DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA

TEMA:

DISEÑO E IMPLEMENTACIÓN DE UNA RED BAN, PARA LA OBTENCIÓN DE DATOS DE SIGNOS VITALES UTILIZANDO EL PROTOCOLO 802.15.4

AUTORES:

RIVERA FONSECA, DIANA ELIZABETH

VINUEZA LOPEZ, WILMA CRISTINA

DIRECTOR: ING. ESPINOSA NIKOLAI

CODIRECTOR: ING. VEGA, CHRISTIAN

SANGOLQUÍ, 2015

CERTIFICACIÓN

Certificamos que el proyecto titulado "DISEÑO E IMPLEMENTACIÓN DE UNA RED BAN, PARA LA OBTENCIÓN DE DATOS DE SIGNOS VITALES UTILIZANDO EL PROTOCOLO 802.15.4", ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Universidad de las Fuerzas Armadas.

Debido a que se trata de un trabajo de investigación se recomienda su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a la Srta. Diana Elizabeth Rivera Fonseca con CI: 1720916665 y a la Srta. Wilma Cristina Vinueza López con CI: 171536373-3 que lo entreguen al Doctor Nikolai Espinosa, en su calidad de Director de la Carrera.

Atentamente,

Ing. Nikolai Espinosa

Ing. Christian Vega

DECLARACIÓN DE RESPONSABILIDAD

DIANA ELIZABETH RIVERA FONSECA
WILMA CRISTINA VINUEZA LÓPEZ

DECLARAMOS QUE:

El proyecto de grado denominado "DISEÑO E IMPLEMENTACIÓN DE UNA RED BAN, PARA LA OBTENCIÓN DE DATOS DE SIGNOS VITALES UTILIZANDO EL PROTOCOLO 802.15.4", ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

En virtud a esta declaración, nos responsabilizamos del contenido, veracidad y alcance del proyecto de grado en mención.

Sangolquí, 18 mayo del 2015

Diana Elizabeth Rivera Fonseca

Wilma Cristina Vinueza López

AUTORIZACIÓN

DIANA ELIZABETH RIVERA FONSECA
WILMA CRISTINA VINUEZA LÓPEZ

Autorizamos a la Universidad de las Fuerzas Armadas "ESPE" la publicación, en la biblioteca virtual de la institución el trabajo "DISEÑO E IMPLEMENTACIÓN DE UNA RED BAN, PARA LA OBTENCIÓN DE DATOS DE SIGNOS VITALES UTILIZANDO EL PROTOCOLO 802.15.4", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 18 mayo del 2015

Diana Elizabeth Rivera Fonseca

Wilma Cristina Vinueza López

DEDICATORIAS

A todas las personas que directa o indirectamente me han apoyado durante el transcurso de mi vida universitaria, a aquellos que siempre me motivaron a concluir la carrera con sus palabras de aliento y su apoyo incondicional.
Dedico la culminación de este Proyecto principalmente a mi madre, Gladys Fonseca, mujer valerosa a la cual amo ya que con su ejemplo me enseño que siempre hay que terminar lo que uno inicia, pese a los obstáculos y adversidades que se presenten día a día y finalmente pero no menos importante a mi amiga Cris con la cual tendré el gusto de disfrutar el cumplimiento de esta meta.

Rivera Fonseca Diana Elizabeth.

Dedico este trabajo a los seres más preciados, mi Madre Vilma, que siempre ha estado junto a mí apoyándome y brindándome su amor incondicional, a mi Hermana Silvana que pese a sus locuras me ha sacado una sonrisa en momentos difíciles, a mi Padre Fernando que ha sido mi amigo, soporte, motor y ejemplo en este largo camino, desde el cielo cuidas y guías mis pasos, para lograr este y más anhelos que compartíamos, sé que hubiese sido tan especial para ti, como lo es para mí. A mi Novio Marcelo por empujarme a salir adelante y ser una persona incondicional en todo momento. A mi amiga Ely, por formar un gran equipo de trabajo, que nos llevó a culminar esta meta.

Vinueza López Wilma Cristina

AGRADECIMIENTOS

Agradezco a todos los miembros de mi familia por haber sido el pilar fundamental para el cumplimiento de esta meta. A mis padres Gladys y Enrique que han sabido guiarme de forma correcta y apoyarme en todo aspecto, A mis hermanos Eduardo, Bryan y Kevin quienes siempre han estado para mí, en las buenas y en las malas, siendo más que hermanos amigos incondicionales, y finalmente a mis hijas Kamyla y Alejandra mi principal motivación para culminar la carrera pues el anhelo de un mejor futuro para ellas me ha permitido llegar hasta este punto. A los docentes de la ESPE, los cuales han sabido compartir de forma desinteresada su conocimiento con el fin de generar buenos profesionales, por lo mencionado y mucho más, a Todos. Gracias de corazón.

Rivera Fonseca Diana Elizabeth.

Primero agradecer a Dios que me ha permitido culminar con éxito esta etapa de mi vida, por darme fuerzas para seguir adelante. Agradezco a mi familia, por ser un pilar muy importante en mi vida. A mi Madre por su paciencia, fortaleza, compresión y amor en todo momento, por alegrarse de mis triunfos y brindarme su mano en mis caídas. A mi Padre que me ofreció sus consejos, enseñanzas, y cariño, quien me hizo comprender que para lograr una meta no importa lo duro del camino, ni los obstáculos que se presenten, sino el nunca rendirse. A mi hermana, que ha sabido soportar mi mal humor, por ser mi cómplice, confidente y amiga. A Marcelo por su apoyo y amor incondicional, por compartir conmigo momentos de alegría y tristeza, demostrándome que podré contar con él en cada momento.

Un agradecimiento a nuestro director, Nikolai Espinosa y codirector Christian Vega, quienes con sus sugerencias y enseñanzas, han colaborado con el avance y culminación de este proyecto. Por último un agradecimiento a todos mis amigos, por su amistad, ayuda, apoyo, por compartir momentos de apremio y risas. A mi amiga Ely por apoyarme y trabajar arduamente para lograr este sueño en común.

Vinueza López Wilma Cristina

INDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El presente proyecto busca diseñar e implementar un prototipo de una red de sensores para el monitoreo de cinco signos vitales, como sensor de pulso y oxigenación en la sangre, temperatura corporal, posición corporal, flujo de aire, respuesta galvánica de la piel (sudoración), y la futura implementación de un glucómetro, utilizando un Kit de Sensores, que incluye un escudo E-Health, conectado a una plataforma Arduino y un módulo de recepción y transmisión inalámbrica Zigbee que se basa en el estándar IEEE 802.15.4. A su vez las muestras de los signos vitales obtenidos serán almacenadas en una base de datos generada en SQL, para su posterior visualización en una interfaz web, creada en Visual Studio. Dentro del desarrollo se compara las opciones entre Arduino y Raspberry, que permite conocer la mejor opción, para la configuración de los sensores mencionados. Se detallan las conexiones de los sensores hacia el escudo E-Health, códigos de configuración tanto para los sensores, como para la base de datos y la página web, así como la arquitectura de la red implementada. Para verificar el funcionamiento y fidelidad en el envío de datos de la aplicación desplegada, se realizaron pruebas de radio entre varios metros, con su respectivo análisis. El propósito de este trabajo es el poder realizar futuras mejoras en el servicio médico, y ajustarlo a las necesidades de cada paciente y doctor según corresponda, ya que los elementos empleados tanto en software como hardware son de desarrollo libre.

PALABRAS CLAVES:

- **E-HEALTH**
- **ARDUINO**
- **ZIGBEE**
- **RED DE SENSORES**
- **BASE DE DATOS**

ABSTRACT

The present project seeks to design and implement a prototype of a sensor network to monitor five vital signs, like pulse and oxygen in blood sensor, body temperature, body position, air flow, galvanic response of the skin (sweat), and the future implementation of a glucometer, that will measure the amount of glucose in blood, by using a Sensor Kit, that includes an E-heatlh shield, connected to an Arduino platform and a wireless reception, transmission module Zigbee which is based on the standard IEEE 802.15.4. At the time the samples obtained from the vital signs will be storage on a data base generated on SQL, for a further visualization on a web interface, created on Visual Studio. Inside the development is a comparison between Arduino and Raspberry that allows knowing the best option for the configuration of the mentioned sensors. The connections of sensors to the E-Health shield, configuration codes, for sensors, data base and web page, and the architecture of the net is detailed. To verify the functionality and fidelity of the sent data, radio tests were made between many meters, and their analysis. The purpose of this work is to improve medical service, and adjust it according the needs of patients or doctors, because the employed elements such as hardware and software are for free development.

KEY WORDS:

- **E-HEALTH**
- **ARDUINO**
- **ZIGBEE**
- **SENSOR NETWORK**
- **DATA BASE**

CAPITULO I: INTRODUCCIÓN

1.1.Antecedentes

En los últimos años, las investigaciones y desarrollos de redes de sensores inalámbricos han ido aumentando tanto en el nivel comercial como en el académico para aplicaciones médicas, específicamente para la monitorización de pacientes. En la actualidad existen varios tipos de sensores inalámbricos que ayudan a monitorizar de forma remota y discreta los parámetros vitales del paciente, tales como la frecuencia cardíaca, frecuencia respiratoria, presión arterial; permitiendo la movilidad de los mismos (Martínez, D., Simo, J., Blandes, F., & Crespo, a. 2009).

En el ámbito de la salud, las redes de sensores pueden llevar a cabo diferentes acciones como son: monitoreo de pacientes, diagnóstico de enfermedades, en la administración de los medicamentos o en el monitoreo del movimiento de los pacientes en los hospitales.

El uso de estas redes de sensores inteligentes en el ámbito de la monitorización de la salud reduce considerablemente el número de hospitalizaciones y visitas médicas y con ello los costes anuales del cuidado de la salud en todo el mundo.

Actualmente se dispone de un gran número de estándares inalámbricos (BLUETOOTH, WIFI,WIMAX, LMDS, etc.) orientados hacia aplicaciones con altos requerimientos de ancho de banda (redes domésticas y de oficina, videoconferencia, VOIP, etc.), pero falta algún estándar inalámbrico específico para redes de sensores en aplicaciones de salud, industriales y domésticas (Flores, E., 2012).

El inconveniente de utilizar cualquiera de los estándares inalámbricos antes mencionados radica en el gran consumo de energía y el ancho de banda que utilizan frente a la baja tasa de bits enviados por cualquier aplicación sensora o de control y sus bajos requerimientos de energía.

En un principio, cada fabricante de nodos sensores ha optado por utilizar soluciones propietarias, lo que trajo problemas de interoperabilidad entre los diversos fabricantes.

La industria notó que hacía falta un nuevo estándar que aúne autonomía, envío de

datos de baja capacidad y un bajo costo, por esta razón nació el estándar IEEE 802.15.4 comercialmente llamado ZigBee. Se ha convenido llamar a esta clase de redes LR-WPAN (Low Rate Wireless Personal Area Network), dado sus bajas tasas de transmisión y su corto alcance (Barriga, W., 2006).

Para supervisar a pacientes con enfermedades crónicas, tales como diabetes, y asma, es importante monitorear permanentemente los signos vitales. La cardiología es el área más prominente de aplicación del monitoreo de los pacientes a través de Electrocardiogramas (ECGs), que indican el estado general del corazón del paciente.

Las redes de Área Corporal (Body Area Networks: BAN), son la estructura básica para el cuidado de la salud utilizando medios electrónicos.

En la mayoría de casos el monitoreo médico requiere de más de un sensor que es conectado al cuerpo humano. Por ejemplo los pacientes con problemas cardíacos deben tener un monitoreo de la presión sanguínea, saturación de oxígeno de la sangre, pulsación cardíaca, temperatura, peso, etc. (Barriga, W., 2006).

Actualmente la tecnología de monitoreo requiere que el paciente utilice una serie de cables para adquirir y procesar las señales vitales. Usando la tecnología de monitoreo inalámbrico, se pueden prevenir y tratar un gran número de enfermedades, pues pueden ser controladas eficazmente. La tecnología más apropiada para esta aplicación es la LR-WPAN (*Low Rate Wireless Personal Area Network*).

La mayoría de los equipos médicos modernos ya comienzan a incluir los medios para comunicarse de forma nativa con redes de este tipo. Esta es una característica a tener en cuenta para ser utilizada

1.2.Justificación e importancia

Desde finales de la década de los 90 ha surgido un gran interés por el uso de redes inalámbricas para la interconexión de los nodos, permitiendo concebir nuevas

aplicaciones que anteriormente eran poco viables debido a limitaciones en la portabilidad de la aplicación, la movilidad de los nodos y la autoconfiguración del sistema.

Inicialmente se desarrollaron aplicaciones de monitorización, dando origen al término redes de sensores inalámbricas (Wireless Sensors Networks - WSN). Posteriormente los nodos fueron dotados de interfaces de actuación, dando origen a las redes inalámbricas de sensores y actuadores (Wireless Sensors and Actuators Networks - WSAN) (Martínez, D. et al., 2009).

En la actualidad hay una fuerte investigación en el campo de las redes de sensores inalámbricas, debido a las grandes ventajas que supone su uso en diversos campos, es por esto importante para la Universidad de las Fuerza Armadas, la constante investigación en aplicaciones que se desarrollen a la par de la nueva tecnología.

Se ha producido una constante evolución en el mundo, pudiendo llegar a encontrar en el mercado dispositivos de tamaño reducido, con un bajo consumo de energía, tales como las plataformas Arduino y Xbee; sin dejar de brindar fiabilidad en la toma de datos del dispositivo, lo cual se plasmará en esta investigación.

1.3.Alcance del proyecto

Diseñar una red BAN (Body Address Network) que permita el registro de pacientes y el flujo de la información de 5 parámetros vitales, como son: Frecuencia cardíaca, frecuencia respiratoria, temperatura, presión arterial y glucosa con la ayuda del Kit de sensores E-health V2.0, los cuales se configurarán desde su punto inicial de tal manera que tomen datos del paciente y los envíen inalámbricamente al nodo, donde serán validados para su posterior uso.

Los datos obtenidos podrán ser manipulados, validados y visualizados en una Laptop, mientras que en un teléfono inteligente únicamente los podremos visualizar.

La solución incluye el diseño y construcción del elemento coordinador, el cual configura y establece la red, determina el canal de acceso, la red de transporte y proporciona los servicios de direccionamiento; además de una base de datos, donde se almacenará la información intercambiada para su análisis y verificación.

Cabe mencionar que el costo de la implementación de la red es alto, debido a los dispositivos necesarios, tales como: sensores, plataformas inalámbricas, licencias, entre otros.

1.4.Objetivos:

1.4.1. General

Diseñar e implementar una red de sensores que permita la obtención de datos de los principales signos vitales del cuerpo humano, así como su transmisión mediante dispositivos que utilicen el protocolo IEEE 802.15.4.

1.4.2. Específicos:

- Diseñar una Red Single-hop, que permita transportar las señales médicas tales como temperatura, presión, ritmo cardíaco, frecuencia respiratoria y glucosa a través de dispositivo con el estándar IEEE 802.15.4.
- Realizar la programación de los sensores que permita tomar las diferentes señales médicas del cuerpo humano.
- Validar la red de sensores para monitorización del paciente en diferentes escenarios.
- Implementar la red BAN y realizar pruebas de funcionamiento.
- Diseñar una base de datos que sirva como central, para guardar los valores obtenidos de los signos vitales a través de los sensores, que permitirá llevar un registro de pacientes.

- Validar, regular y almacenar los datos obtenidos en el nodo de la red diseñada.
- Integración de la Base de Datos en un entorno Web para la visualización.

CAPITULO II: MARCO TEÓRICO

2.1.Estado del arte de redes inalámbricas de sensores en aplicaciones e-Health

El desarrollo de esta tecnología (WBAN) inició alrededor de 1995 en torno a la idea de utilizar las tecnologías de redes inalámbricas para implementar comunicaciones cerca y alrededor del cuerpo humano (Docsetools, 2014).

Para el 2001, el término "BAN" llegó a referirse sistemas en los que la comunicación se realice enteramente dentro, sobre y en las inmediaciones de un cuerpo humano. Sin embargo esto funciona a distancias cortas para la solución a esto un sistema WBAN puede utilizar tecnologías inalámbricas WPAN como puertas para alcanzar distancias más largas.

Con el pasar del tiempo fue aumentado el desarrollo de las redes de sensores inalámbricos tanto en el nivel comercial como en el académico sobre todo para aplicaciones médicas, específicamente para la monitorización de pacientes, en la actualidad existen varios tipos de sensores inalámbricos que ayudan a monitorear de forma remota y discreta los parámetro vitales del paciente.

Actualmente las redes de área corporal han logrado una especial relevancia, en particular por las tendencias que muestran el uso de la medicina en aplicaciones de monitoreo y emergencia. Han surgido también esquemas de estandarización y regulación que han permitido la creación de nuevos estándares de comunicación tales como Zigbee, UWB, Wifi, LTE, etc. los cuales dan soluciones a los distintos retos y necesidades que surgen a medida que se incorporan estas tecnologías al estilo de vida actual del ser humano.

2.2.Aplicaciones de las redes inalámbricas de sensores y actuadores.

Actualmente, Las redes inalámbricas de sensores se están aplicando con éxito en diferentes escenarios, tales como: sistemas de automoción, aplicaciones industriales, militares, aviónica, entornos inteligentes, identificación de productos, domótica y seguridad, control de consumo energético, monitorización de invernaderos y muchas aplicaciones más.

La domótica trata sobre la automatización de viviendas y edificios y es uno de los campos más atractivos para la aplicación de las redes de sensores inalámbricas, brindándonos beneficios como ahorro energético o aplicaciones de seguridad y protección tanto de personas como de bienes, Seguramente la innovación en este ámbito dotará muchos hogares de la comodidad de interacción con ella misma.

La agricultura es una de las áreas donde las redes de sensores tienen una gran repercusión pues en base a la medición de determinados parámetros se puede controlar la cantidad de agua, fertilizante o pesticida que las plantas necesiten, decidir el momento óptimo para la cosecha, optimizar la producción y la calidad de una cosecha o simplemente gestionar alarmas por introducción de animales o daños provocados por las heladas.

La inteligencia ambiental trata buscar el bienestar para el ciudadano y conseguir una nueva relación más amigable, racional, productiva, sostenible y segura del individuo con su entorno.

Pues cuida al máximo el entorno en el que las personas estamos envueltas y asistidas por inteligentes interfaces embebidos en objetos cotidianos ayudándonos a monitorear zonas de riesgo, detectar inundaciones, explotación de animales en su hábitat natural, etc.

El monitoreo de estructuras es otra aplicación muy importante pues las estructuras físicas como puentes, edificios y construcciones experimentan vibraciones ya sea

ocasionadas por actividades normales o por fenómenos naturales, las variaciones en los comportamientos indican fatiga o cambios mecánicos urgentes.

2.3.Aplicaciones en el sector de la salud

Definitivamente las aplicaciones médicas son el núcleo actual del desarrollo de las tecnologías de área corporal. La posibilidad de interactuar con el sistema nervioso central del ser humano no está lejos y la gran cantidad de beneficios que traerá el buen aprovechamiento de este tipo de conocimientos incrementará la calidad de vida de los pacientes, en temas como prótesis y reemplazo artificial de órganos de los sentidos y extremidades corporales. En el ámbito del diagnóstico se mejorará la precisión de los exámenes y se incrementará notablemente la efectividad de las cirugías y tratamientos médicos para reducir los efectos colaterales.

En este sector se las utiliza especialmente para el control continuo y registro de parámetros vitales de los pacientes que sufren de enfermedades crónicas tales como ataques de la diabetes, el asma y el corazón.

- Una red BAN en un paciente puede alertar al hospital, incluso antes de tener un ataque al corazón, a través de medición de los cambios en sus signos vitales.Una red BAN en un paciente diabético puede auto inyectarse insulina a través de una bomba, tan pronto como se detecte la caída de nivel de insulina.
- Gestión y control de tratamientos, prótesis inteligentes y mecanismos que mejoran la calidad de vida de los pacientes.
- Mejorar la interacción de personas que tienen limitaciones físicas o problemas con órganos internos o de los sentidos.
- Asistencia en cirugías y diagnóstico de enfermedades.

2.4.Redes BAN

Una Body Area network (Red de área corporal), también conocida como una red de área corporal inalámbrica, es una red inalámbrica de dispositivos informáticos portátiles de baja potencia. La mencionada red se compone de varias unidades de sensores miniaturizados en el cuerpo junto con una sola unidad central.

Las redes de área corporal son sistemas de comunicaciones de pequeña escala. Las transmisiones se realizan dentro, alrededor o sobre el cuerpo humano, por consiguiente las transmisiones de área corporal tienen características evidentes. Una de las primeras es que la cobertura se encuentra confinada a distancias no superiores a los 2 ó 3 metros, y en segundo lugar, las emisiones de energía de este tipo de tecnologías son muy bajas, lo que contribuye a la larga vida de las baterías de los dispositivos, la reducción de los niveles de interferencia para la coexistencia con otras tecnologías, y la operación con potencias por debajo de los niveles que pueden ser dañinos para el ser humano.

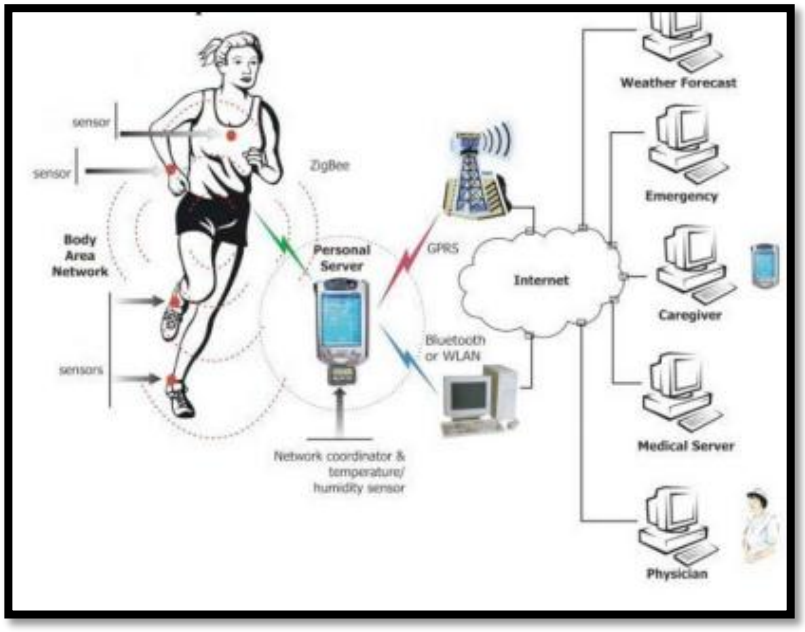


Figura 1. Red BAN
Fuente (Zabala G.)

2.5.Redes inalámbricas de sensores

Una red de sensores, es la colección de un conjunto de nodos sensores, su función es proveer una infraestructura de comunicación inalámbrica para poder monitorear algo en específico (temperatura, presión, movimiento, etc.).

Dentro de una red se tienen diferentes tipos de sensores tales como los sísmicos, térmicos, visuales e infrarrojos, y estos monitorean una amplia variedad de condiciones ambientales y hasta ciertas características de objetos que se encuentran en movimiento.

Los retos que involucra una red inalámbrica de sensores son el tipo de arquitectura, la diseminación y recolección de información, las técnicas adoptadas por los sensores para, localización y aumentar la eficiencia en el consumo de energía, Este último es factor primario que limita la red.

