



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS

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

**SISTEMA INALÁMBRICO DE MONITOREO DE TEMPERATURA
CORPORAL PARA PACIENTES DE TERAPIA INTENSIVA DEL
HOSPITAL “SAN VICENTE DE PAÚL” DE IBARRA**

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA
Y REDES DE COMUNICACIÓN**

AUTOR: TREJO PAGUAY LUIS ALEXANDER

DIRECTOR: GAVILANEZ VILLALOBOS MILTON, ING.

IBARRA-ECUADOR

ENERO 2013



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

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

1. IDENTIFICACIÓN DE LA OBRA

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

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

DATOS DEL CONTACTO	
Cédula de Identidad	1003012059
Apellidos y Nombres	Trejo Paguay Luis Alexander.
Dirección	Avenida Colon y Versailles.
Email	alexandertrej@gmail.com
Teléfono Fijo	062650907
Teléfono Móvil	0981488997

DATOS DE LA OBRA	
Título	SISTEMA INALÁMBRICO DE MONITOREO DE TEMPERATURA CORPORAL PARA PACIENTES DE TERAPIA INTENSIVA DEL HOSPITAL "SAN VICENTE DE PAÚL" DE IBARRA
Autor	Trejo Paguay Luis Alexander
Fecha	20 de Enero de 2013
Programa	Pregrado <input checked="" type="checkbox"/> Posgrado. <input type="checkbox"/>
Título por el que se aspira	Ingeniero en Electrónica y Redes de Comunicación
Director	Gavilánez Villalobos Milton Alejandro, Ing.

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD.

Yo, Luis Alexander Trejo Paguay, con cédula de identidad Nro. 1003012059, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad de material y como apoyo a la educación, investigación y extensión, en concordancia con la ley de Educación Superior Artículo 143.

3. CONSTANCIA

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

Ibarra, a los 20 días del mes de Enero del 2013

El Autor:

.....

Nombre: Alexander Trejo P.

C.I. 100301205-9

Aceptación

.....

Nombre:.....

JEFE DE BIBLIOTECA

Facultado por resolución de Consejo Universitario.....



UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, **Luis Alexander Trejo Paguay**, con cédula de identidad Nro. 1003012059, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, Artículos 4,5 y 6, en calidad de autor del trabajo de grado denominado. **SISTEMA INALÁMBRICO DE MONITOREO DE TEMPERATURA CORPORAL PARA PACIENTES DE TERAPIA INTENSIVA DEL HOSPITAL “SAN VICENTE DE PAÚL” DE IBARRA**, que ha sido desarrollado para optar por el título de: **Ingeniero en Electrónica y Redes de Comunicación**, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Ibarra, a los 20 días del mes de Enero de 2013

Firma

Nombre: Trejo Paguay Luis Alexander.

Cédula: 1003012059.

DECLARACIÓN

Yo, Luis Alexander Trejo Paguay, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; y que este no ha sido previamente presentado para ningún grado o calificación personal.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondiente al presente trabajo a la Universidad Técnica del Norte, según lo establecido en las leyes de propiedad intelectual, reglamentos y normas vigentes en la universidad.

Alexander Trejo

C.I. 100301205-9

AUTOR

CERTIFICACIÓN

INGENIERO MILTON GAVILANEZ VILLALOBOS, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN

CERTIFICA:

Que, el presente Trabajo de Titulación “Sistema Inalámbrico de Monitoreo de Temperatura Corporal para Pacientes de Terapia Intensiva del Hospital San Vicente de Paúlde Ibarra”, ha sido desarrollado por el señor Luis Alexander Trejo Paguay bajo mi Supervisión.

Ing. Milton Gavilánez V.

DIRECTOR

CERTIFICADO

Una vez revisado el CD, con el trabajo de grado del Egresado: **Trejo Paguay Luis Alexander**, con el tema del proyecto de titulación: **“SISTEMA INALÁMBRICO DE MONITOREO DE TEMPERATURA CORPORAL PARA PACIENTES DE TERAPIA INTENSIVA DEL HOSPITAL “SAN VICENTE DE PAÚL” DE IBARRA”**. El CD funciona en su totalidad.

Contenido del CD:

- Documento del Proyecto
 - Parte teórica
 - Parte experimental.
 - Anexos

- Instaladores: CCs, netbeans-6.9.1-ml-windows, jdk-6u16-windows-i586, appserv-win32-2.5.10
- Driver COM FTDI
- Base de Datos App_medico, App_recibe_datos;
- Programas: Flex_lcd, Transmisor.

Atentamente:

Ing. Edgar Maya

Ing. Nancy Cervantes

Ing. Edison Jácome

MIEMBROS DEL TRIBUNAL

DEDICATORIA

El presente Trabajo va dedicado para mi redentor, que con infinita sabiduría y benevolencia me dio la fuerza de lograr tan favorable sueño. Faltándome poco para fracasar, y consumirme en la mediocridad apretó fuertes mis brazos sin dejarme caer, diciendo suavemente al oído que las consecuencias de mis actos son la suma de un resultado que puede variar con la persistencia y fe de mi corazón.

De igual forma dedico este trabajo a mis más grandes regalos, mi hijo Alejandro Jesse y su madre que con su lucha y constancia diaria, ilumina cada día de mi vida mostrándome que ser parte de su vida es el más bello privilegio que puedo tener; sirviendo como resultado un ejemplo que mostrar para su futuro cercano.

Sería por demás dejar a fuera a mis padres y hermanas que con su sacrificio y valentía depositaron no solo su confianza en mí, si no sus anhelos e ilusiones para sentirse dichosos y orgullosos de ver en el hombre que cada día me convierto.

Y por último este trabajo lo dedico para todos aquellos que no tienen voz como algún día no la tuve; para que tengan la

valentía de luchar por lo única y cierta libertad que tiene un hombre sus sueños; valor inagotable que despierta los más profundos deseos; llevando al camino de excelencia.

Alexander.

AGRADECIMIENTO

Un humilde y más sincero agradecimiento, a mi Director de Tesis que tuvo la paciencia y voluntad de sostenerme, ayudarme y guiarme en este caminar; logrando cimentar en mí la vocación profesional y el ferviente deseo de continuar cosechando éxitos junto a personas que como él sirven de luz y ejemplo.

Mi agradecimiento al que se convirtió en mi segundo Hogar , la Universidad Técnica del Norte que no solo prepara profesionales, sino entes productivos en materia gris y personalidad; plantando retos y valores que más adelante permiten ayudar a seres humanos que entiende que la educación no solo es un requisito, sino es una forma de vida que engrandece la sociedad.

De la misma forma agradezco a cada una de las personas que pusieron un granito de arena para que este sueño sea hoy mi realidad.

Alexander.

INDICE

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	II
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .IV	
DECLARACIÓN.....	V
CERTIFICACIÓN	VI
CERTIFICADO	VII
DEDICATORIA	VIII
AGRADECIMIENTO.....	X
INDICE	XI
RESUMEN.	XXIV
PRESENTACIÓN	XXVI
CAPITULO 1.....	1
1. FUNDAMENTACION TEORICA	1
1.1. EL MUNDO DE LAS REDES INALMBRICAS.....	1
1.2. WLAN.....	2
1.2.1 ARQUITECTURA DE LA IEEE 802.11.....	3
1.2.2 TOPOLOGÍA DE UNA LAN INALAMBRICA	4
1.2.3 ESPECIFICACIONES DE UNA WLAN.....	5
1.2.3.1.....	6
1.2.3.2 802.11b.....	6
1.2.3.3 802.11c.....	6
1.2.3.4 802.11d.....	6
1.2.3.5 802.11e.....	7
1.2.3.6 802.11f.....	7
1.2.3.7 802.11g.....	7
1.2.3.8 802.11h.....	7
1.2.3.9 802.11i.....	8
1.2.3.10 802.11k.....	8
1.2.3.11 802.11m.....	8
1.2.3.12 802.11n.....	8

1.3.	WPAN	9
1.3.1	<i>BLUETOOTH</i>	10
1.3.1.1	Topología De Red	10
1.3.1.2	Especificaciones Del Estándar Bluetooth	11
1.3.2	<i>WIMEDIA</i>	121.3.2.1
	<i>Topológica.12</i>	
1.3.3	<i>ZIGBEE</i>	12
1.3.3.1	Topología en una Red ZIGBEE.	13
1.3.4	<i>IRDA</i>	14
1.3.4.1	Topología IRDA.....	14
1.3.5	<i>RFID</i>	14
1.3.5.1	Arquitectura RFID.....	15
1.3.5.2	Bandas de Frecuencia.	16
1.4.	EL USO DE LAS REDES INALÁMBRICAS EN LA MEDICINA.	16
1.4.1	<i>IMPLANTE DE MOTORIZACION INALAMBRICO EN EL CORAZON</i>	16
1.4.2	<i>SISTEMA DE RADIO TELEMETRÍA DE TEMPERATURA EN LA PIEL</i>	17
1.4.3	<i>SISTEMA PRTATIL DE TELEMONITOREO REMOTO DEL PACIENTE</i>	18
1.4.4	<i>TIRA ELECTRÓNICA QUE MIDE LA TEMPERATURA CORPORAL Y EL RITMO CARDIACO SIN ALIMENTACIÓN ELECTRÓNICA</i>	19
	CAPITULO II	21
	DISEÑO DE LA RED	21
2.1	REQUERIMIENTOS PARA EL DISEÑO DE MONITORIZACIÓN DE LA RED DE TEMPERATURA CORPORAL.....	21
2.1.1	<i>ANÁLISIS DE LOS PARÁMETROS DE LA ZONA DE TERAPIA INTENSIVA</i>	21
2.1.2	<i>OBJETIVO DE LA RED</i>	22
2.1.3	<i>SERVICIOS DE LA RED</i>	22
2.1.3.1	Movilidad	22
2.1.3.2	Número de usuarios.....	23
2.1.3.3	Transmisión sin línea de vista.	23
2.1.3.4	Velocidad.	24
2.1.3.5	Disponibilidad	24
2.1.4	<i>TIPO DE RED A UTILIZARSE EN LA ZONA DE TERAPIA INTENSIVA</i>	24
2.1.4.1	Red Mesh	24
2.1.4.2	Mobile Ad Hoc Network.....	25
2.1.4.3	Red De Sensores	26
2.1.4.3.1	<i>Características de una Red de Sensores</i>	26
2.1.4.4	Red a implementarse.	28

2.1.5	<i>DISPOSITIVO DE TRANSMISIÓN INALÁMBRICA</i>	28
2.1.5.1	Dispositivos Zigbee	30
2.1.5.2	Los Equipos Zigbee En Una Red De Sensores.....	31
2.1.5.2.1	El Coordinador.....	31
2.1.5.2.2	Los Routers.....	32
2.1.5.2.3	Dispositivo Final.....	32
2.1.5.3	Características De Los Dispositivos Xbee.....	32
2.1.5.3.1	Circuito básico para el Xbee.....	32
2.1.5.4	Modos de Operación.....	33
2.1.5.4.1	<i>Modo Recepción/ Transmisión</i>	34
2.1.5.4.2	<i>Sleep Mode</i>	34
2.1.5.4.3	<i>Modo de Comando</i>	34
2.1.5.4.4	<i>Modo Idle</i>	35
2.1.5.5	DIRECCIONAMIENTO Y TOPOLOGÍA DE RED EN XBEE.....	35
2.1.5.5.1	<i>Direccionamiento de 16 bits</i>	35
2.1.5.5.2	<i>Direccionamiento de 64 bits</i>	35
2.1.5.5.3	<i>Topología Punto a Punto</i>	36
2.1.5.5.4	<i>Topología Punto a Multipunto</i>	37
2.1.5.6	Arquitectura ZigBee.....	39
2.2	MODULO INFORMATICO DEL SISTEMA MEDICO.....	40
2.2.1	<i>ARQUITECTURA DEL SISTEMA</i>	40
2.2.1.1	<i>Cliente – Servidor</i>	40
2.2.1.2	BASE DE DATOS MYSQL 5.....	42
2.2.1.2.1	<i>Historia</i>	42
2.2.1.2.2	Versiones.....	43
2.2.1.3	<i>Modelo Entidad – Relación</i>	44
2.2.1.4	Diseño de la Base de Datos.....	44
2.2.1.4.1	Diccionario de la Base de Datos.....	44
2.2.1.4.2	Aplicación.....	47
2.2.1.5	Ide de programación Netbeans 6.9.1.....	47
2.2.1.6	La Plataforma NetBeans.....	47
2.2.1.7	Lenguaje de Programación JAVA.....	48
2.2.1.7.1	<i>Filosofía</i>	48
2.2.1.7.2	<i>Orientado a Objetos</i>	48
2.2.1.7.3	<i>Plataforma</i>	49
2.2.1.7.4	<i>JAVA en Aplicaciones de Escritorio</i>	49
2.3	MEDICIÓN DE TEMPERATURA CORPORAL.....	49
2.3.1	<i>TEMPERATURA CORPORAL</i>	49
2.3.2	<i>ALTERACIONES DE DE LA TEMPERATURA CORPORAL</i>	50

2.3.2.1	Hipotermia	50
2.3.2.2	Fiebre	50
2.3.3	<i>FORMAS DE MEDIR LA TEMPERATURA DEL CUERPO HUMANO.....</i>	<i>51</i>
2.3.3.1	<i>Temperatura axilar.</i>	<i>51</i>
2.3.3.2	Temperatura Rectal.	52
2.3.3.3	Temperatura Oral.	52
2.3.3.4	Temperatura en el Oído.....	53
2.3.4	<i>SITUACIONES QUE PERJUDICAN LA TOMA DE LA TEMPERATURA CORPORAL</i>	<i>53</i>
2.4	DISEÑO DEL MÓDULO ELECTRÓNICO DE MEDICIÓN DE TEMPERATURA CORPORAL	53
CAPITULO III.....		54
3 DESARROLLO DE LA APLICACIÓN		54
3.1	TIPOS DE SISTEMA OPERATIVOS UTILIZADOS PARA EL DISEÑO DE MONITORIZACIÓN DE TEMPERATURA CORPORAL.....	54
3.1.1	<i>SOFTWARE LIBRE.</i>	<i>54</i>
3.1.2	<i>SOFTWARE CON LICENCIA.</i>	<i>56</i>
3.1.2.1	Adobe.....	57
3.1.2.2	Aspel	57
3.1.2.3	Athenea.....	57
3.1.2.4	Avg	57
3.1.2.5	Hob Rd Vpn	57
3.1.2.6	Imlock.....	57
3.1.2.7	Ingressio.....	58
3.1.2.8	Lojack.....	58
3.1.2.9	Microsoft.....	58
3.1.2.10	Symantec	58
3.1.2.11	Versadial	58
3.1.2.12	Vmware.....	58
3.1.3	<i>SOFTWARE DE CONTROL INDUSTRIAL</i>	<i>58</i>
3.1.3.1	Labview	59
3.1.3.1.1	Aplicaciones de LabVIEW	59
3.1.3.1.2	ExpeMed	59
3.1.4	<i>REQUERIMIENTOS Y ESPECIFICACIONES TÉCNICAS DEL MÓDULO ELECTRÓNICO DE MEDICIÓN DE TEMPERATURA CORPORAL</i>	<i>60</i>
3.1.4.1	Sensor de temperatura.....	61
3.1.4.1.1	RTD	61
3.1.4.2.2	Sensor de Piel Ellab.....	68
3.1.4.2.3	Sensor de Temperatura recusable.....	69

3.1.4.2.4	Sensor de Piel EXACON SCIENTIFIC.....	70
3.1.4.2.5	Termistor MA100GG103A NTC.....	70
3.1.4.3	Amplificador Operacional De Voltaje.....	71
3.1.4.3.1	LM 324.....	72
3.1.4.3.2	INA122.....	72
3.1.4.4	Microcontroladores.....	73
3.1.4.5	Comunicación Serial.....	74
3.1.5	ESTRUCTURA DEL TRANSMISOR.....	76
3.1.5.1	Selección del sensor de temperatura.....	77
3.1.5.1.1	Circuito amplificador de temperatura corporal.....	78
3.1.5.1.2	Análisis de Circuito amplificador de temperatura corporal.....	79
3.1.5.2	Configuración del Amplificador operacional.....	79
3.1.5.2.1	Cálculo de ganancias de 3 voltajes para dar mayor fialidad a los datos.....	81
3.1.5.2.2	Cálculo del voltaje Vo de salida.....	81
3.1.5.3	Gráfica de temperatura Vs resistencia.....	82
3.1.5.3.1	Análisis de la temperatura y elección de las resistencias.....	82
3.1.5.4	Cálculos de Calibración.....	84
3.1.5.5	Calculo de la ecuación de la recta.....	86
3.1.5.6	Configuración de los dispositivos XBEE para la transmisión de los datos.....	87
3.1.5.6.1	Receptor.....	89
3.1.5.6.2	Transmisor:.....	89
3.1.5.7	Microcontrolador 18F2550.....	90
3.1.5.8	Simulación Del Programa.....	92
3.1.5.9	Visualización de los datos.....	93
3.1.5.9.1	Modulo LCD 2 x 8.....	93
3.1.5.9.2	Diagrama técnico de una pantalla Lcd.....	94
3.1.5.9.3	Especificaciones del módulo LCD 2x8.....	94
3.1.5.9.4	Grafica de la pantalla de la LCD.....	95
3.1.5.9.5	Comandos de trabajo en una LCD.....	96
3.1.5.10	Etapas de regulación de voltaje y máxima corriente consumida por el circuito.....	101
3.1.5.10.1	Selección y gráficas de baterías seleccionadas.....	101
3.1.5.10.2	Baterías para teléfono móvil Nokia serie 5300 n80.....	102
3.1.6	Sistema de Recepción de Datos.....	103
3.1.6.1	Circuito receptor.....	103
3.1.6.2	Estructura de la aplicación.....	104
3.1.6.2.1	Aplicación de Recepción de Datos:.....	104
3.1.6.3	Proceso para Ingresar temperaturas a la BDD.-.....	105
3.1.6.4	Aplicación Médica:.....	105

CAPITULO IV:	109
4. PRUEBAS EXPERIMENTALES	109
4.1. SISTEMA DE MEDICION DE TEMPERATURA CORPORAL	109
4.1.1 <i>Características de transmisor de temperatura corporal</i>	110
4.1.2 <i>Características del Receptor de temperatura de temperatura corporal</i>	111
4.1.3 <i>Características de la Base de Datos</i>	111
4.1.4 <i>Lógica del funcionamiento del Circuito</i>	112
4.1.5 <i>Consumo de Corriente</i>	112
4.1.6 <i>Duración de Baterías</i>	113
4.2. TERMOMETRO DE MERCURIO.	114
4.2.1 <i>Características de un termómetro de Mercurio</i>	115
4.2.1.1 Comparación de medidas.	115
4.3. TERMÓMETRO DIGITAL FLEXIBLE	116
4.3.1 <i>Características de un termómetro Digital</i>	116
4.3.2 <i>Comparación de medidas</i>	116
4.4. MULTÍMETRO FLUKE 179.	117
4.4.1 <i>Comparación de medidas</i>	118
4.5. TABLA DE LAS DIFERENTES MUESTRAS TOMADAS.....	119
4.6. ALCANCE DE LOS DISPOSITIVOS INALÁMBRICOS.	119
4.7. CUADRO DE COSTO DEL SISTEMA DE MONITORIZACIÓN DE TEMPERATURA CORPORAL.....	120
4.8. SISTEMA DE ALERTA DE TEMPERATURA.	122
CAPITULO V:	123
5. CONCLUSIONES Y RECOMENDACIONES	123
REFERENCIAS BIBLIOGRAFICAS	125

ANEXOS

INDICE DE FIGURAS

FIGURA 1. COMPONENTES DE UNA RED IEEE 802.11.	3
FIGURA 2. CONFIGURACIÓN DE UNA RED AD HOC	4
FIGURA 3. TOPOLOGÍA CON PUNTO DE ACCESO EN UNA LAN.	5
FIGURA 4.AMPLIACIÓN DE COBERTURA MEDIANTE ANTENAS DIRECTIVAS.	5

FIGURA 5. APLICACIONES DE LA WPAN.....	9
FIGURA 6. CONEXIÓN PUNTO A PUNTO.....	11
FIGURA 7. CONEXIÓN PUNTO A MULTIPUNTO.	11
FIGURA 8. TOPOLOGÍAS ZIGBEE.	13
FIGURA 9. TOPOLOGÍAS IRDA.	14
FIGURA 10. ARQUITECTURA RFID.	15
FIGURA 11. RECEPTOR DE TEMPERATURA CORPORAL. SISTEMA DE MONITORIZACIÓN DE TEMPERATURA CORPORAL.....	17
FIGURA 12. TRANSMISOR DE TEMPERATURA CORPORAL. SISTEMA DE MONITORIZACIÓN DE TEMPERATURA CORPORAL.....	18
FIGURA 13.SISTEMA DE MONITORIZACIÓN DE TEMPERATURA CORPORAL	19
FIGURA 14. SISTEMA DE MONITORIZACIÓN CON RFID.....	20
FIGURA 15. MONITORIZACIÓN DE VARIABLES FISIOLÓGICAS.....	23
FIGURA 16. REPRESENTACIÓN DE UNA RED MESH EN UN ÁREA URBANA.	25
FIGURA 17. REPRESENTACIÓN DE UNA RED AD HOC EN UN DESPLIEGUE MILITAR. L.....	26
FIGURA 18. RED DE SENSORES EN LA MONITORIZACIÓN UN VOLCÁN.	27
FIGURA 19. RED MULTIPUNTO – PUNTO:.....	28
FIGURA 20. DISPOSITIVO XBEE SERIES 2:	30
FIGURA 21. RED ZIGBEE.	31
FIGURA 22. ALIMENTACIÓN DE UN DISPOSITIVO XBEE.....	33
FIGURA 23.MODOS DE OPERACIÓN DE UN XBEE	33
FIGURA 24. ALIMENTACIÓN DE UN DISPOSITIVO XBEE.....	36
FIGURA 25. ALIMENTACIÓN DE UN DISPOSITIVO XBEE.....	37
FIGURA 26. ALIMENTACIÓN DE UN DISPOSITIVO XBEE.	38
FIGURA 27. ALIMENTACIÓN DE UN DISPOSITIVO XBEE.....	39
FIGURA 28.. ARQUITECTURA DE LA TECNOLOGÍA ZIGBEE.	39
FIGURA 29. INTERFAZ SERVIDOR CLIENTE.....	41

FIGURA 30. VERSIONES MYSQL.....	43
FIGURA 31. ESTRUCTURA DE BASE DE DATOS.....	44
FIGURA 32. INTERACCIÓN BASE DE DATOS USUARIO.....	47
FIGURA 33TEMPERATURA AXILAR.	51
FIGURA 34. TEMPERATURA RECTAL.....	52
FIGURA 35. TEMPERATURA ORAL	52
FIGURA 36. MARCAS DE PROGRAMAS CON LICENCIA.....	56
FIGURA 37. MARCAS DE PROGRAMA DE CONTROL INDUSTRIAL.	59
FIGURA 38. ETAPAS DEL DISEÑO DEL TRANSMISOR DE TEMPERATURA.....	60
FIGURA 39. RTD MEDIDOR DE TEMPERATURA.....	62
FIGURA 40. TERMISTORES PARA MEDIR DE TEMPERATURA.	64
FIGURA 41. LM35 SENSOR MEDIDOR DE TEMPERATURA.....	64
FIGURA 42. TERMOPILA MLX90615.....	65
FIGURA 44. SENSOR DE PIEL ELLAB.	68
FIGURA 45. SENSOR DE PIEL REUSABLE.....	69
FIGURA 46 SENSOR DE PIEL.....	70
FIGURA 47. TERMISTOR MA100GG103A NTC.....	70
FIGURA 48. TERMOPILA INFRARROJA.	71
FIGURA 49. CONECTORES DB9.	75
FIGURA 50. TRANSMISOR DE TEMPERATURA CORPORAL.....	76
FIGURA 52. TERMISTOR DE TEMPERATURA CORPORAL.....	77
FIGURA 53. DIAGRAMA DEL AMPLIFICADOR DE TEMPERATURA CORPORAL.	78
FIGURA 54. AMPLIFICADOR EN MODO NO INVERSOR.....	80
FIGURA 55. GRÁFICA DEL CAMBIO DE LA TEMPERATURA VS LA RESISTENCIA RTD	82
FIGURA 56. ACONDICIONAMIENTO DEL TERMISTOR	84
FIGURA 57. CALIBRACIÓN Y AMPLIFICACIÓN DE LOS DATOSANÁLOGOS DE TEMPERATURA.	85
FIGURA 58. INTERFAZ X-CTU PARA CONFIGURACIÓN DE MÓDULOS XBEE.	88

FIGURA 59. CONFIGURACIÓN ELECTRÓNICA DEL MÓDULO XBEE.....	90
FIGURA 60. PINES DEL MICROCONTROLADOR 18F2550.	91
FIGURA 61. SIMULACIÓN UTILIZANDO PROTEUS	92
FIGURA 62. DIAGRAMA ELECTRÓNICO DEL MICROCONTROLADOR 18F2550	93
FIGURA 63. PANTALLA LCD 2 X8.....	93
FIGURA 64. MEDIDAS DE LCD 2 X 8	94
FIGURA 65. CONFIGURACIÓN ELECTRÓNICA DE LA LCD	95
FIGURA 66. INTERACCIÓN LCD CON MICROCONTROLADOR.....	96
FIGURA 67. REGISTROS DE LA LCD.	96
FIGURA 68. DIAGRAMA ELECTRÓNICO DEL ACONDICIONAMIENTO DE LA BATERÍA.....	101
FIGURA 69. BATERÍAS GENÉRICAS DE 3.5V.....	101
FIGURA 70. PROCESOS DE RECEPCIÓN DE DATOS.	103
FIGURA 71. RECEPTOR DE TEMPERATURA CORPORAL.	104
FIGURA 72. ESTRUCTURA DEL SOFTWARE DE APLICACIÓN	104
FIGURA 73. PROCESO DE INTEGRACIÓN CON LA BASE DE DATOS.	105
FIGURA 74. PROCESO DE INGRESO A LA BASE DE DATOS.....	106
FIGURA 75. FICHA MÉDICA DEL PACIENTE.	106
FIGURA 76. DATOS DE MONITOREO DE TEMPERATURA CORPORAL.	107
FIGURA 77. GRÁFICO ESTADÍSTICO DE LA TEMPERATURA CORPORAL	108
FIGURA 78. SISTEMA DE TEMPERATURA CORPORAL.	109
FIGURA 79. TERMÓMETRO DE MERCURIO.....	114
FIGURA 80. TERMÓMETRO DIGITAL.....	116
FIGURA 81. GRAFICO ESTADÍSTICO DE LA TEMPERATURA CORPORAL.	117

INDICE DE TABLAS

TABLA 1	22
PARÁMETROS DE LA ZONA DE TERAPIA INTENSIVA.....	22

TABLA 2.	29
CARACTERÍSTICAS DE MÓDULOS DE TRANSMISIÓN INALÁMBRICA XBEE Y BLUETOOTH	29
TABLA 3.	44
DATOS INICIALES DEL USUARIO.	44
TABLA 4	45
DATOS DE IDENTIFICACIÓN DEL USUARIO.	45
TABLA 5	45
DATOS DE EVALUACIÓN DEL USUARIO.	45
TABLA 6	46
DATOS DE TERAPIA INTENSIVA DEL PACIENTE	46
TABLA 7	46
<i>DATOS MÉDICOS DEL PACIENTE.</i>	46
TABLA 8	50
TEMPERATURA CORPORAL DEL SER HUMANO EN ETAPAS NORMALES	50
TABLA 9	55
TIPOS DE SISTEMAS OPERATIVOS DE SOFTWARE LIBRE.	55
TABLA 10.	61
DISPOSITIVOS DE MEDICIÓN DE TEMPERATURA.....	61
TABLA 11	67
CARACTERÍSTICAS YSI 427.....	67
TABLA 12	68
CARACTERÍSTICAS SENSOR DE PIEL ELLAB.....	68
TABLA 13	69
CARACTERÍSTICAS SENSOR DE DESECHABLE.	69
TABLA 14	70
CARACTERÍSTICAS SENSOR DE EXACON.	70
TABLA 15	72

CUADRO DE CARACTERÍSTICAS LM 324.....	72
TABLA 16	73
CUADRO DE CARACTERÍSTICAS INA122.....	73
TABLA17	74
CARACTERÍSTICAS DE MICROCONTROLADORES CON DIFERENTES FABRICANTES.....	74
TABLA 18	75
PINES DEL CONECTOR DE UN DB9.	75
TABLA 19	85
EQUIVALENCIAS DE VOLTAJE Y TEMPERATURA.	85
TABLA 20	91
DATOS DEL MICROCONTROLADOR 18F2550	91
TABLA 21	94
DATOS DEL MÓDULO LCD 18F2550.	94
TABLA 22.	97
CÓDIGOS DE EJECUCIÓN COMANDOS CLEAR DISPLAY.	97
TABLA 23.	97
CÓDIGOS DE EJECUCIÓN COMANDOS HOME.	97
TABLA 24.	98
CÓDIGOS DE EJECUCIÓN COMANDOS ENTRY MODE SET.	98
TABLA 25.	98
CÓDIGOS DE EJECUCIÓN COMANDOS DISPLAY ON/OFF CONTROL.	98
TABLA 26.	99
CÓDIGOS DE EJECUCIÓN COMANDOS CURSOR OR DISPLAY SHIFT.	99
TABLA 27.	99
CÓDIGOS DE EJECUCIÓN COMANDOS FUNCTION SET.	99
TABLA 28.	100
CÓDIGOS DE EJECUCIÓN COMANDOS WRITE DATA.	100

TABLA 29.	100
CÓDIGOS DE EJECUCIÓN COMANDOS WRITE DATA.	100
TABLA 30	102
COMPONENTES DE BATERÍA DE CELULAR	102
TABLA 31	110
ESPECIFICACIONES TRANSMISOR DE TEMPERATURA CORPORAL.....	110
TABLA 32.	111
ESPECIFICACIONES RECEPTOR TEMPERATURA CORPORAL.....	111
TABLA 33	111
ESPECIFICACIONES BASE DE DATOS.	111
TABLA 34	112
CONSUMO DE CORRIENTE TRANSMISOR DE TEMPERATURA CORPORAL.....	112
TABLA 35	114
OPERACIÓN DE BATERÍAS AL 100 %.	114
TABLA 36.	114
OPERACIÓN DE BATERÍAS CON MODIFICACIÓN AL SISTEMA.	114
TABLA 37	115
ESPECIFICACIONES DEL TERMÓMETRO DE MERCURIO.....	115
TABLA 39	116
ESPECIFICACIONES DE UN TERMÓMETRO DIGITAL.	116
TABLA 40	117
COMPARACIÓN DE MUESTRAS CON TERMÓMETRO DE DIGITAL.	117
TABLA 41.	118
ESPECIFICACIONES DE UN TERMÓMETRO FLUKE	118
TABLA 42.	118
COMPARACIÓN DE MEDIDAS CON UN TERMÓMETRO FLUKE.	118
TABLA 43.	119

COMPARACIÓN DE MUESTRAS DE MEDIDORES DE TEMPERATURA CORPORAL.....	119
TABLA 44	120
ALCANCES DISPOSITIVOS INALÁMBRICOS.....	120
TABLA 45.	121
COTIZACIÓN DE COSTOS DEL SISTEMA DE TEMPERATURA CORPORAL	121
TABLA 46.	122
DESCRIPCIÓN DE ALARMAS DE TEMPERATURA CORPORAL.	122

RESUMEN.

El Sistema de monitorización de temperatura corporal nació con la necesidad de ayudar a monitorear a los pacientes dentro de la zona terapia intensiva, basándose en las necesidades del personal médico, aplicando su desarrollo mediante tres procesos; censado, transmisión y alarmas.

El sistema de Monitorización de temperatura corporal contiene un transmisor que censa la variación de temperatura de un ser humano dentro de un rango establecido, mostrando sus resultados en una pantalla LCD y enviando los datos a un receptor en forma inalámbrica el cual está conectado a un computador, que es el encargado de almacenar los datos y alertar con alarmas audibles, en caso de existir variación de temperatura.

El ordenador almacena los datos recibidos por el transmisor que pertenece a un determinado paciente, visualiza los datos en tiempo real de la temperatura y muestra en cuadros estadísticos la variación de la misma, así como los datos médicos de ficha médica del paciente.

La versatilidad del transmisor permite al paciente tener movilidad dentro del área de cobertura ya posee un sistema de alimentación con baterías recargables.

El sistema de base de datos esta desarrollado en programas de licencia libre, que facilita tener el código abierto para su futura actualización y especialización.

Por la responsabilidad y rápida respuesta que se debe tener en la zona de terapia intensiva el sistema de monitorización de temperatura corporal, es un fiel aliado del personal médico para desempeñar y realizar la prevención de salvar vidas que es su meta final.

ABSTRACT.

System monitoring body temperature, was born with the need to help monitor patients, within the intensive care area, based on the needs of the medical staff, applying their development through three processes; census, transmission and alarms.

Monitoring system body temperature, contains a transmitter which senses the variation of temperature of a human being within a set range, Displaying their results on a LCD screen and sending data to a wireless receiver, which is connected to a computer that is responsible for storing the data and with audible alert, if any temperature variation.

The computer stores the data received by the transmitter, which belongs to a particular patient, the data displayed in real time and displays the temperature variation in statistical tables of the same, as well as medical data of the patient's medical file.

The versatility of the transmitter, allows the patient to be mobile within the coverage area, and has a power system with rechargeable batteries.

The database system was developed on freely licensed programs that have provided open source for future updating and specialization.

For quick response and responsibility that should be taken in the intensive care unit, the system monitoring body temperature, is a staunch ally of the medical staff to perform and perform lifesaving prevention is the ultimate goal.

PRESENTACIÓN

El presente proyecto tiene como objetivo realizar el diseño e implementación de un sistema de monitorización de temperatura corporal, para el hospital “San Vicente De Paúl De Ibarra” constando de los siguientes contenidos.

El primer capítulo redacta sobre los principales sistemas de transmisión inalámbrica, sus estándares y regulaciones, argumentado cuál tecnología posee las mejores prestaciones para el sistema de temperatura corporal, así como también un análisis a los avances médicos más relevantes en el campo de la medicina.

El segundo capítulo trata sobre el diseño de la red, Diseño de la base de datos y la configuración de dispositivos inalámbricos a utilizarse. De igual forma se realiza un completo análisis de los diferentes dispositivos electrónicos que se implementarán en el diseño del transmisor.

En capítulo tercero hace referencia a la construcción, del transmisor de temperatura corporal con los diferentes materiales electrónicos analizados en el segundo capítulo junto con sus mejores características obteniendo la mejor funcionalidad y compactación de los elementos. De igual forma se configura el receptor de temperatura corporal para la recepción de datos.

En el capítulo cuarto muestra las pruebas y tablas a las que se expuso el sistema de monitorización de temperatura corporal para poder lograr la implementación del sistema en el hospital “San Vicente De Paul De Ibarra”.

En el capítulo quinto se realiza las respectivas conclusiones y recomendaciones que género el proyecto de titulación del sistema de monitorización de temperatura corporal.

CAPITULO 1

1. FUNDAMENTACION TEORICA

En este capítulo, se procede a realizar un análisis de los diferentes estándares inalámbricos y protocolos más utilizados; así como también, avances tecnológicos en la medicina, de mayor relevancia en el área de la comunicación inalámbrica.

1.1. EI MUNDO DE LAS REDES INALMBRICAS.

En el mundo actual, los avances tecnológicos transforman y evolucionan culturas enteras dotando de varios servicios y satisfaciendo a los usuarios, en especial el deseo de estar comunicados y llegar a más destinos. Con el inmenso auge de los servicios móviles los usuarios demandan una mayor calidad de comunicación y libertad de desplazamiento sin ningún cable de por medio; partiendo de esta ideología se han desarrollado un sin número de tecnologías inalámbricas que crecen de forma exponencial sin límite de terminar.

1.1.1 INTRODUCCIÓN A REDES INALÁMBRICAS

Ingresar a las redes de comunicación inalámbrica, es analizar y comprender un universo de infinitas posibilidades y requerimientos técnicos comenzando desde los primeros enlaces vía radio, entre cortas y grandes distancias, pasando por la comunicación de estaciones satelitales, estaciones de comunicación móvil para luego detenernos a mirar cómo se condena a desaparecer al teléfono fijo siendo su sustituto el móvil actual o celular.

En corporaciones, empresas públicas y privadas las redes de comunicación inalámbrica se han vuelto un actor principal de apoyo y trabajo, creándose así estándares como el IEEE¹ 802.11 más conocido como Wi-Fi² que aumento un gran

¹ IEEE-Instituto de Ingenieros Eléctricos Y Electrónicos.

² Wi-Fi.- Fidelidad Inalámbrica, Tecnología de comunicación.

estatus social y económico a los hombres de negocios; superando a sus antecesores de cableado estructurado en movilidad y prestaciones similares.

Por otra parte, las comunicaciones inalámbricas han tenido una gran aceptación en el convivir diario de las personas en especial en el monitoreo de la salud como son las WPAN³ con tecnologías como Bluetooth⁴ y más. Y finalmente una tendencia muy aceptada se ve en la industria como son los protocolos en aplicaciones de control y monitoreo de procesos con estándares como Zigbee⁵ o UWB⁶.

1.2. WLAN.

WLAN son la siglas en inglés de Wireless Local Area Network, que es un sistema de comunicación flexible utilizado como alternativa de la transmisión por cobre, la cual supone costes de infraestructura más bajos y una excelente adaptación a las necesidades de movilidad. Por estas prestaciones nacieron las WLANS, cuyo estándar es más conocido como el 802.11⁷ en cualquiera de sus variantes.

El nacimiento de la WLANS, se remonta a 1979 con la publicación de los resultados de un experimento realizado por los ingenieros de IBM en Suiza; el cual se basó en interconectar varios infrarrojos en forma de una red para la adquisición de datos en una fábrica.

El desarrollo de las redes inalámbricas, tomó su rumbo tanto con las tecnologías de infrarrojos como de microonda, lo que dio lugar por el año de 1985 y tras cuatro años de estudio de la FCC⁸, el ente regulador y administrador de los procesos de Telecomunicaciones en asignar las bandas ISM (Industrial, Scientific and Medical), bandas médicas para uso comercial sin costo de frecuencia.

El desarrollo en el mercado de los equipos de telecomunicaciones y de tecnología, promovió el crecimiento de las WLANS y su normalización y

³ WPAN.- Red Inalámbrica De Comunicación Personal.

⁴ Bluetooth.- Tecnología de Radio Corto.

⁵ Zigbee.- Tecnología de Radiodifusión Digital

⁶ UWB.- Tecnología inalámbrica de banda ultra ancha

⁷ 802.11 Conjunto de Especificaciones inalámbricas de una red LAN.

⁸ FCC-Comisión Federal De Comunicación en regulación de redes inalámbricas.

estandarización extendiéndose al mundo de las redes inalámbricas en la banda de frecuencia 2.4 GHz perteneciente al estándar IEEE 802.11 más sus diversas variables.

A pesar de que en la actualidad, existen diversos estándares de redes LAN inalámbricas, que se han establecido en los diferentes entornos es considerable analizar el estándar IEEE 802.11, el cual asegura una mejor comprensión para la presente aplicación.

1.2.1 ARQUITECTURA DE LA IEEE 802.11.

(Roldan, 2009). La red 802.11 se divide en celdas llamadas BSS⁹, que son círculos de cobertura administradas por una estación base o AP¹⁰. Logrando así que los dispositivos de cada zona de cobertura BSS puedan comunicarse entre sí con los diferentes AP los cuales necesitan un sistema troncal de distribución. De esta manera forman la estructura básica de una WLAN. Sin embargo, cuando existe una agrupación superior de segmentos se la denomina ESS¹¹. A continuación en la figura 1.1, se muestra la arquitectura de una red 802.11 básica

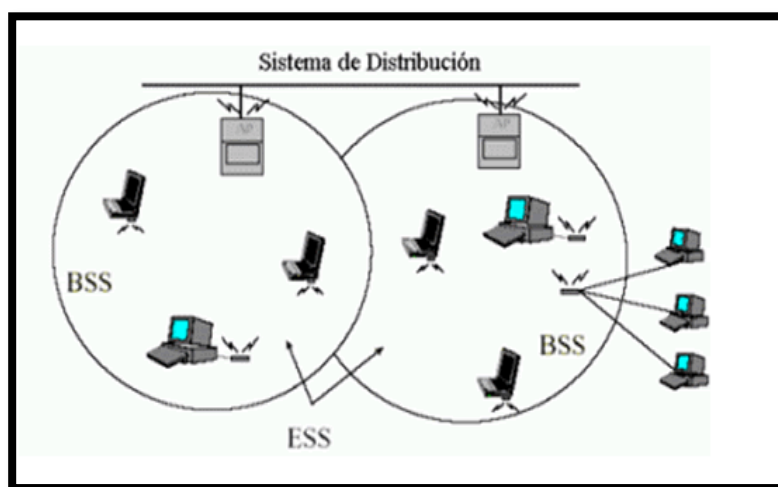


Figura 1. Componentes de una red IEEE 802.11. Fuente: Laser Wifi (2010). Recuperado de: <http://www.laserwifi.com/estander802.11.html>.

⁹ BSS- Servicio Básico De Cobertura inalámbrica.

¹⁰ AP-Punto de Acceso a redes inalámbricas.

¹¹ ESS Extensión De Servicios De Cobertura Inalámbrica

1.2.2 TOPOLOGÍA DE UNA LAN INALÁMBRICA

En una red WLAN, se distinguen dos tipos de configuraciones diferentes, en función de la utilización de puntos de acceso o no.

Entre las topologías más sencillas se maneja las redes ad-hoc¹² en la que las terminales móviles se comunican una a otra, utilizando una tarjeta inalámbrica con la desventaja que se deben encontrar dentro de una zona de cobertura.

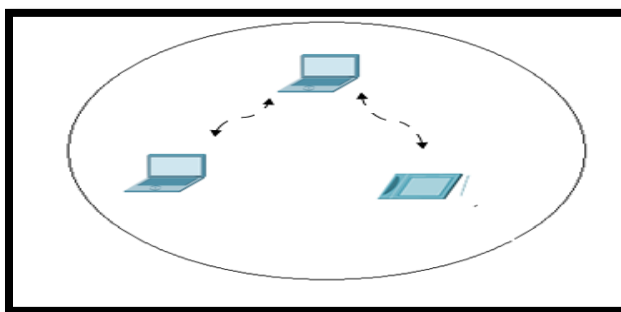


Figura 2. Configuración de una red ad hoc.

(Limachi, 2010) Otra alternativa de topología de una WLAN, es la de infraestructura, en la cual un dispositivo se encarga de centralizar las comunicaciones denominándose punto de acceso AP o Access Point, haciendo las veces de un repetidor inalámbrico. Estas formas de red son mucho más complejas ya que requieren de un diseño y planificación de puntos de acceso, distribuidos estratégicamente para evitar que algunas zonas se queden sin cobertura, asegurando así un ancho de banda y velocidad de transmisión para cada usuario de la Wlan.

¹² Ad-hoc- Configuración Punto a Punto entre dispositivos.

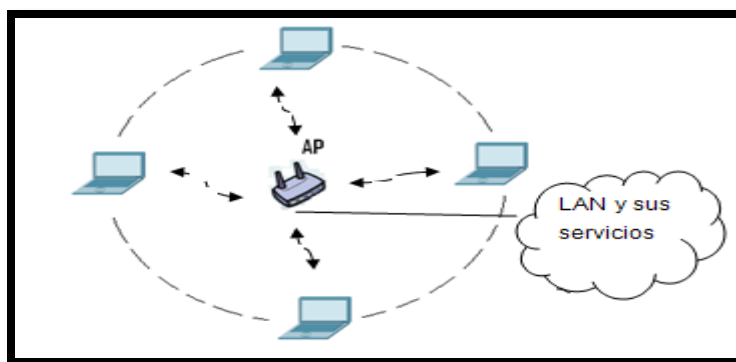


Figura 3. Topología con punto de acceso en una Lan.

Los terminales se conectan a través del AP, el cual permite intercambiar información con dispositivos conectados al mismo AP.

