44	98	0,0552169	0,7590	46670,903782
8	46	57	0,0980343	0,4425	46692,323585
9	47	88	0,0157487	0,6825	46668,245658
10	50	82	0,0627333	0,6410	46701,132841
11	52	82	0,0276981	0,6410	46683,188190
12	54	82	0,0480168	0,6385	46685,607406
13	56	87	0,0243196	0,6805	46672,061656
14	57	88	0,0174263	0,6885	46670,036151
15	62	97	0,023792	0,7520	46673,161581
16	64	76	0,0766033	0,5940	46724,287233
17	66	79	0,0861704	0,6155	46714,538542
18	67	93	0,0459686	0,7235	46681,848307
19	72	73	0,0982693	0,5685	46730,552203
20	74	64	0,2644035	0,4975	46743,831205
21	76	99	0,0185254	0,7720	46669,775354
22	77	76	0,1000313	0,5900	46725,644404
23	82	71	0,0875079	0,5565	46738,009951
24	84	77	0,1454495	0,6020	46726,867843

25	86	65	0,1116446	0,5090	46753,407250
26	87	65	0,1441265	0,5080	46747,807666
27	92	67	0,136837	0,5255	46742,111987
28	94	66	0,2882289	0,5135	46720,768643
29	96	80	0,0410743	0,6245	46692,424297
30	97	87	0,0192798	0,6770	46679,305476

Fuente: Simulador ns2.

Tabla VII.XIII: Muestra de los datos Escenario 2

N° MUESTRA	TAMAÑO PAQUETE CBR	RENDIMIENTO (%)	DELAY (ms)	THROUGHPUT (kbps)	ENERGIA CONSUMIDA (J)
1	32	52	0,0457787370	0,40750	46677,106038
2	34	46	0,4185345840	0,36000	46701,738189
3	36	54	0,0885274670	0,41950	46688,032297
4	37	58	0,0653597500	0,44900	46682,461268
5	40	50	0,1483892190	0,38850	46687,629197
6	42	50	0,2420166700	0,38750	46694,293716
7	44	48	0,2050329560	0,37550	46694,033497
8	46	31	0,4404890000	0,01750	46657,110382
9	47	43	0,1798124570	0,33450	46717,749945
10	50	54	0,2038457400	0,42500	46688,796054
11	52	44	0,1062588080	0,34750	46685,761989
12	54	35	0,1335073300	0,27700	46690,794054
13	56	33	0,1174054640	0,26250	46686,717753
14	57	62	0,0363148640	0,48150	46692,410692
15	62	35	0,1163949270	0,27450	46706,775066
16	64	47	0,0964452390	0,36400	46715,022372
17	66	43	0,1322138350	0,33500	46697,403884
18	67	49	0,1871095630	0,38350	46696,478366
19	72	40	0,1119539760	0,31600	46697,307525
20	74	72	0,6164522310	0,55800	46775,044541
21	76	58	0,0892894530	0,45000	46704,418459
22	77	38	0,1641746370	0,29600	46712,178959
23	82	38	0,3470962080	0,29500	46706,754897
24	84	48	0,5214752220	0,37300	46709,992426
25	86	55	0,0466499480	0,43200	46703,254153
26	87	48	0,1525509840	0,37700	46695,647077
27	92	46	0,1014504670	0,36300	46716,672257
28	94	50	0,4534974660	0,38800	46696,917560

29	96	40	0,2506232540	0,31700	46684,734236
30	97	53	0,2047248180	0,41300	46702,874189

Fuente: Simulador ns2.

Tabla VII.XIV: Muestra de los datos Escenario 3

N° MUESTRA	TAMAÑO PAQUETE CBR	RENDIMIENTO (%)	DELAY (ms)	THROUGHPUT (kbps)	ENERGIA CONSUMIDA (J)
1	32	78	0,019496876	0,6040	46675,641612
2	34	58	0,050893107	0,4565	46678,885071
3	36	49	0,093091453	0,3835	46687,895954
4	37	48	0,076385453	0,3765	46698,893083
5	40	79	0,257725805	0,6125	46741,302134
6	42	66	0,047308418	0,5165	46682,721726
7	44	41	0,131451319	0,3215	46711,168269
8	46	47	0,22761193	0,3640	46686,110504
9	47	62	0,027322256	0,4840	46683,732067
10	50	66	0,033853033	0,5110	46686,621763
11	52	77	0,308801871	0,5960	46770,853946
12	54	52	0,03864503	0,4085	46669,205254
13	56	55	0,028128763	0,4310	46685,060745
14	57	51	0,494450038	0,3955	46689,139179
15	62	65	0,022738215	0,5040	46677,914350
16	64	53	0,261250503	0,4170	46672,319144
17	66	51	0,070233127	0,3955	46689,461933
18	67	58	0,061645056	0,4550	46679,589547
19	72	48	0,411852069	0,3750	46715,476204
20	74	69	0,310609117	0,5360	46744,043979
21	76	56	0,07815118	0,4340	46687,610704
22	77	68	0,249080528	0,5295	46757,806009
23	82	70	0,205268391	0,5495	46701,426481
24	84	45	0,170549819	0,3485	46677,198938
25	86	66	0,093153382	0,5145	46681,597412
26	87	53	0,03514944	0,4145	46696,802549
27	92	48	0,090835042	0,3760	46711,183421
28	94	50	0,040756225	0,3895	46709,014398
29	96	53	0,057424911	0,4110	46701,615774
30	97	61	0,043360191	0,4735	46684,973958

Fuente: Simulador ns2.

Tabla VII.XV: Promedios de las muestras de los tres escenarios con 100s.

ESCENARIOS	RENDIMIENTO (%)	RETARDO(ms)	THROUGHPUT (Kbps)	ENERGÍA CONSUMIDA (J)
PROMEDIO ESCENARIO 1	81,13333300	0,075150500	0,632100000	46701,385753
PROMEDIO ESCENARIO 2	58,10000000	0,134574100	0,452800000	46697,842204
PROMEDIO ESCENARIO 3	47,33333333	0,200779176	0,362266667	46698,870368

Fuente: Elaboración propia.

Tabla VII.XVI: Promedios de las muestras de los tres escenarios con 3600s.

ESCENARIOS	RENDIMIENTO (%)	RETARDO(ms)	THROUGHPUT (Kbps)	ENERGÍA CONSUMIDA (J)
PROMEDIO ESCENARIO 1	84,0333333	0,05957333	25,06505	48774,1402
PROMEDIO ESCENARIO 2	43,6	0,20912134	12,4386167	47166,818
PROMEDIO ESCENARIO 3	31,8	0,48688542	9,35925	47124,8115

➤ RENDIMIENTO CON 100s

Planteamiento de la hipótesis

H_0 : El rendimiento es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El rendimiento es diferente en el escenario 1, escenario 2 y escenario 3.

$$H_1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XVII. Varianza del rendimiento (100s)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
ESCENARIO 1	30	2434	81,1333333	138,395402
ESCENARIO 2	30	1743	58,1	106,162069
ESCENARIO 3	30	1420	47,3333333	82,3678161

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XVIII. Anova del Rendimiento (100s)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	17888,95556	2	8944,477778	82,07818	9,3652E-21	3,10129576
Dentro de los grupos	9480,833333	87	108,9750958			
Total	27369,78889	89				

Fuente: Elaboración propia

Gráfico

En la figura 61 se observa la variación del rendimiento. El escenario 1 tiene un rendimiento 20% mayor que el escenario 2, mientras que el escenario 3 tiene un rendimiento 10 % menor que el escenario 2.

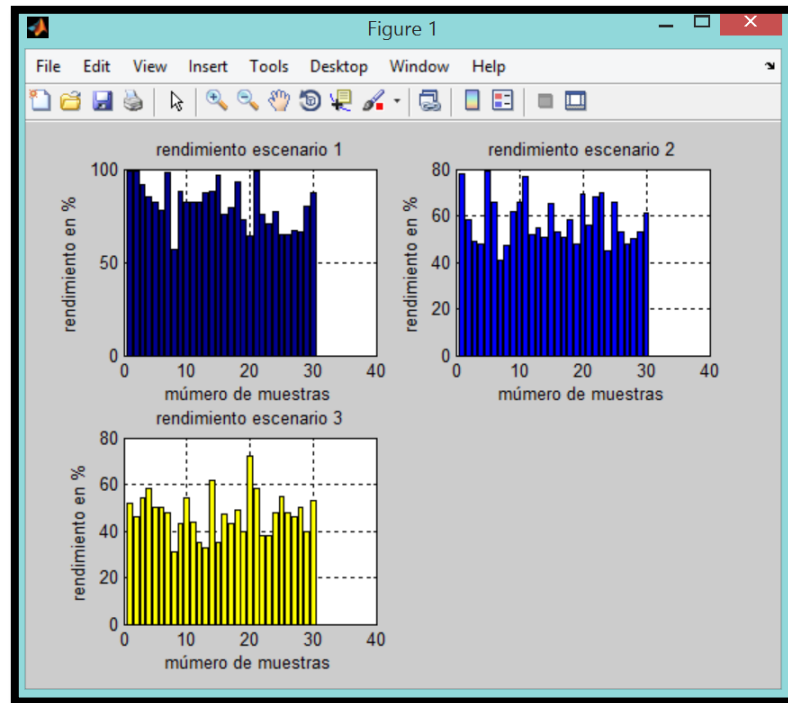


Figura 61: Promedio del Rendimiento (100s)

Fuente: Elaboración propia.

Interpretación

Por medio de la obtención de la tabla ANOVA se sabe que la probabilidad es de $9,3652E-21$ y entonces definimos que $p < 0.05$; así que la hipótesis H_0 se rechaza. Por lo tanto se dice que el escenario 1 presenta un mejor rendimiento que los otros escenarios.

➤ **RETARDO CON 100s**

Planteamiento de la hipótesis

H_0 : El retardo es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El retardo es diferente en el escenario 1, escenario 2 y escenario 3.

$$H1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XIX. Varianza del retardo (100s)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
ESCENARIO 1	30	2,254515899	0,0751505	0,00458677
ESCENARIO 2	30	4,037222548	0,1345741	0,01599673
ESCENARIO 3	30	6,023375274	0,2007792	0,02262487

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XX. Anova del Retardo (100s)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	0,236968296	2	0,1184841	8,22647296	0,00053429	3,101296
Dentro de los grupos	1,253042576	87	0,0144028			
Total	1,490010872	89				

Fuente: Elaboración propia

Gráfico.