Áreas que hacen de las redes una categoría distinta dentro de las redes inalámbricas son:

- Movilidad de los nodos
- Tamaño de la red
- Densidad de la red
- Fusión de datos e información
- Limitación de energía

2.6.Redes 802.15.4 Zigbee

Zigbee no es una red más bien es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4, Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías, es decir bajo consumo.
Zigbee tiene 3 principales características que lo diferencian de otras tecnologías:

- Bajo consumo.
- Topología de red en malla.
- Fácil integración

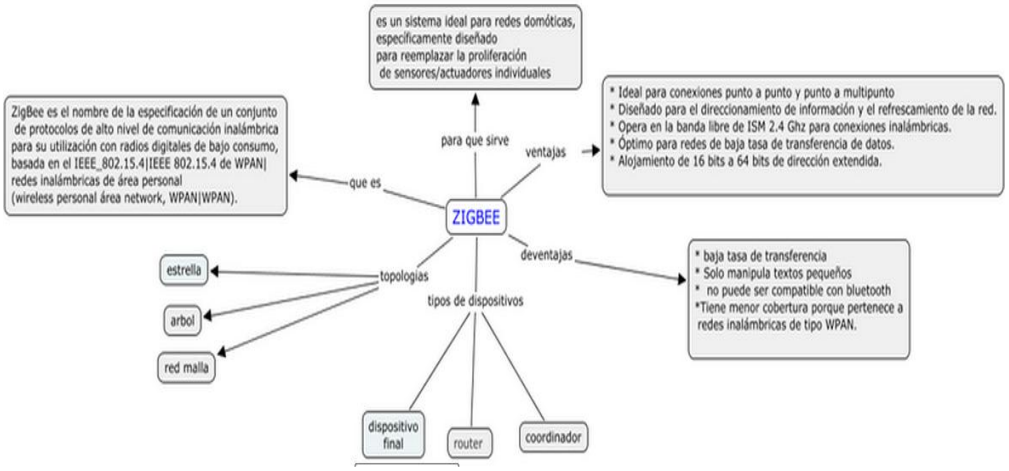


Figura 2. Ventajas Zigbee
Fuente (Glen, M. & Moreno J, 2012)

2.7.Factores de diseño en redes de sensores inalámbricos

2.7.1. Topología de la red de sensores

El manejo de la topología es muy importante y debe ser tratado cuidadosamente, en primera instancia los nodos sensores pueden ser desplegados de diferentes formas en la zona de estudio, más en nuestro caso será depositados uno a uno por tratarse de una red que toma datos de un humano. Luego de este despliegue la red de sensores puede tener frecuentes cambios por diferentes factores provocando que la topología varié también, ya sea por los cambios en la posición de los nodos sensores, capacidad de alcance debido a obstáculos, mal funcionamiento de la red o simplemente porque no cumple con las tareas propuestas. Como solución a esto se debe desplegar nuevamente los sensores ya sea para reemplazar aquellos defectuosos que han fallado o por necesidades de cambio en la dinámica de una tarea.

2.7.2. Entorno

Los nodos sensores son desplegados ya sea muy cerca o directamente en el interior del fenómeno a ser observado, en este caso estarán sometidos a varias condiciones adversas ya sea entornos con una gran cantidad de ruido electromagnético, cambios bruscos de temperaturas, movimientos inesperados por parte del humano etc. Esta variedad de escenarios se debe tomar en cuenta pues afectan aspectos como la comunicación entre nodos y la tasa de fallos.

2.7.3. Sincronización:

Dado que los nodos de una red inalámbrica de sensores operan de forma independiente pueden o no sincronizarse, esto podría causar dificultades cando se trata de integrar e interpretar información censada en diversos nodos.

2.7.4. Medio de transmisión

En una red de sensores conectados de forma inalámbrica, la comunicación se puede establecer mediante radio, sistemas ópticos o infrarrojos, en el caso de las restricciones de hardware aplicables a redes de sensores de bajo consumo y bajo coste, existe un gran número de componentes basados en circuitos RF operando en el rango de 2,4 GHZ.

Otro modo de comunicación entre nodos en WSN es comunicándose mediante infrarrojos, la cual no necesita licencia y es robusta, los transceptores basados en infrarrojos son baratos y fáciles de construir.

Tabla 1.*Tecnologías para redes corporales*

TECNOLOGÍAS DE REDES CORPORALES							
Tecnología	**Espectro**	**Capa física**	**Acceso de canal**	**Máxima velocidad de datos**	**Cobertura**	**Potencia**	**Interferencia**
IrDA	Infrarrojos 850nm	Rayos ópticos	Polling	4Mbps	< 10 m	Depende de la distancia	SI
MIT/IBM PAN	0.1 – 1Mhz	DSSS con ASK/FSK	CSMA-CA/CD, TDMA, o CDMA NO	417 Kbps (teórico) 2400bps (IBM), 9600bps (MIT)	El cuerpo humano	50 pA	NO
BodyLAN		OOC (espectro disperso en el tiempo)	Esquema TDMA complejo	32 Kbps	<10 m	5,4 mA	SI
Redes de tejidos (FAN)	125 KHz RFID	Campos RF	Polling	1-10 Kbps (duplex) 100 Kbps (simplex)	Múltiples antenas (<2cm)	Extrem. baja	NO
IEEE 802.15.1	2.4 GHz banda ISM	FHSS; 1600 hops x segundo	Polling Maestro – esclavo TDD	< 1Mbps	< 10m	1mA -60mA	SI
IEEE 802.15.3	2, 402 - 2,480 GHz banda ISM	QPSK sin codificar QPSK código trellis, o esquema QAM 16/32/64	CSMA-CA y GTS en estructura superframe	11 – 55 Mbps	<10m	<80 mA	SI
IEEE 802.15.4	2,4 GHz y 868/915 Mhz	DSSS con BPSK o MSK (O-QPSK)	CSMA-CA y GTS en estructura superframe	868 MHz-20Kbps, 915 MHz-4ª Kbps, y 2,4 Ghz-250 Kbps	<20m	20-50 pA	SI

Fuente (Digi International Inc.2009)

2.8.Arquitectura de un nodo sensor

2.8.1. Nodo sensor.

Un nodo sensor es un elemento computacional con capacidad de procesamiento, memoria, interface de comunicación y puede formar conjuntos de sensores.

2.8.2. Componentes de un Nodo Sensor Inalámbrico

Un nodo sensor inalámbrico (WSN) se compone de la detección, procesamiento de la señal, toma y almacenamiento de datos, y conexiones inalámbricas, las cuales

operan gracias al abastecimiento de energía proporcionado por una batería o fuente de energía.

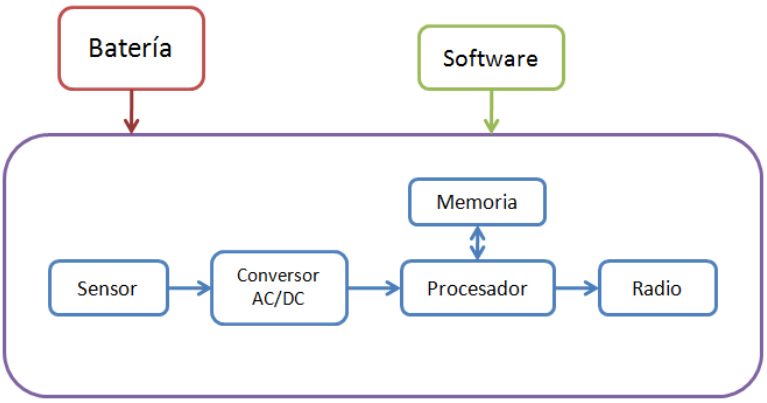


Figura 3. Esquema Nodo Sensor Inalámbrico

2.8.3. Fuente de Energía

Un aspecto importante en el desarrollo de un nodo de sensor inalámbrico es el de garantizar que exista la cantidad adecuada de energía disponible para potenciar el sistema. El nodo sensor consume potencia debido a la detección de los sensores, el procesamiento y comunicación, ya que más energía es requerida para la transmisión de datos que para cualquier otro proceso.

Las fuentes de energía comúnmente utilizadas son las baterías, sin embargo debido a sus limitaciones, los nodos se construyen teniendo en cuenta la conservación de energía, por lo que trabajan consumiendo poca potencia en modo "*sleep*".

Las fuentes de alimentación en redes de sensores se clasifican en:

- **Recargables:**

Las baterías recargables usan reacciones electroquímicas que son eléctricamente reversibles, es decir: cuando la reacción transcurre en un sentido, se agotan los materiales de la pila mientras se genera una corriente eléctrica.

Para que la reacción transcurra en sentido inverso, es necesaria una corriente eléctrica para regenerar los materiales consumidos.

Las baterías recargables vienen en diferentes tamaños y emplean diferentes combinaciones de productos químicos. Las celdas secundarias ("batería recargable") utilizadas con más frecuencia son las de plomo-ácido, la de níquel-cadmio (NiCd), la de níquel-metal hidruro (NiMH), la de iones de litio (Li-ion), y la de polímero de iones de litio (polímero de Li-ion).

Las baterías recargables pueden ofrecer beneficios económicos y ambientales en comparación con las pilas desechable.

- **No recargables:**

Son aquellas que tienen una sola vida útil, pues una vez descargadas no funcionan más y pueden llegar a contaminar grandes cantidades de agua.

- **Regenerativas:**

Aquellas que tienen la capacidad de regenerar energía a partir de un parámetro físico de estudio.

2.8.4. Microcontrolador

Dispositivo que ejecuta tareas, realiza el procesamiento, almacenamiento de datos y controla la funcionalidad de otros componentes en un nodo sensor. Las necesidades de cómputo y de almacenamiento en un nodo sensor inalámbrico (WSN) dependen de la aplicación, para lo cual pueden utilizarse microcontroladores desde los 8 a los 64 bits, mientras que los requerimientos de almacenamiento oscilan entre los 0.01 hasta los 100 GB.

En el mercado no es extraño encontrar modelos de microcontroladores que incluyan una memoria externa, como las memorias flash. En la actualidad los procesadores más utilizados son:

- Atmel AVR: Los AVR pertenecen a la familia de microcontroladores RISC de Atmel. El AVR es un CPU de arquitectura Harvard que posee 32 registros de 8 bits.
- MSP430: Procesadores de ultra bajo consumo conformados por conjuntos de periféricos específicos para varias aplicaciones. Dispositivo de la familia RISC de 16 bits, así como registros de 16 bits y generadores de constantes que contribuyen a la eficiencia del código.
- Intel 8051: Microcontrolador desarrollado por Intel para el uso de sistemas embebidos. Microcontrolador basado en la arquitectura Harvard, que permite direccionar 64 KB de ROM externa.
- PIC: Familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. Desarrollados por la microelectrónica General Instruments.

2.8.5. Sensores

Los sensores son dispositivos que producen una respuesta medible a un cambio de una condición física, como la temperatura o presión, convirtiéndola en una señal. Dicha señal es análoga, la cual es digitalizada por un convertidor ADC y enviada al microcontrolador para ser procesada. Las WSN por lo común son dispositivos electrónicos muy pequeños, por lo que solo pueden ser equipados con una fuente de poder limitada de menos de 0.5-2 amperios/hora y 1.2-3.7 voltios.

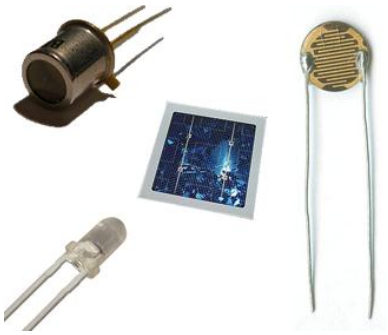


Figura 4. Sensores

Los sensores se dividen en tres categorías:

- Sensores Pasivos Omnidireccionales: Adquieren los datos sin necesidad de manipular el ambiente. Se autoalimentan, lo que quiere decir que la energía solo se utiliza para amplificar la señal análoga.

- Sensores Pasivos Unidireccionales: Estos sensores conocen de manera definida la dirección de medida, por ejemplo una cámara fotográfica.

- Sensores Activos: Los sensores activos examinan y prueban el ambiente, tal como lo hace un sonar o un radar. Estos sensores requieren de una fuente de energía continua para su funcionamiento.

2.8.6. Transceptor

Los nodos de sensores inalámbricos a menudo hacen uso de la banda ISM, lo que significa el uso del espectro radioeléctrico. Las posibles opciones de medios de transmisión inalámbrica son la radio frecuencia, comunicaciones ópticas y el infra rojo. Las WSN's tienden a utilizar las frecuencias de comunicación: 173, 433, 868 y 195 MHz y 2.4 GHz. La funcionalidad de un transmisor y un receptor combinados dan como resultado un solo dispositivo conocido como transceptor, el cual carece de identificadores únicos y trabaja en cuatros estados que son el de transmitir, recibir, modo inactivo y sleep.

2.9.Protocolos de enrutamiento para WSN

2.9.1. Modelos de enrutamiento

Una red se encuentra conformada por varios nodos, los cuales no poseen conocimiento de la topología, ya que deben descubrirla. Para dicha acción, cuando un nodo nuevo se una a la red, anuncia su presencia y escucha los mensajes de

broadcast que envían sus vecinos, de esta forma el nodo conoce a los demás nodos de la red, así como su alcance y de la manera de crear rutas a través de ellos. De igual manera el nuevo nodo anuncia al resto de nodos, como acceder a la red a través de él. De esta manera se realiza la convergencia en la red, donde cada nodo sabrá cuál es su vecino y las formas en las que pueden ser alcanzados.

Los modelos de enrutamiento para WSN deben cumplir con:

- Conservar una tabla de enrutamiento lo más pequeña posible
- Escoger la mejor ruta, ya sea la más confiable, rápida o de menor costo
- Actualizar su tabla de enrutamiento regularmente, para conocer los nuevos nodos en la red, posición o caída de los mismos.
- Requerir tiempo para la convergencia de la red

2.9.1.1.Modelo de un salto

El modelo de un salto es el más sencillo, ya que su funcionamiento se basa en la comunicación directa. Todos los nodos que conforman la red transmiten información a la estación base. Los inconvenientes de este modelo son el alto consumo energético, así como un limitado rango de transmisión. Muchas de las veces las transmisiones no logran alcanzar a la estación base, debido a que poseen una distancia máxima de radio, haciendo de esta comunicación directa una solución poco factible en redes inalámbricas.

2.9.1.2.Modelo Multi-hop

En el modelo Multi-salto, un nodo transmite sus datos reenviándolos a uno de sus nodos vecinos, que se encuentre más cercano a la estación base, y este a su vez enviará la información a otro nodo próximo, hasta alcanzar la mota base. Este comportamiento hace que los datos viajen desde el origen hacia el destino de salto en salto, desde un nodo a otro, hasta alcanzar su punto final.

Figura 5. Modelo Multi-Salto

2.10. Componentes plataforma de sensores E-Health

2.10.1. Escudo E-Health

La plataforma de sensores E-Health permite la ejecución de aplicaciones biométricas y médicas, con el uso de Arduino y Raspberry Pi, donde el monitoreo del cuerpo humano es necesario, usando diez diferentes sensores como: pulso, oxígeno en la sangre, respiración, temperatura corporal, electrocardiograma, glucómetro, respuesta galvánica de la piel (sudor), presión arterial,electro miógrafo (músculos) y posición del paciente.

Es un Sistema fácil y sencillo para el manejo de sensores que se desempeñan en actividades biométricas y médicas.

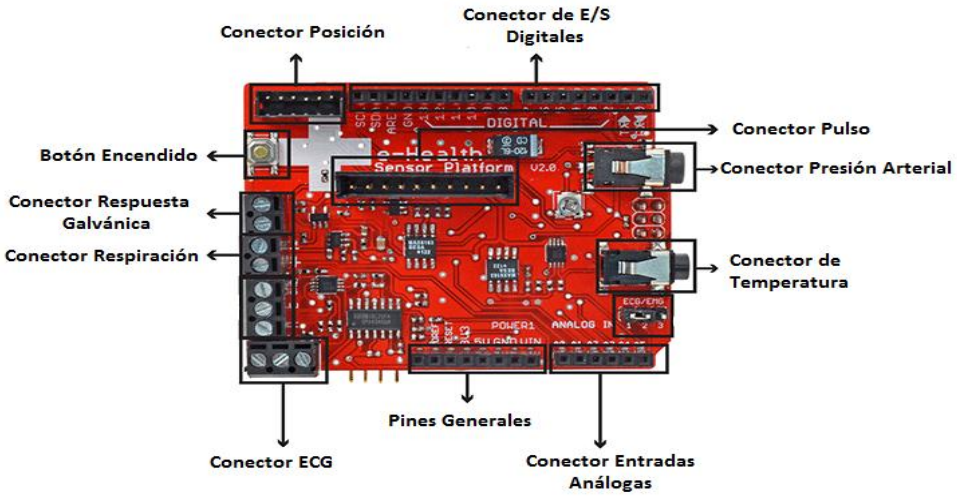


Figura 6. Plataforma E-health, vista posterior
Fuente (Cooking Hacks.2013)

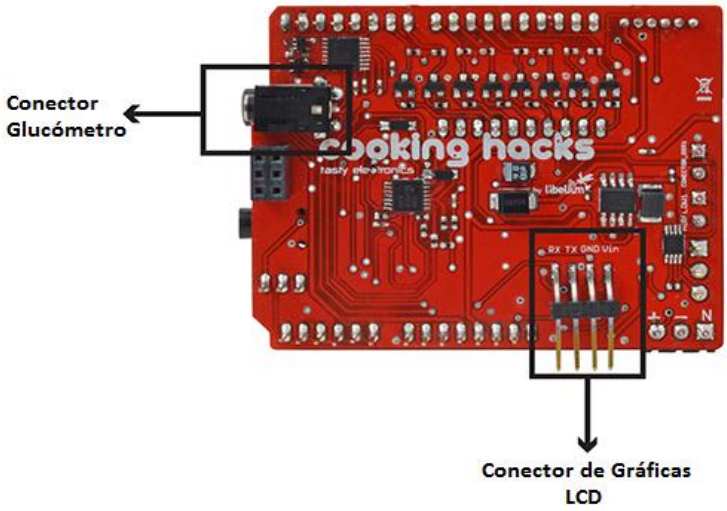


Figura 7. Plataforma E-health, vista anterior
Fuente (Cooking Hacks.2013)

2.10.2. Plataforma Arduino y Raspberry

Arduino es una plataforma de Hardware Libre, establecida en una placa sencilla compuesta por cinco pines de entradas analógicas, 6 pines de salidas analógicas, trece entradas/salidas digitales, un puerto serial, conexión USB, un microcontrolador Atmegaxx8 como elemento principal, un pulsador que permite reiniciar la placa ante cualquier falla de procesos. El software que utiliza Arduino es un lenguaje de programación llamado Processing/Wiring (lenguaje de programación de código abierto basado en Java), así como el cargador de arranque que se ejecuta en la placa.

Entre sus especificaciones técnicas se encuentran:

- Voltaje de Entrada 7-12 Voltios
- Voltaje de operación 5 Voltios
- Memoria Flash de 32 KB
- Velocidad de Reloj 16 MHz
- Memoria RAM de 1 KB
- EEPROM 512 bytes

Figura 8. Plataforma Arduino

Raspberry Pi es un ordenador del tamaño de una tarjeta de crédito de bajo costo, creado en Reino Unido, por la fundación Raspberry Pi, el cual se conecta a un monitor, televisión o computadora, y utiliza un mouse y teclado estándar. Es un dispositivo pequeño que permite a desarrolladores de todas las edades a explorar la computación, y aprender como programar en lenguajes como Scratch y Phyton. Es capaz de realizar todo lo que se espera de una computadora de escritorio, desde navegar en el internet, procesar palabras, hasta jugar video juegos de alta definición.

Sus especificaciones técnicas son:

- Sistema operativo Linux (Debian, Fedora)
- Procesador Gráfico VideoCore IV
- Memoria RAM 512 MB
- Procesador Central ARM1176JZF-S a 700 MHz
- Capacidad de almacenamiento SD o SDHC

2.10.3. Módulo Zigbee

El módulo está diseñado a partir de un chip de Ember con una radio y un microcontrolador, el microcontrolador interno saca al exterior sus pines a través de

las patillas del módulo, en sudatasheet se puede encontrar la descripción detallada de cada una de ellas.

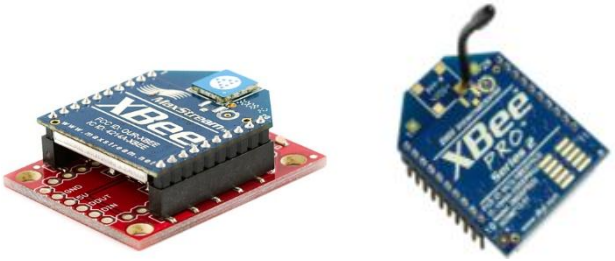


Figura 9. Módulos Zigbee

El microcontrolador interno del módulo se comunica con el exterior mediante una UART, esta UART la podemos utilizar para cambiar el firmware del módulo a través de un bootloader que lleva ya grabado, o para conectar el microcontrolador interno con la UART de un microcontrolador externo para que éste envíe y reciba datos a través del módulo XBee.

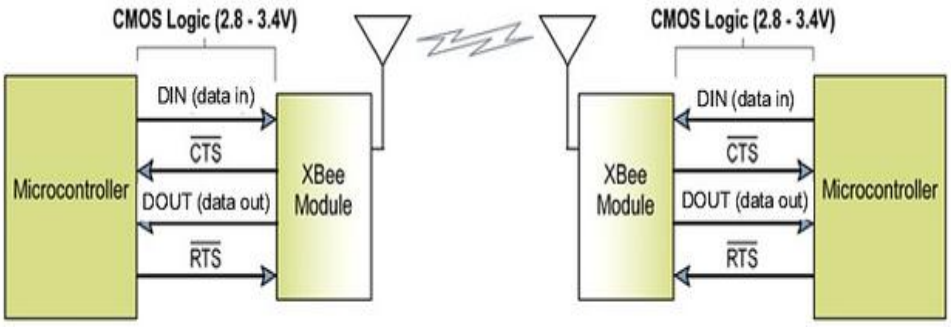


Figura 10.Comunicación módulo Zigbee
Fuente (Digi International Inc.2009)

2.10.4. Sensores E-Health

A continuación detallaremos cada uno de los sensores E-Health:

- Tensiómetro:

Figura 11.Sensor Tensiómetro
Fuente (Cooking Hacks.2013)

La presión arterial es la presión de sangre en las arterias, la presión arterial se registra como dos valores, presión sistólica (cuando el corazón late) y la diastólica (cuando el corazón se relaja).

En la siguiente tabla se detallan los valores de presión arterial para mayores de 18 años:

Tabla 2. *Niveles de presión arterial*

	Sistólica (mmHg)	**Diastólica (mmHg)**
Hipotensión	<90	<60
Normal	90-119	60-79
Pre-hipertensión	120-139	80-89
Hipertensión Nivel 1	140-159	90-99
Hipertensión Nivel 2	160-179	100-109
Crisis Hipertensiva	>=180	>=110

Fuente (Alliance, 2008)

Para el correcto funcionamiento se debe colocar la muñeca en un plano horizontalcon el corazón como muestra la figura.