Otra posibilidad es conectar varios segmentos de red, a través de un radioenlace, el cual puede ser punto o multipunto. Lo que incorpora utilizar equipos de telecomunicaciones como antenas directivas¹³ con el alcance de grandes coberturas y la interconexión de redes LAN, como se muestra en la figura 1.4:

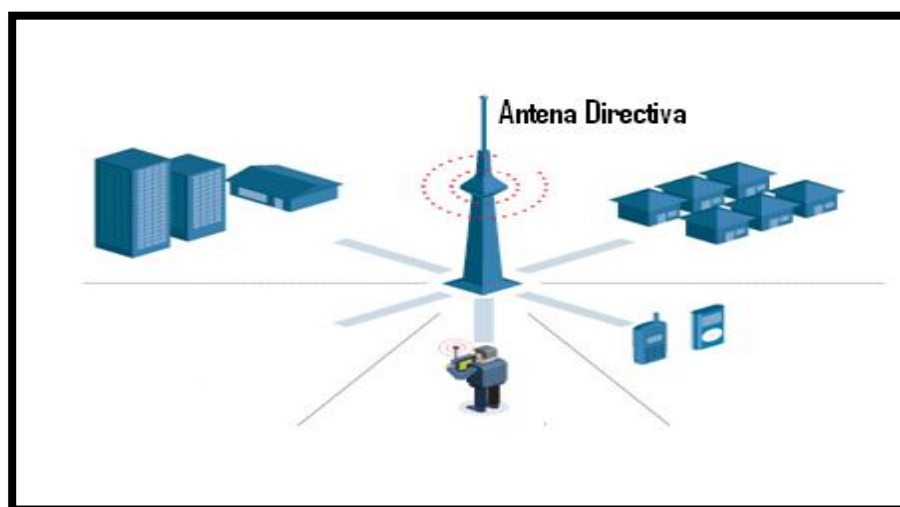


Figura 4. Ampliación de cobertura mediante antenas directivas.

1.2.3 ESPECIFICACIONES DE UNA WLAN

El estándar 802.11, es un conjunto de especificaciones que abarcan todas las características de una red WLAN. En la actualidad hay 4 especificaciones que conforman a una red WLAN, las cuales están agrupadas en diferentes áreas como las especificaciones de nivel físico la 802.11a, 802.11b y 802.11g que definen las técnicas de modulación y el procesamiento de la señal a nivel bajo. En otro aspecto están las de

¹³ Antenas Directivas.- Centralizan gran parte de la radiación en el lugar solicitado.

calidad de servicio (QoS) y seguridad que son estudiadas por la 802.11e y en 802.11i. Así como también la 802.11h y 802.11j que procura la interoperabilidad entre diferentes dispositivos de largo alcance. Terminando con 802.1X que soporta la autenticación de usuarios; a continuación un detalle de cada una de estas variantes.

1.2.3.1 802.11a

(Vocal, 2010). Es una variante, a 5 GHz denominado WIFI 5 que emplea modulación 64-QAM¹⁴ y codificación OFDM¹⁵, que ofrece una transmisión de datos a una velocidad de 6, 9, 12, 18, 24, 36, 48 hasta 54 Mbps con un alcance limitado, a 50m. Utiliza 52 canales de subportadora cada uno de ellos con aproximadamente 300 KHz de ancho de banda. De los cuales se distinguen 48 canales para de transmisión de datos y los 4 canales restantes para el control de errores.

1.2.3.2 802.11b.

El estándar 802.11b es el más conocido en el mundo de las redes inalámbricas como WIFI; actualmente es el más utilizado y brinda mayor prestaciones de servicio en redes inalámbricas. Ofrece velocidades normalizadas de 11 Mbps y tiene un alcance de hasta 300 metros en un espacio abierto. Maneja el rango de frecuencia de 2,4 GHz con tres canales de radio disponibles.

1.2.3.3 802.11c.

El estándar 802.11c, es una versión modificada del estándar 802.11d el cual indica la información que se necesita para conectar dos redes entre sí. Es decir, facilita aspectos como los canales disponibles de comunicación, calidad de servicio y filtrado de tramas.

1.2.3.4 802.11d.

El estándar 802.11d, especifica los requisitos para que los dispositivos intercambien información con las limitaciones regulatorias fuera de Europa, Japón, y

¹⁴ 64-QAM- Modulación de amplitud en cuadratura para obtener la mayor eficiencia de mi señal al momento de tomar muestras de esta.

¹⁵ OFDM- Multiplexación por División de Frecuencia Ortogonal que envía un conjunto de ondas portadoras para ser moduladas.

Estados Unidos. Enfocándose en el desarrollo de prototipos a 5 GHz, puesto que las frecuencias varían de un país a otro, logrando así comunicar dispositivos en cualquier parte del mundo.

1.2.3.5 802.11e.

El objetivo del estándar 802.11e, es mejorar la calidad del servicio en la mayoría de sus aspectos permite priorizar los diferentes tipos de tráfico, y controlar los retardos de transmisión para optimizar la comunicación en audio y video.

1.2.3.6 802.11f.

Este protocolo, especifica las recomendaciones para proveedores de puntos de acceso que proporcionan la información necesaria para efectuar roaming¹⁶; entre diferentes operadores lo que permite la compatibilidad entre productos evitando la falta de comunicación en cualquier punto geográfico.

1.2.3.7 802.11g.

El estándar 802.11g es una variante de las más completas, que alcanza velocidades de 22 Mbps en un rango de frecuencia de 2,4 GHz. Una de las funcionalidades principales de este estándar es que es permite coexistir con los todos los equipos instalados, que trabajan con el estándar 802.11b; con la desventaja que disminuye prestaciones a los equipos del 802.11g; ya que le obliga a trabajar en velocidades de 11Mbps y no soporta mecanismos de envío OFDM; lo cual produce colisiones de datos.

1.2.3.8 802.11h.

El estándar *802.11h*, se centra en acoplar el estándar 802.11 con el estándar inalámbrico europeo, cumpliendo con sus regulaciones de frecuencias y operaciones energéticas.

¹⁶ ROAMING.- Capacidad de cambiar de un área de cobertura a otra.

1.2.3.9 802.11i.

Uno de los aspectos más importantes en una red inalámbrica es la seguridad de sus datos, que al ser comunicados vía radio son propensos a tener acceso a la información. El estándar *802.11i* define los mecanismos que mejoran la autenticación y cifrado de datos definidos en el estándar WEP¹⁷ logrando una transmisión segura y fiable de datos.

1.2.3.10 802.11k.

El estándar 802.11k, trabaja en consolidar en un solo modo los estándares a, b y g; tanto en las operaciones radioeléctricas y de red; ayudando de igual manera a la gestión de la red y la detección de fallos.

1.2.3.11 802.11m.

El estándar 802.11m, es el encargado de realizar un conjunto de normas de mantenimiento, clasificación, interpretación; con todo lo referente al estándar 802.11 y sus nuevas versiones.

1.2.3.12 802.11n.

El estudio de los estándares anteriores, se basa en velocidades de transmisión limitadas en la práctica a 11Mbps en 802.11b y a 20 Mbps en 802.11a/g; por lo cual existe una gran desventaja a las redes LAN cableadas que trabajan en velocidades de transmisión de 100 Mbps. Por lo cual el estándar 802.11n se centra en poner velocidades de transmisión cercanas a los 100 Mbps, para interactuar con cualquier equipo de los estándares inalámbricos 802.11a, 802.11b y 802.11g y así satisfacer las necesidades en entornos corporativos.

¹⁷ WEP-Privacidad Equivalente Ha Cableado. Trabaja en la seguridad inalámbrica de una red.

1.3. WPAN

(Catarina., 2010). La palabra WPAN es un término utilizado para describir a una red inalámbrica de área personal, que encierra a las redes con un rango de operación limitado a unos pocos metros. Por lo cual, se suele utilizar en lugares reducidos para la comunicación entre dispositivos que se encuentran en constante movilidad como teléfonos, celulares, dispositivos de audio, impresoras, PDA¹⁸ y otros que necesitan transmitir información.

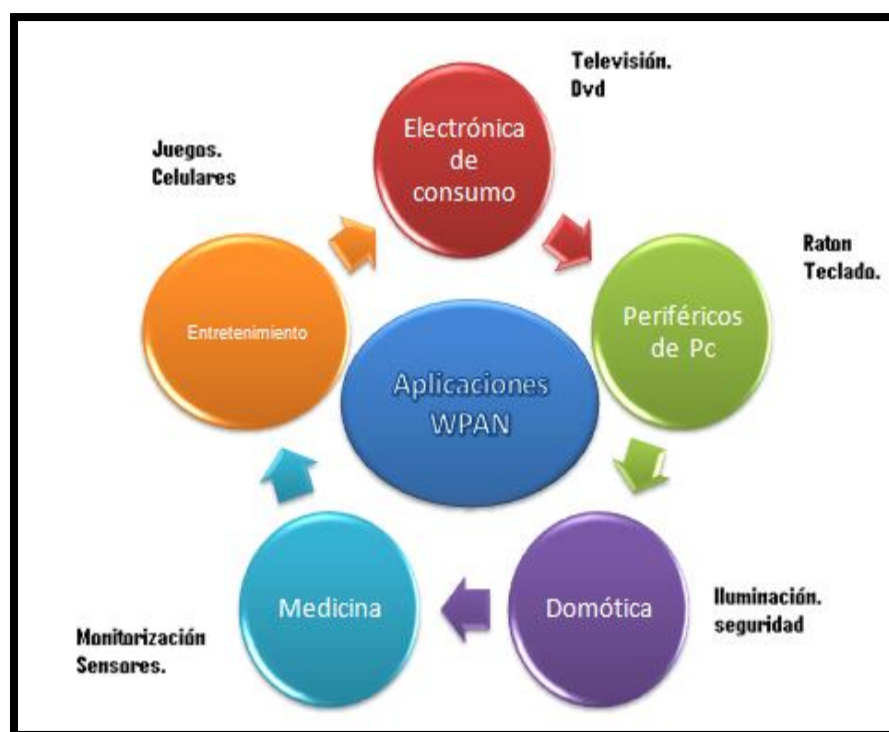


Figura 5. Aplicaciones de la WPAN.

Las características fundamentales de este tipo de redes, es que centran su sistemas de comunicación en un área mínima de 10 metros a la redonda, emitiendo o recibiendo información; sea que se encuentren en movimiento o no. Esta ventaja a permitido su

¹⁸ PDA- Asistente Personal Digital cumple la función de un minicomputador.

rápida evolución absorbiendo nuevas aplicaciones, que van aumentando cada año en entornos de corta distancia y velocidades de baja transmisión.

Cada una de las aplicaciones de una WPAN presenta necesidades diferentes, de acuerdo al entorno de comunicación que encuentren. Bajo este propósito se han creado una amplia gama de tecnologías en las cuales se destacan los estándares IEEE 802.15; como Bluetooth, Wimedia¹⁹, Zigbee y otros menos utilizados como HomeRF, IrDA y RFID.

1.3.1 BLUETOOTH

Bluetooth es un estándar de conectividad inalámbrica, utilizado en las WPAN que nació con el fin de eliminar los cables de los periféricos como por ejemplo de las impresoras; y que luego por su bajo coste y su pequeño alcance que apenas era de 10 metros fue evolucionando para posibilitar la transmisión de voz y datos; mediante enlaces de radiofrecuencia y en la banda ISM de los 2,4 GHz con una velocidad de transmisión aproximadamente de 1Mbps. Logro estandarizarse como uno de los estándares inalámbricos más importantes del mercado, logrando así proporcionar las comunicaciones entre equipos móviles y fijos a más de otorgar la posibilidad de crear pequeñas redes inalámbricas en lugares de difícil acceso y geografía.

1.3.1.1 Topología De Red

Al ser Bluetooth un protocolo de comunicación de cobertura limitada, soporta comunicaciones punto a punto y comunicaciones multipunto.

Los dispositivos Bluetooth forman pequeñas agrupaciones, a las cuales se las conoce como piconet que se conectan entre si, y se encuentran encendidos en la misma zona; todos ellos sincronizados tanto en tiempo como en frecuencia. En la zona de un piconet, existen diferentes topologías de transmisión entre los dispositivos Bluetooth. La red más simple es la conexión punto a punto en la que un dispositivo Bluetooth trabaja como maestro y el otro dispositivo bluetooth como esclavo. Otra

¹⁹ WiMedia.- Tecnología de transmisión inalámbrica de pulsos cortos.

forma de conectividad, es punto a multipunto donde múltiples esclavos se conectan a un maestro; compartiendo el ancho de banda para todos los dispositivos Bluetooth.



Figura 6. Conexión Punto a Punto

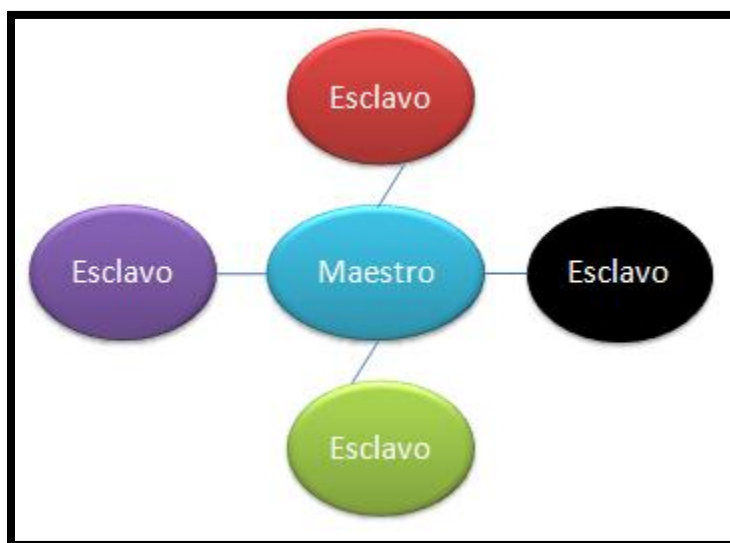


Figura 7. Conexión Punto a Multipunto.

1.3.1.2 Especificaciones Del Estándar Bluetooth

Bluetooth, opera con una tecnología conocida como espectro ensanchado por salto de frecuencia o FHSS, donde la señal se transfiere por un salto aleatorio de frecuencias hasta llegar al transmisor en el caso del Bluetooth se lo realiza en la banda de frecuencia de los 2,4 GHz con una modulación GFSK²⁰, con una potencia del

²⁰ GFSK.- Modulación Por Desplazamiento De Frecuencia Gausiana.

transmisor de 1 mW para un alcance de 10 metros y 100 mW, para un alcance de hasta 100 metros, donde el consumo de amperaje va desde los 30 μ A hasta los 30 mA transmitiendo.

1.3.2 WIMEDIA

Wimedia conocida también como UWB²¹, es una tecnología que se la utiliza en aplicaciones multimedia que necesitan un gran ancho de banda y calidad de servicio, basada en la transmisión de pulsos cortos que trabajan en la frecuencia de 2,4 Ghz y transmitiendo en diferentes velocidades como 11 Mbps, 22 Mbps, 33 Mbps, 44 Mbps, 55 Mbps para dispositivos de bajo coste. El alcance de una UWB está comprendido en una zona de 5 metros hasta 55 metros con una potencia de transmisión aproximada de 10 dBm. La principal ventaja de UWB en una red WPAN, es su robustez de propagación con dispositivos de bajo coste y baja potencia de transmisión.

1.3.2.1 Topológica.

(Hadaschik, 2010). UWB trabaja como una red ad-hoc, en la cual utiliza OFDM para obtener una transmisión robusta con bajas atenuaciones e interferencias. Con gran ancho de banda y calidad de servicio.

1.3.3 ZIGBEE

La tecnología Zigbee, conocida como el estándar IEEE 802.15.4 se utiliza en aplicaciones en las que no se necesitan un gran ancho de banda, con un bajo consumo de potencia y en espacios reducidos. Principalmente nació como una aplicación para hogares en el área de Domótica²².

La tecnología Zigbee puede operar en tres bandas diferentes en la 868 MHz para Europa, la 915 MHz para Estados Unidos y la 2,4 GHz para todo el mundo aplicando su mayor uso en la monitorización de variables, con velocidades de transmisión de datos de 250 Kbps para la banda de 2,4 GHz; de 40 Kbps en los 915 MHz y los 20 Kps en la banda de 868 Mhz.

²¹ UWB.- Banda Ultra Ancha

²² DOMOTICA.- Termino utilizado para denominar la automatización de viviendas.

1.3.3.1 Topología en una Red ZIGBEE.

La tecnología zigbee se adapta a diferentes topologías de red, siendo las más utilizadas la topología Estrella, la topología Punto a Punto y la topología en forma de malla. En la tecnología zigbee existen dispositivos completamente funcionales y parcialmente funcionales; en los cuales a los dispositivos que operan completamente se los denomina FDD²³ y los parcialmente funcionales como RFT²⁴.

Las características principales de un dispositivo que opera como FDD, es que puede ser el coordinador de toda la red en cualquier tipo de topología con la capacidad de comunicarse con cualquier dispositivo dentro de su red; y a la vez intercambiar información con otros dispositivos fuera de su red, trabajando como Router Zigbee conocido como *ZigBee Router (RZ)*. De igual manera, la forma de trabajar de un dispositivo RFT es limitada no puede operar como coordinador de una red y su uso se ve más aplicado en topología estrella. Logrando así distancias de comunicación de 30 metros en las distintas topologías de red y distancias más largas en diferentes aplicaciones.

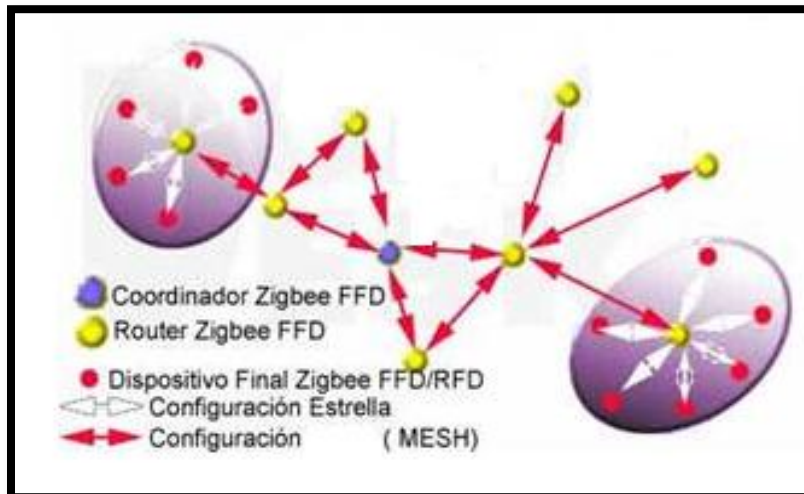


Figura 8. Topologías Zigbee. Fuente: Monografías.com (2011). Recuperado de: <http://www.monografias.com/trabajos61/zigbee-estandar.coml>

²³ FDD.- Dispositivo Completamente Funcional en la tecnología Zigbee.

²⁴ RFT.- Dispositivo de Función Reducida en la tecnología Zigbee.

1.3.4 IRDA

(Wikipedia., 2010). Es una de las tecnologías inalámbricas más extendidas, denomina IRDA o asociación de datos por infrarrojo, la cual se utiliza para proporcionar la conectividad entre equipos. Utilizando una transmisión bidireccional de extremo a extremo a velocidades que se sitúan entre los 9.600 bps y los 16 Mbps.

1.3.4.1 Topología IRDA

La topológica que se utiliza en la tecnología IRDA, es punto a punto con un ángulo direccional de 30° grados con la desventaja que no permite traspasar las paredes, operando a una distancia máxima de 0 a 1 metros, trabajando en un espectro óptico de 850 nm²⁵.

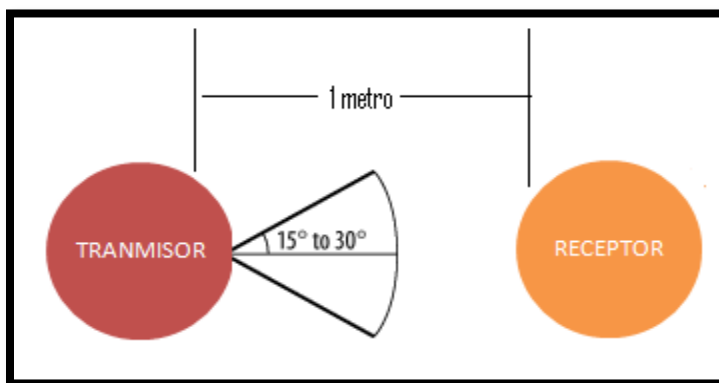


Figura 9. Topologías IRDA.

1.3.5 RFID

Tecnología conocida como identificación por Radiofrecuencia RFID, que aplica el uso de etiquetas que pueden almacenar información y ser leídas por radiofrecuencia; logrando así reemplazar al código de barras cumpliendo con su objetivo principal que es la identificación automática de un objeto.

Las etiquetas RFID, son dispositivos sumamente pequeños iguales a una moneda de un centavo o inclusive más pequeñas, que pueden ser pegadas o insertas en los diferentes objetos a ser identificados. Las etiquetas RFID se clasifican en dos

²⁵ nm.- Nanómetro, unidad de medida utilizada para medir la radiación en espectros y longitudes de ondas.

clases, las activas y las pasivas las cuales contienen antenas para poder comunicarse con el transmisor y receptor en la que no necesitan de una visión directa para establecer la comunicación.

1.3.5.1 Arquitectura RFID.

Un sistema de aplicación RFID, está formado por un grupo de etiquetas, uno o más lectores de etiquetas, un puesto de control y el programa de monitorización y control de los equipos. Las etiquetas activas, tienen la funcionalidad de enviar la información por sí mismas que se encuentran almacenadas ya que poseen un sistema de alimentación eléctrica propio a diferencia de las etiquetas pasivas, las cuales deben ser interrogadas por el lector para mirar la información almacenada ya que no poseen una fuente de alimentación eléctrica. Una vez con la información captada por los lectores en las etiquetas sean pasivas o activas, estas se relacionan con el puesto de control donde se encuentra el respectivo programa el que verifica con la base de datos del sistema y procede a tomar las respectivas acciones, dependiendo de la aplicación en la que se trabaje.

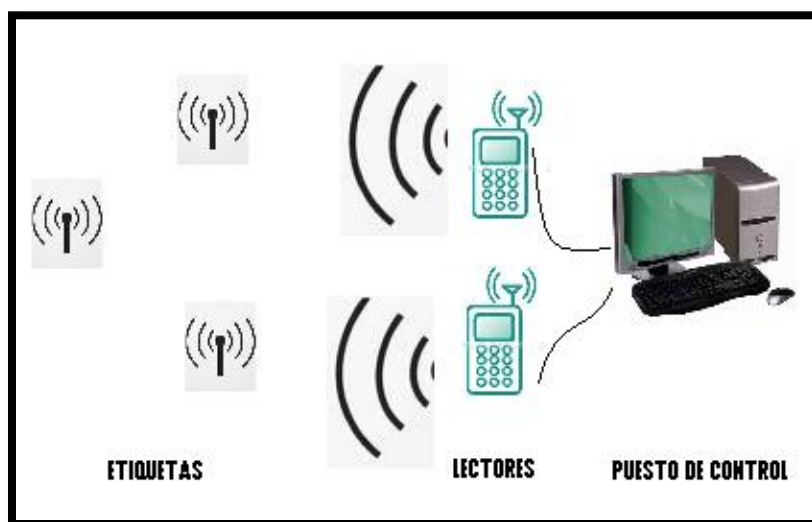


Figura 10. Arquitectura RFID.

1.3.5.2 Bandas de Frecuencia.

Las bandas de frecuencia, se categorizan en tres grupos en baja frecuencia que van desde los 125 KHz a los 134 KHz, en alta frecuencia en los 13,56 MHz y muy alta frecuencia en 5,8 GHz para Europa y los 2,45 GHz para Estados Unidos en ambos casos se requieren licencia de utilización de la banda.

1.4. EL USO DE LAS REDES INALÁMBRICAS EN LA MEDICINA.

En la actualidad la medicina tiene una gran convergencia con las redes inalámbricas, facilitando así cada vez su mayor uso y penetración en la sociedad; mejorando el control de los pacientes y su forma de vivir. A continuación se describe algunos avances tecnológicos:

1.4.1 IMPLANTE DE MOTORIZACION INALAMBRICO EN EL CORAZON

La compañía cardioMes con su matriz en Atlanta, desarrollo un sensor inalámbrico, el cual se implanta en una arteria pulmonar que monitorea la presión del fluido pulmonar y transmite los datos en forma inalámbrica a los médicos encargados, reduciendo así problemas de ataques tanto respiratorios y de corazón.

El sensor se coloca en la arteria pulmonar, con un micro-implante el cual no requiere alimentación ni cables para sensar las muestras de presión. Dos aros de metal colocados a los lados de la arteria junto con un transductor de presión, toman el flujo que transitan por los vasos sanguíneos y transmiten inalámbricamente los datos de variación en el sensor; mediante un receptor que se encuentra en la almohada que entra en funcionamiento en cuanto el paciente se recuesta sobre esta y ejerce presión. Logrando así registra los datos para su respectivo análisis.

(Review, 2011). Los estudios realizados dan muestras, que el sensor reduciría la insuficiencia cardíaca en los pacientes otorgándoles un mejor estilo de vida. Esto es una muestra que en la mayoría de los países desarrollados, se está trabajando en la monitorización de variables fisiológicas del cuerpo humano.

1.4.2 SISTEMA DE RADIO TELEMETRÍA DE TEMPERATURA EN LA PIEL

COCATRE-ZILGIEN JAN H (Febrero 2011). Este sistema establece, que la patente **US5844862** describe el funcionamiento de una radio telemetría, alertando con un sistema de alarmas la elevación de la temperatura de la piel en niños; especialmente para ser utilizado en las noches.

La patente **US5844862** está compuesto por un transmisor y receptor; donde el receptor tiene la opción de conectarse a un reloj de alarma digital o a un reloj de computadora; donde el transmisor posee una alarma acústica, la cual se activa sea por el reloj digital o por el computador utilizando la conexión de forma serial.

Este sistema de medición de temperatura, posee sistemas de seguridad contra daños y desperfectos. Su operación es libre sin requerimientos de licencia, pero su adquisición tiene costo.

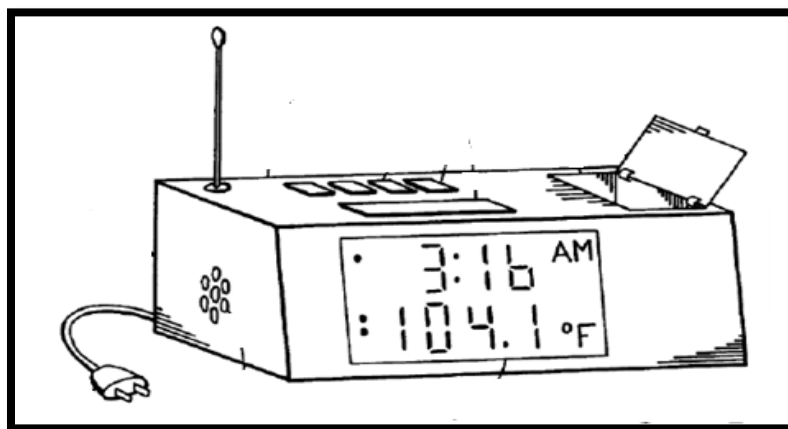


Figura 11. Receptor de temperatura Corporal. Sistema de Monitorización de Temperatura corporal.

Fuente: KUMAR HARPAL S; JOHNSON PAUL (2011). Recuperado de: <http://v3.espacenet.com>

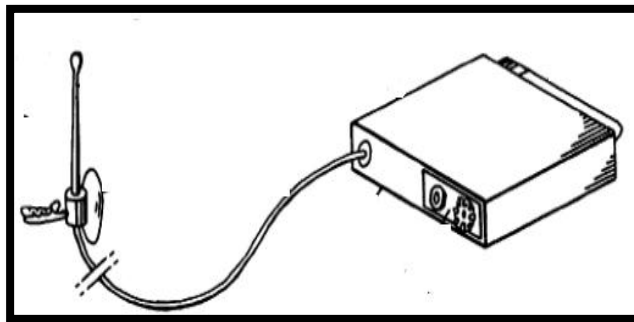


Figura 12. Transmisor de temperatura Corporal. Sistema de Monitorización de Temperatura corporal.

Fuente: KUMAR HARPAL S; JOHNSON PAUL (2011). Recuperado de: <http://v3.espacenet.com>

1.4.3 SISTEMA PORTATIL DE TELEMONITOREO REMOTO DEL PACIENTE

(Kumar Harpal, 2011). El sistema de la patente **US6416471** estudia el funcionamiento de un equipo de monitorización de signos vitales que trabaja remotamente utilizando técnicas de telemetría²⁶. La cual se caracteriza por poseer una banda de ensado que es desechable, que utiliza un sensor de medición de temperatura corporal el cual funciona como un cinturón; que esta sensando constantemente al paciente y enviando sus datos de temperatura a un receptor; el cual abarca hasta unos 60 metros de distancia

La estación base se conecta a la línea telefónica, la cual recibe los datos y los va almacenando para poder llevar un control médico del paciente y sus variables; como lo describe en la imagen a continuación:

²⁶ TELEMETRIA.- Es la medición remota de magnitudes físicas y el posterior envío de información a un centro de control.

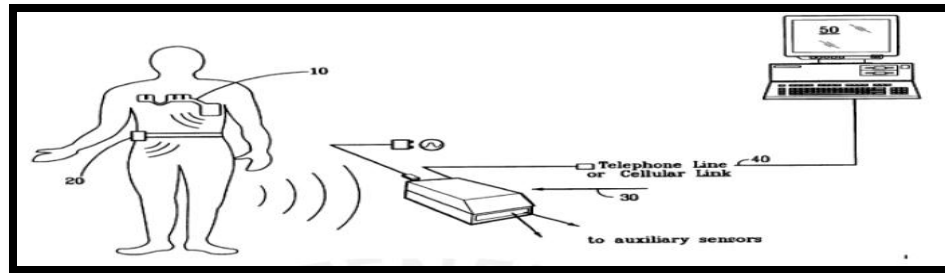


Figura 13. Sistema de Monitorización de Temperatura corporal. Fuente: KUMAR HARPAL S; JOHNSON PAUL (2011). Recuperado de: <http://v3.espacenet.com>

1.4.4 TIRA ELECTRÓNICA QUE MIDE LA TEMPERATURA CORPORAL Y EL RITMO CARDIACO SIN ALIMENTACIÓN ELECTRÓNICA

Un prototipo desarrollado por el Instituto de ingenieros de Microelectrónica de Sevilla (IMSE-CNM), permite medir la temperatura corporal y el ritmo cardiaco; colocando una tira electrónica en la piel la cual transmite los datos en forma inalámbrica sin necesidad de baterías.

(Godoy, 2011). Es un dispositivo el cuál monitorizará a pacientes tanto en sus casas como en centros de salud, mediante tecnología RFID (del inglés *Radio Frequency IDentification*). Este instrumento identificará en tiempo real con un microcircuito RFID los valores de variables fisiológicas de la temperatura y del ritmo cardíaco en pulsaciones por minuto. Así, logran que el paciente se encuentre monitoreado en los lugares de cobertura que la tecnología lo permita.

El trasmisor incluye un transpondedor pasivo el cual reutiliza la señal de transmisión para medir los patrones fisiológicos y almacenar la energía. El principio de este funcionamiento, basa su eje en la reflexión controlada como el fenómeno de una linterna que ilumina un objeto. Cuando la linterna está apagada no se logra ver nada. En cuanto se enciende la linterna se puede distinguir objetos; de igual manera los dispositivos mientras sienten cambios de temperatura al que se encuentran calibrados activan su sistema tanto de transmisión como de alimentación.



Figura 14. Sistema de Monitorización con RFID.*Fuente:* Alvaro Godoy (2011). Recuperado de:

<http://alvarogodoy.bligoo.com/>

CAPITULO II

DISEÑO DE LA RED

En éste capítulo se procede a realizar el diseño de la red inalámbrica, que servirá para medir la temperatura corporal de los pacientes que se encuentran en la zona de terapia intensiva del Hospital San Vicente De Paul; así como también un análisis de los principales problemas que se presentan en un paciente por la variación de la temperatura corporal; junto con la arquitectura y diseño de la base de datos a implementar.

2.1 REQUERIMIENTOS PARA EL DISEÑO DE MONITORIZACIÓN DE LA RED DE TEMPERATURA CORPORAL.

El diseño de la red, se centra en analizar los parámetros requeridos en la zona de terapia intensiva del hospital “San Vicente De Paúl”, para la monitorización de la temperatura corporal tomando en cuenta los siguientes aspectos:

- *Análisis de los parámetros de la zona de terapia intensiva.*
- *Objetivo de la red.*
- *Servicios de la red.*
- *Tipo de red.*
- *Dispositivo de transmisión inalámbrica.*

2.1.1 ANÁLISIS DE LOS PARÁMETROS DE LA ZONA DE TERAPIA INTENSIVA

Actualmente, el departamento de zona de terapia intensiva del hospital San Vicente De Paúl no cuenta con un sistema de conexiones redes inalámbricas. Este cuenta con la mayoría de implementos de monitorización de variables fisiológicas del paciente, las cuales se encuentra junto a la camilla de reposo.

El ingreso de los datos, síntomas y prescripciones del paciente se lo lleva en fichas las cuáles posteriormente se las ingresa en un computador, que posee el jefe de la zona de terapia intensiva dónde no son archivados en una base de datos

especializada para la zona para este tipo de parámetros; si no que se archiva los datos en carpetas creadas en el sistema operativo, que en este caso es Windows XP para llevar un registro de las aspectos más importantes

Tabla 1

Parámetros de la Zona de terapia intensiva.

Parámetros de la Zona de Terapia Intensiva del Hospital San Vicente de Paul	
Área de la Zona de Terapia Intensiva.	40 m ²
Número de Habitaciones.	3

Nota: Elaborado a partir de la información proporcionada por el Departamento de Zona de terapia intensiva del Hospital San Vicente De Paul Ibarra.

2.1.2 OBJETIVO DE LA RED.

El objetivo principal de la red de monitorización de temperatura corporal es la transmisión de los datos por medio inalámbrico; el cual nos muestre fiabilidad²⁷, seguridad y claridad en los datos.

2.1.3 SERVICIOS DE LA RED.

Los servicios principales de nuestra red deben estar basados en movilidad, número de usuarios, transmisión sin línea de vista, velocidad y disponibilidad.

2.1.3.1 Movilidad

La movilidad, es uno de los servicios más importantes en una red inalámbrica que permite que los equipos incorporados o asociados a la red puedan trasladarse de un lugar a otro, ejecutando la transmisión de los datos. Es por esto que los transmisores de temperatura corporal tienen que brindar este servicio; que permitirá tener la monitorización de los pacientes al momento de trasladarse en las instalaciones internas de la zona de terapia intensiva.

²⁷ FIABILIDAD.- Probabilidad de buen funcionamiento de un equipo.

Para poder cumplir con este objetivo el diseño y la construcción del equipo que tomará la temperatura del paciente, debe tener un sistema de alimentación electrónica independiente y que se adapte y movilice junto con el paciente dentro de la zona de terapia intensiva.

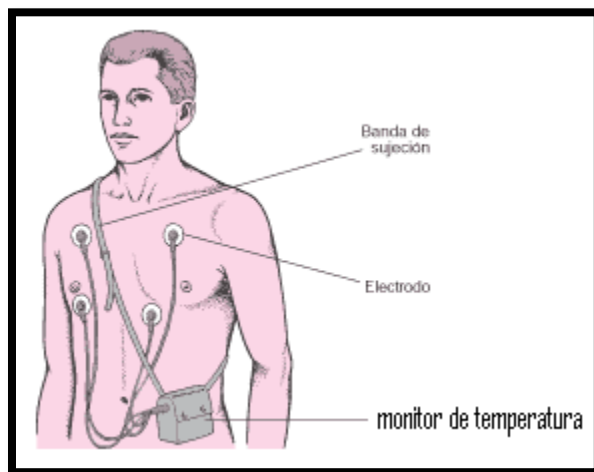


Figura 15. Monitorización de variables fisiológicas. *Fuente:* Taringa (2011). Recuperado de: <http://www.taringa.net/posts/info/Diagnostico-de-enfermedades.html>.

2.1.3.2 Número de usuarios.

El número de usuarios en una red, es uno de los parámetros más importantes al momento de dimensionar la red. Para el caso de la zona de terapia intensiva el número de usuarios dependerá del dispositivo de transmisión inalámbrica a implementar, en la red.

Para poner en marcha el funcionamiento y pruebas del sistema, la red de monitorización de temperatura corporal contará con tres transmisores y la central de recepción, la cual estará diseñada para monitorear y alertar en caso de cambios en la temperatura del paciente.

2.1.3.3 Transmisión sin línea de vista.

La transmisión sin línea de vista es una de las características esenciales que tiene que cumplir el dispositivo, ya que al permitir la movilidad y la monitorización de

temperatura corporal estos deben estar siempre transmitiendo dentro de la zona de terapia intensiva, a la central sin necesidad de estar en visibilidad directa.

2.1.3.4 Velocidad.

La transmisión de los datos deben adaptarse a una velocidad a la cual se pueda tomar acciones de prevención, por parte de los encargados del centro médico ayudando a realizar un trabajo más eficiente.

Este parámetro, de igual manera depende del tipo de dispositivo de comunicación inalámbrico a utilizarse y su tecnología de aplicación.

2.1.3.5 Disponibilidad

La disponibilidad de la red de monitorización de temperatura corporal va a depender de dos parámetros: el primero; que la alimentación electrónica o suministros de baterías de los transmisores inalámbricos tengan un perfecto funcionamiento durante su tiempo de aplicación. Y el segundo que la central en la cual se encuentra la base de datos con su software de monitorización, mantenga todas sus interfaces alertas y en buen estado.

2.1.4 TIPO DE RED A UTILIZARSE EN LA ZONA DE TERAPIA INTENSIVA

Para poder escoger el mejor tipo de red inalámbrica, que se adapte al sistema de monitorización de temperatura corporal de la zona de terapia intensiva, se debe conocer que tipos de redes inalámbricas cumplen con las mejores características; a continuación se describen algunas de las redes inalámbricas utilizadas en este tipo de aplicaciones para poder realizar la mejor elección.

2.1.4.1 Red Mesh

Las redes mesh son aquellas que combinan dos tipos de topologías, la topología Ad Hoc y la topología de infraestructura; permitiendo así trabajar en una área en la cual el centro de transmisión o punto de acceso no necesita tener comunicación directa, sino que sus dispositivos ubicados en diferentes zonas pueden funcionar como

repetidores haciendo llegar la información al destino; creando una red más flexible y dinámica y de amplia cobertura.

(Guatewireless-Org, 2011). Para que en una red mesh la comunicación sea exitosa esta debe compartir los mismos protocolos de enrutamiento y operar en la misma banda de frecuencia. La red debe estar diseñada de una forma en la cual se pueda realizar el menor número de saltos o pasos por repetidores; ya que al paso por una repetidora o saltos pueden ocurrir riesgos como pérdida de información, fallos de los equipos y aumento de retardos por cada salto que ocurre de un nodo a otro. A continuación se muestra una grafica de la representación de una re Mesh.

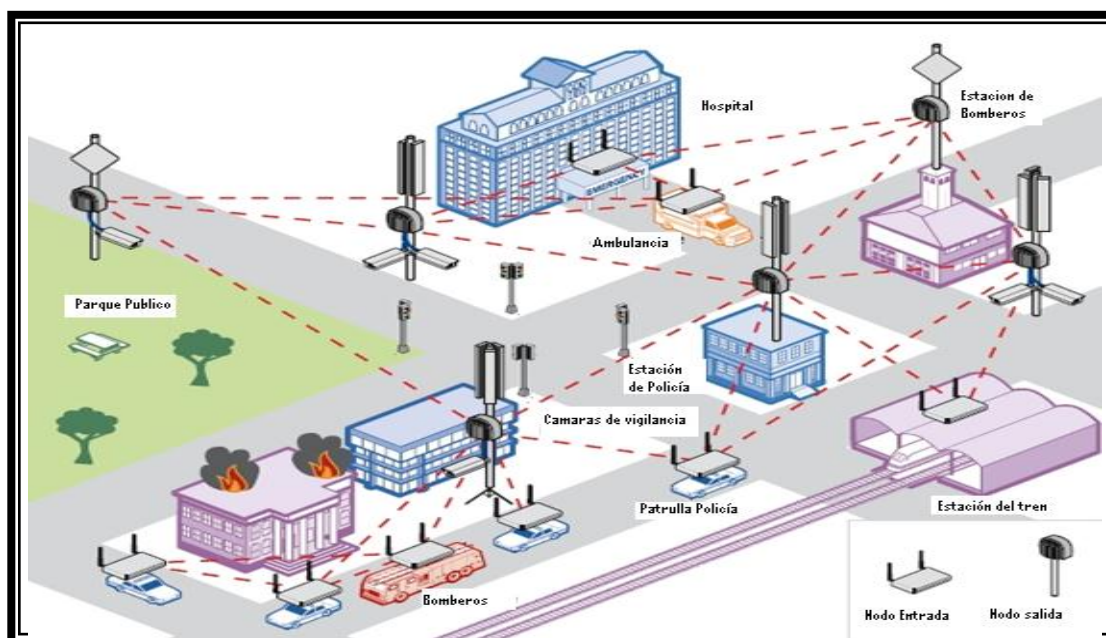


Figura 16. Representación de una Red Mesh en un área urbana. Fuente: Freedom from Wires. (2011)

Recuperado de: <http://digitalcivilization.blogspot.com/2010/11/freedom-from-wires.html>

2.1.4.2 Mobile Ad Hoc Network

(Wikipedia, Mobile ad hoc network., 2011). Una Mobile Ad Hoc Network, o también denominadas como MANET es una red que posee dispositivos móviles donde sus nodos tienen la capacidad de autoconfiguración y un enrutamiento dinámico, sin la necesidad de una administración central; si no que cada nodo puede obtener la

información requerida y administrarla según la necesidad solicitada. Con la desventaja que la mayoría de estas redes no tiene acceso a internet y enfocándose en aplicaciones militares, médicas y de campo.

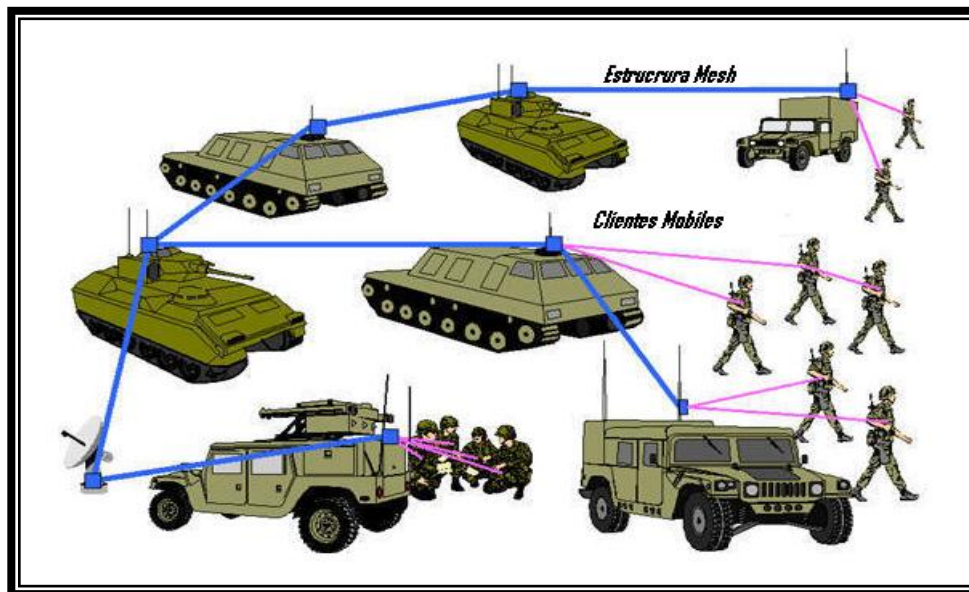


Figura 17. Representación de una red Ad Hoc en un despliegue militar. *Fuente:* Freedom from Wires. (2011) Recuperado de: <http://digitalcivilization.blogspot.com/2010/11/freedom-from-wires.html>

2.1.4.3 Red De Sensores

Una red de sensores, es aquella que está destinada y diseñada para determinando fin; donde sus transmisores o sensores tienen características específicas como sensibilidad, bajo consumo de energía, autoconfiguración sea como emisor o receptor y comunicación inalámbrica. Las redes de sensores se enfocan en múltiples aplicaciones como entornos industriales, domótica, medioambiente y otros.

2.1.4.3.1 Características de una Red de Sensores.

Entre las principales características, de una red de sensores tenemos:

✚ Topología Dinámica.

La topología de una red de sensores siempre se va a encontrar en constante movimiento, ya que esto le permite adaptarse para comunicar los datos que está tomando.

En la mayoría de una red de sensores, no se utiliza una infraestructura por lo que sus nodos al trabajar como emisores, receptores o enrutadores de información tienen la capacidad de recolectar los datos y transmitir a un ordenador, el cual puede ir recolectando la información de acuerdo al lugar que se encuentren.