En la Figura 62 se visualiza la variación del retardo que presentan en los tres escenarios propuestos. En el escenario 1 se observa que el retardo es casi la mitad que los otros escenarios, mientras que el escenario 2 y escenario 3 la diferencia retardo del retardo es poca, porque hay solo una diferencia de cuatro nodos.

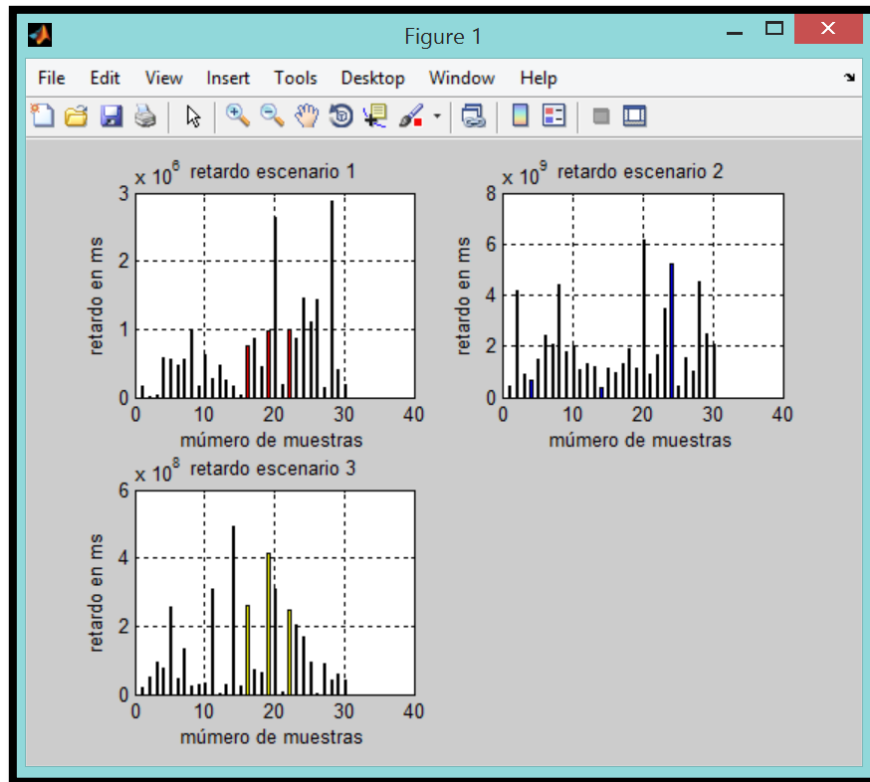


Figura 62. Promedio del Retardo (100s).

Fuente: Elaboración propia.

Interpretación

Por medio de la tabla ANOVA se obtiene que la probabilidad de 0,00053429 y se define que $p < 0.05$; así que la hipótesis H_0 se rechaza. Por lo tanto se dice que el escenario 1 presenta un menor retardo que los otros escenarios.

➤ **THROUGHPUT CON 100s**

Planteamiento de la hipótesis

H_0 : El throughput es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H1: El throughput es diferente en el escenario 1, escenario 2 y escenario 3.

$$H1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XXI. Varianza del throughput (100s)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
ESCENARIO 1	30	18,964	0,63211667	0,00825
ESCENARIO 2	30	13,584	0,4528	0,0063
ESCENARIO 3	30	10,868	0,36226667	0,00848

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXII. Anova del Throughput (100s)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	1,13169774	2	0,56584887	73,7047	1,8844E-19	3,10129576
Dentro de los grupos	0,66792001	87	0,00767724			
Total	1,79961775	89				

Fuente: Elaboración propia

Gráfico.

En este caso vamos a observar el THROUGHPUT real que han presentado cada escenario durante la simulación en el ns2.

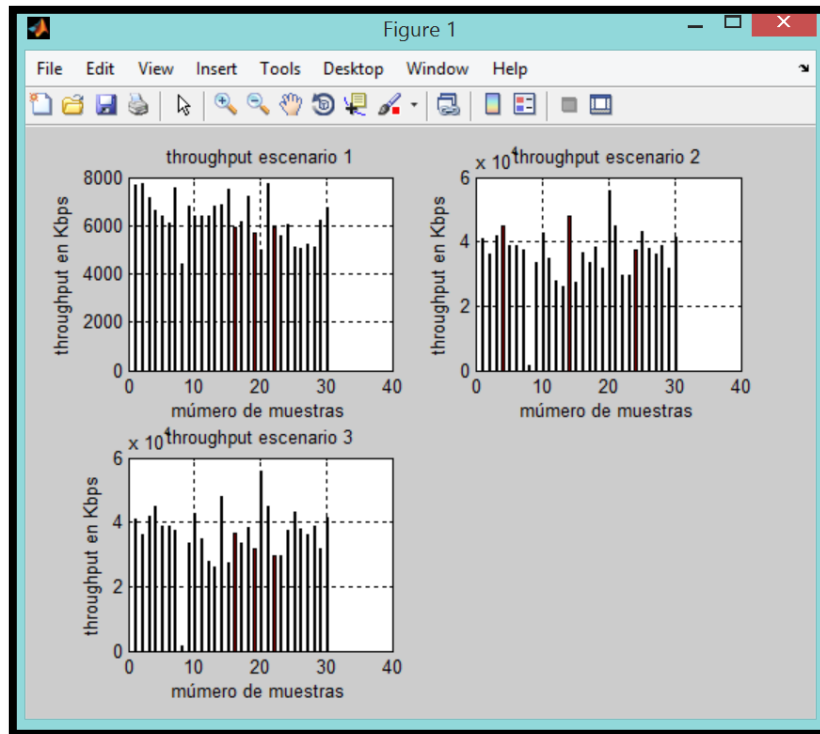


Figura 63. Promedio de THROUGHPUT (100s).

Fuente: Elaboración propia.

Interpretación

A través de la tabla ANOVA se observa que la probabilidad de $1,8844E-19$ y se obtiene que $p < 0.05$; así pues la hipótesis H_0 se rechaza. Por lo tanto se dice que el escenario 1 presenta un mayor THROUGHPUT que los otros escenarios.

➤ CONSUMO DE ENERGÍA CON 100s

Planteamiento de la hipótesis

H_0 : El consumo de energía es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El consumo de energía es diferente en el escenario 1, escenario 2 y escenario 3.

$$H1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XXIII. Varianza del consumo de energía (100s)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
ESCENARIO 1	30	1401041,57	46701,3858	818,430279
ESCENARIO 2	30	1400935,27	46697,8422	651,179849
ESCENARIO 3	30	1400966,11	46698,8704	371,16482

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXIV. Anova del Consumo de Energía (100s)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	199,410307	2	99,7051536	0,16249431	0,8502783	3,10129576
Dentro de los grupos	53382,4735	87	613,591649			
Total	53581,8838	89				

Fuente: Elaboración propia

Gráfico.

En este caso vamos a observar que el consumo de energía es el mismo en cada escenario durante la simulación en el ns2.

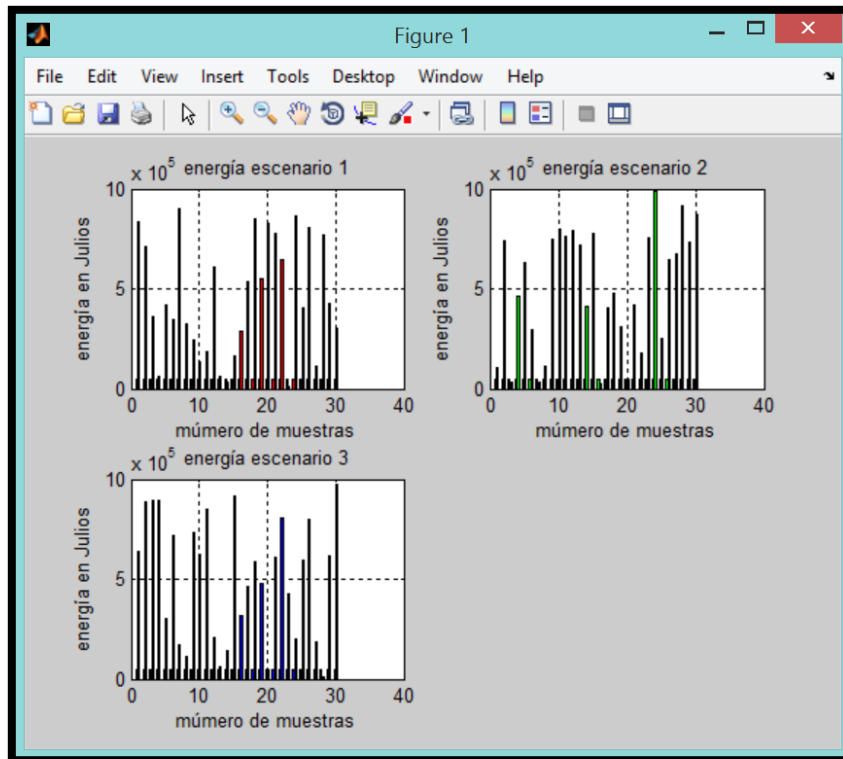


Figura 64. Promedio de consumo de energía (100s).

Fuente: Elaboración propia.

Interpretación

A través de la tabla ANOVA se observa que la probabilidad de 0,8502783 y se obtiene que $p > 0.05$; así pues la hipótesis H_0 se acepta. Por lo tanto se dice que el escenario 1, escenario 2 y escenario 3 presenta un mismo consumo de energía como se puede observar en la gráfica anterior hay una diferencia mínima.

➤ RENDIMIENTO CON 3600s

Planteamiento de la hipótesis

H_0 : El rendimiento es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H1: El rendimiento es diferente en el escenario 1, escenario 2 y escenario 3.

$$H1: \mu_1 \neq \mu_2 \neq \mu_3$$

Cálculo de la varianza

Tabla VII.XXV. Varianza del rendimiento (3600s)

Grupos	Cuenta	Suma	Promedio	Varianza
ESCENARIO1	30	2521	84,0333333	50,8609195
ESCENARIO2	30	1308	43,6	287,627586
ESCENARIO3	30	954	31,8	239,544828

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXVI. Anova del Rendimiento (3600s)

ANÁLISIS DE VARIANZA						
Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	45024,1556	2	22512,0778	116,837956	2,2657E-25	3,10129576
Dentro de los grupos	16762,9667	87	192,677778			
Total	61787,1222	89				

Fuente: Elaboración propia

Gráfico.