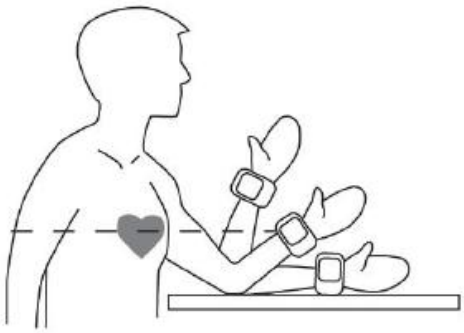


Figura 12. Colocación del tensiómetro
Fuente (Cooking Hacks.2013)

- Sensor de temperatura:

Figura 13.Sensor medidor de temperatura
Fuente (Cooking Hacks.2013)

Este es el sensor más conocido, permite conocer la temperatura corporal del paciente y en base a ello conocer alguna anomalía física. La razón es que gran número de enfermedades están acompañadas por características en los cambios de temperatura.

Los rangos de temperatura son:

- Hipotermia < 35.0°C
- Norma 36.5 a 37.5 ° C
- Fiebre o Hipertermia 37.5 a 38.3 °C
- Hiperpirexia>40.0 a 41.5 °C

Cuando usamos este sensor en realidad medimos una tensión, por cada tensión se tiene una correspondencia de temperatura, por lo que es importante la calibración de este sensor para así obtener valores fiables.

- Sensor de pulso y oxigenación de la sangre:

Figura 14.Sensor Oxigenación y pulsímetro
Fuente (Cooking Hacks.2013)

La saturación de oxígeno se refiere a la medición de la cantidad de oxígeno disuelto en la sangre, basado en la detección de la hemoglobina y desoxihemoglobina. Se utilizan dos longitudes de onda diferentes para medir la diferencia de absorción de la hemoglobina y la desoxihemoglobina.

Los coeficientes de absorción se miden usando 2 longitudes de onda de 660nm (luz roja) y 940nm (luz infrarroja)

- Glucómetro:

Figura 15. Sensor Glucómetro
Fuente (Cooking Hacks.2013)

El glucómetro es un dispositivo médico que determina la concentración aproximada de glucosa en la sangre. Con una pequeña gota de sangre, depositada en las tiras reactivas es posible leer y calcular los niveles de glucosa, los cuales presentan medidas de mg/dL o mmol/L.

- Electro miógrafo:

Figura 16. Sensor para medir las contracciones musculares
Fuente (Cooking Hacks.2013)

Este sensor conocido por sus siglas “EMG” sirve para la medición de las contracciones musculares que se presentan en las diferentes partes del cuerpo para poder visualizar si el paciente sufre de desórdenes musculares como ejemplo tenemos el parquin son que es un movimiento continuo de los músculos por fallas neurológicas.

- Electrocardiógrafo:

Figura 17. Sensor para electrocardiograma
Fuente (Cooking Hacks.2013)

El electrocardiograma (ECG) se ha convertido en una de las pruebas médicas más comúnmente utilizados en la medicina moderna. Su utilidad en el diagnóstico de una miríada de patologías cardíacas que van desde la isquemia de miocardio y el infarto al síncope y palpitaciones ha sido de gran valor para los médicos durante décadas.

Se basa en la medición de pulsos eléctricos en base a 3 sondas o electrodos.

Se obtiene un valor de tensión medido en voltios, siempre y cuando los electrodos estén correctamente conectados al paciente.

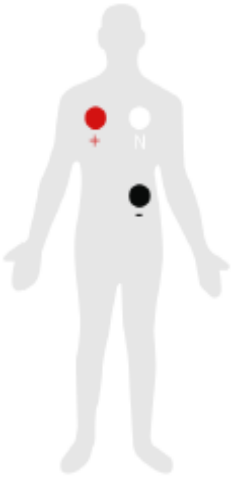


Figura 18. Conexión de los electrodos al paciente
Fuente (Cooking Hacks.2013)

- Sensor de posición corporal:

Figura 19. Sensor posición corporal
Fuente (Cooking Hacks.2013)

Permite obtener la posición del cuerpo en base a 3 acelerómetros (eje X,Y y Z)

La posición del cuerpo del sensor E-health supervisa cinco posiciones diferentes de pacientes (de pie/sentado, supina, prona, izquierda y derecha). En muchos casos, es necesario supervisar las posiciones del cuerpo y los movimientos realizados a causa de su relación con enfermedades concretas (es decir, apnea del sueño y el síndrome de piernas inquietas). Analizando los movimientos durante el sueño también ayuda en la determinación de la calidad del sueño y los patrones de sueño irregulares. El sensor de posición del cuerpo puede ayudar también a detectar desmayos o caídas de las personas mayores o personas con discapacidad.

Tiene rangos definidos para las tensiones:

- Tensión de alimentación:1,95-3,6 V
- Tensión de interfaz: 1,6 V- 3,6 V

- Sensor de flujo de respiración:

Figura 20. Sensor frecuencia respiratoria
Fuente (Cooking Hacks.2013)

Los índices anormales dentro del rango de la respiración, indican una gran inestabilidad física, por lo tanto es crucial monitorear cambios, para conocer el estado del paciente. Utilizado para detectar enfermedades como la hipoxemia o apnea.

Este dispositivo consiste en un sensor termopar, que detecta con precisión los cambios de flujo de aire, así como la temperatura del aire expulsados por las fosas nasales.

Un adulto normal tiene un rango respiratorio entre 15 y 30 respiraciones por minuto.

- Sensor de Respuesta Galvánica de la Piel:

Figura 21. Sensor Sudoración
Fuente (Cooking Hacks.2013)

La respuesta galvánica de la piel, es un método de medir la conductancia eléctrica de la piel, que varía dependiendo del nivel de humedad. Esto es importante, ya que las glándulas de sudor son controladas por el sistema nervioso simpático, así que en momentos de emociones fuertes, la resistencia eléctrica de la piel varía.

El sudor de la piel es utilizado como un indicador de alteraciones físicas o psicológicas, por lo que este sensor mide la conductancia eléctrica entre dos puntos, cuando los niveles de sudoración son altos, la resistencia de la piel decrece, pero al tener una piel seca la resistencia aumenta.

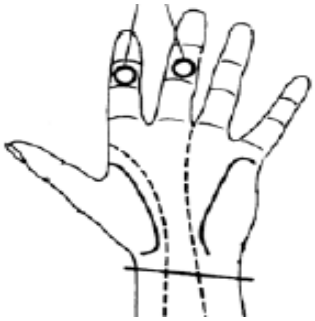


Figura 22. Colocación del sensor GSR
Fuente (Cooking Hacks.2013)

- **Conexión en conjunto de los sensores**

Figura 23. Escudo E-Health con sensores conectados
Fuente (Cooking Hacks.2013)

2.11. Tecnologías para la implementación de la base de datos.

Una base de datos es un conjunto de información interrelacionada, que pertenecen a un mismo contexto, la cual se almacena sistemáticamente para su posterior acceso.

Existen programas llamados “Sistemas Gestores de Base de Datos” (DBMS), que almacenan los datos para su acceso de manera rápida y estructurada. El objetivo de los DBMS es brindar la interfaz entre el conjunto de datos y los usuarios.

Entre las ventajas de las bases de datos se tiene:

- Disponibilidad de datos
- Facilidad para compartir datos
- Consistencia, integridad y seguridad
- Acceso sencillo para usuarios finales

2.11.1. Common Gateway Interface (CGI)

Es una tecnología que permite acceder a los datos mediante la comunicación del servidor Web con la base de datos. Entre sus ventajas se encuentra su sencillez, ya que es fácil de entender, así como su versatilidad, puesto que su elaboración puede realizarse en varios lenguajes.

Sin embargo el CGI, presenta una desventaja en términos de su eficiencia, debido a que el Servidor Web debe conectarse y desconectarse de la base de datos, cada vez que se realice una petición.

2.11.2. Interfaz de Programación de Aplicaciones (API)

Es un grupo de rutinas para la construcción de una interfaz, define como llamar desde un programa un servicio prestado. Una buena interfaz hace más fácil el desarrollar un programa, ya que provee todos los bloques para dicha construcción, por lo que el programador solo debe unir los bloques, evitando así el programar desde cero. Una API, son funciones que el sistema operativo ofrece al programador, por ejemplo escribir un carácter o un fichero.

2.11.3. Java, JDBC

Es una aplicación que consiste en un grupo de interfaces, que permite ejecutar operaciones sobre otras bases de datos, utilizando el lenguaje de programación Java,

y el dialecto SQL. Una ventaja de JDBC, es que al estar escrito en Java es independiente de la plataforma, lo que quiere decir que no es necesario crear un programa para cada tipo de base de datos, por ejemplo una misma aplicación hecha en JDBC, puede manipular una base de datos creada en Oracle, Sybase, etc.

2.11.4. JavaScript

JavaScript es un lenguaje similar al lenguaje C, diseñado para crear escritos, eventos, acciones, etc. dentro de un programa HTML y bajo cualquier plataforma.

CAPÍTULO III: NODO SENSOR

3.1.Nodo sensor E-Health

Un nodo sensor E-Health es un elemento computacional con capacidad de procesamiento, memoria e interface de comunicación y forma un conjunto con los 6 sensores utilizados (flujo de aire, temperatura, respuesta galvánica de la piel, oxigenación, posición corporal y glucómetro)

3.1.1. Hardware del nodo sensor E-Health

El hardware del nodo sensor E-Health se compone de un transceptor, procesador, 6 sensores E-Health, memoria y batería.

Los componentes brindan la opción de comunicación y ejecutar tareas que requieren procesamiento más allá de efectuar funciones de censado.

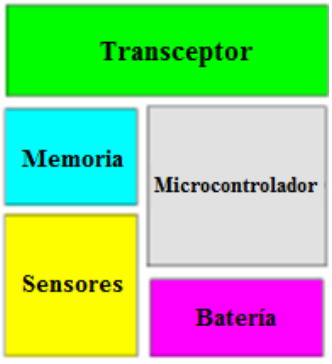


Figura 24. Componentes de hardware de un nodo sensor

La capacidad de procesamiento está dada por el procesador que posee el ARDUINO (EEPROM 512 bytes), así mismo posee una memoria interna en el microprocesador (Memoria RAM de 1 KB), La comunicación se realiza mediante el módulo Zigbee que es el transceptor (transmisor/receptor), además se tiene la fuente de alimentación que es una batería la cual detallaremos más adelante. En cuanto a los

sensores estos son los responsables de monitorear el parámetro de interés e informar del mismo.

3.1.2. Alimentación del nodo sensor

En este punto existen limitaciones debido a la forma de alimentar el nodo sensor E-Health.

Sin embargo debido a la facilidad, movilidad, precio, funcionalidad, tipo y regulación de carga, hemos decidido utilizar una batería recargable de litio de 3000 mili Amperios y 7.4 Voltios, la cual después de realizar pruebas de funcionamiento garantiza que la red trabaje sin problemas más de 4 horas, lo que es suficiente para una visita médica si se diese el caso, y si se quisiera implementar el sistema en un centro médico la mejor forma de alimentar seria directo a una toma de energía.

3.2.Diseño del nodo

3.2.1. Selección de Sensores.

La selección de sensores para este proyecto, se basó en la importancia que tienen dichos dispositivos en la medición de los signos vitales más importantes, como la respiración, pulso, sudoración, posición, temperatura y glucosa.

Al alterarse estos signos vitales, es posible detectar anomalías relacionadas a la salud del paciente, lo que permite actuar con rapidez, para ayudar y estabilizar a la persona enferma.

Para el desarrollo del nodo era posible escoger entre diez sensores médicos, uno de ellos el sensor de presión el cual forma parte de la medición de un signo vital muy importante.

Sin embargo el sensor de Presión no podía trabajar en conjunto con los demás dispositivos, debido a que el escudo E-health es un sistema de código abierto, en el cual sus sensores han sido creados a partir de sensores convencionales.

Esto genera que ciertos sensores no sean estables, y deban ser trabajados de manera independiente, con el fin de no perder datos importantes en el momento de la transmisión de información.

Figura 25. Sensores Seleccionados
Fuente (Cooking Hacks.2013)

3.2.2. Selección entre Arduino o Raspberry Pi.

El escudo E-Health es compatible con plataformas de hardware como Arduino o Raspberry Pi, para el desarrollo de aplicaciones médicas. Ambos dispositivos fueron diseñados originalmente para ser herramientas de enseñanza, lo que los hace fáciles de aprender a usar.

Diferencias entre Arduino y Raspberry Pi

Al realizar una comparación de características entre estas dos herramientas, se puede conocer a qué tipo de proyectos están orientadas:

Tabla 3. *Características de Arduino y Raspberry*

	Arduino	**Raspberry Pi**
Precio	$ 30	$ 65 - $ 80
Tamaño	7.6 x 1.9 x 6.4 cm	8.6 x 5.4 x 1.7cm
Memoria	0.002 MB	512 MB
Velocidad de Reloj	16 MHz	700 MHz
OnBoard Network	Ninguna	Ethernet RJ45
Voltaje de Entrada	7 - 12 V	5 V
Memoria Flash	32 KB	Tarjeta SD (2 – 16 G)
Puertos USB	Uno	Dos
Sistema Operativo	Ninguno	Distribuciones de Linux
Entorno de Desarrollo Integrado (IDE)	Arduino	Scratch, IDLE, cualquiera con soporte Linux

Fuente (Hacedores, 2014)

Arduino tiene la capacidad analógica y en tiempo real, que Raspberry no posee, lo que permite el trabajo con cualquier tipo de sensor o chip. Si se desea realizar una lectura de sensores analógico con Raspberry Pi, se requiere de la asistencia de un hardware adicional.

Una ventaja de utilizar Arduino es que se poseen más recursos para el aprendizaje, más no así con Raspberry Pi, del cual se debe tener conocimientos de programación en Linux.

El encendido y apagado de Arduino se lo puede hacer en cualquier momento de forma segura, pero si se hace lo mismo con Raspberry Pi, la configuración se puede dañar al desconectarlo sin un apagado adecuado.

Al ver las características entre Arduino y Raspberry, se podría decir que este último es superior, sin embargo la simplicidad de Arduino, hace que sea la mejor opción para implementar proyectos de hardware.

3.2.3. Desarrollo

Se mostrarán las conexiones y ajustes necesarios de cada sensor

- Conexión Sensorde Pulso y Oxígeno en la Sangre (SPO2)

Se debe conectar el sensor en el Escudo E-Health. El módulo tiene una sola forma de conexión, para prevenir errores y hacerla más fácil

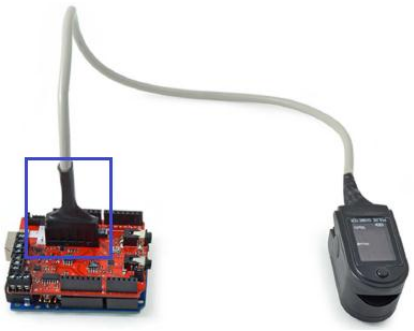


Figura 26. Conexión del sensor al Escudo E-Health
Fuente (Cooking Hacks.2013)

Se debe insertar el dedo en el sensor, y presionar el botón de ON. Después de unos segundos, se desplegarán los valores de pulso y oxígeno en la sangre, en la pantalla del sensor.

Figura 27.Valores del sensor de Pulso y Oxígeno en la Sangre
Fuente (Cooking Hacks.2013)

- *Librerías*

El sensor de Pulso y Oxigenación trabaja mediante interrupciones, por lo que es necesario incluir una librería especial, la cual se indica a continuación.

#include < PinChangeInt.h>

La línea de código que permite obtener datos desde el sensor es:

PCintPort::attachInterrupt (6, readPulsiometer, RISING);

El pin número 6 del Arduino, es el pin desde donde el sensor envía la interrupción y desde donde la función readPulsioximeter será ejecutada.

Void readPulsiometer () { cont ++; if (cont == 50)

Antes de empezar a usar el sensor de Pulso y Oxigenación de la sangre, esté debe ser inicializado, por lo que se deben configurar parámetros básicos de comunicación entre Arduino y el sensor.

Para leer los valores que entrega el sensor se utiliza la siguiente función:

{eHealth.readPulsiometer ();}

Para visualizar los datos del sensor que están almacenados en variables privada se utilizan las siguientes funciones:

{int SPO2 = eHealth.getOxygenSaturation () int BPM = eHealth.getBPM ()}

- Sensor de Flujo de Aire (Respiración)

Este sensor tiene dos conectores, el positivo y negativo. Se debe conectar el cable de color rojo en la terminal marcada con el signo +, mientras que el cable de color negro se conecta en la terminal marcada con el signo -. Estas conexiones deben ser aseguradas ajustando los tornillos de las terminales.

Figura 28. Conexión sensor de Flujo de Aire
Fuente (Cooking Hacks.2013)

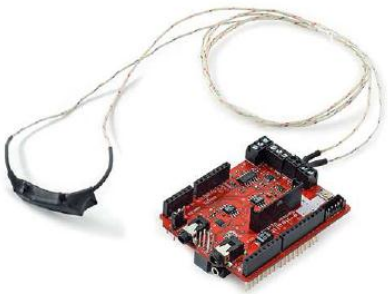


Figura 29. Sensor conectado al Escudo E-health
Fuente (Cooking Hacks.2013)

- *Librerías*

El sensor de Flujo de Aire, se conecta al Arduino mediante una entrada analógica y retorna valores entre 0 y 1024, con las siguientes funciones se pueden obtener estos valores directamente.

{int airFlow = eHealth.getAirFlow (); eHealth.airFlowWave (air);}

- Conexión Sensor de Temperatura Corporal

La precisión del sensor de temperatura corporal es suficiente para la mayoría de aplicaciones. Pero se puede mejorar esta precisión al calibrarla.

Cuando se usa el sensor de temperatura, lo que se mide en realidad es un voltaje, se pueden evitar errores en las mediciones de voltaje, y representar la relación entre el voltaje y la temperatura de manera más exacta, obteniendo así mejores lecturas de las temperaturas.

El proceso de calibración es una medición de valores reales de voltaje y resistencia. En el archivo de eHealth.cpp, se puede encontrar la función de getTemperature, donde los valores de Rc, Ra, Rb, RefTension se encuentran definidos por defecto.

eHealth.cpp

```
231
232     //|*****************************************************************************
233     //|   Name:    getTemperature()
234     //|   Description: Returns the corporal temperature.
235     //|   Param : void
236     //|   Returns: float with the corporal temperature value.
237     //|   Example: float temperature = eHealth.getTemperature();
238     //|*****************************************************************************
239
240     float eHealthClass::getTemperature(void)
241     {
242       //Local variables
243       float Temperature; //Corporal Temperature
244       float Resistance;  //Resistance of sensor.
245       float ganancia=5.0;
246       float Vcc=3.3;
247       float RefTension=3.0; // Voltage Reference of Wheatstone bridge.
248       float Ra=4700.0; //Wheatstone bridge resistance.
249       float Rc=4700.0; //Wheatstone bridge resistance.
250       float Rb=821.0; //Wheatstone bridge resistance.
251       int sensorValue = analogRead(A3);
```

Figura 30. Valores de Resistencias definidas por defecto
Fuente (Cooking Hacks.2013)

Si se miden estos valores con un multímetro, y se los modifica en la librería, se obtendrán mejores lecturas.

- *Librerías*

La temperatura corporal puede ser tomada como una función simple. Está función retorna un flotante (float), con el último valor de temperatura medido por el Arduino.

{float temperatura = e.Health.getTemperature (); }

- Conexión de Sensor de Posición

La conexión del sensor de posición es bastante sencilla, se debe conectar la cinta de cables en el Escudo E-health, como se muestra en la figura. Para luego ser colocada alrededor del torso del paciente, con el conector hacia abajo.

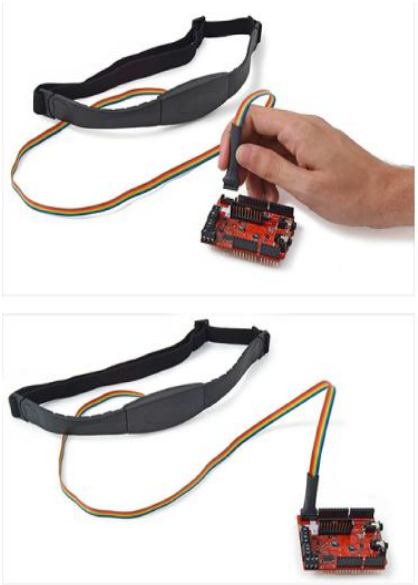


Figura 31. Conexión Sensor de Posición
Fuente (Cooking Hacks.2013)

Figura 32. Sensor de Posición colocado en el paciente
Fuente (Cooking Hacks.2013)

- *Librerías*

Para empezar a utilizar el sensor de posición, se debe inicializar el mismo con la siguiente función:

```
{ eHealth.initPositionSensor () ; }
```

La función descrita, retorna un valor que representa la posición del cuerpo que se encuentran almacenados en variables privadas de la clase e-Health.

```
{uint8_t position = eHealth.getBodyPosition (); }
```

Las posiciones del paciente con su respectiva numeración son:

- 1== Supina
- 2== Izquierda
- 3== Derecha
- 4 == Prona
- 5== Parado o Sentado

Para representar las posiciones de una manera amigable en la pantalla, se usa la función descrita a continuación:

{Serial.print { “Current position: ”); uint8_t position = eHealth.getBodyPosition ();
eHealth.print position (position); }

- Conexión del Sensor de Respuesta Galvánica de la Piel

El sensor de respuesta galvánica de la piel se mide en la punta de los dedos. Emociones como excitación, estrés, o shock, pueden resultar en la fluctuación de la conductividad de la piel.

Se debe realizar la calibración del sensor, en el archivo eHealth.cpp, se encuentran las funciones getSkinConductance y getSkinResistance, está última tiene un valor por defecto de 5 voltios. Si se mide este voltaje con un multímetro y se modifica en la librería, los valores obtenidos serán más precisos.

Figura 33. Voltaje en la Función getSkinResistance
Fuente (Cooking Hacks.2013)

Para calibrar se debe medir el valor del voltaje que entregan la medición entre las terminales de 0.5 voltios y tierra (GND), que pueden variar entre 0.5 y 0.498 voltios, como indica la figura.

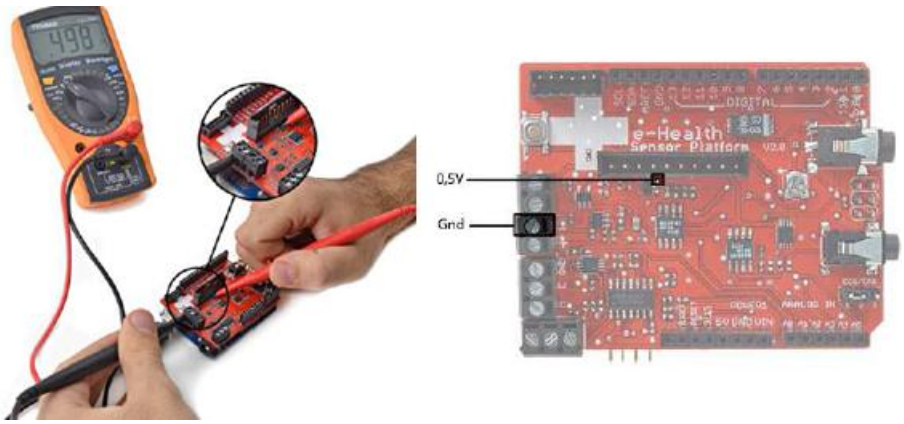


Figura 34. Calibración Sensor de Respuesta Galvánica de la Piel
Fuente (Cooking Hacks.2013)

La conexión del sensor, se la realiza al ajustar los cables en los contacto GSR, estos contactos no tiene polaridad.

