



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

**“ALGORITMO DE ACCESO AL MEDIO EN CAPA APLICACIÓN PARA
TECNOLOGÍA SENSOR NODE DEVICE (MOTE), APLICADO A ESTÁNDARES DE
COMUNICACIÓN RF, ZIGBEE, WIFI Y BLUETOOTH.”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA
EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

AUTOR: JUAN PABLO LOZA ACOSTA

DIRECTOR: MSC. JAIME ROBERTO MICHILENA CALDERÓN

Ibarra-Ecuador

2017

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

1. IDENTIFICACIÓN DE LA OBRA.

La Universidad Técnica del Norte dentro del Proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital, con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

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

DATOS DEL CONTACTO	
Cédula de identidad	1003578133
Apellidos y Nombres	Loza Acosta Juan Pablo
Dirección	Salinas 1-68 y Mejía
E-mail	jploza@utn.edu.ec
Teléfono móvil	0996734505
DATOS DE LA OBRA	
Título	ALGORITMO DE ACCESO AL MEDIO EN CAPA APLICACIÓN PARA TECNOLOGÍA SENSOR NODE DEVICE (MOTE), APLICADO A ESTÁNDARES DE COMUNICACIÓN RF, ZIGBEE, WIFI Y BLUETOOTH.
Autor	Loza Acosta Juan Pablo
Fecha	

Programa	Pregrado
Título	Ingeniero en Electrónica y Redes de Comunicación
Director	Ing. Jaime Michilena Calderón, MSc.

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD.

Yo, Loza Acosta Juan Pablo, con cédula de identidad Nro. 1003578133, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad de material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.

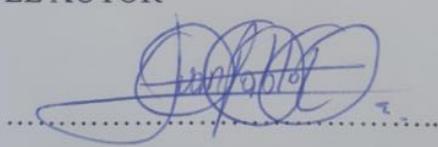
3. CONSTANCIAS.

Yo, LOZA ACOSTA JUAN PABLO declaro bajo juramento que el trabajo aquí escrito es de mi autoría; y que éste no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se presentan en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Universidad Técnica del Norte, según lo establecido por las leyes de propiedad intelectual, reglamentos y normatividad vigente de la Universidad Técnica del Norte.

En la ciudad de Ibarra, noviembre de 2017.

EL AUTOR



Loza Acosta Juan Pablo

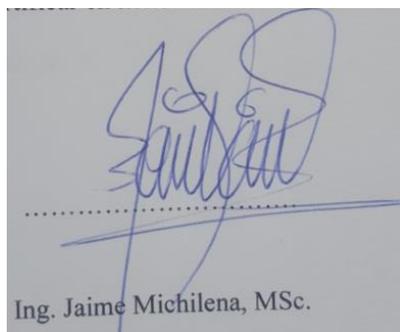
CI: 1003578133

UNIVERSIDAD TÉCNICA DEL NORTE**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS****CERTIFICACIÓN.**

MAGISTER JAIME MICHILENA, DIRECTOR DEL PRESENTE TRABAJO DE TITULACIÓN CERTIFICA:

Que, el presente trabajo de Titulación “ALGORITMO DE ACCESO AL MEDIO EN CAPA APLICACIÓN PARA TECNOLOGÍA SENSOR NODE DEVICE (MOTE), APLICADO A ESTÁNDARES DE COMUNICACIÓN RF, ZIGBEE, WIFI Y BLUETOOTH”. Ha sido desarrollado por el señor Loza Acosta Juan Pablo bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.



Ing. Jaime Michilena, MSc.

1002198438

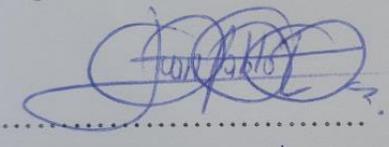
DIRECTOR

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

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Yo, Loza Acosta Juan Pablo, con cédula de identidad Nro. 1003578133, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador artículos 4, 5 y 6, en calidad de autor del trabajo de grado con el tema: ALGORITMO DE ACCESO AL MEDIO EN CAPA APLICACIÓN PARA TECNOLOGÍA SENSOR NODE DEVICE (MOTE), APLICADO A ESTÁNDARES DE COMUNICACIÓN RF, ZIGBEE, WIFI Y BLUETOOTH. Que ha sido desarrollado con el propósito de obtener el título de Ingeniero en Electrónica y Redes de Comunicación de la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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



Loza Acosta Juan Pablo

1003578133

Ibarra, noviembre 2017

Agradecimiento.

Agradezco a Dios por acompañarme y guiarme a lo largo de mis estudios además de brindarme una vida llena de aprendizaje, y experiencias fortalecedoras.

Al Ingeniero Jaime Michilena MSc., director del presente Trabajo de Grado quien ha compartido su tiempo, su conocimiento y me ha apoyado para desarrollar y culminar la presente tesis, además de darme la oportunidad de crecer profesionalmente y aprender cosas nuevas.

A mi madre y hermanos por ser parte importante de mi vida, por haberme apoyado en las buenas y en las malas, sobre todo con su paciencia y amor incondicional, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida.

Finalmente, a cada uno de docentes que supieron compartirme su conocimiento y haberme guiado de la mejor manera para convertirme en una gran persona y un excelente profesional.

Dedicatoria

Dedico mi presente trabajo, a mi madre: María Acosta quien me ha sabido guiar siempre por el camino al éxito a pesar de las dificultades o tropiezos que existen en la vida, además de ser un gran ejemplo de perseverancia y fuerza para que cada día de mi vida sea lleno de bendiciones.

A mis hermanas, Anita Belén, Alexandra, y mi hermano Francisco Saúl por apoyarme incondicionalmente en cada momento importante de mi vida, y demostrarme que todo se puede lograr con solo proponérselo.

A mi grupo de amigos quienes con sus consejos y positivismo han sabido apoyarme y hacerme seguir adelante en cada uno de mis momentos de dificultad; a mi novia Sylvia Changoluisa quien ha estado presente en toda mi carrera universitaria y en cada instante de mi vida apoyándome y fortaleciéndome para lograr una meta de muchas que me he propuesto.

Índice.

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.	I
AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD.....	III
CONSTANCIAS.....	IV
CERTIFICACIÓN.....	V
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	VII
Agradecimiento.....	VII
Dedicatoria.....	VIII
Índice.....	IX
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS	XV
Resumen.....	XVII
Abstract	XVIII
Capítulo 1. Antecedentes.	1
1.1. Tema.....	1
1.2. Problema.....	1
1.3. Objetivos.	3
1.3.1. Objetivo general.	3
1.3.2. Objetivos específicos.....	3
1.4. Alcance.....	4
1.5. Justificación.....	5
Capítulo 2. Justificación Teórica	7
2.1. Sistema Embebido.....	7
2.1.1. Concepto.....	8
2.1.2. Características	8
2.1.3. Entradas	9
2.1.4. CPU	11
2.1.5. Arduino.....	14
2.1.6. Salida	18

3.4. METODOLOGÍA EN V: Fase 3 Diseño del Sistema Electrónico	82
3.4.1. Diagrama de bloques general	83
3.4.2. Nodo Central	84
3.4.3. Nodo Estación	88
3.5. METODOLOGÍA EN V: Fase de prueba	90
3.5.1. Escenario de pruebas para los sistemas de comunicación inalámbricas	90
3.5.2. Explicación del funcionamiento de los sistemas.	95
3.5.3. Fase de prueba uno “Estándar De Comunicación Inalámbrica RF”	96
3.5.4. Fase de prueba dos “Estándar de comunicación inalámbrica Zigbee”	103
3.5.5. Fase de prueba tres “Estándar de comunicación inalámbrica WiFi”	111
3.5.6. Fase de prueba cuatro “Estándar de comunicación inalámbrica Bluetooth”	118
.....	118
3.6. Desarrollo del nuevo algoritmo de comunicación.....	124
Capítulo 4. Conclusiones y Recomendaciones	131
4.1. Conclusiones	131
4.2. Recomendaciones.....	132
Referencias.....	134
Glosario de Términos y Acrónimos.....	139
Anexos	141

ÍNDICE DE FIGURAS

Figura. 1 Diagrama de Flujo de sus componentes de una CPU.....	11
Figura. 2 Ejemplo de placa Arduino	15
Figura. 3. Entorno de Software de desarrollo	18
Figura. 4 Formación de operación en el funcionamiento de Bluetooth.	25
Figura. 5 Localización del espectro mediante su longitud de onda operativa.	26
Figura. 6 Topologías en IEEE 802.15.4 en estándar Zigbee.....	28
Figura. 7 Estructura mediante un diagrama de bloques de un MOTE.....	31
Figura. 8. Análisis Aloha puro de colisiones	34
Figura. 9. Rendimiento de Aloha puro representado en un gráfico en línea.....	36
Figura. 10. Diagrama de flujos de Aloha puro.....	37
Figura. 11. Análisis Aloha Ranurado de colisiones	38
Figura. 12 Diferenciación de rendimientos entre Aloha puro y ranurado mediante un gráfico en línea.	39
Figura. 13. Throughput en función del número de estaciones	42
Figura. 14. Diagrama de flujo de CSMA/CA.	43
Figura. 15 Diagrama de desarrollo en bloques de una metodología en cascada.....	44
Figura. 16 Fases de trabajo en el Modelo de Prototipos.	48
Figura. 17. Niveles para desarrollo de la Metodología en V	49
Figura. 18. Puertos de conexión de Modulo ESP8266 visto desde la parte superior.....	79
Figura. 19. Pines de configuración que cuenta un módulo Xbee S2C.....	81
Figura. 20. Esquema de montaje de un módulo RF NRF24L01	82
Figura. 21. Diagrama de bloques general del sistema electrónico.....	83
Figura. 22. Diagrama de bloques del Nodo Central con sus respectivos subsistemas.....	84
Figura. 23. Diagrama de flujo del Nodo Central con sus respectivos procesos.....	87
Figura. 24. Diagrama de bloques de un Nodo Estación.....	88
Figura. 25. Diagrama de flujo del nodo estación con sus procesos.	89
Figura. 26. Diagrama del entorno de análisis controlado.....	90
Figura. 27. Batería de 9V para sistemas embebidos.	92
Figura. 28. 4 pilas AA para sistemas embebidos.	93

Figura. 29. Baterías de polímetro litio para subsistemas.	94
Figura. 30. Batería USB 5 Voltios	94
Figura. 31. Esquema electrónico del Nodo Central en fritzing.....	98
Figura. 32. Esquema de conexión del Nodo Central y sus componentes en fritzing.....	99
Figura. 33. Esquema eléctrico del nodo estación en fritzing.	100
Figura. 34. Esquema de conexiones del nodo estación en fritzing.	100
Figura. 35. Visualización de datos en los nodos estación.	101
Figura. 36. Visualización de datos en el nodo central.....	102
Figura. 37. Red de sensores utilizando módulos RF.....	103
Figura. 38. Configuración de Módulos Xbee en XCTU	106
Figura. 39. Esquema eléctrico de un nodo central con Xbee en fritzing.	107
Figura. 40. Esquema de conexión del nodo central con Xbee en fritzing.....	107
Figura. 41. Esquema eléctrico de un nodo estación con Xbee en fritzing.	108
Figura. 42. Esquema de conexión del nodo estación con Xbee en fritzing.	109
Figura. 43. Divisor de tensión para protección de los módulos Xbee.....	109
Figura. 44. Recepción de datos de nodo central en el sistema Xbee.	110
Figura. 45. Comunicación en la red de sensores Xbee.	111
Figura. 46. Gestor de tarjetas ESP8266 para el módulo de comunicación WiFi	113
Figura. 47. Esquema de conexión eléctrico del nodo central para el sistema WiFi en Fritzing.....	114
Figura. 48. Esquema de conexiones del sistema WiFi en Fritzing.	115
Figura. 49. Esquema eléctrico de conexión del nodo estación en Fritzing.	115
Figura. 50. Esquema de conexión del nodo estación del sistema WiFi en Fritzing.....	116
Figura. 51 Datos recolectados en el sistema WiFi – Nodo central	117
Figura. 52. Red de sensores WiFi, nodo central de la red.....	118
Figura. 53. Configuración mediante comandos AT de un módulo Bluetooth	120
Figura. 54. Esquema eléctrico del Nodo central en fritzing.....	121
Figura. 55. Esquema de conexión del sistema bluetooth en fritzing.....	121
Figura. 56. Esquema eléctrico del nodo estación en fritzing.	122
Figura. 57. Esquema de conexión del nodo estación en fritzing.....	122
Figura. 58. Datos recolectados en la red de sensores Bluetooth.	123

Figura. 59. Red de sensores Bluetooth.....	124
Figura. 60. Recepción de datos Arduino del nodo central.	127
Figura. 61. Trasmisión de datos Arduino en nodo estación.	128
Figura. 62. Tiempo de procesamiento de datos con y sin algoritmo implementado.....	129

ÍNDICE DE TABLAS

Tabla 1. Características importantes en un Sistema Embebido.	9
Tabla 2. Parámetros que conforma una unidad de control en un Sistema Embebido.	13
Tabla 3. Plataformas de Arduino	16
Tabla 4. Parámetros de comunicación entre sistemas.	20
Tabla 5. Familia de protocolos IEEE 802.11	22
Tabla 6. Características Bluetooth	24
Tabla 7. Características ZigBee	27
Tabla 8. Descripción y valores de parámetros a utilizar en el cálculo de muestra	58
Tabla 9. Lista de Acrónimos y Abreviaturas	63
Tabla 10. Lista de Stakeholders	64
Tabla 11. Requerimientos operacionales del sistema a generar.....	65
Tabla 12. Requerimientos de usuario del sistema a generar.	65
Tabla 13. Requerimientos iniciales del sistema electrónico	66
Tabla 14. Requerimientos funcionales de Software y Hardware en el desarrollo del sistema electrónico.	68
Tabla 15. Elección en base a los requerimientos de Software	71
Tabla 16. Elección de Hardware para el subsistema de procesamiento.....	72
Tabla 17. Elección de subsistema de comunicación WiFi.....	73
Tabla 18. Elección de subsistema de comunicación ZigBee	73
Tabla 19. Elección de subsistema de comunicación Bluetooth	74
Tabla 20. Elección de subsistema de comunicación RF	75
Tabla 21. Elección del subsistema de detección	76
Tabla 22. Componentes importantes de un Arduino Uno.....	77
Tabla 23. Características importantes en un módulo WiFi ESP8266	78
Tabla 24. Especificaciones del Módulo S2C Zigbee Xbee.....	80
Tabla 25. Especificaciones técnicas del módulo NRF24L01.....	81
Tabla 26. Explicación del funcionamiento general de los sistemas de prueba.	95
Tabla 27. Componentes que forman parte para el desarrollo del sistema RF.....	96
Tabla 28. Equipamiento del sistema embebido Zigbee.	104

Tabla 29. Elementos que conformará la red de sensores WiFi.....	112
Tabla 30. Equipamiento del sistema Bluetooth.	119
Tabla 31. Identificadores de trama de comunicación.....	125
Tabla 32. Identificadores de trama.....	128

Resumen.

Se desarrolló un algoritmo de acceso al medio en capa aplicación para tecnología sensor node device (MOTE), aplicado a estándares de comunicación RF, ZIGBEE, WIFI Y BLUETOOTH; con la finalidad de mejorar la trasmisión de grandes volúmenes de datos en redes de sensores inalámbricos que utilizan este tipo de tecnologías, a través de identificadores como: prioridad de envío de paquetes, indicador de batería, número de sensores análogos o digitales que conformen la red, además de una verificación de envío de datos mediante el tiempo procesamiento.

El estudio de protocolos de acceso al medio genera una ventaja competitiva para la construcción de nuevos algoritmos que ayudan con la elaboración de procesos que desean agilizar la transferencia de información en redes de sensores inalámbricas. Uno de los aspectos a considerar es la correcta implementación de mecanismos y protocolos de comunicación que van a operar para no limitar su acceso a la red y la escalabilidad de dispositivos a conectarse, ya que al aumentar su número, mayor volumen en la transferencia de información, ocasiona un alto número de colisiones en caso de no tener una red organizada adecuadamente, teniendo en cuenta que son dispositivos con bajas capacidades computacionales siendo necesario generar propias formas de acceso a la red.

Con las pruebas de laboratorio que se realiza a cada una de las tecnologías antes mencionadas se crearon escenarios en donde se pudo realizar la comparación entre el método tradicional con el nuevo algoritmo de acceso al medio de trasmisión de datos con lo cual se logró corroborar que la reducción de paquetes perdidos es mucho menor.

Abstract

An application layer medium access algorithm was developed for sensor node device technology (MOTE), applied to RF, ZIGBEE, WIFI and BLUETOOTH communication standards; With the purpose of improving the transmission of large volumes of data in networks of wireless sensors that use this type of technology, identified as: priority of sending packets, battery indicator, number of analog and digital sensors that make up the network, In addition to a verification of sending of data.

The study of these types of access to the environment generates a competitive advantage for the construction to streamline processes in the elaboration of these networks. One of the aspects to consider is the correct implementation of communication mechanisms and protocols that will operate in order not to limit their access to the network and the scalability of devices to be connected, since by increasing their number, more information volume will exist, causing a High number of collisions in case of not having a properly organized network, taking into account that they are devices with low computational capacities and it is necessary to generate own forms of access to the network.

With the laboratory tests that were carried out to each one of the technologies mentioned above, scenarios were created in which it was possible to make the comparison between the traditional method and the new algorithm of access to the means of transmission of data with which it was possible to corroborate that the Reduction of lost packets is much less.

Capítulo 1. Antecedentes.

En este capítulo se encuentran detalladas las bases para el desarrollo del presente trabajo de titulación, siendo éstos: el tema, la problemática, los objetivos, el alcance y la justificación, con la finalidad de expresar la importancia de la realización del algoritmo de acceso al medio en capa aplicación para tecnología sensor Node Device (MOTE), aplicado a estándares de comunicación RF, Zigbee, WiFi y Bluetooth.

1.1. Tema.

ALGORITMO DE ACCESO AL MEDIO EN CAPA APLICACIÓN PARA TECNOLOGÍA SENSOR NODE DEVICE (MOTE), APLICADO A ESTÁNDARES DE COMUNICACIÓN RF, ZIGBEE, WIFI Y BLUETOOTH.

1.2. Problema.

El rendimiento de protocolos y arquitecturas en diseño de redes de sensores inalámbricos se ha convertido en un campo de investigación muy importante ya que el desarrollo de estas tecnologías junto con estándares inalámbricos de comunicación, han permitido la aparición de nuevos sistemas como la computación ubicua. Con ello se prevé un futuro en el que minúsculos sensores realicen un monitoreo al medio continuamente y reporten su información a los nodos próximos o a una estación base central. Las aplicaciones en este campo son enormes, por ejemplo, en el campo de domótica y control inteligente de edificios, medicina, agricultura, etc. Estos tipos de aplicaciones en los distintos campos crecerán de gran manera, orientándolas en generación de conocimiento, aprendizaje y construcción de sistemas tecnológicos. (Castillo, 2012)

Gracias a los avances tecnológicos actualmente existe un sin número de aplicaciones que se enfocan a IoT todo esto con la finalidad de mejorar el estilo de vida de las personas, una de las preocupaciones existentes es el aumento de dispositivos y el impacto que va a generar en la población, en el año 2015 la empresa Gartner dedicada a la consultoría de tecnologías de la información realizó un estudio acerca de los intereses que generará el Internet de las Cosas en el amplio desarrollo de aplicaciones en los años presentes y futuros (Gartner, 2015), donde determinan el reto del desarrollo e interoperabilidad de una red de sensores inalámbricos ya que jugarán un papel importante para el crecimiento del Internet de las Cosas mediante dispositivos inteligentes integrados, uno de los aspectos a considerar es la correcta implementación de mecanismos y protocolos de comunicación que van a operar para no limitar su acceso a la red y la escalabilidad de dispositivos a conectarse, ya que al aumentar su número, mayor volumen de información existirá ocasionando un alto número de colisiones en caso de no tener una red organizada adecuadamente, teniendo en cuenta que son dispositivos con bajas capacidades computacionales y es necesario generar propias formas de acceso a la red, además los dispositivos en un hogar mantendrá su crecimiento donde la mayoría son diseñados con sensores inalámbricos para evitar la instalación de cable, estos se utilizan para controlar diversas condiciones del hogar, su correcta comunicación es una parte fundamental para su correcto funcionamiento donde se pueden implementar con diferentes tecnologías pero se desconoce en qué aplicación cuál es la más idónea.

Para ello se desea realizar un estudio del rendimiento de acceso al medio y su impacto para así analizar, comparar, y brindar resultados que nos ayude determinar que estándar entre (RF, ZIGBEE, WIFI y BLUETOOTH) trabaja mejor con protocolos de control de acceso al medio como Aloha y CSMA/CA que son los métodos de acceso al medio en redes inalámbricas. Esto permitirá

tener un claro conocimiento de que estándares son más eficientes al momento de trabajar con dichos protocolos y su desenvolvimiento frente a aplicaciones que se generaran a futuro, además de lograr interconectar mayor cantidad de sensores sin sacrificar el rendimiento de la red con una fácil administración.

1.3. Objetivos.

1.3.1. Objetivo general.

Desarrollar un algoritmo de acceso al medio en capa aplicación para tecnología Sensor Node Device (MOTE), aplicado a estándares de comunicación RF, ZIGBEE, WIFI Y BLUETOOTH para protocolos de control de acceso al medio CSMA/CA Y ALOHA.

1.3.2. Objetivos específicos.

Realizar una revisión bibliográfica para determinar las bases teóricas correspondiente a redes de sensores, protocolos de control de acceso al medio y evolución de nuevas tecnologías con redes inalámbricas referente a estándares de comunicación RF, ZIGBEE, WIFI Y BLUETOOTH.

Implementar los algoritmos en capa aplicación de CSMA/CA y ALOHA para tecnologías Sensor Node Device (MOTE) en un microcontrolador.

Comparar los algoritmos diseñados de control de acceso al medio en tecnologías RF, ZIFBEE, WIFI Y BLUETOOTH.

Desarrollar un algoritmo que sintetice los dos métodos de acceso al medio CSMA/CA y ALOHA al momento de trabajar con estándares de comunicación RF, ZIFBEE, WIFI Y BLUETOOTH.

1.4. Alcance.

En el presente proyecto de tesis se llevará a cabo la creación de un algoritmo de acceso al medio en capa aplicación para tecnologías Sensor Node Device (MOTE), todo aplicado a redes de sensores en estándares de comunicación, RF, Zigbee, WIFI, BLUETOOTH diseñados en protocolos CSMA/CA y ALOHA.

En primera instancia se estudiará las tecnologías que se aplicarán en la red de sensores, investigando las limitaciones y comportamientos de cada uno de estos obteniendo fundamentaciones teóricas necesarias para lograr a implementación satisfactoria del proyecto, además del algoritmo de acceso al medio en capa aplicación basado en CSMA/CA y ALOHA. Luego se diseñara el algoritmo para la respectiva comparación de las tecnologías RF, Zigbee, WIFI y BLUETOOTH realizando pruebas pertinentes de cada uno con las tecnologías mencionadas, todo esto llevándose a cabo a través de una plataforma Arduino basado en lenguaje C, tomando en cuenta los parámetros que permitirán analizar su comportamiento y conjuntamente recopilando la información que será documentada para lograr una mejor apreciación de los resultados aplicado a la red de sensores examinadas.

Finalmente, con la investigación y las pruebas elaboradas se desarrollará y pondrá a prueba un nuevo algoritmo que contenga lo mejor de CSMA/CA y ALOHA acorde a los estándares de comunicación para tecnologías Sensor Node Device (MOTE), aplicando a redes de sensores.

1.5. Justificación.

En el país, promover la investigación y el conocimiento científico además de la innovación tecnológica cumple con el objetivo 2 del Plan Nacional del Buen Vivir el cual dicta el mejorar las capacidades y potencialidades de la ciudadanía, promoviendo a la transformación de la matriz productiva y desarrollando capacidades estratégicas y conocimiento, impulsando al uso de Tecnologías de la información y la comunicación(TIC), además de posibles mejoras para el crecimiento del país siendo de vital importancia permitir mejorar los procesos de educación. (Plan Nacional Del Buen Vivir, 2009-2013)

Cuando damos referencia a una red de sensores nos centramos a nodos equipados con sensores que colaboran una tarea común y a la carga computacional que generan estos, el desarrollo de esta tecnología ha evolucionado de una manera elevada por sus múltiples aplicaciones en distintos sectores como: seguridad, medio ambiente, agricultura entre otras y hoy en día es más conocida como Internet de las Cosas, nuevas innovaciones tecnológicas que hacen el uso de la Internet y estas redes para relacionar la forma que actúa en cada campo según su interconexión y la elevada cantidad de datos a tiempo real que genera, para ello es necesario un estudio de una correcta utilización de esta tecnología en protocolos de acceso al medio, posibilitando una ventaja al momento de trabajar con los diversos estándares inalámbricas y facilitando un beneficio para una correcta elaboración e innovación de Internet de las Cosas, en sectores educativos e incluso emprendimiento.

El estudio de estos tipos de acceso al medio genera una ventaja competitiva para la construcción a agilizar procesos en la elaboración de estas redes ya que la evolución de este tipo de tecnología facilita el estilo de vida cotidiano de las personas y abre una brecha a la innovación

de esta tecnología con protocolos de control de acceso al medio CSMA/CA y ALOHA en estándares como: RF, ZIGBEE, WIFI y BLUETOOTH orientándolo profundamente a Internet de las Cosas.

Capítulo 2. Justificación Teórica

En el presente capítulo se realiza una revisión bibliográfica de los temas a fines al desarrollo de la investigación con definiciones teóricas resumidas de manera objetiva, además de métodos de desarrollo que se tomara en cuenta para realizar el capítulo de diseño usando un debido proceso de investigación.

2.1. Sistema Embebido

En la actualidad existen varios métodos de Ingeniería con soluciones tecnológicas que ayuden a mejorar el estilo de vida y frecuentes innovaciones para la población, en su mayoría han nacido basados en sistemas embebido, ya que cuentan con un alto desempeño y con fuertes restricciones temporales, se los puede definir como una máquina computacional que ejerce tareas específicas, diseñados para controlar una o rangos específicos de funciones, estos dispositivos están desde un taxímetro, un sistema de control de acceso, copiadoras, hasta en un satélite o planta nuclear, es por ello la importancia evolutiva de esta tecnología.

Un sistema embebido está basado en un microprocesador a diferencia de un computador personal, en esta tecnología un usuario puede hacer la elección sobre su funcionalidad, pero no puede cambiar ésta, ya que están diseñados para una función en particular, el manejo de pocas tareas no es un sinónimo de simpleza, por el contrario, esto se refiere más bien a que el sistema se desarrolla para el manejo de pocas tareas y propósitos específicos tan complejos como su procesamiento de datos lo permita, estos pueden ser actualizados para mejorar su funcionalidad. La clave está en la capacidad para agregar funciones nuevas, no depende de cambiar el hardware,

puede ser realizado por un simple cambio en el software. Si el sistema está conectado a una vía de comunicación como un teléfono o una PC en red.

Cerca del 99 % de todos los sistemas basados en microcontroladores están hechos como sistemas embebidos para propósito especiales, esto comparado con apenas el 1 % para computadores de propósito general. (Bernabé, 2011)

2.1.1. Concepto

Consiste básicamente en un sistema de computación que está diseñado para ejecutar una o algunas funciones que son dedicadas con frecuencia en un sistema de computación a tiempo real. Al contrario de como ocurre con ordenadores que poseen un propósito general como un computador personal, así los sistemas embebidos se diseñan prácticamente para cubrir específicas necesidades.

Estos se pueden programar en lenguaje ensamblador del microcontrolador o microprocesador que se encuentra incorporado sobre el mismo, además se utilizan los compiladores específicos que pueden utilizar lenguajes C o C++, existen casos en los que el tiempo de respuesta de la aplicación no es considerado como un factor crítico. En el diseño de un sistema implica ingeniería y técnicas especiales tanto en el diseño electrónico tanto de hardware como software. (Valvano, 2009)

2.1.2. Características

Una característica principal es que un sistema embebido son sistemas reactivos, en si su funcionamiento dependerá de la continuidad e interacción con su entorno el cual determinará las

posibilidades de respuestas al sistema, para esto se utilizan sensores o transductores que emitirán tipos de respuesta dependiendo de la aplicación y las diferentes características que están dado por este tipo de sistemas. A continuación, se muestra en la Tabla 1 un resumen de las características más importantes de un sistema embebido.

Tabla 1. Características importantes en un Sistema Embebido.

CARACTERÍSTICAS	DESCRIPCIÓN
Confiabilidad	<p>Confiabilidad en el tiempo: Mide que trabaje correctamente en el instante $t=0$.</p> <p>Mantenibilidad: Que el sistema vuelva a trabar en el correctamente a fallas.</p> <p>Disponibilidad: Que el sistema funcione en el tiempo t.</p>
Eficiencia	Buen manejo de energía, tamaños reducidos, tamaño reducido del código del sistema, minimización del peso, velocidad de respuesta.
Tiempo Real	La reacción “inmediata” del sistema ante estímulos externos predefinidos.
Interfaces de usuario	La mayoría de los sistemas embebidos poseen alguna manera de interactuar con el usuario como pantallas gráficas, botones, teclados alfanuméricos, sensores, etc.

Fuente: (Ferreira & Arias, 2010)

2.1.3. Entradas

En las diversas aplicaciones existentes de los sistemas embebidos, como parte importante son dispositivos de entradas que se emplean para su ejecución de dichas aplicaciones, gracias a estos dispositivos el procesamiento puede entrar a interactuar con el entorno obteniendo diferentes señales o datos que se emplean para su procesamiento y posterior señal de salida.

2.1.3.1 Sensores

Un sensor es un dispositivo capaz de establecer magnitudes generalmente físicas o químicas con señales o datos al momento de interactuar con el entorno en el que se los ejecute. Existe una gran variedad de sensores que entregan datos analógicos y digitales. Un sensor está diseñado para medir variables como temperatura, humedad, presión, corriente eléctrica entre otros, estos dispositivos han ido desarrollándose en la última década ayudando a la construcción de sistemas embebidos muy eficientes y confiables al entorno en que se lo aplique. (Ferreira & Arias, 2010)

2.1.3.2 Muestreadores:

El propósito de los muestreadores de señal es procesar la información en tiempos definidos por lo que se procede manipulando la señal proveniente del sensor para entregar la información de señales analógicas y convertirse en una señal en el dominio del tiempo. (Ferreira & Arias, 2010)

2.1.3.3 Conversores A/D:

Cuando la señal es procesada o tomados muestras de entrada en un tiempo discreto, esta señal toma infinitud de valores, es decir sigue siendo una señal analógica por lo que los conversores A/D realizan este proceso en dos fases: cuantización y codificación existiendo métodos de conversión, dependerá mucho de las velocidades y la resolución requerida. (Ferreira & Arias, 2010)

2.1.4. CPU

La unidad central de procesamiento y sus componentes pueden brindar un entendimiento de toda la estructura que forma parte de las arquitecturas que existe, justificando el enfoque del curso al emplear microcontroladores.

Se puede observar en el diagrama de flujo que muestra la Figura. 1 de una manera más técnica como forman parte los componentes en una CPU.

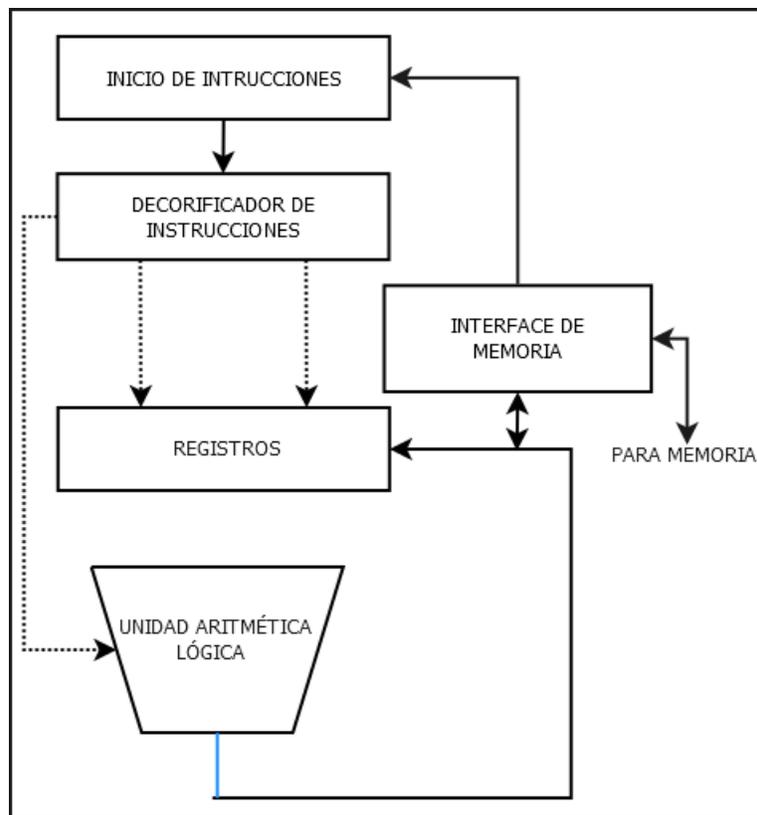


Figura. 1 Diagrama de Flujo de sus componentes de una CPU.
Fuente: (Amdahl, 2012)

Entre los componentes que conforman la unidad central de procesamiento tenemos:

2.1.4.1 *Unidad Aritmética Lógica(ALU)*

Es la encargada de operar funciones tanto aritméticas como lógicas sobre los operadores que provienen de la memoria principal, estos pueden almacenarse de forma temporal en los registros que forman parte de la unidad para diversos propósitos. Existen diferentes tipos y esto dependerá de la aplicación por la cual se encuentran especializadas.

2.1.4.2 *Registros*

Básicamente es una memoria de alta velocidad que se encuentra integrada en el microprocesador que permite guardar o acceder a valores muy utilizados en diversas operaciones, es la manera más rápida que tiene un sistema para almacenar datos y su capacidad dependerá del número de bits que almacena, por ejemplo, registros de 8 o 32 bits.

2.1.4.3 *Unidad de Control*

Forma parte de la unidad de procesamiento que su función es leer las instrucciones en lenguaje máquina que se almacenan en la memoria principal además de generar señales necesarias para controlar el manejo y coordinación y se encuentra conformada los siguientes puntos detallados en la Tabla 2 que se muestra a continuación.