En este caso vamos a observar que el rendimiento de cada escenario durante la simulación en el ns2. En el escenario el rendimiento de 30% mayor que en los otros escenarios., mientras que en los escenario 2 y escenario 3 existe una variación del 10%.

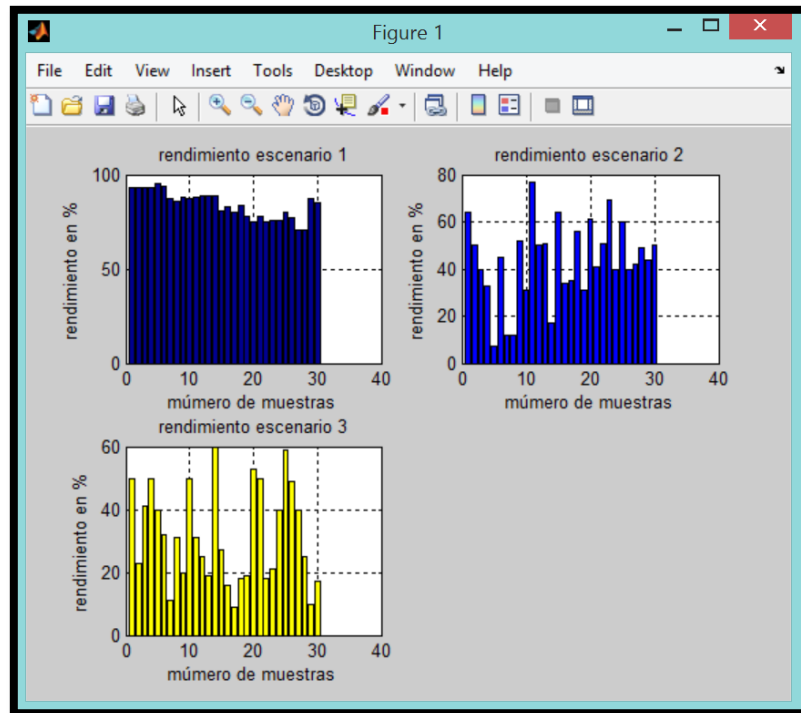


Figura 65. Promedio de rendimiento (3600s).

Fuente: Elaboración propia.

Interpretación

Con los datos obtenidos de la tabla ANOVA se observa que la probabilidad es de 2,2657E-25 y se observa que $p < 0.05$; entonces la hipótesis H_0 se rechaza. Por lo tanto se dice que el escenario 1 tiene el mejor rendimiento de los tres escenarios.

➤ **RETARDO CON 3600s**

Planteamiento de la hipótesis

H_0 : El retardo es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El retardo es diferente en el escenario 1, escenario 2 y escenario 3.

$$H1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XXVII. Varianza del retardo (3600s)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
ESCENARIO1	30	1,78720004	0,05957333	0,0012541
ESCENARIO2	30	6,2736403	0,20912134	0,17583953
ESCENARIO3	30	14,6065626	0,48688542	0,2979372

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXVIII. Anova del Retardo (3600s)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	2,82113107	2	1,41056553	8,90825683	0,00030227	3,10129576
Dentro de los grupos	13,775894	87	0,15834361			
Total	16,597025	89				

Fuente: Elaboración propia

Gráfico.

En este caso vamos a visualizar el retardo que tiene cada escenario durante la simulación en el ns2. Se va incrementando mientras más nodos se incluyen en la red, aunque en el escenario2 y tres la diferencia es poca; ya que solo aumenta en 4 nodos.

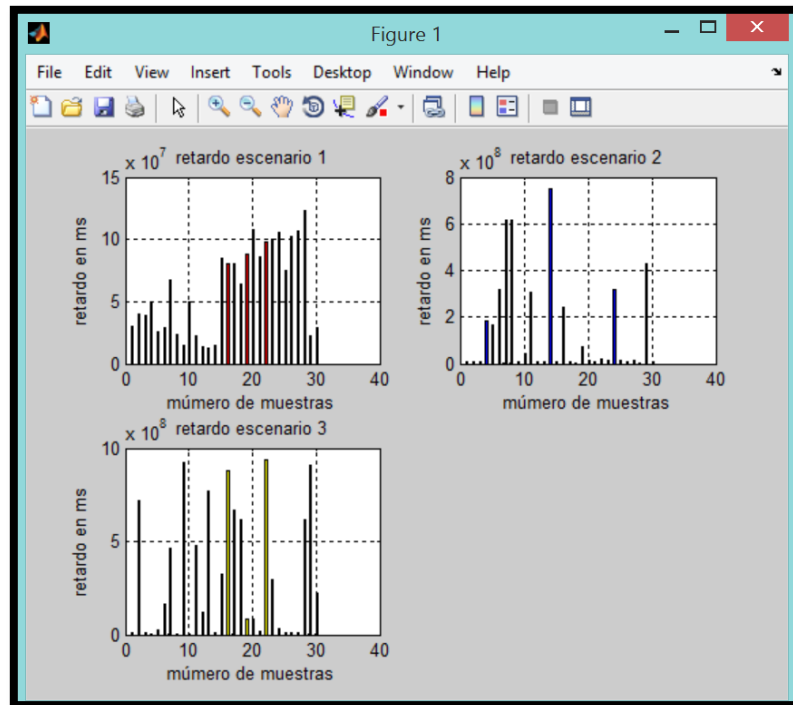


Figura 66. Promedio de retardo (3600s).

Fuente: Elaboración propia.

Interpretación

Con los datos obtenidos de la tabla ANOVA se observa que la probabilidad es de 0,00030227 y define que $p < 0.05$; entonces la hipótesis H_0 se rechaza. Por lo tanto se dice que el escenario 1 tiene el menor retardo en él envió de paquetes de los tres escenarios.

➤ **THROUGHPUT CON 3600s**

Planteamiento de la hipótesis

H_0 : El throughput es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El throughput es diferente en el escenario 1, escenario 2 y escenario 3.

$$H1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XXIX. Varianza del throughput (3600s)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
ESCENARIO1	30	751,9515	25,06505	5,75724556
ESCENARIO2	30	373,1585	12,4386167	26,9564263
ESCENARIO3	30	280,7775	9,35925	24,3818198

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXX. Anova del Throughput (3600s)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	4155,81471	2	2077,90736	109,180636	1,9041E-24	3,10129576
Dentro de los grupos	1655,76926	87	19,0318305			
Total	5811,58397	89				

Fuente: Elaboración propia

Gráfico.

En la Figura 67 observaremos el THROUGHPUT tiene cada escenario durante la simulación en el ns².

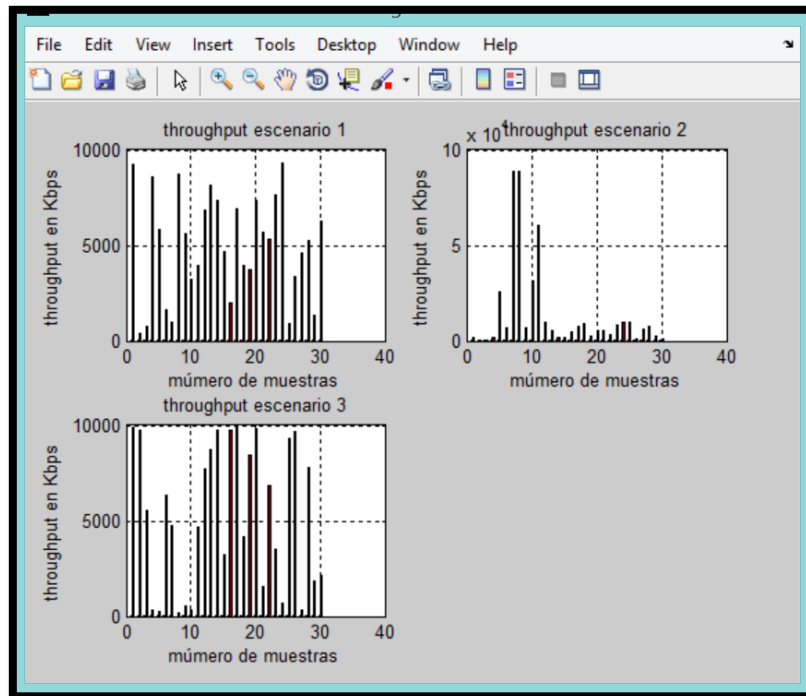


Figura 67: Promedio de throughput (3600s).

Fuente: Elaboración propia.

Interpretación

Con los datos obtenidos de la tabla ANOVA se observa que la probabilidad es de $1,9041E-24$ y define que $p < 0.05$; entonces la hipótesis H_0 se rechaza. Por lo tanto se dice que el escenario 1 tiene mayor throughput que los otros escenarios.

➤ CONSUMO DE ENERGÍA CON 3600s

Planteamiento de la hipótesis

H_0 : El throughput es igual en el escenario 1, escenario 2 y escenario 3.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El throughput es diferente en el escenario 1, escenario 2 y escenario 3.

$$H1: \mu_1 \neq \mu_2 \neq \mu_3$$

Cálculo de la varianza

Tabla VII.XXXI. Varianza del consumo de energía (3600s)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
ESCENARIO1	30	1463224,21	48774,1402	830634,305
ESCENARIO2	30	1415004,54	47166,818	31072,5317
ESCENARIO3	30	1413744,34	47124,8115	54008,5276

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXXII. Anova del Consumo de Energía (3600 s)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	53055343	2	26527671,5	86,9080258	1,8136E-21	3,10129576
Dentro de los grupos	26555745,6	87	305238,455			
Total	79611088,6	89				

Fuente: Elaboración propia

Gráfico.

En la Figura 68 observaremos el consumo de energía tiene cada escenario durante la simulación en el ns2.