Figura 35. Conexión Sensor de Respuesta Galvánica de la Piel
Fuente (Cooking Hacks.2013)

- *Librería*

Con la esta función simple es posible leer el valor del sensor de Respuesta Galvánica de la Piel. La librería retorna valores de la conductancia y resistencia de la piel.

{float conductance = eHealth.getSkinConductance (); float resistance = eHealth.getSkinResistance();

- Conexión del Glucómetro

Antes de realizar la conexión del glucómetro con el Escudo E-health, es necesario tener al menos una medición en la memoria del glucómetro. Después de esto se puede obtener toda la información contenida en el glucómetro como la fecha y el valor de la glucosa.

Se enciende el glucómetro y se coloca la tira reactiva cuando el dispositivo esté listo, se inserta la tira una vez que el indicador lo señale.

Figura 36. Glucómetro
Fuente (Cooking Hacks.2013)

Para tomar la muestra de sangre, se debe limpiar el dedo con alcohol, y picarlo con una lanceta o aguja.

Figura 37. Muestra de sangre
Fuente (Cooking Hacks.2013)

La gota de sangre debe ser colocada a un lado o encima de la tira reactiva, como se indica en la figura.

Figura 38. Colocación muestra de sangre en la tira reactiva
Fuente (Cooking Hacks.2013)

El glucómetro tomará unos segundos en calcular el azúcar en la sangre y almacenar este valor en su memoria.

Figura 39. Cálculo de azúcar en la sangre
Fuente (Cooking Hacks.2013)

Para extraer los datos del glucómetro hacia el Arduino, se debe conectar el cable, como se indica en la figura.

Figura 40. Conexión de Glucómetro al Escudo E-health
Fuente (Cooking Hacks.2013)

Una vez realizada la conexión, se debe observar el mensaje P-C en la pantalla del glucómetro, lo que señala la correcta conexión.

- *Librerías*

Con una función simple es posible realizar la lectura de las mediciones almacenadas en el glucómetro, y desplegarlas en la pantalla

{eHealth.readGlucometer (); Serial.begin(115200); }

La cantidad de datos leídos es accesible mediante la siguiente función pública.

{uint8_t numberOfData eHealthClass.getGlucometerLength () }

Para todos los programas que se detallaran a continuación se requiere importar la librería “eHealth.h”, la cual procede a importar todos los métodos y atributos disponibles para el escudo de sensores médicos.

Por otro lado se debe iniciar la comunicación serial de la tarjeta Arduino con una velocidad definida por el usuario en este caso se tiene que está en “9600 bps”

CAPITULO IV: DISEÑO DE LA BASE DE DATOS

4.1.Sistema Híbrido.

Hemos trabajado con un sistema hibrido entre SQL-Server, Visual Studio y Netbeans debido a sus prestaciones, tanto para la base de datos, interface web y conexión, pues las características de los software mencionados son suficientes para cumplir con los objetivos propuestos en nuestro proyecto.

4.1.1. SQL-Server:

Es fácil de manipular y amigable con el usuario pues ofrece una especie de paleta de herramientas, razón por la cual es pagado sin embargo, existe la versión light que es la que utilizamos para nuestro proyecto, la única limitación de esta versión es que almacena máximo 2000 registros, y no ofrece tanta seguridad sin embargo eso no interfiere con el cumplimiento de los objetivos.

4.1.2. Visual Studio:

Ofrece una plantilla web por defecto, la cual podemos editar acorde a nuestros requerimientos, ahorrándonos tiempo, ya que iniciar desde cero implicaría demasiado tiempo y conocimiento de programación.

4.1.3. Netbeans:

Permite la comunicación entre la base de datos y el Arduino el cual también se comunica con el módulo Zigbee para así poder receptar los datos obtenidos mediante los sensores. Además de tener bases previas gracias a los conocimientos adquiridos en la universidad.

4.1.4. Diseño de la base de datos (SQL)

Para el diseño de la base de datos hemos utilizado SQL-SERVER, debido a las características ya antes mencionadas, a continuación detallaremos paso a paso creación de la base de datos:

1. Abrimos SQL-SERVER, y creamos una base de datos llamada: Hospital.
2. Creamos 3 tablas primarias para poder relacionarlas entre sí, las tablas son: Paciente, diagnóstico, cita.

Figura 41. Tablas BD hospital

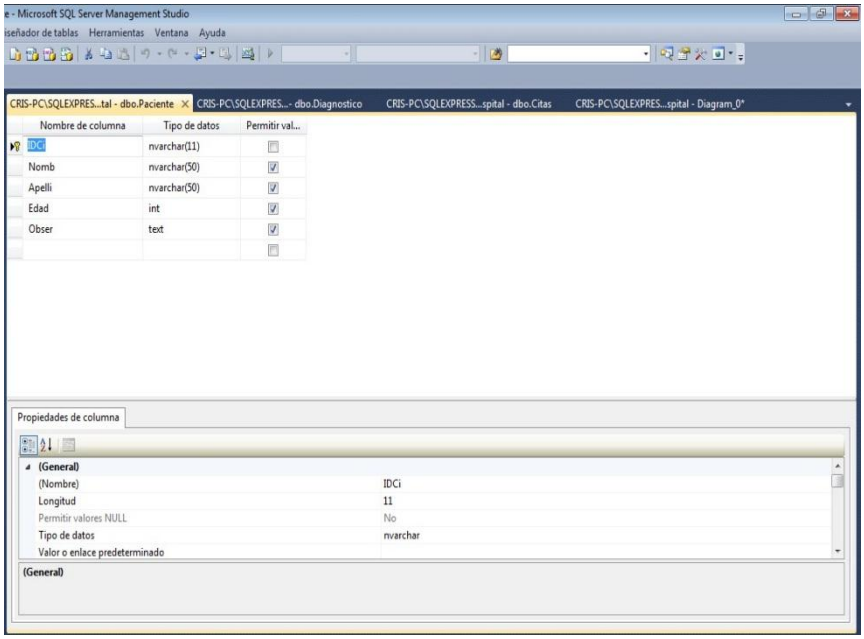


Figura 42. Tabla Paciente y datos

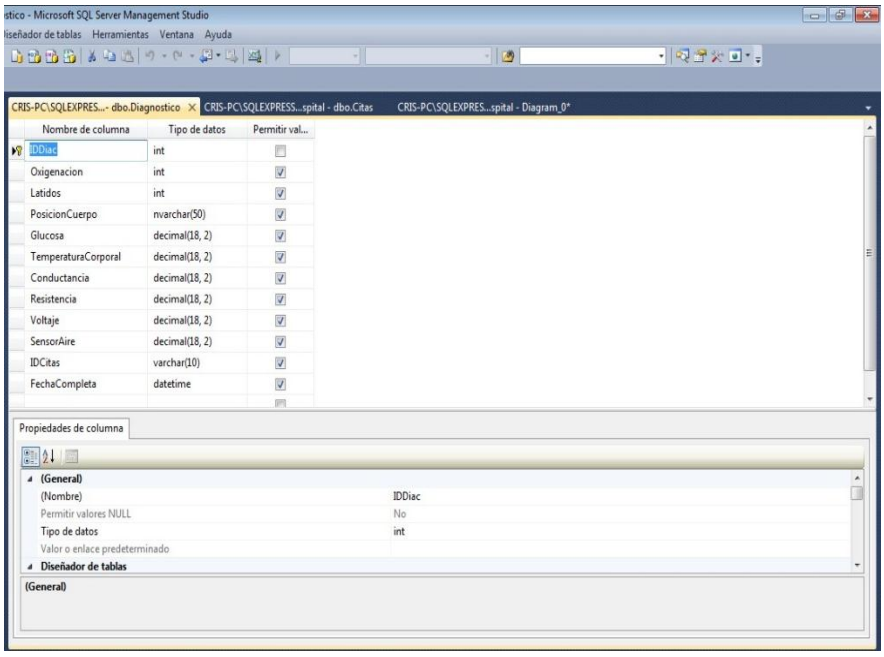


Figura 43. Tabla Diagnóstico y datos.

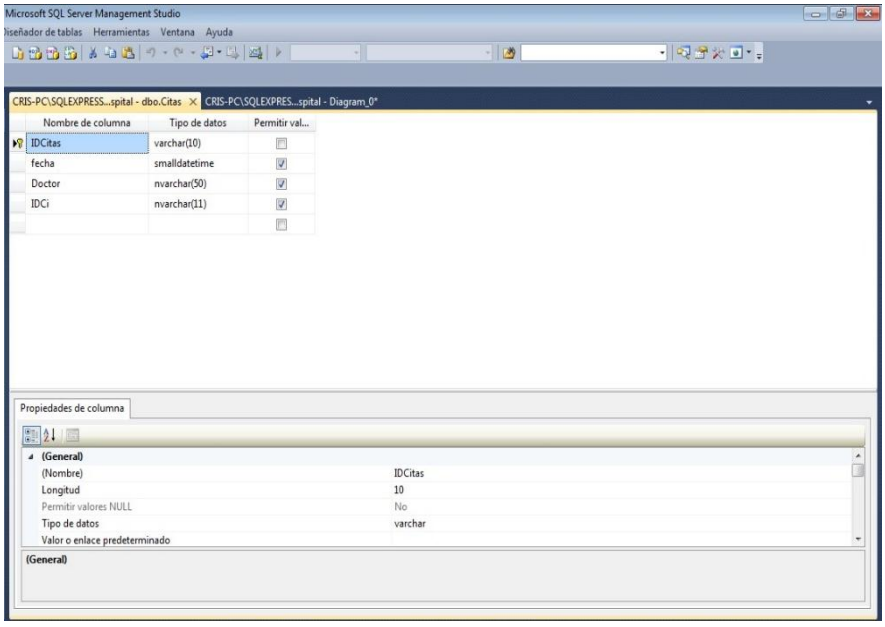


Figura 44. Tabla citas y datos.

3. Una vez creadas nuestras tablas, vamos a crear el diagrama para poder relacionarlas entre sí.

Para crear el diagrama, vamos a la izquierda de la pantalla de visual, debajo de la base de datos creada hay una opción, *crear diagrama de base de datos*, damos click en crear y se mostrará inmediata mente en el siguiente diagrama.

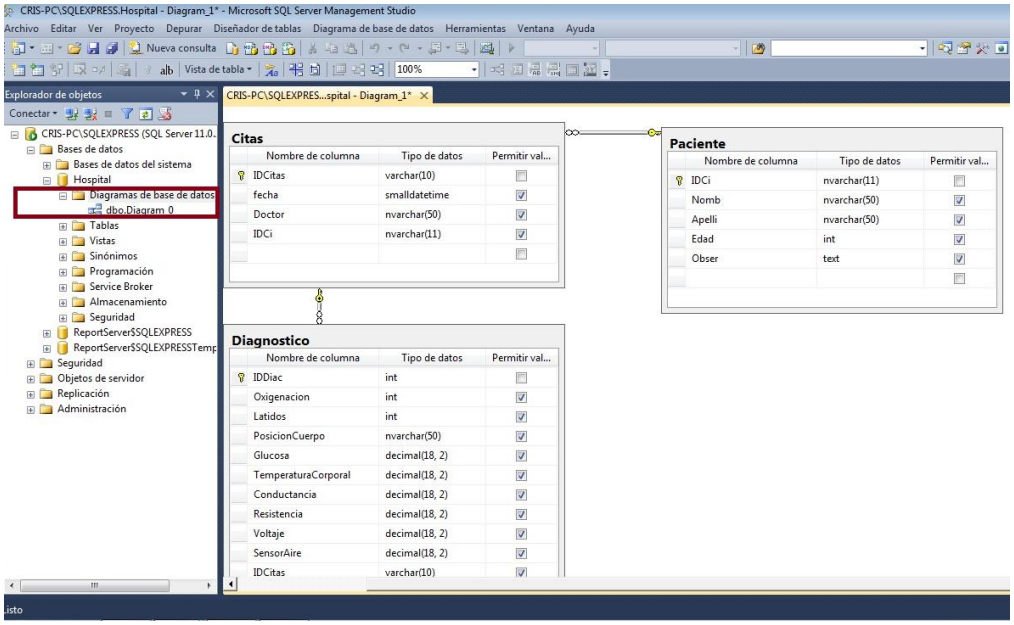


Figura 45. Diagrama de base de datos HOSPITAL.

4. Realizamos las relaciones en base a los siguientes requerimientos:

1 cita puede tener varios Diagnósticos (Relación de uno a muchos)

1 paciente puede tener varias de citas (Relación de uno a muchos)

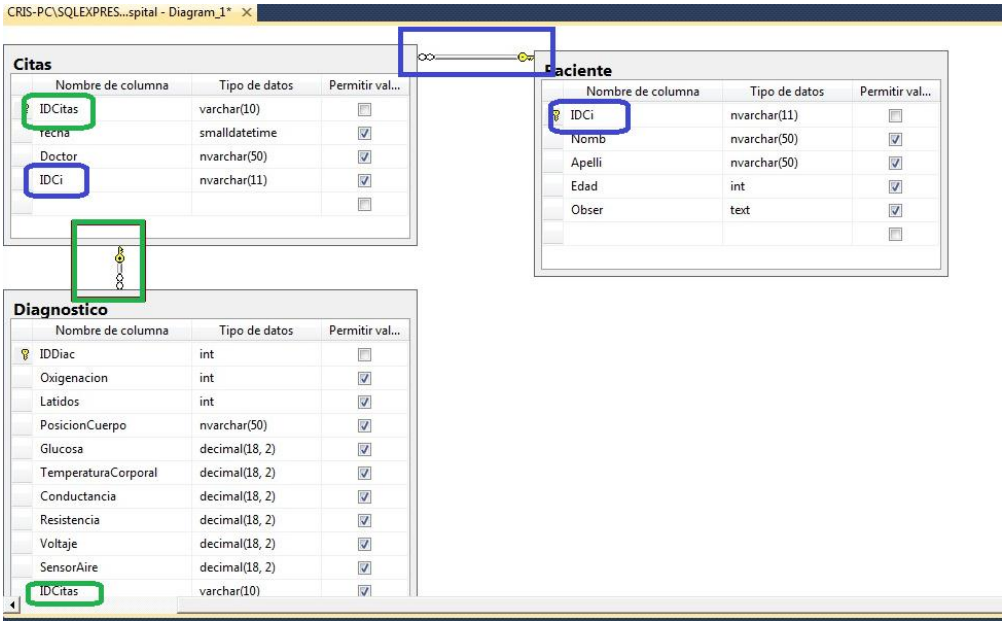


Figura 46. Relaciones de tablas entre sí.

5. Como podemos observar las tablas se relacionan gracias a las variables en común de cada tabla ejm: la tabla CITAS y DIÁGNOSTICO se relacionan mediante la variable **IDCitas** la cual representa un código de cita mientras que las tablas PACIENTE y CITAS se relacionan mediante la variable **IDCi**, la cual es la cédula de identidad del paciente estos dos códigos servirán para la posterior búsqueda de la cita y los signos vitales en el sistema.

4.1.5. Diseño de la interface web (Visual Studio)

Visual Studio ofrece una plantilla para la creación de una página web de manera rápida y sencilla, la cual normalmente tomaría varios días, permitiendo programar únicamente lo necesario.

1. Como primer paso, para la creación de un nuevo proyecto, dentro de la pestaña de Archivo se escoge la opción Nuevo Proyecto, como se indica en la gráfica.

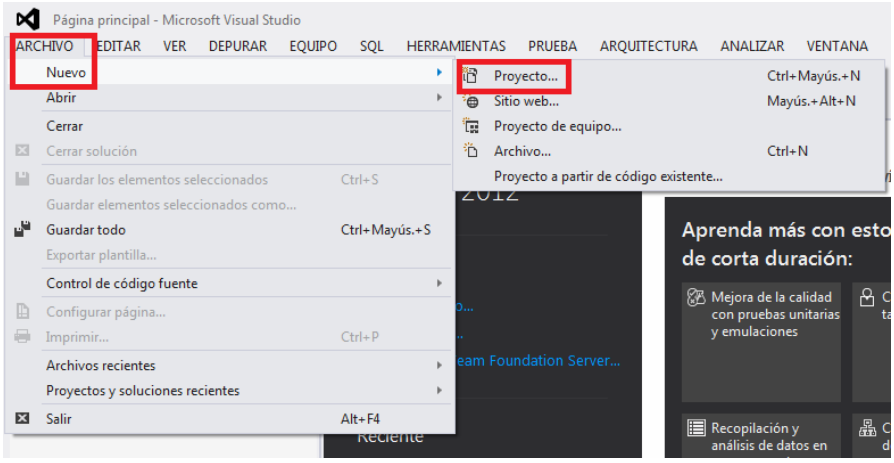


Figura 47. Creación de nuevo Proyecto en Visual Studio

2. Paso seguido se escoge la opción de Aplicación ASP.NET Web Forms, la cual crea una carpeta de tipo CAD (Clase de Acceso a Datos), que permite la conexión con la Base de Datos. Está opción genera una plantilla gráfica.

Se guarda el nuevo proyecto con el nombre deseado.

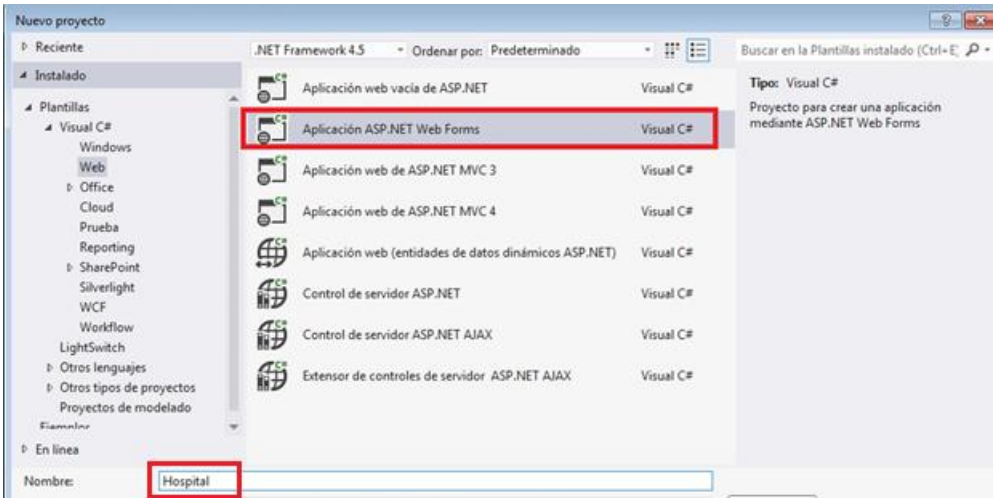


Figura 48. Opción ASP.NET Web Forms

3. Para agregar una página al sitio web, en el Explorador de Soluciones, se da click derecho en el sitio web creado.

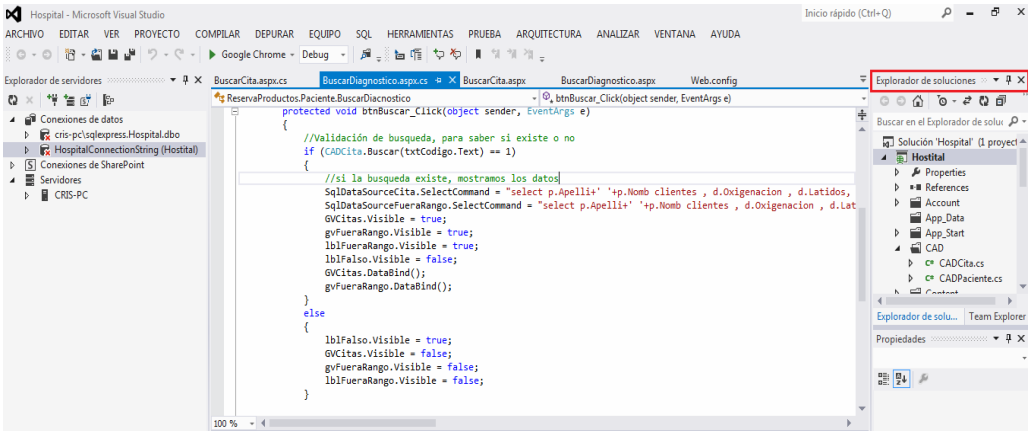


Figura 49. Explorador de Soluciones

4. Se abrirá el cuadro de diálogo, para Agregar un Nuevo Elemento, de donde se escoge la opción de Formulario Web Forms que usa una página maestra, la que permite copiar el código para la creación de páginas futuras.

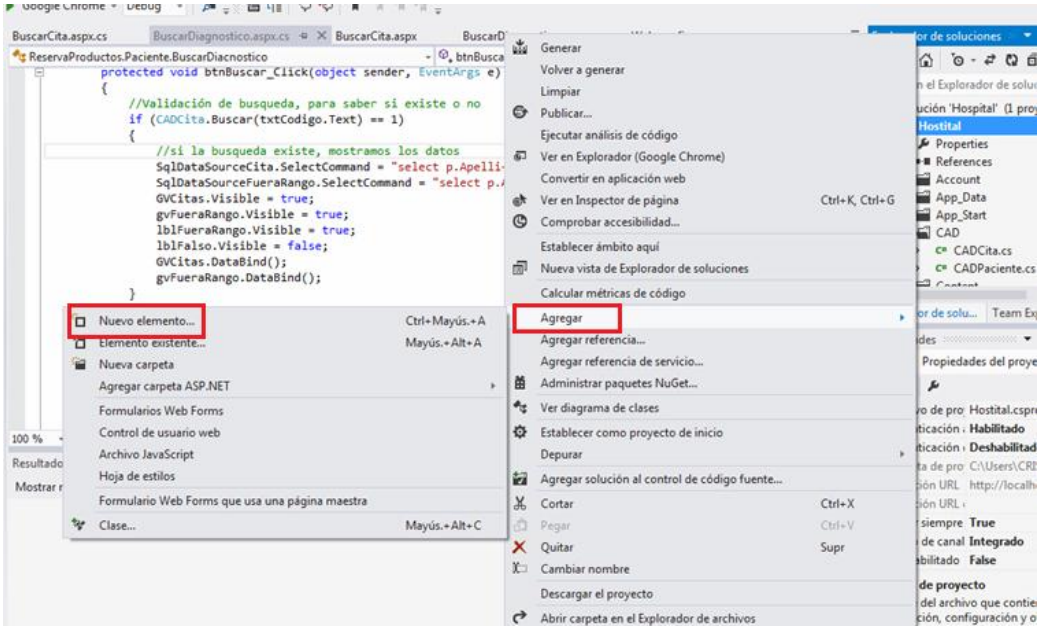


Figura 50. Agregar Nuevo elemento

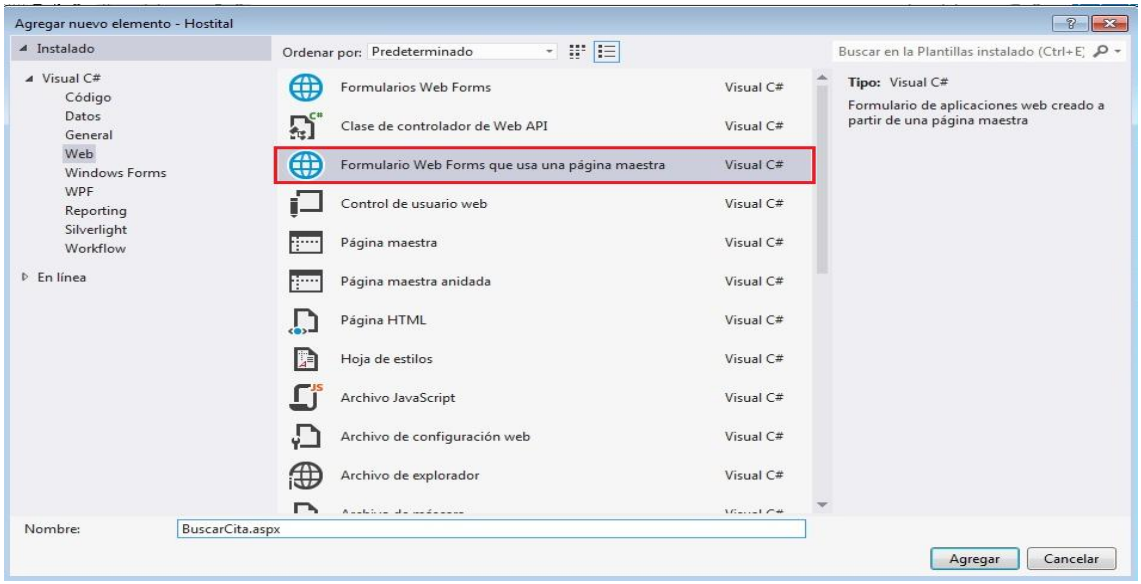


Figura 51. Opción Formulario Web Forms que usa una página maestra

5. Se crea la plantilla maestra (SiteMaster), con el nombre de WebForm1, el cual puede ser cambiado, de acuerdo a los nombres de cada página que se genere.