✚ Comunicación Flexible

Las redes de sensores operan sea con una comunicación multisalto o de broadcast²⁸ para el paso de su información, la mayoría de estas trabajando bajo un mismo protocolo de enrutamiento.

(Wikipedia, Red de Sensores., 2011). El hardware utilizado en las redes de sensores, debe ser fabricado de la manera que ocupe el menor espacio posible y resistente a daños ya que se ve expuesto a usos limitados y en zonas de difícil acceso con bajos consumo de energía y con una gran fiabilidad en la adquisición de sus datos.

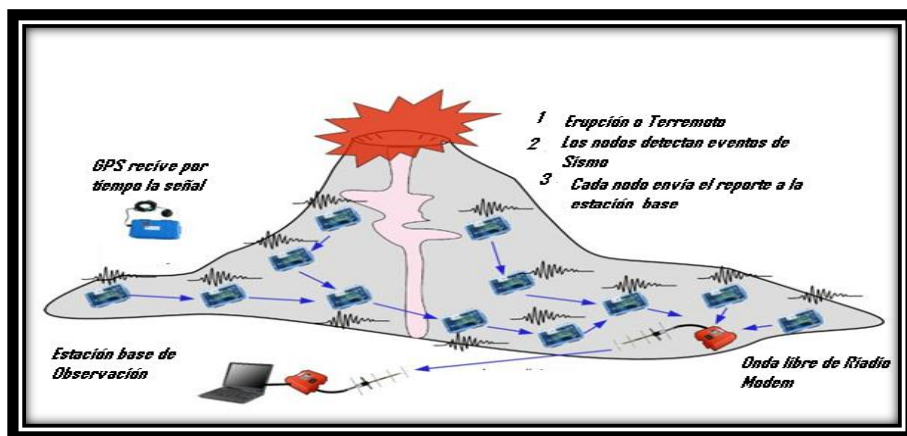


Figura 18. Red de sensores en la monitorización un volcán. Fuente: Freedom from Wires. (2011)

Recuperado de: <http://digitalcivilization.blogspot.com/2010/11/freedom-from-wires.html>

²⁸BROADCAST.- Transmisión de un paquete que será recibido por todos los equipos en una red

2.1.4.4 Red a implementarse.

La red a utilizarse en la monitorización de temperatura corporal, es una red en la cual sus nodos o transmisores deberán estar conectados a una central. Mezclando una red de malla inalámbrica de dos tipos de topologías la add-Hoc que permita la movilidad de los transmisores y de Infraestructura en la cual se va encontrar la central de monitorización; en otras palabras una red de sensores. Por los requerimientos analizados se debe utilizar una topología Multipunto – Punto como se muestra en la figura 2.5:

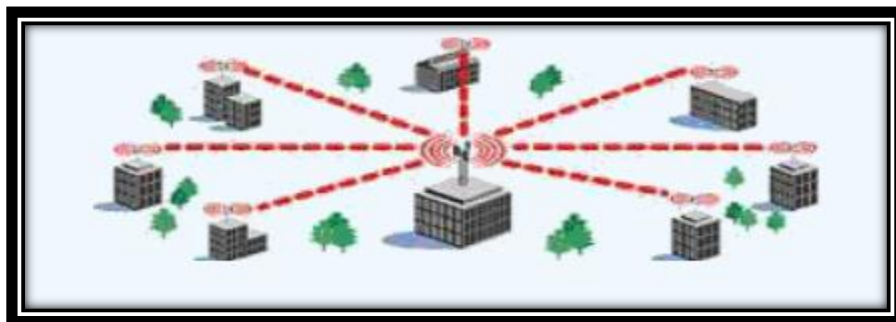


Figura 19. Red Multipunto – Punto *Fuente:* Freedom from Wires. (2011) Recuperado de:

<http://digitalcivilization.blogspot.com/2010/11/freedom-from-wires.html>

2.1.5 DISPOSITIVO DE TRANSMISIÓN INALÁMBRICA.

Una de las partes fundamentales, para transmitir los datos de temperatura corporal, es de forma inalámbrica, por lo que encontrar un dispositivo indicado que se adapte a las necesidades requeridas del diseño; se centra en analizar dispositivos que se encuentran en el mercado, y entre ellos están los módulos Bluetooth y los módulos Zigbee por su fiabilidad, fácil implementación, costo y más prestaciones en aplicación de monitorizaciones de variables, a continuación se muestra la tabla 2 cada una de sus características y necesidades de funcionamiento:

Tabla 2.

Características de módulos de Transmisión inalámbrica Xbee y Bluetooth

CARACTERÍSTICAS		MODULOS	
			
POTENCIA DE SALIDA		63mW(+18dBm) norteamericana 10mW(+10dBm) VersiónInternacional	versión 1mW (0 dbm) en 10m. Versión corto alcance. 1 W (20- 30 dbm) versión largo alcance
MÁXIMO TRANSMITIDO	RANGO	Interior/Urbanorange: 30 Al aire libre /RFde la línea derangode visión: 120 m	Hasta 10 m. Rango normal de 1 a 10 m.
VELOCIDAD TRANSMICIÓN	DE	250 Kbps.	723 Kbps
VOLTAJE ALIMENTACIÓN	DE	2.8 - 3.4 VDC	2.85 – 3.6 VDC
CORRIENTE TRANMISIÓN	DE	30 mA (@ 3.3 V)	40 mA
REDES SOPORTADAS		Puntoa punto, punto amultipunto, ypeer-to-peer	Estrella, multipunto
OPCIONES DE INTERFAZ		CMOS,UART	UART
VERSIÓN		802.15.4	802.15.1

Nota: Elaborado por Alexander Trejo.

Como se observa en los parámetros analizados en la tabla 2 de los dispositivos que se utilizan en redes de sensores, los equipos de tecnología Zigbee cumplen con las mayores características para la monitorización de temperatura corporal; a continuación se detalla como funcionan y se configurar este tipo de dispositivos para la implementación en una red de sensores.

2.1.5.1 Dispositivos Zigbee

La familia de los Zigbee, posee diferentes versiones de dispositivos según aplicaciones técnicas y zonas de cobertura como se muestra en la dirección www.digi.com. Para el caso de la monitorización de la temperatura corporal, se usa el módulo Zigbee series 2 que cumplen con los requerimientos de transmisión, cobertura y operación de frecuencia. La mayoría de los detalles técnicos y de fabricación de este dispositivo se puede observar en el Anexo 5.

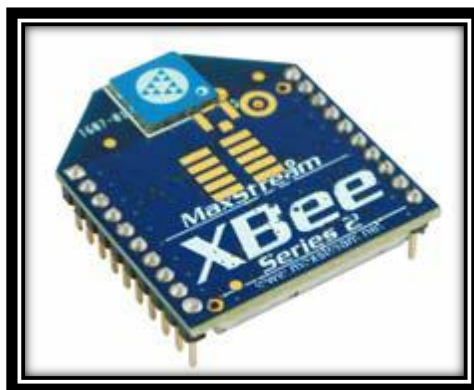


Figura 20. Dispositivo xbee series 2 Fuente: Digi Xbee (2011). Rescatado de:

<http://www.digi.com/products/wireless-wired-embedded-solutions>.

(Digi, Xbee series 2, 2011). El módulo Xbee, opera en la Frecuencia ISM de 2.4 GHz; en la cual la comunicación la hace a través de un canal de entre los 16 posibles canales que permite Zigbee. El alcance del dispositivo al momento de la transmisión de los datos, depende del tipo de antena a utilizarse como de la potencia del dispositivo; para este caso los series 2 que se muestra en la figura 2.6; que se encuentran en un rango de 40 metros en lugares cerrados y de 90 metros en lugares abiertos con línea de vista directa. En una red de sensores Zigbee teóricamente se puede formar hasta 65535 equipos, en otras palabras controlar mediante el protocolo Zigbee todos estos equipos dentro de una sola red.

Entre las principales necesidades que satisface este equipo son:

- Bajo Precio.
- Bajo Consumo de Potencia.

- Uso de bandas libres sin requerimiento de licencias.
- Fácil instalación y configuración.

2.1.5.2 Los Equipos Zigbee En Una Red De Sensores.

(OYARCE, 2010). En una red de sensores Zigbee, su parte de estructuración se centra en analizar dos partes fundamentales el uso del protocolo con el tipo de topología a utilizar sea esta punto a punto, multipunto, peer- to peer o redes complejas de sensores; y el segundo punto la configuración del equipo para su uso final. Una red de sensores zigbee básicamente, consta de 3 tipos de elementos. Un dispositivo coordinador, dispositivos routers y dispositivos finales como nos muestra la grafica siguiente:

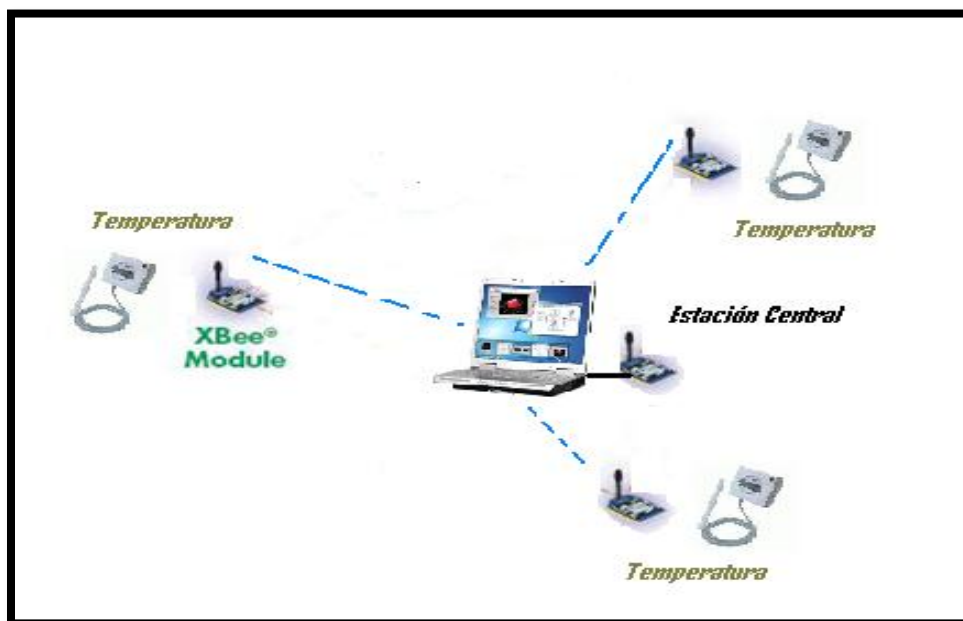


Figura 21. Red Zigbee.

2.1.5.2.1 El Coordinador.

(OYARCE, 2010). El nodo de red Coordinador, es el encargado de formar la red estableciendo el canal de comunicaciones y el identificador de red para toda la red

Zigbee. Permitiendo así unir los dispositivos routers²⁹ y dispositivos finales; operando sea como destinatario u origen.

2.1.5.2.2 Los Routers.

Este dispositivo opera, brindando la información de la red y determina la mejor ruta para poder transmitir el paquete de información. De igual manera puede retransmitir información de otros routers para dispositivos finales o coordinadores.

2.1.5.2.3 Dispositivo Final.

Los dispositivos finales, no poseen la capacidad de enrutar o retransmitir paquetes son configurados para determinada acción; tienen que realizar su petición o interactuar siempre con el nodo coordinador. Normalmente estos dispositivos van conectados a baterías ya que su consumo es bajo al no tener que realizar el enrutamiento de información.

2.1.5.3 Características De Los Dispositivos Xbee.

Los dispositivos Zigbee, poseen una dirección única como en las direcciones MAC de los dispositivos Ethernet³⁰. En el caso de los dispositivos Zigbee su dirección única es de 64 bits que vienen de fábrica, las cuales se utilizan en los algoritmos de ruteo de direcciones de 16 bits, es por eso que el número máximo teórico de elementos que se integran a una red Zigbee es $2^{16} = 65535$.

2.1.5.3.1 Circuito básico para el Xbee.

Un dispositivo Zigbee, debe cumplir un mínimo de conexiones requeridas como alimentación, circuito a tierra y pines de transmisión y recepción como se muestra en la figura 2.8 de a continuación:

²⁹ ROUTER.- Equipo encargado de direccionar paquetes de datos, en una red.

³⁰ETHERNET.- Estándar de red local para comunicarse entre computadoras.

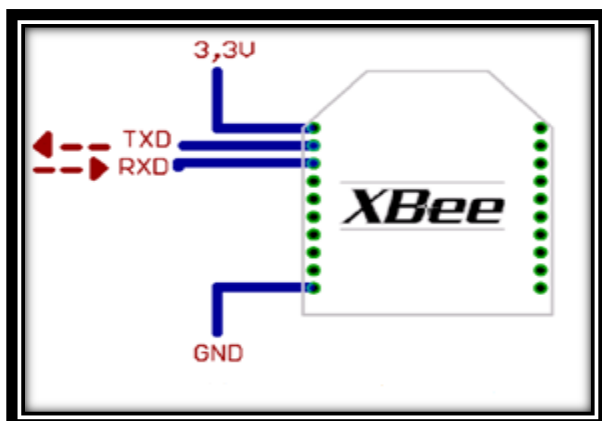


Figura 22. Alimentación de un dispositivo Xbee. Fuente: Andres Oyarce (2010). Guía del Usuario xbee.

Recuperado de: <http://www.olimex.com>.

La alimentación del circuito Xbee, se encuentra en un rango de 2.8 V a 3.4 V. los pines de transmisión y recepción del UART³¹ (TXD y RXD); deben ir conectados a un microcontrolador, o al puerto serial con el convertor de voltajes adecuado; que el fabricante recomiende. Una vez controlados estos parámetros se debe elegir el modo de operación del xbee.

2.1.5.4 Modos de Operación.

Los dispositivos Xbee, operan en 5 modos como se muestra en la figura a continuación:

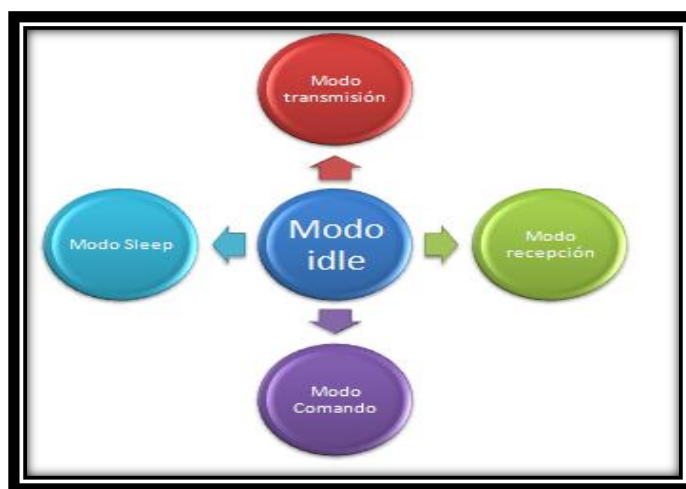


Figura 23. Modos de operación de un Xbee

³¹UART.- Controlador universal de transmisión y recepción serial.

2.1.5.4.1 *Modo Recepción/ Transmisión.*

Se trabaja en este modo, cuando el dispositivo envía o recibe paquetes, siendo la transmisión de paquetes de forma directa o indirecta. En cuanto la información es enviada de forma directa, la información llega inmediatamente al nodo destino; cuando se la realiza en forma indirecta la información es retenida por un periodo de tiempo y es enviada cuando el nodo destino la requiere.

En este modo de operación, es posible enviar la información en dos modos por Broadcast y Unicast, en donde el modo unicast recibe un paquete de confirmación de que la información solicitada llegó con éxito. En el modo Broadcast no se recibe paquetes de información de que la información llegó con éxito solo se espera la acción del dispositivo.

2.1.5.4.2 *Sleep Mode.*

En este tipo de modo el dispositivo, entra en bajo consumo de energía y la transmisión o recepción la realiza únicamente cuando la red lo requiere. Para entrar en este modo de operación, se necesita realizar la configuración del dispositivo en base a la asignación de pines y tiempos controlados por el microcontrolador.

El modo de sueño, hace posible que el módulo RF entre en un modo de bajo consumo de energía cuando no se encuentra en uso.

2.1.5.4.3 *Modo de Comando.*

Este tipo de operación, permite configurar al dispositivo mediante comandos AT para modificar sus parámetros como su dirección propia o dirección destino; su modo de operación u otros parámetros.

El ingreso de los comandos AT, se lo realiza mediante el hiperterminal de Windows o por el programa X-CTU o ya sea por un dispositivo que maneje comunicación UART.

2.1.5.4.4 *Modo Idle.*

Este tipo de operación se le denomina, cuando no se encuentra ni transmitiendo ni recibiendo ni ahorrando energía, ni modo de comandos entonces esta en modo IDLE.

2.1.5.5 DIRECCIONAMIENTO Y TOPOLOGÍA DE RED EN XBEE.

Los equipos xbee, permiten dos tipos de direccionamiento el direccionamiento de 16 bits y el direccionamiento de 64 bits, la diferencia radica principalmente en el número de equipos a controlar en la red. El direccionamiento de 16 bits se encuentra en el rango de direcciones (0x0 – 0xFFFE) y el direccionamiento de 64 bits se encuentra en el rango de (0xFFFF – 0xFFFE).

2.1.5.5.1 *Direccionamiento de 16 bits.*

Para poder utilizar un direccionamiento de 16 bits, se maneja en base a ciertos comandos y la restricción de algunas direcciones. Cumpliendo estas normas se utiliza direccionamiento de 16 bits; con una dirección menor a 0xFFFE junto con el comando MY e ingresando con los comandos AT.

Introduciendo el comando ATMY acompañado del número en formato hexadecimal, sin utilizar el “0x”. Por ejemplo si deseamos a un modulo asignarle la dirección 0x3B1F, ingresamos el comando ATMY3B1F Esto va acompañado de los comandos DL y DH los cuales deben ir asignados a 0 en un direccionamiento de 16 bits.

2.1.5.5.2 *Direccionamiento de 64 bits*

Para poder utilizar el direccionamiento de 64 bits, necesitamos desactivar el direccionamiento de 16 bits mediante comandos AT y habilitar el uso de 64 bits. En este tipo de direccionamiento no es posible definir la dirección de origen del módulo, ya que esta se asigna automáticamente y es el número serial que viene por defecto de fábrica; imposible de cambiar lo que da a la red una mayor asignación de equipos con la utilización de este direccionamiento.

En este tipo de direccionamiento, se debe utilizar el comando MY con la dirección 0xFFFF o el 0xFFFE encontrando una dirección destino con los comandos (DL+ DH), dentro del rango de una dirección de 64 bits; lo que corresponde al número serial de otro módulo formado por (SL + SH), el cual se consulta con el comando ATSL, quien nos devuelve los números seriales del módulo en formato hexadecimal.

2.1.5.5.3 Topología Punto a Punto.

Este tipo de conexión, es ideal cuando se requiere eliminar el cable serial y realizar control o envío de información en redes pequeñas. Para esto se debe configurar la dirección haciendo uso de los comandos MY y el DL así el Modulo de dirección MY se comunicará con el módulo de dirección DL arbitrariamente. En el siguiente grafico se muestra como se configuro una red eligiendo las direcciones arbitrariamente:

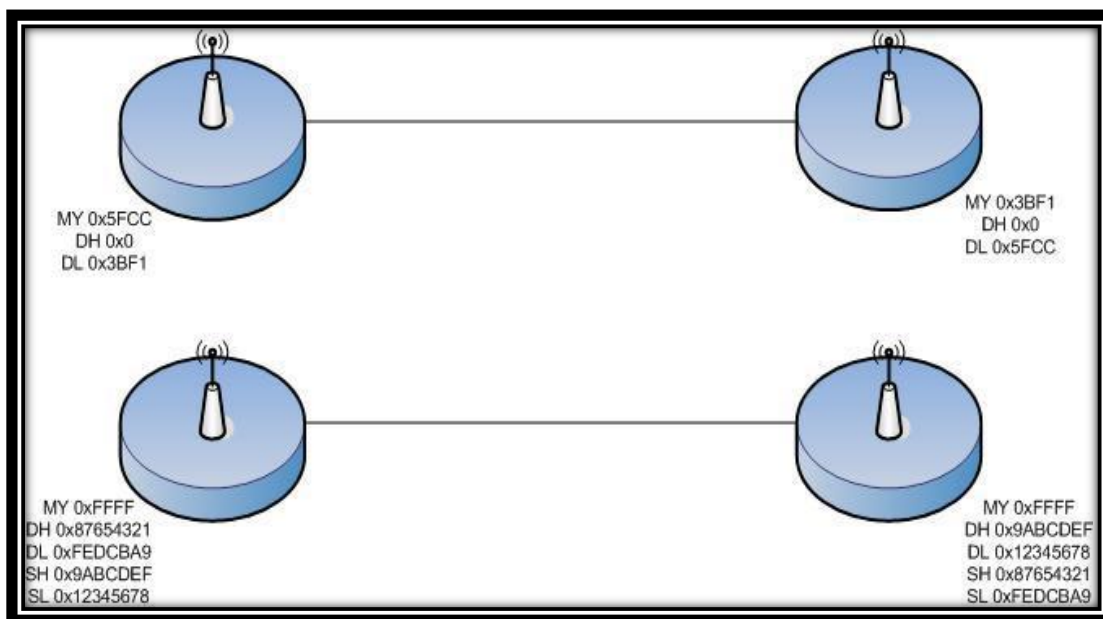


Figura 24. Alimentación de un dispositivo Xbee. Fuente: Andres Oyarce (2010). Guia del Usuario xbee.

Recuperado de: <http://www.olimex.com>.

En la primera red se observó, que se utiliza un direccionamiento de 16 bits y en la segunda configuración de red se utiliza un direccionamiento de 64 bits. Para que

este tipo de conexiones funcione, estas tienen que pertenecer al mismo identificador de red PAN ID y al mismo canal de comunicación.

2.1.5.5.4 Topología Punto a Multipunto.

Este tipo de configuración de red permite mayor prestaciones, como seguridad, fiabilidad de datos y más. Permitiendo tener un mayor control sobre la transmisión de datos, logrando así que la información transmitida llegue solo a ciertos dispositivos de manera controlada. Para este tipo de configuración de red necesitamos que la red se encuentre en el direccionamiento de 16 bits y que nuestro Pan ID o identificador de red pertenezca a la misma red. El comando que se utiliza para la configuración del PAN ID es el ATID y su rango va desde 0x0 a 0x FFFF.

Otro comando que debe ser de esencial configuración, es el tipo de canal que a utilizar según el protocolo IEEE 802.15.4 esta norma especifica que entre canal y canal debe existir un una diferencia de 5 MHz, comenzando desde la frecuencia base 2.405 GHz, hasta la Frecuencia 2.408 GHz como se muestra en la figura 2.11 siguiente:

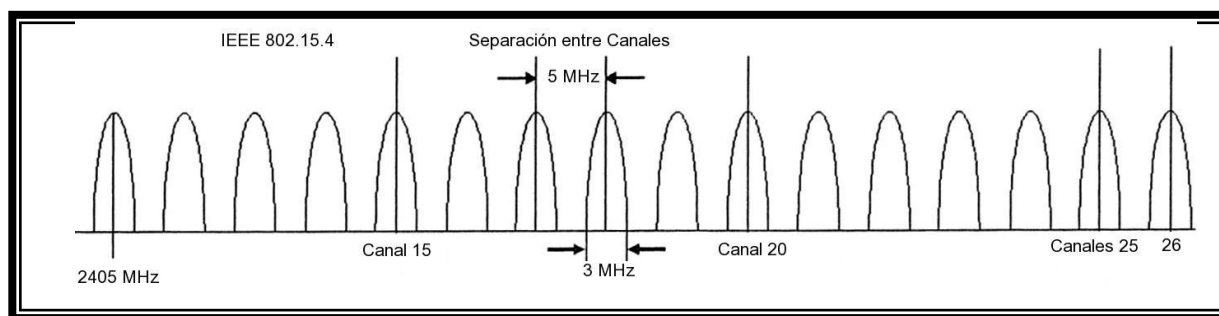


Figura 25. Alimentación de un dispositivo Xbee. Fuente: Andres Oyarce (2010). Guía del Usuario xbee.

Recuperado de: <http://www.olimex.com>.

Como la gráfica, se dispone de 16 canales pero la asignación se la realiza desde al canal 11 hasta el canal 26; para la asignación de la frecuencia central se utiliza la siguiente fórmula:

$$\text{Canal} = 2.405 + (\text{CH} - 11) \times 0.05 \text{ [GHz]}.$$

Donde CH es el equivalente al canal a escoger entre el rango de 11 a 26 esto se lo realiza en forma hexadecimal; por ejemplo se quiere utilizar el canal 15 se ingresa en su formato hexadecimal como ATCH10; a continuación se muestra una figura con la tabla de equivalencias y tipos de canales que se puede utilizar:

Frecuencia [GHz]					
canal	hexadecimal	inferior	central	superior	Comando AT
11	0x0B	2,4025	2,4050	2,4075	ATCH0B
12	0x0C	2,4075	2,4100	2,4125	ATCH0C
13	0x0D	2,4125	2,4150	2,4175	ATCH0D
14	0x0E	2,4175	2,4200	2,4225	ATCH0E
15	0x0F	2,4225	2,4250	2,4275	ATCH0F
15	0x10	2,4225	2,4250	2,4275	ATCH10
17	0x11	2,4325	2,4350	2,4375	ATCH11
18	0x12	2,4375	2,4400	2,4425	ATCH12
19	0x13	2,4425	2,4450	2,4475	ATCH13
20	0x14	2,4475	2,4500	2,4525	ATCH14
21	0x15	2,4525	2,4550	2,4575	ATCH15
22	0x16	2,4575	2,4600	2,4625	ATCH16
23	0x17	2,4625	2,4650	2,4675	ATCH17
24	0x18	2,4675	2,4700	2,4725	ATCH18
25	0x19	2,4725	2,4750	2,4775	ATCH19
26	0x1A	2,4775	2,4800	2,4825	ATCH1A

Frecuencia Base	2,405	GHz
-----------------	-------	-----

Figura 26. Alimentación de un dispositivo Xbee. Fuente: Andres Oyarce (2010). Guia del Usuario xbee.

Recuperado de: <http://www.olimex.com>.

De esta manera una vez configurados los principales parámetros de una red multipunto se vería de la siguiente forma:

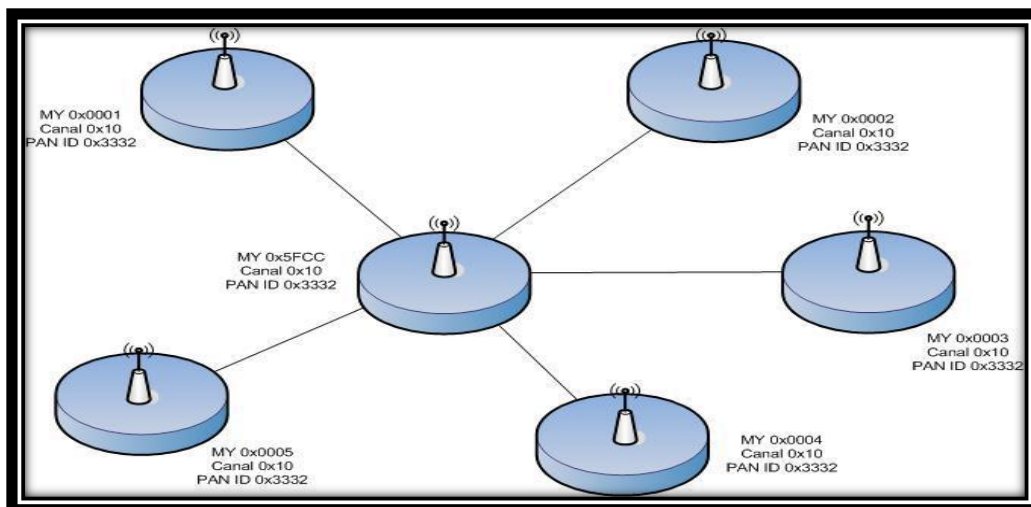


Figura 27. Alimentación de un dispositivo Xbee. Fuente: Andres Oyarce (2010). Guía del Usuario xbee.

Recuperado de: <http://www.olimex.com>.

Como se observa, en la figura cada uno de los módulos tiene una configuración de dirección MY distinta, pero su PAN ID y su canal son iguales para que se pueda realizar la comunicación dentro de una misma red y no exista errores de transmisión. En dispositivos más avanzados como los dispositivos XBEE PRO el reconocimiento de la red se la realiza automáticamente.

2.1.5.6 Arquitectura ZigBee

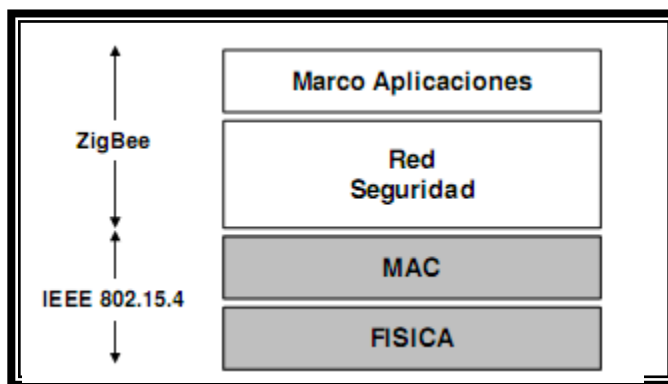


Figura 28.. Arquitectura de la tecnología Zigbee. Fuente: Infoab. Recuperado de:

http://www.infoab.uclm.es/labeledc/solar/Comunicacion/Wifi/Interconexion_de_redes.htm

Estableciendo como referencia el modelo del estándar OSI³², podemos observar la arquitectura estructurada en capas. Las dos primeras capas tanto la Física como la MAC vienen definidas bajo el estándar IEEE 802.15.4 y sus estructuras, las capas siguientes vienen dadas por el alianza Zigbee quienes trabajaron en adecuar la capa de red en base a las necesidades de los equipos y sus futuras aplicaciones, de igual manera las subsiguientes capas en sus presentaciones como promover el uso de la especificación y certificación de los equipos.

La capa física que en secuencia con la capa de acceso al medio, los realizan los servicios de transmisión de datos por el aire punto a punto, operando el estándar en la banda ISM, con radios en un espectro de dispersión en secuencia directa con tasas de transmisión de por lo menos 250 Kbps.

La capa de red basa su función principal en controlar el correcto uso del subnivel MAC y un buen funcionamiento en los métodos de inicio de red, con el enrutamiento de paquete hacia los otros nodos de red, filtrando, recibiendo y autenticando la información. Además la capa de red es la encargada de crear nuevas redes y asignar direcciones de los dispositivos.

El marco de aplicaciones mantiene la relación entre otros dispositivos simplificando el envío de datos bajo las normas ZigBee Alliance.

2.2 MODULO INFORMATICO DEL SISTEMA MEDICO

2.2.1 ARQUITECTURA DEL SISTEMA

2.2.1.1 *Cliente – Servidor.-*

(Wikipedia, Modelo cliente servidor, 2011). La edificación cliente-servidor es una forma de aplicación distribuida en el que las operaciones se reparten por los proveedores de servicios, denominados servidores, y los solicitantes, denominados clientes. Un cliente ejecuta un pedido al programa servidor, el cuál analiza y ejecuta la respuesta. Este tipo de aplicaciones se las puede observar en un computador que

³² OSI.-Modelo de interconexión de sistema abiertos que define la arquitectura de comunicación entre sistemas de conexión.

está configurado de punto a punto donde uno es el administrador y el otro equipo es el usuario normal.

(Wikipedia, Modelo cliente servidor, 2011) El proceso de trabajos en este modo, se encuentra realizado por los clientes y los servidores. Llevar los procesos de forma separada tanto para el cliente y el servidor es una forma lógica para ejecutar varios programas al mismo tiempo. En los cuáles se incluyen diversos tipos de servidores web, de archivos de correo etc.

Un sistema muy utilizado en el modo cliente servidor es la forma multicapa, donde el servidor se encuentra en diferentes programas que suelen ser accionados por varios computadores dando mayor robustez al sistema.

El modo cliente-servidor reemplaza a la arquitectura monolítica³³ en la que no hay estructuración ni a nivel físico ni a nivel lógico.

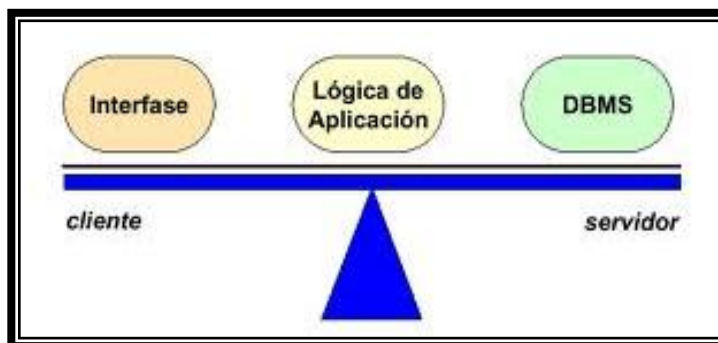


Figura 29. Interfaz servidor cliente. Fuente : Wikipedia (2011). Recuperado de:

<http://es.wikipedia.org/wiki/cliente-servidor>.

Las comunicaciones en una red cliente servidor todos los clientes están relacionados con el servidor y entre sí mediante privilegios brindados por el servidor logrando centralizar sus recursos y acciones a ejecutar. Provocando que la gestión se

³³Monolítica.- Perteneciente a un solo cliente.

realice únicamente desde el servidor el cuál ejecuta las acciones según la prioridad para la que fueron programadas tanto en archivos de uso público como restringido.

- *Cliente de la Aplicación.*-La interfaz ha sido desarrollada en Java
- *Servidor de Base de Datos.*-El servidor de BDD en este caso es MySQL, a continuación se analiza detenidamente cada uno de estos componentes [69].

2.2.1.2 BASE DE DATOS MYSQL 5

Es una plataforma de gestión, administración y ejecución de bases de datos relacional, multihilo³⁴ y multiusuario con un sinnúmero de instalaciones.

Dividiendo su adquisición en dos grandes potencialidades del software bajo la GNU GPL y para empresas privadas que desean hacer el uso comercial de cierto producto.

(Wikipedia, MySQL, 2011). MySQL es un controlador por dos partes, una pequeña parte por el creador del código y la mayor parte del MySQL por la empresa patrocinadora. A diferencia de Apache que es un software desarrollado por la empresa publica.

Teniendo como ventaja MySQL soporte y servicios al momento de la adquisición para comercialización de la misma o de un producto que derive de esta.

2.2.1.2.1 *Historia.*

(Wikipedia, MySQL, 2011). El lenguaje de consulta estructurado SQL³⁵ fue puesto en el mercado por primera vez por IBM en 1981, respaldado por ANSI lo cual consagró al lenguaje como un estándar para las bases de datos logrando desarrollarse en diferentes versiones : SQL:92, SQL:99, SQL:2003.

(Wikipedia, MySQL, 2011). Opensource MySQL establecida en Suecia en el 95 con sus fundadores David Axmark, y Allan Larsson, persiguieron su objetivo de que el software sea compatible con el estándar SQL manteniendo una gran velocidad,

³⁴ Multihilo.- Que puede escalar en diferentes etapas de relaciones o códigos.

³⁵ SQL.- Lenguaje de Consulta Estructurado

almacenamiento y fiabilidad. Naciendo con esto la posibilidad de sacar un sin número de versiones al mercado.

2.2.1.2.2 Versiones

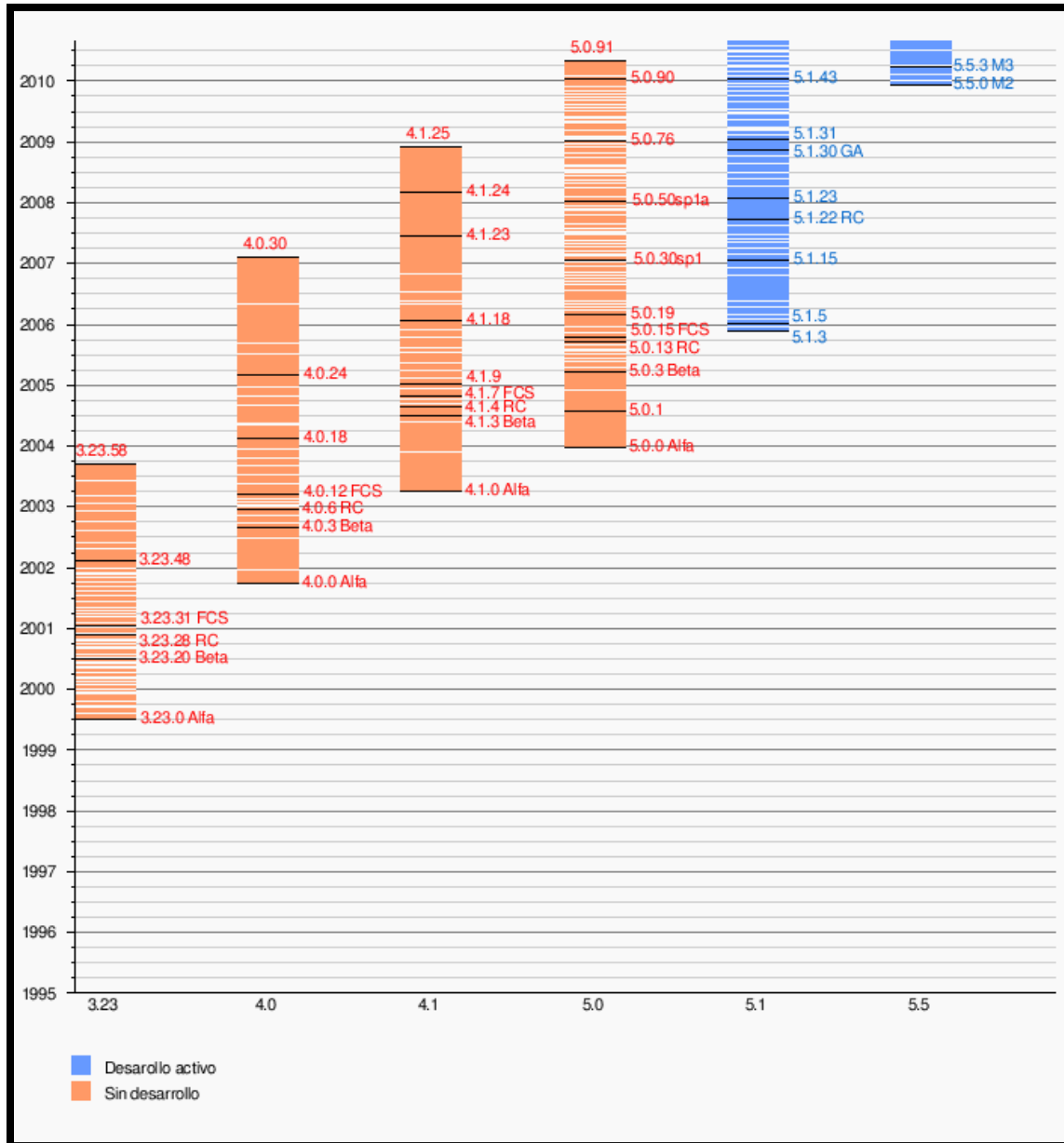


Figura 30. Versiones MySQL. Fuente: Wikipedia (2011). Recuperado de: <http://es.wikipedia.org/wiki/mysql>

2.2.1.3 Modelo Entidad – Relación

(Wikipedia, Modelo Entidad Relación, 2011). Los modelos entidad – relación denominado por ejemplificación E-R provee mayor calidad a los datos de un sistema de información expresando entidades con mayor lista de atributos y propiedades, brindando ventajas de interrelaciones entre sistemas.

Por lo cual el sistema de temperatura corporal entra en este tipo de modelo

2.2.1.4 Diseño de la Base de Datos

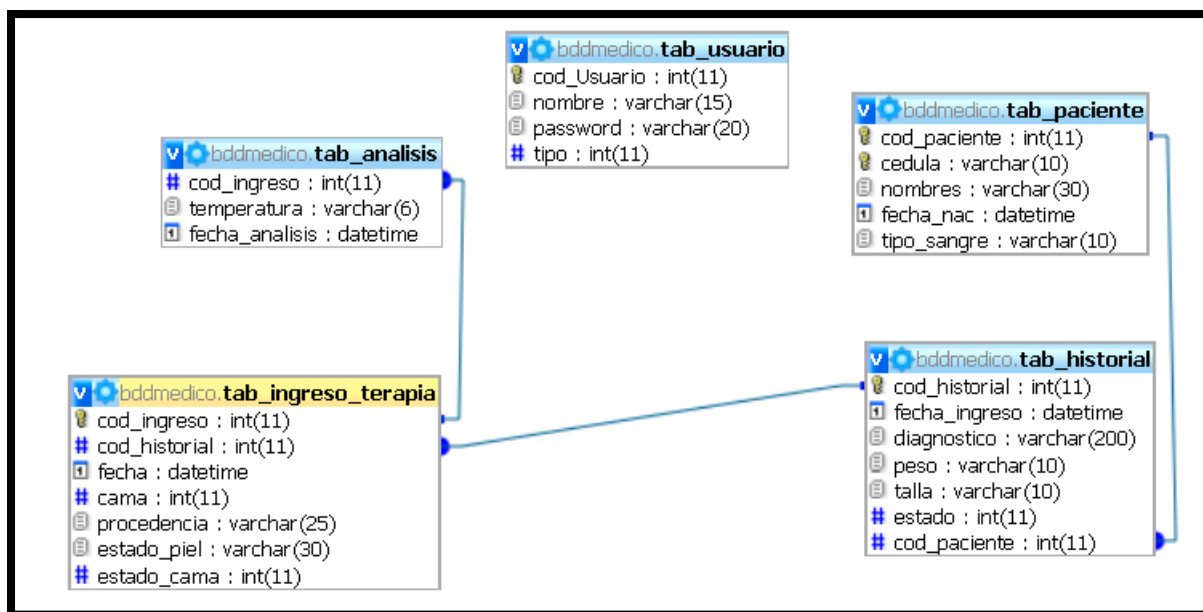


Figura 31. Estructura de Base de Datos

2.2.1.4.1 Diccionario de la Base de Datos

Tabla 3.

Datos iniciales del usuario.

Nombre de la Tabla: tab_usuario	
CAMPO	DETALLE
Cod_usuario	Código único del usuario
Nombre	Nombre del usuario de acceso al

	sistema
Password	Contraseña del usuario
Tipo	El sistema está abierto a manejar perfiles de usuario con este campo

Nota: Elaborado por Alexander Trejo.

Tabla 4

Datos de identificación del usuario.

Nombre de la Tabla: tab_paciente	
CAMPO	DETALLE
Cod_paciente	Código paciente en la Institución
Cédula	Cédula del paciente
Nombres	Nombres del paciente
Fecha_nac	Fecha de nacimiento del paciente
Tipo_sangre	El tipo de sangre del paciente

Nota: Elaborado por Alexander Trejo

Tabla 5

Datos de evaluación del usuario.

Nombre de la Tabla: tab_historial	
CAMPO	DETALLE
Cod_historial	Código del historial paciente
Fecha_ingreso	Fecha de Ingreso del paciente a la Institución.
Diagnostico	Síntomas con los que ingresa el paciente al hospital
Peso	Peso actual del paciente
Talla	Estatura actual del paciente
Estado	Condiciones de ingreso del paciente
Cod_paciente	Código paciente

Nota: Elaborado por Alexander Trejo

Tabla 6

Datos de terapia Intensiva del paciente

Nombre de la Tabla: tab_ingreso_terapia	
CAMPO	DETALLE
cod_ingreso	Código del ingreso al área de terapia intensiva.
cod_historial	Código del historial del paciente
Fecha	Fecha de Ingreso del paciente al área de terapia intensiva
Cama	Número de cama asignada al paciente
Procedencia	Campo obligatorio para realizar la hospitalización.
Estado_piel	Deberá ingresar el estado de la piel, quemada, puntos rojos, etc.
Estado_cama	Campo que nos indica si la cama esta ocupada o vacía.

Nota: Elaborado por Alexander Trejo

Tabla 7

Datos médicos del paciente.

Nombre de la Tabla: tab_analisis	
CAMPO	DETALLE
cod_ingreso	Código del Ingreso al área de terapia intensiva del paciente.
Temperatura	Temperatura emitida por medio del sensor colocado en el paciente hacia la computadora.
Fecha_analisis	Campo que permite ver la hora y fecha exacta de ingreso de la temperatura del paciente a la BDD.

Nota: Elaborado por Alexander Trejo.

2.2.1.4.2 Aplicación.

La aplicación contiene la siguiente estructura:



Figura 32. Interacción base de datos usuario.