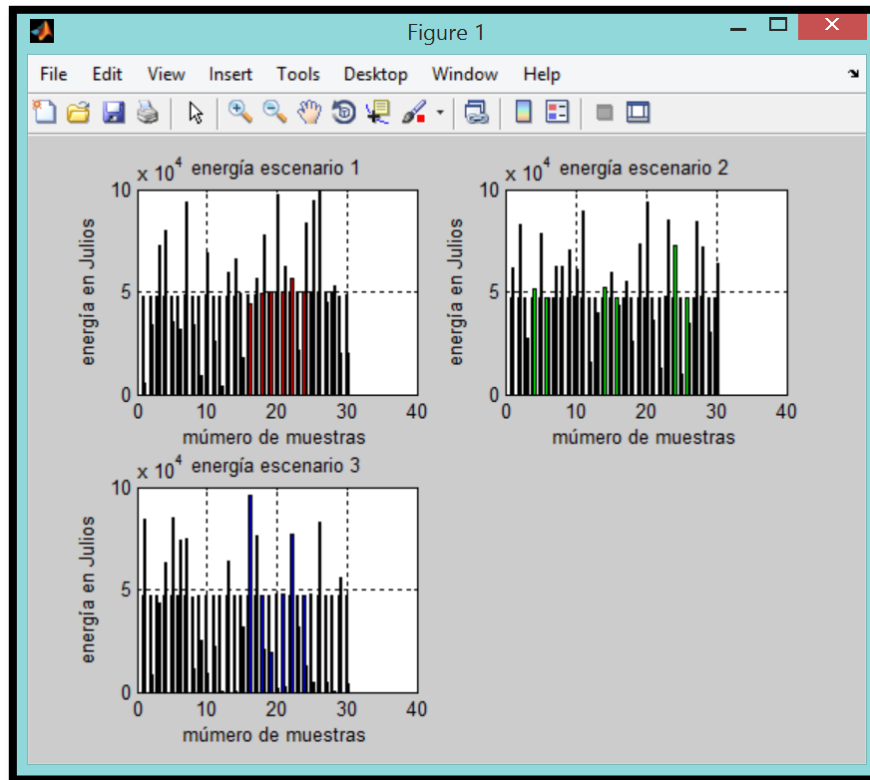


Figura 68: Promedio del consumo de energía (3600s).

Fuente: Elaboración propia.

Interpretación

Con los datos obtenidos de la tabla ANOVA se observa que la probabilidad es de $1,8136E-21$ y define que $p < 0.05$; entonces la hipótesis H_0 se rechaza. Por lo tanto se dice que el escenario 1 tiene un mayor consumo de energía que los otros escenarios al transcurrir mayor tiempo.

7.3 COMPROBACIÓN DE LA IMPLEMENTACIÓN DEL PROTOTIPO DE RED WSN

Después de obtener los datos de temperatura, pulso y respiración en el Centro de Atención Integral en Salud captados por el prototipo y además de medir los mismos signos de manera tradicional; se va a determinar el tiempo de respuesta de los mismos para determinar el mejor sistema.

7.3.1 Métodos a utilizar

El método aplicado para analizar el tiempo de respuesta obtenido en tres distancias diferentes de 2m, 15m y 20m, este es el Método Estadístico de Diseño Completamente Aleatorizado también llamado modelo de análisis de la varianza para clasificación simple de las observaciones. Este método es muy fácil de implementar y se lo realiza de la siguiente manera:

7.3.1.1 Población y Muestra

7.3.1.1.1 Población

La población viene dada por todos los pacientes que asisten al centro de salud durante el semestre.

7.3.1.1.2 Muestra

En la muestra se analiza el tiempo de respuesta que tiene el prototipo frente a la toma de datos manual de los signos vitales, donde se capturan los datos a diferentes distancias de 2m porque es la más homogénea a la toma de datos manual, 15m es la mitad de la cobertura descrita en el datasheet de los módulos XBee S1 que son 30m y 20 metros que es el máximo alcance que tiene el prototipo.

7.3.2 Análisis Comparativo

Se realizaron 30 tomas de cada distancia para realizar el análisis comparativo de los mismos. En donde el número de muestras es igual:

$$n = (t * r) = 3 * 30 = 90$$

Los datos obtenidos se puede observar a continuación:

Tabla VII.XXXIII. Captura de datos fisiológicos de la manera manual

# Paciente	Nombre	Apellido	Pulso	Temperatura	Tiempo (seg)
1	Johana	Ucles	88	36	120
2	Tobias	Cando	62	36	180
3	Rocio	Rojas	64	35,6	150
4	Dayana	Villareal	70	36,2	138
5	Pico	Juan	60	36,1	180
6	Washington	Santos	83	37,6	180
7	Jeferson	Barragan	68	36,2	150
8	David	Humandante	75	36,9	120
9	Marcela	Castillo	80	36,1	150
10	Kimberly	Niama	84	34,4	150
11	Julio	Mejia	73	36,2	120
12	Jorge	León	81	36,1	180
13	Braulio	Zarabia	68	34,4	180
14	Jhon	Tene	65	34,8	150
15	Erica	Zurita	60	36	120
16	Viviana	Guamanquispe	67	35,7	180
17	Cristian	Vasquez	78	35,5	180
18	Mayra	Carrillo	84	35,6	120
19	Carlos	Chamba	74	36,4	120
20	Miguel	Miranda	93	36,5	150
21	Fernanda	Pilamunga	68	35,6	150
22	Naybol	Machado	88	36,2	150
23	Marely	Cruz	80	36	180
24	viviana	Robalino	65	36,5	120
25	Jorge	Pazmiño	65	36,1	150
26	Erick	Villacres	72	35,3	180
27	Fernando	Marquez	60	35,9	180
28	Franklin	Lopez	80	36,8	180
29	Angel	Ortega	83	35,4	180
30	Fernando	Cudco	86	36,2	150
31	Freddy	Lopez	70	35,5	120
32	Tatiana	Chavez	70	36	150
33	Richard	Montesdeoca	97	35	180
34	Dario	Jordan	86	36,5	180
35	Ricardo	Aguilar	80	35,1	150
36	Cristian	Tarco	101	36,5	180

37	Daniel	Ripalda	79	35,7	180
38	Felipe	Valle	66	35,9	180
39	Katherine	Gavilanez	82	35,3	180
40	Isabel	Castillo	87	35,5	150
41	Mayra	Ortiz	75	38,9	120
42	Marllory	Cobos	82	36,4	120
43	Cristian	Jacho	61	35	180
44	Alvaro	Veintimilla	70	34,7	180
45	Ruben	Paucar	60	35,9	180
46	Cristian	Pazmiño	95	35,8	180
47	Juliana	Moya	73	36,5	120
48	Jose	Manobanda	78	36,9	120
49	Andrea	Montesdeoca	93	37,2	120
50	Lenin	Tierra	81	35,3	180
51	Andres	Solis	59	36,7	150
52	Jeferson	Iza	65	34,6	180
53	Edison	Borja	68	35,5	180
54	Santiago	Murillo	64	36,5	150
55	Marlon	España	81	36	150
56	Pamela	Barrazueta	78	35,6	150
57	Ivan	Armijo	73	36	180
58	David	Ojeda	60	36	180
59	Maria	Zambrano	52	35,1	180
60	Johana	Barreno	78	36,8	150
61	Pamela	Gusqui	56	35,3	120
62	Marilyn	Carrion	102	39,8	120
63	Gabriela	Rodriguez	55	37	150
64	Mercy	Loachamin	70	35,1	180
65	Walter	Crow	60	37	120
66	Jhon	Loaiza	90	35,4	180
67	Alex	Yautibuto	60	36,2	150
68	Liliana	Arias	74	35,5	150
69	Jonatan	Quezada	89	36	120
70	Karen	Paredes	70	36,9	120
71	Valeria	Sanchez	83	36,2	120
72	Cristian	Monar	74	35,7	180
73	Cristian	Viteri	70	36,5	180
74	Jairo	Arce	60	36	180
75	Edison	Bonilla	76	37,2	150
76	Andres	Carranza	58	36,2	150

77	Galo	Hurtado	70	35,6	150
78	Miriam	Vega	61	36,9	120
79	Paola	Obregon	85	35,4	120
80	Geovanna	Guerrero	63	35,7	180
81	Tatiana	Padilla	77	37,5	120
82	Geovanny	Cherrez	75	35,7	150
83	Cristian	Yugcha	74	35,5	150
84	Jessica	Proaño	80	36,7	150
85	Mario	Suarez	71	35,8	180
86	Liseth	Cevallos	74	35,1	180
87	Genova	Quispe	97	34,8	180
88	Alexandra	Parreño	80	37,9	120
89	Rebeca	Tenenaula	62	38,4	120
90	Jonatan	Leon	83	37,7	120

Fuente: Elaboración propia

Tabla VII.XXXIV. Captura de datos fisiológicos con el prototipo

# Paciente	Nombre	Apellido	Tiempo (seg)	Distancia	Temperatura	Pulso
1	Johana	Ucles	23	2	36,22	87
2	Tobias	Cando	20	2	36,19	65
3	Rocio	Rojas	15	2	35,95	74
4	Dayana	Villareal	40	2	37,43	71
5	Pico	Juan	17	2	36,1	59
6	Washington	Santos	10	2	38,33	93
7	Jeferson	Barragan	44	2	36,5	70
8	David	Humandante	25	2	37,53	73
9	Marcela	Castillo	23	2	36,07	78
10	Kimberly	Niama	45	2	34,76	85
11	Julio	Mejia	10	2	35,67	75
12	Jorge	León	53	2	36,55	74
13	Braulio	Zarabia	24	2	34,9	76
14	Jhon	Tene	45	2	34,99	76
15	Erica	Zurita	40	2	36,13	61
16	Viviana	Guamanquise	35	2	35,81	75
17	Cristian	Vasquez	30	2	35,72	81
18	Mayra	Carrillo	29	2	35,87	87
19	Carlos	Chamba	40	2	36,95	77