Tabla 2. *Parámetros que conforma una unidad de control en un Sistema Embebido.*

Paramento	Definición
Contador de programa o PC	Es un registro interno del microprocesador en el que se almacena la dirección de la instrucción que se está ejecutando
Registro de instrucción	Al igual que el PC, el registro de instrucción forma parte de la unidad de control y contiene la instrucción que se está ejecutando en cada momento
Decodificador de instrucciones	Se encarga de decodificar la instrucción para interpretar el tipo de instrucción a ejecutar que se encuentra en el registro de instrucción
Reloj Interno	Señal de reloj a una frecuencia específica que marca cada ciclo de ejecución del procesador
Secuenciador	Encargada de generar señales de control para la ejecución de una instrucción que se ha decodificado previamente y comparado con las instrucciones disponibles de la CPU

Fuente. (Ferreira & Arias, 2010)

2.1.4.4 Unidad de Ejecución

Forma parte de la unidad central de procesamiento realizando operaciones y cálculos invocados por los programas que se encuentran en ejecución, generalmente tiene su propia unidad de control, registros entre otros componentes.

2.1.4.5 Memorias

Existen gran variedad de memorias, aunque dependerá de las diversas características que posea una memoria para su función entre ellas tenemos memorias: RAM, DRAM, Memoria Caché, ROM, PROM, EPROM, Memorias FLASH, MMU.

2.1.5. Arduino

Es una plataforma electrónica desarrollada de computación física de código abierto, basado en una placa con un microcontrolador y un entorno de desarrollo de software(programas) para la placa, estos programas se implementan tomando uso de un entorno de desarrollo de programación propio de Arduino. Las placas Arduino son muy utilizadas como componentes electrónicos además de convertir datos analógicos a digitales para el desarrollo de pequeños y grandes sistemas innovadores. (Baeza, 2010)

Gracias al modelo abierto de Arduino se han generado una infinidad de versiones y clones del mismo, dando a conocer un gran abanico de opciones y presentaciones en el mercado, se puede decir que Arduino está conformado por 3 cosas:

- Una placa hardware
- Un software de desarrollo
- Un lenguaje de programación

2.1.5.1 Placa hardware

Placa de circuito impreso que está conformada por distintos pines o puertos que son utilizados para una función como puertos de Entrada o Salida para uso de los sensores de comunicación, además de que incorpora un microcontrolador siendo un modelo estándar del fabricante Atmel, los cuales vienen pre-programados con bootloader o gestores de arranque para poder utilizarlos de una manera sencilla, uno de los diseños de estas placas se muestra en la Figura. 2 que resalto a continuación. (González, 2014)



Figura. 2 Ejemplo de placa Arduino
Fuente: (Cytron Technologies, 2011)

En la actualidad se han desarrollado y existen 17 plataformas de Arduino, que ha crecido considerablemente para ciertos fines o características en común como niveles de entrada, plataformas que ofrecen mejores características, Internet de las cosas y sistemas wearables. En la siguiente Tabla 3 se explica los grupos, plataformas y características importantes además a que fines de desarrollo son empleados.

Tabla 3. Plataformas de Arduino

GRUPO	PLATAFORMA	CARACTERÍSTICAS
NIVELES DE ENTRADA	ARDUINO UNO	Procesador: ATmega328P Velocidad de CPU: 16 MHz E/S Análogas: 6/0 IO digitales/PWM: 14/6
	ARDUINO LEONARDO	Procesador: ATmega32U4 Velocidad de CPU: 16 MHz E/S Análogas: 12/0 IO digitales/PWM: 20/7
	ARDUINO 101	Procesador: Intel Curie Velocidad de CPU: 32 MHz E/S Análogas: 6/0 IO digitales/PWM: 14/4
	ARDUINO ROBOT	Procesador: ATmega32u4 Velocidad de CPU: 16 MHz E/S Análogas: 6/0 IO digitales/PWM: 20/6
	ARDUINO ESPLORA	Procesador: ATmega328u4 Velocidad de CPU: 16 MHz USB: Micro
MEJORAR CARACTERÍSTICAS	ARDUINO MEGA	Procesador: ATmega2560 Velocidad de CPU: 16 MHz E/S Análogas: 16/0 IO digitales/PWM: 54/15
	ARDUINO CERO	Procesador: ATSAMD21G18 Velocidad de CPU: 48 MHz E/S Análogas: 6/1 IO digitales/PWM: 14/10
	ARDUINO DUE	Procesador: ATmelSAM3x8E Velocidad de CPU: 84 MHz
	ARDUINO MEGA ADK	Procesador: ATmega2560 Velocidad de CPU: 16 MHz E/S Análogas: 16/0 IO digitales/PWM: 54/15

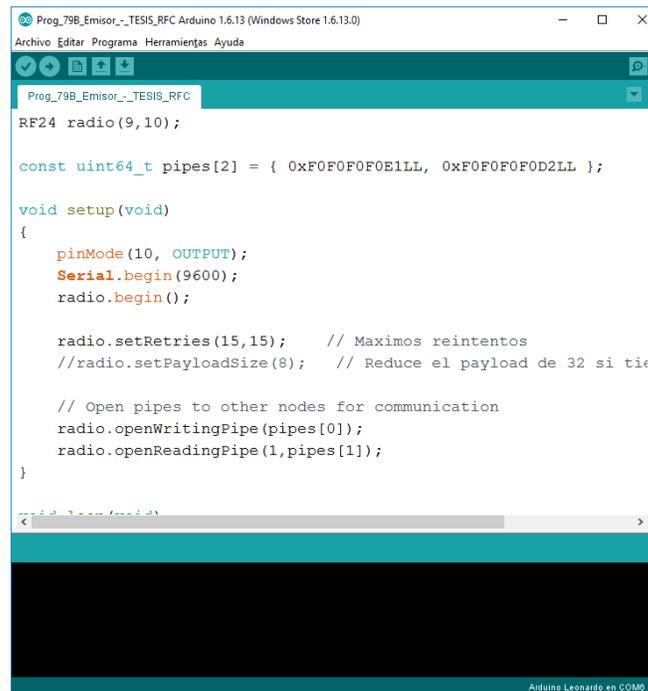
	ARDUINO PRO	Procesador: ATmega168 Velocidad de CPU: 8-16 MHz E/S Análogas: 6/0 IO digitales/PWM: 14/6
INTERNET DE LAS COSAS	ARDUINO YUN	Procesador: ATmega32U4 Velocidad de CPU: 16 a 400 MHz E/S Análogas: 12/0 IO digitales/PWM: 20/7
	ARDUINO ETHERNET	Procesador: ATmega328P Velocidad de CPU: 16 MHz E/S Análogas: 6/0 IO digitales/PWM: 14/4
	ARDUINO GEMMA	Procesador: ATtiny85 Velocidad de CPU: 8 MHz E/S Análogas: 1/0 IO digitales/PWM: 3/2
WEARABLE	LILYPAD ARDUINO USB	Procesador: ATmega32U4 Velocidad de CPU: 8 MHz E/S Análogas: 4/0 IO digitales/PWM: 9/4
	LILYPAD ARDUINO PLACA	Procesador: ATmega328P Velocidad de CPU: 8 MHz E/S Análogas: 4/0 IO digitales/PWM: 9/4
	LILYPAD ARDUINO SIMPLE	Procesador: ATmega168V - ATmega328P Velocidad de CPU: 8 MHz E/S Análogas: 6/0 IO digitales/PWM: 14/6

Fuente: (ARDUINO.cl, 2017)

2.1.5.2 *Software de desarrollo*

Es un entorno de desarrollo libre, multiplataforma y funcional en sistemas operativos Windows, Linux y MacOs, en el cual se puede programar en la placa que se esté utilizando, cuenta con librerías que facilitan el desarrollo de sistemas. Se puede compilar y programar la placa que

esté conectada a nuestro ordenador mediante un cable USB permitiendo la alimentación de la placa y la conexión al software de programación, es muy amigable y de forma sencilla de utilizar utilizando un IDE de carga del programa a la placa.



```

Prog_79B_Emisor_-_TESIS_RFC Arduino 1.6.13 (Windows Store 1.6.13.0)
Archivo Editar Programa Herramientas Ayuda
Prog_79B_Emisor_-_TESIS_RFC
RF24 radio(9,10);

const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };

void setup(void)
{
  pinMode(10, OUTPUT);
  Serial.begin(9600);
  radio.begin();

  radio.setRetries(15,15); // Maximos reintentos
  //radio.setPayloadSize(8); // Reduce el payload de 32 si tie

  // Open pipes to other nodes for communication
  radio.openWritingPipe(pipes[0]);
  radio.openReadingPipe(1,pipes[1]);
}
  
```

Figura. 3. Entorno de Software de desarrollo
Fuente: Propia

2.1.5.3 Lenguaje de programación

El lenguaje de Arduino tiene una gran similitud con C++ ya que este comparte las mismas estructuras de sintaxis además de control y operadores entre otros, es mucho más adaptable y fácil de manejar y adaptarse a él.

2.1.6. Salida

Para la relación de salidas en Sistemas Embebidos refiere a emplear dispositivos análogos y digitales, se tiene muy en cuenta que para dispositivos análogos estas salidas deben ser

procesadas por la correlación Digital a Análogos. Entre elementos o dispositivos de salida que se emplean en este tipo de sistemas tenemos:

2.1.6.1 *Dispositivos electromecánicos*

Estos dispositivos combinan la parte eléctrica como mecánica con la finalidad de controlar sistemas que pueden activarse o desactivarse ya sea manual o automáticamente, pueden ser relays, bobinas, servomecanismos, interruptores, termostatos, motores entre muchos elementos. Los dispositivos interactúan bajo control pueden ser previamente programados con un fin único para una eficiencia ante cambio percibidos. (Francisco, 2014)

2.1.6.2 *Elementos de Visualización*

Forman parte primordial en sistemas embebidos el cual despliega información al usuario relevante para la toma de decisiones o acciones, sirve como interfaz para la programación de dispositivos y otros elementos. Dispositivos de gran variedad como display de 7 segmentos o alfanumérico a partir de leds, además de LCD que genera una visualización más desarrollada como dispositivos de salida en un sistema entre otros.

2.1.7. Comunicación

Al manejar sistemas embebidos, involucra una gran cantidad de datos que deben disponerse para la comunicación a través de canales, estos son entidades abstractas muy esenciales en sistemas de comunicación con parámetros de gran importancia como, capacidad máxima de comunicación o probabilidad de error. Para propagar o enviar información se necesita de un medio en común dependiendo de las propiedades y propósitos del sistema, existen propiedades para la propagación

de datos que pueden ser por medios cableados, medios ópticos o medios inalámbricos como fibra óptica, ethernet, radio frecuencia, bluetooth entre otros. (Miñarro, 2009)

Para una correcta comunicación entre sistemas embebidos existen parámetros importantes que se toman muy en cuenta, estos parámetros están referidos en la Tabla 4 que se muestra a continuación.

Tabla 4. Parámetros de comunicación entre sistemas.

PARÁMENTROS	IMPORTANCIA
Eficiencia	Tiene un sentido económico al momento de implementar la comunicación entre diferentes sistemas por lo que el desarrollo de sistemas eficientes es importante por los diferentes componentes de un sistema.
Ancho de banda apropiado	Varía dependiendo del campo aplicativo entre sistemas ya que es muy importante evitar retrasos en la comunicación.
Privacidad	La privacidad en la comunicación es necesaria en sistemas por lo que usan diferentes técnicas de seguridad.
Robustez	Los sistemas deben poseer suficiente robustez para mantener una comunicación confiable.
Tolerancia a Fallas	Un sistema debe recuperarse de fallas de comunicación de forma rápida y para esto existen mecanismos para solucionarlo.

Fuente: (Ferreira & Arias, 2010)

2.1.7.1 *Comunicación inalámbrica*

En los últimos años se han establecido y desarrollado importantes medios de comunicación inalámbricas diseñados en sistemas de bajo consumo, pero con gran rendimiento. Cuando sensores se juntan formando un fenómeno de datos o información imprescindible en un sistema se forma

una red de comunicación, pero en el caso de que se encuentre en un área geográfica amplia o de gran distancia entre sistemas se necesita de dispositivos o elementos que hagan más económico, eficiente e ideal una correcta comunicación en este caso inalámbricas, en si consiste en elementos de sensores, de cómputo y de computación que ayude a monitorear y tomar decisiones eficientes a un evento de un suceso, a esto se lo llama una red inalámbricas de sensores. Está diseñado con requisitos considerables en un sistema como tolerancia a fallas, restricciones, limitaciones de capacidad, costo, topología, escalabilidad o consumo de energía. (Pahdaman, 2014)

Sistemas embebidos que forman redes de comunicación inalámbricas han desarrollado una gran variedad de aplicaciones con necesidades y características distintas en nodos que pueden detectar el entorno y comunicar con los nodos vecinos realizando cálculos básicos con todo lo obtenido. Algunas de las áreas funcionales de estos sistemas son: medicina, tecnología militar, sensores ambientales, domótica hasta en industrias actuando como pequeños sistemas recopilando datos y analizando o realizando eventos a éstos.

2.1.8. WiFi (Wireless Fidelity)

WiFi trabaja en el estándar IEEE¹ 802.11, este fue diseñado con el objetivo de complementarse con las conexiones de cable en redes LAN² o de poco alcance como en oficinas o casas típicamente, aunque también es muy utilizado para largas distancias por el alcance de esta tecnología, tiene una amplia gama en el mercado y un impacto muy importante en sistemas de comunicación. Ya que nació en el grupo IEEE 802.11 tenía como meta la creación de estándares que pudieran construir una red inalámbrica y esto fue hasta 1997 que el estándar surgió como tal,

¹ Instituto de Ingenieros Eléctricos y Electrónicos

² Redes de Área Local

desde ese entonces se han desarrollado otros grupos o derivados los cuales se designan para mejorar las características de la tecnología específicamente. (Nuñez, Peña, & Garzón)

Al pasar los años se han consolidado como estándares de IEEE 802.11 con tareas específicas. Se puede observar la familia de protocolos más usados en este estándar en la siguiente tabla.

En la presente Tabla 5 se resume los conceptos de la familia de estándares IEEE 802.11 por los cuales se rige la tecnología WiFi.

Tabla 5. Familia de protocolos IEEE 802.11

ESTÁNDAR	DESCRIPCIÓN
802.11	Estándar WLAN original Soporta de 1 a 2 Mbps
802.11a	Estándar WLAN de altas velocidades. Soporta hasta 54Mbps.
802.11b	Estándar WLAN. Banda de 2.4GHz Soporta 11Mbps
802.11e	Está dirigido a los requerimientos de Calidad de servicio para todas las interfaces IEEE WLAN de radio.
802.11f	Define la comunicación entre puntos de acceso para facilitar redes WLAN de diferentes proveedores.
802.11g	Técnicas entre punto de acceso para facilitar redes WLAN

802.11i

Esta dirigido a abatir la vulnerabilidad actual en la seguridad para protocolos.

Fuente: (Núñez, Peña, & Farzón)

Una gran cualidad de WiFi es permitir una aplica gama de aplicaciones que se extiende en todas las áreas además de beneficios que ayudan a la interconexión o envío de mayor cantidad de datos, su interoperabilidad con sistemas dependerá de las funcionalidades originalmente creadas por el fabricante. Es por ello por lo que dispositivos existentes certifican que soporta y proporción de confianza nos da esta tecnología.

2.1.8.1 *Funcionamiento*

Existen módulos o dispositivos WiFi que tienen la capacidad de describir las diversas redes inalámbricas, utilizando ondas de radio como nivel físico de transporte, gracias a esto puede atravesar paredes o suelos deduciéndolo como un medio público al cual se puede tener un acceso a una red o independientemente a otros fines. Ya que esta red puede ser vulnerable ataques se establece en algunos casos políticas de seguridad que se adaptarían a un uso más reservado.

2.1.9. BLUETOOTH

Dentro de las redes inalámbricas existe la tecnología Bluetooth que cada vez se encuentran en más dispositivos y con gran variedad ya que este posee algunas características superiores en otras aplicaciones, bluetooth posee un enfoque omnidireccional que permite que exista una comunicación entre varios dispositivos sin necesidad de poseer línea de vista.

2.1.9.1 Características

Bluetooth posee la capacidad de formar pico redes entre varios dispositivos que se encuentren en la red de operación además de su facilidad de penetrar objetos, este permite una conexión a distancias de hasta 10 metros, es la flexibilidad y su naturaleza omnidireccional lo que ha hecho que puedan instalarse repetidores para alcanzar distancias de hasta 100 metros.

El estándar IEEE 802.15.1 utiliza tecnología inalámbrica Bluetooth presente en una WPAN, este utiliza un radio de corto alcance que ha sido optimizado para obtener un mayor ahorro de energía, es capaz de soportar canales síncronos de comunicación para telefonía de voz y canales asíncronas para comunicación de datos permitiendo una amplia gama de aplicaciones con gran desempeño en dicha tecnología, en forma resumida se presenta las características principales que conlleva Bluetooth en la Tabla 6.

Tabla 6. Características Bluetooth

Características	Bluetooth
Consumo de corriente (mA)	65 – 170
Capacidad de red(nodos)	30
Vida útil de batería(días)	7
Frecuencia de radio(Ghz)	2.4
Cobertura(metros)	10
Potencia de transmisión	1-100
Tecnología	Spread Spectrum

Fuente: (Roque, Zaldívar, & Olimpia, 2013)

2.1.9.2 Funcionamiento

Bluetooth funciona cuando cada dispositivo cuenta con un microchip CMOS³ que trasmite y recibe señales estimadas en frecuencias de 2.4 Ghz estos contienen 48bits trabajando dentro del rango de los 10 metros, cuando se lleva la transmisión de datos se da intercambiando en cada slot entre la unidad maestro y uno de esclavo, estos paquetes contienen acceso que corresponde a la identidad, cabecera dirección de control de acceso al medio, tipo de paquetes, bits de control, cabecera de error los cuales pueden cubrir los slots de transmisión, sus nodos se comunican como se muestra en la Figura. 4 que se muestra a continuación. (Ramírez, s.f.)

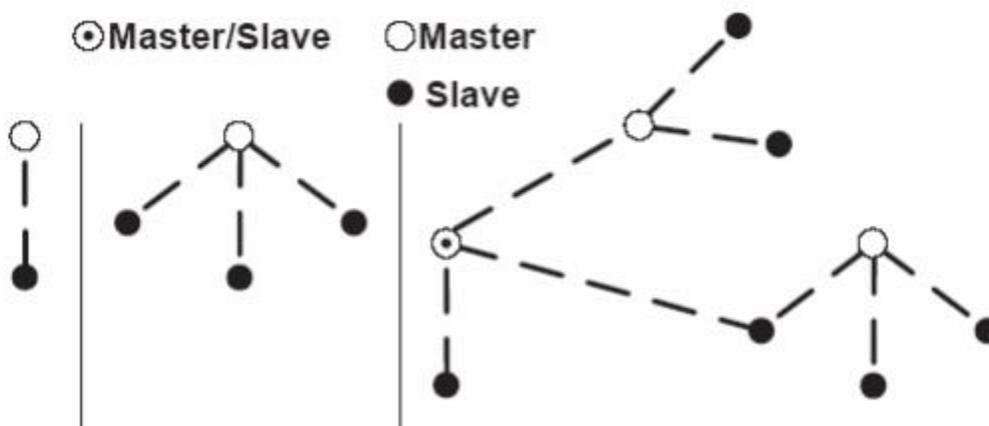


Figura. 4 Formación de operación en el funcionamiento de Bluetooth.

Fuente: (UDLAP, 2011)

2.1.9.3 Asignación de espectro

La tecnología Bluetooth trabaja en la banda mundial sin licencia ISM⁴ de 2.4 GHz, pertenece la banda UHF (Ultra High Frequency) abarca a los 300 MHz a 3 GHz como indica la Figura. 5 que se muestra en el siguiente apartado.

³ Complemento de óxido de metal semiconductor.

⁴ Industrial, científico y médico.

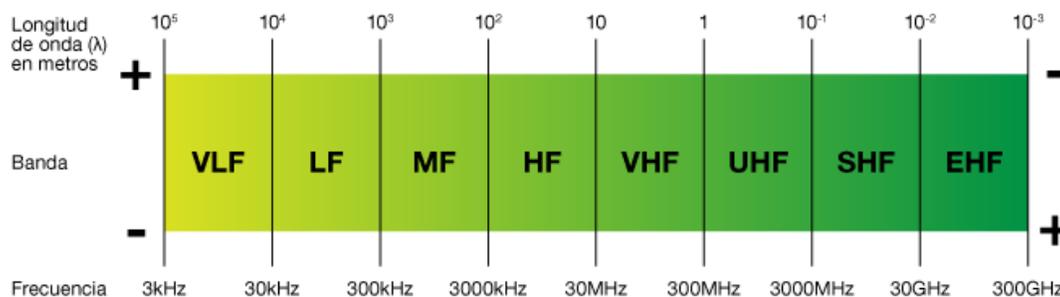


Figura. 5 Localización del espectro mediante su longitud de onda operativa.
Fuente: (UDLAP, 2011)

El estándar Bluetooth en particular ocupa rangos de 2.4 a 2.484 GHz, al operar en bandas que no necesitan de una licencia esta tecnología puede ser muy barata debido a que no requiere pagar por la utilización de esta parte del espectro y al ser definido una banda mundial.

2.1.10. ZigBee

Es un estándar que define protocolos en conjunto para redes inalámbricas de corta distancia baja velocidad, fue desarrollado por Alianza ZigBee y adopto el estándar IEEE 802.15.4 para sus 2 primeras capas: Capa Física y subcapa de acceso al medio.

2.1.10.1 Características generales

Los propósitos de diseño del estándar Zigbee se presentan en la Tabla 7 mencionando las características de gran importancia.

Tabla 7. Características ZigBee

Características	Zigbee
Rango(metros)	10 – 100
Perfil de tiempo	Años
Topología	Árbol, Estrella, Malla
Consumo de transmisión	30mA
Técnica de Modulación	OQPSK
Energía	Bajo consumo de potencia
Bandas	2,4 GHz
Taza de transmisión	250Kbps

Fuente: (Jorge, 2011)

Gracias a las características que brinda esta tecnología es idónea para formar redes de sensores y obtener una gran aplicabilidad en dispositivos actuales o en campos como: Domótica, automatización industrial, reconocimiento remoto, medicina, seguridad, sistema de riego automático entre otros.

2.1.10.2 Tipos de dispositivos

Existen 3 tipos de dispositivos que juegan un rol importante en una red de sensores:

Coordinador Zigbee(ZC): Este dispositivo debe existir uno por red ya que su funcionalidad es encargarse de controlar la red y los caminos que deben seguir, requiere de gran memoria y capacidad computacional.

Router Zigbee(ZR): Su funcionalidad es interconectar los dispositivos separado en la topología de red además que ofrece un nivel de aplicación.

Dispositivo Final(ZED): Posee funciones de comunicación entre su nodo padre pero no puede transmitir información destinada a otros dispositivos.

2.1.10.3 Funcionalidad

Este estándar define dos tipos de funcionalidad con el principal objetivo de reducir costos:

FFD (Full Function Device): Son capaces de realizar su función en cualquier topología estos pueden ser coordinadores o coordinadores de red, brindando la funcionalidad de que los dispositivos puedan dialogar con cualquier otro. (Garrido)

RFD (Reduced Function Device): En este tipo de dispositivos pueden solo ser miembros de una red con topología estrella, solo pueden conectarse con el coordinador de red, por lo que son dispositivos de baja complejidad. (Garrido)

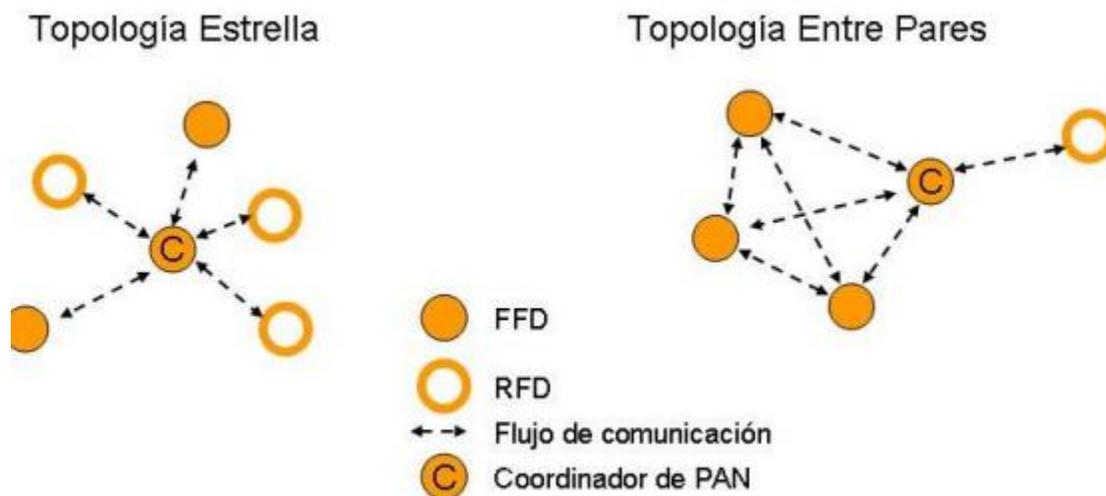


Figura. 6 Topologías en IEEE 802.15.4 en estándar Zigbee.
Fuente: (Jorge, 2011)

2.1.10.4 Modos de funcionamiento

El protocolo MAC soporta dos modos de funcionamiento:

Modo balizado: Es generado periódicamente por el coordinador y distribuida por toda la red gracias a los router, sirve para sincronizar todos los nodos de la red, de modo que estos puedan despertarse en un momento determinado, envía los datos almacenados y vuelta al modo de ahorro de energía. Así tanto el coordinador como los routers y los dispositivos pueden pasar gran parte del tiempo en modo de bajo consumo. (Garrido)

Modo no balizado: En este modo los dispositivos no están sincronizados unos con otros. Así únicamente los dispositivos finales pueden entrar en el modo sleep mientras que los router y el coordinador deben estar continuamente con el sistema en modo recepción así estar preparados para recibir datos en cualquier momento. Este modo es más simple, pero hace que gran parte de nodos tengan un mayor consumo energético. (Garrido)

2.1.11. RF (Radio Frecuencia)

Radiofrecuencia, también denominado espectro de radiofrecuencia, ondas de radio o RF, se aplica a la porción menos energética del espectro electromagnético, situada entre unos 3 Hz y unos 300 GHz. Las ondas electromagnéticas de esta región del espectro se pueden transmitir aplicando la corriente alterna originada en un generador a una antena.

2.1.11.1 Ondas de Radio

Las ondas de radio tienen longitudes que van de tan sólo unos cuantos milímetros (décimas de pulgadas), y pueden llegar a ser tan extensas que alcanzan cientos de kilómetros (cientos de millas). En comparación, la luz visible tiene longitudes de onda en el rango de 400 a 700 nanómetros, aproximadamente 5 000 menos que la longitud de onda de las ondas de radio. Las ondas de radio oscilan en frecuencias entre unos cuantos kilohertz (kHz) y unos cuantos terahertz (THz).

Varias frecuencias de ondas de radio se usan para la televisión y emisiones de radio FM y AM, comunicaciones militares, teléfonos celulares, radioaficionados, redes inalámbricas de computadoras, y otras numerosas aplicaciones de comunicaciones. La mayoría de las ondas de radio pasan libremente a través de la atmósfera de la Tierra. Sin embargo, algunas frecuencias pueden ser reflejadas o absorbidas por las partículas cargadas de la ionosfera.

2.1.11.2 Transmisión y recepción

Una onda de radio se origina cuando una partícula cargada (por ejemplo, un electrón) se excita a una frecuencia situada en la zona de radiofrecuencia (RF) del espectro electromagnético. Otros tipos de emisiones que caen fuera de la gama de RF son los rayos gamma, los rayos X, los rayos infrarrojos, los rayos ultravioletas y la luz.

Cuando la onda de radio actúa sobre un conductor eléctrico (la antena), induce en él un movimiento de la carga eléctrica (corriente eléctrica) que puede ser transformado en señales de audio u otro tipo de señales portadoras de información.

El emisor tiene como función producir una onda portadora, cuyas características son modificadas en función de las señales (audio o video) a transmitir. Propaga la onda portadora así modulada. El receptor capta la onda y la “desmodula” para hacer llegar al espectador auditor tan solo la señal transmitida.

2.1.12. Nodo sensor inalámbrico (Mote)

Un nodo sensor inalámbrico, también llamado Mote, constará con subsistemas que formará un sistema embebido o sistema computacional de baja capacidad como muestra la Figura. 7.

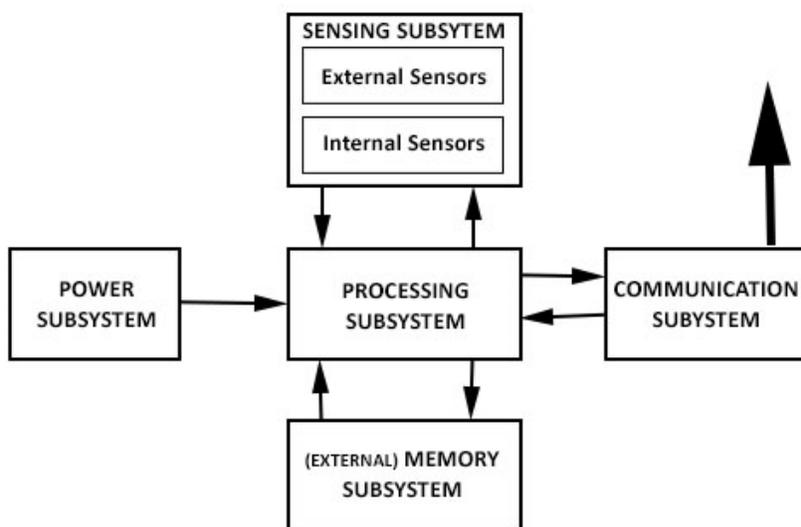


Figura. 7 Estructura mediante un diagrama de bloques de un MOTE.
Fuente: (IEEE Design & Test of Computers, 2001)

2.1.12.1 Subsistema de procesamiento

Es la CPU de un nodo sensor inalámbrico. También se conoce como unidad controladora. El controlador más comúnmente utilizado en el nodo sensor inalámbrico es el microcontrolador, ya que sus características como tamaño pequeño, un bajo consumo de potencia y facilidad de programación. Y, la mayoría de los microcontroladores tienen un convertidor analógico

incorporado a digital (ADC), DSP (Digital Signal Processor) y (Field Programmable Gate Array) FPGA.

2.1.12.2 Subsistema de comunicación

Consiste en la antena inalámbrica. Las diversas tecnologías de comunicación utilizados en WSN son ZigBee (IEEE 802.15.4), WiFi, RF y Bluetooth ya que son eficientes al momento de ensamblar o realizar un sistema embebido.

2.1.12.3 Subsistema de energía

Las baterías son la principal fuente de suministro de energía para un nodo sensor inalámbrico. Debido a la restricción de tamaño, las baterías usadas también tienen que estar en pequeño tamaño, lo que implica que el tiempo de vida de un nodo sensor inalámbrico será relativamente baja.

2.1.12.4 Subsistema de detección

Esta unidad consta de sensores, que son generalmente pequeños dispositivos que son capaces producir alguna respuesta medible (a menudo un cambio en el voltaje). Estos por lo general la respuesta es alimentada a un convertidor de analógica digital (ADC). La señal digitalizada se envía entonces a la unidad de procesamiento para el procesamiento posterior.

2.2. Control de acceso al medio

Las arquitecturas de redes en el nivel de acceso al medio es el encargado de convertir la información suministrada por el nivel de red en señales que serán transmitidas, esto tiene gran participación con algoritmos que determinan como las estaciones comparten el canal.

Para el control de acceso al medio se debe disponer de medios para solucionar problemas relacionados con los errores en la transmisión. Para ello, se dividen los datos suministrados por el nivel de red, que son transmitidos de forma secuencial y que llevan asociado un acuse de recibo que también gestiona la capa de acceso al medio. Puesto que el nivel físico no interpreta (únicamente los procesa como información que enviar o recibir), es el nivel de acceso al medio el que debe poner inicio y fin a los marcos que envía o recibe, asignando secuencias especiales de bits al inicio y al final de cada marco, que sean fácilmente reconocibles. Otro problema que debe resolver la capa de acceso al medio es el supuesto en que un emisor veloz pueda llegar a saturar a un receptor más lento.

2.2.1. PROTOCOLO ALOHA

Aloha es un sistema de redes desarrollado en la Universidad de Hawái, sus inicios dieron en 1970 por Norman Abramson y fue diseñada con el objetivo de permitir a personas acceder a los principales sistemas o nodo principal en una secuencia o serie de nodos. Este se basa en usar un medio compartido para transmitir en el que se usa mismas frecuencias para todos los nodos que estén conectados.

2.2.1.1. Protocolo Aloha Puro

Dependiendo de la distribución de los nodos y el nodo principal y tengan una comunicación se planteó que las diferentes estaciones compartiesen el mismo canal sin preocuparse si este se encuentra libre o no, por ende, si una estación desea transmitir simplemente emite un frame y una vez finalizado quedaba en espera de recibir la confirmación de que si fue receptada correctamente por el destinatario. Si después de un tiempo no se recibe la confirmación el emisor supone que había ocurrido una colisión por lo que se espera un tiempo aleatorio y procede a reenviar nuevamente los frames todo esto funciona como muestra Figura. 8. (Mireya, 2012)

El mayor problema que presenta principalmente este protocolo es que al enviar frames por parte de los nodos se hace una forma confusa y basta que dos frames colisionen o solapen solo en un bit para que ambos no se reciban y nuevamente se deba retransmitir, ya que los nodos simplemente se percatan del problema después de haber terminado la transmisión.

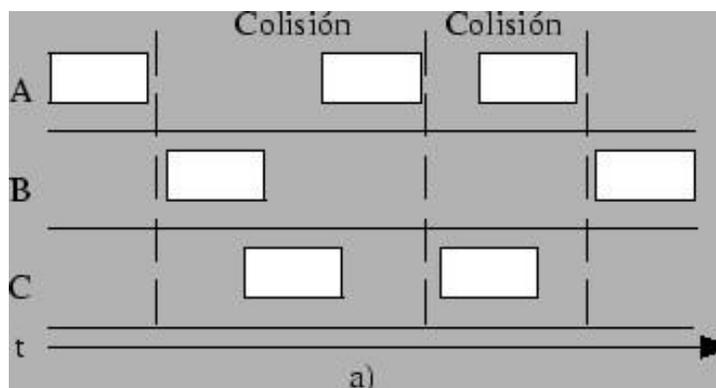


Figura. 8. Análisis Aloha puro de colisiones

Fuente: (Geocities, 2012)

2.2.1.2. Rendimiento de Aloha Puro

Se define el rendimiento como la fracción del tiempo en el que el canal se utiliza para transportar información útil.

Se asume que hay un número infinito de estaciones y el tiempo de transmisión de cada trama es el mismo para todas las estaciones, T . Además, el tráfico generado por las estaciones es de Poisson, con tasa de λ tramas por unidad de tiempo.

Hay que tener en cuenta las retransmisiones en tráfico real generado que será g , con $g \geq \lambda$, ha esto se asume que dicho tráfico se distribuye según la probabilidad de Poisson.