Para el desarrollo de la Aplicación se utilizaron las siguientes herramientas.

- IDE de Programación: Netbeans 6.9.1
- Lenguaje de Programación: JAVA
- Librería Puerto Serial: GiovynetDriver.zip
- Librería de Conexión a la BDD: MySQL-conector-java-5.0.8

2.2.1.5 Ide de programación Netbeans 6.9.1

(PlanetNetbeans, 2011). NetBeans es una plataforma flexible de generación y entorno libre, desarrollada para diferentes lenguajes principalmente java con la posibilidad de adaptar diferentes ides y módulos para robustecer sus aplicaciones.

Basado en código abierto tiene una gran comunidad de adeptos y desarrolladores a nivel mundial, fortaleciendo la comunidad del software libre.

2.2.1.6 La Plataforma NetBeans

(Corporación, 2011)La plataforma NetBens brinda prestaciones de escritorio que permite desarrollar una lógica específica y flexible para cada aplicación basando sus principales características de la aplicación en.

- ✓ Gestión de las interfaces de usuario.
- ✓ Gestión de la configuración de usuario.
- ✓ Gestión de almacenamiento de diferentes tipos de datos.

- ✓ Gestión de ventanas y archivos.

2.2.1.7 Lenguaje de Programación JAVA

(Corporción, 2011). Microsystems³⁶ desarrolló el lenguaje de programación java orientado a objetos, alrededor de los 90, tomando parte de su estructura de las sintaxis de C y C ++ que son sus antecesores con modelos de objetos más simples y con secuencias de bajo nivel

Las aplicaciones Java se compilan bajo códigos máquina, denominados bytecode que permite la determinación de errores o terminación de la aplicación.

2.2.1.7.1 Filosofía

El lenguaje Java tiene su pilar en cinco objetivos principales:

- ✓ Su metodología es orientada a la programación de objetos.
- ✓ Permite diferentes aplicaciones de un mismo programa en múltiples sistemas operativos.
- ✓ Soportes en red y ayudas en líneas.
- ✓ Orientado a trabajar en entornos remotos.
- ✓ Versatilidad y fácil acople con otros lenguajes.

2.2.1.7.2 Orientado a Objetos

(Corporción, 2011). Un lenguaje orientado a objetos basa su estructura a la forma de programación y el diseño del lenguaje. Donde los diferentes datos usados tengan acceso a múltiples operaciones, dando como resultado a funciones y métodos que combinan entidades llamadas objetos. Un objeto encapsula el comportamiento del código y el estado de los datos. Un objeto encapsula el comportamiento del código y el estado de los datos; lo cual ofrece una base más estable para el diseño de un sistema.

2.2.1.7.3 Plataforma

La plataforma brinda una independencia de ejecución para cualquier tipo de hardware, con la capacidad de escribir una sola vez el programa sin necesidad de adaptar a nuevos dispositivos de ejecución.

La forma de independencia de la plataforma de java debe su gran éxito a las aplicaciones en ambientes, como servidores web, webmail, los java beans y sistemas empotrados con entornos java.

2.2.1.7.4 JAVA en Aplicaciones de Escritorio.-

(Corporción, 2011). En la actualidad se hallan diferentes aplicaciones gráficas de usuario basadas en java con el entorno de ejecución java (JRE), un componente virtual de los PC de usuario de los sistemas operativos que permite la ejecución y el uso de la plataforma java en cualquier tipo y lugar del mundo.

2.3 MEDICIÓN DE TEMPERATURA CORPORAL

La medición y monitorización de temperatura corporal en un paciente es sumamente importante al momento de dar un diagnóstico de su salud. Por lo tanto analizar sus diferentes formas de medición y características de la temperatura humana nos brindará una mayor comprensión de su funcionamiento.

2.3.1 TEMPERATURA CORPORAL.

(Erazo, 2009). La temperatura corporal en un ser humano es el cambio que experimenta su cuerpo a diferentes circunstancias como cambios climáticos, actividades físicas, estrés, enfermedades, edad y que se la relaciona a la medida del grado de calor que irradia el cuerpo.

(Lozada, 2010). La temperatura corporal normal de un ser humano varía ente los 36.5 – 37.5 grados Centígrados. Dependiendo de la edad del paciente como se muestra en la siguiente tabla.

Tabla 8

Temperatura corporal del Ser Humano en etapas normales

Edad	Temperatura en Grados Centígrados (°C)
Recién Nacido	36,1 - 37,7
Lactante	37,2
Niños de 2 a 8 años	37,0
Adulto	36,0 - 37,0

Nota: Elaborado a partir de investigaciones médicas de Lozada G (Noviembre 2011). Recuperado de: <http://www.slideshare.net/anama.krpio/temperatura-corporal>.

2.3.2 ALTERACIONES DE DE LA TEMPERATURA CORPORAL

La variación de la temperatura en un ser humano, es un signo vital de alta importancia en el cual se deben conocer sus diferentes denominaciones.

2.3.2.1 Hipotermia.

(Ferreira, 2010). Se le denomina hipotermia al cambio de temperatura que experimenta el ser humano al descender su temperatura corporal por debajo de los 36 °C, esto sucede a que el Hipotálamo³⁷ pierde su función de regulación; debido a diferentes trastornos sean ambientales o de salud, como por ejemplo; cuando el cuerpo humano es expuesto a temperaturas relativamente bajas y prolongadas lo que produce paralización de los músculos en el cuerpo, somnolencia³⁸ y por último el estado de coma; lo que inhabilita los órganos internos del cuerpo y el sistema nervioso central.

2.3.2.2 Fiebre.

(Tobón, 2010). Se le considera fiebre al aumento de temperatura corporal que sobrepasa el cuerpo al pasar los 37,8 °C, por lo cual el hipotálamo pierde su capacidad

³⁷ HIPOTÁLAMO.- Glándula del cuerpo humano situada en la parte interna del cerebro encargada de regular la temperatura del cuerpo humano.

³⁸ SOMNOLENCIA.- Es un exagerado comportamiento del cuerpo por el sueño lo que provoca una buena coordinación de los diferentes partes internas.

de regulación esto se debe principalmente a trastornos de salud que sufre el cuerpo por diferentes tipos de patologías infecciosas en base a virus, bacterias, enfermedades respiratorias, hereditarias, intoxicaciones y cambios exagerados del clima de lo cual produce síntomas como; sudoración, escalofríos, dolores de cabeza, mareos, vómitos, somnolencia ataques epilépticos y por último la muerte. La variación de la temperatura se divide en las siguientes:

- ✚ Fiebre Leve 38 a 38,5 °C.
- ✚ Fiebre Moderada 38,5 a 39 °C.
- ✚ Fiebre alta 39 a 41 °C.

2.3.3 FORMAS DE MEDIR LA TEMPERATURA DEL CUERPO HUMANO.

La medición de la temperatura corporal se la puede realizar en 4 partes del cuerpo humano; la axila, la boca, el recto y el oído a continuación; se describen cada una de estas formas de medición:

2.3.3.1 Temperatura axilar.

Este es una de las formas de medición de temperatura más cómoda, con la cual el paciente no tiene ningún comprometimiento moral. La única desventaja frente a las demás formas de medir la temperatura es que tiene 0.1°C de error por lo cual no es tan segura.

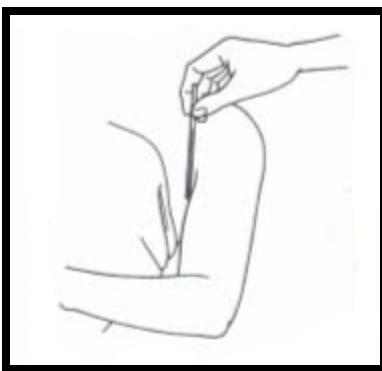


Figura 33 Temperatura axilar. *Fuente:* Fistera (2010). Recuperado de:

<http://www.fisterra.com/material/tecnicas/temp/temp.asp>

2.3.3.2 Temperatura Rectal.

Es una de las mediciones mas incomodas en la cual el paciente puede tener daños físicos como morales; pero al mismo tiempo es la forma de medición más exacta ya que tiene un contacto más interno con el paciente. No se recomienda este tipo de medición de temperatura a pacientes que sufran de trastornos rectales como cirugías en extremidades inferiores.



Figura 34. Temperatura rectal *Fuente:* Fistera (2010). Recuperado de:

<http://www.fistera.com/material/tecnicas/temp/temp.asp>

2.3.3.3 Temperatura Oral.

Es la forma de medición más utilizada ya que no provoca ningún tipo de incomodidad al paciente ni moral ni físico. De igual manera es muy fiable y fácil de aplicar.

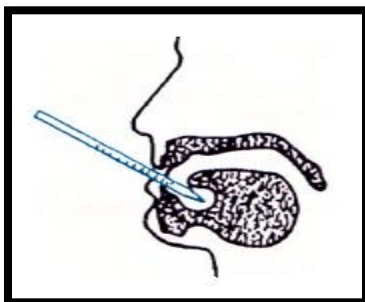


Figura 35. Temperatura oral *Fuente:* Fistera (2010). Recuperado de:

<http://www.fistera.com/material/tecnicas/temp/temp.asp>

2.3.3.4 Temperatura en el Oído.

La medición de la toma de la temperatura en el oído no es muy utilizada en la mayoría de los casos; ya que se la aplica es en los bebés porque aún no entienden ni obedecen ordenes como un niño o adulto.

2.3.4 SITUACIONES QUE PERJUDICAN LA TOMA DE LA TEMPERATURA CORPORAL

- ✚ **Edad.-** (Espinal, 2010). Tanto en los niños que acaban de nacer como en los ancianos se presentan problemas con la estabilización de la temperatura, ya que en el primer caso no tienen la completa madurez de sus órganos para tener datos exactos. En cambio en los ancianos no presentan un correcto funcionamiento de sus órganos por lo que suele disminuir su temperatura corporal a 36°C.
- ✚ **Horas del día.-** (Espinal, 2010). Al transcurso del día la variación de temperatura suele variar en 1.5 °C a los parámetros normales, dependiendo del lugar en que se encuentra la persona.
- ✚ **El sexo.-** (Espinal, 2010). Este factor radica principalmente en las mujeres ya que en el ciclo menstrual su temperatura varía en un rango entre 0.3 – 0.5 °C.
- ✚ **El estrés.-** Las diferentes emociones a las que se enfrenta un ser humano hace que su sistema nervioso actúe elevando su temperatura.
- ✚ **El ejercicio físico.-** La actividad física hace que el metabolismo del ser humano incremente su temperatura corporal.

2.4 Diseño del Módulo electrónico de medición de temperatura corporal.

El diseño del módulo electrónico a efectuarse, permitirá medir la temperatura corporal de los pacientes de terapia intensiva en un rango de 33 °C a 40 °C con la capacidad que le paciente pueda trasladarse de un lugar a otro dentro de las instalaciones de terapia intensiva del Hospital “San Vicente De Paul” el diseño del módulo se lo estructura en el siguiente capítulo.

CAPITULO III

3 DESARROLLO DE LA APLICACIÓN

En este capítulo se centra en estructurar el sistema de monitorización de temperatura corporal, junto con un estudio de los requerimientos técnicos, características y especificaciones de los dispositivos y sistema operativo a que mejor se adapten al sistema.

El sistema de monitorización de temperatura corporal consta de dos partes fundamentales, las cuales están constituidas por el sistema de transmisión de datos y el sistema de recepción de datos, a continuación se detalla cada uno de los sistemas.

3.1 TIPOS DE SISTEMA OPERATIVOS UTILIZADOS PARA EL DISEÑO DE MONITORIZACIÓN DE TEMPERATURA CORPORAL.

Es de vital importancia, que un sistema de monitorización cuente con una interfaz gráfica, la cual muestre los resultados que se están procesando para poder tomar las respectivas precauciones en la zona de terapia intensiva. Existen en el mercado global de las tecnologías diferentes tipos de software que permiten el control y la aplicación de sistemas de monitorización, a continuación se detalla en forma general algunas opciones, las cuales se adaptan al sistema de monitorización de temperatura.

3.1.1 SOFTWARE LIBRE.

(*Wikipedia, Software Libre* , 2011). El software libre es un término que se denomina, para identificar libertad de utilización de un software, donde las personas que lo utilizan pueden reproducirlo, aprenderlo, modificarlo y redistribuirlo con libertad, concentrándose en las siguientes reglas establecidas por su creador.










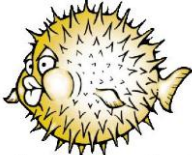
- ✚ La autonomía de producir el software para cualquier tarea específica.
- ✚ La autonomía de aprender cómo funciona el programa, y modificarlo para acoplarlo a la necesidad requerida.

- ✦ La autonomía de redistribuir copias modificadas o sin modificar, que puedan ayudar a otros desarrolladores del software libre.
- ✦ La autonomía de compartir nuevos desarrollos al corazón del software para poder publicar mejores actualizaciones.

En el mercado existen diferentes, tipos de denominaciones que utilizan software libre a continuación mostramos en la tabla algunas de ellas:

Tabla 9

Tipos de Sistemas operativos de Software Libre.

DESARROLLADORES		DISTRIBUCIONES	
GNU		DEBIAN	
LINUX		UBUNTU	
KDE		CENTOS	
MySQL		OPENSOLARIS	
APACHE		OPENBSD	

Nota: Datos emitidos de la sociedad CNU. Fuente: GNU Operating System. Fuente: <http://www.gnu.org/philosophy/free-sw.es.html>.

3.1.2 SOFTWARE CON LICENCIA.

Una de las alternativas, para poder desarrollar la base de monitorización de temperatura corporal es el Software con licencia.

El término de Software licenciado se refiere a un contrato que se realiza entre el creador y el que adquiere el producto, comprometiéndose a cumplir las cláusulas y condiciones que el fabricante establezca:

- ✚ **Desarrollador:** El software creado por el autor, tiene la instancia de tener la respectiva licencia como dueño y ejecutante de la misma.
- ✚ **Derechos de reproducción:** Es el ente natural o jurídico, el cual adquiere derechos sobre la licencia del software que son otorgados por su creador para reproducirlos o mejorarlos.
- ✚ **El autor,** tiene la potestad de establecer un tiempo límite en el cuál se pueda utilizar la utilización de su software.
- ✚ **El autor** se rige a las leyes que se establecen en cada gobierno. De igual forma a recibir la protección del mismo en caso de que su producto o software este siendo utilizado ilegalmente.

A continuación se describe algunas empresas las cuales desarrollan una serie de productos que trabajan bajo licencia:



Figura 36. Marcas de Programas con licencia.

3.1.2.1 Adobe.

Es una empresa líder en el mundo del diseño gráfico mostrando su jerarquía en mercados de ventas y promociones.

3.1.2.2 Aspel

Su principal objetivo es proveer dispositivos informáticos para la dirección de empresas en aspectos de costos y finanzas como facturas, cuentas, inventarios, contabilidad general y estrategias de producción.

3.1.2.3 Athenea.

Es una aplicación muy completa que se centra en la administración de sistemas escolares, mejorando sus aspectos desde la imagen hasta la interacción de padres de familia fortaleciendo así un mejor aprendizaje.

3.1.2.4 Avg

Es una de las empresas líderes en el mundo web de protección de virus informáticos, basando su estrategia en estar en la mayoría de los ordenadores haciendo fácil su descarga e instalación.

3.1.2.5 Hob Rd Vpn

Es una empresa que brinda la ventaja de trabajar al usuario en forma remota desde su hogar, o desde las sucursales de las empresas con gran seguridad y servicios de red.

3.1.2.6 Imlock

Es una empresa que desarrolla programas de prohibición y restricción de aplicaciones web como facebook, twitter, kazza, Lime Wire etc. Aumentando los rendimientos del sistema.

3.1.2.7 Ingressio

Es una empresa líder en controles biométricos de asistencia y control accesos de empleados, utilizando detección por huellas digitales determinando así horarios de salidas y entradas de los empleados.

3.1.2.8 Lojack

Es uno de los mejores sistemas en localización y recuperación de autos, ampliando su segmento de mercado en localización de portátiles trabajando su mayor parte con sistemas satelitales.

3.1.2.9 Microsoft

Es la empresa líder y más exitosa en sistemas operativos teniendo su software instalado en la mayoría de ordenadores del mundo; instalando sus licencias a empresas privadas, de servicio, instituciones públicas y educativas.

3.1.2.10 Symantec

Es un líder mundial en seguridad de ordenadores con el mejor software antivirus con protección personalizada y acceso remoto.

3.1.2.11 Versadial

Es un software especializado en grabación y administración de llamadas mejorando y midiendo la calidad de servicio.

3.1.2.12 Vmware

Es un software ideal en la virtualización, realizando la utilización de diversos sistemas operativos sea en un ordenador o servidores.

3.1.3 SOFTWARE DE CONTROL INDUSTRIAL

En sistemas de control y monitoreo de variables existen empresas especializadas en esta área. Las cuales trabajan bajo la adquisición de licencias y

paquetes de software en diferentes campos de la tecnología; a continuación citamos algunas dos de ellas.



Figura 37. Marcas de Programa de control industrial. Fuente: Henry Roncancio (2008). Recuperado de: <http://es.scribd.com/doc/6539343/016-Tutor-Lab-View>.

3.1.3.1 Labview

LabVIEW (Laboratory Virtual Instrument Engineering Workbench), es un código gráfico de programación especializado en la adquisición de datos, la instrumentación y control. Permitiendo al usuario realizar interfaces graficas de una forma interactiva amigable y segura. LabVIEW está orientado a trabajar en diagramas de bloques en una infraestructura de ingeniería de colocación de herramientas para determinadas acciones, de igual manera es tan flexible que permite trabajar con otros lenguajes de programación de otras aplicaciones, fortaleciéndose con la integración de hardware con tarjetas de adquisición, medición y procesamiento de datos.

3.1.3.1.1 Aplicaciones de LabVIEW

Labview posee varias aplicaciones especialmente en sistemas de medición y monitoreo de procesos, variables y control de las mismas; desplegando su campo de acción en control industrial, laboratorios de universidades, sistemas de transportación. De igual forma Labview es aplicado en procesamiento digital de señales, sistemas biomédicos, manipulación de imágenes, audio, filtros digitales, entre otras.

3.1.3.1.2 ExpeMed

Es un software utilizado en aplicaciones médicas, el cual sirve como una base de datos en la que se consulta, organiza, y guarda datos de los pacientes con acceso confidencial como consultas, tratamientos dentales, padecimientos, alergias, tipo de

sangre, medicamentos, vacunas etc. Facilitando así el trabajo del personal médico. Brindando información importante y oportuna la que puede salvar la vida de una persona.

(Expemed, 2011). Un dato importante que ofrece este tipo de sistema, es que el paciente puede tramitar la donación de órganos y tejidos de una forma legal y respetando su voluntad.

3.1.4 REQUERIMIENTOS Y ESPECIFICACIONES TÉCNICAS DEL MÓDULO ELECTRÓNICO DE MEDICIÓN DE TEMPERATURA CORPORAL.

La medición de temperatura corporal mediante un módulo electrónico, requiere de bastantes aspectos a tomar en cuenta. Para ello es de vital importancia conocer sus diferentes requerimientos técnicos a continuación se muestra la figura 2.19 donde se da un esquema de sus diferentes etapas:

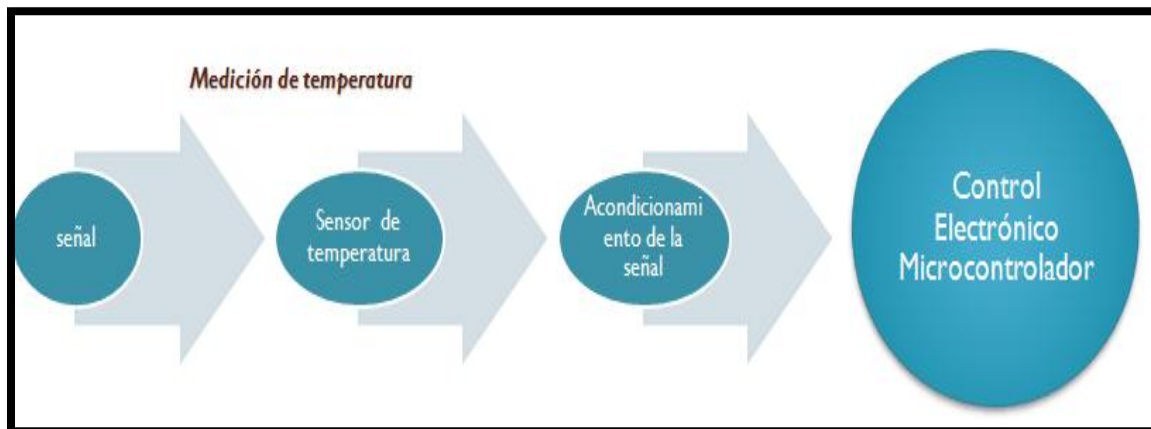


Figura 38. Etapas del diseño del transmisor de temperatura.

La tecnología presenta varios dispositivos y elementos que se utilizan para la implementación de un transmisor de temperatura corporal, por lo cual es importante conocer el funcionamiento, y características de estos dispositivos; para poder escoger los que mejor se adapten al diseño y se encuentren en el mercado.

3.1.4.1 Sensor de temperatura.

Adentrarse en el mundo de los sensores de temperatura es encontrar un sin número de dispositivos utilizables en diferentes campos como la industria, el medioambiente, la economía, la medicina y otras. Como se muestra una tabla de clasificación de los sensores de temperatura más utilizados.

Tabla 10.

Dispositivos de medición de temperatura.

DISPOSITIVOS DE MEDICIÓN DE TEMPERATURA	
Eléctricos	Termocuplas, Termorresistencias, Termistores, Diodos, Transistores, etc.
Mecánicos	Sistemas de Dilatación Termómetros de vidrio con líquidos, Termómetros bimetálicos
Radiación Térmica.	Pirómetros de radiación: Total (banda ancha), Espectral o radiación parcial, Fibra óptica, etc.
Varios	Indicadores de color, indicadores pirométricos, cristales líquidos, etc.

Nota: *Elaborado por Alexander Trejo*

Conocer el funcionamiento y características de los sensores de temperatura nos facilitara la elección del sensor a implementar; el cual permite mayor adaptabilidad y precisión para la aplicación. A continuación se describe algunos dispositivos aplicables al sistema de monitorización de temperatura corporal de temperatura para tener una mayor comprensión de estos:

3.1.4.1.1 RTD

(Omega, 2010). Resistencia Detectora de Temperatura (**RTD**).- Son dispositivos que varían su resistencia en función de la temperatura en los cuales sus principales materiales de construcción suelen ser el cobre, el níquel, o el platino. Estos sensores al sentir un cambio de la temperatura experimentan una elevada agitación térmica, y por

ende aumentan su resistencia estableciendo así una elevada temperatura y mayor resistencia.

✚ RTD características Generales.

Los Rtds tienen características generales en cuanto a margen de cobertura, y buena linealidad, con un amplio rango de temperaturas y gran resistividad pero con la desventaja de una baja sensibilidad; y un costo alto en el mercado.

La variación de la resistencia que presenta los Rtds se ve expresada en una ecuación polinómica en forma lineal:

$$R = R_0 \cdot (1 + \alpha \cdot \Delta T).$$

Donde:

R_0 es la resistencia a la temperatura de referencia T_0

ΔT es la desviación de temperatura respecto a T_0 ($\Delta T = T - T_0$)

α es el coeficiente de temperatura del conductor especificado a 0 °C.

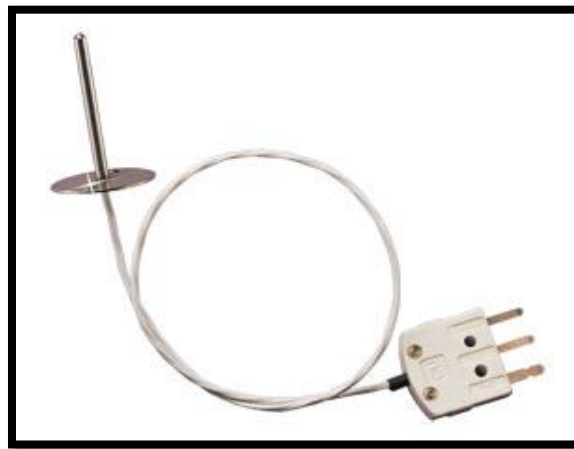


Figura 39. RTD medidor de temperatura. Fuente: Scribd (2011). Recuperado de:
<http://www.scribd.com/doc/52643532/44/Microcontrolador-18F2550>

3.1.4.1.2 Termistores.

Los termistores son sensores de temperatura, que presentan una resistencia sensible a la variación en función de los cambios de temperatura. Se componen de una mezcla de óxidos metálicos como hierro, níquel, cobalto formando un semiconductor con coeficiente térmico elevado. Los termistores presentan una característica especial que poseen un coeficiente de temperatura positivo (PTC) y un coeficiente de temperatura negativo (NTC). Para los termistores NTC, al elevarse la temperatura, se producirá una menor resistencia al flujo de electrones. Para los termistores PTC al elevarse la temperatura presenta una mayor resistencia al flujo de electrones.

Los termistores sirven para la detección, medición de temperatura tanto en líquidos, gases y sólidos especialmente diseñados para trabajar en lugares pequeños en cualquier tipo de medio con grandes ventajas como una gran sensibilidad, buena estabilidad y una precisión. Con algunas desventajas como alinealidad, rango limitado de temperatura, y una autocalentamiento.

La diferencia principal entre los RDT y los termistores es su no linealidad concentrándose en características matemáticas de formas hiperbólicas, que varían en pequeños incrementos en su relación entre la temperatura y la resistencia mediante parámetros como se muestra en la siguiente ecuación:

$$R_T = R_o e^{[\beta(\frac{1}{T} - \frac{1}{T_0})]}$$

Donde:

T₀: Es la temperatura de referencia para el termistor.

R₀: Es la resistencia de referencia para la temperatura.

B: Es un parámetro conocido como la temperatura característica del material y se le considera constante dentro el margen de temperatura de estudio.



Figura 40. Termistores para medir de temperatura. Fuente: Mecatrónica Actual (2010). Recuperado de:<http://www.mecatronicaatual.com.br/secoes>.

3.1.4.1.3 Semiconductores de temperatura.

Este tipo de semiconductores varía su conectividad eléctrica, de acuerdo al cambio de temperatura. La mayoría de estos dispositivos vienen encapsulados para ser utilizados en ambientes de rangos limitados de temperatura. Entre sus características tenemos que son fáciles de implementar, bajo costo, buena linealidad, con mínimos requisitos de acondicionamiento y una velocidad de respuesta aceptable con la desventaja que no tienen una buena reacción a los cambios bruscos de temperatura.



Figura 41. LM35 sensor medidor de temperatura. Fuente: Consultaing (2010). Recuperado de:
<http://consultaing.blogspot.com/2009/10/sensor-de-temperatura-lm35.html>.

3.1.4.1.4 Pirómetros.

Existen aplicaciones en las cuales no se requiere tener un contacto para la medición de temperatura, como por ejemplo cuando el área a medir tiene difícil acceso, o se mueve mucho.

(Pirómetros de Radiación y Ópticos, 2010). Pirómetro es un término que se deriva de un compuesto griego Pyros, que significa “Fuego” y Metron que es “Medida”; donde su principal objetivo es sensar la temperatura en forma de radiación, a través de la propagación de la energía, a la emisión continua de energía irradiada por la superficie de algunos cuerpos, que se encuentran en una temperatura superior a los 0⁰ grados kelvin. Existen diversos tipo de pirómetros entre ellos tenemos, los de radiación, los infrarrojos y los ópticos en los cuales los pirómetros por radiación trabajan con muestreos de la radiación total de calor, emanado por una superficie lo cual es el sensor térmico determinando así su temperatura. El pirómetro óptico funciona comparando la emisión de la luz de calor que emite un cuerpo con su fuente de luz estándar, es decir con diferentes espectros visuales lo que le permite determinar la temperatura del objeto. Y por último el pirómetro infrarrojo tiene el mismo funcionamiento del óptico de comparación de espectros solo que este lo realiza en el segmento infrarrojo.

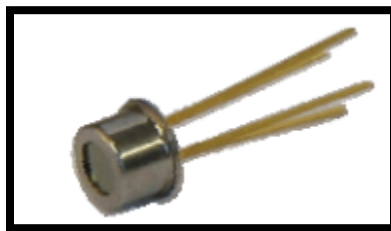


Figura 42. Termopila MLX90615. Fuente: Rincón del Vago (2010). Recuperado de:
<http://html.rincondelvago.com/pirometros-de-radiacion-y-opticos.html>

3.1.4.1.5 Termocuplas.

Son sensores de temperatura, los cuales tienen dos metales de materiales diferentes que al unirlos y aplicarles una gradiente de temperatura produce circulación de

corriente y diferencia de voltajes, a lo que se le conoce como efecto seebek lo que me permite obtener voltajes en el orden de los mV que varía con el aumento de temperatura. Hay 7 tipos de termocuplas que vienen identificadas por letras elaboradas por la Sociedad Instrumental de América (ISA). Las cuales especifican, tipo de materiales, rangos de temperaturas para ser utilizado en las diferentes aplicaciones industriales o de control como se muestra en la figura 2.24.

Tc	Cable + Aleación	Cable - Aleación	°C	Rango (Min, Max) mV	Volts Max
J	Hierro	cobre/nickel	(-180, 750)	42.2	
K	Nickel/cromo	Nickel/alumnio	(-180, 1372)	54.8	
T	Cobre	cobre/nickel	(-250, 400)	20.8	
R	87% Platino 13% Rhodio	100% Platino	(0, 1767)	21.09	
S	90% Platino 10% Rhodio	100% Platino	(0, 1767)	18.68	
B	70% Platino 30% Rhodio	94% Platino 6% Rhodio	(0, 1820)	13.814	

Figura 43. Tabla de los tipos de TermocuplasFuete: Rincón del Vago (2010). Recuperado de:
<http://html.rincondelvago.com/pirometros-de-radiacion-y-opticos.html>.

3.1.4.2 Termistores Utilizados En Biomedicina.

Como se analizó en los diferentes sensores de temperatura los termistores, tienen características de tener una alta sensibilidad, así como un costo no muy alto por lo que analizaremos diferentes tipos de termistores utilizados en la biomedicina para aplicarlos en la de medición de temperatura corporal.

3.1.4.2.1 Sensor YSI 427.



Figura 43. Sensor YSI. Fuente: Catalogo YSI 427 (2010). Recuperado de: <http://www.sorevan.com>

Un sensor YSI 427 es una sonda de temperatura que contiene termistores utilizados en aplicaciones biomédicas, para medir la temperatura corporal de los cuerpos.

Se destacan por poseer una buena linealidad en el rango de temperatura establecido con una buena respuesta de precisión. Fabricado por el líder mundial en control biomédico Yellow Spring Instruments (YSI); a continuación se muestra una tabla de sus principales características:

Tabla 11

Características YSI 427.

<i>Características técnicas del Termistor YSI 427</i>	
Fabricante	Yellow Spring Instruments
Rango de Acción	0° C a 60° C
Precisión	+/-0,1 de 25° C a 40° C; +/-0,2 de -1° C a 60° C
Tiempo de respuesta	0.8 segundos / °C
Terminal	Conector estándar “ (Roldan, 2009) (Erazo, 2009) (Society, Septiembre) (Society, Septiembre) Phone plug” 0.25 pulgadas, compatible con muchos monitores de pacientes.

Nota: Elaborado a partir de investigaciones médicas Instituto YSI (Noviembre 2010). Recuperado de: <http://www.sorevan.com>

3.1.4.2.2 Sensor de Piel Ellab.



Figura 44. Sensor de piel ellab. Fuente: Pmedical (2010). Recuperado de:
<http://www.pmedicalonline.net/servlet/the-Philips-Medical-cln-6.-Sondas-Temperatura/Categories>.

Este tipo de sensor de piel, se lo utiliza como una sonda superficial para medir la temperatura tanto en las manos, piel del cuerpo y pies, existiendo dos categorías para adultos y niños a continuación se realiza una tabla de sus características:

Tabla 12

Características sensor de piel ellab.

Características	Sensor de Piel Ellab
Termistor	Termistor sencillo de 2 hilos, 2252 ohms ³⁹ a 25,0°C
Tipo-Adulto	Un diámetro de 18mm. Con un espesor de 5.8mm
Tipo-Bebe	Un diámetro de 10mm. Con un espesor de 5.0mm
Material Adulto	Placa de Acero inoxidable con un casco de polioximetileno ⁴⁰
Material Bebe	Acero Inoxidable
Exactitud	+/- 0,1°C en un rango de 15 a 45 °C
Tipo de Cable	Cable de silicona activo que no produce irritación

Nota: Elaborado a partir de las investigaciones realizadas por Ellab.

³⁹Ohms.- Tipo de unidad utilizada para identificar a las resistencias.

⁴⁰Polioximetileno.- Es un termoplástico cristalino que me permite gran rigidez.

3.1.4.2.3 Sensor de Temperatura reusable.



Figura 45. Sensor de piel reusable. Fuente: Ellab (2010). Recuperado de:
<http://www.ellab.com/Default.aspx?ID=386>.

Existen sensores de temperatura, que tienen un tiempo de vida corto los cuales se los puede cambiar luego de haber realizado una función. Su principal ventaja es su bajo costo y su facilidad de implementación entre sus principales características tenemos:

Tabla 13

Características Sensor de Desechable.

Sensor Desechable Philips 21091^a

Fabricante	Phillips
Rango de Acción	25° C a 45° C
Precisión	+/-0,1 de 25° C a 45C ^o
Terminal	Conector con cables adaptadores que ofrece el fabricante.

Nota: Tomado de las investigaciones realizadas del Laboratorio Ellab.

3.1.4.2.4 Sensor de Piel EXACON SCIENTIFIC.



Figura 46 Sensor de piel Fuente: Exacon (2010). Recuperado de <http://www.exacon.com/Global/Global11>

Este tipo de sensores son muy utilizados en aplicaciones médicas tanto por el precio como su adaptabilidad, con el inconveniente que la sonda debe permanecer pegado a la piel por lo menos entre 8 a 10 minutos para comenzar a determinar los datos exactos; a continuación se muestra una tabla de sus características:

Tabla 14

Características Sensor de Exacon.

<i>Sensor de piel EXACON SCIENTIFIC</i>	
Fabricante	EXACON SCIENTIFIC
Rango de Acción	25° C a 45° C
Precisión	+/-0,2 de 25° C a 50C°
Medición	de 2,252 K ohmios a 25° C
Referencia	

Nota: Elaborado a partir de datos del laboratorio Exacon.

3.1.4.2.5 Termistor MA100GG103A NTC.



Figura 47. Termistor MA100GG103A NTC

(Digikey, 2010). Es un termistor pasivo que está diseñado para aplicaciones biomédicas que posee una sensibilidad, y precisión fiable a los cambios que experimenta su resistencia por la variación de la temperatura; presentando una curva característica no lineal operando en un rango de 0 °C a 50 °C. Con una resistencia nominal de coeficiente de temperatura negativo NTC de 10K Ohmios.

3.1.4.2.6 Termopila MLX90615

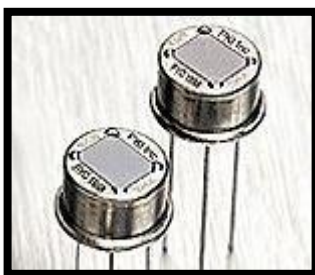


Figura 48. Termopila infrarroja. Fuente: Directindustry (2010). Recuperado de:
<http://www.directindustry.es/prod/perkinelmer-optoelectronics/termopares-infrarrojos>

La termopila es un sensor herméticamente sellado, con un diámetro de 545 micras el cual contiene un filtro óptico que realiza las medidas en un rango espectral, por encima de 5 micras de longitud. Las termopilas poseen una resistencia de referencia adicional para la temperatura la cual va conectada a un sistema de tierra o polo negativo. Las aplicaciones de estas termopilas se centran en desarrollar termómetros clínicos especialmente para el oído; de igual manera se lo utiliza en microondas de aire así como también en controles de temperatura de los automóviles.

3.1.4.3 Amplificador Operacional De Voltaje.

Como se conoce en el ambiente existen cantidades y variaciones analógicas, que pueden ser tomadas o sensadas por dispositivos que tienen la capacidad de detectar sus variaciones; en este caso como los sensores de temperatura los cuales toman una variación de calor y la expresan en voltajes relativamente bajos; cuyos voltajes necesitan ser aumentados por un amplificador de voltaje para que posteriormente sean tratados y visualizados por un sistema electrónico.

El amplificador operacional es un circuito electrónico que se utiliza para aumentar el voltaje o la corriente de un determinado circuito, estableciéndolo en los rangos requeridos.

Para el caso de la implementación del transmisor de temperatura se debe utilizar un amplificador operacional que posea una sola fuente de alimentación debido a que su diseño debe ser lo más compacto posible. A continuación procedemos analizar diferentes amplificadores operacionales que servirán para la implementación del transmisor:

3.1.4.3.1 LM 324

El amplificador LM 324 es un dispositivo que consta con una alta ganancia de compensación con un arreglo de 4 amplificadores internos, los cuales están diseñados para funcionar con una sola fuente en un rango de voltaje de 3 a 32 V. Como muestra la siguiente tabla:

Tabla 15

Cuadro de características LM 324.

Características Técnicas		
Alimentación Simple	2.2V	36V
Alimentación Doble	+/- 0,9 +1.3V	+/- 18 V
Corriente de Alimentación	0 mA	85 μ A
Rango de Temperatura	-40°C	85°C
Entrada de Voltaje Offeset	+/-150 μ V	+/-500 μ V
Entrada de Corriente Offeset	+/-1nA	+/-5Na

Nota: Sacado de las investigaciones de la Industria AMP. Fuente: Datashet LM324.

3.1.4.3.2 INA122

Es un amplificador utilizado en aplicaciones de instrumentación, Con un diseño de dos amplificadores internos los cuales consumen una corriente muy baja centrándose en un rango de alimentación de voltaje de 2 a 36 V. Que tiene la capacidad de funcionar con

fuentes dobles y con un funcionamiento ideal para instrumentos portátiles y adquisición de datos. A continuación se muestra una tabla de sus principales características:

Tabla 16

Cuadro de Características INA122.

Características Técnicas		
Alimentación Simple	2.2V	32V
Alimentación Doble	+/- 1,5 V	+/- 16 V
Corriente de Alimentación	0 mA	50 mA
Rango de Temperatura	0°C	70°C
Entrada de Voltaje Offset	2mV	3mV
Entrada de Corriente Offset	5nA	30nA





Nota: Sacado de las investigaciones de la Industria AMP. Fuente: Datasheet INA122.

3.1.4.4 Microcontroladores.

Los microcontroladores son dispositivos que permiten monitorear, controlar y tratar señales para realizar determinada acción. En el mercado existen industrias que fabrican diferentes líneas de microcontroladores, enfocados en diversas aplicaciones; para la red de monitorización de temperatura corporal analizaremos 3 fabricantes, entre ellos Atmel, Microchip, y Motorola a continuación mostramos características de los microcontroladores que son requeridas para el diseño de monitorización de temperatura. Como conversores análogos Digital, con un número de Bits mayor o igual a 8 bits, puertos de entrada y salida de comunicación serial de procesamiento de datos y almacenamiento de los mismos, bajo consumo de potencias como se presenta a continuación:

Tabla17

Características de Microcontroladores con diferentes fabricantes.

FABRICANTE	MICROCHIP	ATMEL	MOTOROLA
			
MCU	Pic 18F2550	AVR (8 Bits)	68HC908
MIPS	12	10	8
Convertor A/D (bits)	10	8	10
Conectividad	ISP, Enhanced USART	ISP, Enhanced USART	ISP, Enhanced USART
Memoria Flash (KB)	32	64	42
Memoria Ram (KB)	2.048	4	4
Alimentación	2 a 5 V	1.6 a 3.6 V	2 a 5V

Nota:Elaborado por Alexander Trejo.

3.1.4.5 Comunicación Serial.

El puerto serial es un medio de comunicación de datos digitales que opera en computadores y periféricos, en la cual la forma de transmitir es de bit a bit uno, a la vez lo que nos resulta beneficioso para poder hacer la transmisión de los datos de temperatura tomados por el circuito de monitoreo.

A este tipo de comunicación serial se la conoce como RS232 la cual es la norma establecida para transmitir datos binarios entre un DTE⁴¹ y un DCE⁴² en donde la forma de transmisión es mediante un conector tipo DB9 o DB25 con nueve pines de configuración como e muestra a continuación

⁴¹ DTE.- Equipo terminal de datos encargado de transmitir.

⁴² DCE.- Equipo de Comunicación de Datos.

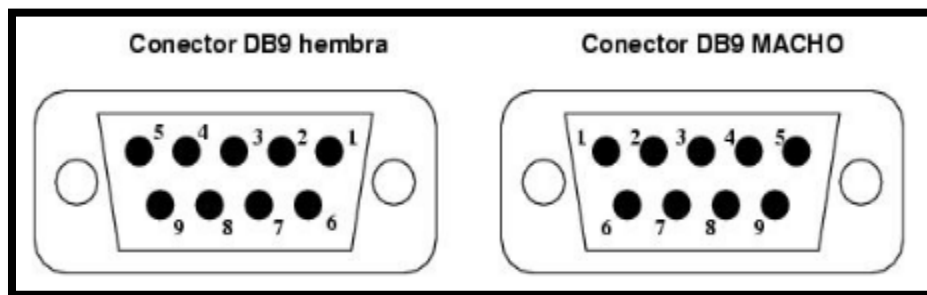


Figura 49. Conectores DB9. Fuente: PCexpertos (2011). Recuperado de: <http://pcexpertos.com/tag/cd/page/2v>.

Tabla 18

Pines del conector de un DB9.

PIN	SEÑAL
1	CD- Detección de portadora
2	RXD- Recepción de Datos.
3	TXD Transmisión de Datos.
4	DTR. Terminal de Datos Lista
5	GND- Nivel de Tierra
6	DSR- Fijación de Datos Lista.
7	RTS- Requerimiento de Envío.
8	CTS- Borrar para envío.
9	RI- indicador de comienzo de transmisión

Nota: Elaborado a partir de las investigaciones PCexpertos.

El sistema de transmisión de datos está constituido por circuitos electrónicos, donde su funcionalidad se basa en tomar la temperatura corporal del cuerpo humano, visualizarla en una pantalla LCD y transmitirla a una base de datos. Por eso se debe realizar, tanto la calibración y respuesta del circuito lo más efectiva posible para tener fiabilidad y seguridad de los datos a transmitir.

3.1.5 ESTRUCTURA DEL TRANSMISOR.

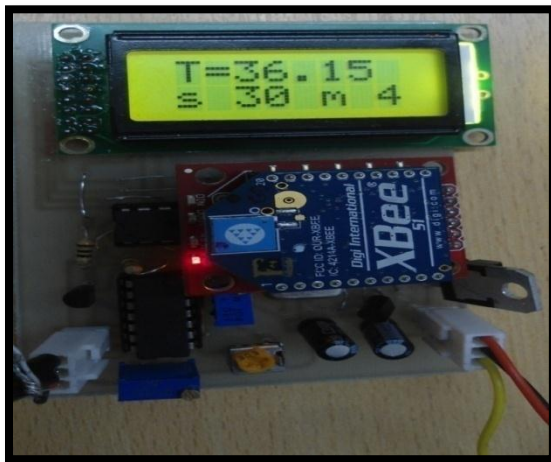


Figura 50. Transmisor de temperatura corporal

El transmisor de Temperatura corporal, consta principalmente de diferentes etapas las cuales, trabajan individualmente pero se suman para dar un producto final:

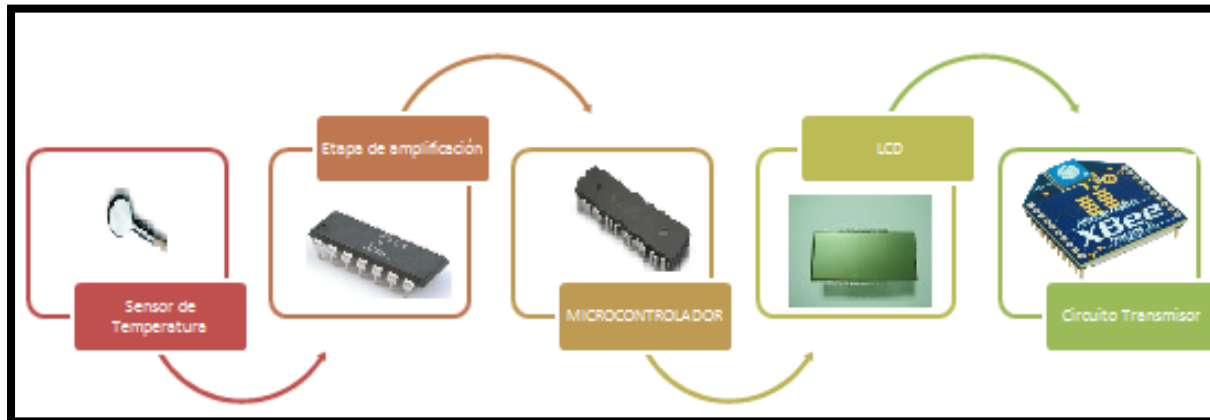


Figura 51. Descripción gráfica del transmisor.