Esta plantilla permite el arreglo y modificación de su diseño, ayudándose del cuadro de herramientas, que posee todos los elementos necesarios para diseñar las páginas web.

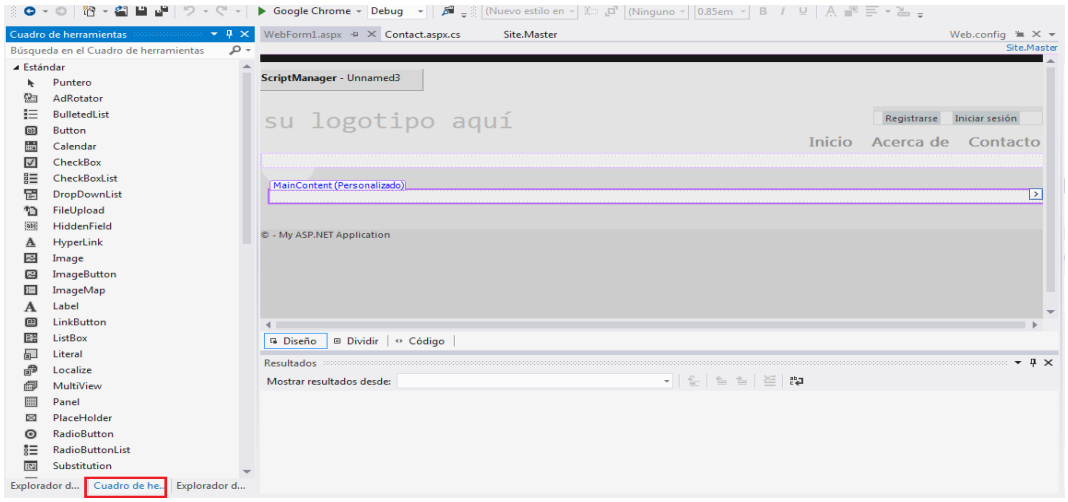


Figura 52. Modificación de Páginas Web

6. Para visualizar los datos obtenidos en la toma de los signos vitales se necesita establecer la conexión con la base de datos, por lo que en el Explorador de Servidores, en la opción de Conexión de datos, se agrega una conexión.

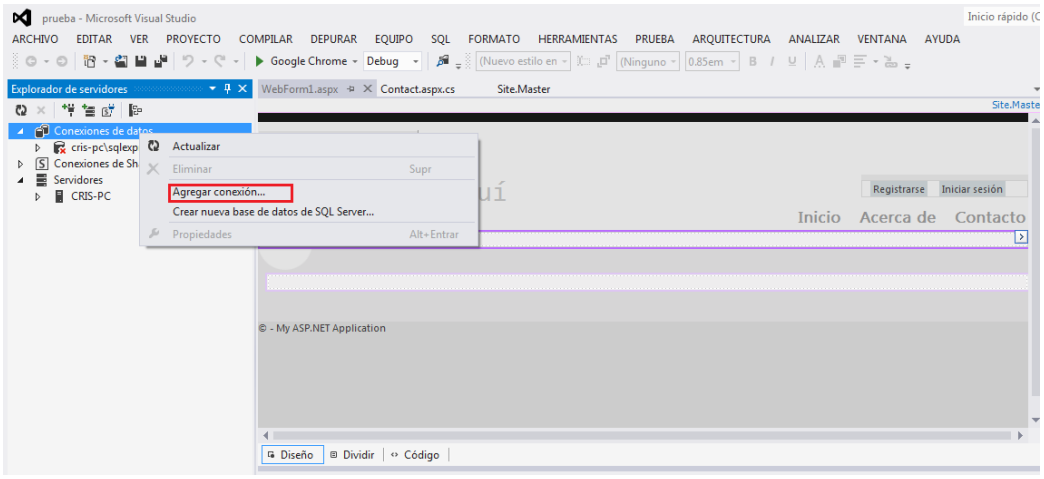


Figura 53. Agregar conexión

7. En la cual escogemos el nombre de nuestro servidor, que es local, así que se copia la ruta del SQL. Para la relación entre la base de datos y la página web, se deben ingresar los valores de Nombre de usuario y contraseña, ya que Java no permite la creación de usuarios, para la validación de datos.

Igualmente se selecciona el nombre de la base de datos con la que se quiere realizar la conexión.

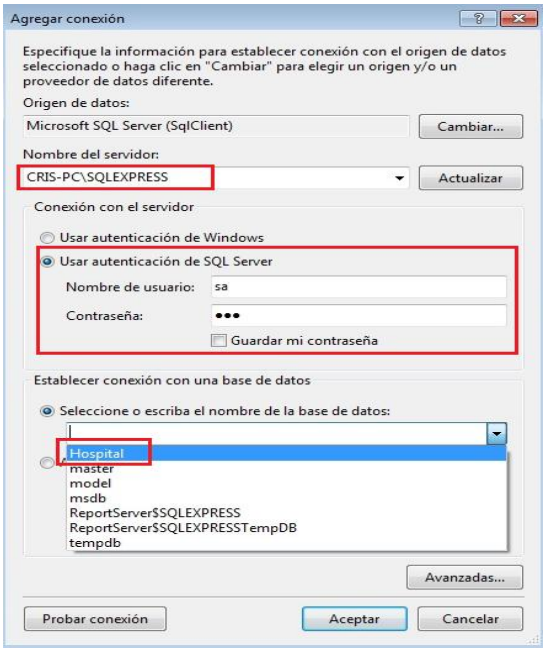


Figura 54. Conexión entre Base de Datos y Servidor Web

8. Para que el diseño de la interfaz web, tenga una apariencia más agradable se debe actualizar el complemento JQuery, que permite la utilización de herramientas de visualización. Para esto dentro de la pestaña Herramientas, se escoge la opción Administrador de paquetes NuGet, y luego la opción Administrar Paquetes NuGet para la Solución, aquí buscamos JQuery y procedemos a instalar.

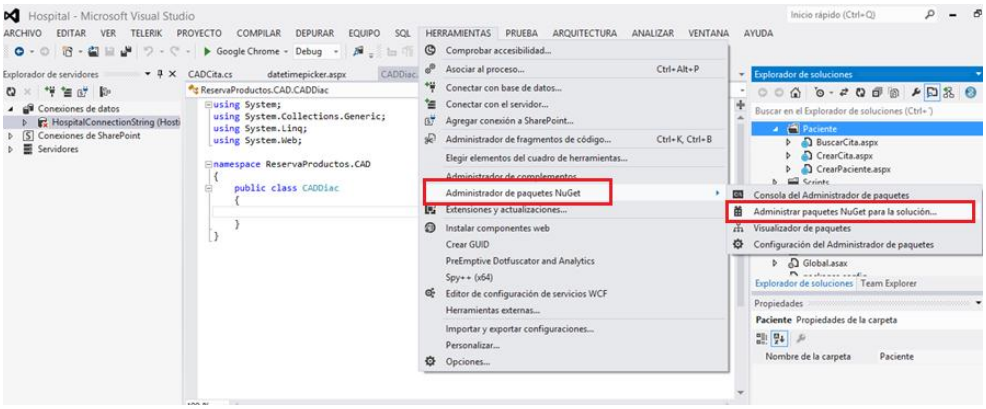


Figura 55. Instalación JQuery

9. Una vez instalado el JQuery, modificamos nuestro proyecto, implementando las referencias necesarias que serán realizadas dentro del SiteMaster. Creamos una plantilla de acuerdo a nuestro gusto, para proporcionarlo a las demás vistas que hemos de crear.

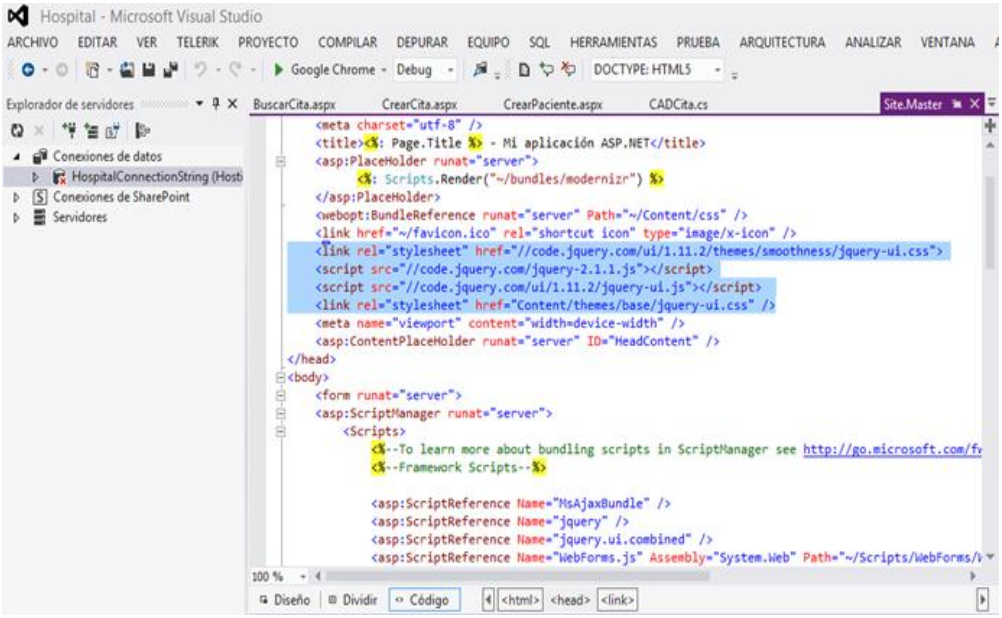


Figura 56. Modificaciones SiteMaster

10. Después de tener el SiteMaster generado, creamos un WebDForm heredada del SiteMaster el cual llamaremos Default.aspx para que sea la página principal al entrar al sistema, el cual va a contener todas las opciones que se pueden realizar para interactuar con los diferentes requerimientos de navegación entre operación y operación.

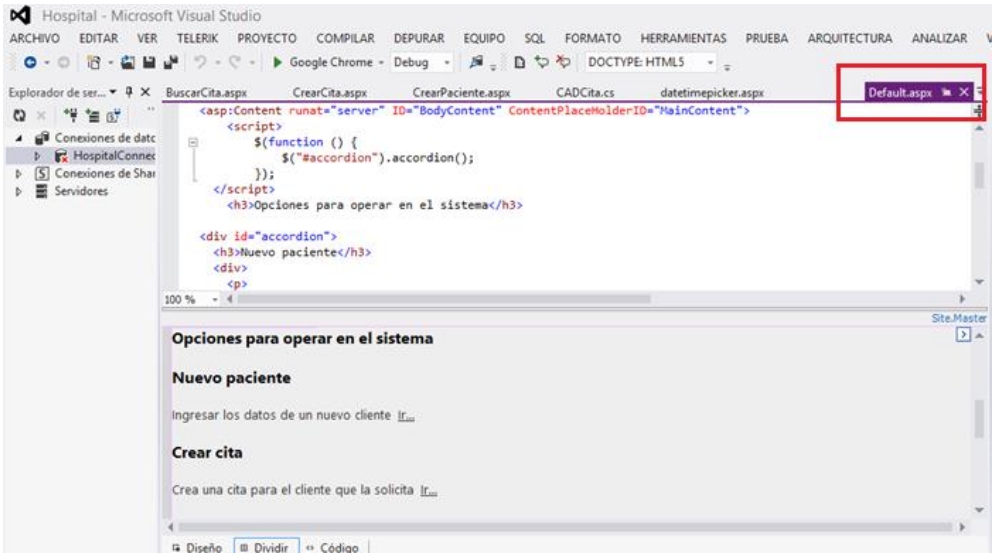


Figura 57. Creación WebDForm

11. Se crean dos carpetas adicionales en el proyecto para organizar las vistas y la conexión hacia la base de datos. Una será nombrada “Paciente”, la cual contendrá todas las vistas del usuario y la otra con el nombre de “CAD” (Clase acceso a datos), que contendrá todas las operaciones posibles para la conexión con la base de datos.

12. En la carpeta CAD, vamos a crear clases, necesarias para conectarnos a la base de datos, según nuestras necesidades, comenzamos con la clase CADPaciente, creamos funciones para las diferentes operaciones que realizamos con la base de datos, implementamos la librería usingSystem.Data.SqlClient, la que permite interactuar con la base de datos de varias maneras según lo requiramos.

13. En la carpeta Paciente creamos un nuevo elemento de WebForm que va hacer heredado del SiteMaster, el cual lo vamos a nombrar como CrearPaciente.aspx. En el comenzamos a crear un formulario para ingresar los datos del paciente, una vez creado el formulario, añadimos un botón guardar, el cual va a tomar todos los datos ingresados, los va a validar y luego que todo se encuentra correcto procederá a guardar en la base de datos, este proceso se realiza gracias a que llamamos a la

clase CADPaciente, la cual tiene una función creada para el ingreso un nuevo paciente.

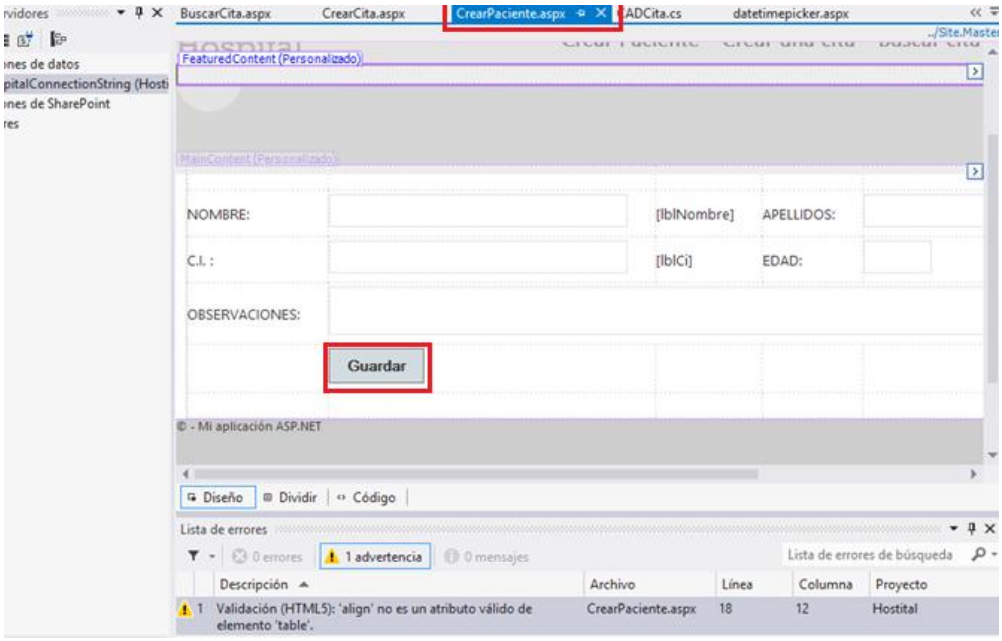


Figura 58. WebForm CrearPaciente

14. En la carpeta Paciente creamos un nuevo elemento de WebForm que va hacer heredado del SiteMaster, el cual lo vamos a nombrar como CrearCita.aspx. En el comenzamos a crear un formulario para ingresar al paciente, la hora de la cita, el código de la cita y además al doctor que va a realizar la intervención, una vez creado el formulario, añadimos un botón guardar, el cual va a tomar todos los datos ingresados, para el campo de hora y fecha implementamos una función en el lado de html que procede de jquery y anidamos con el campo de la hora. Los va a validar y luego que todo se encuentra correcto procede a guardar en la base de datos, este proceso se realiza gracias a que llamamos a la clase CADCita, la cual tiene una función que creamos de ingresar una nuevacita, para poder usar esa clase la llamamos mediante objeto.

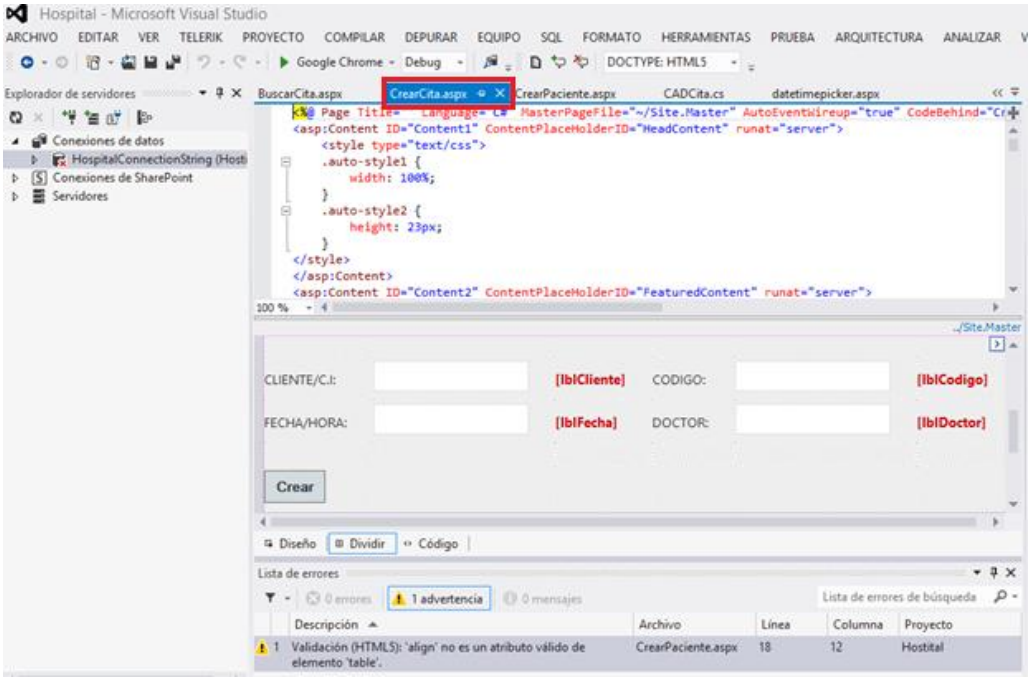


Figura 59. WebForm CrearCita

CAPÍTULO V: IMPLEMENTACION DE LA RED

5.1.Descripción del hardware

En este punto describiremos el soporte hardware en el que se basa este trabajo, las especificaciones y las conexiones reales mediante fotografías de la red implementada.

5.1.1. Equipo host:

Para el buen funcionamiento del sistema a implementar se establece como hardware mínimo el siguiente ordenador:

- Intel Pentium M
- 1,86 GHZ
- 1GB de memoria RAM
- Espacio mínimo de almacenamiento de 1GB.

5.1.2. Arduino UNO

Las especificaciones técnicas del Arduino UNO que se va a utilizar en este trabajo son las siguientes:

- Microcontrolador ATmega328
- Tensión de Funcionamiento (lógica): 5 voltios
- Tensión de alimentación (recomendada): 7 – 12 voltios
- Tensión máxima (no recomendada): 20 voltios
- Pines E/S Digita: 14
- Pines de entrada analógica: 6
- DC I/O Pin: 40 mili amperios
- DC 3.3 voltios Pin: 50 mili amperios

- Memoria flash: 32 KB
- SRAM: 2 KB
- Frecuencia de reloj: 16 MHz
- Comunicación serial UART TTL (5v), RX/TX o USB
- Bootloader: Optiboot
- SPI

5.1.3. Seguridad Escudo E-HEALTH y Sensores

Este escudo nos permitirá la conexión y gestión de los 6 sensores seleccionados con el fin de realizar un monitoreo de signos vitales.

Es de hardware libre, por lo que se tiene plataforma libre para aplicaciones e-salud.

Es importante mencionar que la plataforma E-HEALTH implementa los siguientes mecanismos de protección ya que un punto importante de la gestión de pacientes es la privacidad de los datos:

- **Capa de enlace de comunicación:** Cifrado AES 128 para 802.14.5 Zigbee y WPA2 para WIFI.
- **Capa de aplicación:** Protocolo HTTPS que utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel dependa del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información privada que el HTTP.

5.1.4. Implementación del hardware

Mediante imágenes reales mostraremos a continuación paso a paso, como se realizó la implementación del hardware.

1. Adquirir los implementos necesarios (Arduino, escudo E-Health, módulos Zigbee para emisión y recepción).

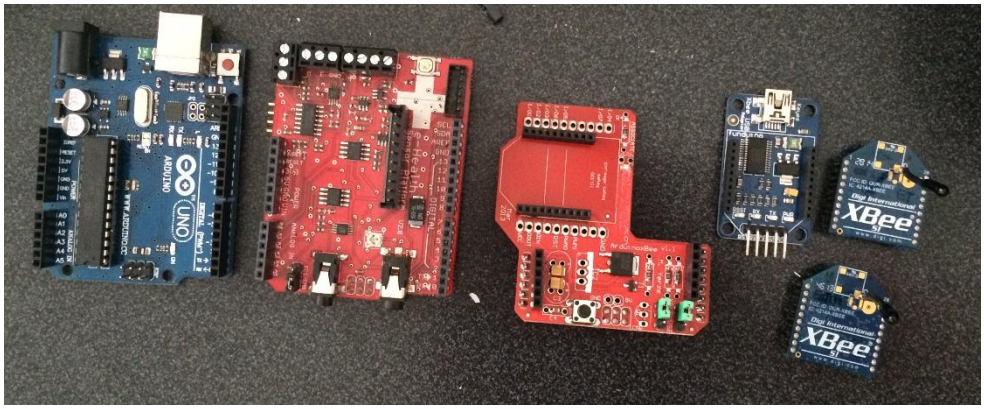


Figura 60. Equipo para la implementación comunicación del nodo sensor.