Consideremos un paquete transmitido en un tiempo arbitrario t , dicha transmisión se recibirá correctamente si no se produce ninguna otra en el intervalo dado por $t - T$, $t + T$, siendo este conocido como intervalo de vulnerabilidad. La probabilidad de que una trama se reciba correctamente coincide con la probabilidad de que haya 0 transmisiones en dicho intervalo,

$$P(\text{exito}) = P(0 \text{ transmisiones en } 2T) = e^{-2gT}$$

Con lo que el rendimiento o throughput se calculará como:

$$S = gT e^{-2gT}$$

Definiendo $G = gt$, se llega finalmente a:

$$S = G e^{-2G}$$

En donde G sería la tasa en paquetes por tiempo de transmisión por trama o la tasa cuando se normaliza el tiempo de transmisiones a la unidad, en donde dicho máximo se alcanza cuando se consigue la utilización de canal al 50% esto significa $G=0,5$ obteniendo un rendimiento máximo de $S = 0,184$ es decir 18,4% como se observará en la Figura. 9, esto significa que el 81.6% del total disponible de ancho de banda se está desperdiciando básicamente debido a estaciones tratando de emitir al mismo tiempo.

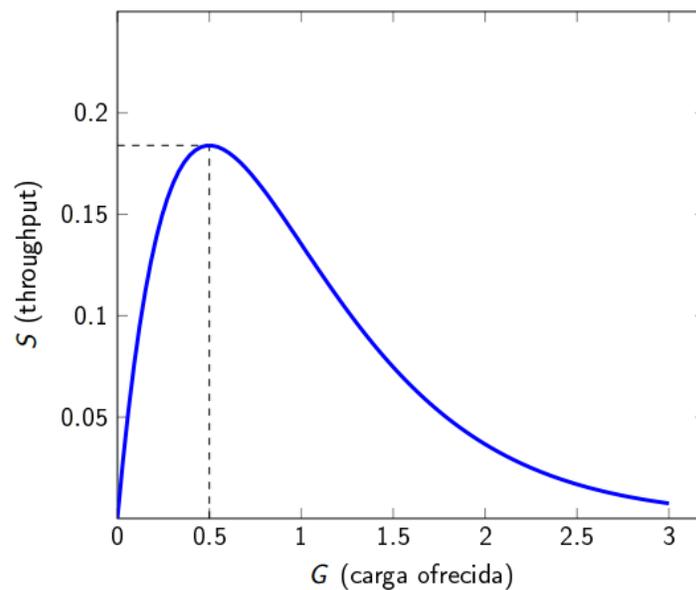


Figura. 9. Rendimiento de Aloha puro representado en un gráfico en línea.
Fuente: (UNICAN, 2015)

En la Figura. 10, se puede apreciar con mayor claridad como es el procedimiento o comportamiento en un protocolo Aloha puro.

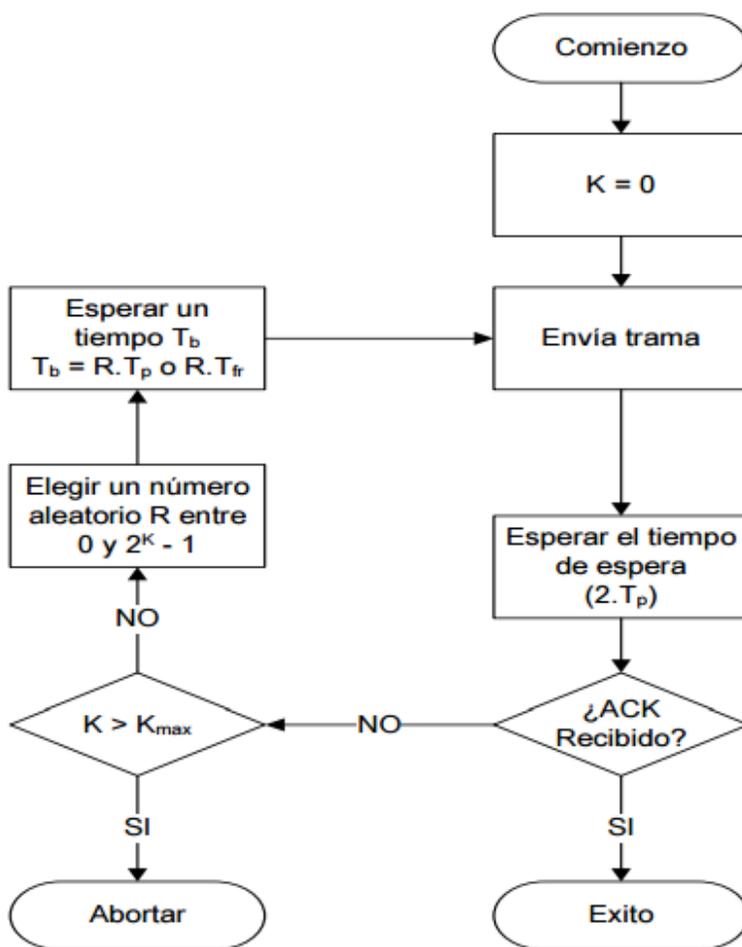


Figura. 10. Diagrama de flujos de Aloha puro.
Fuente. (Redes, 2006)

2.2.1.3. Protocolo Aloha Ranurado

Para realizar unas mejoras de las prestaciones que realiza Aloha se definió un avance con Aloha Ranurado (Slotted) en 1972 con la diferencia que en este caso los nodos pueden transmitir en unos determinados instantes de tiempo o slot, este sincronismo favorece de manera importante cuando una terminal necesite transmitir debe esperar al inicio de un nuevo periodo para realizarlo, provocando de esta manera reducir colisiones a diferencia de Aloha puro como se muestra en la

Como:

$$S = Ge^{-G}$$

El número medio de trasmisiones necesarias para una trama llegue correctamente es e^G , por el rendimiento máximo se alcanza para $G=1$, dado que aumenta a un 50% más en el protocolo y éste se duplica a 36.8% como se puede apreciar en la Figura. 12.

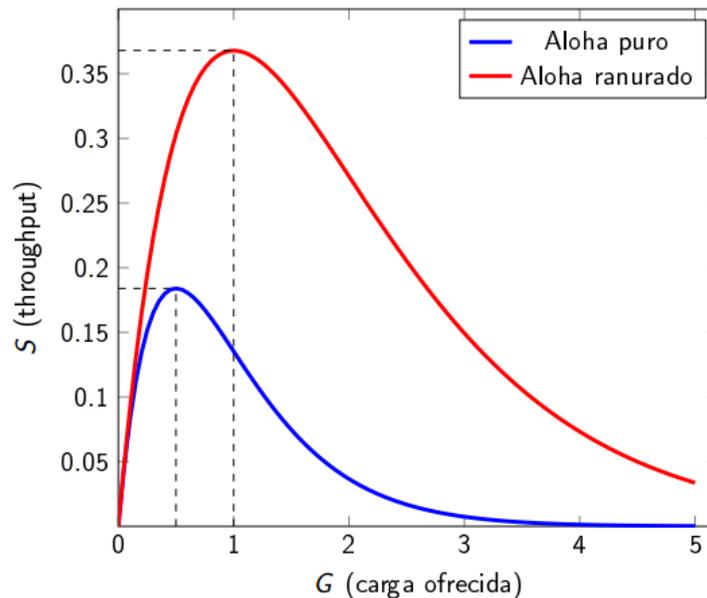


Figura. 12 Diferenciación de rendimientos entre Aloha puro y ranurado mediante un gráfico en línea.
Fuente: (UNICAN, 2015)

2.2.2. CSMA

Acceso Múltiple por detección de portador o CSMA (Carrier Sense Multiple Access) es un protocolo diseñado para reducir la posibilidad de colisiones, comprobando el medio antes de realizar una transmisión, CSMA se basa principalmente en la detección antes de transmitir o escuchar antes de hablar, aunque existe una posibilidad de colisión debido al retardo de propagación.

Aquí se trabaja con métodos de persistencias o métodos desarrollados para dar respuestas a preguntas como ¿Qué debería hacer una estación si el canal está ocupado? o ¿Qué debería hacer una estación si el canal está libre?, el primero método de persistencia hace que la estación encuentre la línea libre y envíe la trama inmediatamente con probabilidad 1, a este método tiene la posibilidad más alta de colisión, otro método es el de no persistencia aquí si la estación encuentra la línea libre, envía las tramas inmediatamente, en casi contrario espera un tiempo y vuelven a comprobar de nuevo el medio reduciendo significativamente colisiones, este método reduce la eficacia de la red debido a que el medio puede permanecer libre cuando hay estaciones con tramas por enviar y por último se ve el método de persistencia este enfoque combina las ventajas de las dos estrategias anteriores, reduciendo la posibilidad de colisión y mejorando la eficacia, Se comprueba si el canal tiene ranuras de tiempo con una duración de la ranura igual o mayor que el tiempo máximo de propagación.

2.2.3. CSMA-CA

Cuando se emplea este algoritmo los dispositivos anuncian que están listos para enviar paquetes de datos antes de acceder al canal. De esta forma se evita la colisión. Dependiendo de unos ciertos parámetros se da prioridad a uno de los candidatos, que podrá acceder al canal para enviar su paquete de datos. El resto de los dispositivos esperarán un tiempo aleatorio (distinto en cada uno de ellos) para volver a intentar acceder al canal. Este fue diseñado para evitar colisiones en redes inalámbricas, debido a que las colisiones no pueden ser detectadas debido a que gran parte de la energía se pierde en la transmisión. Estas colisiones se evitan mediante el uso de tres estrategias como:

2.2.3.1. Espaciado entre trama(IFS):

Es que es cuando un canal está libre la estación espera un periodo de tiempo entre tramas para luego si el canal sigue libre pueda enviar, estos se puede definir la prioridad de una estación o de una trama.

2.2.3.2. Ventana de contención

Es una cantidad de tiempos divididos en ranuras, una estación lista para enviar simplemente elige un número aleatorio de ranuras como “s” tiempo de espera, el número de ranuras cambia de acuerdo con la estrategia de espera binaria exponencial esto significa que se fija a una ranura la primera vez y luego se dobla cada vez que la estación no detecta un canal libre después del tiempo IFS.

2.2.3.3. Confirmaciones

La confirmación positiva y el vencimiento del temporizador ayudan a garantizar que el receptor ha recibido trama. En la siguiente ilustración se puede observar de mejor manera el procedimiento que tiene CSMA/CA y sus parámetros de control.

2.2.3.4. Rendimiento de CSMA/CA

Hay algunos autores que han realizado un análisis de rendimiento de este tipo de sistemas, uno de ellos y con gran relevancia es el de Bianchi asume saturación, ya que todos los nodos tienen en todo momento paquetes para ser transmitidos y propone una cadena de Markov bidimensional para determinar la ranura aleatoria que ha escogido la estación, así como la fase que se encuentra.

Todo esto dependerá del número de estaciones que estén transmitiendo como en función de probabilidad de transmisión. Con lo que se puede calcular el rendimiento, siendo L la longitud de los paquetes, y T_e y T_c la duración de una transmisión exitosa y de una colisión, respectivamente en función a parámetros MAC. Dándonos la siguiente ecuación.

L = Longitud de paquetes

T_e y T_c = Duración de transmisión de éxito y de colisión

S = Probabilidad de éxito

$$S = \frac{P_e * P_{tx} * L}{(1 - P_{tx})\sigma + P_{tx} * p_e * T_e + P_{tx}(1 - P_e)T_c}$$

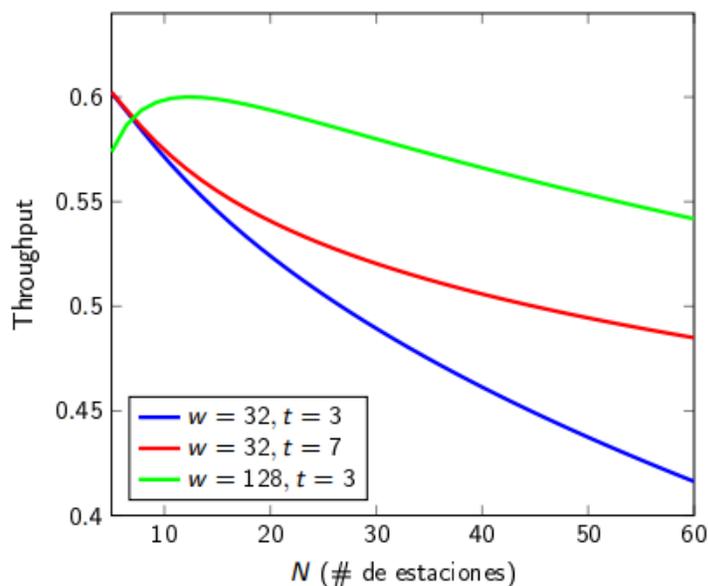


Figura. 13. Throughput en función del número de estaciones
Fuente: (UNICAN, 2015)

Hay que tener en cuenta que en algunos sistemas como tecnologías inalámbricas no es posible detectar colisiones mientras se está transmitiendo, lo que trata el protocolo es evitar las colisiones mediante un proceso como se explica en la siguiente Figura. 14.

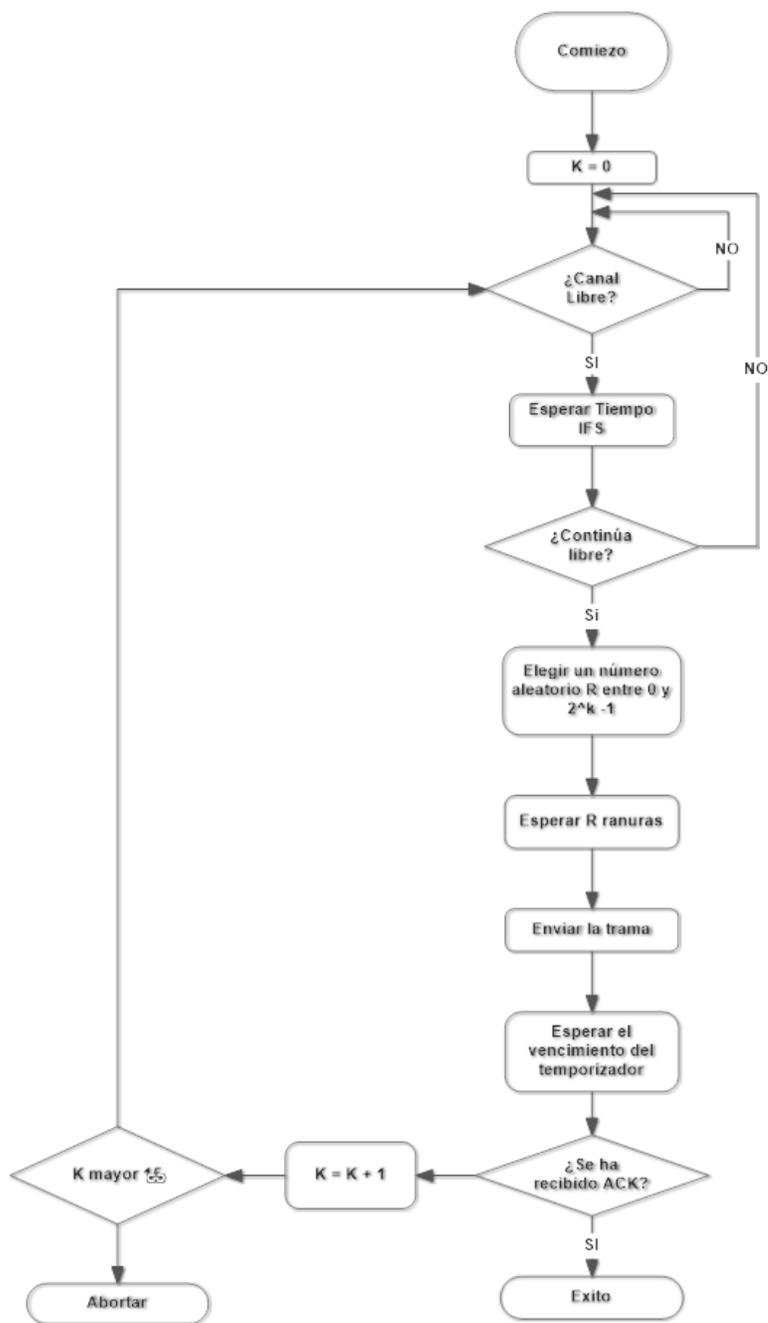


Figura. 14. Diagrama de flujo de CSMA/CA.
Fuente: (UNICAN, 2015)

2.3. Metodología de Investigación

En este tema se especifican métodos o modelos de investigación adecuados para la elaboración del presente proyecto para comparar y tomando en cuenta el mejor de estos, para llevarlo a cabo en el tercer capítulo y obtener una investigación ordenada, eficiente y con gran validez. Se presentan modelos de requerimientos como un apoyo constructivo del sistema embebido ya que en la actualidad no las metodologías de Ingeniería de Requerimientos propuestas no establecen continuidad en su proceso de desarrollo.

2.3.1. Metodología en cascada

Modelo en Cascada o Lineal secuencial, es un enfoque metodológico que ordena rigurosamente las fases del proceso para el desarrollo de un sistema, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior como se detalla en la Figura. 15.

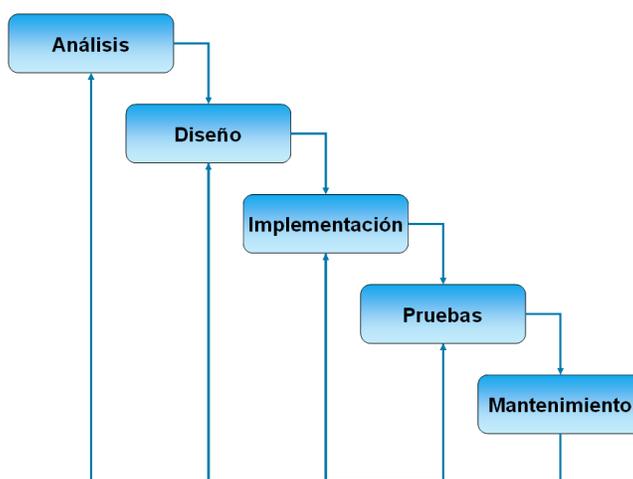


Figura. 15 Diagrama de desarrollo en bloques de una metodología en cascada

Fuente: (Solórzano, 2013)

2.3.2. Fases del modelo en cascada

2.3.2.1. Análisis de requisitos

Se analizan las necesidades de los usuarios finales para determinar qué objetivos debe cumplir. En esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación de lo que debe hacer el sistema sin entrar en detalles internos. Es importante señalar que en esta etapa se debe detallar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas. (Pressman, 2017)

2.3.2.2. Diseño del sistema

En esta fase se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución identificando grandes módulos y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación. (Pressman, 2017)

2.3.2.3. Diseño del Programa

Se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario, así como también los análisis necesarios para saber qué herramientas usar en la fase de codificación. (Pressman, 2017)

2.3.2.4. Fase de codificación

Se implementa el código fuente, haciendo uso de prototipos, así como de pruebas para corregir errores. Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para elaborar un programa con procesos más rápidos. (Pressman, 2017)

2.3.2.5. Fase de pruebas

Los elementos programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de finalizarlo, para ello existen tipos de pruebas como: Pruebas de unidad, Pruebas de integración, Pruebas de sistema o Pruebas de aceptación. (Pressman, 2017)

2.3.2.6. Fase de verificación

Es la fase el usuario final ejecuta el sistema, para ello ya se realizaron exhaustivas pruebas para comprobar que el sistema no falle.

2.3.2.7. Fase de mantenimiento

Una de las etapas más críticas, es el mantenimiento del Software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas. Para ello existen varios tipos de mantenimiento que son preventivos, correctivos o evolutivos. (Pressman, 2017)

2.3.3. Modelo de prototipos

Es también conocido como desarrollo prototipado o modelo de desarrollo evolutivo, se inicia con la definición de los objetivos globales para el sistema, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. (Flores, 2008)

Este modelo se utiliza para dar al usuario una vista preliminar del software. Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba fallo por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho. (Flores, 2008)

2.3.3.1. Etapas

- Recolección y refinamiento de requisitos
- Modelado, diseño rápido
- Construcción del Prototipo
- Desarrollo, evaluación del prototipo por el cliente
- Refinamiento del prototipo
- Producto de Ingeniería
- Cómo se lleva a cabo

Se comienza elaborando un prototipo del producto final, qué aspecto tendrá, cómo funcionará en muchas interfaces de usuario, este modelo puede resultar fácil. Cuanto más próximo se encuentre el prototipo al producto real, mejor será la evaluación, si bien se pueden obtener

magníficos resultados con prototipos de baja fidelidad. En la Figura. 16, se puede observar el inicio y final de las fases de trabajo en este modelo esquematizado. (Flores, 2008)

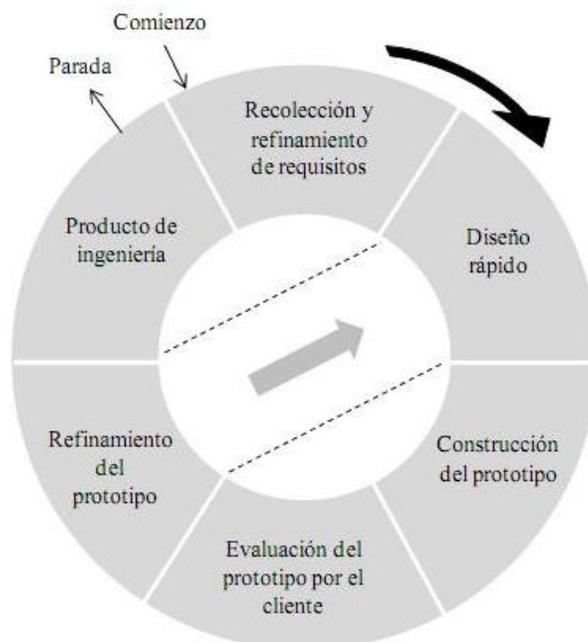


Figura. 16 Fases de trabajo en el Modelo de Prototipos.
Fuente: (Repositorio UPTC, 2015)

2.3.3.2. Ventajas

- No modifica el flujo del ciclo de vida
- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios
- Reduce costo y aumenta la probabilidad de éxito
- Exige disponer de las herramientas adecuadas

Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida. Ofreciendo un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un

algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina. (Flores, 2008)

2.3.4. Metodología en V

Es una variación del modelo en cascada que muestra cómo se relacionan las actividades de pruebas con el análisis y el diseño, la codificación forma el vértice de la V, con el análisis y el diseño a la izquierda y las pruebas y mantenimiento a la derecha como se muestran en la Figura. 17. (Martínez Pérez, 2012)

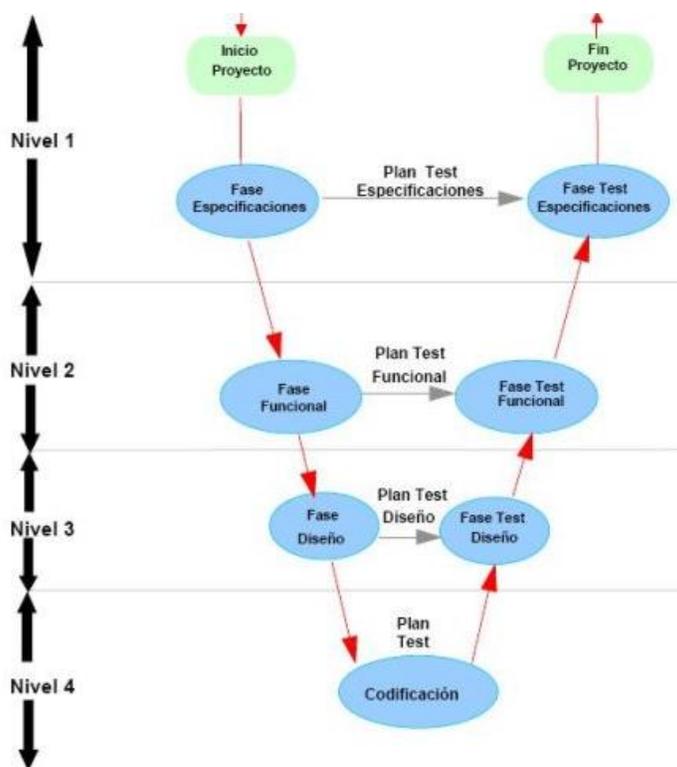


Figura. 17. Niveles para desarrollo de la Metodología en V
Fuente: (Jonathan, 2012)

En cada fase de desarrollo, existe una fase correspondiente o paralela a la verificación o validación. Esta estructura obedece que cada fase de desarrollo debe existir un resultado válido de verificación.

En la misma esquematización se advierte además que la proximidad entre fase de desarrollo y fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V, es decir la longitud de esta separación sea proporcional a la distancia en el tiempo entre fase y verificación. (Martínez Pérez, 2012)

2.3.4.1. Nivel 1

Este nivel se encuentra orientado al cliente ya que el inicio del proyecto y el fin del proyecto contribuyen a los dos extremos del ciclo. Se compone de análisis de requerimiento además de especificaciones que serán expuestas en el documento. (Martínez Pérez, 2012)

2.3.4.2. Nivel 2

El segundo nivel se encuentra orientado a las características funcionales del sistema propuesto. Puede considerarse un sistema como una caja negra y caracterizarla únicamente con funciones que son directas o indirectas visibles por el usuario al finalizar la etapa esto se traduce como el análisis funcional. (Martínez Pérez, 2012)

2.3.4.3. Nivel 3

En este nivel se definen los componentes tanto de hardware como de software del sistema final, a este nivel se lo denomina como arquitectura del sistema.

2.3.4.4. Nivel 4

Este nivel se encuentra orientado a la implementación en la que se desarrolla los elementos unitarios o módulos del programa. (Martínez Pérez, 2012)

En general las metodologías llevan a cabo una serie de procesos comunes que son buenas prácticas para lograr los objetivos antes mencionados independientemente de cómo hayan sido diseñadas.

Las fases que agrupan estos procesos son las siguientes:

- Análisis
- Especificación
- Diseño
- Programación
- Prueba
- Documentación
- Mantenimiento
- Reingeniería

2.3.4.5. Ventajas

Se trata de un proceso ideal, por su robustez, para proyectos pequeños, con equipos de una a cinco personas. También es ideal, por su claridad, para toda esa gente que nunca ha programado siguiendo una metodología. Para el proyecto final de carrera o para ese cliente que te ha conseguido un amigo que te lo pide a ti y no se dirige a una empresa por mayor comodidad, la relación entre

las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos. (Martínez Pérez, 2012)

Capítulo 3. Algoritmo en capa aplicación y pruebas

En el presente capítulo se procede a desarrollar un algoritmo de acceso al medio en capa aplicación para tecnologías Sensor Node Device (MOTE), comparando los algoritmos en estándares de comunicación inalámbrica RF, ZIGBEE, WIFI Y BLUETOOTH en protocolos de acceso CSMA/CA y ALOHA, realizando pruebas pertinentes en cada una de las tecnologías mencionadas, todo esto llevándolo a cabo a través de una plataforma Arduino basado en lenguaje C, para ello se escogerá una metodología de investigación que ayude a sintetizar y desarrollar este capítulo de la mejor manera dado que previamente se realizó una revisión y análisis de estas metodologías.

3.1. Selección de metodología para el desarrollo del algoritmo en capa aplicación y pruebas.

En la actualidad la importancia de adquirir o generar un sistema embebido mediante estándares de comunicación se ha vuelto un tema indiscutible, por lo que el desarrollo de nuevos sistemas se ha vuelto muy relevante, esto ha originado que se establezcan mejoras no solo en la productividad de estos sino en la calidad de un sistema.

Para el desarrollo de sistemas se ha requerido de herramientas o métodos especializados para cumplir con los requerimientos finales tanto software como hardware, dado que el proceso de desarrollo de los sistemas embebidos puede estar lleno de retos.

Por lo tanto, se ha elegido mediante una revisión teórica que la metodología en V cumple con los parámetros desde la especificación, análisis y requerimientos mediante fases funcionales de diseño para la elaboración de este proyecto hasta el mantenimiento de éste.

3.2. METODOLOGÍA EN V: Fase 1 Análisis

Esta fase se encuentra orientada a conocer los factores orientados al cliente o a quien va dirigido gran parte del proyecto, en este caso docentes, estudiantes y tesis que se encuentren orientado a Electrónica y Redes de Comunicación, por lo que es necesario poseer un análisis investigativo que muestre información necesaria acerca de los problemas que intervienen en la investigación para así ayudar a generar un proceso esquematizado en el presente proyecto, utilizando técnicas que determinaran los requerimientos necesarios.

3.2.1. Situación Actual

En este apartado se da a conocer el estado o la situación actual en la población, como parte del proceso de análisis que brindará información importante ayudando a establecer los requerimientos del sistema a desarrollar tanto como hardware, software y componentes esenciales que se dará a lo largo de la investigación.

Dado que el rendimiento de protocolos y arquitecturas en el diseño de redes de sensores inalámbricos son parte importante de investigaciones por la cantidad de datos de información a transmitir y problemas que conlleva ello, nos permite enfocarnos en investigadores, estudiantes, docentes y personas que forman parte o están orientados al proceso de generar conocimiento, aprendizaje y construcción de nuevos sistemas tecnológicos, como parte para la obtención de información útil ya que en la actualidad un sin número de aplicaciones con fines de mejorar el estilo de vida de las personas se genera con mayor crecimiento aumentando el número de dispositivos interconectados y uno de los retos es el desarrollo de una eficiente interoperabilidad

de una red de sensores inalámbricos, siendo necesario realizar un estudio y una posible solución al manejar un mayor volumen de información.

Para la obtención de esta información se realizará la investigación o se tomará como población a estudiantes, investigadores, tesis y docentes de la Universidad Técnica del Norte situados en el sector del Olivo, que se encuentra en la Av. 17 de Julio 5-21 y General José María Córdoba en la Facultad de Ingeniería en Ciencias Aplicadas “Carrera de Ingeniería en Electrónica y Redes de Comunicación” de séptimo a décimo nivel.

La Carrera de Ingeniería en Electrónica y Redes de Comunicación cuenta con una malla curricular que incluye materias referentes a Comunicación Inalámbrica, Sistemas Embebidos, Networking, Sistemas de Trasmisión entre otras y los estudiantes en este proceso se encuentran a fin al tema, además de docentes que brindan el plan investigativo en estas materias y tesis que realizan su proyecto de titulación en redes inalámbricas.

3.2.2. Técnicas de recolección de información útil.

Para la una investigación y análisis adecuado de datos útiles, existen varias técnicas que ayuden al desenvolvimiento o adquisición de estos, una de ellas que se va aplicar es una encuesta a los estudiantes de la Carrera de Ingeniería en Electrónica y Redes de Comunicación, además de una entrevista a docentes y tesis con preguntas muy concretas pero con gran validez, dado que el propósito de dichas técnicas es observar, determinar y analizar un promedio de respuestas en temas de estudios referentes a redes de sensores inalámbricas, algoritmos y métodos de acceso al medio.

3.2.2.1. Entrevista

En la entrevista se busca un dialogo profesional que ayudará a recolectar información útil. Se pretende obtener datos referentes al uso y comunicación de una red de sensores inalámbricos, para de este modo realizar un algoritmo eficiente que ayude al desenvolvimiento de una red de sensores de mejor manera.

Los resultados a obtenerse están sujetos a medidas de nivel de comunicación entre investigadores y docentes que formen parte del tema de estudio, en consecuencia, se plantean preguntas orientadas al tema de trasmisión de información en una red de sensores inalámbricos con gran capacidad mediante protocolos de acceso al medio, con el fin de resolver dudas o cuestionar una previa investigación por lo cual es necesario obtener resultados o información para el desarrollo del presente proyecto.

3.2.2.2. Encuesta

La encuesta aplica a estudiantes de la carrera que ya poseen conocimiento acerca del tema a desarrollar, como problemas en la trasmisión de datos, algoritmos a utilizarse en una red de sensores hasta configuraciones para el desarrollo de un proyecto. Con el fin de obtener datos precisos de los requerimientos de usuario para realizar el diseño del sistema. Para ello se toma una población definida y se una un cálculo de muestra para las encuestas a realizarse.

3.2.3. Cálculo de muestra para encuestas

El cálculo de muestra ayuda a conocer aspectos previos a la investigación y determinará el grado de credibilidad que concederemos al análisis de resultados obtenidos. Para ello se toma como muestra a estudiantes de la Carrera de Ingeniería de Electrónica y Redes de Comunicación en la Universidad Técnica del Norte que ayuda furtivamente a obtener resultados mediante encuestas fáciles de contestar, pero con gran validez, a continuación, en la Ecuación 1 se muestra la fórmula del cálculo de muestra que ayudará con el número de encuestas que toca realizar.

$$n = \frac{N * Z^2 * S^2}{(d^2 * (N - 1)) + Z^2 * S^2}$$

Ecuación 1. Fórmula para el cálculo de muestra.

Fuente: (Aguilar, 2005)

Donde:

n = tamaño de la muestra

N = tamaño de la población

Z= valor de Z crítico, calculado en las tablas del área de la curva normal. Llamado también nivel de confianza. Es un valor constante que, si no se tiene su valor, se lo toma en relación al 95% de confianza equivale a 1,96 (como más usual) o en relación al 99% de confianza equivale 2,58, valor que queda a criterio del investigador.

S= Desviación estándar de la población que, generalmente cuando no se tiene su valor, suele utilizarse un valor constante de 0,5.

d = Límite aceptable de error muestra que, generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09), valor que queda a criterio del encuestador. (Aguilar, 2005)

A continuación, en la Tabla 8 se muestra los valores que se utilizarán para generar el cálculo de muestra y obtener el número de personas que se deberá realizar la encuestas con el 95% de confianza equivalente.

Tabla 8. Descripción y valores de parámetros a utilizar en el cálculo de muestra

Parámetros utilizados	Valor
N	158
Z	0,5
S	1,96
d	0,09

Fuente: (Aguilar, 2005)

Dados los parámetros y valores a utilizarse se realiza el cálculo para conocer el total de encuestas a realizar como se expresa a continuación en la Ecuación 2.

$$n = \frac{158 * 0.5^2 * 1.96^2}{(0.09^2 * (158 - 1)) + 0.5^2 * 1.96^2}$$

Ecuación 2. Fórmula del cálculo de muestra.

Fuente: (Aguilar, 2005)

$n = 66$ encuestas

Donde:

n = el número de encuestas que se realizaran.

Las preguntas realizadas en las encuestas con sus respectivas tabulaciones están en la sección de Anexos. A continuación, se indica las conclusiones llegadas después de realizar dicha encuesta:

El promedio de estudiante que han recibido materias relacionadas a tecnologías inalámbricas o han pasado por materias de electrónica es muy alto por lo que indica que realizan proyectos o trabajos de práctica con frecuencia.

La mayoría de los estudiantes tienen conocimiento acerca de sistemas embebidos o han escuchado de proyectos que se realizan con la ayuda de módulos de comunicación inalámbricos por lo que denota gran interés en este tipo de sistemas.

La mayoría de los estudiantes conoce estándares de comunicación inalámbricos, por lo que en su opinión han utilizado por lo menos uno de los estándares establecidos como RF, Bluetooth, Zigbee y WiFi, y aún más en sistemas IOT.