20	Miguel	Miranda	30	2	36,88	97
21	Fernanda	Pilamunga	17	2	35,78	78
22	Naybol	Machado	30	2	36,1	87
23	Marely	Cruz	20	2	37,87	82
24	viviana	Robalino	14	2	37,28	67
25	Jorge	Pazmiño	21	2	36,61	64
26	Erick	Villacres	15	2	35,54	80
27	Fernando	Marquez	21	2	35,63	63
28	Franklin	Lopez	27	2	36,58	89
29	Angel	Ortega	13	2	35,78	86
30	Fernando	Cudco	19	2	36,67	86
31	Freddy	Lopez	25	15	35,66	71
32	Tatiana	Chavez	27	15	36,1	70
33	Richard	Montesdeoca	16	15	35,84	103
34	Dario	Jordan	31	15	36,64	86
35	Ricardo	Aguilar	15	15	35,13	80
36	Cristian	Tarco	24	15	36,73	105
37	Daniel	Ripalda	21	15	35,25	81
38	Felipe	Valle	18	15	35,78	71
39	Katherine	Gavilanez	23	15	35,45	86
40	Isabel	Castillo	42	15	35,57	87
41	Mayra	Ortiz	21	15	39,03	75
42	Marllory	Cobos	32	15	36,5	82
43	Cristian	Jacho	18	15	35,04	64
44	Alvaro	Veintimilla	27	15	34,87	68
45	Ruben	Paucar	15	15	35,95	64
46	Cristian	Pazmiño	19	15	35,98	95
47	Juliana	Moya	23	15	36,61	73
48	Jose	Manobanda	30	15	37,59	78
49	Andrea	Montesdeoca	21	15	37,96	97
50	Lenin	Tierra	18	15	35,19	93
51	Andres	Solis	21	15	36,64	59
52	Jeferson	Iza	24	15	34,62	63
53	Edison	Borja	30	15	35,72	71
54	Santiago	Murillo	20	15	36,07	66
55	Marlon	España	29	15	36,16	82
56	Pamela	Barrazueta	29	15	35,95	73
57	Ivan	Armijo	23	15	36,13	73
58	David	Ojeda	30	15	36,1	60
59	Maria	Zambrano	40	15	35,07	53

60	Johana	Barreno	34	20	36,88	74
61	Pamela	Gusqui	38	20	35,36	52
62	Marilyn	Carrion	40	20	39,97	106
63	Gabriela	Rodriguez	43	20	37,4	55
64	Mercy	Loachamin	19	20	35,57	71
65	Walter	Crow	23	20	37,19	63
66	Jhon	Loaiza	16	20	35,45	94
67	Alex	Yautibuto	20	20	36,13	64
68	Liliana	Arias	27	20	35,6	74
69	Jonatan	Quezada	32	20	36,16	90
70	Karen	Paredes	38	20	36,88	74
71	Valeria	Sanchez	23	20	36,16	83
72	Cristian	Monar	21	20	35,69	78
73	Cristian	Viteri	20	20	36,43	70
74	Jairo	Arce	21	20	36,01	62
75	Edison	Bonilla	27	20	37,53	75
76	Andres	Carranza	21	20	36,95	58
77	Galo	Hurtado	13	20	35,72	73
78	Miriam	Vega	20	20	37,01	61
79	Paola	Obregon	21	20	35,42	87
80	Geovanna	Guerrero	35	20	35,84	68
81	Tatiana	Padilla	26	20	37,93	77
82	Geovanny	Cherrez	10	20	35,84	75
83	Cristian	Yugcha	15	20	35,48	75
84	Jessica	Proaño	13	20	36,65	81
85	Mario	Suarez	22	20	35,75	71
86	Liseth	Cevallos	15	20	35,07	76
87	Genova	Quispe	21	20	34,75	96
88	Alexandra	Parreño	19	20	37,96	80
89	Rebeca	Tenenaula	22	20	38,4	62
90	Jonatan	Leon	18	20	37,83	80

Fuente: Elaboración propia

➤ **TIEMPO DE RESPUESTA (2 METROS PROTOTIPO Y 1 METRO MANUAL)**

Planteamiento de la hipótesis

H_0 : El tiempo de respuesta es igual en el prototipo y el método manual.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H1: El tiempo de respuesta es diferente en el prototipo y el método manual.

$$H_1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XXXV. Varianza del tiempo de respuesta (2m prototipo y 1m manual)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
PROTOTIPO	30	795	26,5	135,086207
TRADICIONAL	30	4638	154,6	572,731034

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXXVI. Anova del tiempo de respuesta (2m prototipo y 1m manual)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	246144,15	1	246144,15	695,50199	5,4842E-34	4,00687289
Dentro de los grupos	20526,7	58	353,908621			
Total	266670,85	59				

Fuente: Elaboración propia

Gráfico

En la Figura 69 se muestra las mediciones realizadas a 30 personas con su respectivo tiempo, en donde va variando cada muestra con un tiempo distinto, con lo que se determina que nuestro prototipo es más rápido y eficiente.

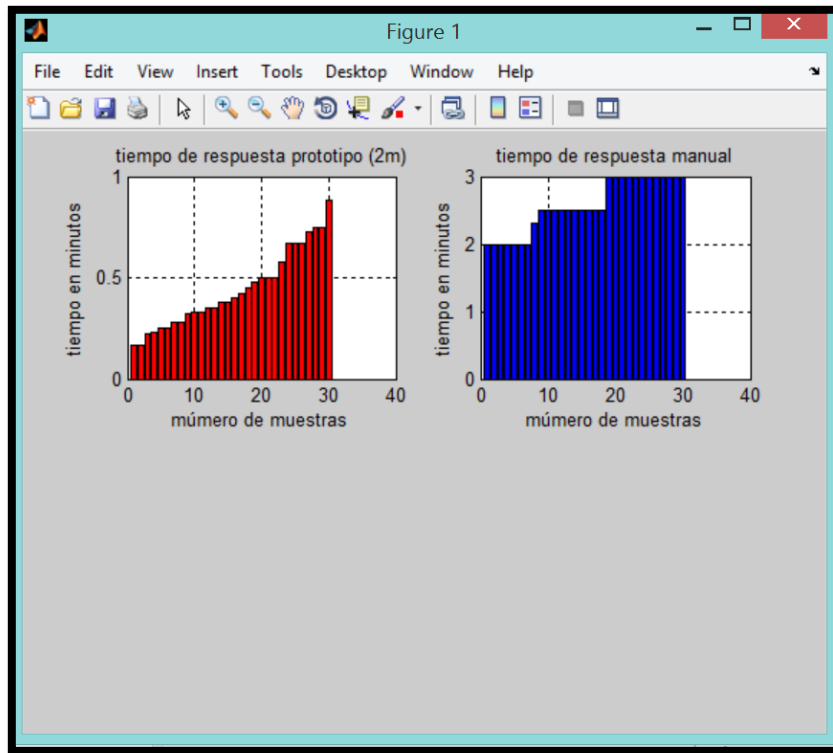


Figura 69: Tiempo de respuesta (2m prototipo y 1m manual)

Fuente: Elaboración propia.

Interpretación

Por medio de la obtención de la tabla ANOVA se sabe que la probabilidad es de $5,4842E-34$ y entonces definimos que $p < 0.05$; así que la hipótesis H_0 se rechaza. Por lo tanto se dice que el prototipo tiene un tiempo de respuesta menor que el método manual de la medición de los signos vitales.

➤ **TIEMPO DE RESPUESTA (15 metros PROTOTIPO Y 1 metro MANUAL)**

Planteamiento de la hipótesis

H_0 : El tiempo de respuesta es igual en el prototipo a 15 metros y el método manual.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H1: El tiempo de respuesta es diferente en el prototipo a 15 metros y el método manual.

$$H_1: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XXXVII. Varianza del tiempo de respuesta (15m prototipo y 1m manual)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
PROTOTIPO	30	746	24,8666667	46,9471264
TRADICIONAL	30	4800	160	579,310345

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXXVIII. Anova del tiempo de respuesta (15m prototipo y 1m manual)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	273915,267	1	273915,267	874,768859	1,1163E-36	4,00687289
Dentro de los grupos	18161,4667	58	313,128736			
Total	292076,733	59				

Fuente: Elaboración propia

Gráfico

En la Figura 70 se observa la variación del tiempo de respuesta del prototipo a una distancia mucho mayor a la anterior, con 15 metros de distancia entre el paciente y el receptor de datos.

Nuestro prototipo tiene un tiempo de respuesta menor.

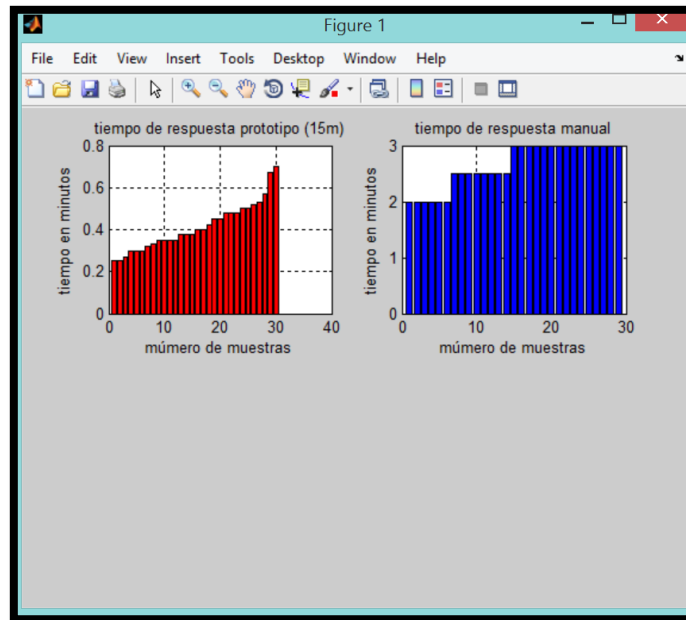


Figura 70: Tiempo de respuesta (15m prototipo y 1m manual)

Fuente: Elaboración propia.

Interpretación

Por medio de la obtención de la tabla ANOVA se sabe que la probabilidad es de 1,1163E-36 y entonces definimos que $p < 0.05$; así que la hipótesis H_0 se rechaza. Por lo tanto se dice que el prototipo tiene un tiempo de respuesta menor que el método manual al medir los signos vitales de un paciente a pesar de que la distancia aumente en la recepción el tiempo de respuesta del prototipo es mucho menor permitiendo usar este sistema en casos de emergencia.