2. Acoplar el Arduino al escudo E-Health.

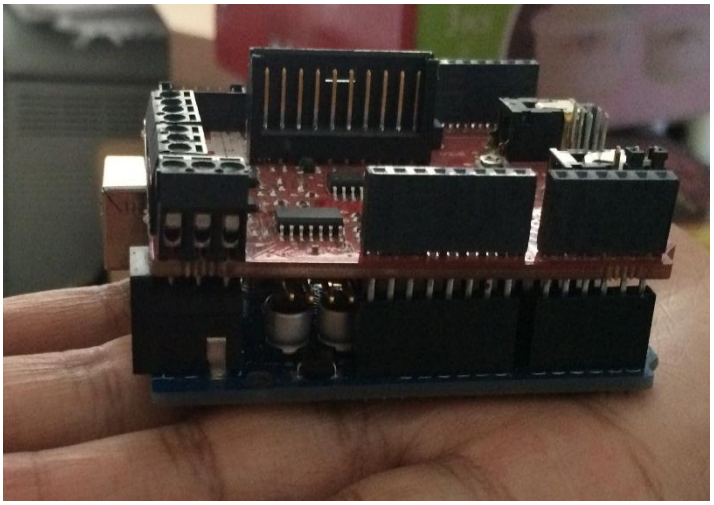


Figura 61. Placa Arduino y Escudo E-Health acoplados.

3. Colocar el módulo Zigbee en la placa para poder adaptarlo al Arduino y escudo E-Health.

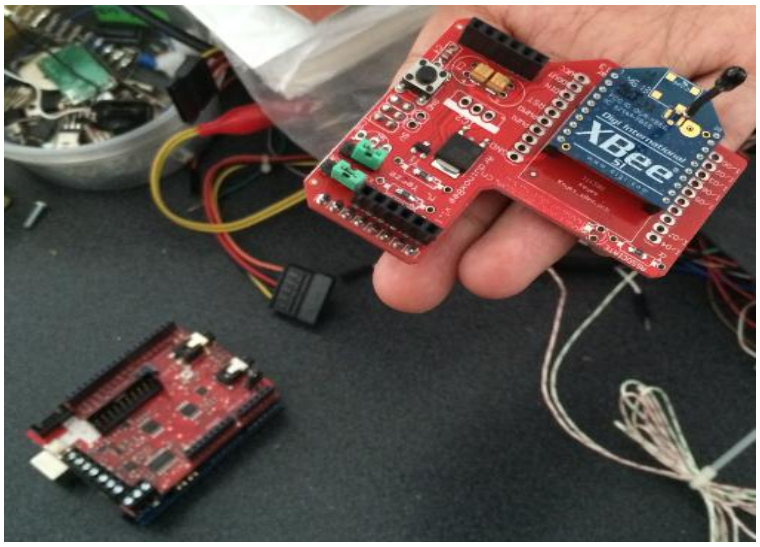


Figura 62. Colocación del módulo Zigbee emisor a la placa

4. Integramos los módulos de los 2 pasos anteriores.

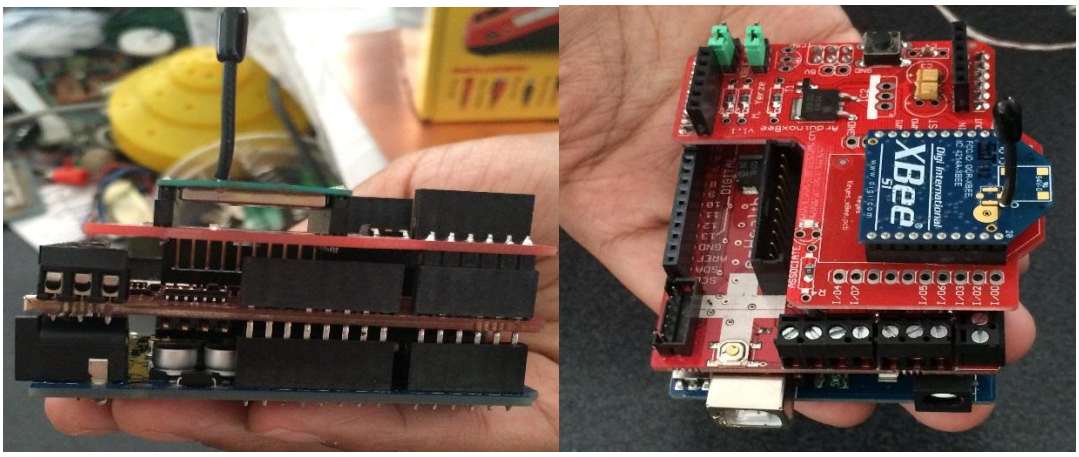


Figura 63. Vista superior y lateral de la estructura básica del nodo sensor.

5. Conectamos el módulo Zigbee receptor a la placa que nos permitirá comunicarnos alámbricamente a la computadora.

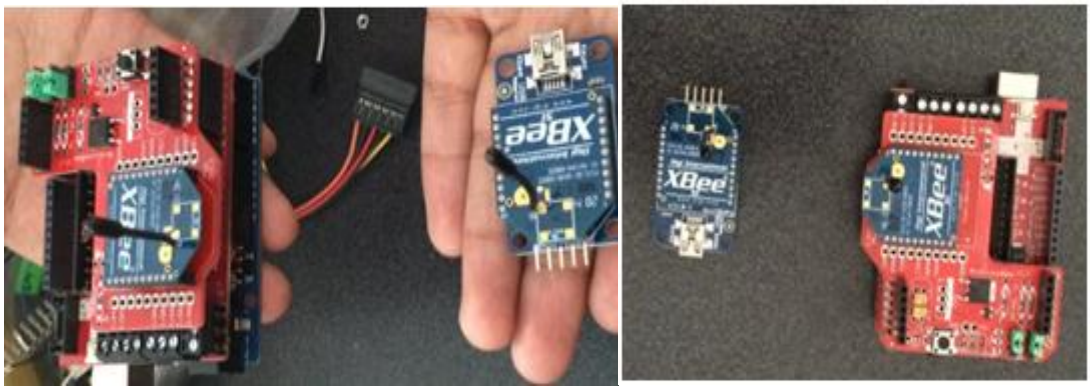


Figura 64. Vista superior y lateral de la estructura nodo sensor más módulo Zigbee receptor.

6. Integramos los sensores según lo detallamos en capítulos anteriores:

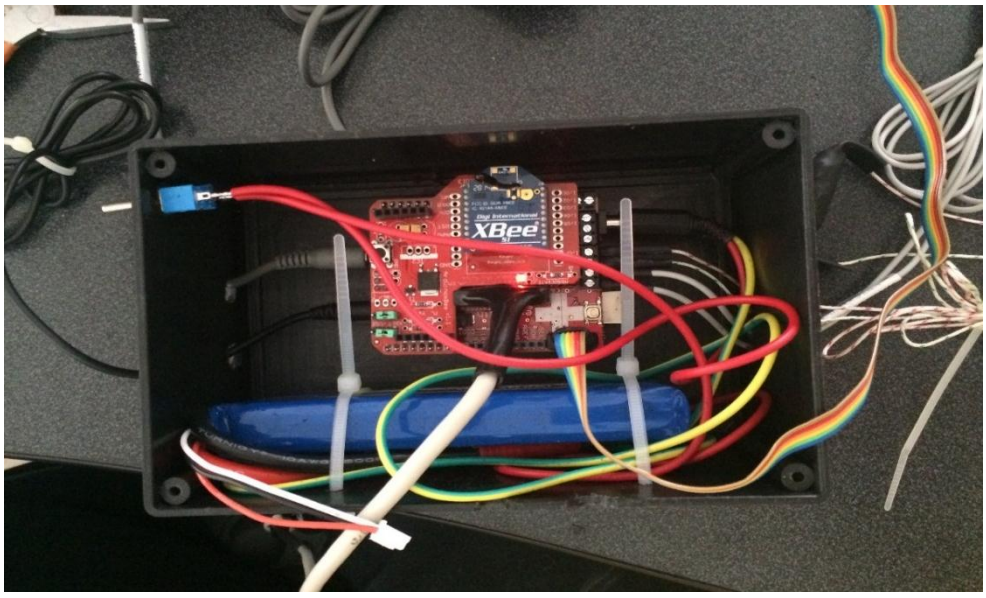


Figura 65. Nodo sensor completo e integrado en caja.

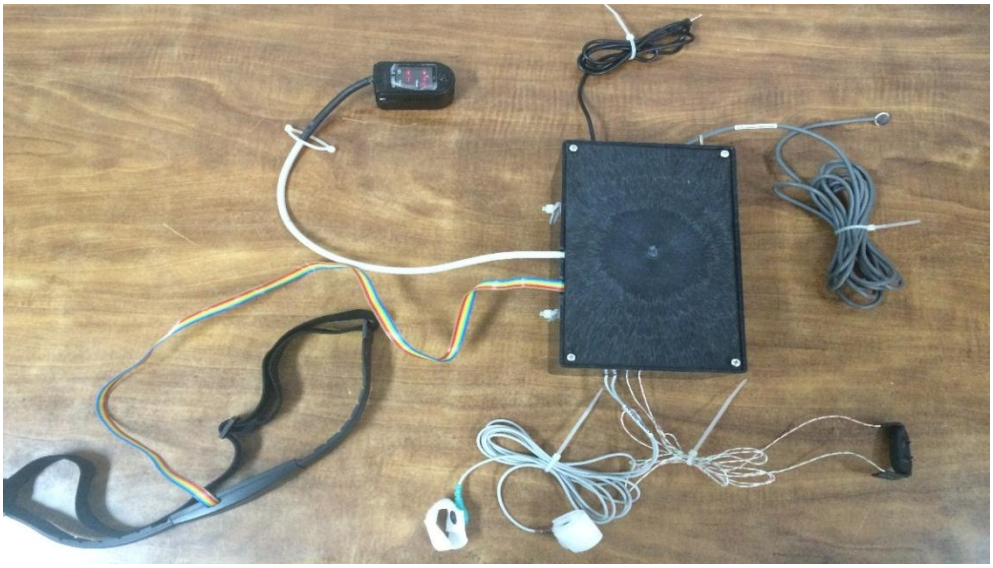


Figura 66. Nodo sensor terminado_ vista final.

7. Conectamos mediante cable el módulo zigbee receptor al computador en el cual se encuentra la interfaz web y base de datos.

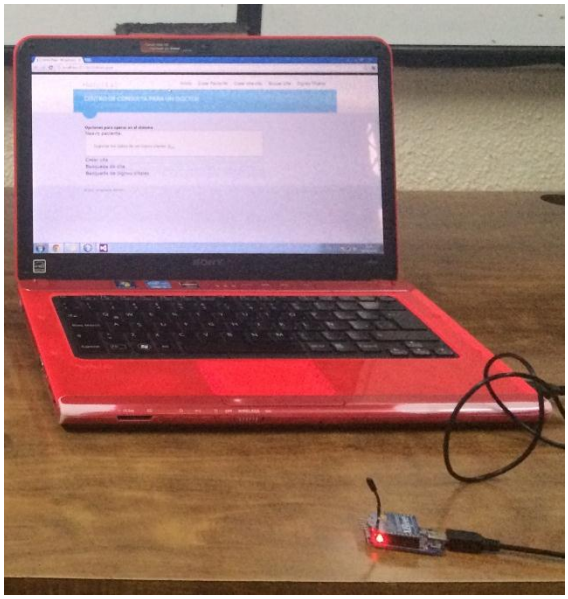


Figura 67. Computador con módulo Zigbee receptor (“servidor”).

5.2.Descripción del software:

A continuación describiremos el soporte software sobre el que se desarrolló este proyecto, se presenta los drivers necesarios para controlar el hardware y las herramientas para el diseño de la interfaz.

- **Arduino IDE:** Entorno integrado de desarrollo para la programación de la plataforma Arduino, Proporciona un editor de texto para la programación y también permite seleccionar la forma de comunicación entre el equipo programador y el dispositivo programado, además de la compilación y carga del programa en el microcontrolador de la plataforma Arduino, cabe mencionar que mediante el puerto serie se puede programar cualquier escudo compatible con la plataforma Arduino, El lenguaje está basado en ***WIRING***, sintaxis similar al C++ sin implementar herencia múltiple ni ***templates***. El usuario debe definir al menos dos funciones setup() y loop().

- **NetBeans IDE 7.3.1:** Entorno de desarrollo utilizado para la programación y compilación de todo el sistema.

- **RXTX:**Es una librería que utiliza una implementación nativa (a través de JNI), que proporciona comunicación serie y paralelo para el kit de herramientas de JDK, se basa en la especificación ***SUN JAVA COMUNICATIONS API,*** sin embargo Java no proporciona sopporte para Windows, ese es el motivo de utilizar RXTX, A diferencia de otras librerías para lectura y escritura del puerto serie, tiene licencia GPI.

- **Java Development Kit:**kit de herramientas para el desarrollo de programas en el lenguaje de programación orientada a objetos.

5.2.1. Programa ejecutado en arduino

A continuación se indica el código programado en el Arduino

1. Para realizar la configuración de los sensores es importante contar con las librerías eHealth.h que permite leer y enviar la información de los sensores utilizando las interfaces de radio disponibles, así también se usa la librería PinChangeInt.h, que es necesaria solo cuando se utiliza el sensor de pulso y oxigenación en la sangre.

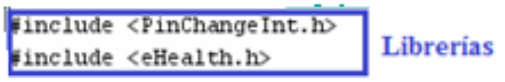


Figura 68. Librerías eHealth.h y PinChangeInt.h
Fuente (Cooking Hacks.2013)

2. Paso seguido se configura la velocidad de los datos de la placa Arduino, que en este caso es de 9600 Baudios. Además se inicializan los sensores de la glucosa, pulso y oxigenación de la sangre y el sensor de posición.

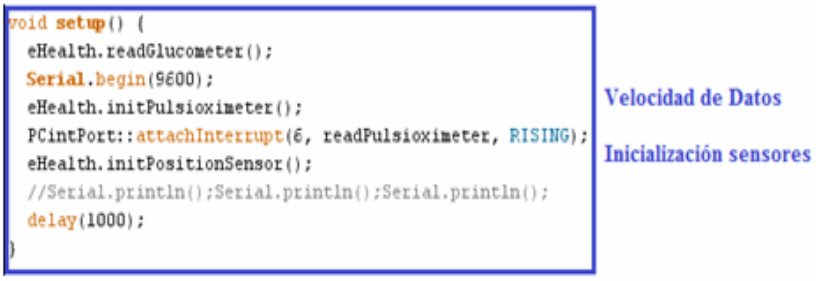


Figura 69. Velocidad de Datos, Inicialización de sensores
Fuente (Cooking Hacks.2013)

3. En la ejecución del programa void loop (), se empieza con la captura de los datos de los sensores, los cuales se configuran con un tiempo de retardo de 1000 ms.

```
void loop() {
  sensor_dedo();
  on_off_pinChange();
  posicion_cuerpo();
  glucosa();
  teperatura_cuerpo();
  GSR_sensor();
  sensor_aire();
  contreset++;
  if(contreset>=20){
    asm volatile ("  jmp 0");
  }
  delay(1000);
```

Figura 70. Captura de Datos de Sensores
Fuente (Cooking Hacks.2013)

4. Para leer el sensor de pulso y oxigenación en la sangre se establece un contador de 50 muestras, para reducir la latencia y se utiliza la función eHealth.readPulsioximeter.

Para obtener los datos de pulso y oxigenación del sensor se utilizan las variables de eHealth.getBPM e eHealth.OxygenSaturation respectivamente.

```
void readPulsioximeter(){
  cont ++;
  if (cont == 50) { //Get only of one 50 measures to reduce the latency
    eHealth.readPulsioximeter();
    cont = 0;
  }
}
void sensor_dedo(){
  if(eHealth.getBPM()>0){
    for(int x=0;x<10;x++){
      rbpm_acu+=eHealth.getBPM();
      spo2_acu+=eHealth.getOxygenSaturation();
    }
```

Figura 71. Funciones Sensor Pulso y Oxigenación en la sangre
Fuente (Cooking Hacks.2013)

5. Para el sensor de flujo de aire, la función utilizada es eHealth.getAirFlow, que permite imprimir el valor de la respiración.

```
void sensor_aire(){
  int air = eHealth.getAirFlow();
  //Serial.print(F("Airflow analogic value : "));
  Serial.println((String)air);
```

Figura 72. Funciones Sensor Flujo de Aire
Fuente (Cooking Hacks.2013)

6. La función usada para el sensor de temperatura corporal, que retorna los valores medidos es eHealth.getTemperature.

```
void teperatura_cuerpo(){
  float temperature = eHealth.getTemperature();
  //Serial.print("Temperature (°C): ");
  Serial.print(temperature, 2);
  Serial.print("+");
  //delay(100);
}
```

Figura 73. Función Sensor de Temperatura Corporal
Fuente (Cooking Hacks.2013)

7. Las funciones utilizadas para obtener los valores medidos de la posición, así como la impresión de los mismos son eHealth.getBodyPosition y eHealth.printPosition.

```
void posicion_cuerpo(){

  //delay(1000);
  //Serial.print("Current position : ");
  uint8_t position = eHealth.getBodyPosition();
  eHealth.printPosition(position);
  aux1+=position;
  Serial.print("+");
  //delay(100);
}
```

Figura 74. Funciones Sensor de Posición
Fuente (Cooking Hacks.2013)

8. El sensor de respuesta galvánica de la piel, usa dos funciones, para devolver los valores de resistencia y conductancia de la piel, las cuales son eHealth.getSkinResistance e eHealth.getSkinResistance

```
void GSR_sensor(){
 float conductance = eHealth.getSkinConductance();
  float resistance = eHealth.getSkinResistance();
  float conductanceVol = eHealth.getSkinConductanceVoltage();
```

Figura 75. Funciones Sensor de Respuesta Galvánica de la piel
Fuente (Cooking Hacks.2013)

9. El sensor de glucosa, opera con la función uint8_t numberOfData, que lee la cantidad de datos obtenidos.

```
void glucosa(){
  uint8_t numberOfData = 1;//eHealth.getGlucometerLength();
  //Serial.print(F("Number of measures : "));
  //Serial.println(numberOfData, DEC);
  //delay(100);
```

Figura 76. Función Sensor de Glucosa
Fuente (Cooking Hacks.2013)

5.2.2. Red inalámbrica

Para la red inalámbrica hemos decidido utilizar tecnología Zigbee, por lo cual usaremos MODULOS ZIGBEE UNO, Para comenzar a configurar estos módulos se tiene que descargar el software X-CTU de los mismos desarrolladores de los módulos X-BEE, que sirve para configurarlos de forma visual o en su defecto el desarrollador puede proceder al uso de comandos AT para realizar las mismas configuraciones sin tener que descargarse dicha aplicación.

5.2.2.1.Configuración

Para la configuración usamos el software X – CTU:

Tabla 4. ***Tabla de Configuraciones***

	Xbee-Envia	Xbee-Recibe
PAN ID	9032	9032
DH	0	0
DL	1	0
MY	0	1

Módulo Emisor (E):

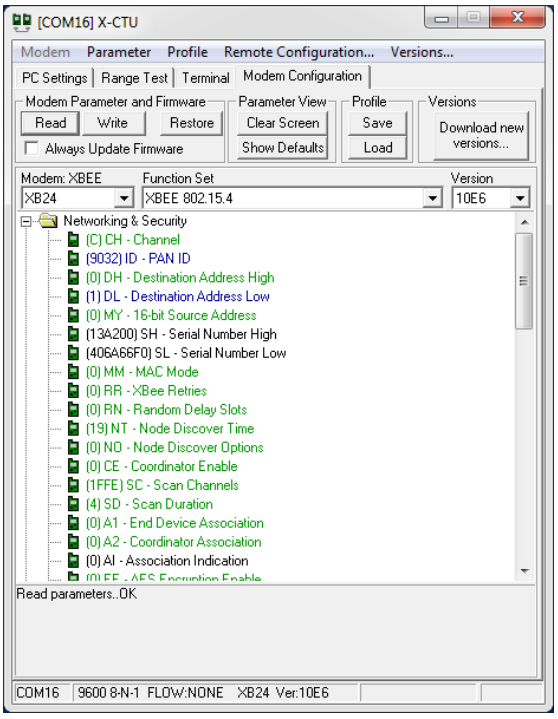


Figura 77.Pantalla Modulo emisor

Módulo Receptor(R):

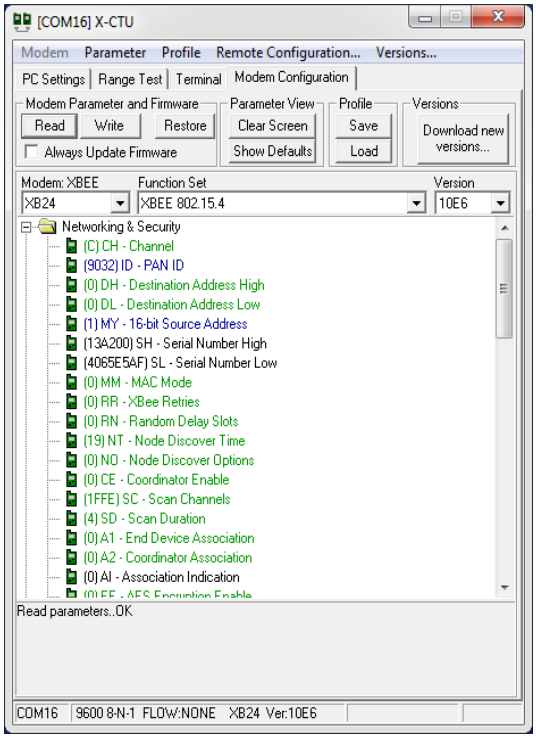


Figura 78.Pantalla modulo receptor.

Como se puede apreciar en las imágenes anteriores tanto para el emisor como para el modulo receptor, se tienen pintadas de color azul las direcciones tanto de origen y destino para realizar la comunicación serial entre los dispositivos.

5.2.3. Arquitectura

Se ha empleado una arquitectura cliente servidor, de manera que se tiene un programa de bajo nivel en el microcontrolador de Arduino que se encarga de leer los datos de los sensores y enviarlos al servidor. El servidor recibe datos por parte de los sensores y los almacena en la base de datos. El cliente en el entorno web se comunica con el servidor para hacer peticiones de lectura y por ultimo base de datos donde se almacenan los datos provenientes de los sensores.

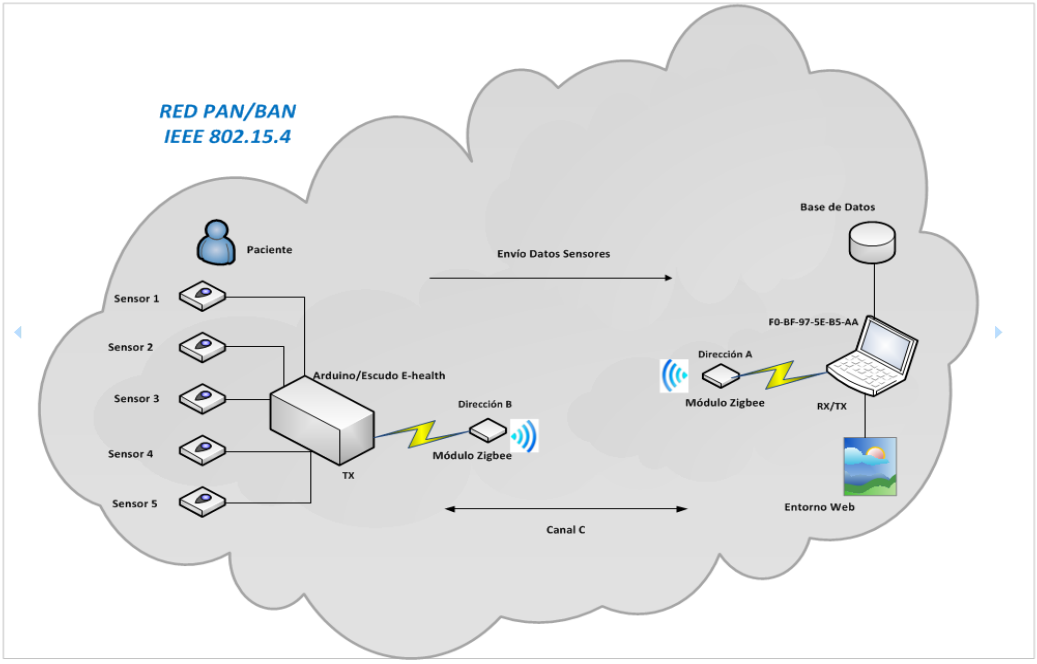


Figura 79. Diseño de la red implementada

5.2.3.1.Montaje del prototipo

A continuación mostraremos una fotografía del prototipo implementado, a la derecha se encuentra el computador que actúa como servidor junto con el módulo Zigbee el cual recepta los datos tomados por los sensores, mientras que a la izquierda se encuentra el nodo sensor compuesto por el escudo E-health, Arduino, módulo Zigbee emisor, batería y 6 sensores fijados en una caja por cuestiones de presentación y movilidad.