Un porcentaje alto de estudiantes considera o tiene conocimiento que el laboratorio de la carrera posee módulos de comunicación inalámbricos o posee plataformas que ayuden al desenvolvimiento de proyectos IOT.

La mayoría de los estudiantes afirman conocer que es una red de sensores o han desarrollado proyectos referentes con este tipo de tecnologías con el fin de presentar proyectos de materia dando un énfasis muy grande a este tipo de redes.

La mayoría de las personas no conoce un algoritmo o no sigue lineamientos en el desenvolvimiento de proyectos en una red de sensores inalámbricos por lo que lo realizan de una manera no muy práctica con errores en sus sistemas de comunicación además de no conocer subsistemas basados en la implementación de estos.

En gran parte un número de estudiantes desean aumentar el rendimiento y eficiencia en una transmisión de datos basado en una red de sensores por lo que afirma una creación de técnicas que ayuden a la eficiencia de este.

La mayoría de los proyectos realizados por tesis, ingenieros y estudiantes se han tenido problemas en el rendimiento de transmisión de datos en una red de sensores trabajando con grandes cantidades de datos en el sistema.

Todos los encuestados desean que exista un algoritmo que les ayude a optimizar la transmisión de datos en una red de sensores, para aplicarlas en proyectos relacionados a Internet de las Cosas o proyectos de titulación.

Un número muy grande de estudiantes desea que sus sistemas cuenten con ayuda a un control de sus sistemas de red de sensores, además de una visualización de cada uno de los nodos colocados en su red.

3.2.4. Características del Sistema

Las redes de sensores o sistemas electrónicos se basan en estándares de comunicación inalámbricos los cuales se evaluará cómo reaccionan al manejar volúmenes de información grandes, todo esto se lo podrá observar mediante un virtual terminal, dependiendo de cada uno de los protocolos que se van a generar en el desarrollo de pruebas, y se desea implementar un nuevo algoritmo que genere mayor eficiencia en los sistemas.

3.2.4.1. Riesgos

- Que el hardware que se utilice para las pruebas no genere lo deseado al momento de su ejecución.
- Que no haya mucha información acerca de las librerías a utilizar en la mayoría de los estándares de comunicación inalámbricos.
- Que la ejecución del prototipo tome más tiempo de lo planificado.
- La adquisición de los módulos tome mayor tiempo de lo esperado.
- El porcentaje de eficiencia luego de implementar el algoritmo no sea tan elevado.
- Que se necesite de fuentes portables para la alimentación de los sensores inalámbricos al momento de realizar las pruebas pertinentes.
- Que los protocolos de comunicación sean un riesgo para la elaboración de dicho algoritmo.

3.2.4.2. Restricciones

- El sistema deberá ser manipulado únicamente por el administrador o el docente que ayudará a realizar las pruebas en cada uno de los sistemas electrónicos.
- El sistema deberá trabajar en una topología estrella en cada uno de los estándares de comunicación inalámbricos.
- Los sensores seleccionados sean de mayor precisión y estabilidad para las pruebas a elaborar.
- Se utilizará una placa de desarrollo en microcontrolador cuya programación se realizará en lenguaje C.
- El hardware necesario deberá ser compatible para cada uno de los protocolos establecidos.

3.2.5. Definición de Acrónimos y Abreviaturas

En el presente capítulo se utilizará una lista de acrónimos y abreviaturas con frecuencia, para ello se los describirá en la Tabla 9, para un mejor entendimiento y manejo de estos términos.

Tabla 9. Lista de Acrónimos y Abreviaturas

Abreviatura	Descripción
UTN	Universidad Técnica del Norte
CIERCOM	Carrera de Ingeniería en Electrónica y Redes de Comunicación
FICA	Facultad de Ingeniería en Ciencias Aplicadas
CSMA/CA	Acceso múltiple con detección de portadora y prevención de colisiones
RF	Radio Frecuencia
StRS	Documento de especificación de requerimientos del Stakeholder
SyRS	Documento de especificación de requerimientos del sistema
SRSH	Documento de especificación de requerimientos de Software y Hardware

Fuente: Autoría.

3.3. METODOLOGÍA EN V: Fase 2 Requerimientos

En este apartado se especificará los requerimientos que cumplan con las necesidades de análisis, entre estos requerimientos se evaluará tanto, requerimientos operacionales y de usuario, requerimientos de sistemas tanto software como hardware, además de requerimientos de arquitectura entre otros. Para ello se tomará como muestra de interesados o stakeholders según la norma IEEE 830m como se muestra en la Tabla 10, a quien se realiza el estudio según el análisis previamente realizado con el fin de especificar los requerimientos.

Tabla 10. Lista de Stakeholders

Lista de Stakeholders	
Universidad Técnica del Norte (CIERCOM)	Docentes
	Estudiantes
Msc. Michilena	Jaime Director de Tesis
Ing. Carlos Pupiales	Codirector de Tesis
Ing. Paúl Rosero	Docente (CIERCOM)
Sr. Juan Loza	Estudiante

Fuente: Autoría

3.3.1. Requerimientos operacionales y de usuario

Las Tabla 11 y Tabla 12 contienen requerimientos iniciales para el desarrollo de pruebas o implementación de un nuevo algoritmo en capa aplicación tanto al usuario, como requisitos operacionales, estos lineamientos dan una visión del funcionamiento de la red de sensores y brindan un nivel de prioridad.

Tabla 11. Requerimientos operacionales del sistema a generar.

StSR					
REQUERIMIENTOS OPERACIONALES					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
StRS 1	Los estándares de comunicación deben ser inalámbricos.	X			
StRS 2	Alta transmisión y recepción en la tasa de datos.	X			
StRS 3	Bajo consumo de energía en los sistemas electrónicos.		X		
StRS 4	El subsistema de detección y comunicación debe ser de bajo costo.	X			

Fuente: Autoría

Tabla 12. Requerimientos de usuario del sistema a generar.

StSR					
REQUERIMIENTOS DE USUARIO					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
StRS 1	Trabajar en lenguaje C	X			
StRS 2	El diseño electrónico sea simple.	X			
StRS 3	Bajo consumo de batería			X	
StRS 4	Software posea visualización Virtual Terminal			X	
StRS 5	Bajo costo de Hardware		X		
StRS 6	Software libre	X			

Fuente: Autoría

3.3.2. Requerimientos iniciales del sistema

Para la elaboración de los sistemas electrónicos propuestos, se debe tener en cuenta cómo se comportará el sistema junto a sus componentes, además definir cada función que se requiere o se desea en el sistema, para ello en la Tabla 13 se plantea cuáles serán los requerimientos iniciales o de función del sistema en su uso, para obtener una limitación del sistema, además cada uno de los requerimientos que se expresan en siguiente tabla son parte importante para realizar el diseño y manejo del sistema.

Tabla 13. Requerimientos iniciales del sistema electrónico

SySR					
REQUERIMIENTOS DE FUNCIONES					
NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
StRS 1	El manejo en la taza de datos sea elevado tanto en la Tx como Rx de estos.	X			
REQUERIMIENTOS DE USO					
StRS 3	Existan manuales de configuración para la red de sensores.		X		
StRS 4	Datasheet de sensores			X	
StRS 5	Manual para agregar nuevos nodos	X			
REQUERIMIENTOS DE PERFORMANCE					

StRS 6	Mayor precisión de los sensores para la lectura de los datos.	X
StRS 7	Ubicación adecuada de los nodos para mayor eficiencia en la lectura de datos.	X
StRS 8	Tasa de datos elevada	X
StRS 9	No exista interferencias con otros equipos de comunicación	X
StRS 10	Capacidad de conectar inalámbricamente varios nodos sensores a un nodo central	X

REQUERIMIENTOS DE INTERFACES

StRS 11	Comunicación a través de puerto serial al PC	X
StRS 12	Fuente de energía mediante puerto serial.	X

REQUERIMIENTOS DE MODOS/ESTADOS

StRS 13	Modo Maestro	X
StRS 14	Modo Esclavo	X
StRS 15	Modo AP	X
StRS 16	Modo Estación	X
StRS 17	Modo Lectura	X
StRS 18	Modo Escritura	X

REQUERIMIENTOS FÍSICOS

StRS 19	Cobertura adecuada en la interconexión de nodo central con las estaciones.	X
StRS 20	Exista constante alimentación eléctrica	X

Fuente: Autoría.

3.3.3. Requerimientos iniciales de Arquitectura

En el siguiente apartado se denota funcionalidades del sistema electrónico a realizar, para ello se necesita describir puntos o limitaciones importantes tanto en el software como en el hardware que se necesitara en sus componentes básicos, además de requerimientos lógicos y de diseño que son necesarios para generar un sistema adecuado a nuestro tema de investigación haciéndolo más fácil de manipular y operar como se puede apreciar en la Tabla 14.

Tabla 14. *Requerimientos funcionales de Software y Hardware en el desarrollo del sistema electrónico.*

REQUERIMIENTOS DE FUNCIONES

SRSH

REQUERIMIENTOS LÓGICOS

NÚMERO	REQUERIMIENTOS	PRIORIDAD			RELACIÓN
		ALTA	MEDIA	BAJA	
SRSH 1	Hardware debe poseer entradas y salidas análogos y digitales	X			
SRSH 2	El subsistema de procesamiento debe realizar la conversión análogo digital		X		

SRSH 3	Comunicación Inalámbrica debe ser mediante topología estrella	X
REQUERIMIENTOS DE DISEÑO		
SRSH 4	Software libre	X
SRSH 5	Hardware de bajo costo	X
SRSH 6	Programación en lenguaje C	X
SRSH 7	Compatibilidad de Equipos	X
REQUERIMIENTOS DE SOFTWARE		
SRSH 8	Sencillo de manejar	X
SRSH 9	Disponibilidad de librerías para sensores y plataformas	X
SRSH 10	Licencias gratuitas en la utilización de software de programación.	X
SRSH 11	Software de programación debe ser compatible con la placa Arduino	X
SRSH 12	Software compatible con el sistema operativo de la PC	X
REQUERIMIENTOS DE HARDWARE		
SRSH 14	Placa basada en microcontrolador	X
SRSH 15	Debe poseer pines de Tx y Rx	X
SRSH 16	Debe manejar estándares de comunicación inalámbrica.	X
SRSH 17	Debe ser compatible con protocolos de comunicación inalámbricos.	X

SRSH 18	Trabaje con voltajes de 3V y 5V	X
SRSH 19	Capaz de realizar lectura de sensores	X
SRSH 20	Fácil Configuración en la interconexión del subsistema de detección y comunicación.	X
SRSH 21	Debe ser estable al manejar constante taza de datos.	X
REQUERIMIENTOS ELÉCTRICOS		
SRSH 21	Tomas eléctricas o uso de baterías.	X

Fuente: Autoría.

3.3.4. Elección de Software y Hardware para el diseño de los sistemas electrónicos.

En este apartado lo que se desea es realizar la elección tanto de software como de hardware dado los requerimientos especificados anteriormente, además de limitar da prioridad a elementos que cumplan las funciones que se desea desarrollar, también evalúa cada uno de los sistemas o equipos para seleccionar de una forma calificada.

3.3.4.1. Elección de software

Para realizar la correcta elección del software a utilizar se tomará en cuenta los requerimientos especificados anteriormente, entre ellos se observará desde el punto de vista funcional, tecnológico y operativo limitando a un software que nos ayude al desarrollo de los sistemas. Para ello se puede apreciar en la Tabla 15 una valoración a cada software sugerido y una total para la selección de uno de estos.

Tabla 15. Elección en base a los requerimientos de Software

SOFTWARE	REQUERIMIENTOS								VALORACIÓN TOTAL
	SySR 11	SRSH 4	SRSH 6	SRSH 7	SRSH 9	SRSH 10	SRSH 11	SRSH 12	
IDE de Arduino	1	1	1	1	1	1	1	1	8
Fritzing	1	1	1	1	0	1	0	1	6
ESPlorer	1	1	0	0	0	1	0	1	4
Cumple = 1									
No Cumple = 0									
Elección: IDE de Arduino									

Fuente: Autoría.

3.3.4.2. Elección de Hardware

Para la elección de hardware hay que tomar en cuenta que se debe obtener 3 subsistemas que son de procesamiento, comunicación y detección para el desarrollo de las pruebas e implementación del nuevo algoritmo, se realizará la elección mediante los requerimientos anteriormente citados y se elegirá el ideal por medio de una valoración que se puede observar en las tabla mostrando si cumple o no los requerimientos necesarios, el primero en seleccionar es el subsistema de procesamiento como se observa en la Tabla 16.

Tabla 16. Elección de Hardware para el subsistema de procesamiento.

HARDWARE	REQUERIMIENTOS									VALORACIÓN TOTAL
	StR S 6	StR S 10	SRS H 1	SRS H 5	SRS H 14	SRS H 13	SRS H 17	SRS H 18	SRS H 19	
Arduino Leonardo	1	1	1	0	1	0	1	1	1	7
Arduino Uno	1	1	1	1	1	1	1	1	1	9
Intel Galileo	1	1	1	0	1	1	1	1	1	8
Raspberry Pi	1	1	0	0	0	1	1	1	0	5
Cumple = 1										
No Cumple = 0										
Elección: Arduino Uno										

Fuente: Autoría.

A continuación, se realiza la elección del subsistema de comunicación de igual manera tomando como base los requerimientos especificados anteriormente, para este subsistema se realizará la elección de 4 estándares de comunicación inalámbricas.

En la Tabla 17 se muestra la elección del subsistema de comunicación WiFi entre ellos está: ESP, WIFLY y FN-LINK ya que poseen especificaciones ideales para el cumplimiento de la fase de pruebas además que son altamente utilizados para elaboraciones de proyectos electrónico por los estudiantes.

Cumple = 1

No Cumple = 0

Elección: Xbee S2C

Fuente: Autoría.

En la Tabla 19 se puede observar la elección del subsistema de comunicación Bluetooth, para ello se han elegido dos módulos comúnmente utilizados y que cuentan con características idóneas para la fase de pruebas, son estables a diferencia de otros y trabajan con una tasa de información grande, además que posee el laboratorio de electrónica de la Universidad y el estudio que se desea realizar será de mucha ayuda al utilizar estos estándares de comunicación en futuros proyectos.

Tabla 19. Elección de subsistema de comunicación Bluetooth

BLUETOOTH	REQUERIMIENTOS							VALORACIÓN TOTAL
	StRS 6	StRS 7	StRS 19	SRSH 5	SRSH 6	SRSH 7	SRSH 16	
HC-05	1	1	1	1	1	1	1	7
HC-06	1	1	1	0	1	1	1	6

Cumple = 1

No Cumple = 0

Elección: HC-05

Fuente: Autoría.

La Tabla 20 muestra la elección del subsistema de comunicación Radio Frecuencia para ello se han tomado módulos que han evolucionado para la elaboración de sistemas embebidos, comúnmente utilizados y con gran apertura en el mercado, además que son de bajo costo y poseen características muy similares.

Tabla 20. Elección de subsistema de comunicación RF

RF	REQUERIMIENTOS							VALORACION TOTAL
	StRS 6	StRS 7	StRS 19	SRSH 5	SRSH 6	SRSH 7	SRSH 16	
TWS-HSP	0	0	0	1	1	1	1	4
RWS	1	0	0	1	1	1	1	5
NRF24L01	1	1	1	1	1	1	1	7
Cumple = 1								
No Cumple = 0								
Elección: NRF24L01								

Fuente: Autoría.

Por último, se realiza la elección del subsistema de detección que es la unidad de sensor que generara datos al subsistema de comunicación para sus procesamientos, este se realizara la elección según los requerimientos establecidos anteriormente como se muestra en la Tabla 21.

Tabla 21. Elección del subsistema de detección

DETECCIÓN	REQUERIMIENTOS						VALORACIÓN TOTAL
	StRS 3	StRS 6	SRSH 5	SRSH 18	SRSH 19	SRSH 20	
LM35	1	0	0	1	1	1	4
DS18B20	1	1	1	1	1	0	5
DTH11	1	1	0	1	1	0	4
Cumple = 1							
No Cumple = 0							
Elección: DS18B20							

Fuente: Autoría.

3.3.5. Hardware seleccionado para la fase de prueba

Luego de realizar la selección de cada uno de los módulos y plataforma a utilizar se desea realizar una breve descripción de cada uno de ellos para poder apreciar sus especificaciones y el desenvolvimiento que pueden tener al momento de realizar antes de la fase de prueba.

3.3.5.1. Arduino Uno

Arduino Uno formará parte del subsistema de procesamiento, es una placa electrónica basada en ATmega328, posee 14 pines digitales de entrada / salida, 6 entradas analógicas, un resonador cerámico de 16MHz, para tener una mejor visión de sus características se puede apreciar en la Tabla 22 que se muestra a continuación.

Tabla 22. Componentes importantes de un Arduino Uno

ARDUINO UNO	DESCRIPCIÓN
MICROCONTROLADOR	ATmega328
ARQUITECTURA	AVR
TENSIÓN DE FUNCIONAMIENTO	5 V
MEMORIA FLASH	32 KB de los cuales 0,5 KB utilizado por cargador de arranque
SRAM	2 KB
VELOCIDAD DE RELOJ	16 MHz
ANALOG PINES I / O	6
EEPROM	1 KB
CORRIENTE CONTINUA PARA PINES I / O	40 mA en pines I / O; 50 mA en 3,3 V Pin
VOLTAJE DE ENTRADA	7-12 V
DIGITAL PINES I / O	20 (de wich 6 proporcionar una salida PWM)
SALIDA PWM	6
TAMAÑO DE PCB	53,4 x 68,6 mm
PESO	25 g
CÓDIGO DE PRODUCTO	A000066 (TH); A000073 (SMD)

Fuente: (ARDUINO AG, 2017)

Básicamente esta plataforma computacional física está basada en una simple tarjeta de I/O y su entorno de desarrollo que implementa el lenguaje Processing/Wiring, puede ser utilizado para desarrollar objetos interactivos o puede conectarse a un software computacional. El IDE es open-source actualmente para Mac OS X, Windows y Linux (ARDUINO.cl, 2017).

3.3.5.2. Módulo WiFi ESP8266

El Módulo WiFi ESP8266 ofrece soluciones completamente económicas para conexiones de sistemas a redes WiFi esta serie es capaz de funcionar como un adaptador de red en sistemas basados en microcontroladores y se comunica a través de interfaz UART, es económica en el mercado además que concentra características necesarias para poder implementar en proyectos embebidos sin problemas, ya que existe documentación tanto en inglés como español acerca del módulo. Es confiable permitiendo al diseñador delegar todas las funciones relacionadas con WiFi y TCP/IP. A continuaciones en la Tabla 27, se puede apreciar de forma resumida las características que ofrece el módulo WiFi en Serial ESP8266.

Tabla 23. Características importantes en un módulo WiFi ESP8266

CARACTERÍSTICAS	DETALLES
PROTOCOLOS SOPORTADOS	802.11 B/G/N
POTENCIA DE SALIDA	+19.5DBM EN MODO 802.11B
SENSOR TEMPERATURA	DE INTEGRADO
CONSUMO EN MODO DE BAJA ENERGÍA	<10 UA
PROCESADOR INTEGRADO	32 BITS
TAMAÑO REDUCIDO	ULTRA (16MM X 22MM)
ALIMENTACIÓN	3.0~3.6V
COMPATIBILIDAD	WPA / WPA2

Fuente: (GeekFactory, 2017)

Para la conexión de este módulo es necesario de una fuente interna de 3.3v del Arduino de un máximo de 50mA, si sobrepasa este consumo suele existir problemas de arranque o es poco fiable, además hay que conectar como se muestra en la Figura. 18, cada periférico para un funcionamiento de modo autónomo que se podría programar e incluso, dispone de un par de puertos GPIO para su uso como activador.

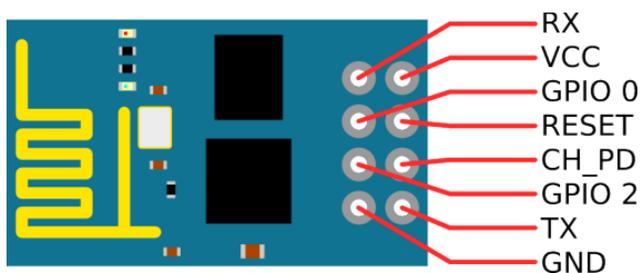


Figura. 18. Puertos de conexión de Modulo ESP8266 visto desde la parte superior.
Fuente: (ARDUINO.cc, s.f.)

3.3.5.3. Módulo Xbee S2C ZigBee

El módulo es un componente electrónico confiable y de conectividad inalámbrica rentable en sistemas embebidos. El S2C Zigbee es adecuado para aplicaciones en las que el consumo de energía es crítico, usa una interfaz SPI de alta velocidad facilitando la integración de microcontroladores y generando ventajas como cortos tiempos de desarrollo de proyectos. La configuración de este módulo es fácil usando software libre a través de comandos AT, entre sus importantes especificaciones se tiene que la velocidad de datos es de 250Kbps y la potencia de transmisión es de 3.1mW. En la Tabla 24 se muestra de una manera resumida cuales son los requerimientos que posee y cuáles son sus características más importantes para trabajar con este módulo.

Tabla 24. Especificaciones del Módulo S2C Zigbee Xbee

ESPECIFICACIONES	DETALLES
FUENTE DE ALIMENTACIÓN	2,1 A 3.6VDC
FRECUENCIA	2.4 GHZ
VELOCIDAD DE DATOS	250 KBPS
CADENA DE TRANSMISIÓN	33 MA
ALCANCE	60 METROS EN INTERIORES Y HASTA 1200 METROS EN EL AIRE ABIERTO
ANTENA	WIRE
PINES	15 PINES DE E / S DIGITALES
DATOS DE INTERFAZ	UART Y SPI
PROTOCOLO	ZIGBEE
DIMENSIONES	27,61 X 24,38 MM

Fuente: (Jose, s.f.)

Los módulos Xbee pueden ser configurados con el número mínimo de conexiones: Power (3.3V), GND y TX/RX de la UAR, y conexiones recomendables como reset y sleep. Gran parte de estos módulos posee otras conexiones como Flow control, input/output, conversor análogo digital e indicador. Este puede funcionar con 3.3V ya que los pines no son tolerantes a 5V. Todos estos pines pueden apreciarse de mejor manera en la siguiente Figura. 19.

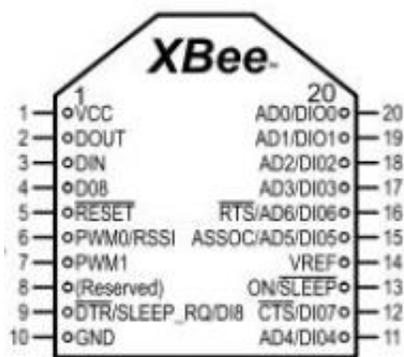


Figura. 19. Pines de configuración que cuenta un módulo Xbee S2C
Fuente: (ARDUINO.ag, 2016)

3.3.5.4. Módulo NRF24L01 RF

El NRF24L01 es un módulo de comunicación inalámbrico compatible con un procesador como Arduino, interfaz un transceptor RF a una frecuencia de 2.4 GHz a 2.5GHz, este tiene una antena integrada con forma de zig-zag con un alcance de 20 a 30 metros, es un módulo actualizado y de gran uso en sistemas electrónicos como medio de comunicación, a continuación, se puede mostrar en la Tabla 25 parámetros importantes que posee este módulo para tener un claro conocimiento de cómo trabajará en la fase de prueba.

Tabla 25. Especificaciones técnicas del módulo NRF24L01

PARAMENTOS	DETALLES
VELOCIDAD TRANSMISIÓN	DE 250 KBPS, 1MBPS, Y 2 MBPS
CONEXIÓN SIMULTÁNEA	6 DISPOSITIVOS.
BANDA DE FRECUENCIA	2400 A 2525 MHZ
CANALES ESPACIADOS	125

TENSIÓN ALIMENTACIÓN	DE 1.9 A 3.6V
ALCANCE MÁXIMO	20-30 METROS
ANTENA	INTEGRADA

Fuente: (Llamas, 2017)

Son empleados ampliamente por su bajo costo y buenas características, un factor de gran impacto en el alcance es la alimentación que se realice al módulo, para conseguir el máximo alcance se conviene realizar la alimentación con una fuente externa de 3.3v, estable y con potencia suficiente.

La conexión del módulo es sencilla, alimentamos el integrado desde Arduino y se conectan los pines de datos como se muestra en la Figura. 20.

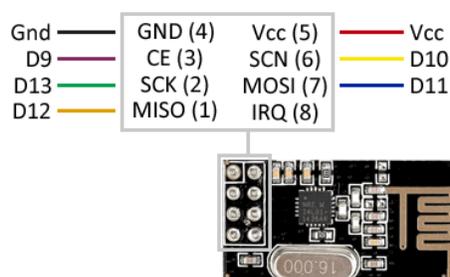


Figura. 20. Esquema de montaje de un módulo RF NRF24L01

Fuente: (Llamas, 2017)

3.4. METODOLOGÍA EN V: Fase 3 Diseño del Sistema Electrónico

Una vez culminado la fase de análisis en donde se pudo determinar la situación actual y adquirir información útil observando y describiendo factores importantes que se deben tratar en la investigación, además de la fase de requerimientos que toma relevancia en la selección de software como hardware para el desarrollo del proyecto basado en lineamientos técnicos y operacionales.

Se desea dar a conocer como estas dos partes se van a interconectar y brindar un funcionamiento tanto con diagrama de bloques como esquemas de interconexión para un funcionamiento adecuado.

3.4.1. Diagrama de bloques general

Se desea brindar una visión general de cómo funciona el sistema electrónico y que partes conllevará cada bloque, además de cómo van a estar interconectados cada uno de los subsistemas, brindando una clara percepción y entendimiento del presente proyecto como se muestra en la Figura. 21.

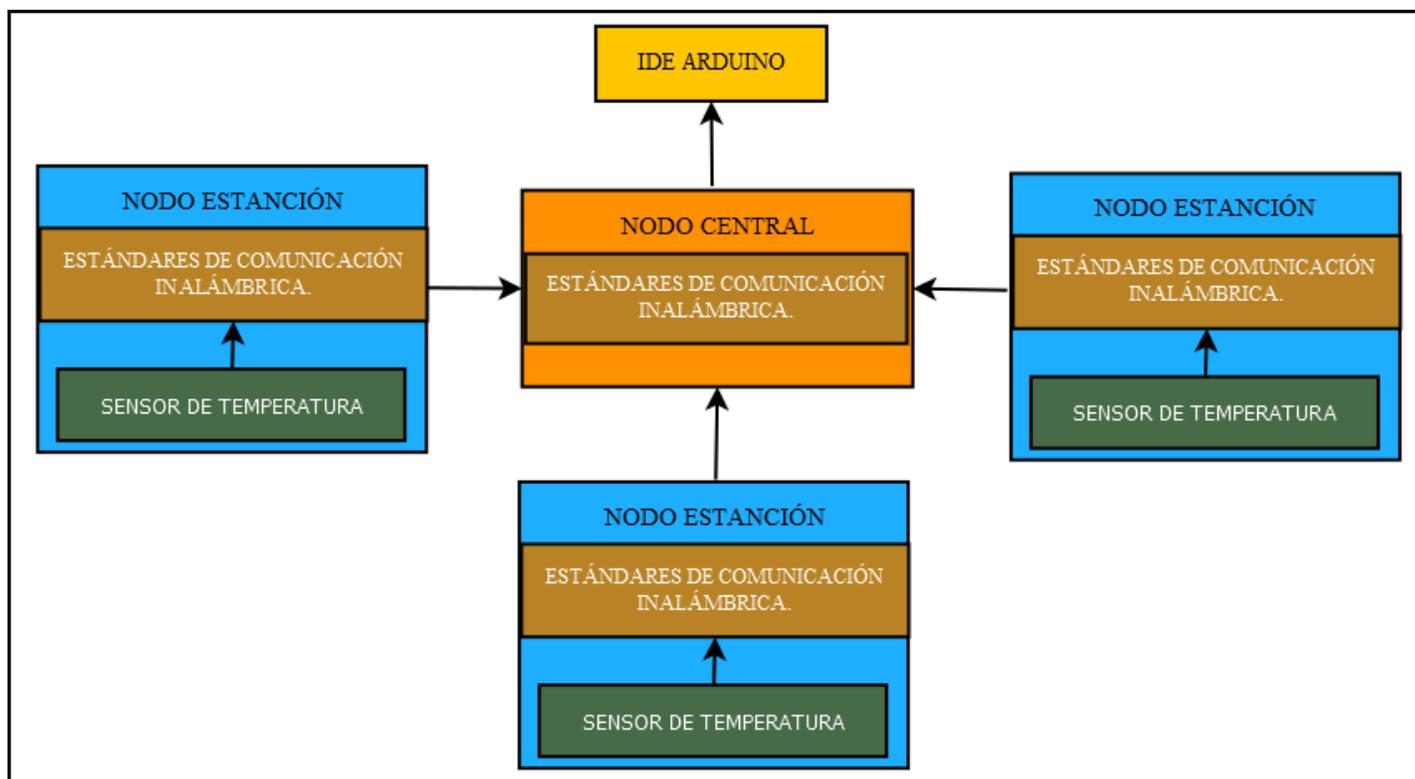


Figura. 21. Diagrama de bloques general del sistema electrónico.

Fuente: Autoría.

Dado que se van a utilizar cuatro estándares de comunicación se lo ha descrito de forma global como muestra la Figura. 21, ya que los estándares tanto Bluetooth, Zigbee, WiFi y RF van a trabajar de la misma manera que se muestra en el diagrama de bloques general y se darán uso en el presente proyecto.

3.4.2. Nodo Central

Dado que se va a utilizar una topología estrella, hay que tener claro de que va a estar conformado el nodo central, ya que este va a ser el encargado de comunicarse con cada uno de los nodos además de ser el responsable de darnos una percepción del procesamiento de datos y envío de éstos a un virtual terminal para conocer cómo está trabajando dicho nodo.

3.4.2.1. Diagrama de bloques Nodo Central

El nodo central recibe, procesa y envía los datos que se van a manejar en toda la red de sensores, va a contener un subsistema de comunicación y un subsistema de procesamiento y conectados a él los distintos nodos sensor, en este caso va a recibir datos de temperatura ya antes establecido, todo esto va a estar conectado como muestra la Figura. 22.



Figura. 22. Diagrama de bloques del Nodo Central con sus respectivos subsistemas.

Fuente: Autoría

3.4.2.2. Diagrama de flujo Nodo central

A continuación, se describirá cada uno de los pasos de cómo trabaja el nodo central para tener una clara percepción y un mejor entendimiento de la red de sensores a implementar para realizar las pruebas posteriormente de funcionamiento.

Lo primero que se realizará es determinar el protocolo o algoritmo que se va a utilizar, ya que dependiendo de ello se va a tener una distinta configuración o análisis de la tasa de datos, como el objetivo de este proyecto es determinar problemas y dar una solución con un nuevo algoritmo de comunicación, se lo realizará tanto con CSMA/CA y como ALOHA protocolos de comunicación inalámbricos.

Para que el nodo central pueda establecer una conexión con los distintos estándares de comunicación primero hay que declarar cada uno de estos a utilizar en el nodo central, como en el caso de los módulos WiFi se declara el nombre de los SSID a constituir el nodo estación o el módulo RF se declara igualmente un nombre hexadecimal para su comunicación de las estaciones con el nodo central, de la misma manera para los estándares Bluetooth y Xbee, luego de realizar las declaraciones de éstos, se procede a configurar los tiempos de envío o recepción de datos en este caso temperatura los cuales serán procesados.

Para este análisis se utilizarán cuatro módulos por cada estándar de comunicación obteniendo tres nodo estaciones y una central, los cuales establecerán una conexión dependiendo del protocolo que se analice, el nodo central será el encargado de realizar el procesamiento y posteriormente la visualización de esta información, así como los nodo estación realizarán el

subsistema de comunicación de la mano al subsistema de detección para solicitar y enviar los datos adquiridos en estos subsistemas.

El nodo central no solo procesará la información, sino que nos dará una percepción de cómo está funcionando la red y que problemas existen en ella como colisiones de información o tiempos de envío, todo esto el IDE de Arduino gracias a su virtual terminal no los mostrará de una forma visual y ayudará a conocer el estado de la red de sensores.

Este proceso se muestra en el diagrama de flujo de la Figura. 23, indicando sus configuraciones, operaciones y análisis de cómo se manejará dicha red en el nodo central con los cuatro estándares de comunicación ya descritos.

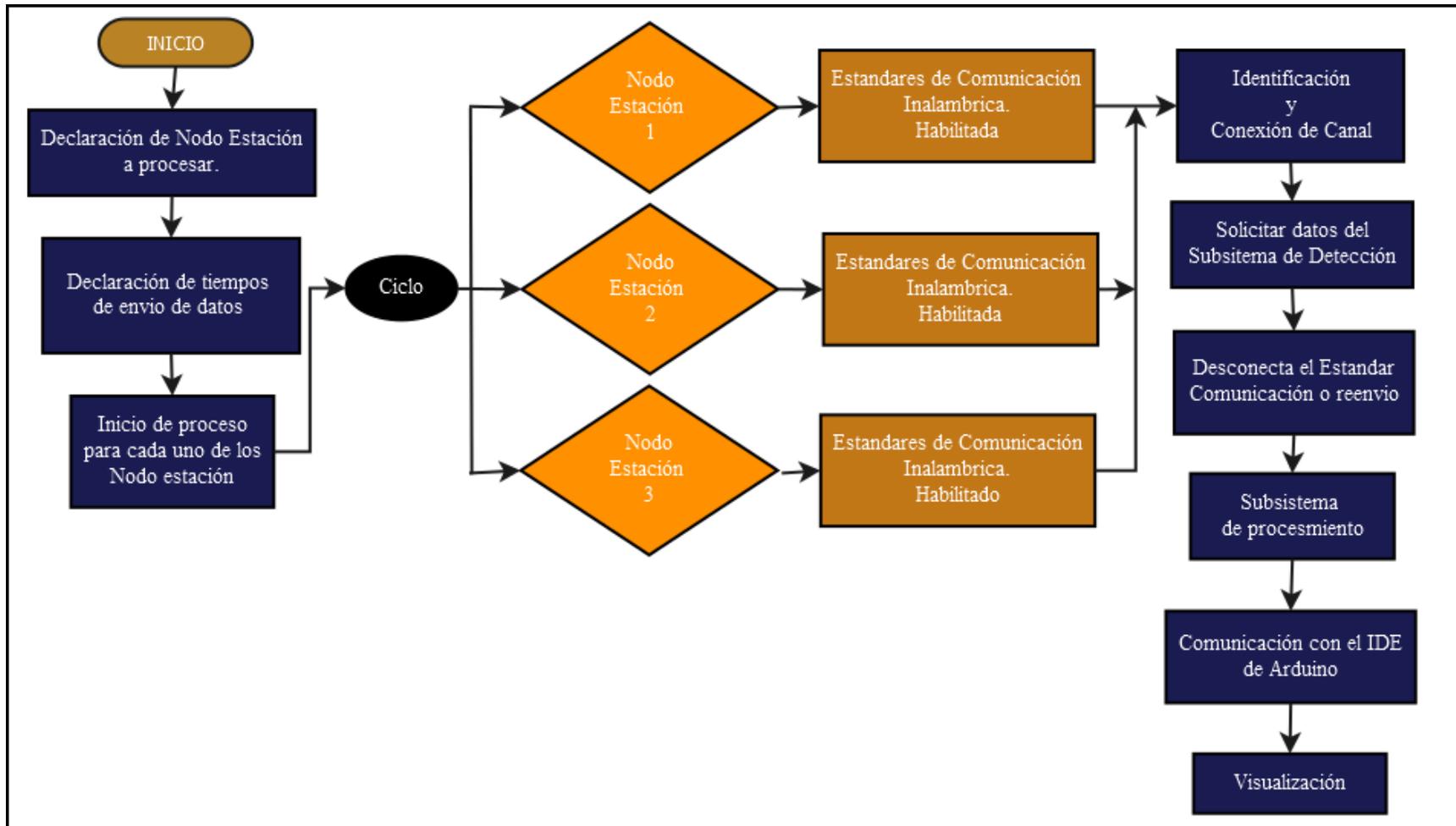


Figura. 23. Diagrama de flujo del Nodo Central con sus respectivos procesos.