3.1.5.1 Selección del sensor de temperatura.

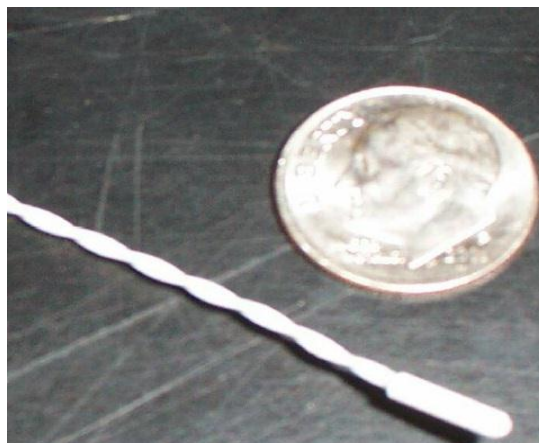


Figura 52. Termistor de temperatura corporal.

Como se detalla anteriormente existen varios dispositivos, los cuales cumplen con la función de medir la temperatura corporal, en este caso para la elección del dispositivo aparte de sus características químicas, exactitud y aplicaciones se realizó en base al precio y la adquisición del sensor en el mercado. Logrando cumplir con todas este tipo de características el Termistor NTC .080"DIA 10K OHM - MA100GG103A.

El termistor MA100GG103A es de coeficiente de temperatura negativo (NTC)⁴³, es decir que cuyo valor decrece con el aumento de la temperatura dando una relación de resistencia vs temperatura no lineal, si no exponencial. Este tipo de sensor trabaja en rangos de temperaturas de 0 °C a 50 °C con una resistencia de 10.000 Ω encapsulado dentro de una punta de plástico diseñados para aplicaciones biomédicas con una respuesta de exactitud de 10 a 15 segundos.

⁴³NTC._ Coeficiente Terminal Negativo para dominación de termistores.

En este tipo de sensor sus características físicas, permiten que se le aplique una venda o una cinta para poder tener adherencia a la piel lo cual mejora la medición de temperatura.

3.1.5.1.1 Circuito amplificador de temperatura corporal.

El termistor MA100GG103A varía su tolerancia o valor de resistencia de acuerdo al rango de temperatura que se encuentre expuesto, por lo tanto se debe encontrar la forma de tener un circuito que permita obtener los valores de variación de esa resistencia y transformar los resultados en parámetros entendibles para el microcontrolador.

Existen muchos tipos de configuraciones de circuitos que permiten, trabajar con termistores, entre ellos tenemos divisores de voltaje, puentes de Wheatstone y más. Es importante enfocar la estructura del transmisor, con el menor número de componentes electrónicos posibles, pero cuidando de que no pierda su funcionalidad, es por eso que mediante diferentes investigaciones se encontró un circuito el cual utiliza un divisor de voltaje con el termistor existente para la fabricación de un sensor de temperatura corporal junto con un amplificador operacional para poder procesar la información mediante un microcontrolador como se muestra a continuación:

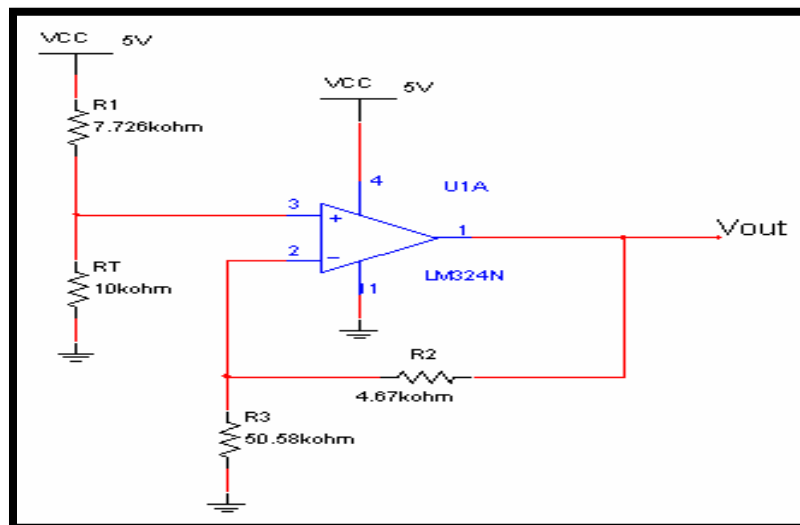


Figura 53. Diagrama del amplificador de temperatura corporal.

Como se puede observar en el gráfico anterior, está compuesto por un divisor de voltaje en el cual está estructurado con un amplificador de tensión acompañado de varias resistencias de precisión. Donde el circuito según su estructura se adapta a las necesidades del proyecto de monitorización de temperatura corporal; por lo cual se analizara tanto matemáticamente como electrónicamente para determinar que este tipo de configuración es adaptable al sistema con sus respectivas resistencias.

3.1.5.1.2 Análisis de Circuito amplificador de temperatura corporal.

Para esto, es necesario utilizar un circuito de instrumentación, que permita obtener los valores requeridos para el sistema de calibración del sensor. Y como parte fundamental de la misma la utilización de un amplificador operacional. Por análisis en secciones anteriores existen diversas gamas de amplificadores operacionales y marcas en el mercado; es por eso que se utilizará el amplificador operacional LM324N que es fácil de adquirir con un costo económico y con las características requeridas para el circuito de monitorización.

(Mimemoria.net, 2011). El amplificador LM324N entre en sus ventajas posee alimentación simple, con un pequeño gasto de energía lo que lo hace apto para alimentarse con baterías.

Es importante, analizar el tipo de configuración que presenta el amplificador operacional LM324, en la calibración del circuito para poder optimizar sus ventajas.

3.1.5.2 Configuración del Amplificador operacional.

En la gráfica de calibración del circuito presenta al amplificador en modo no inversor⁴⁴ como se lo muestra:

⁴⁴No Inversor._ Es una de las configuraciones que puede adaptar un amplificador operacional.

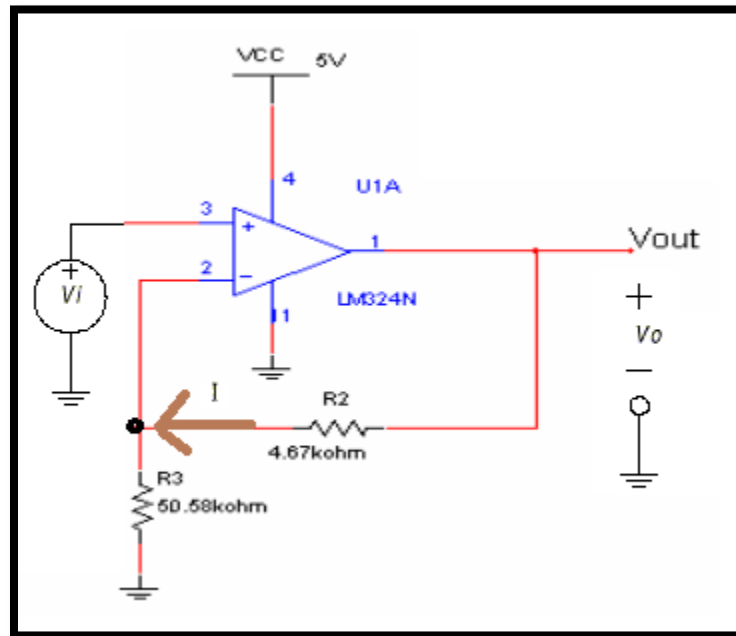


Figura 54. Amplificador en modo no inversor

En el circuito la entrada V_i se aplica el voltaje en la entrada positiva (+), y por arreglo de resistencia R_1 - R_2 como divisor de voltaje se produce una pequeña fluctuación de corriente (I). Puesto que no fluye corriente entre los terminales de entrada, se posee un diferencial de voltaje cero, lo que nos da como resultado que:

$$\checkmark V_i = I \times R_3.$$

Por cuanto:

$$\checkmark V_o = I \times (R_2 + R_3)$$

Estableciendo en términos iguales:

$$\checkmark V_o = \frac{V_i}{R_3} \times (R_2 + R_3)$$

Si todo lo expresamos en términos de ganancia:

$$\checkmark \frac{V_o}{V_i} = \frac{(R_2 + R_3)}{R_3}$$

$$\checkmark \frac{V_o}{V_i} = \frac{(4.67k\Omega + 50.58k\Omega)}{50.58k\Omega}.$$

$$\checkmark \frac{V_o}{V_i} = 1.092$$

$$\text{Ganancia} = 1.092^{45}$$

3.1.5.2.1 Cálculo de ganancias de 3 voltajes para dar mayor fialidad a los datos.

$$V_{nc} = \frac{RT}{RT + R1} \times V_{cc}$$

$$\checkmark V_{ntc} = \frac{6K}{6K+7.72K} \times 5V.$$

$$\checkmark V_{ntc} = 2.10 \text{ v.}(1)$$

$$\checkmark V_{ntc} = \frac{8K}{8K+7.72K} \times 5V.$$

$$\checkmark V_{ntc} = 2.54 \text{ v.}(2)$$

$$\checkmark V_{ntc} = \frac{10K}{10K+7.72K} \times 5V.$$

$$\checkmark V_{ntc} = 2.82 \text{ v.}(3)$$

3.1.5.2.2 Cálculo del voltaje Vo de salida

$$V_o = \frac{V_i}{R3} \times (R2+R3)$$

$$\checkmark V_{ntc} = 2.10 \text{ v.}(1)$$

$$\checkmark V_o = \frac{2.10 \text{ v}}{50.58 \text{ k}\Omega} \times (4.67 \text{ k}\Omega + 50.58 \text{ k}\Omega)$$

$$\checkmark V_o = 2.29 \text{ v}$$

$$\checkmark V_{ntc} = 2.54 \text{ v.}(2)$$

$$\checkmark V_o = \frac{2.54 \text{ v}}{50.58 \text{ k}\Omega} \times (4.67 \text{ k}\Omega + 50.58 \text{ k}\Omega)$$

$$\checkmark V_o = 2.77 \text{ v.}$$

$$\checkmark V_{ntc} = 2.82 \text{ v.}(3)$$

$$\checkmark V_o = \frac{2.82 \text{ v}}{50.58 \text{ k}\Omega} \times (4.67 \text{ k}\Omega + 50.58 \text{ k}\Omega)$$

$$\checkmark V_o = 3.08 \text{ v.}$$

⁴⁵Ganancia.- Término que se da a los voltajes luego de la salida de un amplificador.

3.1.5.3 Gráfica de temperatura Vs resistencia.

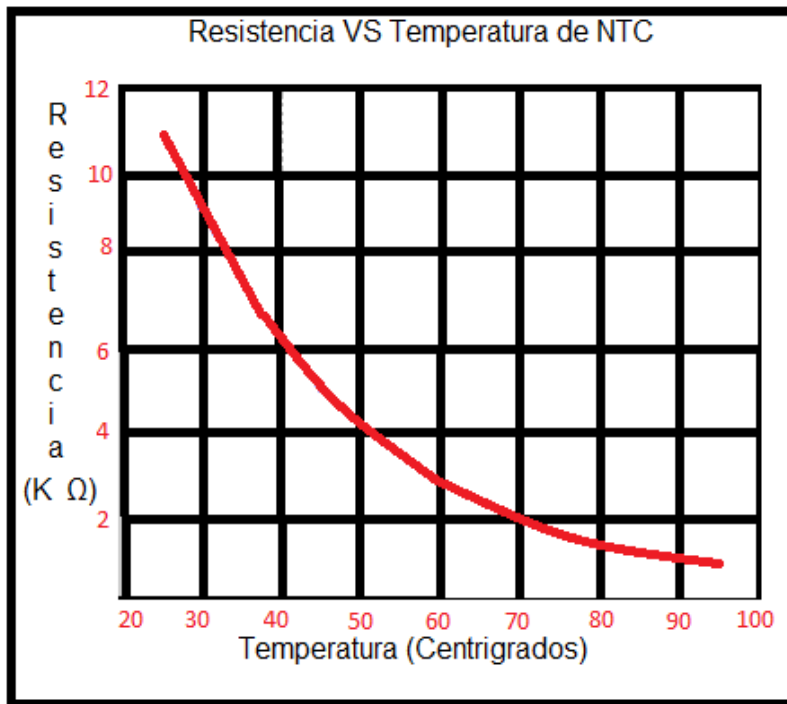


Figura 55. Gráfica del cambio de la temperatura vs la resistencia RTD

3.1.5.3.1 Análisis de la temperatura y elección de las resistencias

$$RT = R_0 \cdot e^{\left\{\beta\left(\frac{1}{T} - \frac{1}{T_0}\right)\right\}} [61]$$

Donde:

RT= Resistencia del termistor NTC al aplicar cierta temperatura ($^{\circ}\text{C}$ ó $^{\circ}\text{K}$).

R₀= Resistencia característica del material del NTC (10 KΩ).

β= Constante de temperatura del material que se calcula en base rangos de temperatura.

T₀= Temperatura de Referencia del material dada por el fabricante a 25 $^{\circ}\text{C}$.

T= Temperatura de referencia para cálculo de β.

En base a la fórmula del termistor NTC, se procede a realizar el cálculo de β el cual se realiza mediante un rango establecido de dos temperaturas dentro de su rango total de tolerancia

Para la aplicación se establecerá un rango de 25 °C a 50 °C; que se extiende dentro de los parámetros de la temperatura corporal del ser humano.

Igualmente se toma un rango de resistencias dentro de la curva característica del termistor por lo tanto:

$$\checkmark \beta = \frac{\ln\left(\frac{R_1}{R_2}\right)}{\left(\frac{1}{T_1} - \frac{1}{T_0}\right)}$$

$$\checkmark \beta = \frac{\ln\left(\frac{10K}{5K}\right)}{\left(\frac{1}{25} - \frac{1}{50}\right)}$$

$$\checkmark \beta = 34.65^\circ\text{C}$$

Con el valor β se puede calcular la R_T , al rango de temperatura requerida para poder linealizar⁴⁶ o captar al cambio de temperatura vs voltaje de la manera más exacta posible.

Ya que los rangos mínimos y máximos de la temperatura corporal se encuentran de 33 °C a 42 °C se determina la resistencia característica del material en el rango establecido, lo permitirá linealizar la variación de la resistencia frente a la variación de la temperatura:

$$R_T = 10 \text{ K}\Omega \cdot e^{\left\{34.65^\circ\text{C}\left(\frac{1}{42^\circ\text{C}} - \frac{1}{33^\circ\text{C}}\right)\right\}}$$

$$\mathbf{R_T = 7985\Omega.}$$

⁴⁶Linealizar.- Escoger dos puntos de una curva para sacar su tramo mas recto.

3.1.5.4 Cálculos de Calibración.

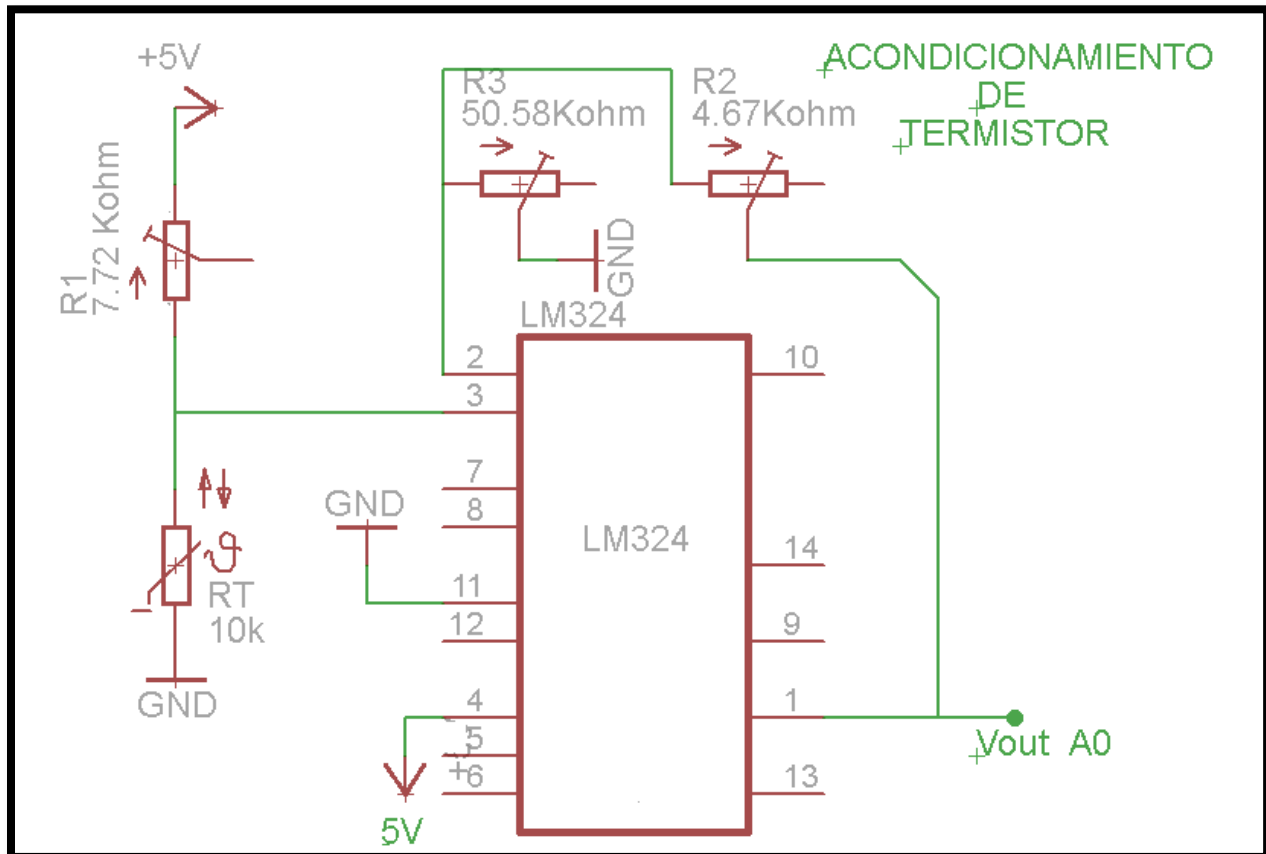


Figura 56. Acondicionamiento del termistor

Para poder determinar los valores de variación que tiene el termistor de acuerdo al cambio de temperatura, se debe utilizar métodos que permitan registrar la temperatura y su equivalencia en este caso debemos relacionar la temperatura con el voltaje emitido. Si se arma la configuración del circuito anterior y la variación de temperatura vs voltaje se introduce los datos en un microcontrolador que realice la conversión analógica digital y esto se visualiza en una LCD, se puede sacar registros de voltaje de acuerdo a la temperatura que se aplique; esto se lo hace con la ayuda de métodos convencionales como introducir el termistor NTC junto con un termómetro de temperatura corporal dentro de una olla de aceite que se encuentre a unos 45°C y alimentando tanto la configuración del amplificador como del microcontrolador en circuito a 5V podemos mirar en la pantalla LCD, como existe la variación de voltaje de acuerdo a como va subiendo o bajando la temperatura y poder sacar un registro de

esta variación, cabe mencionar, que la utilización del aceite se debe a que posee mayor viscosidad⁴⁷ y retiene por mayor tiempo el calor que el agua brindando datos más exactos; a continuación se muestra un gráfico de cómo se realiza este procedimiento y una respectiva tabla de los datos de la temperatura vs voltaje.

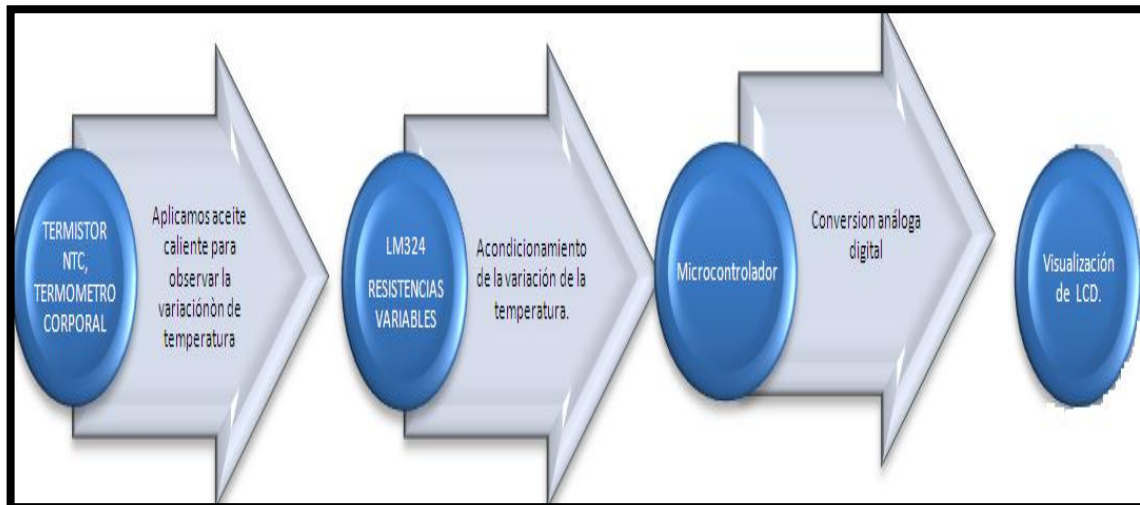


Figura 57. Calibración y amplificación de los datos analógicos de temperatura.

Tabla 19

Equivalencias de voltaje y temperatura.

VOLTAJE VS TEMPERATURA	
TEMPERATURA	VOLTAJE
41	2,14
40,5	2,17
40	2,2
39,5	2,24
39	2,26
38,5	2,3
38	2,33
37,5	2,35
37	2,38
36,5	2,41

⁴⁷Viscosidad.- Líquido pegajoso que retiene los elementos químicos por más tiempo.

36	2,44
35,5	2,47
35	2,5
34,5	2,53
34	2,56
33,5	2,59
33	2,62

Nota: Elaborado por Alexander Trejo.

3.1.5.5 Cálculo de la ecuación de la recta.

Para poder tener los valores de la temperatura mediante un sistema electrónico, se debe trabajar en base a parámetros y ecuaciones; con los valores de conversión obtenidos del voltaje de la tabla 14 junto con la ecuación característica para la temperatura.

Para esto se selecciona dos valores de temperatura y voltajes en pares ordenados para poder tener la ecuación de la pendiente de la recta.

$$y - y_0 = m(x - x_0).^{48}$$

P1(x; y). Primer par ordenado

P2(x0; y0). Segundo par ordenado

P1 (2.66; 33°).

P2 (2.2; 40°).

✓ Calculando la pendiente de la recta:

$$m = \frac{y_2 - y_1}{x_2 - x_1}.^{49}$$

$$m = \frac{40^\circ - 33^\circ}{2.2v - 2.66v}$$

⁴⁸Ecuación de la Recta.- Función matemática para determinar los puntos de unión entre dos puntos distantes.

m= -15.21.

$$y - y_0 = m(x - x_0).$$

$$x = \frac{(y-y_0)}{m} + x_0. \text{ Despejando en términos de } x$$

$$x = \frac{(y-2.2v)}{-15.21} + 40. \text{ Ecuación para la conversión de los datos de voltaje a temperatura corporal.}$$

3.1.5.6 Configuración de los dispositivos XBEE para la transmisión de los datos

De acuerdo a lo tratado en el capítulo anterior, se realizará la transmisión de los datos de forma inalámbrica utilizando dispositivos Zigbee series 2.

Para la configuración de los dispositivos se debe utilizar el software X-CTU⁵⁰ el que permite establecer los parámetros de red para que el dispositivo pueda comenzar la comunicación.

⁵⁰X-CTU.- Software de configuración de módulos xbee.

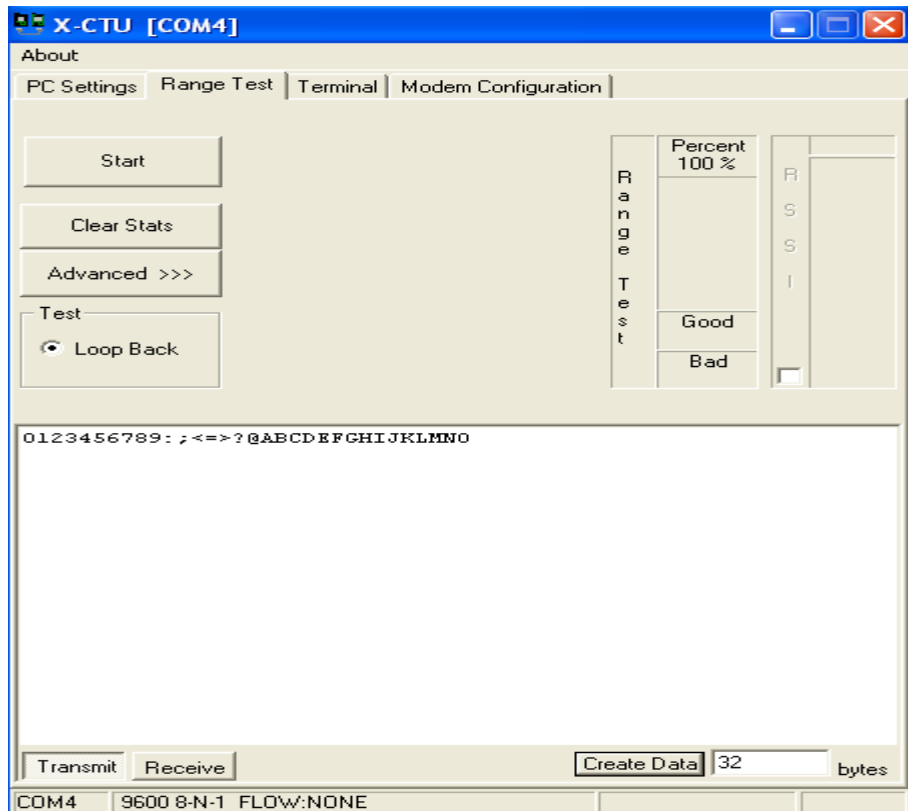


Figura 58. Interfaz X-CTU para configuración de módulos XBEE.

Este tipo de Software se lo descarga directamente de la página web www.digi.com en sus versiones actualizadas, disponible para el sistema operativo con el cual se está trabajando.

Para poder realizar la configuración de los módulos Xbee, se lo debe hacer mediante puerto serial o puerto usb, utilizando el Xbee_explorer que permite interactuar al módulo con el computador.

Tomando en cuenta los parámetros de la zona de terapia intensiva y el número de dispositivos a utilizarse en la red a continuación; se muestra la configuración paso a paso de uno de estos dispositivos para que puedan transmitir los datos dentro de la red Multipunto_ Punto.

Para estos se necesita configurar los 3 módulos Zegbee como transmisores y un cuarto módulo como receptor de la siguiente manera:

3.1.5.6.1 Receptor.

El receptor es el encargado de comunicarse directamente como el computador y recibir todos los datos de temperatura corporal, y se debe configurar con los siguientes parámetros:

- ✓ **Tipo direccionamiento:** Z .net 2.5 Coordinador.
- ✓ **Pan ID⁵¹:** Por Defecto.
- ✓ **Bate Rate** 9.600 Mbps.
- ✓ **Name node:** identifeir central.

3.1.5.6.2 Transmisor:

Para la configuración del transmisor, se selecciona solo parámetros de transmisión ya que la comunicación es de modo simple.

- ✓ **Tipo direccionamiento:** znet 2.5 router/end device AT
- ✓ **Pan Id:** Por Defecto.
- ✓ **Name node:** Tx1/Tx2/Tx3.
- ✓ **Bate Rate:⁵²** : 9.600 Mbps.

Como la comunicación es un solo sentido no se necesita configurar parámetros de comunicación full dúplex, logrando así armar la red de comunicación de temperatura corporal.

La conexión electrónica de un módulo Xbee se la debe realizar de la siguiente manera:

⁵¹PAN ID.- identificador de un dispositivo dentro de una misma red

⁵²Bate Rate.- velocidad de Transmisión

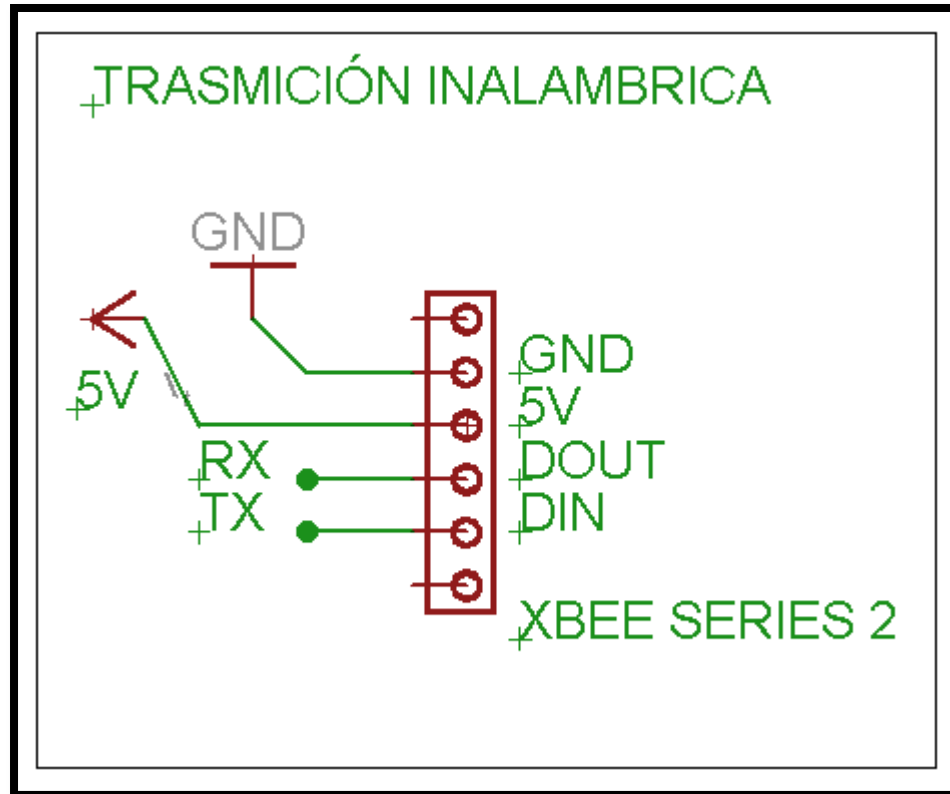


Figura 59. Configuración electrónica del módulo Xbee.

3.1.5.7 Microcontrolador 18F2550

Según análisis en el capítulo 2 existen varios microcontroladores con los cuáles se pueden procesar y trabajar los datos. Dentro de estos se encuentra el 18F2550 por las características, que posee este microcontrolador por su forma compacta y su tamaño es el indicado para procesar los datos enviados por el termistor de temperatura corporal. A continuación se realiza un análisis tanto de su estructura y prestaciones en el circuito de temperatura corporal.

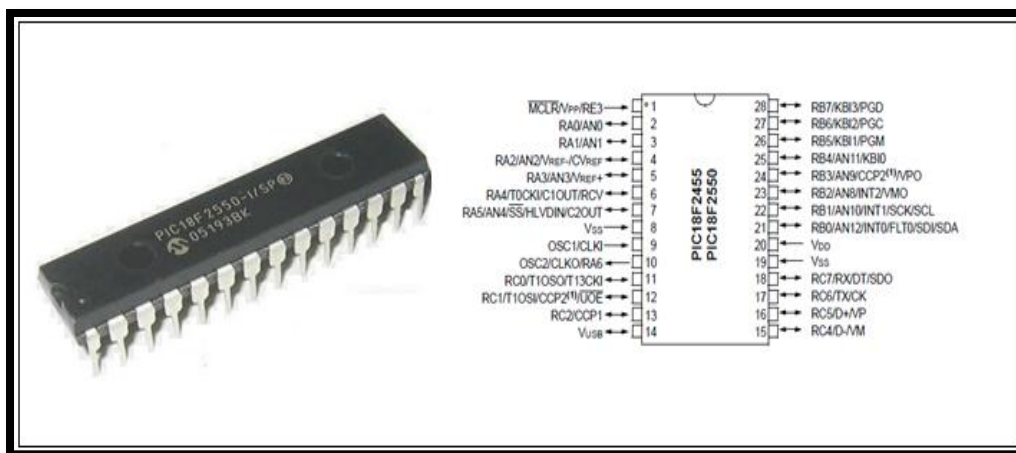


Figura 60. Pines del Microcontrolador 18f2550. *Fuente:* Scribd (2011). Recuperado de: <http://www.scribd.com/doc/52643532/44/Microcontrolador-18F2550>

Tabla 20

Datos del Microcontrolador 18F2550

MICROCONTROLADOR 18F2550.	
Frecuencia de Operación.	Dc 48 MHz.
Programa de memoria Flash (Bytes).	32768
Programa de memoria (Instrucciones).	16384
Memoria de Datos. (Bytes).	2048
Memoria de Datos EEPROM	256
Puertos de E/S.	Ports A, B, C, (E)
Timers	4
Capture/Compare/PWM Modules	2
Comunicación Serial.	MSSP, USART
10-bit Analog-to-Digital Module	10 Input Channels
Comparadores.	4
Voltaje de Operación	2 a 5.5 V
Numero de Pines.	28

Nota: Elaborado a partir de las investigaciones de APM. Recuperado de:

<http://www.scribd.com/doc/52643532/44/Microcontrolador-18F2550>

3.1.5.8 Simulación Del Programa

La comprobación de un sistema de monitorización o de procesos electrónicos se los debe realizar mediante programas de simulación; en el caso de la del transmisor de temperatura corporal se los realizó en el programa Proteus⁵³ para comprobar las acciones y funcionamiento del programa. Como se muestra una imagen en la cual se tiene los componentes del transmisor con el programa ejecutándose, mostrando los resultados en una pantalla LCD y en un terminal virtual, bajo un voltaje de un termistor virtual calibrado bajo parámetros de temperatura corporal.

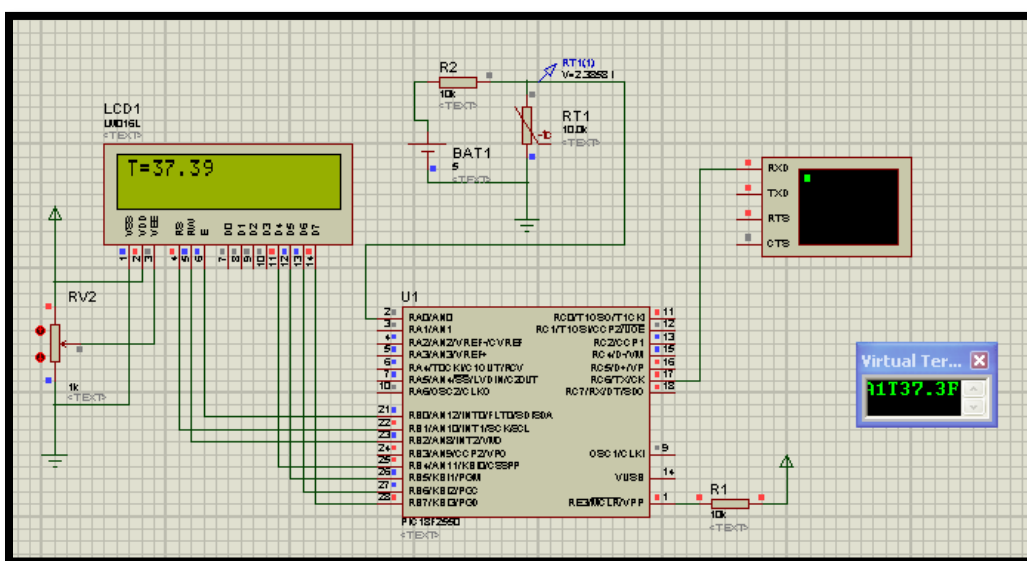


Figura 61. Simulación utilizando Proteus

⁵³Proteus.- programa de simulación de microcontroladores.

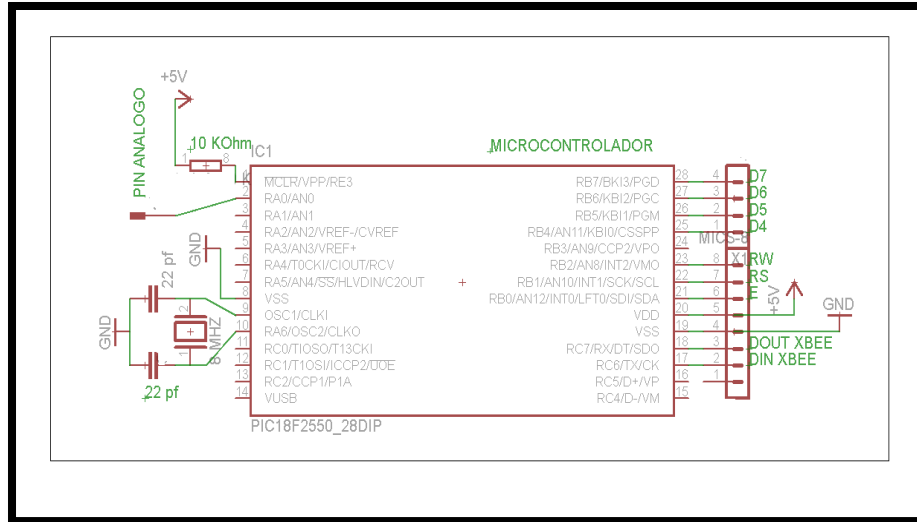


Figura 62. Diagrama electrónico del Microcontrolador 18F2550

3.1.5.9 Visualización de los datos.

Una funcionalidad principal del circuito es la visualización de los datos mediante una pantalla LCD, que se adapte al circuito tanto en forma como tamaño, es por eso que se utilizará la pantalla ICD de 2 x 8 líneas la cuál viene en tamaño reducido, y se le puede programar con el microcontrolador a utilizar:

3.1.5.9.1 Modulo LCD 2 x 8.

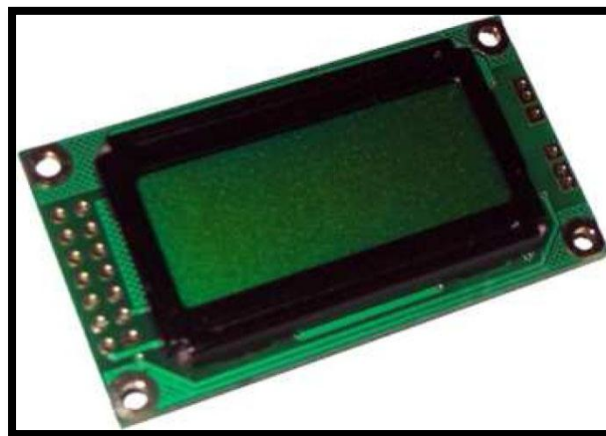


Figura 63. Pantalla LCD 2 x 8. Fuente: Elane (2011). Recuperado de: www.elane.net

La pantalla LCD 2 x 8 tiene 16 pines de configuración, funciona a un voltaje de alimentación de 5 V con ajuste de brillo para visualizar datos, soporta diferentes tipos lenguajes de programación entre ellos CCS, el cual permite su utilización en el circuito de monitorización de temperatura corporal.

3.1.5.9.2 Diagrama técnico de una pantalla Lcd.

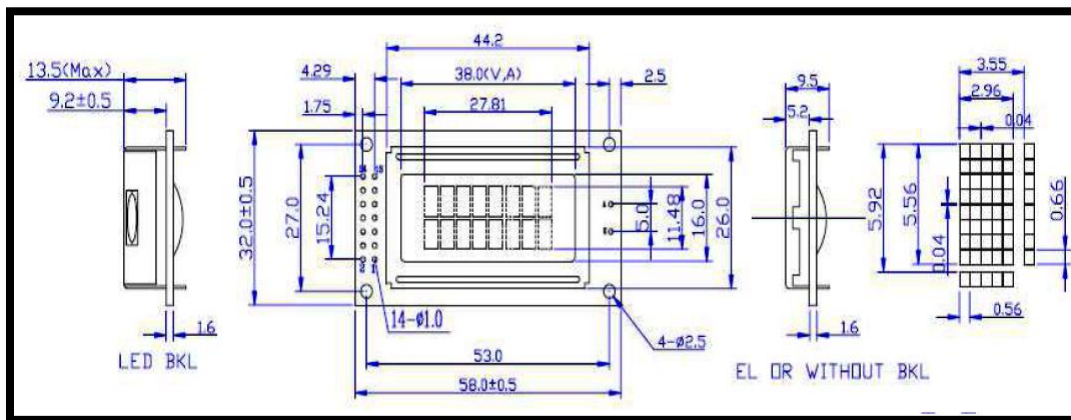


Figura 64. Medidas de LCD 2 x 8 Fuente: Elane (2011). Recuperado de: www.elane.net

3.1.5.9.3 Especificaciones del módulo LCD 2x8.

Tabla 21

Datos del módulo ICD 18F2550.

MODULO LCD 2X8	
CARACTERISTICAS TECNICAS	
Formato Display	8 caracteres (W) x 2 líneas (H)
Dimensiones	: 58 mm (W) x 32 mm (H) x 9.5 mm (T)
Generales	
Rango de Temperatura	de 0 °C to 50 °C (Operación); -20 °C to 70 °C.
PINES DE CONECCIÓN	
1	Pin de masa o tierra.
2	Pin de Voltaje a 5v Vdd
3	Pin de Ajuste de constaste

4	Pin de Selección de Registro.
5	Pin de lectura y escritura.
6	Enable.
7-14	Bus de línea de datos. DB0 – DB7

CARACTERISTICAS ELECTRICAS

Voltaje de Operación | V= 5.0 a Ta=25 °C

(V)

Current Supply 3.0 amp.

Voltage Supply for 2.1 v.

LED

Nota:Elaborado porAlexander Trejo.

3.1.5.9.4 Grafica de la pantalla de la LCD.

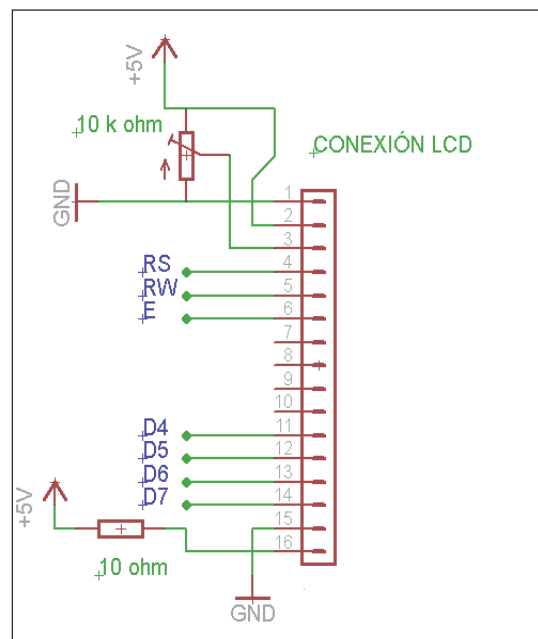


Figura 65. Configuración electrónica de la LCD

3.1.5.9.5 Comandos de trabajo en una LCD.

En los procesos de monitorización se debe tener en cuenta cada una de las funciones, que cumple el equipo es fundamental para la suma de las partes saber que funciones y comandos utiliza la LCD nos ayuda a comprender y tener mejor exactitud en el proceso de datos.