➤ **TIEMPO DE RESPUESTA (20 METROS PROTOTIPO Y 1 METRO MANUAL)**

Planteamiento de la hipótesis

H_0 : El tiempo de respuesta es igual en el prototipo a 20 metros y el método manual.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H1: El tiempo de respuesta es diferente en el prototipo a 20 metros y el método manual.

$$H0: \mu_1 \neq \mu_2 \neq \mu_3$$

Calculo de la varianza

Tabla VII.XXXIX. Varianza del tiempo de respuesta (20m prototipo y 1m manual)

<i>Grupos</i>	<i>Cuenta</i>	<i>Suma</i>	<i>Promedio</i>	<i>Varianza</i>
PROTOTIPO	30	699	23,3	70,837931
TRADICIONAL	30	4410	147	642,413793

Fuente: Elaboración propia

Análisis de los Resultados (ANOVA)

Tabla VII.XXXX. Anova del tiempo de respuesta (20m prototipo y 1m manual)

ANÁLISIS DE VARIANZA						
<i>Origen de las variaciones</i>	<i>Suma de cuadrados</i>	<i>Grados de libertad</i>	<i>Promedio de los cuadrados</i>	<i>F</i>	<i>Probabilidad</i>	<i>Valor crítico para F</i>
Entre grupos	229525,35	1	229525,35	643,60265	4,3572E-33	4,00687289
Dentro de los grupos	20684,3	58	356,625862			
Total	250209,65	59				

Fuente: Elaboración propia

Gráfico

En la Figura 71 se observa la variación del tiempo de respuesta del prototipo a su máximo alcance de cobertura que son 20 metros de distancia entre el paciente y el receptor de datos y el método manual que no permite el desplazamiento ya que se debe obtener los datos desde el mismo punto donde se toman los datos fisiológicos. Se puede observar que el tiempo de respuesta sigue siendo mejor con el prototipo que de manera manual y se puede optimizar el tiempo de la toma de los signos vitales.

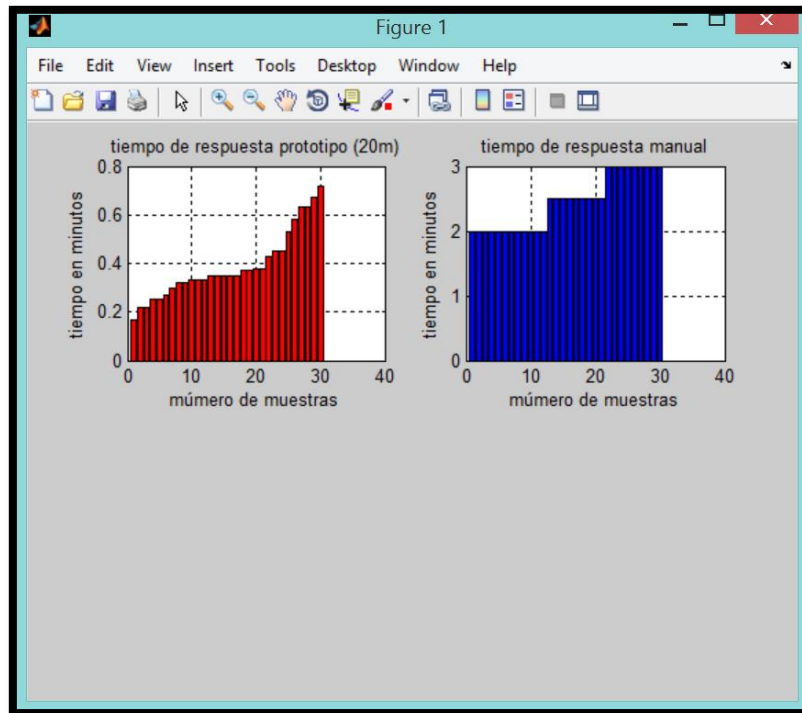


Figura 71: Tiempo de respuesta (20m prototipo y 1m manual)

Fuente: Elaboración propia.

Interpretación

Por medio de la obtención de la tabla ANOVA se sabe que la probabilidad es de $4,3572E-33$ y entonces definimos que $p < 0.05$; así que la hipótesis H_0 se rechaza. Por lo tanto se dice que el prototipo tiene un tiempo de respuesta menor que el método manual. En la medición de los datos fisiológicos de un paciente a pesar de que la distancia aumente para la recepción, siendo este sistema más eficiente y eficaz en la obtención de información.

7.3.3 Comprobación de la Hipótesis

Hipótesis de la Investigación

Es posible mediante el diseño e implementación de un prototipo de red de sensores inalámbricos para el monitoreo de los pacientes en el Centro de atención integral en salud de la ESPOCH mejorar el tiempo de respuesta de atención en casos de emergencia.

Proceso

Para la comprobación de la hipótesis se realiza una tabla de los promedios obtenidos del tiempo de respuesta a las diferentes distancias que se captó los datos tomados por el prototipo y de la manera manual a una misma distancia. Las tablas VII.XXXXI y VII.XXXII nos ayudan a visualizar mejor los resultados.

Tabla VII.XXXI. Valoración

VALOR	SIGNIFICADO
<input checked="" type="checkbox"/>	MAS RÁPIDO

Fuente: Elaboración propia

Tabla VII.XXXXII. Resultados

ESCENARIOS VARIABLES	PROTOTIPO	MÉTODO TRADICIONAL
TIEMPO DE RESPUESTA (segundos)	<input checked="" type="checkbox"/> 26,5 a 2 m	154,6 a 1 m
TIEMPO DE RESPUESTA (segundos)	<input checked="" type="checkbox"/> 24,866666x7 a 15 m	160 a 1 m
TIEMPO DE RESPUESTA (segundos)	<input checked="" type="checkbox"/> 23,3 a 20 m	147 a 1 m
PROMEDIO	24,889	153.867

Fuente: Elaboración propia

$$153.867 - 24.889 = 128.978$$

$$\text{Porcentaje de diferencia} = \frac{128,987}{153,867} = 0,8382 = 83,82\%$$

Después de visualizar los resultados obtenidos, se puede interpretar que el prototipo para la red de sensores inalámbricos permite brindar un tiempo de respuesta mucho menor siendo un 83.82% más rápido que el método tradicional con la que se trabaja en la actualidad en el Centro de Atención Integral en Salud de la ESPOCH.

Para la comprobación de la eficacia del prototipo se comparan los datos del pulso y la temperatura con las medidas tomadas manualmente.

Tabla VII.XXXIII. Promedios

	Pulso (promedio)	Temperatura (promedio)
Prototipo	76,1111111	37,025
Manual	74,3	36,0822222

Fuente: Elaboración propia

Porcentaje de pulso:

$$76,1111 - 74,3 = 1,8111$$

$$\text{Porcentaje de diferencia} = \frac{1,8111}{76,1111} = 0,0237 = 2,37\%$$

Porcentaje de temperatura:

$$37,025 - 36,0822 = 0,9428$$

$$\text{Porcentaje de diferencia} = \frac{0,9428}{36,0822} = 0,0255 = 2,55\%$$

En la Figura 72 y 73 se muestran los signos vitales tomados por el prototipo y la toma manual utilizada en el Centro de atención integral en salud de la ESPOCH, nuestro prototipo tiene un 2,37% de error en el pulso y 2,55 % de error en la temperatura demostrando que este sistema es fiable y eficiente.

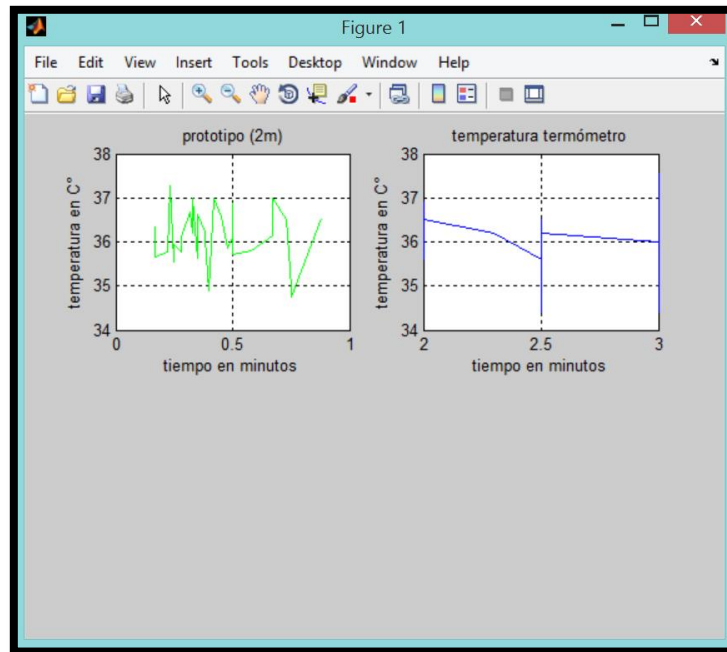


Figura 72: Temperatura (2m prototipo y 1m manual)

Fuente: Elaboración propia.

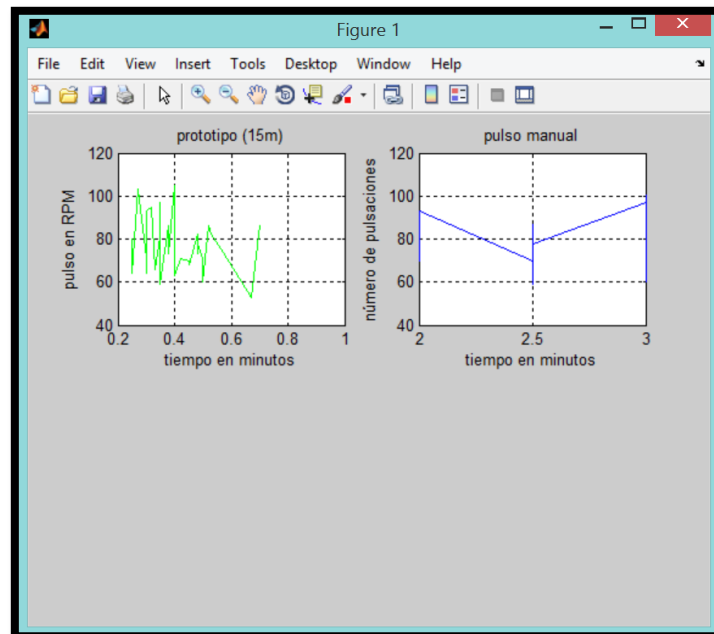


Figura 73: Pulso (15m prototipo y 1m manual)

Fuente: Elaboración propia.

CONCLUSIONES

Se utilizó la tecnología Zigbee que es la más usada en redes de área personal inalámbricas y en aplicaciones de telemedicina, además es el estándar bajo el cual transmiten los módulos XBee S1 usados.

Arduino Uno incorpora su propio software de configuración que es el IDE Arduino basado en C++ y es multiplataforma lo cual permite la posibilidad de interactuar con cualquier software para realizar una interfaz gráfica.

El Simulador de Redes NS2 admite redes cableadas o redes inalámbricas, pero no redes mixtas en un mismo escenario.

En las simulaciones realizadas en software NS2 se determinó que el mejor escenario fue el que incorpora el menor número de sensores, porque al aumentar sensores los paquetes se incrementan y producen una cola de espera en la transmisión, debido que se utiliza una comunicación tipo serie.