Fuente: Autoría

3.4.3. Nodo Estación

Este nodo va a formar parte del sistema electrónico ya que será el encargado de adquirir datos, en este caso de temperatura y estará formado por dos subsistemas de la red de sensores, el de detección y el de comunicación que estará conformado por los estándares de comunicación inalámbricas.

3.4.3.1. Diagrama de bloques Nodo Estación

En el diagrama de bloques que muestra la Figura. 24, se puede observar en separado dichos sistemas que estarán conformados por el hardware ya seleccionado anteriormente, aquí se puede apreciar una perspectiva de cómo estará interconectados cada uno en el nodo estación.

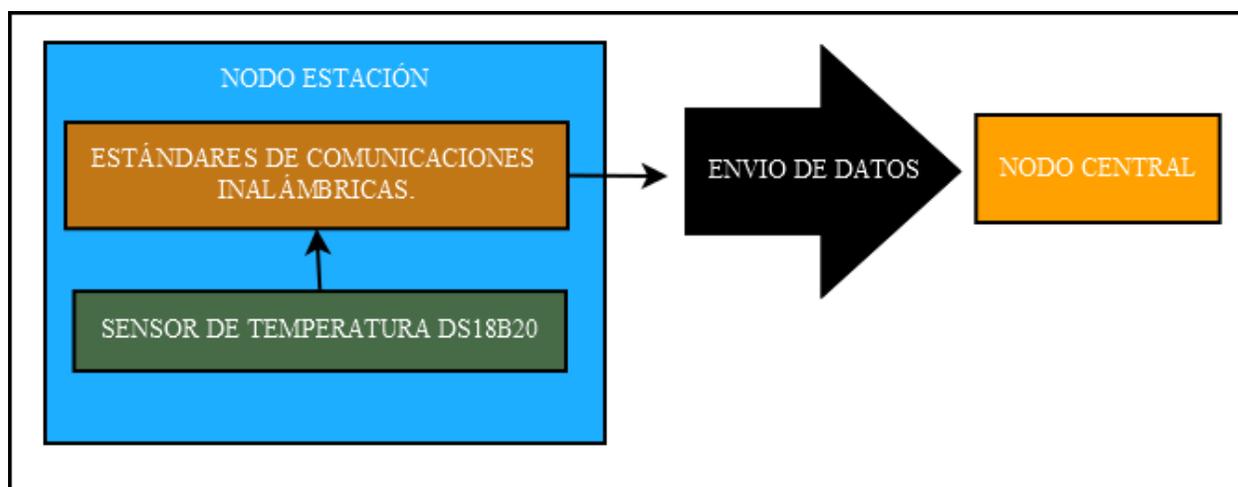


Figura. 24. Diagrama de bloques de un Nodo Estación.

Fuente: Autoría

3.4.3.2. Diagrama de flujo Nodo Estación

Para este diagrama hay que tener en cuenta que lo que se desea es saturar el canal o establecer el envío de grandes volúmenes de información, para ello se utilizan módulos de comunicación y un detector de temperatura para el envío de datos por medio de los módulos a un nodo central donde será procesado y analizado.

Una vez establecida o identificado cada uno de estos nodos en el nodo central es necesario recolectar, y enviar la mayor cantidad de información, para ello se establecerá condiciones o se manejará por medio de tiempos de envío de información para cada uno de estos ciclos en sus distintos estándares de comunicación como se muestra en la siguiente Figura. 25, indicando cada paso a ejecutar para la adquisición de información útil en la fase de prueba.

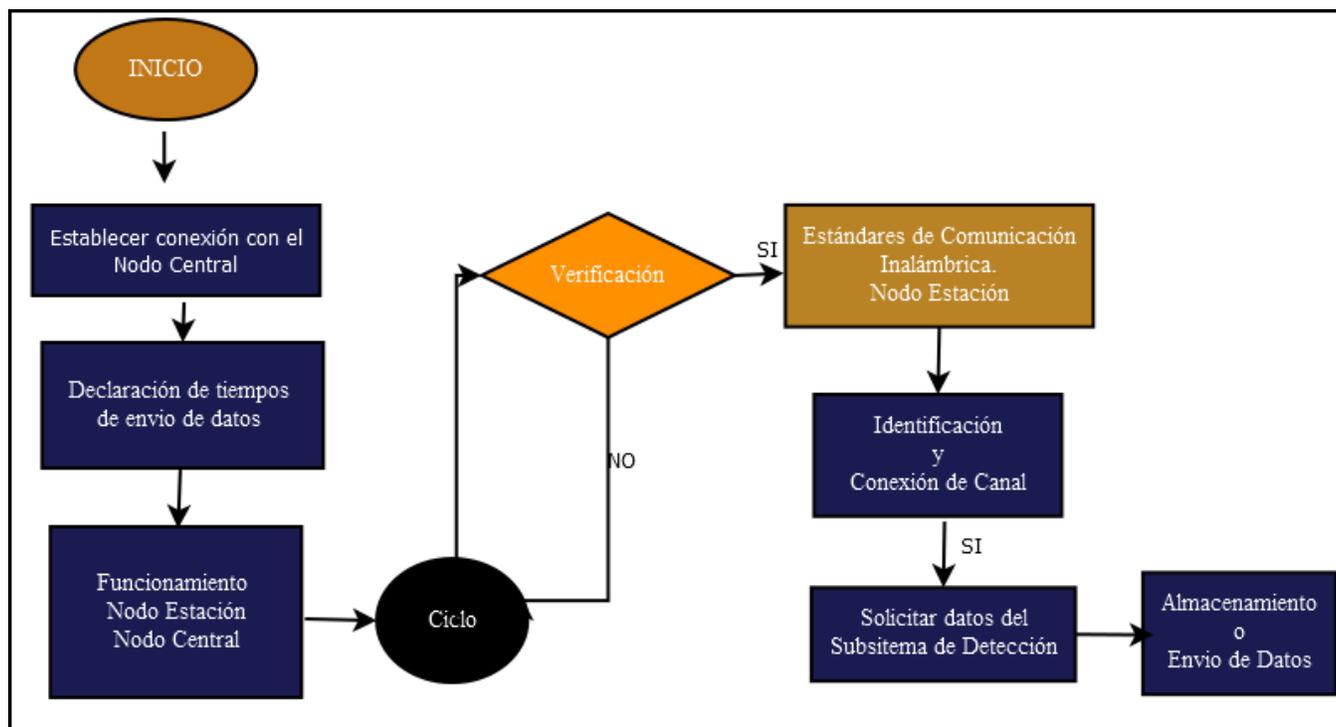


Figura. 25. Diagrama de flujo del nodo estación con sus procesos.

Fuente: Autoría.

3.5. METODOLOGÍA EN V: Fase de prueba

En la presente fase de la metodología en V se realizará las pruebas individuales de cada uno de los sistemas con sus estándares de comunicación inalámbricos para obtener resultados que ayuden al desarrollo del nuevo algoritmo de acceso al medio, el sistema se llevará a cabo en laboratorio de pruebas mediante un escenario controlado independientemente de los datos que se desea adquirir.

3.5.1. Escenario de pruebas para los sistemas de comunicación inalámbricas

El entorno de análisis de cada una de las tecnologías será un laboratorio de prueba controlado, ya que los datos de temperatura que se van adquirir son independientes de los resultados que se desean obtener, para ello parte importante del entorno de análisis es el correcto uso de los elementos de comunicación y la visualización de estos datos en una PC para monitorear el procesamiento como envío y recepción de estos datos en una red de sensores como se puede apreciar en la Figura. 26, que se muestra a continuación.

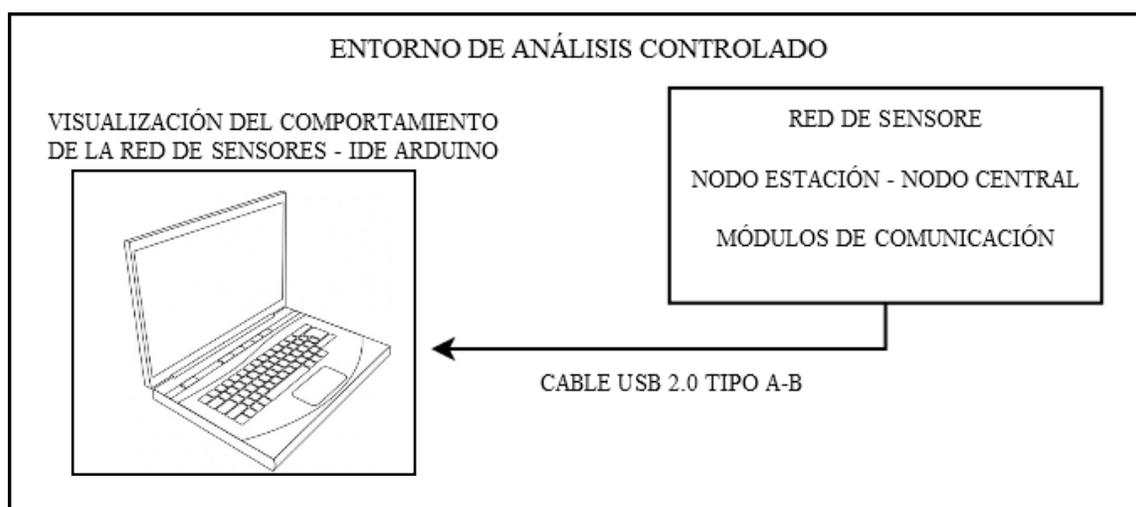


Figura. 26. Diagrama del entorno de análisis controlado

Fuente: Autoría

Para todos los estándares de comunicación inalámbricos tanto RF, WiFi, Bluetooth y Zigbee se va a llevar a cabo el mismo entorno de análisis, lo que van a cambiar es su configuración en software y el módulo de trabajo en hardware.

3.5.1.1. Módulos de alimentación en escenario de pruebas.

El módulo de energía se encarga de generar las diferentes tensiones y corrientes necesarias para alimentar los componentes activos que forman el SE. Lo normal es el empleo de baterías para los dispositivos portátiles y fuentes de alimentación (convertor AC/DC) para los sistemas que disponen de acceso a la red de energía eléctrica.

Dado que la conexión de la energía es clave en muchas aplicaciones y ésta a su vez es fuente de ruido e interferencias, a veces es necesario dotarla de filtros, circuitos integrados supervisores de alimentación, protectores de sobretensión, entre otros.

El consumo de energía puede ser determinante en el desarrollo de algunos sistemas embebidos, que necesariamente se alimentan con baterías y a las que no es posible recargar de forma continua.

Para la fase de pruebas censo sistemas e alimentación para el Arduino uno y para la parte de subsistema de detección, la primera prueba lo realizo con una pila y adaptador de 9V dando una ventaja de ser de ser fáciles de encontrar y usar. Además, hay disponibles cables y porta pilas, que incluso incorporan un conector Jack tipo Arduino, lo que hace que sean fáciles de usar. Y como desventaja las pilas de 9V disponen de baja densidad energética. Una pila tiene una capacidad típica de 500-600mAh. Además, son proporcionan una intensidad de corriente máxima muy baja,

en torno a 300mA, útil solo para proyectos pequeños. Este sistema es muy opcional para pequeños sistemas o en este caso pruebas sencillas. Esta opción se puede apreciar como esta en Figura. 27 que se muestra a continuación.



Figura. 27. Batería de 9V para sistemas embebidos.
Fuente: (Llamas , Ingeniería, informática y diseño, 2016)

Como segunda prueba realizo el censo de baterías con 4 pilas de AA de 1.5 Voltios proporcionando un total de 6V, es otra opción sencilla y ampliamente usado en pequeños proyectos y proyectos de iniciación. Las pilas AA tienen la ventaja de ser fáciles de encontrar. Además, el voltaje de 6V es perfecto para alimentar motores de DC y servos. La carga es superior a las pilas de 9V. Cuatro pilas AA convencionales proporcionan 800-1500 mAh mientras que, en el caso de usar pilas AA alcalinas la capacidad es de 1700-2800mAh. La intensidad máxima que podemos obtener supera 1A, pudiendo llegar a extraer hasta 2 Amperios.

Es valida en proyectos pequeños y a largo plazo resulta no económica. A continuación, en la Figura. 28, se muestra como sería su alimentación.



Figura. 28. 4 pilas AA para sistemas embebidos.
 Fuente: (Llamas , Ingeniería, informática y diseño, 2016)

Como tercera prueba se realizó el análisis con baterías de polímero litio, es la opción mas avanzada para alimentar proyectos se presentan en varios voltajes, en función del número de celdas. Así las baterías de polímetro litio de 2 celdas (2S) proporcionan 7.4V-8.4, y las de 3 celdas (3S) proporcionan 11.1-12.6V. Ambos voltajes son adecuados para alimentar un Arduino o subsistemas de detección.

Las baterías LiPo⁵ presentan la densidad de energía más alta de las opciones planteadas. Podemos encontrar baterías con capacidades desde 500 a 5000mAh. Además, las baterías LiPo tienen la ventaja de proporcionar enormes intensidades. Es posible encontrar baterías 20-25C, lo que se traduce en intensidades de descarga de 50 a 100A, requerida por los motores más grandes. Las baterías LiPo también son la opción más cara, aunque al ser recargables resultan económicas a largo plazo, más aún si tenemos en cuenta sus características eléctricas.

La mayor desventaja de estas baterías es la dificultad y cuidado que hay que tener en su uso. Manipular de forma indebida una batería LiPo puede ser extremadamente peligroso, por la gran cantidad de energía que almacenan. las baterías LiPo son la solución más potente para

⁵ Baterías de polímetro litio

alimentar nuestros proyectos, pero los requisitos de manipulación y carga la hacen adecuada sólo para usuarios avanzados y los proyectos complejos. Este subsistema se muestra en la Figura. 29.

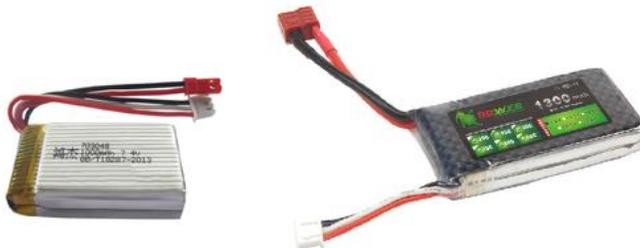


Figura. 29. Baterías de polímetro litio para subsistemas.
Fuente: (Llamas, Ingeniería, informática y diseño, 2016)

Como cuarta opción de análisis esta la batería USB de 5 voltios. El voltaje de 5V es adecuado para alimentar una gran variedad de componentes, como motores DC, servos, así como una gran cantidad de dispositivos (sensores, tiras LED, displays). Un banco de baterías USB es una buena solución de movilidad, como batería para llevar encima que nos permita hacer pruebas y montajes de forma sencilla (y mucho mejor que una batería de 9V). Estos bancos son, por supuesto, recargables. La capacidad de energía es alta, pudiendo encontrar bancos de hasta 17.000 mAh. Este subsistema se puede apreciar en la Figura. 30.



Figura. 30. Batería USB 5 Voltios
Fuente: (Llamas, Ingeniería, informática y diseño, 2016)

Todos estos subsistemas de alimentación en el proyecto mencionado duran entre 5 a 7 días, pero cabe recalcar que dependerá de la corriente que se use en el sistema por ello dejo mencionado

cuanta corriente da cada una de estas alternativas de alimentación y cuanto más o menos durará el sistema que se quiera aplicar, en mi caso el mejor sistema fue la opción de baterías de polímetro litio, pero puede utilizarse cualquiera todo dependerá de que sistema embebido se quiere realizar.

3.5.2. Explicación del funcionamiento de los sistemas.

En todos los sistemas de comunicación inalámbrico a desarrollar lo que se desea es observar el comportamiento de la tasa de información que se va a manejar en toda la red de sensores, para ello todos los subsistemas forman parte importante para el funcionamiento de éste, todo esto se lo expresará mediante la Tabla 26 que se muestra a continuación dando una explicación de cada uno de los subsistemas y que se desea de cada uno de ellos.

Tabla 26. Explicación del funcionamiento general de los sistemas de prueba.

Subsistemas	Funcionamiento
Subsistema de procesamiento	Es la CPU del nodo sensor inalámbrico en este caso la plataforma Arduino UNO que se encarga de procesar enviar y recibir los datos, además de la visualización de estos. Cada nodo sensor tiene su subsistema de procesamiento.
Subsistema de comunicación	Los módulos de comunicación ya seleccionados forman parte del subsistema de comunicación, estos tienen el rol de transmitir los datos de temperatura que se van adquirir manejando la comunicación entre nodo estación y nodo central.
Subsistema de energía	El cable USB 2,0 tipo A-B que está conectado al Arduino uno es el encargado de generar la alimentación a los módulos, sensores y la

plataforma que formara parte de cada uno de los nodos sensores.

Subsistema de detección

El sensor DS18B20 será el encargado de adquirir datos de temperatura monitoreando el entorno de análisis controlado para procesarlo y enviar mediante los módulos de comunicación al nodo central.

Fuente: Autoría.

3.5.3. Fase de prueba uno “Estándar De Comunicación Inalámbrica RF”

La primera tecnología para desarrollar y realizar las pruebas pertinentes son los estándares de comunicación de radio frecuencia, el principal objetivo es observar como interactúa la red sensores con el manejo de datos, y apreciar su comportamiento.

3.6.3.1. Equipamiento del sistema de radio frecuencia

A continuación, en la Tabla 27, se describe cada uno de los componentes que formarán parte de cada uno de los subsistemas para el desarrollo de la red de sensores.

Tabla 27. Componentes que forman parte para el desarrollo del sistema RF.

HARDWARE		
CANTIDAD	COMPONENTES	DESCRIPCIÓN
4	Arduino UNO	Plataforma embebida encargada del control e componentes y parte del subsistema de control
3	DS08L01	Sensor de temperatura
4	NRF01L24	Módulo de Radio Frecuencia

4	CABLE USB Tipo A-B	Conexión y fuente de alimentación de la plataforma Arduino con la PC.
3	Resistores	Protección para los sensores
4	Protoboard	Montaje de los componentes
-	Cable para Protoboard tipo Macho - Hembra y Macho - Macho	Conexión entre componentes.
1	Computador	Monitoreo, fuente y control de datos

SOFTWARE

CANTIDAD	COMPONENTES	DESCRIPCIÓN
-	IDE de Arduino	Software de programación de la plataforma Arduino
-	Fritzing	Diseño de circuitos y esquemas de conexión
-	DIA	Diseño de diagramas de flujo y de esquemas.

Fuente: Autoría.

3.6.3.2. Esquema de conexión en el estándar de comunicación inalámbrica RF

A continuación, se apreciar mediante un esquema de conexiones como va a estar conformado el sistema electrónico, en este caso se van a dar a conocer por cada una de las tecnologías que se desea realizar la fase de prueba las que son WiFi, Bluetooth, Zigbee y RF, esto nos dará una visión de que conformará el sistema electrónico en la red y cómo estará conectado cada uno de éstos.

Para este estándar de comunicación los elementos que conformará el sistema electrónico son: Módulos NRF24I01 como parte del subsistema de comunicación, un Arduino UNO como parte del sistema de procesamiento y un sensor de temperatura DS18B20 como parte del subsistema de detección, los componentes a utilizar fueron seleccionados anteriormente por las distintas características que brindan cada uno de ellos.

3.6.3.3. *Nodo Central*

El nodo central solo conformara la parte de procesamiento y comunicación ya que esta es la encargada de realizar el análisis y la comunicación entre los nodos como se muestra en las Figura. 31 y Figura. 32, que da una visualización de cómo está conformado su parte electrónica.

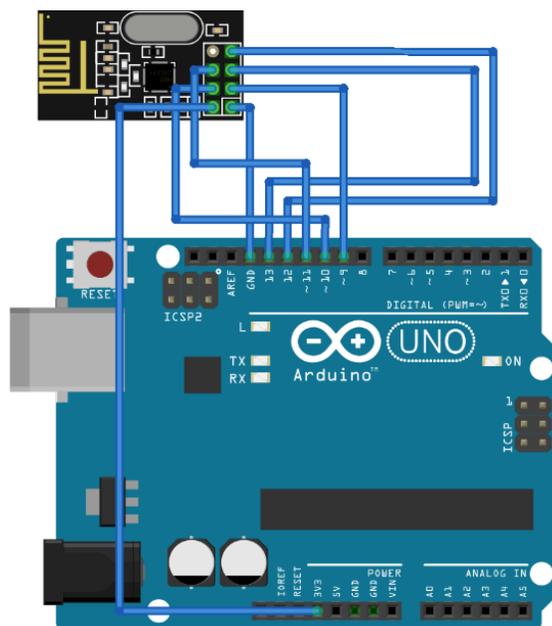


Figura. 31. Esquema electrónico del Nodo Central en fritzing
Fuente: Autoría

A continuación, se muestra una visualización del esquemático de conexión del nodo central y que partes la conforman.

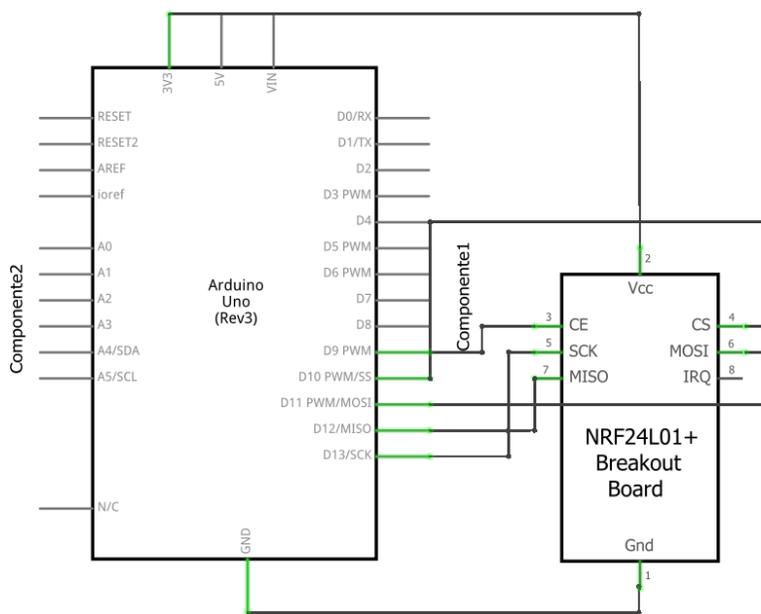


Figura. 32. Esquema de conexión del Nodo Central y sus componentes en fritzing.

Fuente: Autoría

3.6.3.4. *Nodo Estación*

Ya que este nodo es el encargado de enviar datos al nodo central además de los anteriores subsistemas, éste tiene un subsistema de detección y está conformado por un sensor de temperatura DS18B20 como se muestra en la Figura. 33, y brinda una visión de cómo estará construido dicho sistema electrónico en su nodo estación.

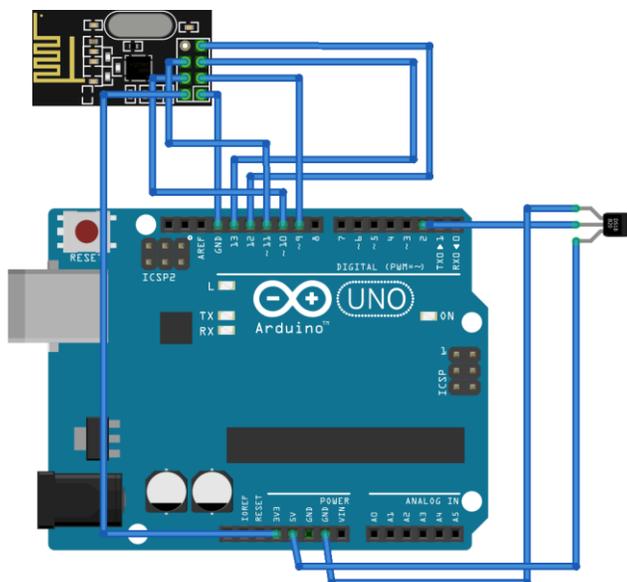


Figura. 33. Esquema eléctrico del nodo estación en fritzing.
Fuente: Autoría

A continuación, en la Figura. 34, se muestra el esquema de conexiones y que elementos van a conformar cada uno de los nodos estación.

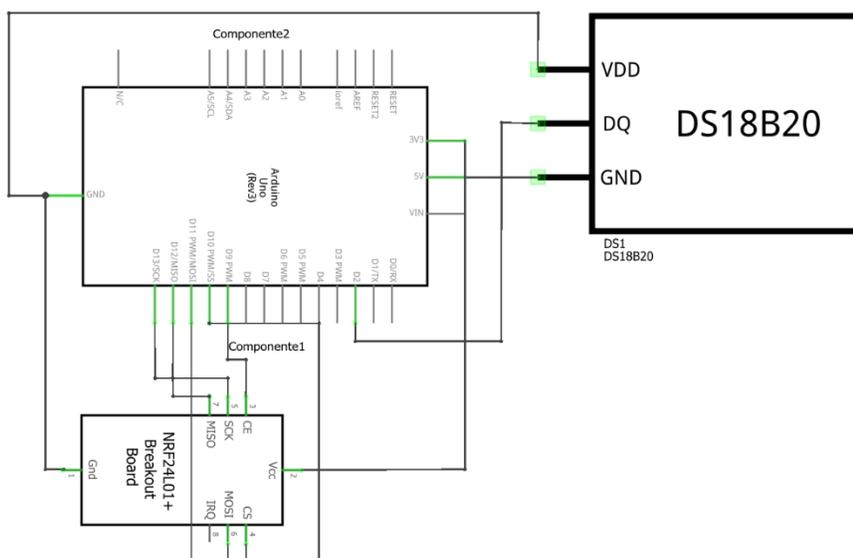
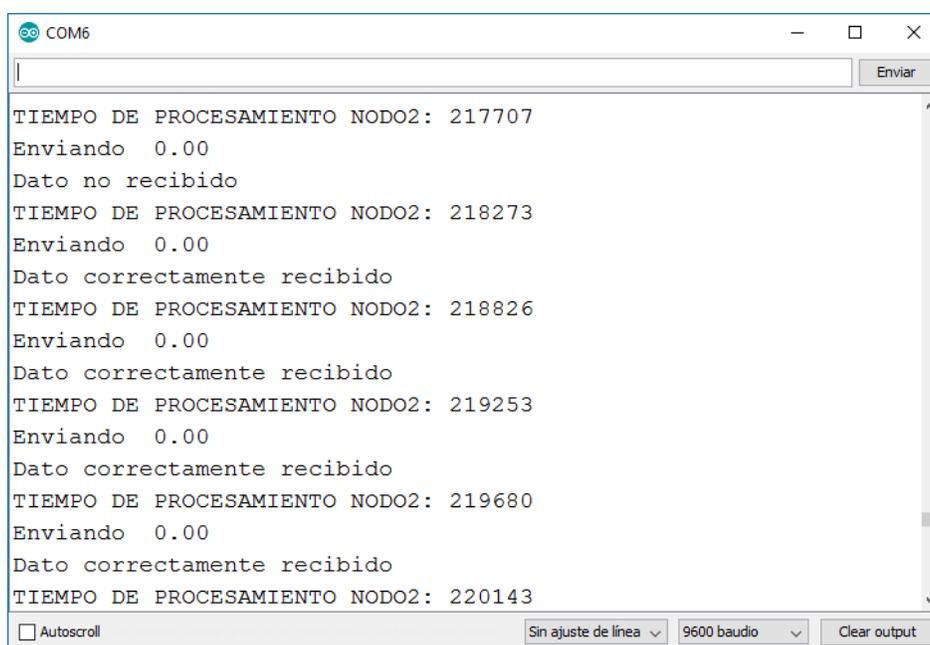


Figura. 34. Esquema de conexiones del nodo estación en fritzing.
Fuente: Autoría

3.6.3.5. Resultados obtenidos

Para conocer el estado de la red de sensores y observar si existen pérdida de datos o el procesamiento que se desea estudiar, se muestra cómo realiza la transmisión los nodo estaciones y al mismo tiempo recibe una respuesta de si el dato llegó o se perdió en el tiempo esperado, para ello se utiliza una función “micro” que devuelve el número en microsegundos desde la placa Arduino comenzó a correr el programa, como se puede apreciar en la Figura. 35, podemos visualizar la monitorización, el dato que ha enviado, un mensaje de si llegó o no el dato y la respuesta en milis de programa, esto se lleva a cabo en el nodo estación.



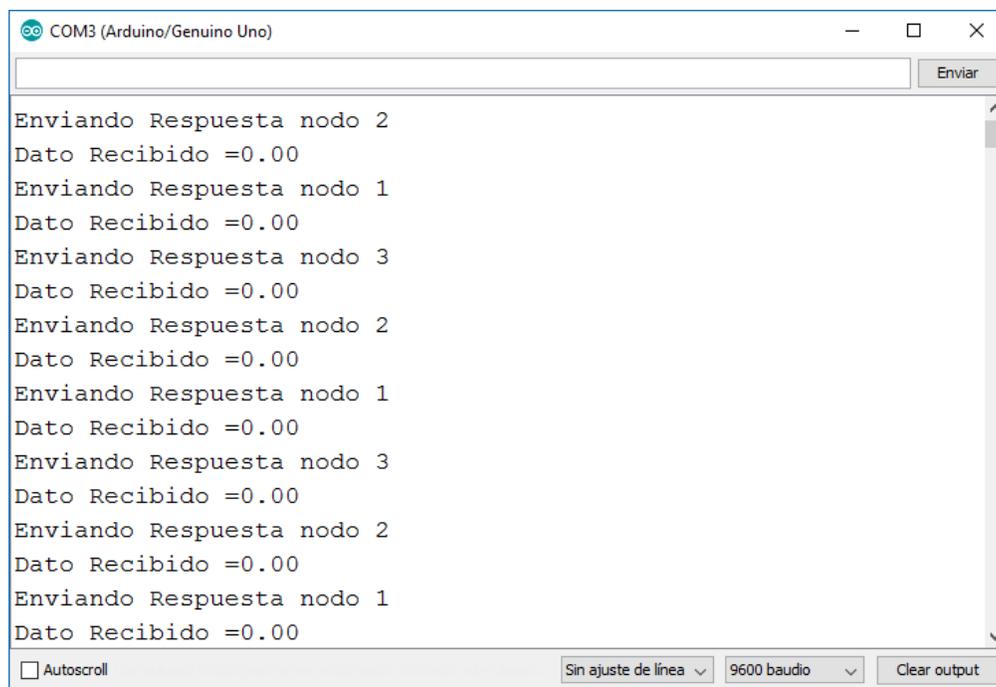
```
COM6
TIEMPO DE PROCESAMIENTO NODO2: 217707
Enviando 0.00
Dato no recibido
TIEMPO DE PROCESAMIENTO NODO2: 218273
Enviando 0.00
Dato correctamente recibido
TIEMPO DE PROCESAMIENTO NODO2: 218826
Enviando 0.00
Dato correctamente recibido
TIEMPO DE PROCESAMIENTO NODO2: 219253
Enviando 0.00
Dato correctamente recibido
TIEMPO DE PROCESAMIENTO NODO2: 219680
Enviando 0.00
Dato correctamente recibido
TIEMPO DE PROCESAMIENTO NODO2: 220143
```

Figura. 35. Visualización de datos en los nodos estación.

Fuente: Autoría.

Además de la visualización que se lleva a cabo en el nodo estación, podemos observar los datos adquiridos y de igual manera visualizados en el nodo central, para ello se puede observar el dato recibido y el envío en micro de la respuesta al momento de adquirir el dato para poder tener

un concepto claro de cómo se está manejando el procesamiento en dicho nodo como se muestra en la Figura. 36, que se encuentra a continuación.



```
COM3 (Arduino/Genuino Uno)
Enviando Respuesta nodo 2
Dato Recibido =0.00
Enviando Respuesta nodo 1
Dato Recibido =0.00
Enviando Respuesta nodo 3
Dato Recibido =0.00
Enviando Respuesta nodo 2
Dato Recibido =0.00
Enviando Respuesta nodo 1
Dato Recibido =0.00
Enviando Respuesta nodo 3
Dato Recibido =0.00
Enviando Respuesta nodo 2
Dato Recibido =0.00
Enviando Respuesta nodo 1
Dato Recibido =0.00
```

Figura. 36. Visualización de datos en el nodo central.

Fuente: Autoria.

Al mismo tiempo que se realiza las pruebas se puede apreciar los problemas o inconvenientes que tiene una red de sensores, para ello se realiza el algoritmo de comunicación entre nodo estación – central, haciendo más flexible la comunicación de datos.



Figura. 37. Red de sensores utilizando módulos RF.
Fuente: Autoría.

3.5.4. Fase de prueba dos “Estándar de comunicación inalámbrica Zigbee”

A continuación, se realiza la fase de prueba para el estándar de comunicación inalámbrica “Zigbee”, para ello se presentará la configuración de los módulos Xbee, además del equipamiento del sistema, los esquemas de conexión y sus resultados.

3.6.4.1. Equipamiento del sistema Zigbee

A continuación, en la Tabla 28 se especifican los componentes tanto de hardware como de software que se utilizará para el desarrollo del sistema embebido con Zigbee.

Tabla 28. Equipamiento del sistema embebido Zigbee.