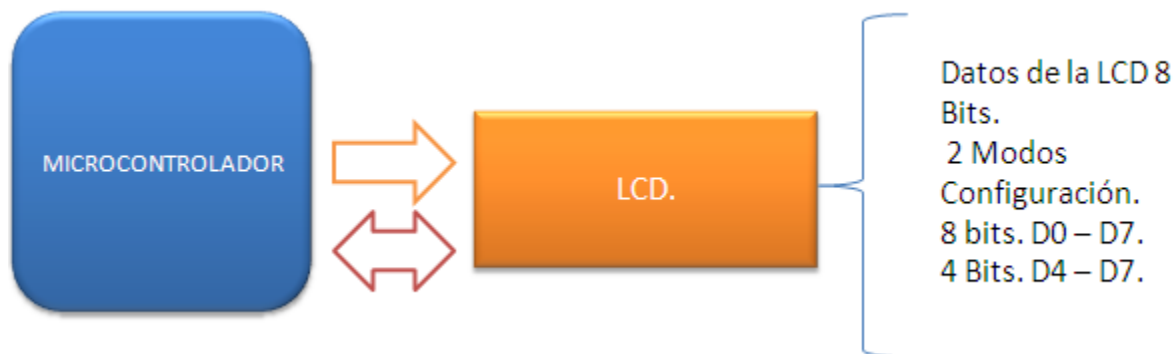


Figura 66. Interacción LCD con microcontrolador

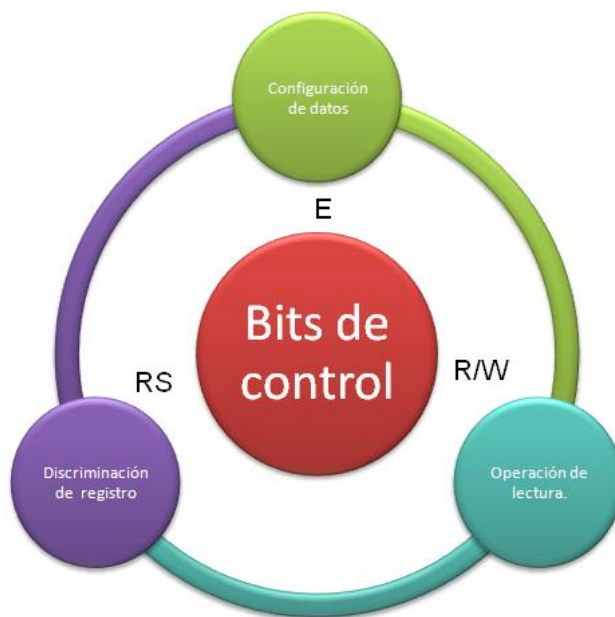


Figura 67. Registros de la LCD.

❖ Comando Borrar Display (Clear Display)

(Elane, 2011). Aplicando una serie de voltajes que digitalmente se aprecian en forma de Bits, lo que realiza este comando borra el contenido de la LCD, colocando el cursor en la primera posición como voltaje en alto 1 y los demás en 0.

Tabla 22.

Códigos de ejecución comandos Clear Display.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Tiempo de ejecución: 1.64 mS.

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

❖ Comando Inicio (Home).

(Elane, 2011). Este comando es de inicialización que coloca el cursor en la posición cero para comienzos de visualización.

Tabla 23.

Códigos de ejecución comandos Home.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	X

Tiempo de ejecución: 1.64 mS

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

❖ Entrar en modo de activación (Entry mode set).

(Elane, 2011) Este comando es para poner en modo de movimiento del cursor, estableciendo lugar de visualización y posición a desplazarse. Utiliza operaciones de lectura o escritura.

Tabla 24.

Códigos de ejecución comandos Entry Mode Set..

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	1/D	S

Tiempo de ejecución: 40 μ S.

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

❖ Control de iluminación (Display on/off control).

(Elane, 2011). Se encarga de encender o pagar el led de iluminación de display y su control de parpadeo.

Tabla 25.

Códigos de ejecución comandos Display on/off control.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Tiempo de ejecución: 40 μ S.

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

❖ Desplazamiento de cursor e iluminación (Cursor or display shift)

(Elane, 2011). Desplaza el cursor sin cambiar el contenido de la memoria de datos moviendo el display a la posición requerida.

Tabla 26.

Códigos de ejecución comandos Cursor or display shift.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	X	X

Tiempo de ejecución: 40 μ s

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

❖ Function set.

(Elane, 2011). Establece la configuración y tamaño del bus de datos a utilizar como de igual forma el número de líneas y en el display.

Tabla 27.

Códigos de ejecución comandos Function set.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	X	X

Tiempo de ejecución: 40 μ s

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

❖ **Comando de escribir y guardar datos (Write data to cg or dd ram).**

(Elane, 2011). Escribe los datos en la memoria DD Ram en código ASCII que se visualizará en la pantalla, de la misma manera trabaja con la memoria CG RAM para visualizar caracteres gráficos según la necesidad del usuario.

Tabla 28.

Códigos de ejecución comandos Write data.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0		1Caracteres gráficos / Código ASCII							

Tiempo de ejecución: 40 μ s.

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

❖ **Comando de lectura de datos (Read data from cg or dd ram)**

(Elane, 2011). Permite leer los datos de la memoria Ram que están se encuentran tanto almacenados en código ASCII como en caracteres gráficos.

Tabla 29.

Códigos de ejecución comandos Write data.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	Caracteres gráficos / Código ASCII							

Tiempo de ejecución: 40 μ s.

Nota: Elaborado a partir de las Investigaciones del centro web Elane. Recuperado de: www.elane.net

3.1.5.10 Etapa de regulación de voltaje y máxima corriente consumida por el circuito.

Para el caso de del transmisor de temperatura corporal, por la utilización de dos baterías de voltaje continuo en forma paralela se tiene un voltaje de casi 8v; por lo cual se debe utilizar un regulador de voltaje que permita dar protección al controlador. Como brindarle el voltaje ideal para su funcionamiento:

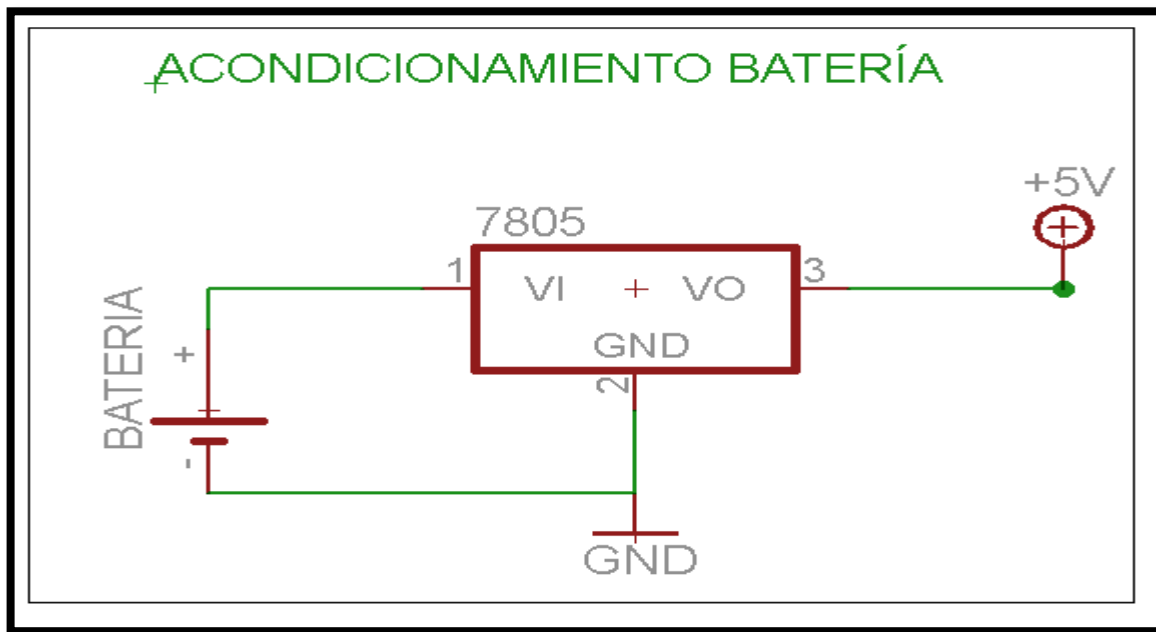


Figura 68. Diagrama electrónico del acondicionamiento de la batería

3.1.5.10.1 Selección y gráficas de baterías seleccionadas.



Figura 69. Baterías genéricas de 3.5v

Existen un sin número de baterías de voltaje continuo en el mercado, pero para el circuito de monitorización de temperatura corporal, se necesita que las baterías cumplan ciertos requisitos entre los cuales se encuentran los siguientes:

- ✓ Un alto contenido de amperaje continuo.
- ✓ Un tamaño compacto y adaptable al transmisor de temperatura corporal.
- ✓ Que sean autorecargables.
- ✓ Que su voltaje sobrepase los 5,2 V para poder alimentar a todo el circuito de temperatura corporal.

Estas son las principales características que debe poseer las baterías del transmisor de temperatura corporal.

3.1.5.10.2 Baterías para teléfono móvil Nokia serie 5300 n80.

Estas baterías cumplen con las características principales que se requiere en el circuito de temperatura corporal, tienen un amperaje continuo de 900 mAh que le permite un largo funcionamiento, en un tamaño compacto para la aplicación.

Tabla 30

Componentes de batería de celular

CARACTERISTICAS TECNICAS	
Amperaje Continuo	1100 mAh
Voltaje Nominal	3.7V
Temperatura de Operación.	15 °C a 50 °C
Compuesto Químico	Litio ⁵⁴ .
Tiempo de carga	1 a 2 horas.
Tiempo de Vida	1 Año

Nota: Elaborado a partir de los productos de Nokia. Recuperado de: http://es.made-in-china.com/co_cnjuzhao/product_Cell-Phone-Battery-for-Nokia-8800-Arte-E66_huururegn.html

⁵⁴ Litio.- compuesto químico que se utiliza en las baterías.

3.1.6 Sistema de Recepción de Datos.

El sistema de recepción de datos, es el encargado de procesar la información enviada por el transmisor, almacenarla y asignarla a los diferentes parámetros relacionados con los datos a procesar por el personal médico del Hospital "San Vicente De Paúl." dando como respuesta una mayor eficiencia tanto en procesos de monitorización como datos de un paciente.



Figura 70. Procesos de recepción de datos.

3.1.6.1 Circuito receptor.

El circuito receptor del sistema de temperatura corporal tiene como objetivo adquirir los datos de los transmisores de temperatura, y pasarlos al computador para que este pueda procesarlos y almacenarlos.

El circuito receptor utiliza un modulo Xbee el cual se conecta con un Xbee_explorer, al computador con una interfaz USB pasando así periódicamente los datos de temperatura corporal según se conecten los transmisores.

El trabajo del receptor de temperatura corporal es únicamente pasar los datos al ordenador por un puerto con disponible en el computador. Las tareas de procesamiento y alerta de cambios de temperatura en los pacientes lo realiza programa y la base de datos del computador.

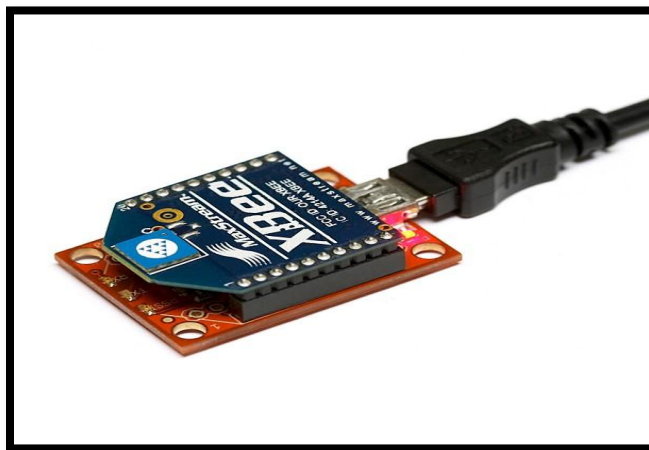


Figura 71. Receptor de temperatura corporal. Fuente: Xbee (2011). Recuperado de: www.xbee.com

3.1.6.2 Estructura de la aplicación

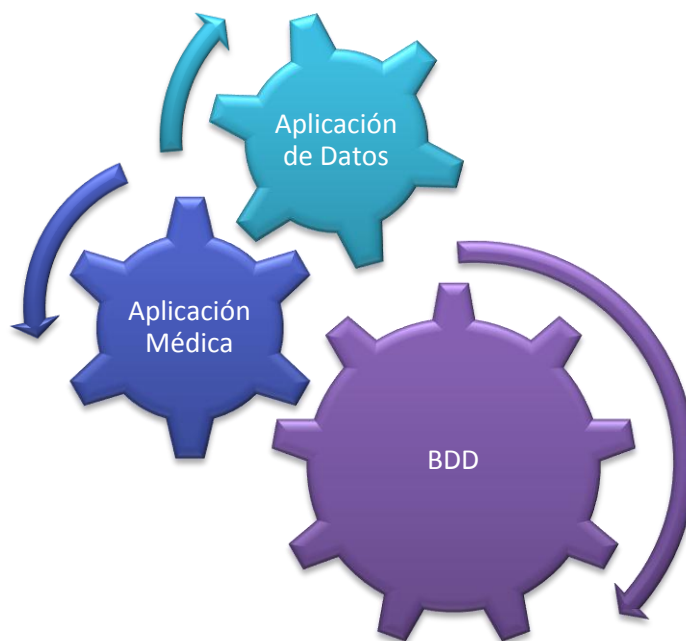


Figura 72. Estructura del software de aplicación

3.1.6.2.1 Aplicación de Recepción de Datos:

Esta aplicación trabaja en la consola para poder recibir la información emitida desde el componente electrónico. Es importante señalar que se interactúa en diferentes aplicaciones para que no exista problema cuando aparezca algún inconveniente en la interfaz de la Aplicación.

Recibir Datos.- Para recibir los datos utiliza la librería GiovynetDriver; esta librería facilita la recepción de datos entre el puerto serial y la computadora.

3.1.6.3 Proceso para Ingresar temperaturas a la BDD.-



Figura 73. Proceso de integración con la base de datos.

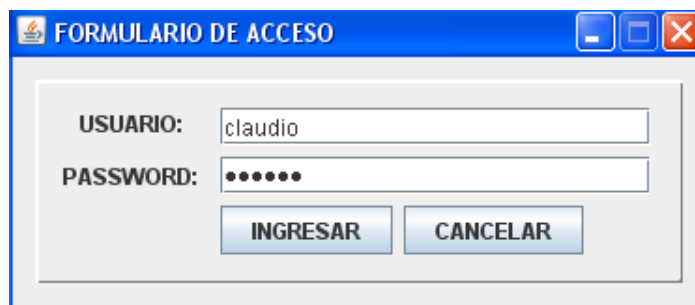
1. Una vez emitida la señal de forma inalámbrica se recepta en el Computador mediante el puerto serial adaptado a la PC.
2. En la Aplicación se decodifica el dato emitido y se le da formato para ingresar a la Base de Datos. El dato emitido muestra el número de la cama.
3. Se ingresa a la tabla tab_analisis, los campos con el formato correspondiente.

3.1.6.4 Aplicación Médica:

La aplicación médica a utilizar en el hospital, muestra la interfaz para ver el estado del Paciente.

- ✓ La Aplicación consta de cuatro partes importantes:
- ✓ Control de Acceso de Usuarios
- ✓ Manejo del Paciente.
- ✓ Monitoreo.
- ✓ Estadísticas

✚ Control de acceso de usuarios:



FORMULARIO DE ACCESO

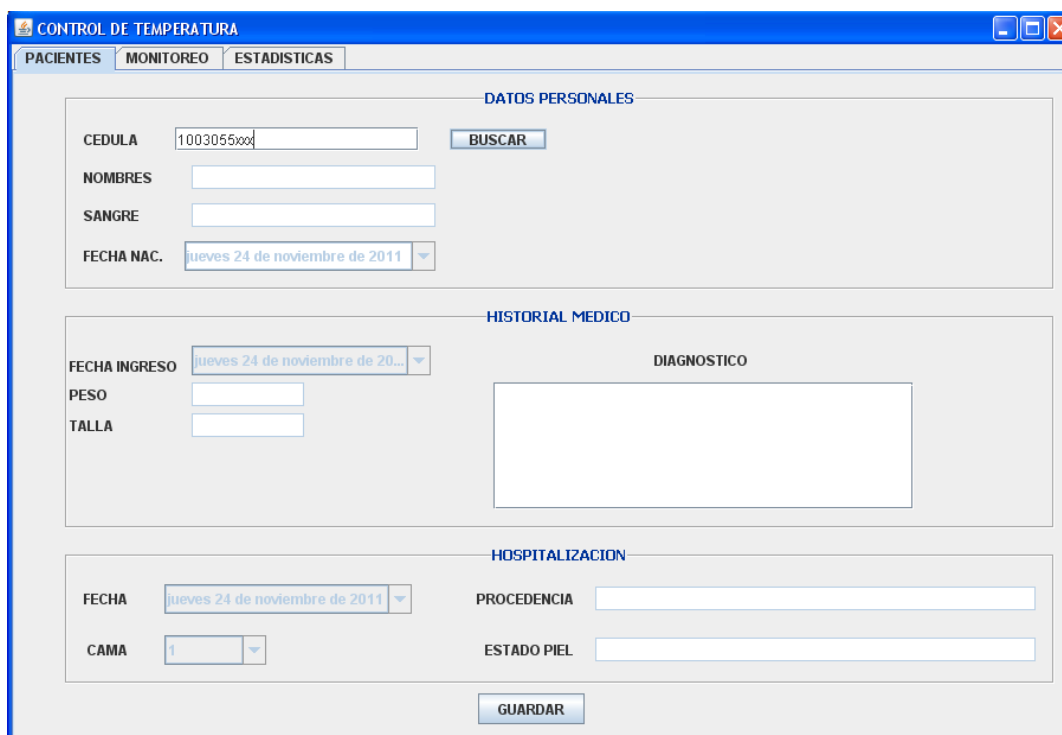
USUARIO: claudio

PASSWORD: ●●●●●●

INGRESAR CANCELAR

Figura 74. Proceso de ingreso a la base de datos.

✚ Manejo del Paciente



CONTROL DE TEMPERATURA

PACIENTES MONITOREO ESTADISTICAS

DATOS PERSONALES

CEDULA 1003055000 BUSCAR

NOMBRES

SANGRE

FECHA NAC. jueves 24 de noviembre de 2011

HISTORIAL MEDICO

FECHA INGRESO jueves 24 de noviembre de 20... DIAGNOSTICO

PESO

TALLA

HOSPITALIZACION

FECHA jueves 24 de noviembre de 2011 PROCEDENCIA

CAMA 1 ESTADO PIEL

GUARDAR

Figura 75. Ficha médica del paciente.

Al ingresar el número de Cédula del Paciente por primera vez este se registra en la base de datos, en caso de que ya exista se procede a llenar o actualizar su historial médico.

✚ Monitoreo.-

CAMA 2

TEMPERATURA ACTUAL

0.0

Datos Personales:

NOMBRES Rafael Andrade

F. NAC 1976-06-02 00:00:00.0

0.0

34.1

34.6

35.0

34.8

34.1

34.0

33.9

33.8

34.0

34.4

34.3

34.2

CAMA 2

Figura 76. Datos de monitoreo de temperatura corporal.

Esta ventana aparece en la parte superior de los datos personales del paciente, la misma que permite enviar una alerta desde aplicación cuando existe un rango fuera de límite en este caso la temperatura.

✚ Estadística

En esta parte de la Aplicación se muestra un gráfico estadístico de la temperatura – tiempo. Es decir se graficarán los datos ingresados a la Base de Datos.

La línea roja representa los cambios de temperatura cada cierto tiempo.

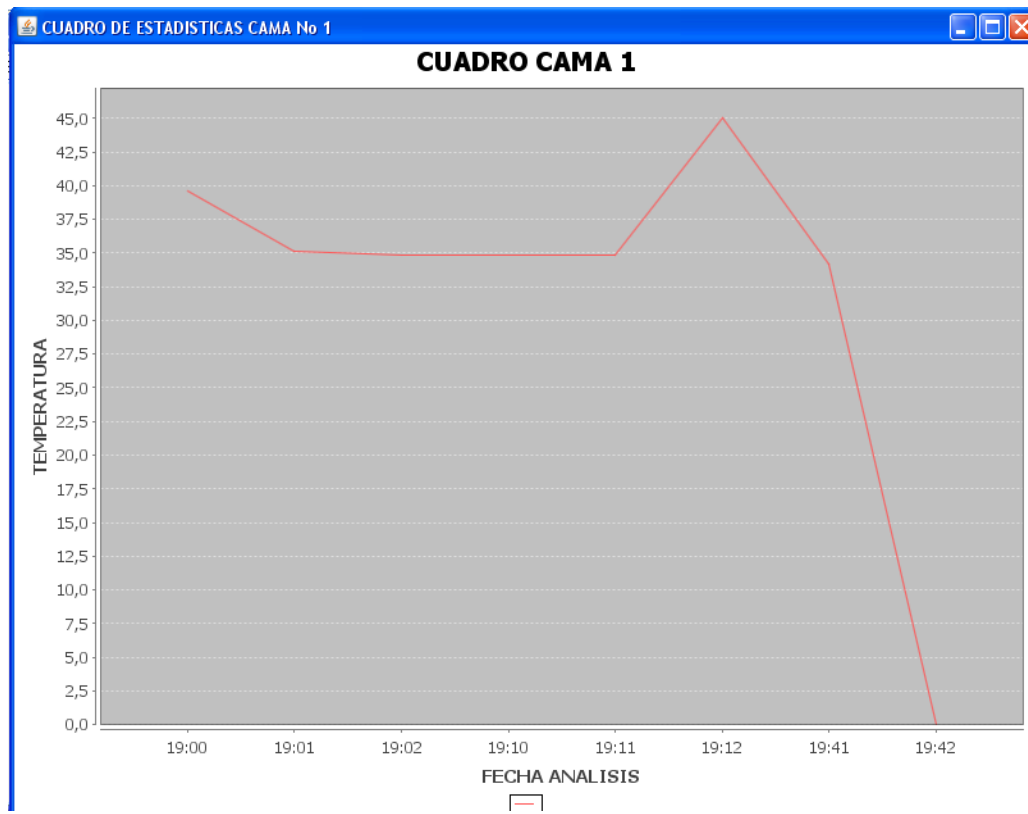


Figura 77. Gráfico estadístico de la temperatura corporal

CAPITULO IV:

4. PRUEBAS EXPERIMENTALES

En este capítulo se analizará el funcionamiento del sistema de medición de temperatura corporal, comparando el transmisor de temperatura con termómetros existentes en el mercado, además se examinara el costo del sistema de temperatura corporal en el mercado, como los resultados arrojados por el sistema en la base de datos, para la implementación en el "Hospital San Vicente De Paúl".

4.1. SISTEMA DE MEDICION DE TEMPERATURA CORPORAL

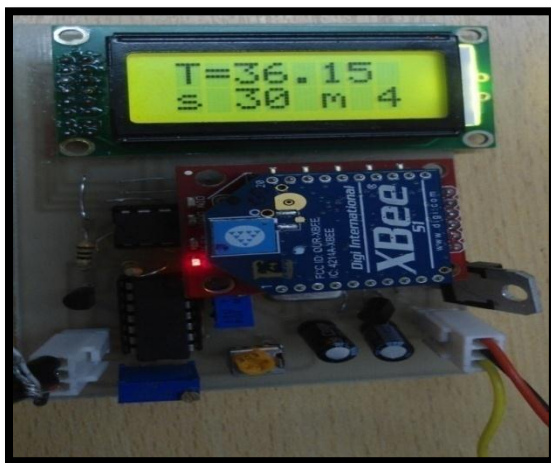


Figura 78. Sistema de temperatura corporal.

El propósito fundamental del sistema de medición de temperatura corporal es transmitir de forma inalámbrica hacia receptor conectado a un ordenador, que posee una base de datos logrando así causar el menor malestar al paciente y alertando a las enfermeras en caso de existir variación de temperatura llevando un registro de las mismas.

4.1.1 Características de transmisor de temperatura corporal

Tabla 31

Especificaciones transmisor de temperatura corporal

<i>Especificaciones.</i>	
Alcance de transmisión:	50m ² .
Rango de medición:	33 C° - 42 C°.
Precisión:	0,1 C°
Sistema de Visualización:	LCD 2x8
Resolución:	0, 01 C°
Led indicador:	Rojo.
Voltaje de Entrada min.	5.5 V.
Voltaje de Entrada max.	8 V.
Zonda de medición:	Termistor NTC
Tecnología de transmisión	Zegbee.
<i>Especificaciones de ambiente.</i>	
Temperatura de trabajo:	-10 °C a +50 °C
Protección contra humedad:	Leve
Materiales tóxicos:	No
<i>Especificaciones mecánicas generales.</i>	
Materiales:	Plástico, Tela
Peso:	200 g.
Tamaño:	50 mm x 40 mm x 30 mm.
Duración de Baterías.	6 horas Recargables.
Garantía:	Vida Útil.

Nota: Elaborado por Alexander Trejo

4.1.2 Características del Receptor de temperatura de temperatura corporal.

El receptor está compuesto por tres partes, la placa de alimentación del Xbee, el cable de interfaz mini usb y el dispositivo receptor Xbee; a continuación se muestra sus principales características:

Tabla 32.

Especificaciones receptor temperatura corporal.

Especificaciones	
Placa de alimentación xbee xplorer	Gamas de Xbee series 1 - Series 2.5
Interfaz de comunicación	Mini Usb.
Corriente de Operación	500 mA.
Tecnología de recepción	Zegbee.

Nota: Elaborado a partir de los datos de Electrónica Embajadores. Recuperado de: <http://www.electronicaembajadores.com/Productos/xbee-explorer-usb>.

4.1.3 Características de la Base de Datos

Tabla 33

Especificaciones base de datos.

Especificaciones.	
Monitoreo de temperatura.	Visualiza en el ordenador la temperatura corporal de los pacientes.
Estadísticas	Visualiza mediante cuadros los cambios efectuados en la temperatura corporal
Formularios de acceso	Restringe el acceso a personal no autorizado.
Datos del paciente	Permite llenar un formulario con los datos clínicos del paciente como nombre, cedula de identidad, fecha de nacimiento, diagnostico entre otros.

Plataforma de Programación	Netbeans 6.9.1
Lenguaje de Programación	JAVA
Librería Puerto Serial	GiovynetDriver.zip
Librería de Conexión a la BDD	MySQL-conector-java-5.0.8

Nota: Elaborado por Alexander Trejo.

4.1.4 Lógica del funcionamiento del Circuito.

- ✓ El circuito de transmisión de temperatura trabaja de la siguiente forma:
- ✓ El rango de medición de temperatura es de es de 33 °C a 40 °C.
- ✓ El rango de alerta de variación de temperatura es de es de 33.5 °C a 37.5 °C.
- ✓ La alerta de alarmas de variación de temperatura se realiza cada 10 s, enviando el dato de temperatura al computador.
- ✓ La alerta de cambio de baterías se realiza cada 5 s, cuando necesitan ser cargadas. Enviando el texto “batería baja”.
- ✓ En caso de no haber alteración de la temperatura se envía la temperatura cada 20 minutos para registro de la base de datos.
- ✓ Los transmisores Xbee entran solo en funcionamiento, cuando se requiere la transmisión de datos, caso contrario permanecen apagados.
- ✓ La pantalla LCD 2 x 8 enciende su led de iluminación y titila solo cuando existe transmisión.

4.1.5 Consumo de Corriente.

El análisis del consumo de corriente es importante para determinar el buen desempeño del dispositivo y horas de funcionamiento como se lo determina en la siguiente tabla.

Tabla 34

Consumo de corriente transmisor de temperatura corporal.

DISPOSITIVOS	CONSUMO VOLTAJE.	CONSUMO CORRIENTE
Microcontrolador	5.0 V	25 mA.

Lcd	5.0 V	35 mA.
Xbee	3.3 V	40 mA.
Amplificador	5 V	0,7 mA.
Led.	2.5 V	10 mA.
Lm24.	5 V	0,7 mA.
Consumo total	5 V.	112 mA.

Nota: Elaborado por Alexander Trejo.

Como se puede ver el circuito tiene una corriente de consumo de 112 mA manteniendo un porcentaje más del 10 % de los materiales indirectos como resistencias, condensadores y otros, se obtiene un valor aproximado de 120 mA si se opera el sistema a un 100 % el circuito permanecerá prendido teóricamente según la siguiente fórmula:

$$\checkmark \text{ Número de horas operación de baterías} = \frac{\text{Corriente de baterías (mAh)}}{\text{Consumo de corriente del circuito}}$$

$$\checkmark \text{ Número de horas operación de baterías} = \frac{900 \text{ mAh}}{112 \text{ mA}}$$

$$\checkmark \text{ Número de horas operación de baterías} = 7 \text{ horas.}$$

4.1.6 Duración de Baterías.

La duración de las baterías depende directamente del consumo de corriente, como se puede observar teóricamente se tiene una duración de la operación del circuito aproximadamente de 7 horas reloj pero en la práctica este periodo de tiempo se reduce a 6 horas por lo cual el circuito se adapta para que opere bajo ciertos parámetros como por ejemplo:

- ✓ Control electrónico de encendido de la LCD 2x 8.
- ✓ Control electrónico de encendido del dispositivo Xbee.

El led de iluminación de la LCD, se encenderá únicamente cuando varíe la temperatura corporal del paciente en el rango establecido.

Los transmisores xbee se encenderán únicamente para realizar la transmisión de datos de la temperatura corporal.

Logrando así obtener un mayor periodo de operación del circuito a continuación mostramos algunas tablas con los diferentes consumos de energía:

Tabla 35

Operación de baterías al 100 %.

Uso de Baterías sin regulación.		
LCD	MICROCONTROLADOR	XBEE
Siempre encendido	Siempre encendido	Transmitiendo en todo momento

Nota: Elaborado por Alexander Trejo.

Tabla 36.

Operación de baterías con modificación al sistema.

Uso de Baterías con regulación.		
LCD	MICROCONTROLADOR	XBEE
Visualiza cada vez que hay variación.	Siempre encendido	Transmite cada vez que hay variación.

Nota: Elaborado por Alexander Trejo

4.2. TERMOMETRO DE MERCURIO.



Figura 79. Termómetro de mercurio. *Fuente:* Wikipedia (2011). Recuperado de: http://es.wikipedia.org/termometros_de_mercurio.

(Termometro, 2011)El termómetro de mercurio generalmente es un aparato que sirve para medir la temperatura de determinados materiales, para la aplicación el termómetro de mercurio mide la temperatura corporal del paciente. El termómetro de mercurio fue creado por Daniel Gabriel Fahrenheit alrededor del año 1714 fabricándolo dentro de un bulbo de vidrio transparente al cual ha exponerlo a determinadas temperaturas del cuerpo humano produce que el elemento químico del mercurio se dilate reflejando su valor en una escala predeterminada de 33 °C a 42 °C .

4.2.1 Características de un termómetro de Mercurio

Tabla 37

Especificaciones del termómetro de mercurio.

Termómetro de Mercurio	
Elemento Químico	Mercurio
Elemento físico	Bulbo de vidrio y metal
Principio de operación	Variación de Volumen
Escala	33 °C – 42 °C
Velocidad de Respuesta.	Lenta

Nota: Elaborado por Alexander Trejo.

4.2.1.1 Comparación de medidas.

Tabla 38

Comparación de muestras con termómetro de mercurio.

Numero de Muestras.	Transmisor de temperatura corporal	Termómetro de mercurio.
1	36,52	36,2
2	36,31	36,5
3	37,14	36,6
4	36,63	36,4
5	36,21	36,3
6	36,62	36,3

Nota: Elaborado por Alexander Trejo.

4.3. TERMÓMETRO DIGITAL FLEXIBLE



Figura 80. Termómetro Digital. Fuente: El Termometro (2011). Recuperado de:
<http://www.termometro.es/termometro-flexible.php>

El termómetro digital cumple con las mismas funciones que un termómetro de mercurio, a diferencia que los termómetros digitales tienen mayor exactitud, alta fiabilidad y precisión de la temperatura corporal y su lectura es más fácil; mostrando el resultado en una pantalla LCD este tipo de termómetros son fáciles de usar y transportar, utilizan pilas de voltaje continuo aproximadamente 1.5 V con una vida útil de 100 horas en uso continuo. Este tipo de termómetros toma el valor real de la temperatura en aproximadamente 45 segundos notificando el resultado a mostrar con sonidos agudos.

4.3.1 Características de un termómetro Digital

Tabla 39

Especificaciones de un termómetro digital.

Termómetro Digital	
Elemento Químico	Cristal Líquido (LCD)
Elemento físico	Plástico, Metal
Principio de operación	Sensibilidad con sensores
Escala de medición	35,5 °C – 42,0 °C
Tiempo de medición.	Mediana.
Alimentación	Batería de 1.5 V.

Nota: Elaborado por Alexander Trejo.

4.3.2 Comparación de medidas.

Tabla 40

Comparación de muestras con termómetro de Digital.

Numero de Muestras.	Transmisor de temperatura corporal	de Termómetro Digital.
1	36,62	36,6
2	36.53	36,4
3	37,12	37
4	36,22	36.4
5	36,81	36,8
6	37,11	37,1
7	36,54	36,3

Nota: Elaborado por Alexander Trejo.

4.4. MULTÍMETRO FLUKE 179.



Figura 81. Grafico estadístico de la temperatura corporal. Fuente: Fluke (2011). Recuperado de: <http://www.fluke.com/fluke>.

El multímetro de marca Fluke serie 179 es uno de los dispositivos de medición más utilizados por los técnicos eléctricos y electrónicos para diferentes actividades tecnológicas. Por su versatilidad, fácil uso y con un gran número de funciones entre ellas un módulo de medición de temperatura de amplio rango de medición, permitiendo así comparar el trasmisor de temperatura corporal con un aparato de precisión técnica y poder medir diferentes resultados.

Tabla 41.

Especificaciones de un termómetro fluke

Termómetro Fluke	
Materiales	Plástico, Metal
Principio de operación	Sensibilidad con sensores
Escala de medición	-40 °C/400 °C
Tiempo de medición.	Rapida.
Resolución de temperatura.	0,1 °C
Precisión	1,0 %

Nota: Elaborado por Alexander Trejo.

4.4.1 Comparación de medidas.

Tabla 42.

Comparación de medidas con un termómetro fluke.

Numero de Muestras.	Transmisor temperatura corporal	Termómetro Fluke 179
1	36,63	37,2
2	36.63	38,1
3	37,22	36,5
4	36,32	35,3
5	36,41	36,4
6	37,21	38,3

Nota: Elaborado por Alexander Trejo.

4.5. TABLA DE LAS DIFERENTES MUESTRAS TOMADAS.

Uno de los objetivos del sistema de medición de temperatura corporal, es que su medida sea exacta, igual o aproximada a los diferentes dispositivos de medición; es por eso que en base a las muestras de temperatura realizadas se determina el rango de error de los diferentes dispositivos de temperatura corporal como se muestra en la siguiente tabla.

Tabla 43.

Comparación de muestras de medidores de temperatura corporal.

MUESTRA	DISPOSITIVO (°C)	DIGITAL (°C)	ERROR (°C)	MANUAL (°C)	ERROR (°C)
1	36,62	36,6	0,02	36,2	0,42
2	36,53	36,4	0,13	36,5	0,03
3	37,12	37	0,12	36,6	0,52
4	36,22	36,4	0,18	36,4	0,18
5	36,81	36,8	0,01	36,3	0,51
6	37,11	37,1	0,01	36,3	0,81
7	36,54	36,3	0,24	36,5	0,04
Promedio			0,1		0,3

Nota: Elaborado por Alexander Trejo.

4.6. ALCANCE DE LOS DISPOSITIVOS INALÁMBRICOS.

El alcance de transmisión de los dispositivos Xbee se adapta en base al área del cuarto de la zona de terapia intensiva del Hospital San Vicente De Paul el cual es aproximadamente es de 40 m². Por lo cual el dispositivo seleccionado cumple con las características de cobertura.

Tabla 44

Alcances dispositivos Inalámbricos

Alcance Xbee	XBee Series 1	XBee Series 2	XBee Pro
Lugares Cerrados	30 m	40 m	50 m
Lugares Abiertos	100 m	120 m	150

Nota: Elaborado por Alexander Trejo.

Según los datos mostrados por la tabla se puede observar que las 43 muestras tomadas por diferentes dispositivos de temperaturas, tiene un rango de error mínimo; por lo cual garantiza que el sistema de medición de temperatura realiza la medición de datos exactos.

4.7. CUADRO DE COSTO DEL SISTEMA DE MONITORIZACIÓN DE TEMPERATURA CORPORAL.

Unos de los factores determinantes al momento de adquirir o implementar un sistema de monitorización son sus beneficios, costos de implementación y garantía de los equipos.

Actualmente en el mercado existen varios sistemas de monitorización de temperatura corporal, los cuales tienen precios muy altos; ya que cumplen con diversas normas y estándares establecidos por diferentes ministerios de salud, es por eso que al momento del desarrollo de un sistema de monitorización se debe determinar su alcance y acción a ejecutar, en el caso del sistema de monitorización de temperatura corporal no necesita entrar a un sistema de evaluación o de estandarización ya que su diseño y alcance no compromete ni expone algún daño a la salud del paciente, por este motivo sus costos tanto de fabricación como de implementación se encuentran al alcance de diferentes jerarquías médicas como se detalla en tabla 45.

Tabla 45.

Cotización de costos del sistema de temperatura corporal

Cotización Del Sistema De Temperatura Corporal			
Descripción	Cantidad	Precio	Total
Pic 18f2550	3	10 \$	30 \$
Pantalla Lcd 2 x8	3	5,36 \$	16,07 \$
Termistor NTC .080"DIA 10K OHM - MA100GG103A	3	30 \$	90 \$
X bee explorer USB	1	39,73\$	39,73
Resistencias variables de precisión 100 k	3	0.85\$	2,55\$
Resistencias variables de precisión 10 k	3	0.85\$	2,55\$
Resistencias variables de precisión 5 k	3	0.40\$	1,20\$
Condensadores cerámicos de 22 pf.	6	0.09\$	0,27\$
Board XBee.	3	5,36\$	16,08\$
Zocalos XBee	3	0,89\$	2,67\$
Oscilador 8MHz	3	0,71\$	2,13\$
Cable USB-mini explorer	1	4,02\$	4,02\$
LM7805	3	0,80\$	2,40\$
Zacalo 28 pines	3	0,31\$	0,93\$
Baquelita Fibra de vidrio	2	7,59\$	15,18\$
Xbee series 2	3	45\$	135\$
Transistores 2N3906	3	0,20\$	0,60\$
Ordenador de gama media.	1	450\$	500\$
Mano de Obra.			400 \$
Total Sistema			1261,38

Nota: Elaborado por Alexander Trejo.

4.8. SISTEMA DE ALERTA DE TEMPERATURA.

El sistema de alarmas de registro de temperatura corporal tiene dos tipos de alertas sonoras, la primera es para aviso de variación de temperatura en los pacientes, y la segunda es para aviso de descargas de baterías con las siguientes características:

Tabla 46.

Descripción de alarmas de temperatura corporal.

Tipo de alarmas	Sonido Midi	Color en interfaz grafica	Características luminosas
Variación de temperatura	“ Cada 10s ”	Rojo.	Parpadeo led de Lcd cada 10s
Descargas de alarmas	“ Cada 5s ”	Naranja.	Parpadeo led de Lcd cada 5s.

Nota: Elaborado por Alexander Trejo.

CAPITULO V:

5. CONCLUSIONES Y RECOMENDACIONES.

- ✚ Se elaboró un sistema de monitorización de temperatura corporal, tomando como referencia otros sistemas y recomendaciones, logrando aproximar y tener el mínimo de errores en el número de muestras.
- ✚ El transmisor de temperatura corporal al implementar un termistor NTC posee una curva características de curva logarítmica lo que elimina su linealidad, pero este principio no fue aplicado al transmisor ya que el rango de temperatura a tomar es demasiado corto por lo cual se toma como referencia parámetros lineales, los cuales mediante ecuaciones de la recta y software de programación se tiene precisión, linealidad y exactitud.
- ✚ Al momento de la utilización de una pantalla LCD de cristal líquido 2 x 8 líneas se debe tomar en cuenta que su librería de programación es diferente a la librería de una pantalla LCD de cristal líquido 2 x 16 líneas.
- ✚ Para que el transmisor de temperatura corporal posea un tamaño compacto se utilizó los dispositivos electrónicos más pequeños del mercado con las más altas características, para el dispositivo; como fue el caso del microcontrolador 18F2550 que es un dispositivo de gama alta, pero medidas reducidas.
- ✚ Al momento de la configuración de los módulos Xbee se debe configurar en el mismo pan ID y el mismo canal ya que esto asegura que no se tenga interferencias con otras tecnologías que trabajan en las misma banda de frecuencia ISM
- ✚ No se debe exponer el Transmisor de temperatura corporal a temperaturas altas ni a derrames de agua ya que esto puede traer daños permanentes, al equipo.
- ✚ Las baterías del transmisor de temperatura corporal tiene, la opción de recargar las baterías al momento de la notificación de la base de datos para mantener el sistema en constante funcionamiento con la característica principal que se las encuentra en cualquier parte del mercado tecnológico ya que son baterías de celular genéricas.

- ✚ La base de datos fue diseñada en base a la consultoría y plantillas del personal de la zona de terapia intensiva del hospital San Vicente de Paul.
- ✚ Cuando se realiza la recarga del sistema de baterías del transmisor, se lo aplica de forma normal utilizando el adaptador estandarizado de sistema de baterías.
- ✚ El uso de tecnología Xbee tienen un rango de cobertura limitada, para aplicaciones de mayor cobertura se recomienda la utilización de tecnología GSM.
- ✚ Los sistemas de alarmas auditivos se activan únicamente desde el computador, para evitar molestias al paciente en su etapa de recuperación.
- ✚ El dispositivo de transmisión de temperatura corporal, se encuentra estructurado para cambiar los dispositivos electrónicos en caso de existir daños, produciendo averías parciales y no totales.
- ✚ En caso de utilizar el sistema de monitorización de temperatura corporal en tablets o teléfonos este deben tener puertos y software que permitan el uso de aplicaciones seriales.
- ✚ El mantenimiento del sistema de temperatura corporal se debe realizar por lo menos cada seis meses y con personal capacitado en este tipo de tecnología.
- ✚ Para la implementación del sistema de temperatura corporal a gran escala se debe realizar un análisis económico junto con las debidas certificaciones y estándares internacionales.
- ✚ El uso de la tecnología Zegbee en el transmisor no produce interferencias con celulares u otros dispositivos, ya que trabajan en frecuencias y autenticaciones diferentes.

REFERENCIAS BIBLIOGRAFICAS

- Actual, M. (2010). *Termómetro con logotipo*.
<http://www.mecatronicaactual.com.br/secoes>.
- Agüero, R. (2011). *WIRELESS SENSOR NETWORKS*.
http://www.infoab.uclm.es/labeledec/solar/Comunicacion/Wifi/Interconexion_de_redes.htm.
- Andes, U. d. (2011). *Tipo de RTDS*. .
<http://webdelprofesor.ula.ve/ingenieria/damarquezg/Temp.pdf>.
- APM. (2011). *Datahet LM 324*.
- barato, Q. (2011). *Pic microcontrolador*.
http://zulia.quebarato.co.ve/maracaibo/pic-microcontrolador-pic18f2550-microchip__5FE3C4.htm.
- Botella, C. (2010). *Determinación de la temperatura corporal*.
<http://www.fisterra.com/material/tecnicas/temp/temp.asp>.
- Catarina. (2010). *Introducción a las WPANs*.
http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capítulo1.pdf.
- China, M. i. (2011). *Baterías Genéricas*. http://es.made-in-china.com/co_cnjuzhao/product_Cell-Phone-Battery-for-Nokia-8800-Arte-E66_huururegn.html.
- desarrollo, C. a. (2011). *Historia del lenguaje Java*.
http://www.cad.com.mx/hitoria_del_lenguaje_java.html.
- Digi. (2011). *Xbee series 2*. www.digi.com.

- Digi. (2011). *Xbee series 2*. <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/xbee-digimesh-2-4#overview>.
- Digikey. (2010). *Distribución de Termistores*. <http://ordering.digikey.com/>.
- Directindustry. (2010). *Características de Termopílas*. <http://www.directindustry.es/prod/perkinelmer-optoelectronics/termopares-infrarrojos>.
- Elane. (2011). *LCD 2x8*. www.elane.net.
- Ellab. (2010). *Descripción de Equipos*. <http://www.ellab.com/Default.aspx?ID=386>.
- Embajadores, E. (2011). *Características Xbee Xplorer*. <http://www.electronicaembajadores.com/Productos/xbee-explorer-usb>.
- Embajadores, E. (2011). *Características Xbee Xplorer*. [http://www \(Embajadores, 2011\).electronicaembajadores.com/Productos/xbee-explorer-usb](http://www.electronicaembajadores.com/Productos/xbee-explorer-usb).
- Erazo, C. (2009). *Diseño y construcción de un módulo inalámbrico, para la adquisición y procedimiento, en tiempo real, de las variables fisiológicas requeridas en atención prehospitalaria*. Quito.
- Espinal, N. (2010). *Fiebre*. <http://www.infermeravirtual.com/es-es/problemas-de-salud/signos-sintomas/fiebre/informacion-relacionada.html>.
- Estudio, I. (2010). *Sensor de Temperatura LM35*. <http://consultaing.blogspot.com/2009/10/sensor-de-temperatura-lm35.htm>.
- Exacon. (2010). *Sensores y Probadores de Temperatura*. <http://www.exacon.com/Global/Global11>.
- Expemed. (2011). *Expediente medio Digital*. <http://www.expemed.net/>.