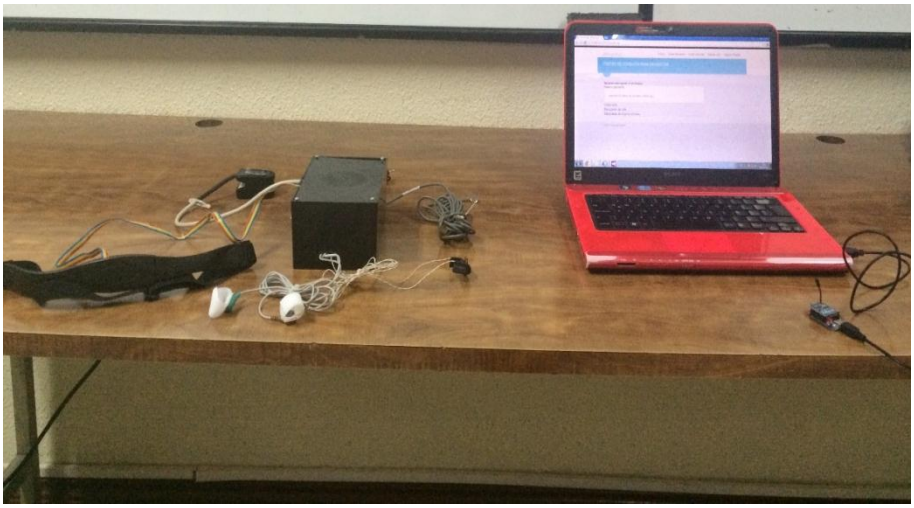


Figura 80. Imagen real del Prototipo realizado.

5.2.4. Conexión con base de datos

Para la conexión de la base de datos con el escudo E-Health utilizamos Java por cuestiones de compatibilidad, a continuación mostraremos a breves rasgos los pasos para realizar la conexión.

*Java precisa especificar mediante qué puerto se va a comunicar, dicha información la obtendremos ingresando a:**sql manager => protocolos SQL EXPRESS =>tcp/ip => direcciones ip=>puertos dinámicos (1029).***

1. Para poder utilizar la librería Arduino es fundamental como paso previo copiar la librería de rxtx en la ruta **C:\Program Files\Java\jre1.8.0_20\bin** en el jdk.

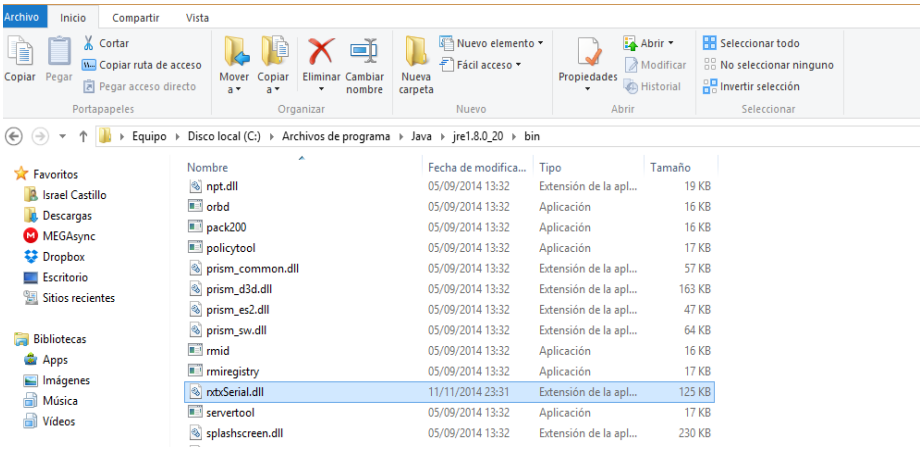


Figura 81. Insertando librería RXTX

2. Creamos un nuevo proyecto en java llamado Proyecto Arduino.

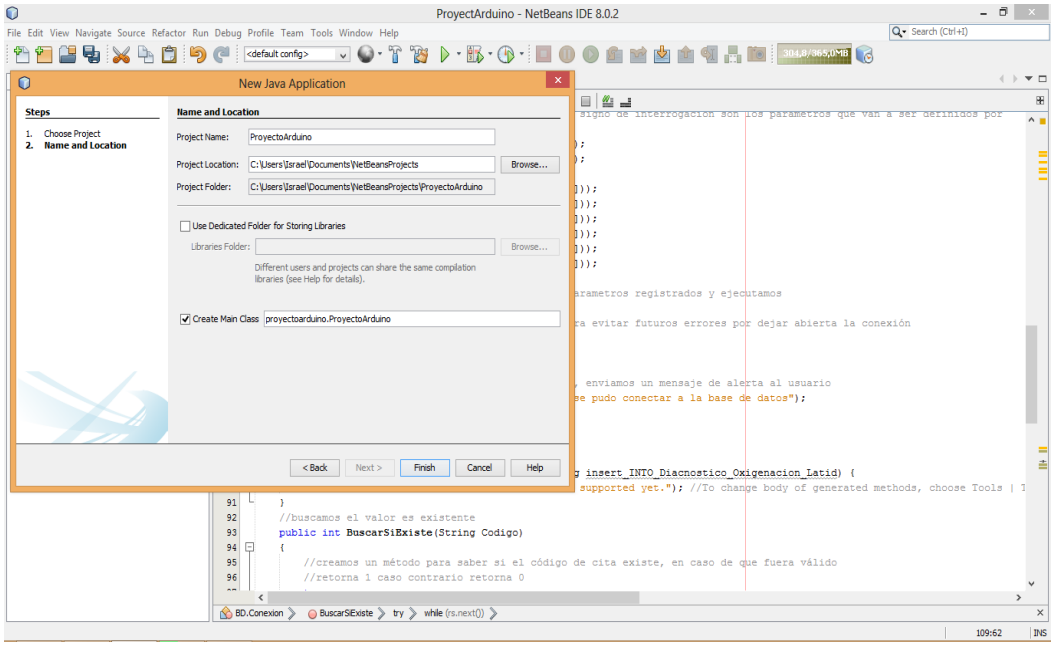


Figura 82. Creando nuevo proyecto

3. Importamos las librerías necesarias que nos permitirán utilizar los componentes necesarios, para que corra nuestro programa de manera eficiente, como:

- **Arduino,rtxt, jcommon** .- Estas librerías contienen el hilo que va a estar preguntando constantemente a nuestros sensores si están enviando datos.
- **Jfreechar.-** Esta librería nos ayuda a mostrar las gráficas según vaya recibiendo datos desde el Arduino.
- **Sqljdbc4.-** Esta librería ayuda a realizar la conexión con la BD (SQL Server.)

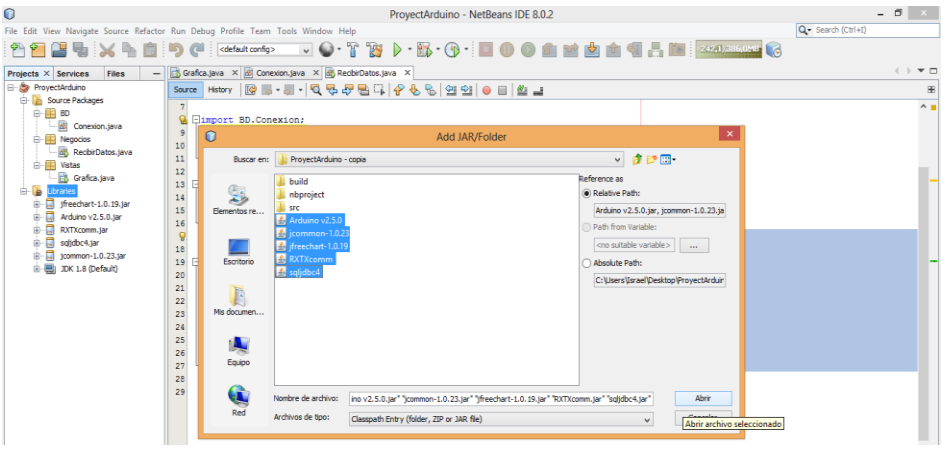


Figura 83. Añadiendo librerías necesarias.

4. Creamos 3 paquetes llamados: BD, negocio y Vistas.

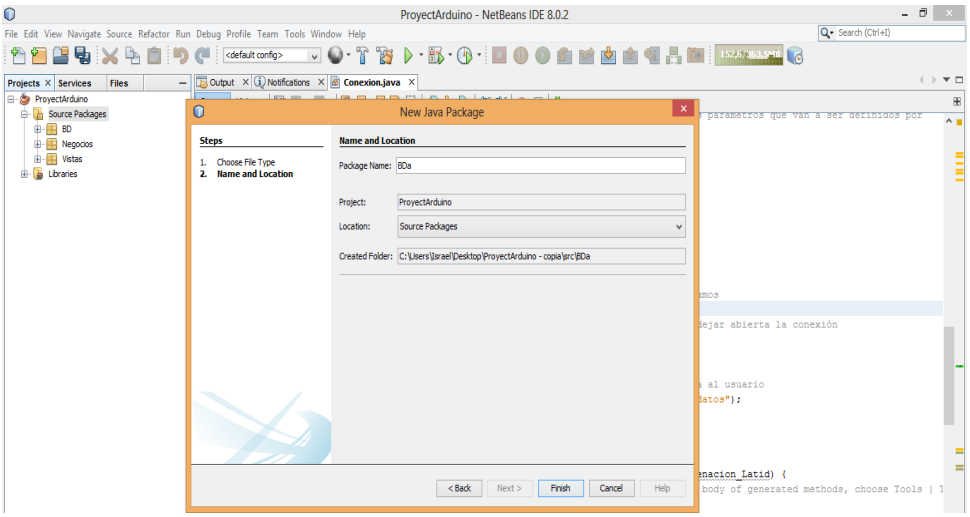


Figura 84. Creando paquetes necesarios.

5. En el paquete BD vamos a ingresar todo referente a la base de datos como conexiones, consultas e inserciones:

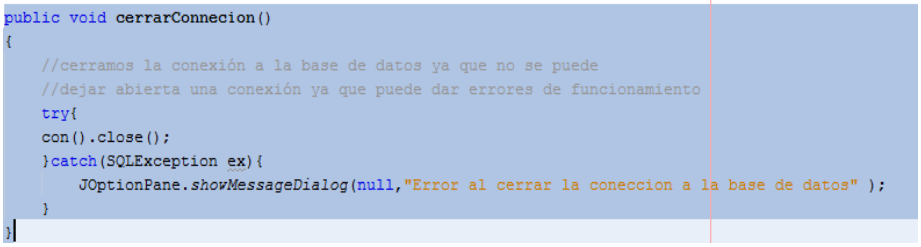
Creamos el método de conexión a la BD al cual pasaremos los parámetros necesarios para establecerla. Ejemplo: nombre de la BD, nombre del servidor, nombre del usuario y contraseña, y finalmente pero no menos importante el puerto establecido en la BD

```
public  Connection con()
{
   Connection con = null;
   try
   {
       //mandamos parametros necesarios para establecer conexión con la base de datos
   SQLServerDataSource ds = new SQLServerDataSource();
   ds.setUser("sa");
   ds.setPassword("123");
   ds.setServerName("CRIS-PC\\SQLEXPRESS");
   ds.setPortNumber(1029);
   ds.setDatabaseName("Hospital");
   con= ds.getConnection();
   }catch(Exception e)
   {
       //en caso de no conectarse o que la base este desahabilitada mandar un mensaje de error
        JOptionPane.showMessageDialog(null,"Error al conectar a la base de datos" );
   }
   //retornamos el pull de conexion para poder realizar todo tipo de consultas a la base de datos apuntada
   return con;
}
```

Figura 85. Parámetros necesarios para BD para el pull de conexión.

Si todos los parámetros son correctos se establecerá y utilizará la conexión sin problema, caso contrario se va a desplegar un mensaje de advertencia la misma indicara que no se pudo establecer la conexión con la BD.

6. Configuramos el cierre de la conexión ya que si se deja abierta, esta puede crear conflictos con la BD, la misma que podría ocasionar fallos hasta llegar al punto de reiniciarla y perder las últimas consultas realizadas.



```
public void cerrarConnecion()
{
    //cerramos la conexión a la base de datos ya que no se puede
    //dejar abierta una conexión ya que puede dar errores de funcionamiento
    try{
    con().close();
    }catch(SQLException ex){
        JOptionPane.showMessageDialog(null,"Error al cerrar la coneccion a la base de datos" );
    }
}
```

Figura 86. Estableciendo cierre de conexión.

7. Realizamos la inserción de los valores enviados de Arduino y este a su vez hace la inserción a la BD.

```
public void saveOperation(String Datos[], String codigo)
{

    try
    { //llamamos al método de llamar a la conexión luego creamos la consulta sql para ingresar datos a la bd
        stmt = con().prepareStatement("Insert INTO Diagnostico "
        + "(Oxigenacion, Latidos, PosicionCuerpo, Glucosa,"
        + " TemperaturaCorporal, Conductancia, Resistencia, "
        + "Voltaje, SensorAire, IDCitas, FechaCompleta) Values (?,?,?,?,?,?,?,?,?,?, GETDATE())");
        //todos los parametros diferenciados por signo de interrogación son los parametros que van a ser definidos por
        // usuario
        stmt.setInt(1,Integer.parseInt(Datos[0]));
        stmt.setInt(2,Integer.parseInt(Datos[1]));
        stmt.setString(3,Datos[2]);
        stmt.setDouble(4, Double.valueOf(Datos[3]));
        stmt.setDouble(5, Double.valueOf(Datos[4]));
        stmt.setDouble(6, Double.valueOf(Datos[5]));
        stmt.setDouble(7, Double.valueOf(Datos[6]));
        stmt.setDouble(8, Double.valueOf(Datos[7]));
        stmt.setDouble(9, Double.valueOf(Datos[8]));
        stmt.setString(10,codigo);
        //ejecutamos la consulta con todos los parametros registrados y ejecutamos
        stmt.executeUpdate();
        // llamamos al método cerrar conexión para evitar futuros errores por dejar abierta la conexión
        cerrarConnecion();
    }catch(SQLException ex)
    {
        //en caso de no poderse conectar a la bd, enviamos un mensaje de alerta al usuario
        JOptionPane.showMessageDialog(null," No se pudo conectar a la base de datos");
    }
}
```

Figura 87. Creando método llamado conexión.

8. Dentro del try catch realizamos llamamos al método conexión y luego le mandamos la consulta de sql para la inserción, nombramos los campos tales y cuales como se llaman en la tabla de la BD, en los campos “values” están definidos con signo de interrogación lo cual significa que va a recibir valores por arduino. Luego de definir todos los valores recibidos realizamos la ejecución de la sententica sql, con “executeUdpade()”

9. Si se ha establecido la conexión y los parámetros son los correctos, la inserción debería ser todo un éxito, caso contrario mandaría un mensaje de que no se pudo insertar los valores a la base de datos.

Método buscar, este nos sirve para realizar una búsqueda referente al código de la cita, si el código existe retorna un valor igual a 1, caso contrario retornaría 0.

```
public int BuscarSiExiste(String Codigo)
{
    //creamos un método para saber si el código de cita existe, en caso de que
    //retorna 1 caso contrario retorna 0
    try
    {
        stmt = con().prepareStatement("select 1 verdad\n" +
        "  from Citas\n" +
        "  where IDCitas = ?");
        stmt.setString(1,Codigo);
        stmt.execute();
        //recorremos la tupla obtenida por la consulta realizada
        ResultSet rs = stmt.getResultSet();
        while(rs.next()){
            //leemos el encabezado de la consulta la cual se llama "verdad"
            //luego obtenemos el valor 1 si es que la cita existe
            //se almacena en la variable llamada "Verdad"
        Verdad = Integer.parseInt(rs.getString("verdad"));
        }
        rs.close();
        if(Verdad == 1)
        {
            cerrarConnecion();
            Verdad = 0;
            return 1;
        }
        else
        {
            cerrarConnecion();
            Verdad = 0;
            return 0;
        }
```

Figura 88. Método Buscar

El objeto llamado ResultSet sirve para retornar el número de filas realizadas por la consulta previa, luego capturamos el valor retornado en una variable para realizar las comparaciones, si el valor es igual a uno retorna 1 y cierra la conexión, si el valor es lo contrario retorna 0 y cierra la conexión.

En caso de que haya un fallo en la base de datos el método despliega un mensaje de error al conectar a la BD.

10. Ahora, en el paquete llamado vistas creamos un JFrame form al cual llamaremos grafica

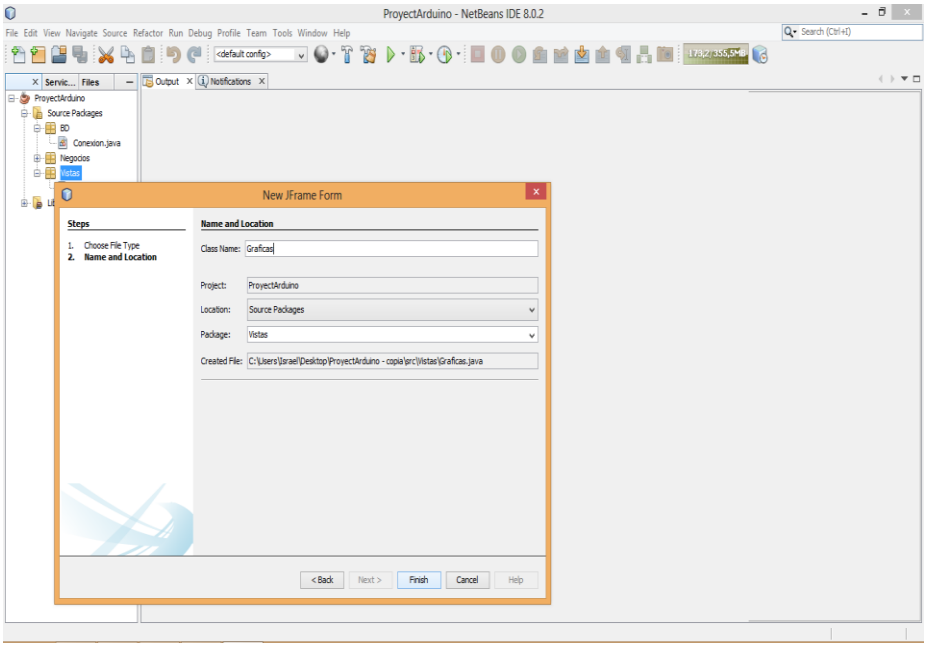


Figura 89. JFrame form llamado gráfica.

11. Una vez creada la gráfica en modo diseño, comenzamos arrastras las diversas herramientas para crear una vista y consulta con jlabel, jTextfield, jButton.

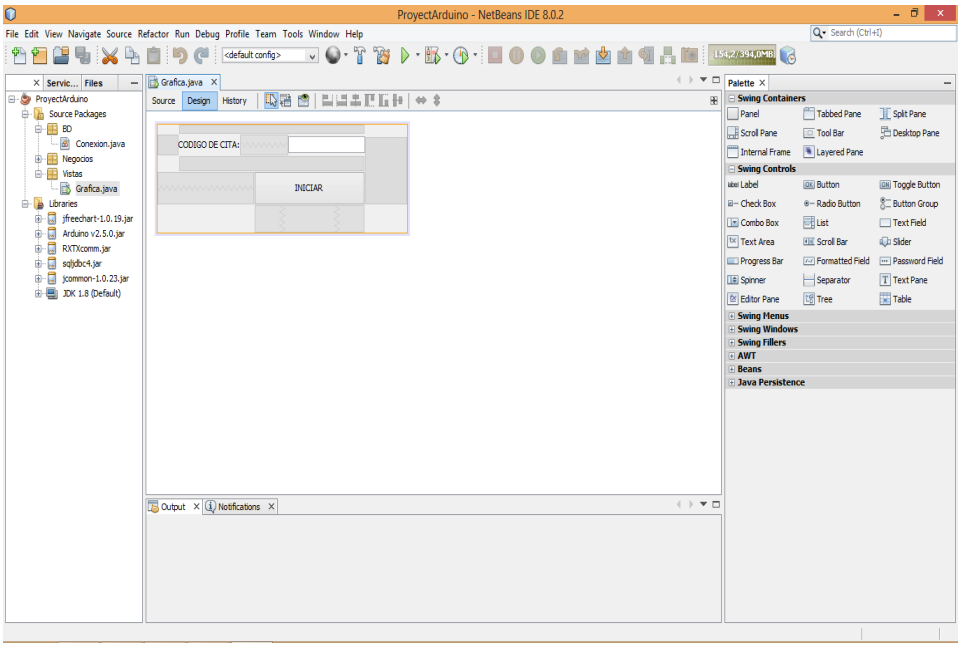


Figura 90. Creando vista.

Ya una vez puesto los elementos gráficos, nos dirigimos al modo código en el cual al iniciar el programa va a tomar los valores por defecto para que se inicialice la librería Arduino,a esta le indicamos los parámetros necesarios para que se conecte con nuestros sensores. También"dibujamos" las dos gráficas y especificamos como van a ir estructuradas.

```
public Grafica() throws Exception {

    initComponents();


        Arduino.ArduinoRX("COM10", 1000, 9600, evento);


    Series.add(0,0);
    Series3.add(0,0);
    Coleccion.addSeries(Series);
    Coleccion.addSeries(Series2);
    Coleccion2.addSeries(Series3);
    Grafica = ChartFactory.createXYLineChart("Latidos ", "", "Rango", Coleccion, PlotOrientation.VERTICAL, true, true, false);
    Grafica2 = ChartFactory.createXYLineChart("Oxigenación ", "", "Rango", Coleccion2, PlotOrientation.VERTICAL, true, true, false);


}
```

Figura 91. Estructurando las 2 graficas a mostrar

12. Establecemos los objetos necesarios para poder usar Arduino, como la conexión a la base de datos y la gráfica a mostrar.

```
//LLamamos a los objetos de arduino, la conexión ,
Arduino Arduino = new Arduino();
Conexion Conn = new Conexion();
//Llamamos a los objetosde de la libreria jfreechart la cual sirve para graficar
final XYSeries Series = new XYSeries("Pulsos");
final XYSeries Series2 = new XYSeries("Limite");
final XYSeries Series3 = new XYSeries("Oxigenación");
final XYSeriesCollection Coleccion = new XYSeriesCollection();
final XYSeriesCollection Coleccion2 = new XYSeriesCollection();
JFreeChart Grafica;
JFreeChart Grafica2;
//lista de vareables para realizar las distintas operaciones del programa
long  i = 0;
int latidos =0;
int oxi =0;
String buscar;
String valor = "";
String Dato ="";
String Valores[] = new String [8];
static  String Guardar = "";
static  String Codigo = "";
```

Figura 92. Definiendo las variables necesarias para las distintas operaciones a realizar.