HARDWARE		
CANTIDAD	COMPONENTES	DESCRIPCIÓN
4	Arduino UNO	Plataforma embebida como parte del subsistema de procesamiento.
3	DS08L01	Sensor de temperatura.
4	Xbee S2 Serie C	Módulo Zigbee.
4	CABLE USB Tipo A-B	Conexión y fuente de alimentación de la plataforma Arduino con la PC.
4	Resistores	Protección para los sensores.
4	Protoboard	Montaje de los componentes.
-	Cable para Protoboard tipo Macho - Hembra y Macho - Macho	Conexión entre componentes.
		Monitoreo, fuente y control de datos.
1	Computador	
SOFTWARE		
CANTIDAD	COMPONENTES	DESCRIPCIÓN
-	IDE de Arduino	software de programación de la plataforma Arduino.
-	Fritzing	Diseño de circuitos y esquemas de conexión.
-	Proteus	Simulador de sistemas electrónicos.
-	DIA0	Diseño de esquemas electrónicos y flujo de datos.
-	XCTU	Software de configuración de módulos.

Fuente: Autoría.

3.6.4.2. Configuración de los módulos Xbee

Para la configuración de los módulos Xbee es necesario poseer tanto en hardware Xbee USB Explorer, como el software XCTU, estos nos permiten realizar las configuraciones de cada uno de los módulos en este caso queremos que trabaje un módulo como nodo central y los demás como nodo estaciones tanto para la transmisión como recepción de datos.

Al ejecutar el programa hay que detectar los dispositivos para el módulo de transmisión las configuraciones establecidas son:

- CH: C
- ID: 1001
- CE: Coordinador
- Velocidad de transmisión: 9600 bps

De igual manera al detectar el módulo que va a interactuar con éstos la configuración debe ser:

- CH: C
- ID: 1001
- CE: Punto final
- Tasa Baund: 9600 bps

Para una mejor apreciación del entorno en el que se va a configurar podemos observar la Figura. 38, que denota los cambios efectuados y el modo de operación de un sensor Xbee

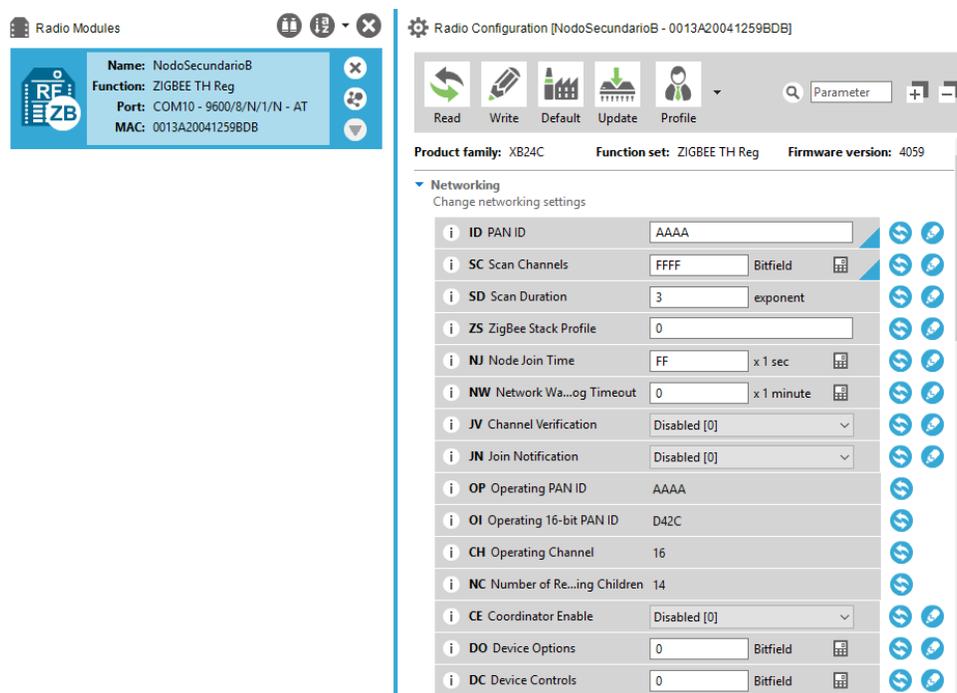


Figura. 38. Configuración de Módulos Xbee en XCTU

Fuente: Autoría.

3.6.4.3. Esquema de conexión en el estándar de comunicación Zigbee

En el esquema de conexión para el estándar de comunicación inalámbrica Zigbee se utilizará un Módulo Xbee S2 tipo C, además de un Arduino UNO y un sensor de temperatura DS18B20, estos componentes en si formarán parte tanto del nodo central como el nodo estación del sistema embebido a realizar las pruebas.

3.6.4.4. Nodo central

Ya que el nodo central será el encargado de procesar los datos adquiridos para su visualización del monitoreo de los demás nodos estaciones cuenta con el módulo Xbee y Arduino uno que estará conectado a la PC, en la Figura. 39 y Figura. 40, muestra como estarán conectados dichos elementos en el nodo central.

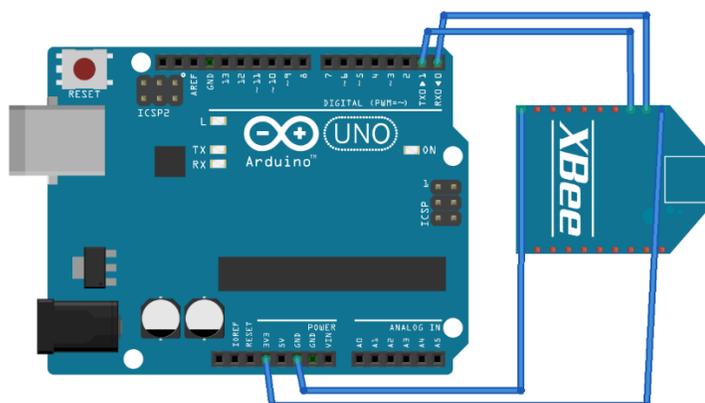


Figura. 39. Esquema eléctrico de un nodo central con Xbee en fritzing.
Fuente: Autoría.

En la que se muestra a continuación se puede apreciar el esquema de conexiones del nodo central y sus respectivos componentes.

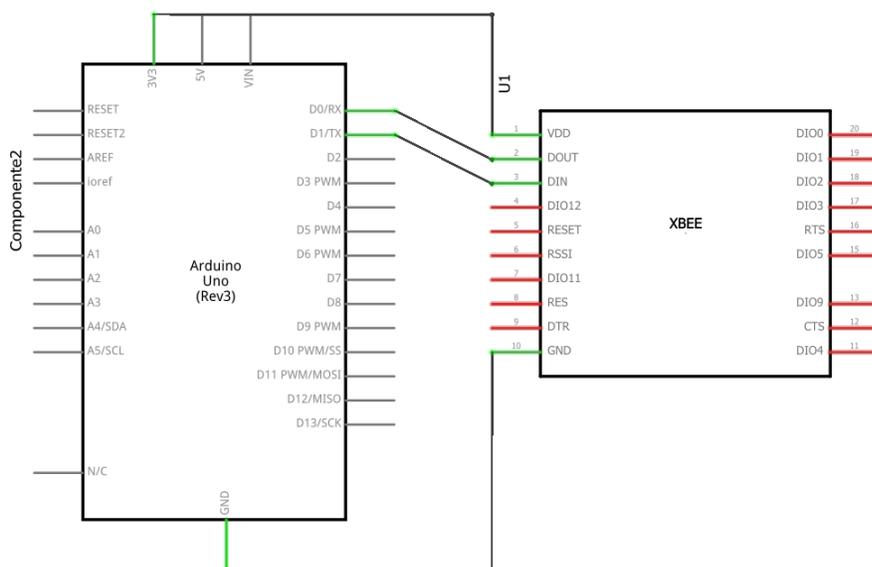


Figura. 40. Esquema de conexión del nodo central con Xbee en fritzing.
Fuente: Autoría.

3.6.4.5. Nodo estación

El nodo estación será el que realice el monitoreo de los sensores de temperatura por lo que éste conlleva subsistemas de detección, procesamiento y comunicación como se muestra en la siguiente Figura. 41.

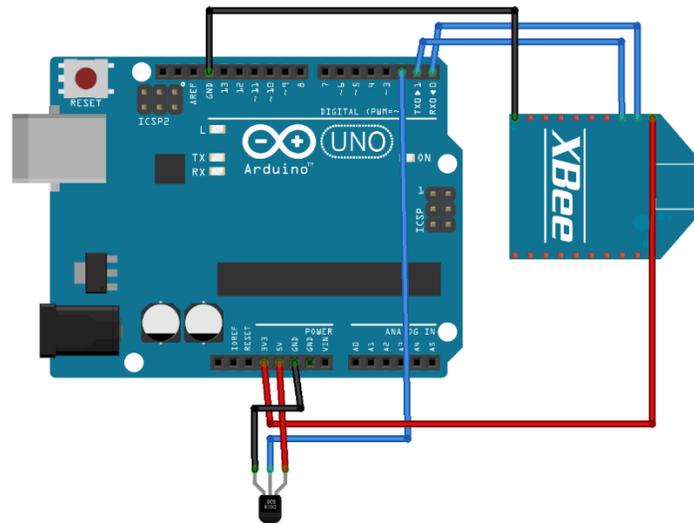


Figura. 41. Esquema eléctrico de un nodo estación con Xbee en fritzing.
Fuente: Autoría.

A continuación, se puede apreciar el esquema de conexión de cada uno de los elementos como se muestra en la Figura. 42, que está a continuación.

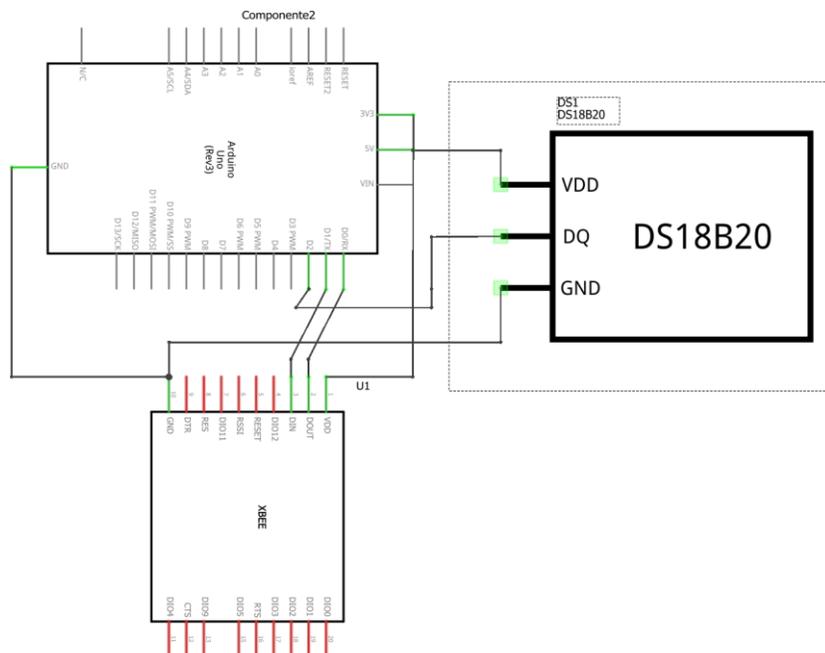


Figura. 42. Esquema de conexión del nodo estación con Xbee en fritzing.
Fuente: Autoría.

Los módulos Xbee funcionan a 3.3V y los pines no son tolerantes a 5V. Desde Arduino podemos alimentar un módulo Xbee, pero la comunicación serie en Arduino es a 5V y en el módulo Xbee es a 3.3V. Para ello se puede utilizar un divisor de tensión para posibilitar el uso de los módulos sin problema como se muestra en la Figura. 43.

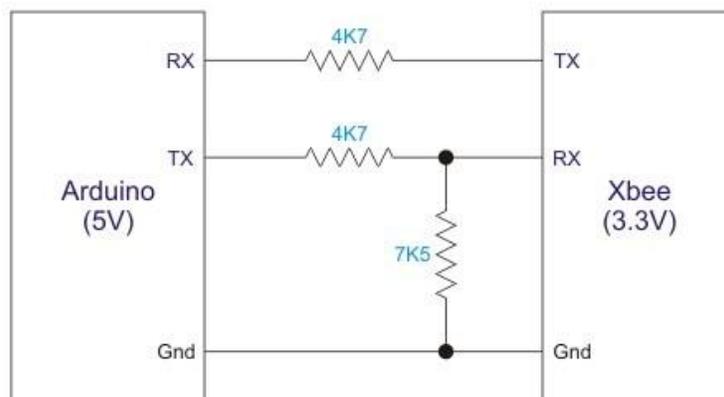
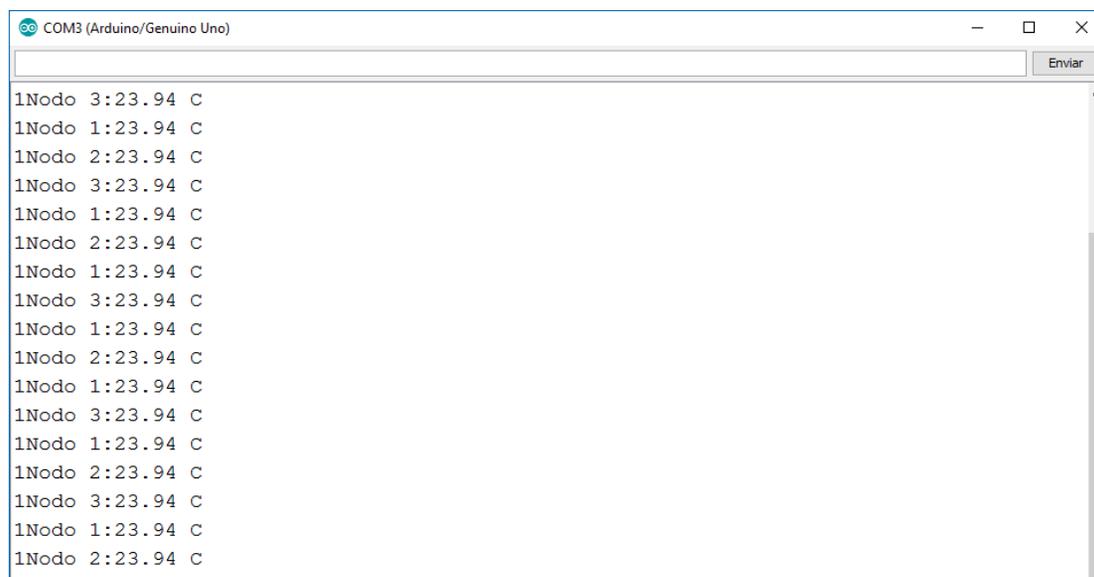


Figura. 43. Divisor de tensión para protección de los módulos Xbee
Fuente: (PARALLAX Z, 2013)

3.6.4.6. Resultados Obtenidos

Como parte de los resultados de la tecnología Xbee, de igual manera para observar su efectividad al manejar datos a velocidades altas, se usa la función "micro" que nos ayuda apreciar el procesamiento de estos datos y cómo se los maneja a lo largo de la prueba. Como se expresó anteriormente se realizó un entorno de análisis controlado. Los datos llegan con gran velocidad y sin problemas de la recepción ni transmisión de estos. Uno de los indicadores que se utiliza es el número de nodos o de clientes que están conectados a la red, en este caso son tres nodos estaciones que están interactuando con el nodo central como se puede apreciar en la siguiente Figura. 44.



```
COM3 (Arduino/Genuino Uno)
1Nodo 3:23.94 C
1Nodo 1:23.94 C
1Nodo 2:23.94 C
1Nodo 3:23.94 C
1Nodo 1:23.94 C
1Nodo 2:23.94 C
1Nodo 1:23.94 C
1Nodo 3:23.94 C
1Nodo 1:23.94 C
1Nodo 2:23.94 C
1Nodo 1:23.94 C
1Nodo 3:23.94 C
1Nodo 1:23.94 C
1Nodo 2:23.94 C
1Nodo 3:23.94 C
1Nodo 1:23.94 C
1Nodo 2:23.94 C
```

Figura. 44. Recepción de datos de nodo central en el sistema Xbee.

Fuente: Autoría.

Además, se puede apreciar la interoperabilidad de los nodos a su nodo central como se aprecia en la Figura. 45.

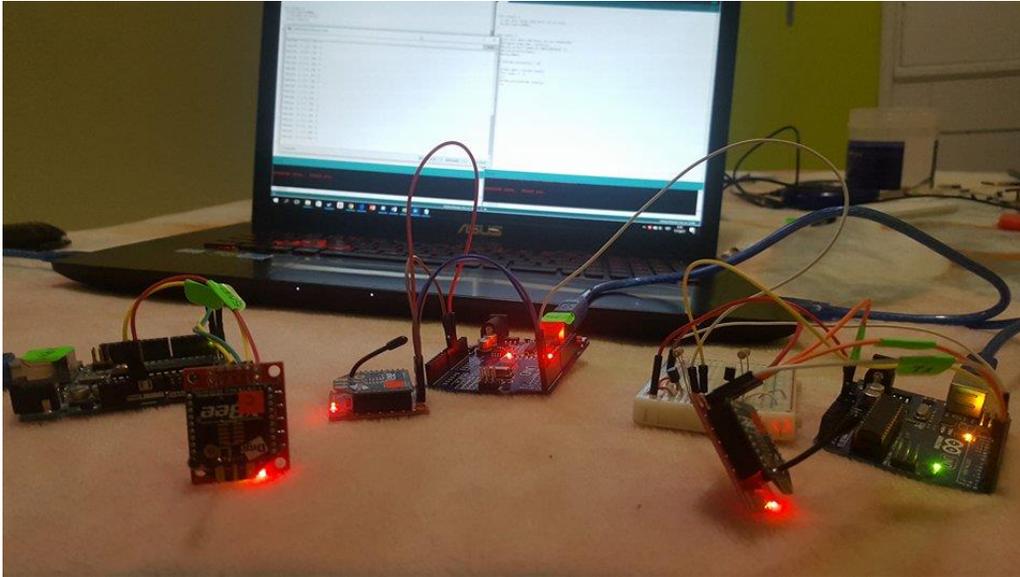


Figura. 45. Comunicación en la red de sensores Xbee.
Fuente: Autoría.

3.5.5. Fase de prueba tres “Estándar de comunicación inalámbrica WiFi”

Esta fase de prueba cuenta con el estándar de comunicación WiFi, para la elaboración de esta red es necesario configurar nuestro IDE de Arduino y configurar nuestro módulo mediante comandos AT, además de que conformara la red de sensores, esquema de conexión y resultados obtenidos.

3.6.5.1. Equipamiento del sistema WiFi

Se describe cada uno de los componentes que van a formar parte para el desarrollo de la red de sensores WiFi, todo esto se detalla en la Tabla 29, que se muestra a continuación.

Tabla 29. Elementos que conformará la red de sensores WiFi

HARDWARE		
CANTIDAD	COMPONENTES	DESCRIPCIÓN
4	Arduino UNO	Plataforma embebida encargada del control e componentes y parte del subsistema de control.
3	DS08L01	Sensor de temperatura
4	ESP8266	Modulo WiFi
4	CABLE USB Tipo A-B	Conexión y fuente de alimentación de la plataforma Arduino con la PC.
4	Protoboard	Montaje de los componentes
-	Cable para Protoboard tipo Macho - Hembra y Macho - Macho	Conexión entre componentes.
1	Computador	Monitoreo, fuente y control de datos.
SOFTWARE		
CANTIDAD	COMPONENTES	DESCRIPCIÓN
-	IDE de Arduino	Software de programación de la plataforma Arduino
-	Fritzing	Diseño de circuitos y esquemas de conexión

Fuente: Autoría.

3.6.5.2. Configuración modulo WiFi ESP8266

Antes de realizar la programación o establecer pines de comunicación, se debe configurar el módulo WiFi ESP8266 para que cumpla s función como un Arduino Uno, para ello hay que descargarse una tarjeta en el IDE de Arduino llamada ESP8266 Community como se puede apreciar en la Figura. 46.

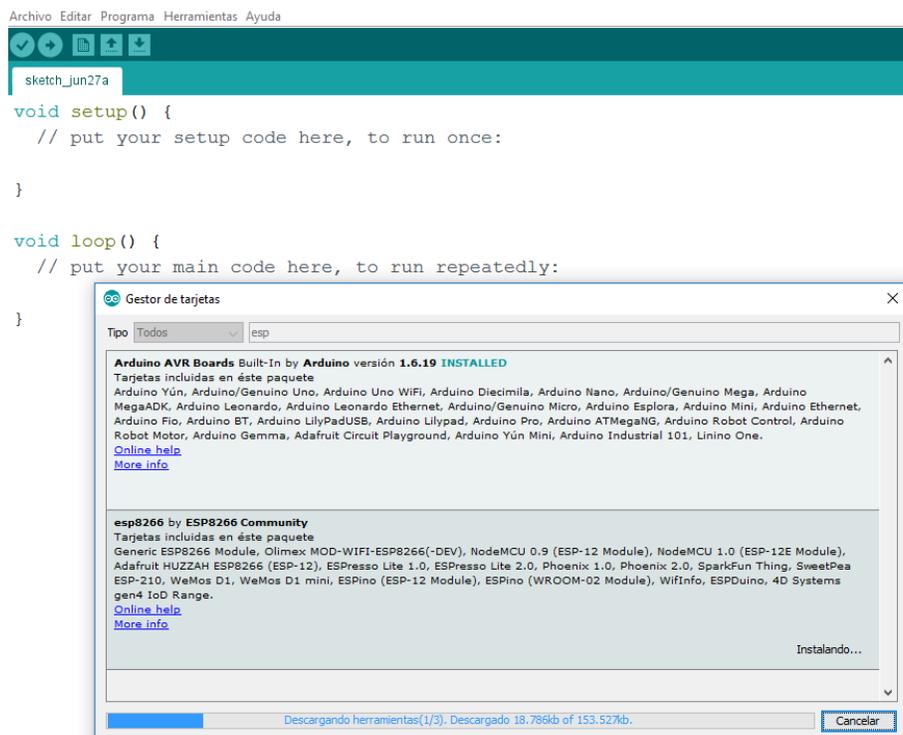


Figura. 46. Gestor de tarjetas ESP8266 para el módulo de comunicación WiFi
Fuente: Autoría

Luego de realizar las configuraciones y tomar el módulo ESP8266 como una tarjeta se puede realizar la programación para las pruebas pertinentes del sistema WiFi.

3.6.5.3. Esquema de conexión en el estándar de comunicación WiFi

El esquema que se presenta a continuación estará conformado con un Módulo ESP826, un Arduino UNO para el nodo central y para el nodo estación contará además de los componentes antes dichos un sensor de temperatura DS15B20, apreciado su conexión.

3.6.5.4. Nodo Central

Dicho anteriormente el nodo central esta conformará por un módulo WiFi ESP8266 y un Arduino Uno y la conexión a realizar para su funcionamiento será como la Figura. 47, que se muestra a continuación.

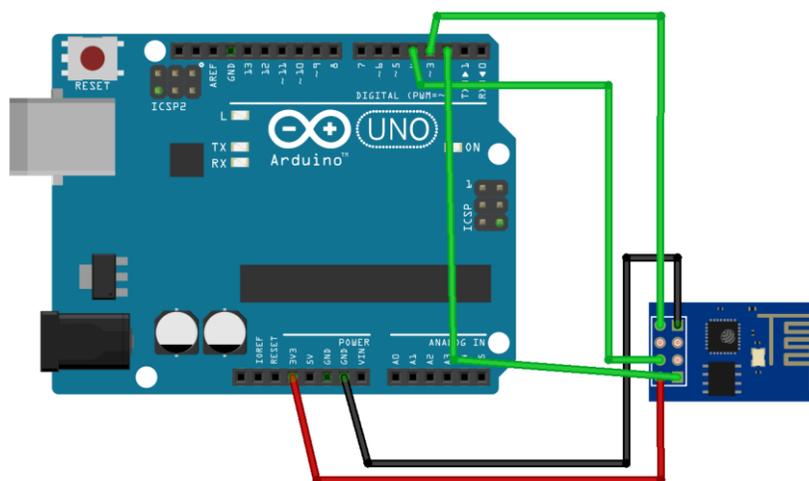


Figura. 47. Esquema de conexión eléctrica del nodo central para el sistema WiFi en Fritzing.
Fuente: Autoría.

De igual manera se puede apreciar el esquemático del nodo central, pines y funciones de cada uno de estos para el sistema WiFi como se muestra en la Figura. 48.

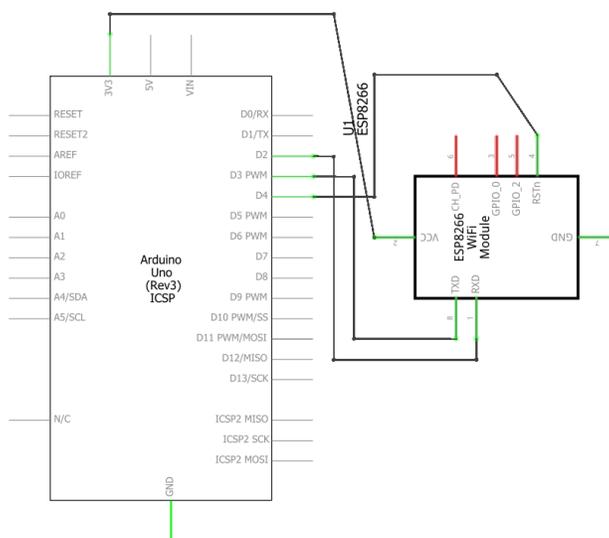


Figura. 48. Esquema de conexiones del sistema WiFi en Fritzing.
Fuente: Autoría.

3.6.5.5. Nodo Estación

El nodo estación contará con el subsistema de procesamiento, detección y control para enviar sus datos al nodo central, este cuenta con cada elemento de comunicación como se muestra en la Figura. 49.

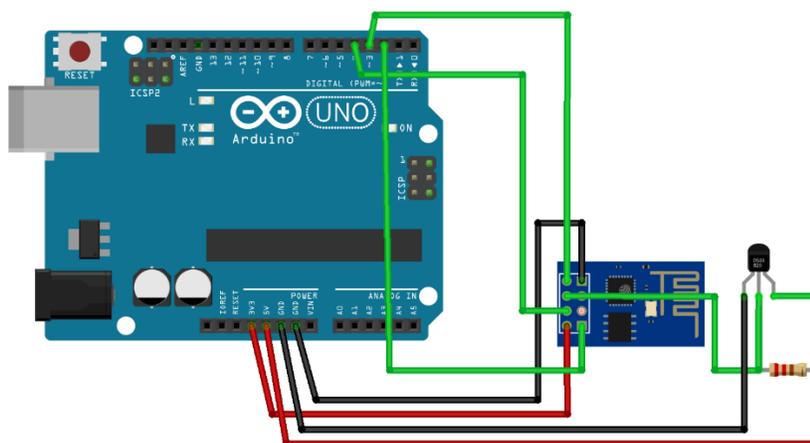


Figura. 49. Esquema eléctrico de conexión del nodo estación en Fritzing.
Fuente: Autoría.

De igual manera se puede apreciar el esquema de conexión de cada uno de los subsistemas en uno solo para crear el Nodo estación como se muestra en la Figura. 50.

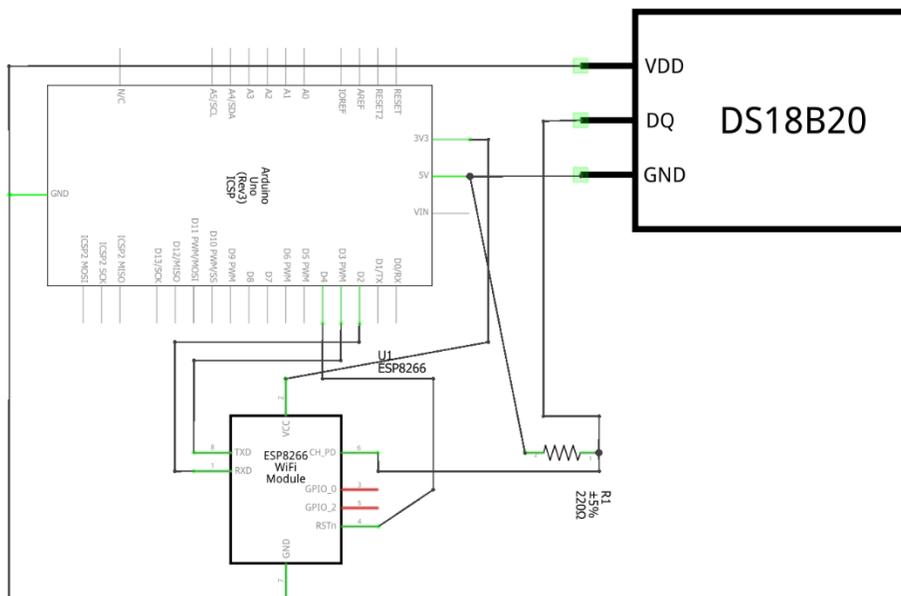


Figura. 50. Esquema de conexión del nodo estación del sistema WiFi en Fritzing.

Fuente: Autoría.

3.6.5.6. Resultados obtenidos

Para apreciar los resultados obtenidos en WiFi, no solo se los podrá visualizar en el terminal del IDE de Arduino, sino que además puede establecer mediante Internet, una visualización en una plataforma que trabaja con este tipo de sensores y proyectos. En éste se puede apreciar como indicador del algoritmo el número de sensores en este caso digitales que trabaja la red de sensores, como se puede apreciar en la Figura. 51 y Figura. 52, que se muestra a continuación.

```

COM3 (Arduino/Genuino Uno)
|
received: Temperatura: 22 #52 _Node151
received: Temperatura: 21 #32 _Node507
received: Temperatura: 21 #33 _Node507
received: Temperatura: 22 #79 _Node848
received: Temperatura: 22 #63 _Node151
received: Temperatura: 22 #34 _Node507
received: Temperatura: 22 #35 _Node507
received: Temperatura: 22 #36 _Node507
received: Temperatura: 22 #61 _Node848
received: Temperatura: 22 #37 _Node507
received: Temperatura: 22 #82 _Node848
received: Temperatura: 21 #38 _Node507
received: Temperatura: 22 #56 _Node151
received: Temperatura: 22 #39 _Node507
received: Temperatura: 22 #83 _Node848
received: Temperatura: 22 #40 _Node507
received: Temperatura: 22 #57 _Node151
received: Temperatura: 22 #42 _Node507
received: Temperatura: 22 #58 _Node151
received: Temperatura: 22 #85 _Node848
received: Temperatura: 22 #43 _Node507
received: Temperatura: 22 #44 _Node507
received: Temperatura: 22 #45 _Node507
received: Temperatura: 22 #46 _Node507
received: Temperatura: 22 #60 _Node151

```

Figura. 51 Datos recolectados en el sistema WiFi – Nodo central

Fuente: Autoría.

Además del indicador de sensores, se puede apreciar un indicador de recibido de información el cual nos permite monitorear cómo está trabajando la información en cada uno de éstos. Dicho anteriormente el nivel de energía es muy importante para cada una de las redes sensores, para este se ha colocado una fuente aparte de la fuente tradicional de Arduino que nos dé mayor amperaje y nos brinde una conexión estable y sólida.

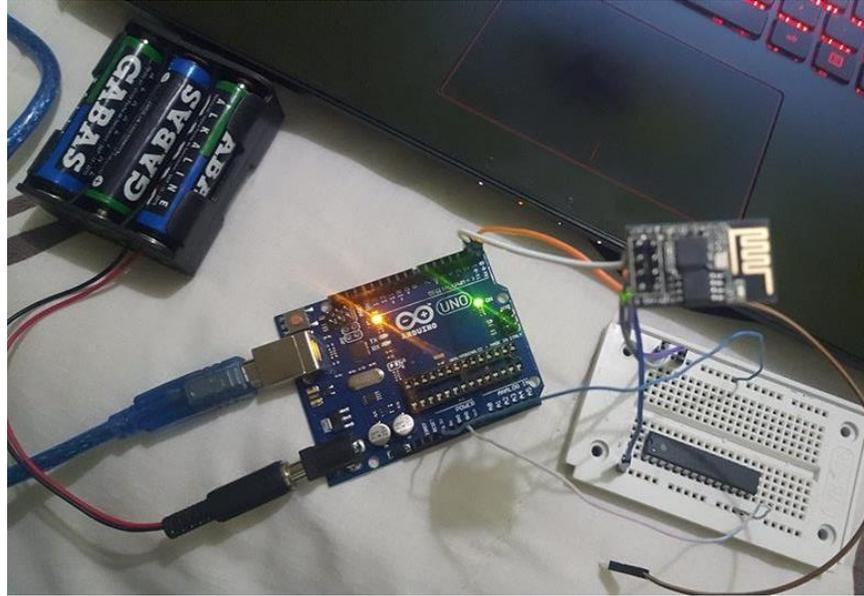


Figura. 52. Red de sensores WiFi, nodo central de la red.
Fuente: Autoría.

3.5.6. Fase de prueba cuatro “Estándar de comunicación inalámbrica Bluetooth”

En este punto se realizará las pruebas pertinentes al estándar de comunicación inalámbrica Bluetooth, para ello hay que realizar las configuraciones correspondientes de cada uno de los módulos mediante comandos AT, tanto su velocidad de trasmisión como el role que tomará en el sistema completo de la red.

3.6.6.1. Equipamiento del sistema Bluetooth

De igual manera se detalla en la Tabla 30 cada uno de los componentes y que función cumplen en el sistema Bluetooth.

Tabla 30. Equipamiento del sistema Bluetooth.

HARDWARE		
CANTIDAD	COMPONENTES	DESCRIPCIÓN
4	Arduino UNO	Plataforma embebida encargada del control e componentes y parte del subsistema de control
3	DS08L01	Sensor de temperatura
4	HC-05	Módulo Bluetooth
4	CABLE USB Tipo A-B	Conexión y fuente de alimentación de la plataforma Arduino con la PC.
4	Protoboard	Montaje de los componentes
-	Cable para Protoboard tipo Macho - Hembra y Macho - Macho	Conexión entre componentes.
1	Computador	Monitoreo, fuente y control de datos.
SOFTWARE		
CANTIDAD	COMPONENTES	DESCRIPCIÓN
-	IDE de Arduino	software de programación de la plataforma Arduino
-	Fritzing	Diseño de circuitos y esquemas de conexión

Fuente: Autoría.

3.6.6.2. Esquema de conexión en el estándar de comunicación Bluetooth

Básicamente para este esquema de conexión se utilizará un Bluetooth HC-05 y un Arduino Uno para el nodo central y para el nodo estación conformará también un sensor de temperatura DS15B20.

3.6.6.3. Configuración Módulo Bluetooth

Antes de realizar la programación de los módulos de comunicación bluetooth hay que realizar la configuración de estos, como modo esclavo para las estaciones y modo maestro además de la velocidad de transmisión en bauds que se van a trabajar que será de 9600 como se muestra a continuación en la Figura. 53.