- Ferreira, M. (2010). *Termoregulación* .
<http://200.89.72.103/documentos/deporte/manual/pd>.
- Fluke. (2011). *Milímetro Fluke* . <http://www.fluke.com/fluke>.
- Godoy, A. (2011). *Tira electrónica que mide la temperatura corporal* .
<http://alvarogodoy.bligoo.com/>.
- Guatewireless-Org. (2011). *La Revolución open Mesh* .
<http://www.guatewireless.org/internetworking/redes/wireless/la-revolucion-open-mesh/>.
- H, C. J. (2011). *Skin temperature radio telemetry and alarms* .
<http://v3.espacenet.com>.
- Hadaschik, N. (2010). *Techniques for UWB-OFDM* . http://www.iss.rwth-aachen.de/Projekte/Theo/OFDM/OFDM_en.html.
- licenciamient. (2011). *El portal del Software*. www.licenciamiento.net.
- Lozada, G. (2010). *Temperatura Corporal*.
<http://www.slideshare.net/anama.krpio/temperatura-corporal-presentation>.
- Medical, S. (2010). *Catalogo YSI 427*. <http://www.sorevan.com>.
- mercurio, T. d. (2011). *Termómetros de mercurio*.
http://es.wikipedia.org/termometros_de_mercurio.
- Mimemoria.net. (2011). *Clases de amplificadores*.
<http://mimemoria.net/2010/11/lm324-o-amplificador-operacional/>.
- Monografía. (2011). *Arquitectura Cliente Servidor*.
<http://www.monografías.com/trabajos24/arquitectura-clienteservidor>.
- Oficio, T. (2011). *Termómetros de Mercurio*.
<http://www.tecnoficio.com/docs/doc63.php>.

- Omega. (2010). *Aplicación y precisión de Rtds* . <http://www.omega.com/>.
- OYARCE, A. (2010). *Guía el usuario Xbee Olimex*.
- PCexpertos. (2011). *Conectores seriales*. <http://pcexpertos.com/tag/cd/page/2v>.
- PlanetNetbeans. (2011). *Plataforma Netbeans*.
<http://www.planetnetbeans.org/es/>.
- Plata, U. M. (2010). *Transductores como elemento de Control*.
<http://www.fi.mdp.edu.ar/>.
- Pmedical. (2010). *Características Técnicas de Sensores*.
<http://www.pmedicalonline.net/servlet/the-Philips-Medical-cln-6.-Sondas-Temperatura/Categories>.
- Review, T. (2011). *Implante Coronario*. <http://avances-tecnologicos.euroresidentes.com/2011/02/implante-coronario-inalambrico.html>.
- Roldan, D. (2009). *Comunicaciones Inalambricas*. Mexico: Alfaomega.
- Roncacio, H. (2008). *Tutorial de Labview*. Universidad Distrital "Francisco José de Calas".
- S, K. H., & PAUL, J. (2001). *Portable remote patient telemonitoring system* .
<http://v3.espacenet.com>.
- Scribd. (2011). *Telemétrica por medio de control electrónico* .
<http://www.scribd.com/doc/52643532/44/Microcontrolador-18F2550>.
- Society, I. C. (Septiembre). *Radio Communications. IEEE Communications* .
- System., G. O. (2011). *La definición del software libre*.
[<http://www.gnu.org/philosophy/free-sw.es.html>].

- Taringa. (2011). *Diagnostico de enfermedades Cardiacas.* .
<http://www.taringa.net/posts/info/835832/Diagnostico-de-enfermedades-cardiacas.html>.
- Termómetro, E. (2011). *Termómetro Digital Flexible.*
<http://www.termometro.es/termometro-flexible.php> .
- Tobón, C. (2010). *Fiebre.* <http://www.aibarra.org/Guias/7-24.htm>.
- Vago, R. d. (2010). *Pirómetros de Radiación y Ópticos.*
<http://html.rincondelvago.com/pirometros-de-radiacion-y-opticos.html>.
- Vocal, T. (2010). *802.11a Wireless LAN.*
http://www.vocal.com/redirect/802_11a.html.
- Wifi, L. (2010). *Descripción para las redes Wi-Fi IEEE.802.11.*
<http://www.laserwifi.com/estander802.11.html>.
- Wikipedia. (2011). *MySQL.* <http://es.wikipedia.org/wiki/MySQL>.
- Wikipedia. (2011). *Red de Sensores.*
http://es.wikipedia.org/wiki/Red_de_sensores.
- Wikipedia. (2011). *Mobile ad hoc network.*
http://es.wikipedia.org/wiki/Red_movil_ad_hoc.
- Wikipedia. (2011). *Modelo cliente servidor.* <http://es.wikipedia.org/wiki/cliente-servidor>.
- Wikipedia. (2011). *Modelo Entidad Relación.*
http://es.wikipedia.org/wiki/Modelo_entidad.
- Wikipedia. (2011). *Software Libre.* http://es.wikipedia.org/wiki/Software_libre.
- Wkipedia. (2010). *Infrared Data Association.*
http://es.wikipedia.org/wiki/Infrared_Data_Association.

- Xbee. (2011). *Xbee series 2*. www.xbee.com.
- Zappala, D. (2011). *Freedom from Wires*.
<http://digitalcivilization.blogspot.com/2010/11/freedom-from-wires.html>.

Anexo A

**CARACTERISTICAS DEL ORDENADOR DEL SISTEMA DE
MONITORIZACION DE TEMPERATURA CORPORAL**

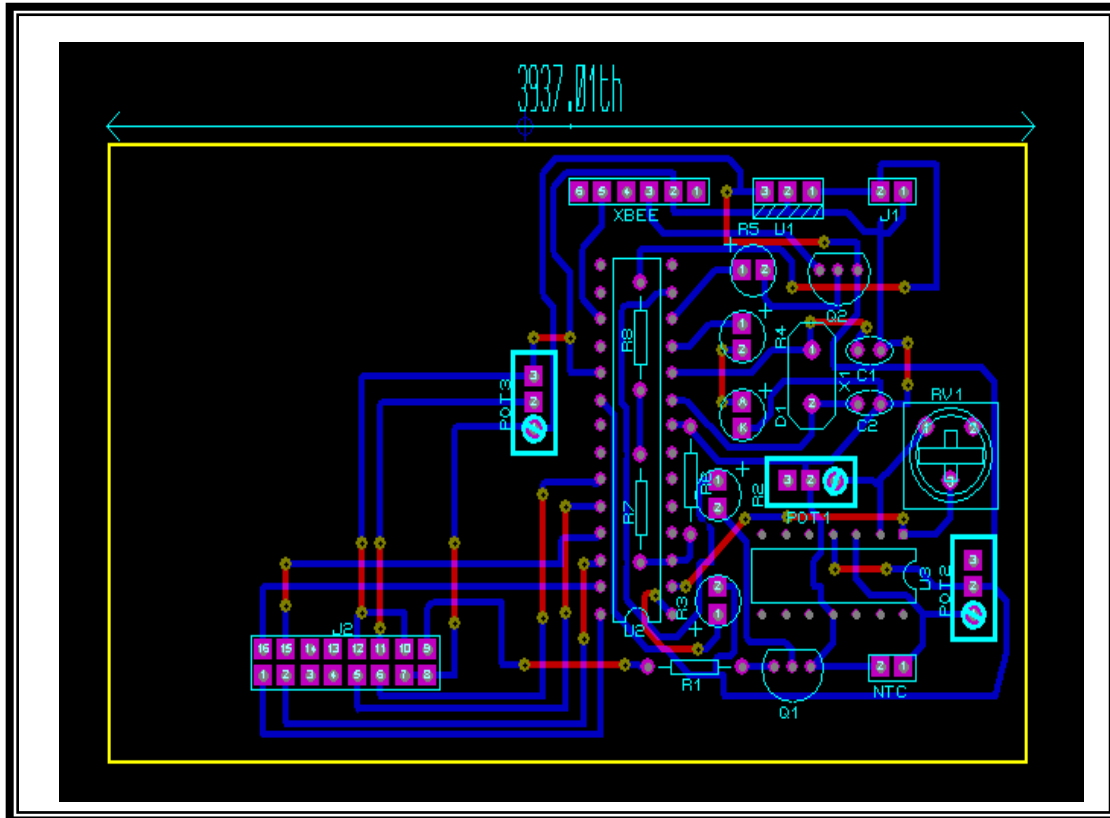


El computador para la implementación de la base de datos de monitorización de temperatura corporal debe poseer las características de un ordenador de gama media como las que se menciona a continuación:

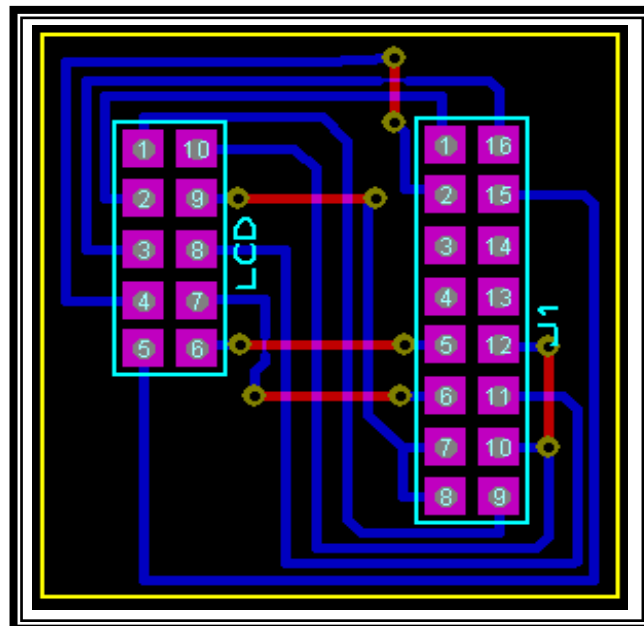
Características Ordenador Temperatura Corporal	
Procesador	Intel® Pentium® processor G630(2.70GHz,3MB L2,1066MHz)
Memoria	4 GB ² Dos canales SDRAM DDR3 a 1333 MHz
Disco Duro	500GB Serial ATA (7200RPM) con/DataBurst Cache™
Tarjeta de Video	Intel® HD Graphics
Monitor	LG/SAMSUM/DELL de 21"
Unidad Óptica	16X DVD+/-RW Drive
Sistema Operativo	Windos xp servipack 3

Anexo B

**DISEÑO DEL TRANSMISOR DE TEMPERATURA
CORPORAL PARA IMPRESIÓN EN BAQUELITA**



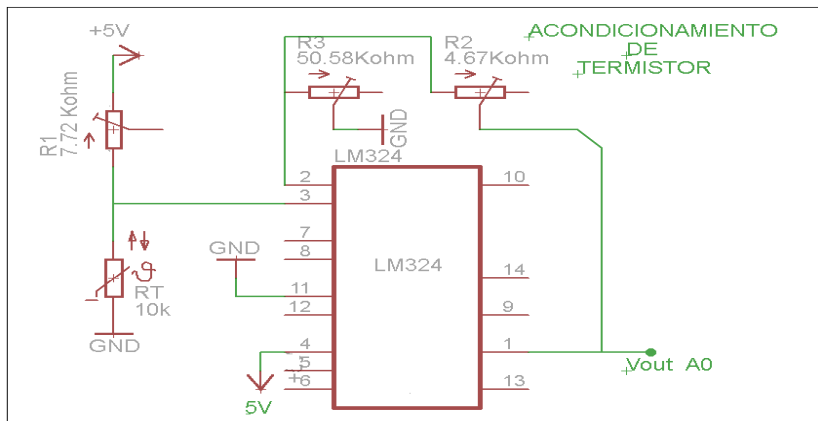
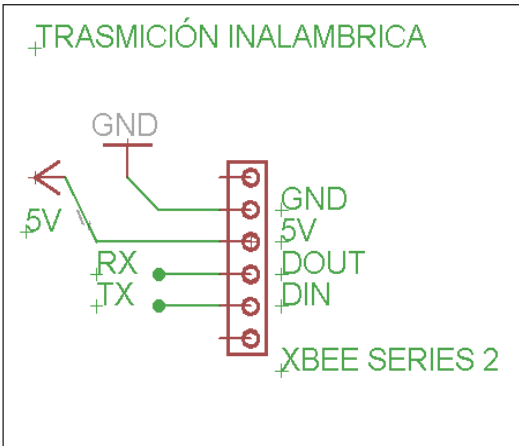
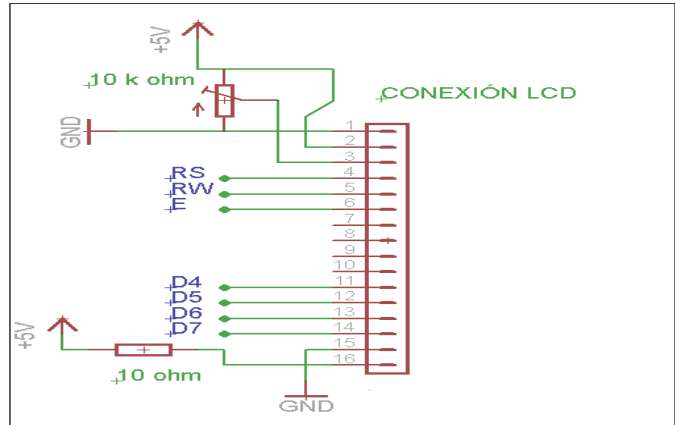
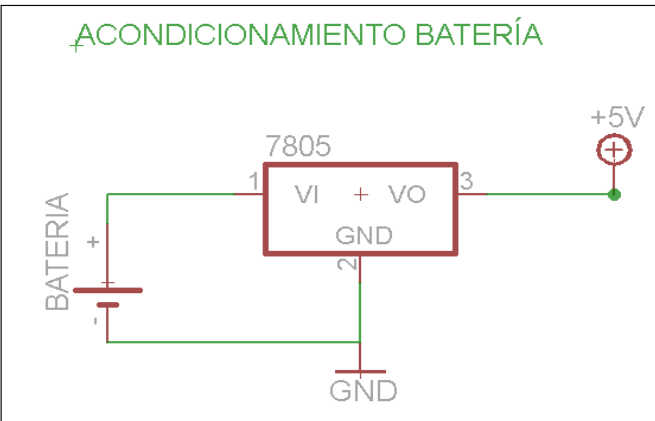
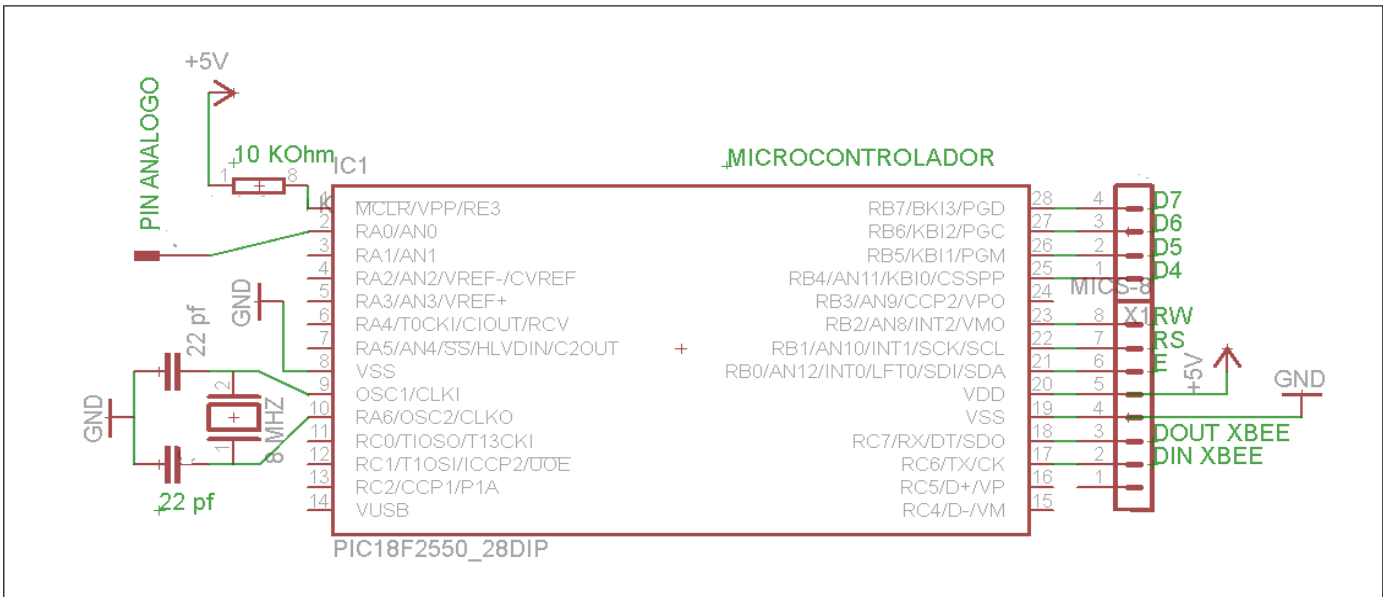
Diseño de transmisor en Proteus.



Diseño de LCD en Proteus.

Anexo C

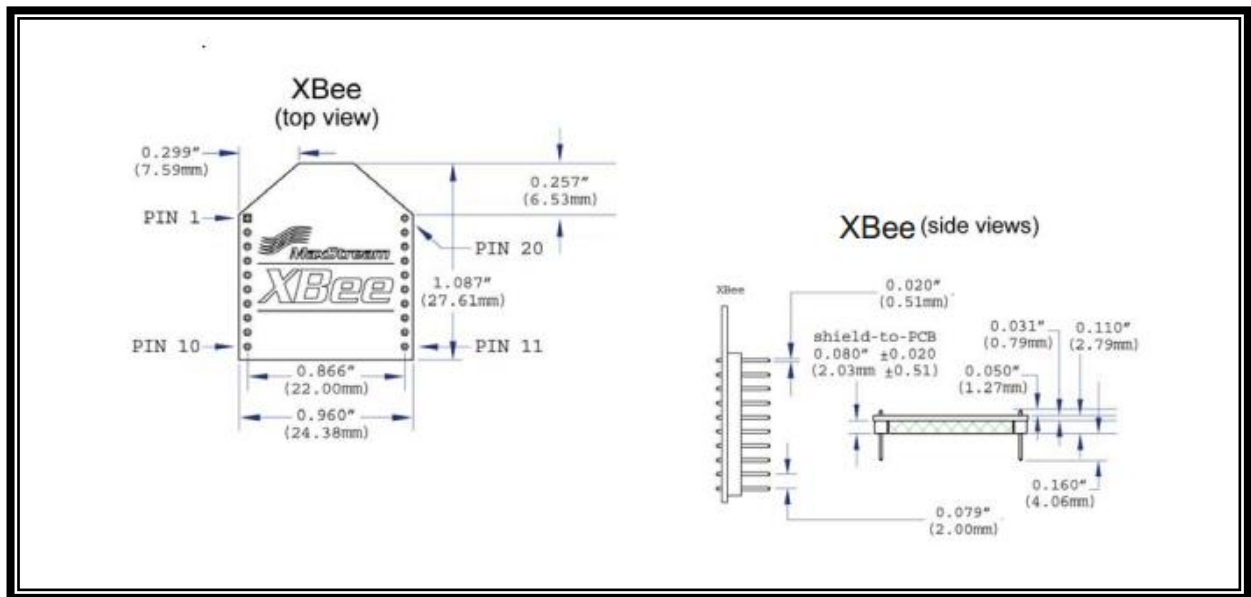
DISEÑO CIRCUITO EAGLE NORMAL



Diseños simulación electrónica.

Anexo D

CARACTERISTICAS DEL XBEE SERIES 2



Especificaciones Xbee Series 2	
Rango Urbano interior	40m
Rango urbano exterior línea recta	120m
Potencia de transmisión de salida	2 Mw
Velocidad de Transmisión	250,000 bps.
Velocidad de puerto	1200230400 bps
Sensibilidad Receptor	-95 dBm (1% de paquete por error)
Requerimientos de Fuente de alimentación	
Voltaje de Alimentación	2.8 – 3.4 V
Corriente de operación de transmisión	40 mA @ 3.3 V
Corriente de operación de Recepción	40 mA @ 3.3 V
General	
Banda de frecuencia de operación	ISM 2.4 GHz
Dimensiones	0,960" x 1087"
Operación de Temperatura.	-45 ° a 85° C
Opción de Antenas	Wchip, Chip, RPMSA

Redes Y Seguridad	
Topologías de Red soportadas	Punto – Punto, Punto a Multipunto Peer to Peer, Mesh
Número de Canales	16 Secuencias de Canales directos
Opción de Direcciones	Pan Id y cluster Ids y puntos Finales
Agencias de Aprobación	
Estados Unidos	Pendiente
Industrias Canadá	Pendiente
Europa	Pendiente

Símbolo	Parámetro	Condición	Mínimo	Típico	Max	Unidades
V _{IL}	Entrada Bajo de Voltaje	Todas la Entradas Digitales	-	-	0.2*VCC	V
V _{IH}	Entrada Alto de Voltaje	Todas la Entradas Digitales	0.8*VCC	-	0.18*VCC	V
V _{OL}	Salida Bajo de Voltaje	2mA a 2.7 V		-	0.18*VCC	V
V _{OH}	Salida Alto de Voltaje	-2mA a 2.7 V	0.82*VCC	-	-	V
TX	Voltaje de Transmisión	3,3 V	-	45	-	V
RX	Voltaje de Recepción	3,3 V	-	50	-	V
PWR-DWN	Corriente de encendido	Parámetros SM = 1.	-	< 10	-	µA.

Anexo E

**PROGRAMACIÓN “CCS” DEL SISTEMA DE
TEMPERARURA CORPORAL**

Código Transmisor.

```
#include <18f2550.h>
```

```
#device adc=10
```

```
#fuses
```

```
HS,MCLR,NOWDT,NOPROTECT,NOPUT,NOBROWNOUT,NOLVP,NOCPD,NODEBUG,NOWRT,NOV  
REGEN
```

```
#use delay(clock=8M)
```

```
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
```

```
#define set output_high
```

```
#define reset output_low
```

```
#use standard_io(a)
```

```
#use standard_io(b)
```

```
#use standard_io(c)
```

```
#byte porta=0x05
```

```
#byte portb=0x06
```

```
#byte portc=0x07
```

```
#define xbee pin_c1
```

```
#define LCD pin_c2
```

```
#include "flex_lcd.c"
```

```
//#include <lcd.c>
```

```
int8 envio=0;
```

```
int16 cont=0;
```

```
int16 ADC_res2=0;

int16 ADC_prom2=0;

float voltio1=0;

int16 ADC_res=0;

int16 ADC_prom=0;

float voltio=0;

float temperatura=0;

int16 contRTCC=0;

int8 valor=20,seg=50,min=0;

int8 enviar=0;

#int_Timer1

void temp1s()

{

    set_timer1(15536);

    if(valor<4)

        valor++;

    else

    {

        if(seg<59)

        { seg++;

            if(seg==5 && enviar==1)

            {enviar=0;

                }

        }

        if(temperatura<35.5 || temperatura>37.5 )
```



```
{if (temperatura<1)

    {temperatura=0;

        envio=0;}

else

    { envio=1;}

}

    if(temperatura>=35.5 && temperatura<=37.5 )

        { envio=0;}

    }

else

{seg=0;

    if(min<1)

        min++;

    else

        {

            enviar=1;

        }

    }

    valor=0;

}

set_adc_channel(0);

cont++; //INCREMENTA EL CONTADOR

ADC_res=ADC_res+read_adc(); // SUMATORIA DE VALORES

if(cont==100)
```

```
{  
  
cont=0x00;  
  
ADC_prom=ADC_res/101;  
  
voltio=5.0*ADC_prom/1024.0;  
  
if((voltio>=2.09)&&(voltio<=2.20))  
{  
    temperatura=(-2*(voltio -2.09)/0.11)+42;  
}  
else if((voltio>2.20)&&(voltio<=2.33))  
{  
    temperatura=(-2*(voltio -2.20)/0.13)+40;  
}  
else if((voltio>2.33)&&(voltio<=2.44))  
{  
    temperatura=(-2*(voltio -2.33)/0.11)+38;  
}  
else if((voltio>2.44)&&(voltio<=2.62))  
{  
    temperatura=(-3*(voltio -2.44)/0.18)+36;  
}  
else
```

```
{  
    temperatura=0.00;  
}  
  
lcd_gotoxy(1,1);  
printf(lcd_putc,"\fT=%01.2f",temperatura);  
ADC_res=0;  
}  
  
set_adc_channel(1);  
ADC_res2=read_adc();  
voltio1=5.0*ADC_res2/1024.0;  
  
    if((voltio1>=1) && (voltio1<=3.75))  
    {  
        LCD_GOTOXY(1,1);  
        printf(lcd_putc,"\fV=%01.2f",voltio1);  
        LCD_GOTOXY(1,2);  
        printf(lcd_putc,"\fBAJO VOL");  
    }  
else  
    {  
        voltio1=0;  
    }  
}  
  
void main ()
```

```
{  
  
    set_tris_a(0x03);  
  
    set_tris_b(0x00);  
  
    set_tris_c(0x00);  
  
    portc=0;  
  
    delay_ms(2000);  
  
    lcd_init();  
  
    delay_ms(10);  
  
    setup_comparator(NC_NC_NC_NC);  
  
    setup_vref(FALSE);  
  
    setup_timer_1(T1_INTERNAL|T1_DIV_by_8);  
  
    set_timer1(55036);  
  
    enable_interrupts(INT_timer1);  
  
    enable_interrupts(GLOBAL);  
  
    setup_adc(adc_clock_internal);  
  
    delay_ms(50);  
  
    SETUP_ADC_PORTS(AN0_TO_AN1);  
  
    delay_us(20);  
  
    set(xbee);  
  
    CONT=0;  
  
    output_bit( PIN_c1, 0);  
  
    delay_ms(1000);  
  
    printf("hola mundo");  
}
```

```
do
{
    lcd_gotoxy(1,2);

    printf(lcd_putc,"s %u m %u ",seg,min);

    delay_ms(100);

    if(envio==1)
    {
        output_bit( PIN_c0, 1);

        output_bit( PIN_c1, 0);

        output_bit( PIN_c2, 1);

        delay_ms(100);

        printf("IA3T%f \n",temperatura);

        output_bit( PIN_c0, 0);

        output_bit( PIN_c6, 0);

        output_bit( PIN_c2, 0);

        output_bit( PIN_c1, 0);
    }

    if(enviar==1)
    {
        if(temperatura>1)
        {
            output_bit( PIN_c0, 1);

            output_bit( PIN_c1, 0);
```

```
    output_bit( PIN_c2, 1);  
    delay_ms(100);  
    printf("IA3T%f \n",temperatura);  
    output_bit( PIN_c1, 0);  
    output_bit( PIN_c6, 0);  
    output_bit( PIN_c2, 0);  
    output_bit( PIN_c1, 0);  
    }  
    }  
  
}while(TRUE);  
  
}
```

Anexo F

LIBRERÍA EN “CCS” DE LA PANTALLA LCD 2 x 8

Código LCD.

```
// flex_lcd.c

// These pins are for the Microchip PicDem2-Plus board,
// which is what I used to test the driver. Change these
// pins to fit your own board.

#define LCD_DB4 PIN_B4

#define LCD_DB5 PIN_B5

#define LCD_DB6 PIN_B6

#define LCD_DB7 PIN_B7

#define LCD_RS PIN_B2

// #Define LCD_RW PIN_B0

#define LCD_E PIN_B3

// If you only want a 6-pin interface to your LCD, then
// connect the R/W pin on the LCD to ground, and comment
// out the following line.

// #define USE_LCD_RW 1

// =====

#define lcd_type 2 // 0=5x7, 1=5x10, 2=2 lines

#define lcd_line_two 0x40 // LCD RAM address for the 2nd line

int8 const LCD_INIT_STRING[4] =
{
    0x20 | (lcd_type << 2), // Func set: 4-bit, 2 lines, 5x8 dots
    0xc, // Display on
```



```

1,          // Clear display

6          // Increment cursor

};

//-----

void lcd_send_nibble(int8 nibble)

{

// Note: !! converts an integer expression

// to a boolean (1 or 0).

output_bit(LCD_DB4, !(nibble & 1));

output_bit(LCD_DB5, !(nibble & 2));

output_bit(LCD_DB6, !(nibble & 4));

output_bit(LCD_DB7, !(nibble & 8));

delay_cycles(1);

output_high(LCD_E);

delay_us(2);

output_low(LCD_E);

}

//-----

// This sub-routine is only called by lcd_read_byte().

// It's not a stand-alone routine. For example, the

// R/W signal is set high by lcd_read_byte() before

// this routine is called.

```

```
#ifdef USE_LCD_RW

int8 lcd_read_nibble(void)

{

int8 retval;

// Create bit variables so that we can easily set

// individual bits in the retval variable.

#bit retval_0 = retval.0

#bit retval_1 = retval.1

#bit retval_2 = retval.2

#bit retval_3 = retval.3

retval = 0;

output_high(LCD_E);

delay_cycles(1);

retval_0 = input(LCD_DB4);

retval_1 = input(LCD_DB5);

retval_2 = input(LCD_DB6);

retval_3 = input(LCD_DB7);

output_low(LCD_E);

return(retval);

}

#endif
```

```
// Read a byte from the LCD and return it.
```

```
#ifdef USE_LCD_RW
```

```
int8 lcd_read_byte(void)
```

```
{
```

```
int8 low;
```

```
int8 high;
```

```
output_high(LCD_RW);
```

```
delay_cycles(1);
```

```
high = lcd_read_nibble();
```

```
low = lcd_read_nibble();
```

```
return( (high<<4) | low);
```

```
}
```

```
#endif
```

```
// Send a byte to the LCD.
```

```
void lcd_send_byte(int8 address, int8 n)
```

```
{
```

```
output_low(LCD_RS);
```

```
#ifdef USE_LCD_RW
```

```
while(bit_test(lcd_read_byte(),7) );
```

```
#else
```

```
delay_us(60);
```

```
#endif
```

```
if(address)
    output_high(LCD_RS);
else
    output_low(LCD_RS);
    delay_cycles(1);
#ifdef USE_LCD_RW
    output_low(LCD_RW);
    delay_cycles(1);
#endif
    output_low(LCD_E);
    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}

void lcd_init(void)
{
    int8 i;
    output_low(LCD_RS);
#ifdef USE_LCD_RW
    output_low(LCD_RW);
#endif
    output_low(LCD_E);
```

```
delay_ms(15);

for(i=0 ;i < 3; i++)

{

    lcd_send_nibble(0x03);

    delay_ms(5);

}

lcd_send_nibble(0x02);

for(i=0; i < sizeof(LCD_INIT_STRING); i++)

{

    lcd_send_byte(0, LCD_INIT_STRING[i]);

    #ifndef USE_LCD_RW

    delay_ms(5);

    #endif

}

}

void lcd_gotoxy(int8 x, int8 y)

{

int8 address;

if(y != 1)

    address = lcd_line_two;

else

    address=0;

address += x-1;

lcd_send_byte(0, 0x80 | address);
```

```
}  
  
void lcd_putc(char c)  
{  
    switch(c)  
    {  
        case '\f':  
            lcd_send_byte(0,1);  
            delay_ms(2);  
            break;  
        case '\n':  
            lcd_gotoxy(1,2);  
            break;  
        case '\b':  
            lcd_send_byte(0,0x10);  
            break;  
        default:  
            lcd_send_byte(1,c);  
            break;  
    }  
}  
  
#ifdef USE_LCD_RW  
  
char lcd_getc(int8 x, int8 y)  
{
```

```
char value;

lcd_gotoxy(x,y);

// Wait until busy flag is low.

while(bit_test(lcd_read_byte(),7));

output_high(LCD_RS);

value = lcd_read_byte();

output_low(lcd_RS);

return(value);

}

#endif

void lcd_setcursor_vb(short visible, short blink) {

    lcd_send_byte(0, 0xC|(visible<<1)|blink);

}
```

Anexo G

LENGUAJE DE PROGRAMACION DE LA BASE DE DATOS

BASE DE DATOS.

Código Conexión Datos App Medico.

```
package aplicacion;

import java.sql.*;

public class Conexion {

    public static Connection Conec;

    Statement St;

    ResultSet reg;

    public Conexion() {

        try {

            Class.forName("com.mysql.jdbc.Driver");

            String url = "jdbc:mysql://localhost:3306/bddmedico";

            Conec = (Connection) DriverManager.getConnection(url, "root", "root");

            } catch (Exception ex) {

                System.out.println("Error: " + ex.getMessage());

            }

        }

    public String ejecutar(String sql) {

        String error = "";

        try {

            St = Conec.createStatement();

            St.execute(sql);

            error="ok";

        } catch (Exception ex) {
```

```

        error = ex.getMessage();
    }

    return (error);
}

public ResultSet consulta(String sql) {

    String error = "";

    try {

        St = (Statement) Conec.createStatement();

        reg = (ResultSet) St.executeQuery(sql);

    } catch (Exception ex) {

        error = ex.getMessage();

    }

    return (reg);

}
}

```

Código Graficos App Medico.

```

package aplicacion;

import com.mysql.jdbc.ResultSet;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.GradientPaint;

import org.jfree.chart.ChartFactory;

```

```

import org.jfree.chart.ChartFrame;

import org.jfree.chart.ChartPanel;

import org.jfree.chart.JFreeChart;

import org.jfree.chart.axis.CategoryAxis;

import org.jfree.chart.axis.CategoryLabelPositions;

import org.jfree.chart.axis.NumberAxis;

import org.jfree.chart.plot.CategoryPlot;

import org.jfree.chart.plot.PlotOrientation;

import org.jfree.chart.renderer.category.BarRenderer;

import org.jfree.data.category.CategoryDataset;

import org.jfree.data.category.DefaultCategoryDataset;

import org.jfree.data.general.DefaultPieDataset;

public class Graficos extends javax.swing.JFrame {

    Conexion con = new Conexion();

    int cama;

    String nombreCama;

    static String nomCama="";

    public Graficos(String title,int cama) {

        super(title);

        this.cama=cama;

        this.nombreCama=title;

        CategoryDataset dataset = crearDataset();

        JFreeChart chart = crearChart(dataset);

        ChartPanel panel = new ChartPanel(chart, false);

```

```

panel.setPreferredSize(new Dimension(800 , 600));

this.setContentPane(panel);

}

private CategoryDataset crearDataset() {

    DefaultCategoryDataset dataset = new DefaultCategoryDataset();

    try {

        //Cambio de cama

        ResultSet res1 = getDatos(cama);

        boolean verificarE1 = res1.next();

        if (!verificarE1) {

        } else {

            while (verificarE1) {

                String [] fCompleta=res1.getString("fecha_analisis").split(" ");

                String [] fHora=fCompleta[1].split(":");

                dataset.addValue(Double.valueOf(res1.getString("temperatura")), "",fHora[0]+":"+fHora[1]);

                verificarE1 = res1.next();

            }

        }

        dataset.getRowCount();

    } catch (Exception e) {

```

```

    }

    return dataset;
}

private static JFreeChart crearChart(CategoryDataset data) {

    JFreeChart chart = ChartFactory.createLineChart(

        nomCama, //Nombre de la gráfica

        "FECHA ANALISIS", //Nombre del eje Horizontal

        "TEMPERATURA", //Nombre del eje vertical

        data, //Data

        PlotOrientation.VERTICAL, //Orientación

        true, //Incluir leyenda

        true, //Información al pasar el mouse

        true

    );

    return chart;
}

public ResultSet getDatos(int cama)

    String consulta = "SELECT temperatura,fecha_analisis FROM tab_ingreso_terapia i, tab_analisis a
    WHERE estado_cama =1 AND cama =" + cama + " and i.cod_ingreso=a.cod_ingreso order by
    fecha_analisis DESC";

    ResultSet res = (ResultSet) con.consulta(consulta);

    return res;
}
}

```

Código Login App Médico.

```
package aplicacion;

import com.mysql.jdbc.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import javax.swing.JOptionPane;

public class frmLogin extends javax.swing.JFrame {

    Conexion con =new Conexion();

    private String tipoUs;

    frmPrincipal prin;

    public frmLogin() {

        initComponents();

        prin=new frmPrincipal();

    }

    @SuppressWarnings("unchecked")

    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();

        txtUsuario = new javax.swing.JTextField();

        jLabel1 = new javax.swing.JLabel();

        jLabel2 = new javax.swing.JLabel();

        btnIngresar = new javax.swing.JButton();

        btnCancelar = new javax.swing.JButton();

        txtPassword = new javax.swing.JPasswordField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
setTitle("FORMULARIO DE ACCESO");

setBackground(new java.awt.Color(153, 153, 153));

setResizable(false);

addWindowListener(new java.awt.event.WindowAdapter() {

    public void windowOpened(java.awt.event.WindowEvent evt) {

        formWindowOpened(evt);

    }

});

jPanel1.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));

txtUsuario.setText("alex");

jLabel1.setText("USUARIO:");

jLabel2.setText("PASSWORD:");

btnIngresar.setText("INGRESAR");

btnIngresar.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnIngresarActionPerformed(evt);

    }

});

btnCancelar.setText("CANCELAR");

btnCancelar.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        btnCancelarActionPerformed(evt);
    }
});

txtPassword.setText("alex");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(21, 21, 21)
                    .addComponent(jLabel1))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(21, 21, 21)
                    .addComponent(jLabel2)))
            .addContainerGap(210, true))
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(21, 21, 21)
            .addComponent(txtPassword, javax.swing.GroupLayout.DEFAULT_SIZE, 210,
                true))
    );
```



```

Short.MAX_VALUE)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())

        .addComponent(btnIngresar)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(btnCancelar)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,      36,
Short.MAX_VALUE))

        .addComponent(txtUsuario,      javax.swing.GroupLayout.DEFAULT_SIZE,      210,
Short.MAX_VALUE))

        .addGap(14, 14, 14))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addComponent(txtUsuario,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtPassword,      javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(jLabel2))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(btnIngresar)

        .addComponent(btnCancelar)))

        .addComponent(jLabel1))

        .addContainerGap()

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

            .addContainerGap()

        );

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

```

```
        .addComponent(jPanel1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(14, Short.MAX_VALUE))

    );

    pack();

private void formWindowOpened(java.awt.event.WindowEvent evt) {

    this.setLocationRelativeTo(null);

    this.setVisible(true);

}

private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {

    System.exit(0);

}

public String SacarPass(char[] pass)

{

    String cadena="";

    for(int i=0;i<pass.length;i++)

    {

        cadena=cadena+pass[i];

    }

    return cadena;

}

private void btnIngresarActionPerformed(java.awt.event.ActionEvent evt) {

    try {
```

```

String usuario = txtUsuario.getText();

char[] password = txtPassword.getPassword();

String pass= SacarPass(password);

String consulta="select * from tab_usuario where nombre="+usuario+"and password="+pass+"";

ResultSet tipoUsuario=(ResultSet)con.consulta(consulta);

boolean lleno=tipoUsuario.next();

int columnaTipo= tipoUsuario.findColumn("tipo");

if(!lleno)

    JOptionPane.showMessageDialog(this, "Usuario no existe");

else

while(lleno)

{

    setTipoUs(tipoUsuario.getString(columnaTipo));

    lleno=tipoUsuario.next();

    txtUsuario.setText(getTipoUs());

}

if (getTipoUs().equals("1") ) {

    prin.setVisible(true);

    this.setVisible(false);

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(this, "Error");

```

```

    }
}

public static void main(String args[]) {

    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {

            new frmLogin().setVisible(true);

        }

    });

}

private javax.swing.JButton btnCancelar;

private javax.swing.JButton btnIngresar;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JPanel jPanel1;

private javax.swing.JPasswordField txtPassword;

private javax.swing.JTextField txtUsuario;

public String getTipoUs() {

    return tipoUs;

}

public void setTipoUs(String tipoUs) {

    this.tipoUs = tipoUs;

}

```

Función Principal App Medico.

```
package aplicacion;

import com.mysql.jdbc.ResultSet;

import java.awt.event.ActionEvent;

import javax.swing.*.*;

import giovynet.nativelink.SerialPort;

import giovynet.serial.Baud;

import giovynet.serial.Com;

import giovynet.serial.Parameters;

import java.awt.Color;

import java.awt.GradientPaint;

import java.awt.event.ActionListener;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import java.awt.event.*;

import org.jfree.ui.RefineryUtilities;

public class frmPrincipal extends javax.swing.JFrame implements ActionListener {

    int cuentaC = 0;

    String cedula;

    Conexion con = new Conexion();
```

```
public frmPrincipal() {  
  
    initComponents();  
  
    cedula = "";  
  
}  
  
private void initComponents() {  
  
    jPopupMenu1 = new javax.swing.JPopupMenu();  
  
    pnlContenedor = new javax.swing.JTabbedPane();  
  
    jPanel1 = new javax.swing.JPanel();  
  
    jPanel3 = new javax.swing.JPanel();  
  
    jLabel1 = new javax.swing.JLabel();  
  
    txtCedula = new javax.swing.JTextField();  
  
    txtNombres = new javax.swing.JTextField();  
  
    jLabel2 = new javax.swing.JLabel();  
  
    txtSangre = new javax.swing.JTextField();  
  
    jLabel3 = new javax.swing.JLabel();  
  
    jLabel4 = new javax.swing.JLabel();  
  
    txtFechaNac = new org.freixas.jcalendar.JCalendarCombo();  
  
    btnBuscar = new javax.swing.JButton();  
  
    jPanel4 = new javax.swing.JPanel();  
  
    jLabel5 = new javax.swing.JLabel();  
  
    jLabel6 = new javax.swing.JLabel();  
  
    jLabel7 = new javax.swing.JLabel();  
  
    jLabel8 = new javax.swing.JLabel();  
  
    jScrollPane1 = new javax.swing.JScrollPane();
```

```
txtDiagnostico = new javax.swing.JTextArea();

txtPeso = new javax.swing.JTextField();

txtTalla = new javax.swing.JTextField();

txtFechaIngreso = new org.freixas.jcalendar.JCalendarCombo();

jPanel5 = new javax.swing.JPanel();

jLabel9 = new javax.swing.JLabel();

jLabel12 = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jLabel14 = new javax.swing.JLabel();

txtProcedencia = new javax.swing.JTextField();

txtEstadoPiel = new javax.swing.JTextField();

cbxCama = new javax.swing.JComboBox();

txtFechaHospitalizacion = new org.freixas.jcalendar.JCalendarCombo();

btnGuardar = new javax.swing.JButton();

jPanel2 = new javax.swing.JPanel();

jPanel6 = new javax.swing.JPanel();

jLabel10 = new javax.swing.JLabel();

txtTemperaturaC1 = new javax.swing.JTextField();

jLabel11 = new javax.swing.JLabel();

lblNombresC1 = new javax.swing.JLabel();

lblFechaC1 = new javax.swing.JLabel();

jLabel15 = new javax.swing.JLabel();

jLabel16 = new javax.swing.JLabel();

jPanel7 = new javax.swing.JPanel();
```



```
jLabel17 = new javax.swing.JLabel();

txtTemperaturaC2 = new javax.swing.JTextField();

jLabel18 = new javax.swing.JLabel();

lblNombresC2 = new javax.swing.JLabel();

lblFechaC2 = new javax.swing.JLabel();

jLabel19 = new javax.swing.JLabel();

jLabel20 = new javax.swing.JLabel();

jPanel9 = new javax.swing.JPanel();

jLabel25 = new javax.swing.JLabel();

txtTemperaturaC3 = new javax.swing.JTextField();

jLabel26 = new javax.swing.JLabel();

jLabel24 = new javax.swing.JLabel();

lblNombresC3 = new javax.swing.JLabel();

jLabel27 = new javax.swing.JLabel();

lblFechaC3 = new javax.swing.JLabel();

jLabel21 = new javax.swing.JLabel();

jLabel22 = new javax.swing.JLabel();

jLabel23 = new javax.swing.JLabel();

btnIniciar = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

listCama2 = new javax.swing.JList();

jScrollPane3 = new javax.swing.JScrollPane();

listCama1 = new javax.swing.JList();

jScrollPane4 = new javax.swing.JScrollPane();
```

```

listCama3 = new javax.swing.JList();

jLabel28 = new javax.swing.JLabel();

jPanel8 = new javax.swing.JPanel();

btnCargarC1 = new javax.swing.JButton();

btnCargarC2 = new javax.swing.JButton();

btnCargarC3 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setTitle("CONTROL DE TEMPERATURA");

addWindowListener(new java.awt.event.WindowAdapter() {

    public void windowOpened(java.awt.event.WindowEvent evt) {

        formWindowOpened(evt);

    }

});

```

```

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(), "DATOS PERSONALES", javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 51, 153))); // NOI18N