En la práctica los módulos XBee S1 alcanzan una cobertura de 20 m sin línea de vista, no de 30 m como se especifica en el datasheet.

Para brindar seguridad a nuestra red WSN se estableció una conexión punto a punto, además de encriptar los datos con AES.

El prototipo de WSN planteado en este tema de investigación para el monitoreo de pacientes brinda un tiempo de respuesta menor a la toma de signos vitales manual usado en el Centro de Atención Integral en Salud de la ESPOCH, logrando ser 80% más rápido que el método actual.

RECOMENDACIONES

Se debe tener en cuenta a cuantos baudios de comunicación se va a trabajar en el Arduino Uno y en los módulos XBee S1, ya que los dos dispositivos deben tener la misma velocidad de comunicación para obtener datos reales, por lo que se recomienda 9600 baudios para información confiable.

Para el diseño de la interfaz gráfica se recomienda Delphi XE7 porque dispone de muchas propiedades que facilitan el diseño, como le configura a través de bloques, además puerto ComPort para una conexión directa con el puerto serial.

El prototipo diseñado e implementado puede ser usado tanto en consulta externa como en pacientes hospitalizados en cualquier institución de Salud o incluso por personas particulares que necesitan tener un control de su estado de salud.

Cuando el área de trabajo es mayor a los 20 m que cubre XBee S1 sin línea de vista, se pueden usar otras variantes del XBee para alcanzar una mayor cobertura como el XBee PRO, etc.

TRABAJO FUTURO

El proyecto de investigación nos permite diferentes líneas futuras de investigación, una es que permita a través de sistemas operativos móviles controlar y monitorear el estado del paciente, permitiendo que los datos de texto se encuentren disponibles a través de su Smartphone.

El diseño y prototipo ofrece la posibilidad de acoplarse a diferentes redes con mayor alcance de cobertura como: WIFI, WIMAX, GSM, etc.

Además este tema de investigación es un sistema escalable debido a que permite añadir más sensores como para la presión, electrocardiogramas, etc. Que ayudan de una mejor manera el control y monitoreo de los pacientes.

BIBLIOGRAFÍA

- [1] **ANDREU, Joaquín,** Servicios en Red., Madrid-España., Ed. EDITEX., 2010., Pp. 214-218.
- [2] **ENRÍQUEZ, Andrés., y otros.,** Banda Ancha Inalámbrica., Madrid-España., Ed. OMNIASCIENCE., 2013., 170p.
- [3] **HUIDOBRO, José.,** Radiocomunicaciones viajando a través de las ondas., Madrid-España., Ed. COPYRIGHT., 2011., Pp. 150.
- [4] **CAPRILE, Sergio.,** Equisbí: Desarrollo de aplicaciones con comunicación remota basadas en módulos ZigBee y 802.15.4., Barcelona-España., Ed. UNIVERSITAT POLITECNICA CATALU., 2009., Pp. 77-80.
- [5] **CASADEMONT, Jordi.,** Redes de comunicaciones: de la telefonía móvil a Internet., Barcelona-España., Ed. UPC., 2010., Pp. 110-111.
- [6] **FERNÁNDEZ, R., y otros.,** Redes inalámbricas de sensores: teoría y aplicación práctica., New York-Estados Unidos., Ed. COPYRIGHT., 2013., Pp. 18-50.
- [7] **GISLASON, D.,** Zigbee Wireless Networking., Washington-Estados Unidos., Ed. Elsevier., 2008., Pp.400-413.
- [8] **MORREALE, P., TERPLAN, K.,** CRC Handbook of Modern Telecommunications., 2a ed., New Jersey-Estados Unidos., Ed. CRC PRESS., 2010., 320p.
- [9] **RASHVAND, H., ALCARAZ, J.,** Distributed Sensor System., Chennai-India., Ed. WILEY., 2012., Pp. 98-110.

[10] **MCROBERTS, M.**, Beginning ARDUINO, 2a ed., Madrid-España., Ed. TECHNOLOGY IN ACTION., 2013., Pp 2-4.

[11] **LI, Yingshu.**, Wireless Sensor Networks and Applications., New York-USA., Ed. SPRINGER., 2008., Pp. 8-11.

BIBLIOGRAFÍA INTERNET

[1] **REDES DE SENSORES INALÁMBRICAS**

<http://www.uv.es/~montanan/ampliacion/trabajos/Redes%20de%20Sensores.pdf>

2014-08-10

[2] **CONTRIBUCIONES AL ESTABLECIMIENTO DE UNA RED GLOBAL DE SENSORES INALÁMBRICOS INTERCONECTADOS.**

http://sedici.unlp.edu.ar/bitstream/handle/10915/4188/Documento_completo.pdf?sequence=1

2014-08-17

[3] **REPORTE: ¿QUE ES UNA RED DE SENSORES INALÁMBRICOS?**

<http://www.ni.com/white-paper/7142/es/>

2014-08-22

[4] **ESTADO ACTUAL DEL MONITOREO REMOTO DE PACIENTES USANDO REDES DE SENSORES INALÁMBRICAS.**

<http://web.b.ebscohost.com/ehost/pdfviewer/pdfviewer?sid=d808072c-7abb-492e-87d7-83931b7f95a3%40sessionmgr115&vid=2&hid=112>

2014-08-27

[5] ZIGBEE Y EL ESTÁNDAR IEEE 802.15.4

http://www.dea.icaupco.es/sadot/Comunicaciones/avanzadas/Alberto_Gasc%C3%B3n_Zigbee%20y%20el%20Est%C3%A1ndar%20IEEE%20802.15.4.pdf

2014-09-02

[6] COMENZANDO CON ZIGBEE.

<http://webdelcire.com/wordpress/archives/1714>

2014-09-05

[7] REDES DE SENSORES INALAMBRICOS APLICADO A LA MEDICINA

<http://repositorio.unican.es/xmlui/bitstream/handle/10902/1288/349251.pdf?sequence=1>

2014-09-10

[8] ZIGBEE

<http://sx-de-tx.wikispaces.com/ZIGBEE>

2014-09-13

[9] USOS PARA ZIGBEE

http://www.ehowenespanol.com/usos-zigbee-info_226594/

2014-09-18

[10] SQUIDBEE Y LAS REDES SENSORIALES

<http://www.microsiervos.com/archivo/tecnologia/squidbee-redes-sensoriales.html>

2014-09-23

[11] REDES DE SENSORES INALÁMBRICOS QUE CUIDAN NUESTRA SALUD Y MEDIO AMBIENTE

<http://www.lacofa.es/blog/2008/09/04/redes-de-sensores-inalambricos-que-cuidan-nuestra-salud-y-medio-ambiente/>

2014-10-15

[12] LIBELIUM, SPIN-OFF DE LA UZ, PIONERA EN TECNOLOGÍA DE REDES SENSORIALES

<http://www.aragoninvestiga.org/libelium-spin-off-de-la-uz-pionera-en-tecnologia-de-redes-sensoriales/>

2014-10-20

RESUMEN

Diseño e Implementación de un prototipo de Red de Sensores Inalámbricos para el monitoreo de los pacientes. Caso práctico: Centro de Atención Integral en Salud de la Escuela Superior Politécnica de Chimborazo.

Este trabajo busca mejorar el tiempo de respuesta en la toma de los signos vitales y envío de información al doctor para monitorear constantemente a los pacientes y usar nuestro sistema en casos de emergencia. En la construcción de este prototipo se utilizó el sensor infrarrojo que mide el pulso, oxigenación en la sangre, el sensor de flujo de aire mide la respiración, el sensor de temperatura mide la temperatura corporal; estos sensores se configuraron en la placa electrónica Arduino Uno. La transmisión inalámbrica de la información es a través de dos módulos XBee S1, el primero es transmisor y el segundo es receptor. En esta Red de Sensores Inalámbricos se realizó una conexión punto a punto con encriptación en los datos de los signos vitales para brindar información real. Para la visualización de los datos se configuró una interfaz gráfica en el software Delphi XE7, que además se conecta a una base de datos en MySQL para almacenar la información.

Para el análisis de resultados se utilizó el método estadístico completamente al Azar, definiendo que este prototipo: 80% más rápido que la toma de signos vitales manual utilizada en el Centro de Atención Integral en Salud de la Escuela Superior Politécnica de Chimborazo.

Concluimos que este proyecto de investigación ofrece una optimización de tiempo en la toma de los signos vitales en los pacientes atendidos en el Centro de Atención Integral en Salud, pudiendo usarse en casos de emergencia.

Recomendamos que el departamento de enfermería a cargo de la Lic. Lupe Zabala utilice este prototipo para brindar una atención más rápida y eficiente a los usuarios.

PALABRAS CLAVES: < ATENCIÓN INTEGRAL EN SALUD> < CONEXIÓN PUNTO A PUNTO> < MYSQL>
<RED DE SENSORES> <SIGNOS VITALES> <TIEMPO DE RESPUESTA> <TRANSMISIÓN INALÁMBRICA>

ABSTRACT

Design and implementation of the Wireless Sensor Network prototype for monitoring patients. Case study: Center for Integrated Health Care of the Polytechnic School of Chimborazo.

This research seeks to improve the response time in taking vital signs and sending information to the doctor to constantly monitoring patients and using our emergency system. An Infrared sensor was used in the construction of this prototype that a measure pulse, blood oxygenation, the air flow sensor measures the respiration, the temperature sensor measure body temperature; these sensors were configured on the electronic board Arduino One. The wireless transmission of information is through two modules: XBee S1; the first is the transmitter and the second is receiver. This Wireless Sensor Network was made with a point-to-point connection with encryption on data of the vital signs in order to provide real information. To display data a graphical interface in the XE7 Delphi software was configured, which further connects to a MySQL database for storing information.

For the analysis of results the statistical method was used completely random, defining this prototype: 80% faster than the manual takes vital signs used in the Center For Integrated Health Care of the Polytechnic School of Chimborazo.

We conclude that this research project provides an optimization of time in taking vital signs in patients treated at the Center for Integrated Health Care, and can be used in emergencies.

We recommend that the nursing department in charge of the Lic. Lupe Zavala uses this prototype to provide faster and more efficient users attention.