- Role=1 Modo Maestro
- Role=0 Modo Esclavo

```
COM5
|
* AT+STOP            Get/Set UART stop bit.           *
* AT+INQ             Search slave model             *
* AT+SHOW            Show the searched slave model.   *
* AT+CONN            Connect the index slave model.   *
* AT+IMME            System wait for command when power on. *
* AT+START           System start working.           *
* AT+UUID            Get/Set system SERVER_UUID .     *
* AT+CHAR            Get/Set system CHAR_UUID .       *
* -----*
* Note: (M) = The command support master mode only.   *
* Copyright@2013 www.bolutek.com. All rights reserved. *
*****
OK
+ROLE=1
+PASS=000000
+ADDR=00:00:00:00:00:00
```

Figura. 53. Configuración mediante comandos AT de un módulo Bluetooth

Fuente: Autoría.

3.6.6.4. Nodo Central

De igual manera este contará con el subsistema de procesamiento que será el Arduino Uno y el subsistema de comunicación que será el Módulo Bluetooth y la conexión de estos se realizará como se muestra en Figura. 54.

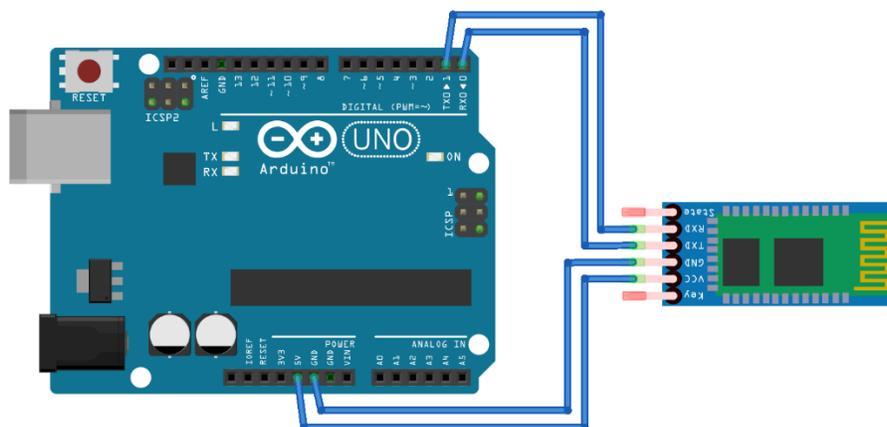


Figura. 54. Esquema eléctrico del Nodo central en fritzing.

Fuente: Autoría.

Además del esquema eléctrico se puede apreciar cada uno de los pines a utilizar en el sistema Bluetooth y de qué manera serán interconectados cada uno como se muestra en la Figura.

55.

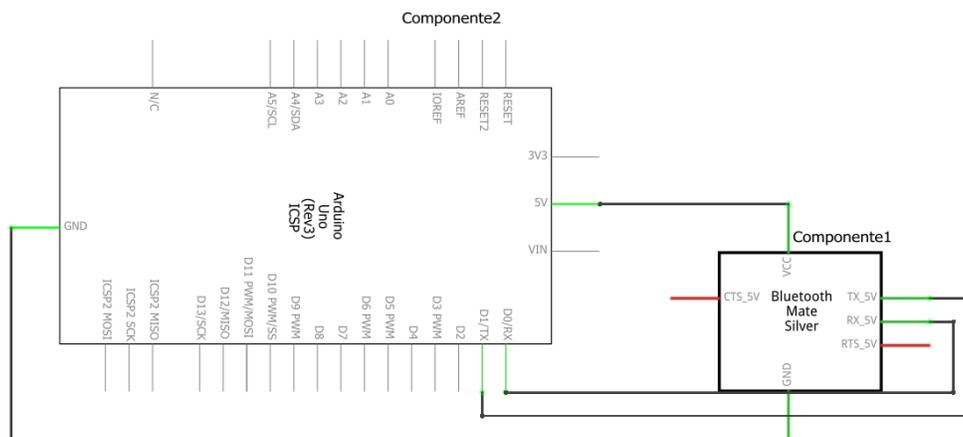


Figura. 55. Esquema de conexión del sistema bluetooth en fritzing.

Fuente: Autoría.

3.6.6.5. Nodo Estación

Este nodo contará con el subsistema de procesamiento” Arduino Uno”, subsistema de detección con el sensor de temperatura, subsistema de comunicación con el módulo Bluetooth como se muestra en la Figura. 56.

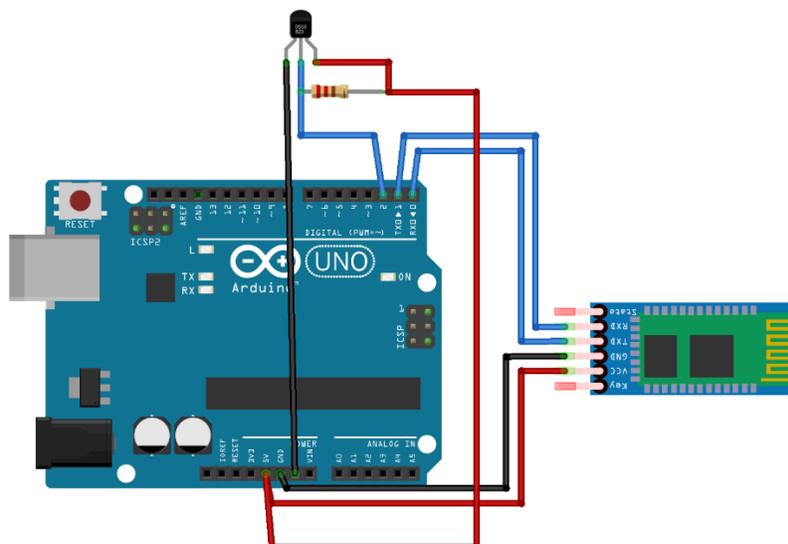


Figura. 56. Esquema eléctrico del nodo estación en fritzing.
Fuente: Autoría.

De igual manera se puede apreciar la interconexión de los subsistemas con sus pines de comunicación como se muestra en la Figura. 57.

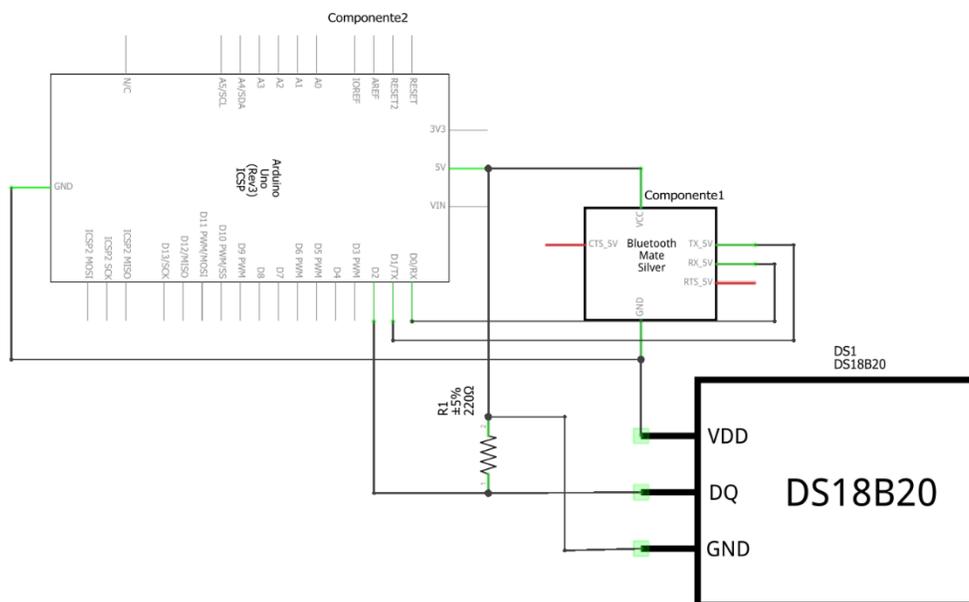
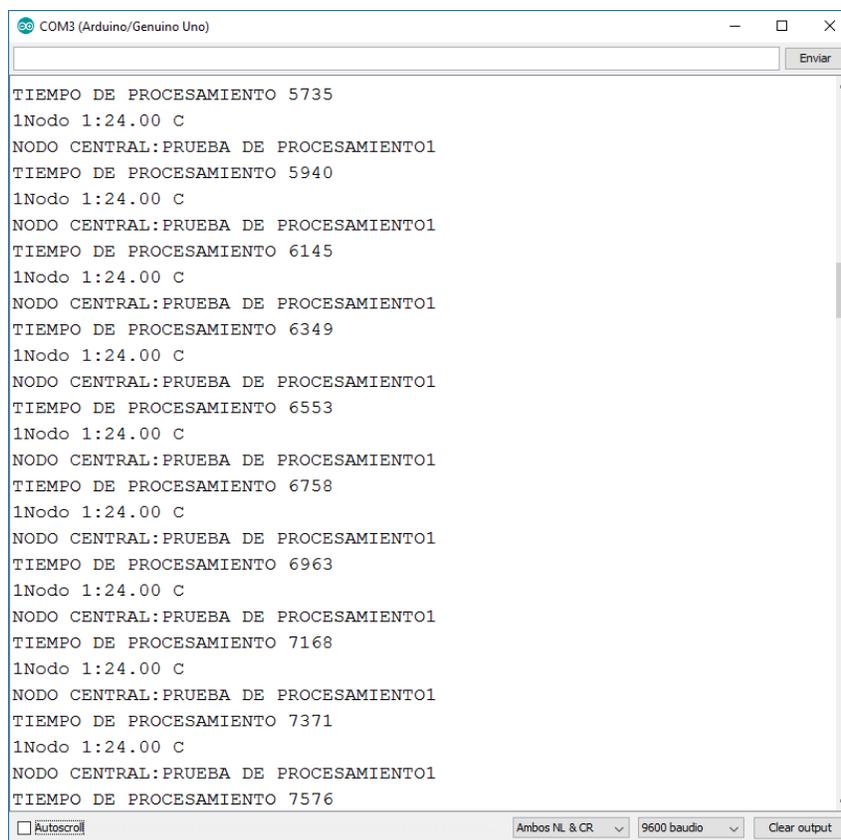


Figura. 57. Esquema de conexión del nodo estación en fritzing
Fuente: Autoría.

3.6.6. Resultados obtenidos

Para los resultados de los nodos se utilizó la función `mili` que nos muestra el tiempo de procesamiento que tiene la red, cabe recalcar que para crear la red se debe establecer la comunicación punto a punto, luego hay que realizar en cambio de la MAC del dispositivo para el siguiente nodo estación, ya que con tecnología bluetooth no es posible establecer una comunicación a todos los nodos sino uno a uno, para este uno de los indicadores del algoritmo es el tipo de dato que se envía en este caso en un dato float decimal que nos expresa la temperatura que nos entrega como se muestra en la Figura. 58.



```
COM3 (Arduino/Genuino Uno)
TIEMPO DE PROCESAMIENTO 5735
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 5940
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 6145
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 6349
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 6553
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 6758
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 6963
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 7168
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 7371
1Nodo 1:24.00 C
NODO CENTRAL:PRUEBA DE PROCESAMIENTO1
TIEMPO DE PROCESAMIENTO 7576
```

Figura. 58. Datos recolectados en la red de sensores Bluetooth.

Fuente: Autoría

Para esta tecnología se puede apreciar que es prescindible utilizar una fuente de 5v, ya que los datos a mejor amperaje y voltaje trabajan con gran eficiencia. Físicamente se puede apreciar como esta en Figura. 59

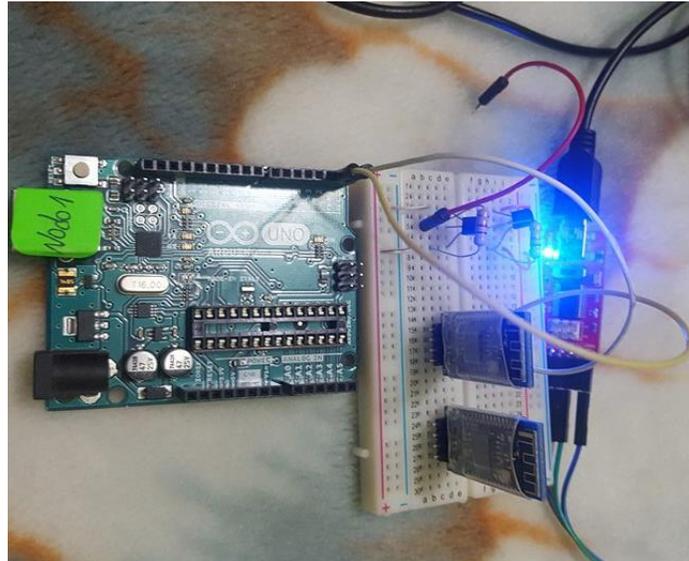


Figura. 59. Red de sensores Bluetooth
Fuente: Autoría.

3.6. Desarrollo del nuevo algoritmo de comunicación

En base a las pruebas realizadas se puede apreciar que se realiza un estudio detallado de cada una de las tecnologías inalámbricas utilizadas, se toma en cuenta que toda la programación está basada para la plataforma Arduino, la explicación y el código que se encuentra en las hojas de Anexo 2, Anexo 3 y Anexo 4, Anexo 5 pueden funcionar en cualquier plataforma de Arduino siempre y cuando posean los periféricos que se deseen utilizar.

El algoritmo cuenta con las variables de comunicación establecidas en los requerimientos y en cada diseño del sistema electrónico. La trama creada en estos protocolos de comunicación cuenta con 11 bytes, de los cuales 2 que son el inicio y el final de la trama son para autenticación

o identificadores de los módulos de comunicación, el segundo y tercero son para el control del uso de la batería en cada sensor de temperatura o sensores que se coloquen dependiendo de cómo se envíe el dato en la trama el receptor indicara su estado; el cuarto, quinto y sexto son los datos analógicos o digitales que genera el sensor de temperatura, el séptimo, octavo, noveno y décimo nos muestran el procesamiento del sistema mientras más alto sea denota que el sistema está en malas condiciones o existe interferencias no deseadas en los puntos de conexión.

En la Tabla 31 se muestra cada uno de los identificadores que forma parte de la trama y ayuda a la codificación y decodificación de cada uno de los datos.

Tabla 31. Identificadores de trama de comunicación

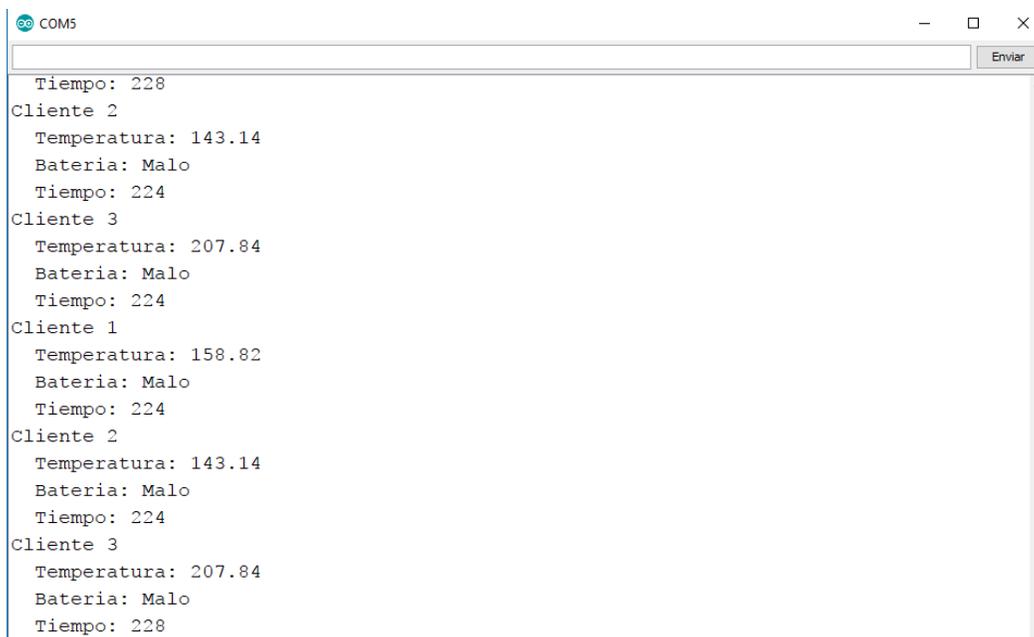
Identificadores	Definición	Estados	Numero de bytes
Primer y último byte	Ayuda a verificar que cliente es, además del inicio y fin de la trama. Se especifica con todas las letras del abecedario, en el caso de que existan más clientes se aumentaría un byte mas a la trama establecida.	A= Cliente Uno B = Cliente dos C= Cliente tres	2
Segundo y tercer byte	Dependiendo del estado de la batería envía un "00" si está en mal estado "01" si está en un estado medio y "10" si está en buenas condiciones. Estos identificadores	00 = Batería en Mal estado 01 = Batería en estado medio	2

		serán decodificados en el nodo central.	10 = Batería en buen estado.
Cuarto, quinto y sexto byte	03u	El dato de temperatura, estos bytes están dado por 3 slots donde indica el dato de temperatura análogo o digital que se esté censando	0 0 0 3
Séptimo, octavo, noveno y décimo byte	04u	Dato de procesamiento, aquí se muestran 4 bytes en donde indica el estado de procesamiento de la red, en si cuento tiempo se demora en decodificar la información.	0224 milisegundos es lo más rápido que decodifica la red, a medida que se emplea 4 mas carga computacional ira subiendo este valor.

Fuente: Propia.

Todo esto se va a ver reflejado en la parte de trasmisión de cada uno de los nodos de comunicación al nodo central.

El nodo central al recoger esta información va a decodificara y procesarla mostrándolo como la Figura. 60 que se muestra a continuación, ya los datos reales y procesados, esto se reflejara en todas las tecnologías inalámbricas ya que es una trama específica y realizada para poder usarla con cualquier plataforma y con los módulos ya especificados.



```
COM5
Tiempo: 228
Cliente 2
  Temperatura: 143.14
  Bateria: Malo
  Tiempo: 224
Cliente 3
  Temperatura: 207.84
  Bateria: Malo
  Tiempo: 224
Cliente 1
  Temperatura: 158.82
  Bateria: Malo
  Tiempo: 224
Cliente 2
  Temperatura: 143.14
  Bateria: Malo
  Tiempo: 224
Cliente 3
  Temperatura: 207.84
  Bateria: Malo
  Tiempo: 228
```

Figura. 60. Recepción de datos Arduino del nodo central.
Fuente: Autoría.

Como se pudo denotar se muestra el cliente que está enviando los datos con sus parámetros establecidos los cuales son: temperatura, batería, tiempo de procesamiento. Dando un control de cada uno de los datos además de su estado actual por parte del nodo central.

En la Figura. 61, podemos apreciar cómo se envía la trama en cada uno de los nodos estaciones, además que se observara cada uno de los identificadores enviados al nodo central.

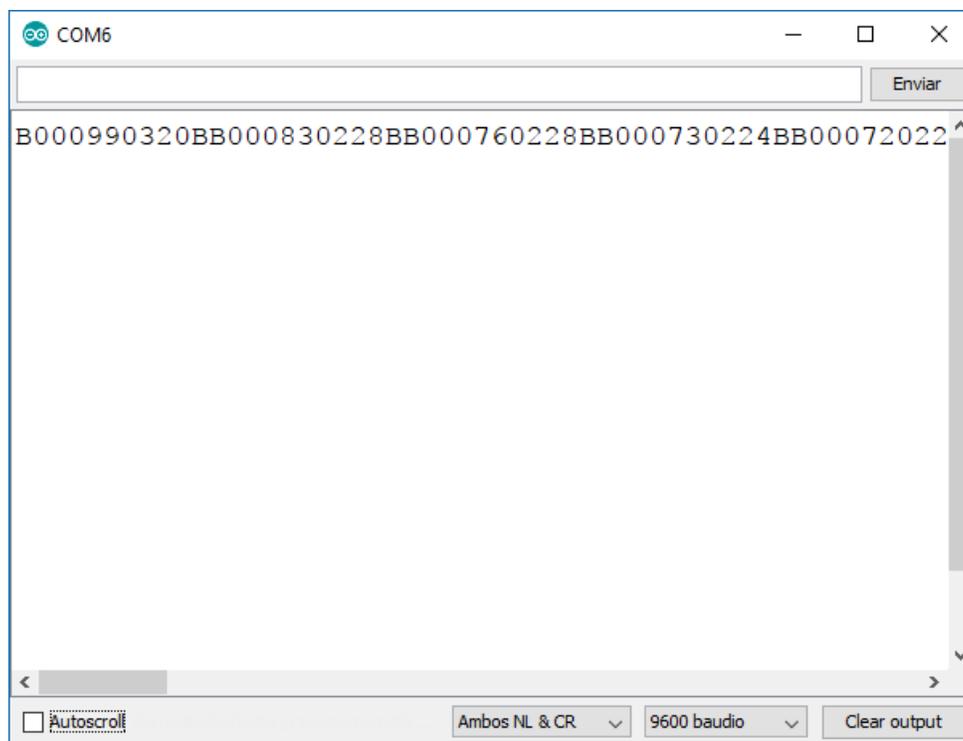


Figura. 61. Trasmisión de datos Arduino en nodo estación.
Fuente: Autoría.

En este caso se envía la trama del nodo B con los identificadores B000990320B, como ya antes mencionado estos datos serán decodificados por parte del nodo central y serán visualizados mediante una interface al usuario. En la Tabla 32 se puede observar con mayor claridad la trama establecida en el nodo B.

Tabla 32. Identificadores de trama

	B	00	099	0320	B
IDENTIFICADOR	INICIO DE CLIENTE	CONTROL DE BATERÍA	CONTROL DE TEMPERATURA	CONTROL DE PROCESAMIENTO	FIN DE PROCESO

Fuente: Autoría.

Establecido todos los parámetros de visualización, codificación y decodificación que se van a dar en el programa para el control de datos en capa aplicación y realizada la fase de pruebas

aplicado el algoritmo en cada una de las tecnologías inalámbricas, podemos validarla mediante el procesamiento de datos que se da en cada una de las tecnologías como existe un cambio de envío o prioridad de paquetes enviados gracias a la trama establecida, además de que los paquetes se envían con mayor eficiencia y menor tiempo, a diferencia de la fase de pruebas sin dicho algoritmo, en la Figura. 62, se puede observar que el tiempo de procesamiento de los datos baja entre las pruebas normales con los protocolos establecidos y las pruebas con el nuevo algoritmo implementado con un promedio de 57,25 en función de microsegundos, dando un menor tiempo de respuesta, además de un envío de paquetes con prioridades.

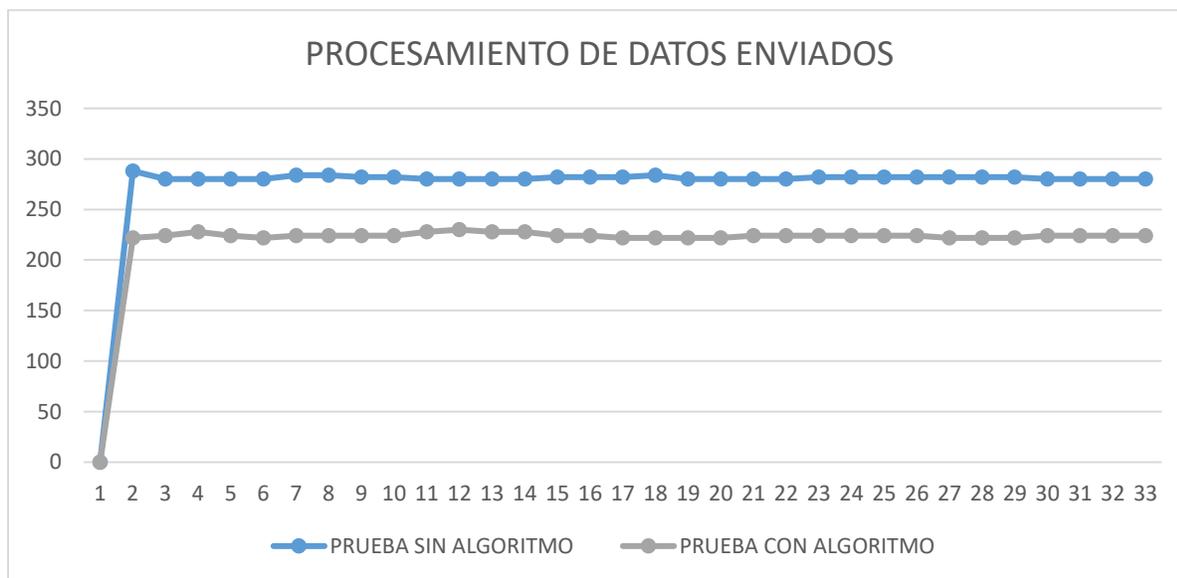


Figura. 62. Tiempo de procesamiento de datos con y sin algoritmo implementado.

Fuente: Autoría.

Luego de realizar las pruebas con cada una de las tecnologías establecidas, se logró demostrar que la creación e implementación del nuevo algoritmo funciona adecuadamente y cumple con los objetivos establecidos, principalmente mejorando el tiempo de respuesta de cada uno de los nodos estación, y ayudando a tener un mejor control de acceso al medio en capa aplicación visualizando el estado de la red de sensores y dando identificadores importantes al

momento de levantar dichas redes inalámbricas, optimizando proyectos basados en Internet de las Cosas o sistemas embebidos, dando una apertura a futuros tesis o proyectos de investigación utilizar dicho algoritmo para una mejor recolección de datos y análisis de estos.

Capítulo 4. Conclusiones y Recomendaciones

4.1. Conclusiones

Se desarrolló un algoritmo flexible a los estándares de comunicación Zigbee, WiFi, Bluetooth y de Radio Frecuencia que, mediante identificadores colocados en la trama de datos como: control de batería, envío de datos analógicos y digitales, identificador de cliente, control de procesamiento y control de dato; la red de sensores tenga mayor rendimiento y control de acceso en capa aplicación, además de que otorga un mejor manejo de volumen de información y que las velocidades de transmisión sean adecuados al sistema embebido empleado.

La investigación del rendimiento de acceso al medio permite generar una ventaja competitiva en el mercado, ya que la implementación de un nuevo algoritmo logra interconectar un mayor número de sensores sin sacrificar el rendimiento de la red, dando prioridad y seguridad al momento de conectar varios nodo estación al nodo central, el tiempo más rápido de decodificación del nuevo algoritmo es de 224 milisegundos a diferencia de algoritmos tradicionales en el que su tiempo es mucho mayor.

Mediante el uso de la metodología del modelo en V y la norma ISO/IEE 29148 permite tener un mayor control del proyecto de titulación al momento de un inconveniente en la elaboración de éste, ya que esto nos permite restablecer la información y enfocarnos en la elaboración eficiente del nuevo algoritmo de comunicación, brindando un proyecto flexible y óptimo para su uso.

Luego de realizar la fase de pruebas en cada uno de los estándares de comunicación y la comparación de cada uno de estos con el nuevo algoritmo diseñado e implementado, se logra reducir el tiempo de respuesta, y el procesamiento de cada uno de los datos como parte de la trama con un promedio de 57,25 milisegundos, además de que el envío de paquetes obtenga prioridades en cada dato.

4.2. Recomendaciones

Es recomendable al momento de realizar la investigación bibliográfica, realizarlas en fuentes de información comprobables o en artículos publicados ya que estos se encuentran en áreas de investigación y son más fiables al realizar proyectos de investigación.

Se recomienda el correcto uso de los divisores de tensión y elementos de protección, para un correcto uso de energía en la red de sensores, ya que si no se tiene un nivel adecuado de estos la red sufre pérdidas de paquetes o una mala comunicación entre subsistemas de detección y comunicación.

Se recomienda que al momento de subir los códigos de IDE de Arduino a su plataforma este se encuentre desconectada de los módulos de comunicación, ya que puede no solo generar problemas al grabarlos sino también puede ocasionar daños a dichos módulos de comunicación.

Es recomendable para el módulo de comunicación WiFi descargar el gestor de tarjeta del ESP8266 para así programarlo desde el IDE de Arduino caso contrario se deberá utilizar otro medio para configurarlos mediante comandos AT y genera mayores inconvenientes al subir dichos y establecer la comunicación deseada.

Antes de realizar el programa deseado con cualquiera de las tecnologías de comunicación inalámbricas, se debe integrar el algoritmo de comunicación y modificar a los parámetros que se van a ejecutar en la red que se desee realizar, para que la red de sensores no genere inconvenientes al momento de compilar o ejecutar el nuevo algoritmo de comunicación.

Es recomendable establecer lineamientos al momento de utilizar este algoritmo ya que cada una de las tecnologías de comunicación inalámbricas posee protocolos de comunicación diferentes y puede ocasionar problemas al momento de grabarlos o censarlos con otras plataformas o módulos de comunicación.

Referencias

- (s.f.). Obtenido de http://zeflo.com/wp-content/uploads/2014/09/esp8266_pinout_h-569x236.png
- Aguilar, S. (2005). Fórmulas para el cálculo de la muestra en investigaciones de salud. *Salud en tabasco*.
- Alloza, J. M. (2014). *Montaje de componentes y periféricos microinformáticos. IFCT0108*. Antequera: IC Editorial.
- Amdahl, G. M. (2012). *Application-customized CPU*. Obtenido de https://upload.wikimedia.org/wikipedia/commons/thumb/5/52/CPU_block_diagram.svg/210px-CPU_block_diagram.svg.png
- AOSONG. (2013). *Aosong (Guangzhou) Electronics Co., Ltd*. Obtenido de <http://www.aosong.com/en/products/index.asp>
- Arduino. (2016). *Arduino*. Obtenido de <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- ARDUINO AG. (2017). Obtenido de <http://www.arduino.org/products/boards/arduino-uno>
- ARDUINO.ag. (16 de 11 de 2016). Obtenido de <https://aprendiendoarduino.wordpress.com/2016/11/16/zigbeexbee/>
- ARDUINO.cc. (s.f.).
- ARDUINO.cl. (2017). Obtenido de <http://arduino.cl/arduino-uno/>
- Baeza, J. P. (2010). Manual de Arduino. *Grupo de Innovación Educativa Automática*.
- Banzi, M. (2011). *Getting Started with Arduino*. Sebastopol: O'Reilly Media, Inc.
- Bernabé, J. D. (2011). Diseño de un sistema embebido para el control en tiempo real. de un convertidor elevador alimentado por una celda de combustible.
- Blanco, J., García, D., Castellnou, M., Molina, D., Grillo, F., & Pous, E. (2008). *Curso Básico de Incendios Forestales*. Lérida: Universitat de Lleida.

Bomberos de Granollers. (Mayo de 2011). *Bombers de Granollers Visita d'escoles al parc.*

Obtenido de http://bombersgranollers.org/documentacio/incendio_forestal/CIF01_texto&imagenes_CAST_Intro.pdf

Castillo, S. (2012). Análisis de tecnologías WIFI y ZIGBEE que optimice las comunicaciones inalámbricas.

CCM Benchmark Group. (Junio de 2014). *CCM - Online Community*. Obtenido de <http://es.ccm.net/faq/9457-benchmark-definicion>

Consejo de Federaciones de Bomberos de Argentina. (2006). *Consejo Nacional Bomberos Argentina*. Obtenido de http://scoutsanpatricio.com.ar/libros/bomberos_03.pdf

Cytron Technologies. (11 de Agosto de 2011). Obtenido de <http://cdn.elgrupoinformatico.com/Manuales/ArduinoDuemilanove300713.png>

Diario La Hora. (21 de Enero de 2002). *LA HORA*. Obtenido de http://lahora.com.ec/index.php/noticias/show/1000049231/-1/Guayabillas_es_bosque_protector_de_Ibarra.html#.Vz4oi_kX3IU

Erazo, J., & Hervas, C. (20 de Octubre de 2014). *Revista Científica Maskana*. Obtenido de <http://www.ucuenca.edu.ec/ojs/index.php/maskana/article/view/532>

F. R. (2014). *ASSA ABLOY*. Obtenido de <http://www.tesa.es/Other/Tesa/PDF/Mec%C3%A1nica%20Nacional/TESA%20dispositivos%20electromecanicos.pdf>

Ferreira, J. V., & Arias, O. V. (2010). 208006 - SISTEMAS EMBEBIDOS.

Flores, C. (2008). MODELO DE PROTOTIPOS. *his pavista*.

freedomdefined.org. (10 de Agosto de 2015). *Definition of Free Cultural Works*. Obtenido de <http://freedomdefined.org/OSHW/translations/es>

Garrido, G. C. (s.f.). Características del consumo en Redes ZigBee. Málaga.

Gartner. (2015). *Internet de las Cosas y ciclo de sobre expectativa de Gartner*. Obtenido de <http://www.iotpreneur.com/internet-de-las-cosas-y-el-ciclo-de-sobreexpectacion-de-gartner-i/>

GeekFactory. (2017). Obtenido de <http://www.geekfactory.mx/tienda/radiofrecuencia/modulo-wifi-serial-esp8266-economico/>

Geocities. (2012). Obtenido de <http://www.geocities.ws/jhdezsan/Fotos/aloha.jpg>

Gobierno Nacional del Ecuador. (2013). *buenvivir.gob.ec*. Obtenido de <http://www.buenvivir.gob.ec/objetivo-7.-garantizar-los-derechos-de-la-naturaleza-y-promover-la-sostenibilidad-ambiental-territorial-y-global>

González, E. D. (2014). *Red de sensores - Internet de las cosas*. Valencia.

IEEE *Design & Test of Computers*. (2001). Obtenido de http://i0.wp.com/wirelessmash.com/blog/wp-content/uploads/2012/10/sensor_node.jpg

IngenieríaElectrónica.Org. (8 de Agosto de 2015). *INGENIERIAELECTRONICA.ORG*. Obtenido de <http://ingenieriaelectronica.org/definicion-y-caracteristicas-de-led-diodo-emisor-de-luz/>

J. C. (s.f.). *FILIFELOP*. Obtenido de <http://www.filipeflop.com/pd-6b876-modulo-xbee-s2c-zigbee-wire-antenna.html>

J. D. (2011). ANÁLISIS DEL PROTOCOLO ZIGBEE .

Jonathan. (16 de Junio de 2012). *Teorías de aprendizaje*. Obtenido de http://2.bp.blogspot.com/-KMEq_f60XBU/T84nplKlc6I/AAAAAAAAAADO/cPFNIFRDGgA/s1600/12121.jpg

- Llamas, L. (2017). *Ingeniería, Informática y diseño*. Obtenido de <https://www.luisllamas.es/comunicacion-inalambrica-a-2-4ghz-con-arduino-y-nrf24l01/>
- Martínez Pérez, L. S. (10 de Septiembre de 2012). *Ingeniería de Software*. Obtenido de <http://softwareverde.blogspot.com/2012/09/modelo-en-v.html>
- Martínez, R. F., Meré, J. O., de Pisón Ascacíbar, F. J., Marcos, A. G., & Elías, F. A. (2009). *Redes Inalámbricas de sensores: teoría y aplicación práctica*. Publicaciones Universidad de la Rioja.
- Ministerio de Ambiente y Desarrollo Sustentable de Argentina. (16 de Junio de 2016). *MAYDS Argentina*. Obtenido de <http://ambiente.gob.ar/servicio-nacional-de-manejo-del-fuego/incendios-forestales/>
- Miñarro, B. Ú. (2009). *Sistemas embebidos*.
- Mireya. (27 de Abril de 2012). *PROTOCOLO ALOHA*.
- Núñez, C. V., Peña, J. C., & Farzón, L. C. (s.f.). Análisis comparativo de tecnologías inalámbricas para una solución de servicios de telemedicina. *ISSN*.
- Núñez, C. V., Peña, J. C., & Garzón, C. L. (s.f.). Análisis comparativo de tecnologías inalámbricas para una solución de servicios de telemedicina.
- OMS. (2004). *GUÍAS PARA LA CALIDAD DEL AIRE*. Lima: Centro Panamericano de Ingeniería Sanitaria y Ciencias del Ambiente.
- Pahdaman, S. M. (2014). Implementación y Evaluación del protocolo de sincronización RBS para las redes inalámbricas de sensores. 7-13.
- PARALLAX Z. (2013). Obtenido de <https://www.parallax.com/sites/default/files/downloads/32401-XBee-5V-3.3V-Adapter-v1.2.pdf>

(2009-2013). *Plan Nacional Del Buen Vivir*. Quito.