13. Validamos el botón de iniciar, de la siguiente manera:

Debe existir algún valor en el campo txtCita, si no ha ingresado nada debe insertar algún valor.

Validamos el valor ingresado en el campo txtCita pues esta debe existir en la BD caso contrario, se manda el mensaje que"*debe ingresar una cita válida"*.

```
private void btngraficarActionPerformed(java.awt.event.ActionEvent evt) {
//Validar si existe el codigo de la cita

    if(txtCita.getText().equals(valor))
    {
        JOptionPane.showMessageDialog(null," Debe ingresar un valor");
        return;
    }
//Obtener valor a buscar
      buscar = txtCita.getText();
     if(Conn.BuscarSiExiste(buscar)==0)
     {
         JOptionPane.showMessageDialog(null,"¡ El código de la cita no existe !");
         return;
     }
```

Figura 93. Validaciones principales

14. Editamos el evento de Arduino para poder recibir los datos según nuestros requerimientos.

15. Preguntamos si Arduino me ha enviado un dato, si ese fuese el caso, el mensaje enviado lo paso a la variable dato, la cual voy a particionar y almacenar en un array separando los datos con un "+".

Los datos referentes a latidos y oxigenación corresponden a los dos primeros valores del array 0 y 1, esos los agregamos en las series que son referentes a las gráficas.

Si el valor estático guardar es igual a 1 entonces almacenamos en la BD utilizando el método creado en la clase conexión llamado "saveOperation", que se muestra a continuación.

```
// evento de arduino, el cual es un hilo ya que se ejecuta repati
SerialPortEventListener evento = new SerialPortEventListener(){
    @Override
    // se reescribe el evento segun a nuestros requerimientos
    public void serialEvent(SerialPortEvent spe)
    {
        i++;
        if (Arduino.MessageAvailable()==true)
            {
                Dato = Arduino.PrintMessage();
                System.out.println(Dato);
                Valores = Dato.split("\\+");

                 latidos = Integer.valueOf(Valores[1]);
                 oxi = Integer.valueOf(Valores[0]);
                 Series.add(i, Integer.valueOf(Valores[0]));
                 Series2.add(i,85);
                 Series3.add(i, Integer.valueOf(Valores[1]));
                  if(Guardar.equals("1"))
                      {
                          Conn.saveOperation(Valores, Codigo);
                      }
                 Valores[0] = "";
                 Valores[1] = "";

            }
    }
```

Figura 94. Método saveOperation

En el paquete llamado negocio se crea un mainClase, la cual inicializa el programa creando una instancia del jframe llamado Grafico, el cual ejecuta de raíz el programa.

CAPÍTULO VI: PRUEBAS Y RESULTADOS

6.1.Parámetros obtenidos en las pruebas de radio experimentales.

6.1.1. PRUEBA 1, Distancia 1m.

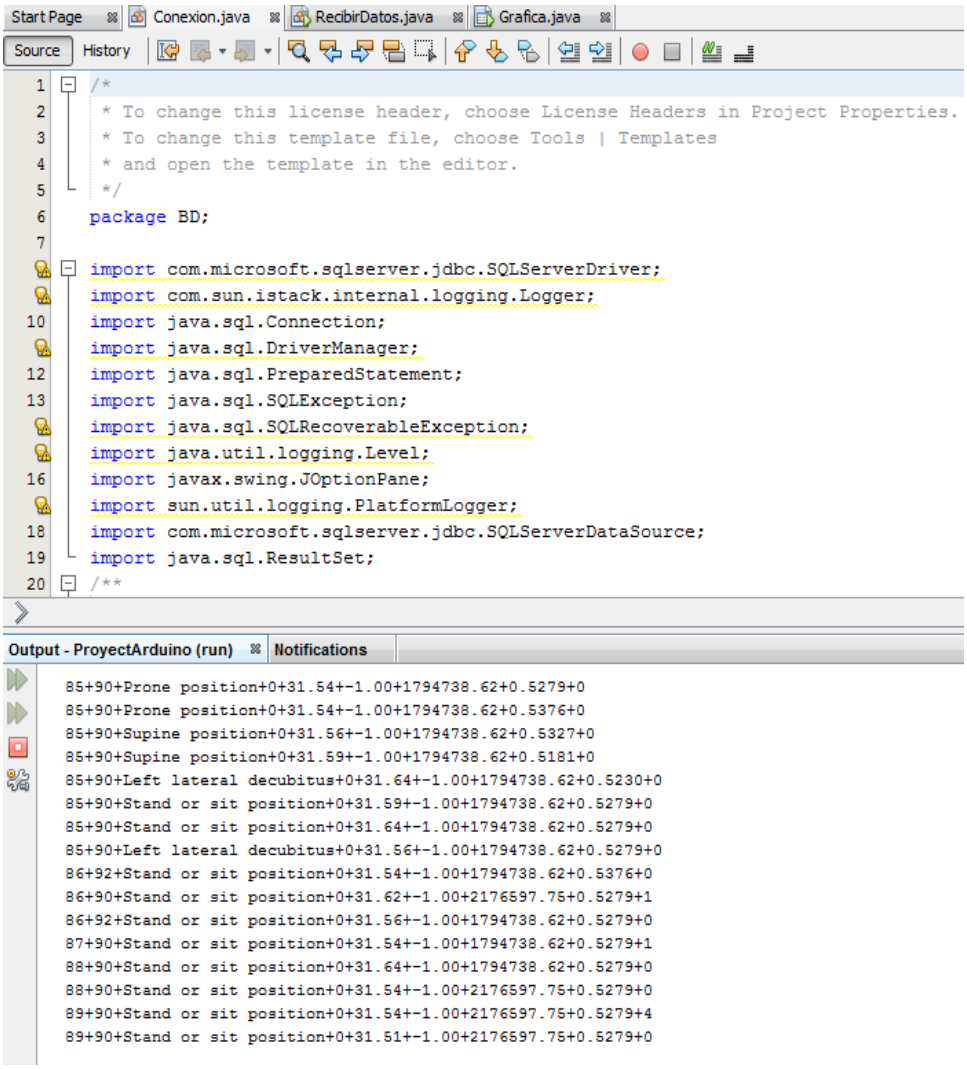


Figura 95. Resultados Distancia 1m

6.1.2. PRUEBA 2, Distancia 5m.

Start Page | Conexion.java | RecibirDatos.java | Grafica.java

Source | History

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package BD;
7
   import com.microsoft.sqlserver.jdbc.SQLServerDriver;
   import com.sun.istack.internal.logging.Logger;
10 import java.sql.Connection;
   import java.sql.DriverManager;
12 import java.sql.PreparedStatement;
13 import java.sql.SQLException;
   import java.sql.SQLRecoverableException;
   import java.util.logging.Level;
16 import javax.swing.JOptionPane;
   import sun.util.logging.PlatformLogger;
18 import com.microsoft.sqlserver.jdbc.SQLServerDataSource;
19 import java.sql.ResultSet;
20 /**
```

Output - ProyectArduino (run) | Notifications

```
80+90+Prone position+0+31.54+-1.00+1328570.50+0.5230+0
80+90+Prone position+0+31.62+-1.00+1794738.62+0.5327+0
80+90+Prone position+0+31.56+-1.00+1526864.37+0.5327+0
80+91+Stand or sit position+0+31.54+-1.00+1526864.37+0.5279+0
80+91+Stand or sit position+0+31.54+-1.00+1794738.62+0.5230+0
80+90+Stand or sit position+0+31.43+-1.00+1526864.37+0.5279+0
79+91+Rigth lateral decubitus+0+31.51+-1.00+1794738.62+0.5327+0
79+91+Rigth lateral decubitus+0+31.51+-1.00+1526864.37+0.5327+0
79+90+Stand or sit position+0+31.51+-1.00+1526864.37+0.5327+1
79+90+Left lateral decubitus+0+31.51+-1.00+1526864.37+0.5279+0
79+90+Left lateral decubitus+0+31.51+-1.00+1794738.62+0.5327+0
79+90+Stand or sit position+0+31.51+-1.00+1526864.37+0.5279+1
```

Figura 96. Resultados Distancia 5m

6.1.3. PRUEBA 3, Distancia 10m.

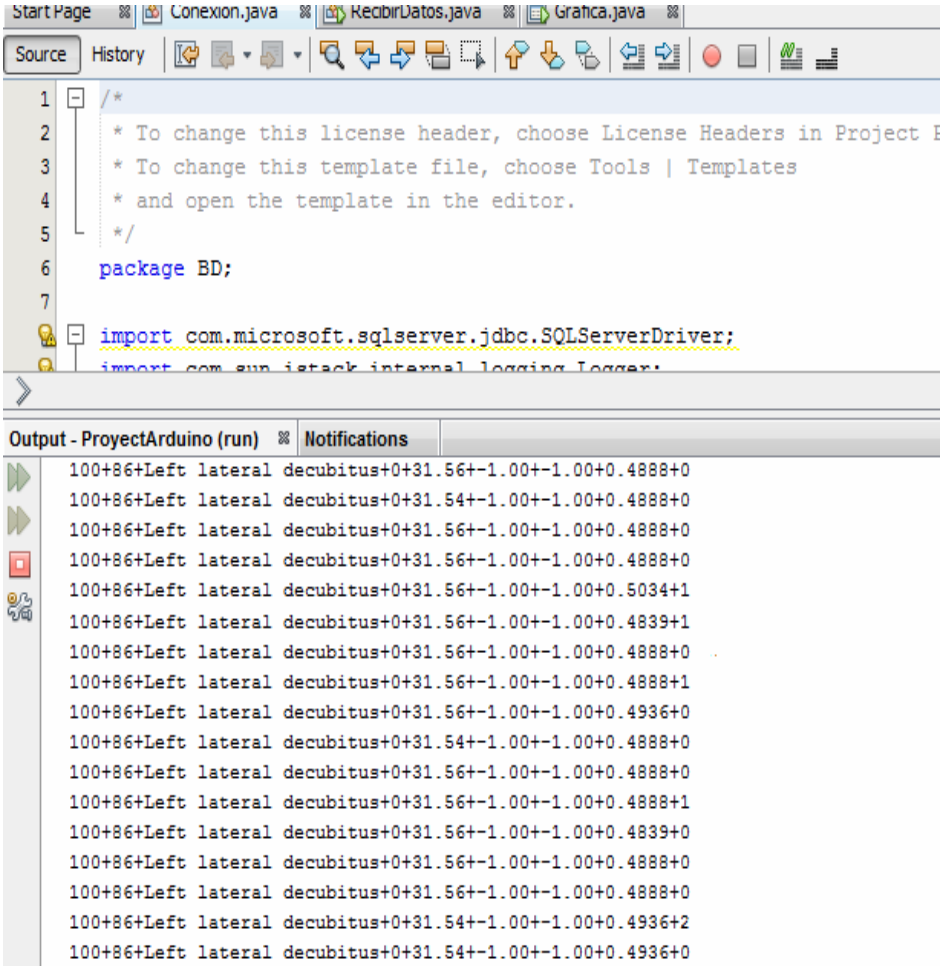


Figura 97. Resultados Distancia 10m

6.1.4. PRUEBA 4, Distancia 15m.

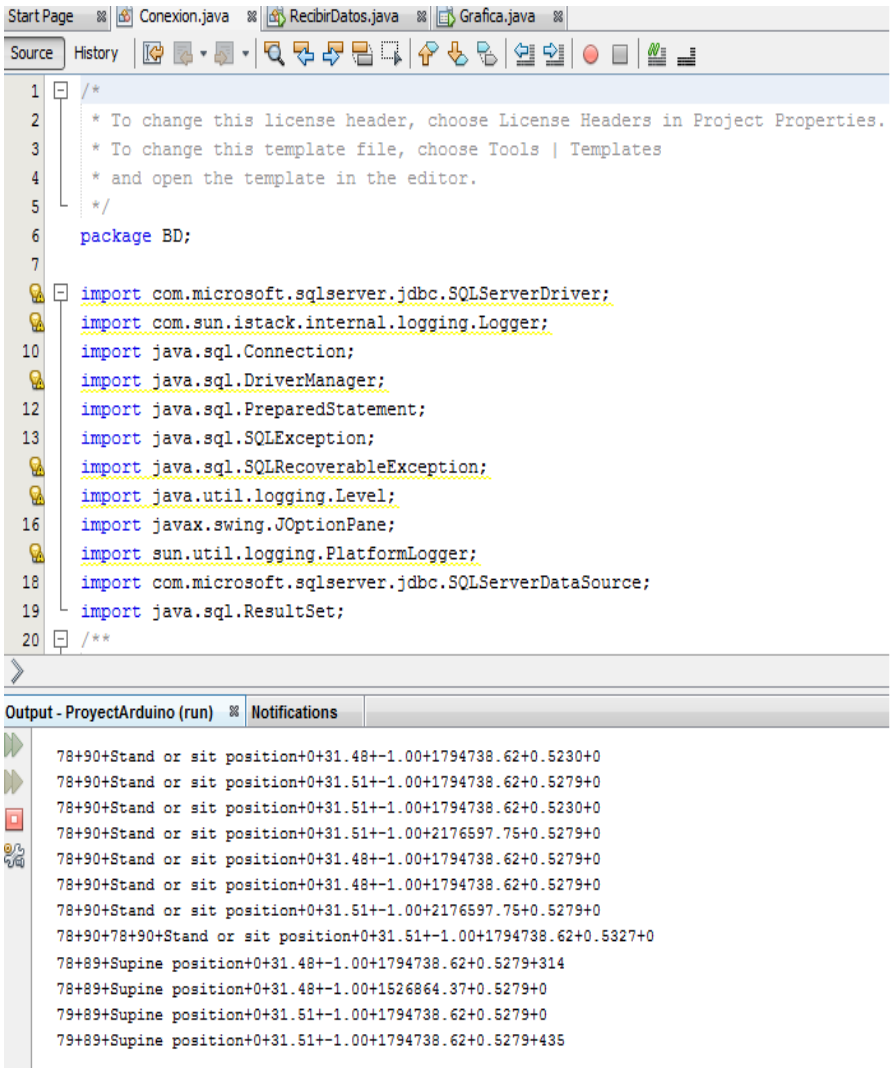


Figura 98. Resultados Distancia 15m

6.1.5. PRUEBA 5, Distancia 20m.

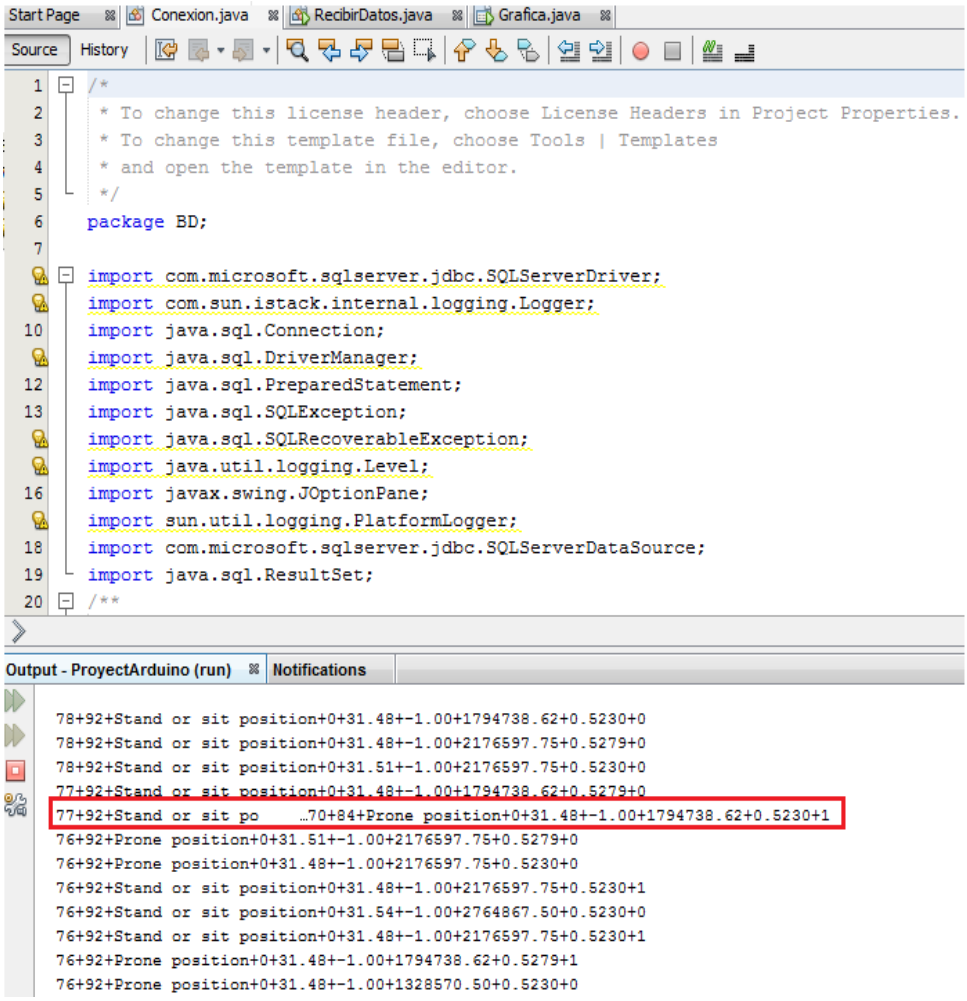


Figura 99. Resultados Distancia 20m.

6.2.Análisis de resultados.

Como podemos observar los resultados de las pruebas obtenidos del prototipo a nivel general muestran un buen rendimiento en cuanto a su funcionamiento, ya que la transmisión de los resultados es fiel pese a la distancia y a los obstáculos ya que las pruebas realizadas fueron hechas en un ambiente que simule el entorno de un centro médico, lo más cercano a la realidad.

Para las 4 primeras pruebas es decir hasta una distancia de 15 metros y con presencia de obstáculos el tiempo de estabilización es de 2 segundos, mientras que a distancias mayores los datos siguen siendo fieles la única variante es el tiempo de estabilización, el cual fue de 4 segundos en este tiempo los datos receptados son “basura”, pero una vez transcurrido, la señal es fiel.

CAPITULO VII: CONCLUSIONES Y RECOMENDACIONES

7.1. Conclusiones:

- Se diseñó una Red punto a punto con éxito, ya que permite el envío y recepción de datos mediante el estándar IEEE 802.15.4. (ZIGBEE).
- Se configuro y calibro 5 sensores de modo que envíen de forma correcta los signos vitales obtenidos.
- Se realizaron pruebas de funcionamiento a corta y larga distancia para posteriormente comparar los datos obtenidos y determinar la distancia exacta en que los valores pierden fidelidad.
- Se diseñó una base de datos la cual almacena información de pacientes como: datos informativos, citas, signos vitales, entre otros, la cual permite llevar un registro ordenado de los pacientes atendidos.
- Se integró la Base de Datos con un entorno web amigable con el usuario de forma que toda persona pueda interactuar con el sistema, sin necesidad de una preparación previa.

7.2. Recomendaciones:

- La librería RXTX fue fundamental para cumplir con el propósito deseado sin embargo debemos tomar en cuenta que al ser una librería gratuita "light“, por lo que la base de datos puede almacenar un máximo de 1500 registros.
- Es recomendable utilizar un sistema hibrido debido a las prestaciones de las versiones light pues nos benefician en tiempo de desarrollo y costos.
- Es mejor enviar los datos entramados y no uno por uno, debido a la forma de decodificación ya que de esta manera se favorece al correcto almacenamiento y sincronización de la base de datos.
- Realizar un estudio previo de la compatibilidad entre el sistema operativo y los softwares utilizados para evitar problemas con la base de datos.
- Debido a la inestabilidad de los sensores se debe trabajar de forma independiente, con el fin de evitar la pérdida de datos.

- Durante la programación de los sensores procurar trabajar en un ambiente de temperatura y humedad constante hasta lograr la estabilidad requerida por el sistema.

Bibliografía

Arévalos Brazas, A. (2011). *Programación de un Nodo Conmutador para la Gestión Remota de Redes 802.15.4/Zigbee.* Universidad de Málaga, Málaga.

Barriga, W. (2006). *Tecnologías Inalámbricas Zigbee, Bluetooth.* Universidad Politécnica Salesiana, Quito.

Betancur, L. (2011). *Redes de Área Corporal. Una Perspectiva al Futuro desde la Investigación.* Universidad Icesi, Calí.

Capella Hernández, J. V. (2010). *Redes Inalámbricas de Sensores: Una Nueva Arquitectura Eficiente y Robusta Basada en Jerarquía Dinámica de Grupos.* Universidad Politécnica de Valencia, Valencia.

Cobos Hernández, E. (2007). *Estudio de las Redes Sensoriales Como una Nueva Alternativa de Comunicación Inalámbrica.* Escuela Politécnica del Ejército, Quito.

Cooking Hacks. (2013). *e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi [Biometric / Medical Applications].* Obtenido de http://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical

Docsetools. (2014). *Red de área Corporal*. Obtenido de http://docsetools.com/articulos-enciclopedicos/article_92513.html

Flores, E. (2012). *Redes de Sensores Inalámbricas Aplicado a la Medicina.* Universidad de Cantabria, Cantabria.

Garbarino, J. (2012). *Protocolos para Redes Inalámbricas de Sensores.* Universidad de Buenos Aires, Buenos Aires.

Glen, M., & Moreno, J. (2012). *Zigbee*. Obtenido de http://sx-de-tx.wikispaces.com/ZIGBEE

Hacedores. (2014). *Arduino o Raspberry Pi, ¿Cuál es la mejor Herramienta para tí?* Obtenido de http://hacedores.com/arduino-o-raspberry-pi-cual-es-la-mejor-herramienta-para-ti/

Kioumars, A., & Tang, L. (2011). *Wireless Network for Health Monitoring: Heart Rate and Temperature Sensor. Proceeding of the International Conference on Sensing Technology.* Massey University, Auckland.

Libelium. (2014). *X-CTU Tutorial*. Obtenido de http://www.libelium.com/es/development/waspmote/documentation/x-ctu-tutorial/

Martínez, D., Simo, J., Blandes, F., & Crespo, A. (2009). *Wireless Sensor and Actuator Networks: Characterization and Cases study for Confined Spaces Healthcare and Control Applications. Scalable Computing: Practice and Experience, Scientific International Journal for Parallel and Distributed.* Universidad Politécnica de Valencia, Valencia.

Moreno, E. (2014). *Plataforma de Desarrollo de WSAN y Middleware para Despliegue de Aplicaciones.* Universidad Politécnica de Madrid, Madrid.

MYSQL. (2015). *Installign and Configuring*. Obtenido de http://dev.mysql.com/doc/connector-net/en/connector-net-visual-studio-install.html

Papacetzi, F. M. (2003). *Tesis Licenciatura Ingeniería en Electrónica y Comunicaciones, Estándar IEEE 802.15.4.* Puebla.

Romano, P. (2013). *Reconocimiento de actividad humana mediante técnicas de Soft Computing.* Universidad de Oviedo, Oviedo.

Zabala, G. (2012). *Hombre Máquina*. Obtenido de http://slideplayer.es/slide/1075551/

ZigBee PRO Stack User Guide. (2014).