KEY WORDS: comprehensive health care, point to point connection, MYSQL, sensor network, vital signs, response time, wireless transmission

ANEXOS

Anexo 1 datasheet Arduino uno

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 μ A
 - Power-save Mode: 0.75 μ A (Including 32 kHz RTC)

Anexo 2 Datasheet Xbee s1

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

Anexo 3 Datasheet Shield XBee

Specifications

PCB size	54.9mm X 58.8mm X 1.6mm
Indicators	PWR State, DI, DO, RSSI, ON, ASS
Power supply	5V DC
Communication Protocol	UART/XBee
RoSH	Yes

Electrical Characteristics

Specification	Min	Type	Max	Unit
Power Voltage	4.5	5	5.5	VDC
Input Voltage VH:	4.5	5	5.5	V
Input Voltage VL:	-0.3	0	0.5	V
Current Consumption	-	20	40	mA

ANEXO 4 Instalación de NS2

Previamente antes de realizar la instalación de NS2 es necesario instalar lo siguiente:

1. Instalación de programas necesarios.
 - `sudo apt-get update`
 - `sudo apt-get install tcl8.5-dev tk8.5-dev`
 - `sudo apt-get install build-essential autoconf automake`
 - `sudo apt-get install perl xgraph libxt-dev libx11-dev libxmu-dev`
2. Descargar desde la página oficial el archivo fuente NS2.
 - <http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/download>
 - El archivo que se descarga es: "ns-allinone-2.35.tar.gz"
3. Descomprimir el archivo que se descargó en el paso 2 dentro de la carpeta "home"
 - `tar -zxvf ns-allinone-2.35.tar.gz -C/home/mary`
4. Instalación de NS2 se ubica dentro de la carpeta que la contiene.
 - `cd /home/mary/ns-allinone-2.35`
 - `sudo ./install`
5. Modificar el archivo `gedit /home/mary/.bashrc`
 - Dentro de este archivo al final se añade lo siguiente:


```
➤ PATH=$PATH:/home/stan/ns-allinone-2.35/bin:/home/mary/ns-allinone-2.35/tcl8.5.10/unix:/home/mary/ns-allinone-2.35/tk8.5.10/unix
```

```
➤ LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/mary/ns-allinone-2.35/otcl-1.14:/home/mary/ns-allinone-2.35/lib
```

```
➤ TCL_LIBRARY=$TCL_LIBRARY:/home/mary/ns-allinone-2.35/tcl8.5.10/library
```

- Activar el PATH ejecutando el comando: `source .bashrc`

6. Modificar el PATH para que se ejecute itm/tclsh/wish/xgraph

- `/home/mary/ns-allinone-2.35/bin:/home/mary/ns-allinone-2.35/tcl8.5.10/unix:/home/mary/ns-allinone-2.35/tk8.5.10/unix`

7. Instalar NS2 ejecutando el comando:

- `cd /home /mary /ns-allinone-2.35`
- `./install`

8. Ejecutar la validación dentro del archive `cd /home /mary /ns-allinone-2.35 /ns-2.35`

- `./validate`

9. Verificar si está instalado correctamente NS2 con el siguiente comando en el terminal:

- `ns`
- Saldrá un símbolo: `"%ns"`

ANEXO 5 DATASHEET SENSOR DE FLUJO DE AIRE

WARRANTY

Sleepmate Technologies warrants the sensor to be free of defects in materials and workmanship for a period of nine (9) months from the date purchased.

GENERAL SPECIFICATIONS (14800 SERIES)

Thermal Sensor

Lead wire:	Two conductors, approximately 200 cm
Terminators:	1.5 mm Safety pins (custom terminators available)
Sensor:	Flexible oral probe. Nasal probe inserts into custom cannula
Time constant:	One second or longer. The use of a shorter time constant may attenuate waveforms.
Filter Settings:	Low Frequency = 0.1 Hz; High Frequency = 15 Hz (1.5 Hz Thermistor)
Sensitivity:	Response is dependent on a variety of variable such as sensor application, patient effort, ambient room temperature, CPAP pressure, etc.

Cannula

Cannula:	Soft, clear and light for a comfortable patient fit
Patient Attachment:	Over-the-ear, "Bolo-Style" adjustable slide
Length:	7ft (213.4cm) kink resistant tubing
Terminator:	Male luer connector

ORDERING INFORMATION

Ambu Reference Number	Product Description	Sleepmate former Number
814310-100	CannuTherm Kit (Thermocouple Sensor & 5 Nasal Cannulas)	1419
814840-100	CannuTherm - Reusable Thermocouple Sensor Only	1418
814930-100	CannuTherm Thermistor Kit (Thermistor Sensor + 5 Nasal Cannulas w/filters)	1420
814960-100	CannuTherm - Reusable Thermistor Sensor only	1458
814360-100	AASM CannuTherm Flow Kit (AC AirFlow Pressure, Reusable Thermocouple Sensor, 10 CannuTherm Cannulas)	1431
814320-119	CannuTherm Alice 3 Thermistor - Sensor Only	1463
814310-121	CannuTherm Alice 5 Thermistor - Sensor Only	1464
814370-100	AASM CannuTherm Flow Kit (DC AirFlow Pressure, Reusable Thermocouple Sensor, 10 CannuTherm Cannulas)	1432
814930-119	Alice 3 CannuTherm Kit (Alice 3 Compatible Thermistor, Thermistor Box & 5 Nasal Cannulas w/filters)	1424
814930-121	Alice 5 CannuTherm Kit (Alice 5 Compatible Thermistor & 5 Nasal Cannulas w/filters)	1425
Disposable		
814360-000	CannuTherm Cannula with filter	14877
814360-000/25	CannuTherm Cannulas with filters (case of 25)	14878

Ambu Inc.

6740 Baymeadow Drive
Glen Burnie, MD 21060
Tel.: 1 410 768 6464
1 800 262 8462
Fax. 1 800 262 8673
www.ambuUSA.com

ANEXO 6 DATASHEET SENSOR DE TEMPERATURA CORPORAL

ABSOLUTE MAXIMUM RATINGS*

Voltage on V_{DD}	GND -0.3V to +6.5V
Output Current	5.0mA
Operating Temperature Range	-40°C to +125°C
Storage Temperature Range	-55°C to +150°C
ESD Susceptibility (Human Body Model)	2kV
Soldering Temperature	215°C for 60s (Vapor Phase)
220°C for 15s (IR)	

* These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

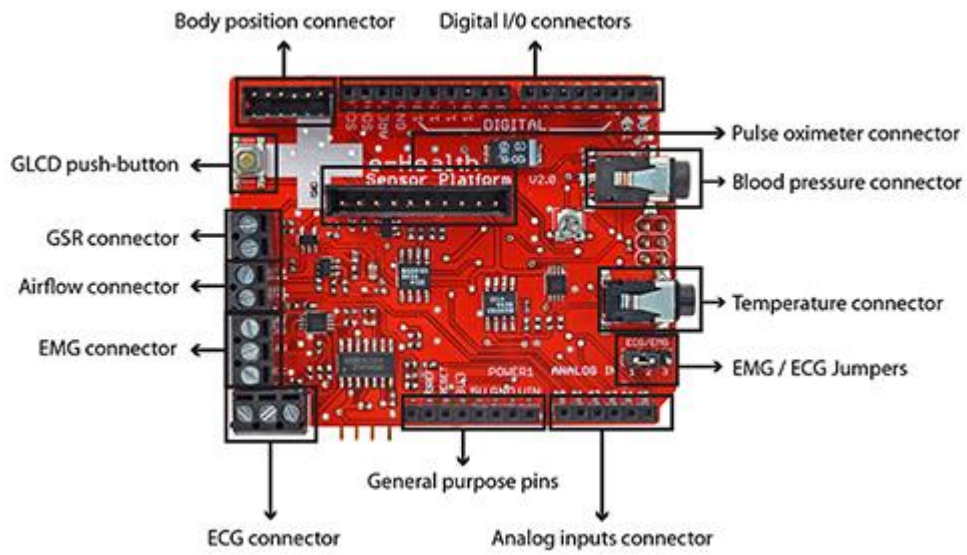
The Dallas Semiconductor DS60 is built to the highest quality standards and manufactured for long-term reliability. All Dallas Semiconductor devices are made using the same quality materials and manufacturing methods. However, the DS60 is not exposed to environmental stresses, such as burn-in, that some industrial applications require.

DC ELECTRICAL CHARACTERISTICS (-40°C to +125°C; $2.7V \leq V_{DD} \leq 5.5V$)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}		2.7		5.5	V	1
Supply Current	I_{DD}			80	125	μA	2
Measurement Error	T_{ERR}	$0^{\circ}C \leq T_A \leq 85^{\circ}C$			± 2	$^{\circ}C$	2
		$-40^{\circ}C \leq T_A \leq 125^{\circ}C$			± 3		
V_O DC Offset		$T = 0^{\circ}C$		424		mV	1, 2
Sensor Gain	$\Delta V/\Delta T$		6.0	6.25	6.5	mV/ $^{\circ}C$	2
Nonlinearity					± 0.8	$^{\circ}C$	2, 3
Power Supply Regulation		$2.7V \leq V_{DD} \leq 3.3V$			± 2.0	mV/V	
		$3.0V \leq V_{DD} \leq 5.5V$			± 0.25	mV/V	
Sensor Drift				± 0.25		$^{\circ}C$	4
Output Impedance					800	Ω	

Anexo 7 Pulse Oximetry Sensor

Anexo 8 e-health sensor shield



Anexo 9 Antena whip



WH2400 Whip Antenna Antenna for Use with 2.4GHz ELPRO Products



Description

The WH2400-SMA Whip Antenna is a 1/4 wave antenna suitable for ELPRO 2.4GHz product range. The antenna is vertically polarized with an omnidirectional radiation pattern.

These antennas are mounted directly onto the ELPRO product and are intended for demonstration use or short range communications only.

Specifications	
Parameters	
Frequency	2380 - 2520MHz
Gain	-2dBi
Bandwidth	140MHz of Center Frequency
VSWR	< 1.5:1
Polarization	Vertical
H Plane	Omnidirectional
Termination	
Termination	SMA Male
General	
Size	54mm (2.1")
Housing	PVC/wire
Mounting	SMA Male Connector Base to Module
Weight	0.1kg (0.22lbs)
Wind Loading	N/A
Note: Specifications subject to change.	

Ordering

To order, select product code from the table and specify country of application.

Product Code	Description	Frequency
WH2400-SMA	Whip Antenna - 54mm (2.1"), SMA Male, -2dBi Gain	2380 - 2520MHz