```

```

jPanel3.setForeground(new java.awt.Color(0, 0, 255));

jLabel1.setText("CEDULA");

txtCedula.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        txtCedulaActionPerformed(evt);

    }

});

```

```
txtCedula.addFocusListener(new java.awt.event.FocusAdapter() {  
    public void focusLost(java.awt.event.FocusEvent evt) {  
        txtCedulaFocusLost(evt);  
    }  
});  
  
txtCedula.addPropertyChangeListener(new java.beans.PropertyChangeListener() {  
    public void propertyChange(java.beans.PropertyChangeEvent evt) {  
        txtCedulaPropertyChange(evt);  
    }  
});  
  
txtCedula.addKeyListener(new java.awt.event.KeyAdapter() {  
    public void keyPressed(java.awt.event.KeyEvent evt) {  
        txtCedulaKeyPressed(evt);  
    }  
});  
  
txtNombres.setEnabled(false);  
jLabel2.setText("NOMBRES");  
txtSangre.setEnabled(false);  
jLabel3.setText("SANGRE");  
jLabel4.setText("FECHA NAC.");  
txtFechaNac.setEditable(true);  
txtFechaNac.setEnabled(false);  
btnBuscar.setText("BUSCAR");
```

```

btnBuscar.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnBuscarActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel3Layout.createSequentialGroup()

            .addGap(15, 15, 15)

            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,

false)

                .addGroup(jPanel3Layout.createSequentialGroup()

                    .addComponent(jLabel1)

                    .addGap(31, 31, 31)

                    .addComponent(txtCedula, javax.swing.GroupLayout.PREFERRED_SIZE, 206,

javax.swing.GroupLayout.PREFERRED_SIZE))

                .addGroup(jPanel3Layout.createSequentialGroup()

                    .addComponent(jLabel2)

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,

javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                    .addComponent(txtNombres, javax.swing.GroupLayout.PREFERRED_SIZE, 206,

javax.swing.GroupLayout.PREFERRED_SIZE))

                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,

jPanel3Layout.createSequentialGroup()

                    .addContainerGap()

                    .addComponent(txtFecha, javax.swing.GroupLayout.PREFERRED_SIZE, 206,

javax.swing.GroupLayout.PREFERRED_SIZE))

            )

        )

    )

);

```

```

        .addComponent(jLabel3)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(txtSangre, javax.swing.GroupLayout.PREFERRED_SIZE, 206,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(jPanel3Layout.createSequentialGroup())

        .addComponent(jLabel4)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(txtFechaNac, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(btnBuscar)

        .addContainerGap(425, Short.MAX_VALUE))

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup())

.addContainerGap())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(txtCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(btnBuscar, javax.swing.GroupLayout.PREFERRED_SIZE, 18,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```

addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel2)

    .addComponent(txtNombres,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel3)

    .addComponent(txtSangre,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel4)

    .addComponent(txtFechaNac,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

);

```

```

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtchedBorder(),
    "HISTORIAL MEDICO",                javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 12), new
java.awt.Color(0, 51, 153))); // NOI18N

```

```

jPanel4.setForeground(new java.awt.Color(0, 51, 153));

```

```

jLabel5.setText("FECHA \nINGRESO");

```

```

jLabel6.setText("DIAGNOSTICO");

```

```

jLabel7.setText("PESO");

```

```

jLabel8.setText("TALLA");

```

```
txtDiagnostico.setColumns(20);

txtDiagnostico.setRows(5);

txtDiagnostico.setEnabled(false);

jScrollPane1.setViewportView(txtDiagnostico);

txtPeso.setEnabled(false);

txtTalla.setEnabled(false);

txtFechaIngreso.setEnabled(false);

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel5)
                .addComponent(jLabel7)
                .addComponent(jLabel8))
            .addContainerGap(161, false))
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(txtPeso, javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(txtTalla,
                        javax.swing.GroupLayout.Alignment.LEADING,
                        javax.swing.GroupLayout.DEFAULT_SIZE, 95, Short.MAX_VALUE))
            ))
    )
);
```

```

        .addComponent(txtFechaIngreso, javax.swing.GroupLayout.PREFERRED_SIZE, 202,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(37, 37, 37)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel4Layout.createSequentialGroup()

                .addGap(152, 152, 152)

                .addComponent(jLabel6))

            .addGroup(jPanel4Layout.createSequentialGroup()

                .addGap(16, 16, 16)

                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 355,
javax.swing.GroupLayout.PREFERRED_SIZE)))

            .addGap(99, 99, 99))
    );

    jPanel4Layout.setVerticalGroup(

        jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel4Layout.createSequentialGroup()

                .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(txtFechaIngreso, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(jLabel6)))

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```



```

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel4Layout.createSequentialGroup())

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel7)

            .addComponent(txtPeso, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel8)

            .addComponent(txtTalla, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))

            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 107,
Short.MAX_VALUE))

            .addContainerGap()

        );

```

```

jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEtche
dBorder(), "HOSPITALIZACION", javax.swing.border.TitledBorder.CENTER,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 12), new
java.awt.Color(0, 51, 153))); // NOI18N

```

```

jPanel5.setForeground(new java.awt.Color(0, 51, 153));

```

```

jLabel9.setText("FECHA");

```

```

jLabel12.setText("CAMA");

```

```

jLabel13.setText("ESTADO PIEL");

```

```

jLabel14.setText("PROCEDENCIA");

```

```

txtProcedencia.setEnabled(false);

txtEstadoPiel.setEnabled(false);

cbxCama.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "1", "2", "3" }));

cbxCama.setEnabled(false);

txtFechaHospitalizacion.setEnabled(false);

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);

jPanel5.setLayout(jPanel5Layout);

jPanel5Layout.setHorizontalGroup(

    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup()

            .addGap(31, 31, 31)

            .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(txtFechaHospitalizacion, javax.swing.GroupLayout.PREFERRED_SIZE,
209, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(cbxCama, javax.swing.GroupLayout.PREFERRED_SIZE, 86,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(55, 55, 55)

            .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                .addComponent(jLabel14)

                .addComponent(jLabel13))

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(txtProcedencia,          javax.swing.GroupLayout.DEFAULT_SIZE,      364,
Short.MAX_VALUE)

            .addComponent(txtEstadoPiel,          javax.swing.GroupLayout.DEFAULT_SIZE,      364,
Short.MAX_VALUE))

        .addContainerGap())

    );

    jPanel5Layout.setVerticalGroup(

        jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel5Layout.createSequentialGroup())

        .addContainerGap()

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel14)

            .addComponent(txtProcedencia,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel9)

            .addComponent(txtFechaHospitalizacion,  javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(txtEstadoPiel,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jLabel13)

            .addComponent(cbxCama,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(jLabel12))

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

);

btnGuardar.setText("GUARDAR");

btnGuardar.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnGuardarActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(jPanel1Layout.createSequentialGroup()

                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(jPanel1Layout.createSequentialGroup()

                            .addGroup(jPanel1Layout.createSequentialGroup()

                                .addGroup(jPanel1Layout.createSequentialGroup()

                                    .addGap(41, 41, 41)

                                )

                            )

                        )

                    )

                .addGroup(jPanel1Layout.createSequentialGroup()

                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

                        .addComponent(jPanel5,

                            javax.swing.GroupLayout.Alignment.LEADING,

                            javax.swing.GroupLayout.DEFAULT_SIZE,

                            Short.MAX_VALUE)

                        .addComponent(jPanel3,

                            javax.swing.GroupLayout.Alignment.LEADING,

                            javax.swing.GroupLayout.DEFAULT_SIZE,

                            Short.MAX_VALUE)

                    )

                )

            )

        )

);

```

```

        .addComponent(jPanel4,                javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 806, Short.MAX_VALUE)))

        .addGroup(jPanel1Layout.createSequentialGroup())

        .addGap(392, 392, 392)

        .addComponent(btnGuardar)))

    .addContainerGap(28, Short.MAX_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addContainerGap()

.addComponent(jPanel3,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jPanel4,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jPanel5,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(btnGuardar)

.addContainerGap(17, Short.MAX_VALUE))

);

pnlContenedor.addTab("PACIENTES", jPanel1);

```

```
jPanel6.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineB  
order(new java.awt.Color(0, 0, 0), "CAMA 1",  
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,  
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 12), new  
java.awt.Color(0, 51, 153))); // NOI18N
```

```
jPanel6.setPreferredSize(new java.awt.Dimension(200, 162));
```

```
jLabel10.setFont(new java.awt.Font("Tahoma", 1, 12));
```

```
jLabel10.setForeground(new java.awt.Color(255, 0, 0));
```

```
jLabel10.setText("TEMPERATURA ACTUAL");
```

```
txtTemperaturaC1.setFont(new java.awt.Font("Tahoma", 1, 14));
```

```
jLabel11.setFont(new java.awt.Font("Tahoma", 1, 11));
```

```
jLabel11.setText("Datos Personales:");
```

```
lblNombresC1.setText("Nombres y Apellidos");
```

```
lblFechaC1.setText("Edad: 17 años");
```

```
jLabel15.setFont(new java.awt.Font("Tahoma", 1, 11));
```

```
jLabel15.setText("NOMBRES");
```

```
jLabel16.setFont(new java.awt.Font("Tahoma", 1, 11));
```

```
jLabel16.setText("F. NAC");
```

```
javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
```

```
jPanel6.setLayout(jPanel6Layout);
```

```
jPanel6Layout.setHorizontalGroup(
```

```
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(jPanel6Layout.createSequentialGroup()
```

```
            .addGap(17, 17, 17)
```

```
            .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addComponent(jLabel10)

        .addGroup(jPanel6Layout.createSequentialGroup())

    .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel15)

        .addComponent(jLabel16))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel11)

        .addComponent(txtTemperaturaC1, javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGroup(jPanel6Layout.createSequentialGroup())

        .addGap(10, 10, 10)

    .addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(lblFechaC1)

        .addComponent(lblNombresC1, javax.swing.GroupLayout.DEFAULT_SIZE, 134,
Short.MAX_VALUE))))))

        .addContainerGap())

    );

    jPanel6Layout.setVerticalGroup(

        jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel6Layout.createSequentialGroup()

            .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 15,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtTemperaturaC1,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel11)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel15)

        .addComponent(lbINombresC1))

        .addGap(11, 11, 11)

addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel16)

        .addComponent(lbIFechaC1))

        .addContainerGap()

);

```

```

jPanel7.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineB
order(new          java.awt.Color(0,          0,          0)),          "CAMA          2",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 12), new
java.awt.Color(0, 51, 153))); // NOI18N

```

```

jPanel7.setPreferredSize(new java.awt.Dimension(200, 162));

```

```

jLabel17.setFont(new java.awt.Font("Tahoma", 1, 12));

```

```

jLabel17.setForeground(new java.awt.Color(255, 0, 0));

```

```

jLabel17.setText("TEMPERATURA ACTUAL");

```

```

txtTemperaturaC2.setFont(new java.awt.Font("Tahoma", 1, 14));

```



```

.addContainerGap()

.addGroup(jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel19)

    .addComponent(jLabel20))

.addGap(18, 18, 18)

.addGroup(jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel7Layout.createSequentialGroup()

        .addComponent(lblFechaC2)

        .addContainerGap())

        .addComponent(lblNombresC2, javax.swing.GroupLayout.DEFAULT_SIZE, 147,
Short.MAX_VALUE)))

);

jPanel7Layout.setVerticalGroup(

    jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel7Layout.createSequentialGroup()

        .addComponent(jLabel17, javax.swing.GroupLayout.PREFERRED_SIZE, 15,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtTemperaturaC2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel18)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

            .addGroup(jPanel7Layout.createSequentialGroup()

```

```

        .addComponent(lblNombresC2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(lblFechaC2))

    .addGroup(jPanel7Layout.createSequentialGroup())

        .addComponent(jLabel19)

        .addGap(11, 11, 11)

        .addComponent(jLabel20))

    .addContainerGap()

);

```

```

jPanel9.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)), "CAMA 3", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT_POSITION, new java.awt.Font("Tahoma", 1, 12), new java.awt.Color(0, 51, 153))); // NOI18N

```

```

jPanel9.setPreferredSize(new java.awt.Dimension(200, 162));

jLabel25.setFont(new java.awt.Font("Tahoma", 1, 12));

jLabel25.setForeground(new java.awt.Color(255, 0, 0));

jLabel25.setText("TEMPERATURA ACTUAL");

txtTemperaturaC3.setFont(new java.awt.Font("Tahoma", 1, 14));

jLabel26.setFont(new java.awt.Font("Tahoma", 1, 11));

jLabel26.setText("Datos Personales:");

jLabel24.setFont(new java.awt.Font("Tahoma", 1, 11));

jLabel24.setText("F. NAC");

lblNombresC3.setText("Nombres y Apellidos");

jLabel27.setFont(new java.awt.Font("Tahoma", 1, 11));

```

```

jLabel27.setText("NOMBRES");

lblFechaC3.setText("Edad: 17 años");

javax.swing.GroupLayout jPanel9Layout = new javax.swing.GroupLayout(jPanel9);
jPanel9.setLayout(jPanel9Layout);
jPanel9Layout.setHorizontalGroup(
    jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel9Layout.createSequentialGroup()
            .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel9Layout.createSequentialGroup()
                    .addGroup(jPanel9Layout.createSequentialGroup()
                        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(jPanel9Layout.createSequentialGroup()
                                .addGroup(jPanel9Layout.createSequentialGroup()
                                    .addGap(36, 36, 36)
                                )
                            )
                        )
                    )
                .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel25)
                    .addGroup(jPanel9Layout.createSequentialGroup()
                        .addGap(10, 10, 10)
                    )
                )
            )
        )
        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel26)
            .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(txtTemperaturaC3, javax.swing.GroupLayout.PREFERRED_SIZE,
                    110, javax.swing.GroupLayout.PREFERRED_SIZE))))))
        .addGroup(jPanel9Layout.createSequentialGroup()
            .addContainerGap()
        )
    )
    .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

        .addComponent(jLabel27)

        .addComponent(jLabel24))

    .addGap(18, 18, 18)

.addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(lblFechaC3)

        .addComponent(lblNombresC3, javax.swing.GroupLayout.DEFAULT_SIZE, 98,
Short.MAX_VALUE))))

    .addContainerGap()

);

jPanel9Layout.setVerticalGroup(

jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel9Layout.createSequentialGroup())

    .addComponent(jLabel25, javax.swing.GroupLayout.PREFERRED_SIZE, 15,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(txtTemperaturaC3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jLabel26)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addGroup(jPanel9Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

            .addGroup(jPanel9Layout.createSequentialGroup())

                .addComponent(lblNombresC3)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

```

```
.addComponent(lblFechaC3))

.addGroup(jPanel9Layout.createSequentialGroup())

.addComponent(jLabel27)

.addGap(11, 11, 11)

.addComponent(jLabel24)))

.addContainerGap())

);

jLabel21.setText("VARIACION DE TEMPERATURA");

jLabel22.setText("CAMA 1");

jLabel23.setText("CAMA 2");

btnIniciar.setText("ACTUALIZAR");

btnIniciar.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnIniciarActionPerformed(evt);

    }

});

jScrollPane2.setViewportViewView(listCama2);

jScrollPane3.setViewportViewView(listCama1);

jScrollPane4.setViewportViewView(listCama3);

jLabel28.setText("CAMA 3");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel2Layout.createSequentialGroup())
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel2Layout.createSequentialGroup())
```

```
.addGap(225, 225, 225)
```

```
.addComponent(jLabel22)
```

```
.addGap(242, 242, 242)
```

```
.addComponent(jLabel23))
```

```
.addGroup(jPanel2Layout.createSequentialGroup())
```

```
.addContainerGap()
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addComponent(jLabel21)
```

```
.addGroup(jPanel2Layout.createSequentialGroup())
```

```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
.addGroup(jPanel2Layout.createSequentialGroup())
```

```
.addComponent(btnIniciar)
```

```
.addGap(40, 40, 40)
```

```
.addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE, 240,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 247,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 9,  
Short.MAX_VALUE)
```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addComponent(jScrollPane2, 0, 0, Short.MAX_VALUE)
    .addComponent(jPanel7, javax.swing.GroupLayout.PREFERRED_SIZE, 238,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addGap(27, 27, 27)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT_SIZE, 199,
Short.MAX_VALUE)
            .addComponent(jPanel9, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 199, Short.MAX_VALUE))
            .addGap(17, 17, 17))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel28)
            .addGap(102, 102, 102))))
);

jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()

```



```
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)

    .addGroup(jPanel2Layout.createSequentialGroup())

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

    .addGroup(jPanel2Layout.createSequentialGroup())

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jPanel6,    javax.swing.GroupLayout.PREFERRED_SIZE,    145,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(jPanel2Layout.createSequentialGroup())

        .addGap(47, 47, 47)

        .addComponent(btnIniciar)))

    .addGap(20, 20, 20)

    .addComponent(jLabel21)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))

.addGroup(jPanel2Layout.createSequentialGroup())

    .addComponent(jPanel9,    javax.swing.GroupLayout.PREFERRED_SIZE,    145,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(40, 40, 40)))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

    .addComponent(jScrollPane4)

    .addComponent(jScrollPane3,    javax.swing.GroupLayout.DEFAULT_SIZE,    234,
Short.MAX_VALUE)))

    .addGroup(jPanel2Layout.createSequentialGroup())
```

```

        .addComponent(jPanel7,          javax.swing.GroupLayout.PREFERRED_SIZE,      145,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(40, 40, 40)

        .addComponent(jScrollPane2)))

.addGap(30, 30, 30)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel22)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel23)

        .addComponent(jLabel28)))

.addContainerGap(56, Short.MAX_VALUE))
);

pnlContenedor.addTab("MONITOREO", jPanel2);

btnCargarC1.setText("CAMA 1");

btnCargarC1.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnCargarC1ActionPerformed(evt);

    }

});

btnCargarC2.setText("CAMA 2");

btnCargarC2.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        btnCargarC2ActionPerformed(evt);
    }
});

btnCargarC3.setText("CAMA 3");

btnCargarC3.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnCargarC3ActionPerformed(evt);

    }

});

javax.swing.GroupLayout jPanel8Layout = new javax.swing.GroupLayout(jPanel8);
jPanel8.setLayout(jPanel8Layout);
jPanel8Layout.setHorizontalGroup(

    jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel8Layout.createSequentialGroup()

            .addGap(111, 111, 111)

            .addComponent(btnCargarC1)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(btnCargarC2)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(btnCargarC3)

            .addGap(539, Short.MAX_VALUE))

        );

jPanel8Layout.setVerticalGroup(

```

```

jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel8Layout.createSequentialGroup()
        .addGap(85, 85, 85)

.addGroup(jPanel8Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnCargarC1)
    .addComponent(btnCargarC2)
    .addComponent(btnCargarC3))
    .addContainerGap(422, Short.MAX_VALUE))
);

pnlContenedor.addTab("ESTADISTICAS", jPanel8);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlContenedor, javax.swing.GroupLayout.DEFAULT_SIZE, 880,
Short.MAX_VALUE)
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pnlContenedor, javax.swing.GroupLayout.DEFAULT_SIZE, 558,
Short.MAX_VALUE)
);

pack();

} // </editor-fold>

```

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {  
  
    try {  
  
        this.setLocationRelativeTo(null);  
  
        this.setVisible(true);  
  
        lblNombresC1.setText(DatosCliente(1)[0]);  
  
        lblNombresC2.setText(DatosCliente(2)[0]);  
  
        lblNombresC3.setText(DatosCliente(3)[0]);  
  
        lblFechaC1.setText(DatosCliente(1)[1]);  
  
        lblFechaC2.setText(DatosCliente(2)[1]);  
  
        lblFechaC3.setText(DatosCliente(3)[1]);  
  
    } catch (Exception e) {  
  
    }  
  
}  
  
private void txtCedulaKeyPressed(java.awt.event.KeyEvent evt) {  
  
}  
  
private void txtCedulaPropertyChange(java.beans.PropertyChangeEvent evt) {  
  
}  
  
private void txtCedulaActionPerformed(java.awt.event.ActionEvent evt) {  
  
}  
  
private void txtCedulaFocusLost(java.awt.event.FocusEvent evt) {  
  
}  
  
public void EstadoInicial() {
```

```
txtDiagnostico.setEnabled(false);

txtEstadoPiel.setEnabled(false);

txtFechaIngreso.setEnabled(false);

txtFechaNac.setEnabled(false);

txtNombres.setEnabled(false);

txtPeso.setEnabled(false);

txtProcedencia.setEnabled(false);

txtSangre.setEnabled(false);

txtTalla.setEnabled(false);

cbxCama.setEnabled(false);

txtDiagnostico.setText("");

txtEstadoPiel.setText("");

txtCedula.setText("");

txtPeso.setText("");

txtProcedencia.setText("");

txtSangre.setText("");

txtTalla.setText("");

txtNombres.setText("");

}

public void Historial() {

    txtDiagnostico.setEnabled(true);

    txtEstadoPiel.setEnabled(true);

    txtFechaIngreso.setEnabled(true);

    cbxCama.setEnabled(true);
```

```
txtPeso.setEnabled(true);

txtProcedencia.setEnabled(true);

txtTalla.setEnabled(true);

txtFechaHospitalizacion.setEnabled(true);

}
```

```
public void DatosPersonales() {

    txtFechaNac.setEnabled(true);

    txtNombres.setEnabled(true);

    txtSangre.setEnabled(true);

    txtDiagnostico.setEnabled(true);

    txtEstadoPiel.setEnabled(true);

    txtFechaIngreso.setEnabled(true);

    cbxCama.setEnabled(true);

    txtPeso.setEnabled(true);

    txtProcedencia.setEnabled(true);

    txtTalla.setEnabled(true);

    txtFechaHospitalizacion.setEnabled(true);

}
```

```
int ban, banEstado;
```

```
private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {

    try {

        cedula = txtCedula.getText();

        String consulta = "select * from tab_paciente where cedula='" + cedula + "'";
```

```

ResultSet tipoUsuario = (ResultSet) con.consulta(consulta);

int nomb = tipoUsuario.findColumn("nombres");

int ts = tipoUsuario.findColumn("tipo_sangre");

int cod = tipoUsuario.findColumn("cod_paciente");

boolean lleno = tipoUsuario.next();

if (cedula.length() == 10) {

    if (!lleno) {

        JOptionPane.showMessageDialog(this, "EL PACIENTE NO EXISTE!!!");

        DatosPersonales();

        ban = 1;

    } else {

        txtNombres.setText(tipoUsuario.getString(nomb));

        txtSangre.setText(tipoUsuario.getString(ts));

        consulta = "select estado from tab_historial where cod_paciente=" +
tipoUsuario.getString(cod) + """;

        ResultSet estadoHis = (ResultSet) con.consulta(consulta);

        boolean verificarE = estadoHis.next();

        if (!verificarE) {

            banEstado = 1;//Cuando no existe nada de datos se puede ingresar el hisotrial

        } else {

            int est = estadoHis.findColumn("estado");

            if (Integer.parseInt(estadoHis.getString(est)) == 1) {

                banEstado = 1;

            } else {

```



```

        banEstado = 2;
    }

}

ban = 2;

Historial();

}

} else {

    Estadoinicial();

    JOptionPane.showMessageDialog(this, "VERIFIQUE SU NUMERO DE CEDULA!!!");

    ban = 3;

}

} catch (Exception e) {

}

}

private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {

    if (txtCedula.getText().equals("") || txtDiagnostico.getText().equals("") ||
txtEstadoPiel.getText().equals("") || txtNombres.getText().equals("") || txtPeso.getText().equals("") ||
txtProcedencia.getText().equals("") || txtSangre.getText().equals("") || txtTalla.getText().equals("")) {

        ban = 3;

    }

    if (ban == 3) {

        JOptionPane.showMessageDialog(this, "VERIFIQUE SUS DATOS!!!");

    } else {

```

```

if (ban == 2) {

    if (banEstado == 1) {

        JOptionPane.showMessageDialog(this, "EL PACIENTE NO SE ENCUENTRA
HOSPITALIZADO");

    } else {

        JOptionPane.showMessageDialog(this, "EL PACIENTE SE ENCUENTRA
HOSPITALIZADO");

    }

} else {

    java.util.Date fnac = txtFechaNac.getDate();

    java.sql.Date ffnn = new java.sql.Date(fnac.getTime());

    String insertarPaciente = "INSERT INTO bddmedico.tab_paciente VALUES (default, " +
txtCedula.getText() + ", " + txtNombres.getText() + ", " + ffnn + ", " + txtSangre.getText() + ")";

    if (con.ejecutar(insertarPaciente).equals("ok")) {

        java.util.Date ffii = txtFechaIngreso.getDate();

        java.sql.Date fi = new java.sql.Date(ffii.getTime());

        String insertarHistorial = "INSERT INTO bddmedico.tab_historial VALUES (default, " + fi +
", " + txtDiagnostico.getText() + ", " + txtPeso.getText() + ", " + txtTalla.getText() + ",0," +
CodigoPaciente() + ")";

        if (con.ejecutar(insertarHistorial).equals("ok")) {

            java.util.Date ffhh = txtFechaHospitalizacion.getDate();

            java.sql.Date fh = new java.sql.Date(ffhh.getTime());

            String actualizarEstado = "update bddmedico.tab_ingreso_terapia set estado_cama=0
where cama= " + cbxCama.getSelectedItem() + """;

            if (con.ejecutar(actualizarEstado).equals("ok")) {

```

```
String insertarTerapia = "INSERT INTO bddmedico.tab_ingreso_terapia VALUES  
(default, "" +CodigoHistorial() + "", "" + fh + "", "" + cbxCama.getSelectedltem() + "", "" +  
txtProcedencia.getText() + "", "" + txtEstadoPiel.getText() + "",1)";
```

```
if (con.ejecutar(insertarTerapia).equals("ok")) {
```

```
    JOptionPane.showMessageDialog(this, "PACIENTE INGRESADO CON EXITO");
```

```
    EstadoInicial();
```

```
}
```

```
int terminar = 0;
```

```
private void btnIniciarActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    try {
```

```
        //CAMA 3
```

```
        int acum = 0;
```

```
        DefaultListModel modelo = new DefaultListModel();
```

```
        ResultSet res3 = DatosAnalisis(3);
```

```
        boolean verificarE = res3.next();
```

```
        if (!verificarE) {
```

```
            //Cuando no existe nada de datos se puede ingresar el hisotrial
```

```
        } else {
```

```
            txtTemperaturaC3.setText(res3.getString("temperatura"));
```

```
            while (verificarE) {
```

```
                int nom = res3.findColumn("temperatura");
```

```
                modelo.addElement(res3.getString("temperatura"));
```

```
                verificarE = res3.next();
```

```
                if (acum == 10) {
```

```
        acum = 0;

        break;

    }

    acum++;

}

listCama3.setModel(modelo);

}

DefaultListModel modeloC1 = new DefaultListModel();

ResultSet res1 = DatosAnalysis(1);

boolean verificarE1 = res1.next();

if (!verificarE1) {

} else {

    txtTemperaturaC1.setText(res1.getString("temperatura"));

    while (verificarE1) {

        int nom = res1.findColumn("temperatura");

        modeloC1.addElement(res1.getString("temperatura"));

        verificarE1 = res1.next();

        if (acum == 10) {

            acum = 0;

            break;

        }

        acum++;

    }

}
```

```

        listCama1.setModel(modeloC1);
    }

    //CAMA 2

    DefaultListModel modeloC2 = new DefaultListModel();

    ResultSet res2 = DatosAnalysis(2);

    boolean verificarE2 = res2.next();

    if (!verificarE2) {

        //Cuando no existe nada de datos se puede ingresar el hisotrial

    } else {

        txtTemperaturaC2.setText(res2.getString("temperatura"));

        while (verificarE2) {

            int nom = res2.findColumn("temperatura");

            modeloC2.addElement(res2.getString("temperatura"));

            verificarE2 = res2.next();

            if (acum == 10) {

                acum = 0;

                break;

            }

            acum++;

        }

        listCama2.setModel(modeloC2);

    }

    lblNombresC1.setText(DatosCliente(1)[0]);

    lblNombresC2.setText(DatosCliente(2)[0]);

```

```
        lblNombresC3.setText(DatosCliente(3)[0]);

        lblFechaC1.setText(DatosCliente(1)[1]);

        lblFechaC2.setText(DatosCliente(2)[1]);

        lblFechaC3.setText(DatosCliente(3)[1]);

    } catch (Exception w) {

    }

}

private void btnCargarC1ActionPerformed(java.awt.event.ActionEvent evt) {

    Graficos demo1 = new Graficos("CUADRO DE ESTADISTICAS CAMA No 1", 1);

    demo1.pack();

    RefineryUtilities.centerFrameOnScreen(demo1);

    iniciar();

    demo1.setVisible(true);

}

private void btnCargarC2ActionPerformed(java.awt.event.ActionEvent evt) {

    Graficos demo1 = new Graficos("CUADRO DE ESTADISTICAS CAMA No 2", 2);

    demo1.pack();

    RefineryUtilities.centerFrameOnScreen(demo1);

    iniciar();

    demo1.setVisible(true);

}
```

```

private void btnCargarC3ActionPerformed(java.awt.event.ActionEvent evt) {

    Graficos demo1 = new Graficos("CUADRO DE ESTADISTICAS CAMA No 3", 3);

    demo1.pack();

    RefineryUtilities.centerFrameOnScreen(demo1);

    iniciar();

    demo1.setVisible(true);

}

public void iniciar() {

    Timer t = new Timer(10000, new ActionListener() {

        public void actionPerformed(ActionEvent e) {

        }

    });

    t.start();

}

public int CodigoPaciente() {

    int codigo = 0;

    try {

        String consulta = "select cod_paciente from tab_paciente where cedula=" + txtCedula.getText() + "";

        ResultSet res = (ResultSet) con.consulta(consulta);
    }
}

```

```

boolean verificarE = res.next();

if (!verificarE) {

    codigo = -1;//Cuando no existe nada de datos se puede ingresar el hisotrial

} else {

    int co = res.findColumn("cod_paciente");

    codigo = Integer.parseInt(res.getString(co));

}

} catch (Exception e) {

}

return codigo;

}

```

```

public intCodigoHistorial() {

    int codigo = 0;

    try {

        String consulta = "SELECT max(cod_historial) as Ultimo FROM tab_historial";

        ResultSet res = (ResultSet) con.consulta(consulta);

        boolean verificarE = res.next();

        if (!verificarE) {

            codigo = -1;//Cuando no existe nada de datos se puede ingresar el hisotrial

        } else {

            int co = res.findColumn("ultimo");

```



```

        codigo = Integer.parseInt(res.getString(co));
    }
} catch (Exception e) {
}
return codigo;
}

public String[] DatosCliente(int cama) {
    //cama=3;

    String[] datos = new String[3];

    try {
        String consulta = "SELECT nombres,fecha_nac,tipo_sangre FROM tab_ingreso_terapia i,
tab_historial h,tab_paciente p WHERE estado_cama =1 AND cama =" + cama + " AND h.cod_historial =
i.cod_historial AND h.estado =0 AND h.cod_paciente=p.cod_paciente";

        ResultSet res = (ResultSet) con.consulta(consulta);

        boolean verificarE = res.next();

        if (!verificarE) {

            //Cuando no existe nada de datos se puede ingresar el hisotrial

        } else {

            int nom = res.findColumn("nombres");

            int fn = res.findColumn("fecha_nac");

            int san = res.findColumn("tipo_sangre");

            datos[0] = res.getString(nom);

            datos[1] = res.getString(fn);

            datos[2] = res.getString(san);

        }
    }
}

```

```

    } catch (Exception e) {

    }

    return datos;

}

public ResultSet DatosAnalisis(int cama) {

    String consulta = "SELECT temperatura,fecha_analisis FROM tab_ingreso_terapia i, tab_analisis a
WHERE estado_cama =1 AND cama =" + cama + " and i.cod_ingreso=a.cod_ingreso ORDER BY
fecha_analisis DESC";

    ResultSet res = (ResultSet) con.consulta(consulta);

    return res;

}

public int CodigoIngreso(int cama) {

    int codigo = 0;

    try {

String consulta = "SELECT cod_ingreso FROM tab_ingreso_terapia i, tab_historial h WHERE
estado_cama =1 AND cama =" + cama + " AND h.cod_historial = i.cod_historial AND h.estado =0";

        ResultSet res = (ResultSet) con.consulta(consulta);

        boolean verificarE = res.next();

        if (!verificarE) {

            codigo = -1;//Cuando no existe nada de datos se puede ingresar el hisotrial

        } else {

            int co = res.findColumn("cod_ingreso");

            codigo = Integer.parseInt(res.getString(co));

        }

    }

```

```
    } catch (Exception e) {  
  
    }  
  
    return codigo;  
}  
  
public static void main(String args[]) {  
  
    java.awt.EventQueue.invokeLater(new Runnable() {  
  
        public void run() {  
  
            new frmPrincipal().setVisible(true);  
  
        }  
  
    });  
}  
  
private javax.swing.JButton btnBuscar;  
  
private javax.swing.JButton btnCargarC1;  
  
private javax.swing.JButton btnCargarC2;  
  
private javax.swing.JButton btnCargarC3;  
  
private javax.swing.JButton btnGuardar;  
  
private javax.swing.JButton btnIniciar;  
  
private javax.swing.JComboBox cbxCama;  
  
private javax.swing.JLabel jLabel1;  
  
private javax.swing.JLabel jLabel10;  
  
private javax.swing.JLabel jLabel11;  
  
private javax.swing.JLabel jLabel12;
```

```
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
```

```
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JPanel jPanel4;  
private javax.swing.JPanel jPanel5;  
private javax.swing.JPanel jPanel6;  
private javax.swing.JPanel jPanel7;  
private javax.swing.JPanel jPanel8;  
private javax.swing.JPanel jPanel9;  
private javax.swing.JPopupMenu jPopupMenu1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JScrollPane jScrollPane2;  
private javax.swing.JScrollPane jScrollPane3;  
private javax.swing.JScrollPane jScrollPane4;  
private javax.swing.JLabel lblFechaC1;  
private javax.swing.JLabel lblFechaC2;  
private javax.swing.JLabel lblFechaC3;  
private javax.swing.JLabel lblNombresC1;  
private javax.swing.JLabel lblNombresC2;  
private javax.swing.JLabel lblNombresC3;  
private javax.swing.JList listCama1;  
private javax.swing.JList listCama2;  
private javax.swing.JList listCama3;  
private javax.swing.JTabbedPane pnlContenedor;
```

```
private javax.swing.JTextField txtCedula;

private javax.swing.JTextArea txtDiagnostico;

private javax.swing.JTextField txtEstadoPiel;

private org.freixas.jcalendar.JCalendarCombo txtFechaHospitalizacion;

private org.freixas.jcalendar.JCalendarCombo txtFechaIngreso;

private org.freixas.jcalendar.JCalendarCombo txtFechaNac;

private javax.swing.JTextField txtNombres;

private javax.swing.JTextField txtPeso;

private javax.swing.JTextField txtProcedencia;

private javax.swing.JTextField txtSangre;

private javax.swing.JTextField txtTalla;

private javax.swing.JTextField txtTemperaturaC1;

private javax.swing.JTextField txtTemperaturaC2;

private javax.swing.JTextField txtTemperaturaC3;

public void actionPerformed(ActionEvent e) {

    throw new UnsupportedOperationException("Not supported yet.");

}

}
```

App Recibe Datos.

```
public class Conexion {

// Conexion con = new Conexion();

    Connection Conec;

    Statement St;

    ResultSet reg;

    public Conexion() {

        try {

            Class.forName("com.mysql.jdbc.Driver");

            String url = "jdbc:mysql://localhost:3306/bddmedico";

            Conec = (Connection) DriverManager.getConnection(url, "root", "root");

                } catch (Exception ex) {

                    System.out.println("Error: " + ex.getMessage());

                }

        }

    }

    public String ejecutar(String sql) {

        String error = "";

        try {

            St = Conec.createStatement();

            St.execute(sql);

            error="ok";

        } catch (Exception ex) {

            error = ex.getMessage();

        }

    }

}
```

```

    }

    return (error);
}

public ResultSet consulta(String sql) {

    String error = "";

    try {

        St = (Statement) Conec.createStatement();

        reg = (ResultSet) St.executeQuery(sql);

    } catch (Exception ex) {

        error = ex.getMessage();

    }

    return (reg);

}

public int getCodigoIngreso(int cama) {

    int codigo = 0;

    try {

        String consulta = "SELECT cod_ingreso FROM tab_ingreso_terapia i, tab_historial h WHERE
estado_cama =1 AND cama ="+cama+" AND h.cod_historial = i.cod_historial AND i.estado_cama =1";

        ResultSet res = (ResultSet) consulta(consulta);

        boolean verificarE = res.next();

        if (!verificarE) {

            codigo = -1;//Cuando no existe nada de datos se puede ingresar el hisotrial

        } else {

            int co = res.findColumn("cod_ingreso");

```



```
        codigo = Integer.parseInt(res.getString(co));
    }

    } catch (Exception e) {
    }

    return codigo;
}
}
```

App Datos Función Principal.

```
package apprecibirdatos;

import com.mysql.jdbc.ResultSet;

import javax.swing.JOptionPane;

import giovynet.nativelink.SerialPort;

import giovynet.serial.Baud;

import giovynet.serial.Com;

import giovynet.serial.Parameters;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import sun.io.Converters;

import java.io.*;

import sun.audio.*;
```

```
public class Main {  
  
    public static void main(String[] args) throws Exception{  
  
        Conexion con = new Conexion();  
  
        int codigoIngreso = 0;  
  
        try {  
  
            String recibido = "";  
  
            String caracter = "";  
  
            String data = "";  
  
            int abrir = 1;  
  
            SerialPort serialPort = new SerialPort();  
  
            List<String> portsFree;  
  
            portsFree = serialPort.getFreeSerialPort();  
  
            if (portsFree != null && portsFree.size() > 0) {  
  
                for (String free : portsFree) {  
  
                    System.out.println("Puertos Libres: " + free);  
  
                }  
  
                for (int i = 0; i < portsFree.size(); i++) {  
  
                    if (!portsFree.get(i).isEmpty()) {  
  
                        abrir = i;  
  
                    }  
  
                }  
  
            }  
  
            Parameters parameters = new Parameters();  
  
            parameters.setPort(portsFree.get(2));  
  
            parameters.setBaudRate(Baud._9600);  
  
        }  
    }  
}
```

```
Com com = new Com(parameters);

System.out.println("PUERTO ABIERTO: " + portsFree.get(2));

char seperador = '\n';

do {

    String sonido="C:/Blip.wav";

InputStream in=new FileInputStream(sonido);

AudioStream audio=new AudioStream(in);

    recibido = "";

    caracter = "x";

    int i = 0;

do {

    caracter = com.readPortC(parameters.getPort());

    if (caracter.equals("")) {

        caracter = "x";

    } else {

        if (caracter.toCharArray()[0] != '\n') {

            recibido += caracter;

        }

    }

    i++;

} while (caracter.charAt(0) != seperador);

if (recibido.length() == 10) {
```

```

System.out.println(recibido);

String[] t1 = recibido.split("T");//para separar la cadena recibida

String[] t2 = t1[1].split("F");//sacar la temperatura

String[] sacarCama = t1[0].split("A");

int cama = Integer.parseInt(sacarCama[1]);

System.out.println(t2[0] + " cama" + cama);

codigoIngreso = con.getCodigoIngreso(cama);

Date fecha = new Date();

DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

java.sql.Date fa = new java.sql.Date(fecha.getTime());

double limInferior = 35.50;

double limSuperior = 37.50;

if (Double.valueOf(t2[0]) <= limInferior) {

    AudioPlayer.player.start(audio);

}

if (Double.valueOf(t2[0]) >= limSuperior) {

    AudioPlayer.player.start(audio);

}

String insertarAnalisis = "INSERT INTO bddmedico.tab_analisis VALUES (" +
codigoIngreso + ", " + t2[0] + ", " + dateFormat.format(fa) + ")";

if (con.ejecutar(insertarAnalisis).equals("ok")) {

    System.out.println("ingreso ok");
}

```

```

    }

    } else {

        if(recibido.length() == 4){

            String[] transmisor = recibido.split("B");//para separar la cadena recibida

            // System.out.println(recibido);

            JOptionPane.showMessageDialog(null, "CARGAR TRANSMISOR "+transmisor[1]);

        }

    }

    } while (true);

    System.out.println("<Datos Enviados>");

    for (int i = 0; i < data.toCharArray().length; i++) {

        Thread.sleep(400);

        com.sendSingleData(data.toCharArray()[i]);

        System.out.print(data.toCharArray()[i]);

    }

    } else {

        System.out.println("No hay puertos libres!!!");

    }

} catch (Exception e) {

    e.printStackTrace();

}

}

}

```

Anexo H

FOTOGRAFÍAS DEL DISPOSITIVO.



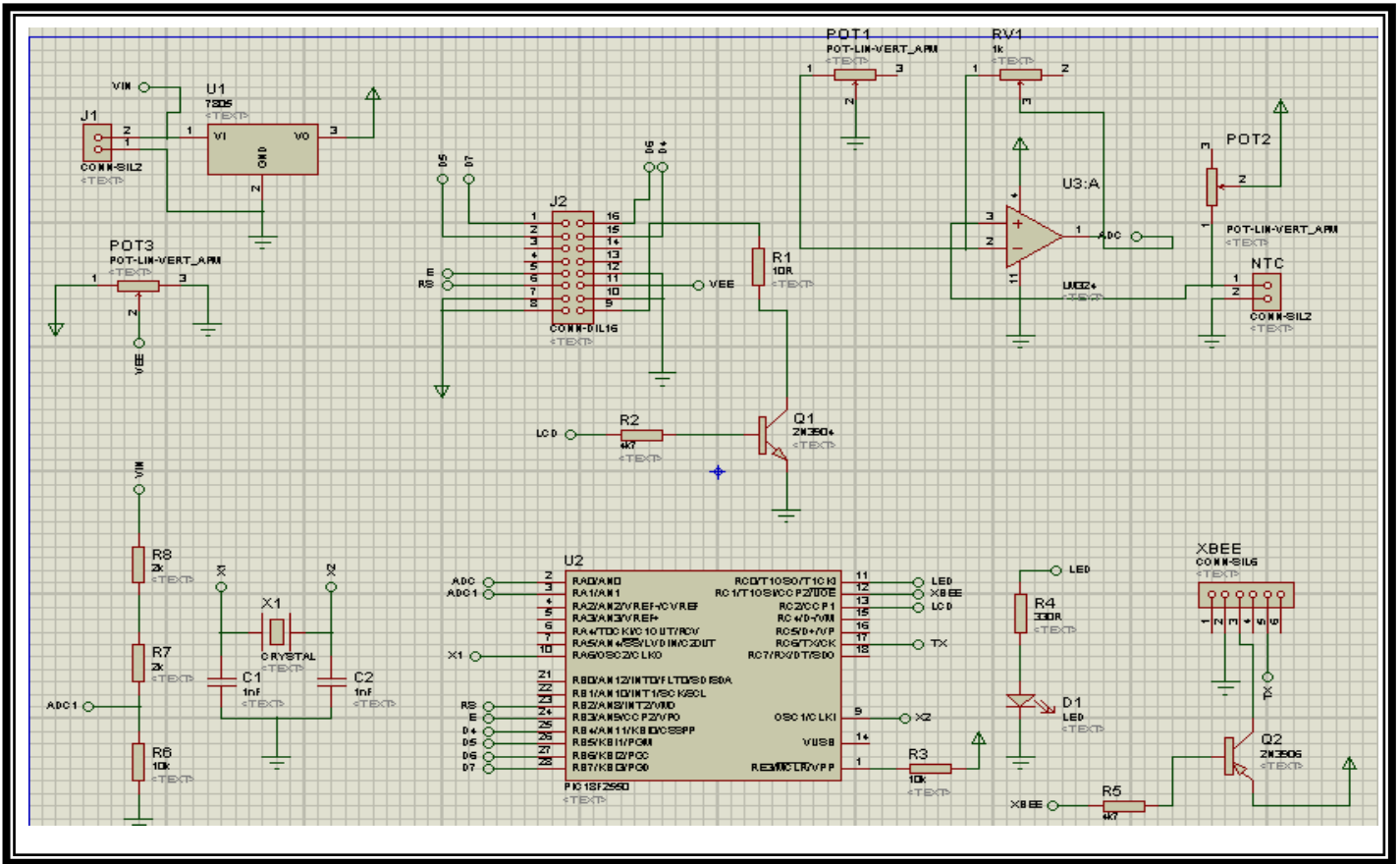
Transmisor portatil.



Receptor de datos de temperatura corporal.

Anexo I

ESQUEMA GENERAL DE LA BAQUELITA PROTEUS



Conexiones y puros de simulación del transmisor.