Pressman, R. S. (2017). *Ingeniería de Software*. *EcuRed*.

Ramírez, M. M. (s.f.). *Tecnología Bluetooth*. Obtenido de <http://www.gestiopolis.com/tecnologia-bluetooth/>

Real Academia Española. (Enero de 2017). *Diccionario de la lengua española | Real Academia Española*. Obtenido de <http://www.rae.es/ayuda/diccionario-de-la-lengua-espanola>

Redes. (2006). Obtenido de http://www.mfbarcell.es/docencia_uned/redes/tema_10/redes_cap_10.pdf

Repositorio UPTC. (2015). Obtenido de <https://sisteminformacii.wikispaces.com/file/view/Figura1.JPG/270812306/Figura1.JPG>

Roque, F. P., Zaldívar, E. V., & O. A. (2013). *Sistema de Adquisición de Datos con comunicación inalámbrica*.

Secretaría de Gestión de Riesgos. (2015). *Secretaría de Gestión de Riesgos | Ecuador*. Obtenido de <http://www.gestionderiesgos.gob.ec/>

Solorio, M. (16 de Abril de 2013). *Metodología en cascada*. Obtenido de [http://3.bp.blogspot.com/-](http://3.bp.blogspot.com/-VCq5TNHp5HY/UWzPqIr2JYI/AAAAAAAAAAU/LJ7QSJnMOck/s1600/modelo-en-cascada.png)

[VCq5TNHp5HY/UWzPqIr2JYI/AAAAAAAAAAU/LJ7QSJnMOck/s1600/modelo-en-cascada.png](http://3.bp.blogspot.com/-VCq5TNHp5HY/UWzPqIr2JYI/AAAAAAAAAAU/LJ7QSJnMOck/s1600/modelo-en-cascada.png)

UDLAP. (2011). Obtenido de http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo3.pdf

UNICAN. (2015). Obtenido de <https://www.tlmat.unican.es/siteadmin/submaterials/2061.pdf>

Valvano, J. (2009). *Introducción a los Sistemas de Microcomputadora Embebidos*.

Glosario de Términos y Acrónimos

Arduino: Plataforma de creación de prototipos electrónicos de código abierto que permite a los usuarios crear objetos electrónicos interactivos.

CIERCOM: Carrera de Ingeniería en Electrónica y Redes de Comunicación

Hardware Libre: Hardware de código abierto, electrónica libre o máquinas libres a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público.

IDE: Un entorno de desarrollo integrado o entorno de desarrollo interactivo

MOTE: Es un nodo en un sensor de red que es capaz de realizar algún procesamiento, reuniendo información sensible y comunicando con otros nodos conectados en la red

RF: Denominado espectro de radiofrecuencia es un término que se aplica a la porción menos energética del espectro electromagnético

WIFI: Mecanismo de conexión de dispositivos electrónicos de forma inalámbrica

Interconexión: Conexión entre sí de dos o más elementos

ALU: Unidad aritmética lógica o unidad aritmética-lógica

Divisor de tensión: Configuración de circuito eléctrico que reparte la tensión de una fuente entre una o más impedancias conectadas en serie

Procesamiento: Conjunto de operaciones que un ordenador realiza partiendo de un programa.

Microcontrolador: es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como microcomputadora.

Zigbee: 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 de redes inalámbricas de área personal.

Subsistema: Es un sistema que es parte de otro sistema mayor que lo contiene. En otras palabras, un subsistema es un conjunto de elementos interrelacionados que, en sí mismo, es un sistema, pero a la vez es parte de un sistema superior.

Interfaz: Dispositivo capaz de transformar las señales generadas por un aparato en señales comprensibles por otro.

Anexos

Anexo 1. Encuesta realizada a estudiantes de CIERCOM y tabulación de resultados.



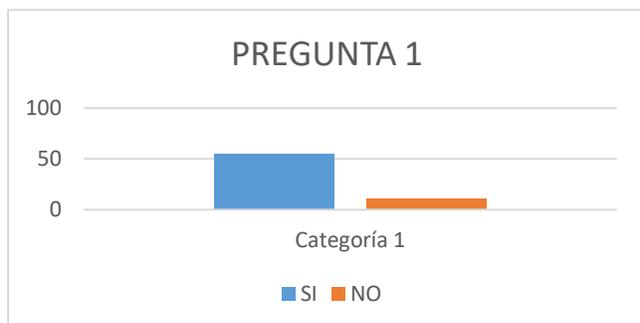
UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

ENCUESTA

Indicaciones: Por favor contestar la presente encuesta según su criterio u opinión marcándola con una X.

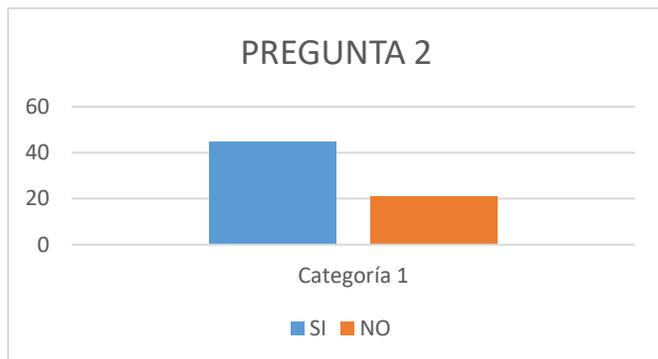
¿Usted ha recibido alguna materia referente a tecnologías inalámbricas o sistemas electrónicos?

SI: _____ NO: _____



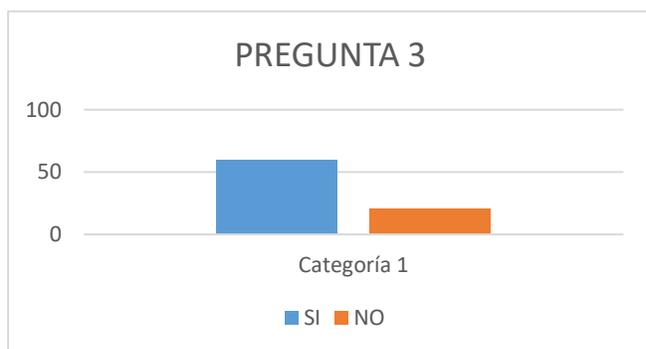
¿Ha escuchado o tiene conocimiento acerca de Sistemas Embebidos con módulos de comunicación inalámbricos?

SI: _____ NO: _____



¿Conoce alguno de estos estándares de comunicación inalámbricos: ¿RF, BLUETOOTH, Zigbee o WIFI?

SI: _____ NO: _____



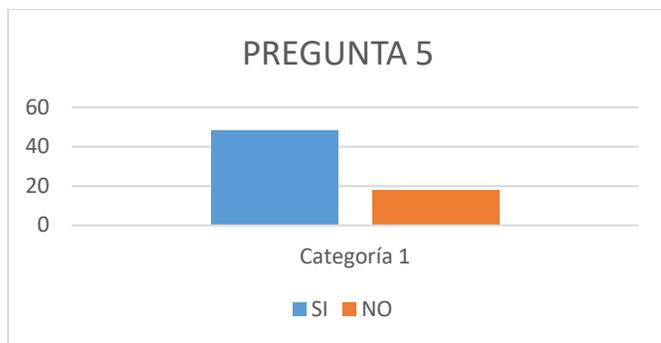
¿Considera que los laboratorios de CIERCOM posee este tipo de tecnología?

SI: _____ NO: _____



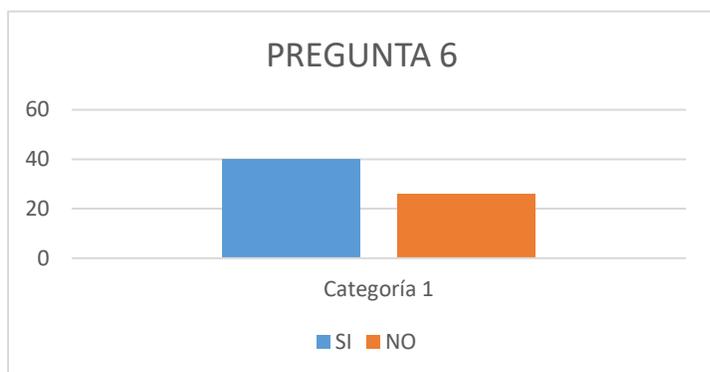
¿Conoce que es una red de sensores y ha utilizado este tipo de red en proyectos de clase?

SI: _____ NO: _____



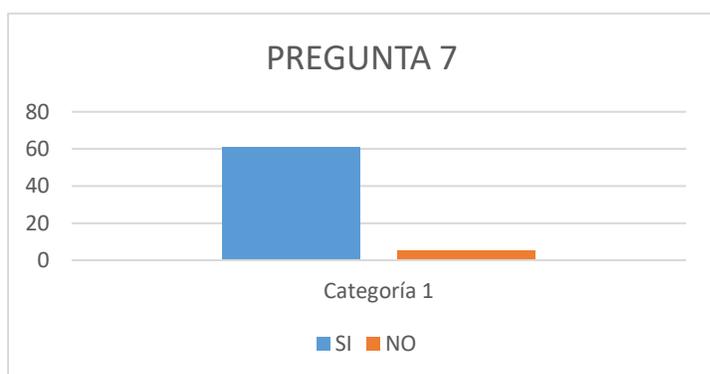
¿Conoce algún algoritmo o esquema técnico que le ayude al desenvolvimiento eficiente de proyectos en una red de sensores inalámbricos?

SI: _____ NO: _____



¿Te gustaría aumentar la eficiencia de transmisión de datos en una red de sensores?

SI: _____ NO: _____



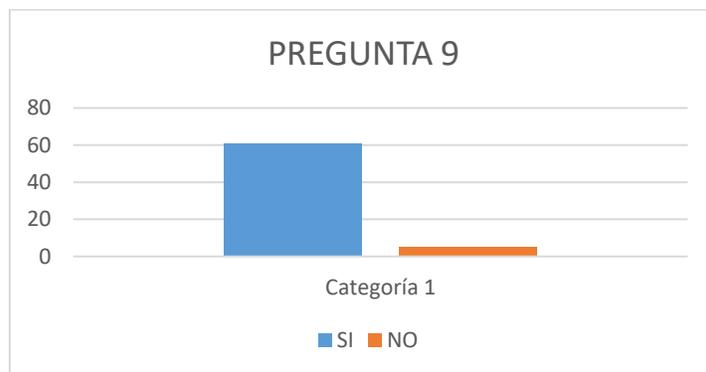
¿Conoce usted alguna estructura o subsistema para levantar una red de sensores?

SI: _____ NO: _____



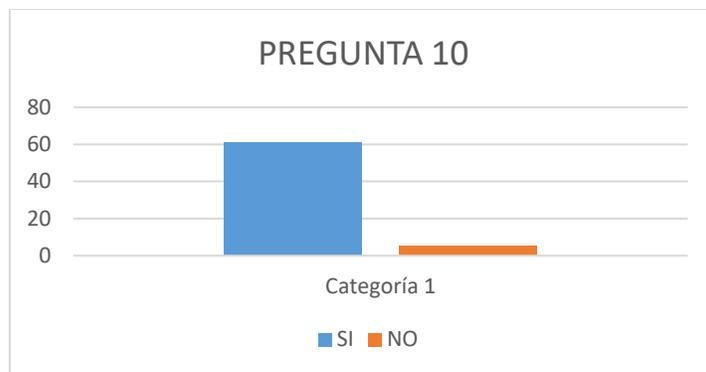
¿Alguna vez ha tenido problemas en el rendimiento de una red de sensores al trabajar con grandes cantidades de datos en el sistema?

SI: _____ NO: _____



¿Usted utilizaría un algoritmo en capa aplicación para optimizar la transmisión de datos en una red de sensores?

SI: _____ NO: _____



¿Te gustaría que tu sistema cuente con un algoritmo que te ayude con la visualización del rendimiento de tu red de sensores además de un control de los tipos y número de sensores que utilices?

SI: _____

NO: _____



Anexo 2. Datasheet Módulo NRF24L01

PRELIMINARY PRODUCT SPECIFICATION 

Single chip 2.4 GHz Transceiver **nRF24L01**

FEATURES

- True single chip GFSK transceiver
- Complete OSI Link Layer in hardware
- Enhanced ShockBurst™
- Auto ACK & retransmit
- Address and CRC computation
- On the air data rate 1 or 2Mbps
- Digital interface (SPI) speed 0-8 Mbps
- 125 RF channel operation
- Short switching time enable frequency hopping
- Fully RF compatible with nRF24XX
- 5V tolerant signal input pads
- 20-pin package (QFN20 4x4mm)
- Uses ultra low cost +/- 60 ppm crystal
- Uses low cost chip inductors and 2-layer PCB
- Power supply range: 1.9 to 3.6 V

APPLICATIONS

- Wireless mouse, keyboard, joysti
- Keyless entry
- Wireless data communication
- Alarm and security systems
- Home automation
- Surveillance
- Automotive
- Telemetry
- Intelligent sports equipment
- Industrial sensors
- Toys

GENERAL DESCRIPTION

nRF24L01 is a single chip radio transceiver for the world wide 2.4 - 2.5 GHz ISM band. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator, a demodulator, modulator and Enhanced ShockBurst™ protocol engine. Output power, frequency channels, and protocol setup are easily programmable through a SPI interface. Current consumption is very low, only 9.0mA at an output power of -6dBm and 12.3mA in RX mode. Built-in Power Down and Standby modes makes power saving easily realizable.

QUICK REFERENCE DATA

Parameter	Value	Unit
Minimum supply voltage	1.9	V
Maximum output power	0	dBm
Maximum data rate	2000	kbps
Supply current in TX mode @ 0dBm output power	11.3	mA
Supply current in RX mode @ 2000 kbps	12.3	mA
Temperature range	-40 to +85	°C
Sensitivity @ 1000 kbps	-85	dBm
Supply current in Power Down mode	900	nA

Table 1 nRF24L01 quick reference data

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

PIN FUNCTIONS

Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin
7	VDD	Power	Power Supply (+3V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal Pin 2
10	XC1	Analog Input	Crystal Pin 1
11	VDD_PA	Power Output	Power Supply (+1.8V) to Power Amplifier
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+3V DC)
16	IREF	Analog Input	Reference current
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+3V DC)
19	DVDD	Power Output	Positive Digital Supply output for de-coupling purposes
20	VSS	Power	Ground (0V)

Table 3 nRF24L01 pin function

PIN ASSIGNMENT

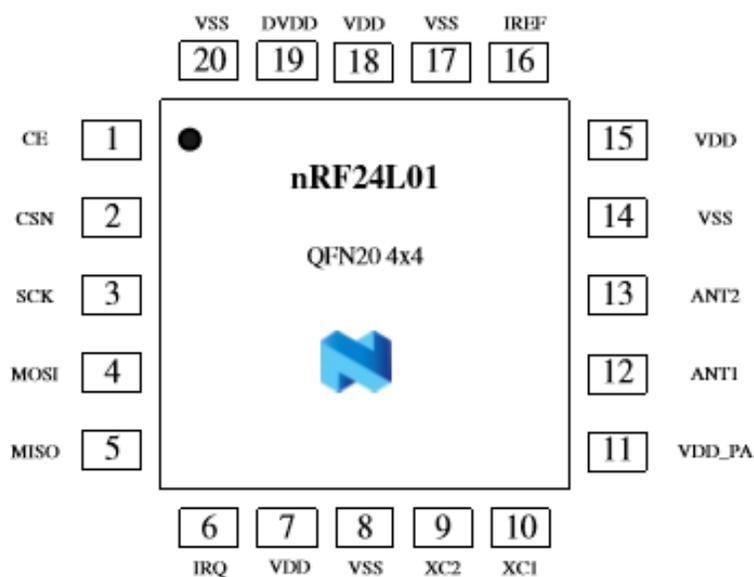


Figure 2 nRF24L01 pin assignment (top view) for a QFN20 4x4 package.

Anexo 3. Datasheet Módulo ESP8266

Table 1 below describes the major parameters.

Table 1 Parameters

Categories	Items	Values
WiFi Parameters	WiFi Protocoles	802.11 b/g/n
	Frequency Range	2.4GHz-2.5GHz (2400M-2483.5M)
Hardware Parameters	Peripheral Bus	UART/HSPI/I2C/I2S/Ir Remote Control
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	14.3mm*24.8mm*3mm
	External Interface	N/A
Software Parameters	Wi-Fi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network) / download and write firmware via host
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Table 2 Pin Descriptions

NO.	Pin Name	Function
1	GND	GND
2	GPIO2	GPIO,Internal Pull-up
3	GPIO0	GPIO,Internal Pull-up
4	RXD	UART0,data received pin RXD
5	VCC	3.3V power supply (VDD)
6	RST	1) External reset pin, active low 2) Can loft or external MCU
7	CH_PD	Chip enable pin. Active high
8	TXD	UART0,data send pin RXD

Anexo 4. Datasheet Módulo Xbee S2 C

Performance specifications

The following table describes the performance specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Indoor/urban range	Up to 60 m (200 ft)	Up to 90 m (300 ft)	Up to 60 m (200 ft)
Outdoor RF line-of-sight range	Up to 1200 m (4000 ft)	Up to 3200 m (2 mi)	Up to 1200 m (4000 ft)
Transmit power output (maximum)	6.3 mW (+8 dBm), boost mode 3.1 mW (+5 dBm), normal mode channel 26 max power is +3 dBm	63 mW (+18 dBm)	6.3 mW (+8 dBm) channel 26 max power is +1 dBm
RF data rate	250,000 b/s		
Receiver sensitivity	-102 dBm, boost mode -100 dBm, normal mode	-101 dBm	-102 dBm, boost mode -100 dBm, normal mode

Power requirements

The following table describes the power requirements for the XBee/XBee-PRO ZigBee RF Module.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Adjustable power	Yes		
Supply voltage	2.1 - 3.6 V 2.2 - 3.6 V for programmable version	2.7 - 3.6 V	2.1 - 3.6 V
Operating current (transmit)	45 mA (+8 dBm, boost mode) 33 mA (+5 dBm, normal mode)	120 mA @ +3.3 V, +18 dBm	45 mA
Operating current (receive)	31 mA (boost mode) 28 mA (normal mode)	31 mA	31 mA
Power-down current	< 1 μ A @ 25°C		< 3 μ A @ 25°C

General specifications

The following table describes the general specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Operating frequency band	ISM 2.4 - 2.5 GHz		
Form factor	through-hole, surface-mount		surface-mount

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Dimensions	through-hole: 2.438 x 2.761 cm (0.960 x 1.087 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	through-hole: 2.438 x 3.294 cm (0.960 x 1.297 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)
Operating temperature	-40 to 85 °C (industrial)		
Antenna options	through-hole: PCB antenna, U.FL connector, RPSMA connector, or integrated wire surface-mount: RF pad, PCB antenna, or U.FL connector		

Networking and security specifications

The following table describes the networking and security specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Supported network topologies	Point-to-point, point-to-multipoint, peer-to-peer, and DigiMesh		
Number of channels	16 Direct sequence channels	15 Direct sequence channels	16 Direct sequence channels
Interface immunity	Direct Sequence Spread Spectrum (DSSS)		
Channels	11 to 26		
Addressing options	PAN ID and addresses, cluster IDs and endpoints (optional)		

Communication interface specifications

The following table provides the device's communication interface specifications.

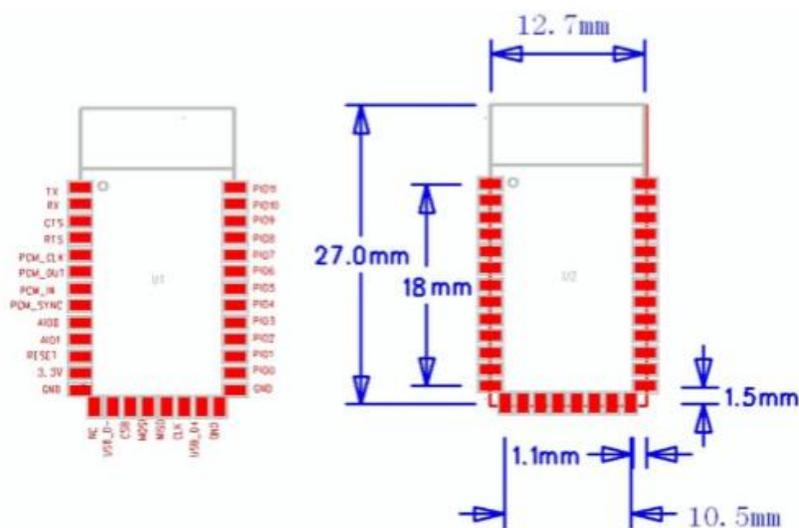
Interface options	
UART	250 Kb/s maximum
SPI	5 Mb/s maximum (burst)

Anexo 5. Datasheet Módulo Bluetooth

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



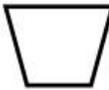
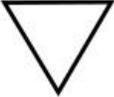
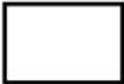
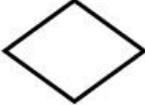
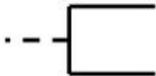
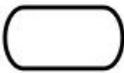
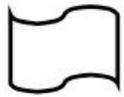
Anexo 6. Diagrama de flujos

SIMBOLOGIA DIAGRAMAS DE FLUJO

RODRIGUEZ TENORIO JHOISETH ALINA SEC: 4IV1

INFORMATICA APLICADA A LA INGENIERIA

SIMBOLOGIA DIAGRAMAS DE FLUJO

SÍMBOLO	REPRESENTA	SÍMBOLO	REPRESENTA
	Terminal. Indica el inicio o la terminación del flujo, puede ser acción o lugar; además se usa para indicar una unidad administrativa o persona que recibe o proporciona información.		Documento. Representa cualquier tipo de documento que entra, se utilice, se genere o salga del procedimiento.
	Disparador. Indica el inicio de un procedimiento, contiene el nombre de éste o el nombre de la unidad administrativa donde se da inicio		Archivo. Representa un archivo común y corriente de oficina.
	Operación. Representa la realización de una operación o actividad relativas a un procedimiento.		Conector. Representa una conexión o enlace de una parte del diagrama de flujo con otra parte lejana del mismo.
	Decisión o alternativa. Indica un punto dentro del flujo en que son posibles varios caminos alternativos.		Conector de página. Representa una conexión o enlace con otra hoja diferente, en la que continúa el diagrama de flujo.
	Nota aclaratoria. No forma parte del diagrama de flujo, es un elemento que se adiciona a una operación o actividad para dar una explicación.		Línea de comunicación. Proporciona la transmisión de información de un lugar a otro mediante?
SÍMBOLO	REPRESENTA	SÍMBOLO	REPRESENTA
	Operación con teclado. Representa una operación en que se utiliza una perforadora o verificadora de tarjeta.		Dirección de flujo o línea de unión. Conecta los símbolos señalando el orden en que se deben realizar las distintas operaciones.
	Tarjeta perforadora. Representa cualquier tipo de tarjeta perforada que se utilice en el procedimiento.		Cinta magnética. Representa cualquier tipo de cinta magnética que se utilice en el procedimiento.
	Cinta perforada. Representa cualquier tipo de cinta perforada que se utilice en el procedimiento.		Teclado en línea. Representa el uso de un dispositivo en línea para promocionar información a una computadora electrónica u obtenerla de ello.
NOTA: Los símbolos marcados con * son utilizados en combinación con el resto cuando se está elaborando un diagrama de flujo de un procedimiento en el cual interviene algún equipo de procesamiento electrónico.			

Anexo 7. Código nodo central Módulo RF

```

#include <SPI.h> //Librería del módulo NRF24101
#include "nRF24L01.h" //Librería del módulo NRF24101
#include "RF24.h" //Librería del módulo NRF24101
float temperature; //variable del valor de temperatura
float NODO2; //variable del Nodo estable 2 de temperatura

RF24 radio(9, 10); //pines que se van a utilizar para el modulo
const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL }; //clientes de conexión
void setup(void)
{
  pinMode(10, OUTPUT); //pin en modo salida de los módulos NRF
  Serial.begin(9600); //velocidad de transmisión serial
  radio.begin(); //velocidad de modulo
  radio.setRetries(15, 15); //Máximo de reintentos
  radio.startListening(); //inicio para escuchar canal
  radio.openWritingPipe(pipes[1]); //canal de escritura
  radio.openReadingPipe(1, pipes[0]); //canal de lectura

  //radio.openWritingPipe(pipes[0]);
}
void loop(void)
{
  if ( radio.available() ) // Si hay datos disponibles
  {

    float temperature2 = 0;
    bool done = false;
    while (!done) // Espera aquí hasta recibir algo
    {
      done = radio.read( &NODO2, sizeof(float));
      delay(100);
      Serial.print("Dato Recibido =");
      Serial.println(NODO2);
      Serial.println("Enviando Respuesta nodo 2");
      delay(150); // Para dar tiempo al emisor
    }

    // doneTwo = radio.read( &temperature2, sizeof(float));
    bool doneTwo = false;
    while (!doneTwo) // Espera aquí hasta recibir algo

```

```
{
  delay(100);
  Serial.print("Dato Recibido =");
  Serial.println(temperature2);
  Serial.println("Enviando Respuesta nodo 1");
  delay(20); // Para dar tiempo al emisor
}
bool doness = false;
while (!doness) // Espera aquí hasta recibir algo
{
  delay(100);
  Serial.print("Dato Recibido =");
  Serial.println(temperature2);
  Serial.println("Enviando Respuesta nodo 3");
  delay(20); // Para dar tiempo al emisor
}
```

Anexo 8. Código nodo estación Módulo RF

```

#include <SPI.h> //Librería del módulo NRF24101
#include "nRF24L01.h" //Librería del módulo NRF24101
#include "RF24.h" //Librería del módulo NRF24101
#include <DallasTemperature.h> //Librería del sensor de temperatura
#define ONE_WIRE_BUS 2 //Librería del sensor de temperatura

OneWire oneWire(ONE_WIRE_BUS); //habilitación de bus del sensor
DallasTemperature sensors(&oneWire);
RF24 radio(9, 10); //pines de habilitación para NRF21101
const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };
void setup(void)
{
  pinMode(10, OUTPUT); //pin en modo salida de los módulos NRF
  Serial.begin(9600); //velocidad de transmisión serial
  radio.begin(); //velocidad de modulo
  radio.setRetries(15, 15); //Máximo de reintentos
  radio.startListening(); //inicio para escuchar canal
  radio.openWritingPipe(pipes[1]); //canal de escritura
  radio.openReadingPipe(1, pipes[0]); //canal de lectura
}

void loop(void)
{
  radio.stopListening(); // Paramos la escucha para poder hablar
  unsigned long time = millis();
  Serial.print("TIEMPO DE PROCESAMIENTO ");
  Serial.println(time); //función para medir el tiempo de
  //procesamiento

  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);

  Serial.print("Enviando");
  Serial.println(temperature);

  delay(200);
  radio.write( &temperature, sizeof(float) ); //Escritura para envío
  bool ok = radio.write( &temperature, sizeof(float) );
  if (ok) //Evento de recepción y falla
    Serial.println("Dato correctamente recibido");
  else
    Serial.println("Dato no recibido");
  radio.startListening(); //Volvemos a la escucha
}

```

Anexo 9. Código nodo central Módulo Xbee

```
void setup() {  
  Serial.begin(9600);           //se establece la velocidad de comunicación de los nodo  
                                sensor  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  if (Serial.available() > 0)   //se limpia los caracteres de llegada para que no exista  
                                interferencias  
  {  
    char dato = Serial.read();  
    if (dato == 1)  
    {  
      Serial.write(Serial.read()); //al recibir un indicar este procede a escribir el dato recibido  
    }  
  }  
}
```

Anexo 10. Código nodo estación Módulo Xbee

```

#include <DallasTemperature.h> //Librería del sensor de temperatura
#define ONE_WIRE_BUS 2 //Librería del sensor de temperatura

OneWire oneWire(ONE_WIRE_BUS); //habilitación de bus del sensor
DallasTemperature sensors(&oneWire);

void setup(void)
{
  pinMode(10, OUTPUT); //pin en modo salida de los módulos NRF
  Serial.begin(9600); //velocidad de trasmisión serial
  sensors.begin();
  radio.begin();
  radio.openWritingPipe(pipe);
}

void loop(void)
{
  sensors.requestTemperatures(); //se realiza la captura del dato de temperatura
  TemperaturaA = sensors.getTempCByIndex(0); //se le añade a una variable

  if (TemperaturaA > 0) //condición para el envío de datos
  {

    Serial.print(1);
    Serial.print("Nodo 3:");
    Serial.print (TemperaturaA); //se imprime el dato recolectado para recibir
    //en el nodo central

    Serial.println (" C");
    delay (800);

  }
}

```

Anexo 11. Código nodo central Módulo WiFi

```

#include <ESP8266WiFi.h> //librería para programar directamente en el módulo
                          //WiFi
int temp;                //variable donde se almacena valor temperatura
int nodo;                //variable donde se almacena el número del nodo

//Conexión WiFi
const char* ssid = "Kirito"; //nombre del AP
const char* password = "Kirito1993"; //contraseña para que se conecten otros dispositivos

const char* ssid1 = "FLIA.LOZA"; //ssid de la red que nos conectaremos en el modo
                                  //estación para subir los datos a Internet
const char* password1 = "030793JP"; //contraseña del AP al que nos conectaremos para
                                      //subir los datos a internet

const char* server1 = "api.thingspeak.com"; //servidor de la plataforma IoT en Internet
String apiKey = "4VDEGO8TI25BZZRY"; //apiKey que nos da thinspeak

WiFiServer server(23); //puerto que utilizaremos para comunicarnos con los
                       //nodos sensores

int i = 0; //contador de veces que lee el dato del nodo 1
int temp1; //almacenara la temperatura del nodo 1
int j = 0; //contador de veces que lee el dato del nodo 2
int temp2; //almacenara la temperatura del nodo 1
int k = 0; //contador de veces que lee el dato del nodo 1
int temp3; //almacenara la temperatura del nodo 1

WiFiClient cliente2;

void setup()
{
  Serial.begin(74880); //iniciamos el puerto serial para ver los datos en la
                       //PC

  delay(10);

  WiFi.mode(WIFI_AP_STA); //indicamos que el módulo trabajará como AP y
                            //estación
  WiFi.softAP(ssid, password); //inicializamos el modo AP con el ssid y password

  IPAddress myIP = WiFi.softAPIP(); //Obtenemos la IP del modo AP
  Serial.print("AP IP address: "); //impresión del dialogo por el puerto seial

```

```

Serial.println(myIP); //impresión por el puerto serial de la
                      Ip del AP
server.begin();      //iniciamos el servidor
}
void loop()
{
  WiFiClient client = server.available(); //creamos nuestro cliente y comprobamos
  if (client)        //si no hay un cliente sigue
    return;

  String req = client.readStringUntil('\r'); //lee el mensaje
  temp = atoi(&req[0]); //extraemos el valor número 0 del vector request
  nodo = atoi(&req[5]); //extraemos el valor número 5 del vector request
  Serial.println("Nuevo cliente");
  Serial.print(temp); //imprimimos la temperatura separados de una coma
                      con el id del nodo que envía el mensaje

  Serial.print(" , ");
  Serial.println(nodo);

  if (nodo == 220882)
  {
    i++;
    if (i == 1)
      temp1 = temp;
  }

  if (nodo == 507040)
  {
    j++;
    if (j == 1)
      temp2 = temp;
  }

  if (nodo == 16299647)
  {
    k++;
    if (k == 1)
      temp3 = temp;
  }

  client.flush();

  client.print("Recibido \r");
  delay(100);

```

```

if (temp1 && temp2 && temp3 > 0)
{
  WiFi.begin(ssid1, password1);           //código para que se conecte a un AP
  cliente2.connect(server1, 80);         //para conectarse a la paltforma

  String postStr = apiKey;
  postStr += "&field1=";
  postStr += String(temp1);
  postStr += "&field2=";
  postStr += String(temp2);
  postStr += "&field3=";
  postStr += String(temp3);
  postStr += "\r\n\r\n";

  cliente2.print("POST /update HTTP/1.1\n");
  cliente2.print("Host: api.thingspeak.com\n");
  cliente2.print("Connection: close\n");
  cliente2.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
  cliente2.print("Content-Length: ");
  cliente2.print(postStr.length());
  cliente2.print("\n\n");

  cliente2.print(postStr);

  cliente2.stop();
  ESP.reset();
  delay(5000);                          //tiempo para cargar en lapsos de 15seg
}
}

```

Anexo 12. Código nodo central Módulo Wifi

```

#include <ESP8266WiFi.h> //librerías del módulo ESP8266
#include <WiFiClient.h> //librerías del módulo ESP8266
#include <OneWire.h> //librerías del sensor de temperatura
#include <DallasTemperature.h> //librerías del sensor de temperatura

OneWire ds(2); //pin de conexión del sensor de temperatura
DallasTemperature sensors (&ds);
const char* ssid = "Kirito"; //nombre del ssid del AP al cual nos vamos
const char* password = "Kirito1993"; //contraseña del AP

void setup() {

Serial.begin(74880); //iniciamos el puerto serial para ver los datos
//en la PC

delay(10);
WiFi.mode(WIFI_STA); //definimos en que trabaja el nodo
delay(100);

sensors.begin();
WiFi.begin(ssid, password); //inicializamos la conexión desde esta
//estación

while(WiFi.status() != WL_CONNECTED) //Espera por una conexión wifi
{
delay(500);
}
Serial.print(WiFi.localIP()); //Imprimir la IP
}

void loop() {

int celsius; //variable para obtener el valor de la
//temperatura

sensors.requestTemperatures(); //Solicita las temperaturas
celsius += sensors.getTempCByIndex(0); //Añade el valor de la temperatura
char request[20];
sprintf(request, "%d , %d\r", celsius, ESP.getChipId());
const int port=23; //Puerto por donde enviaremos el mensaje
const char* host= "192.168.4.1"; //ip o dns del seridor
WiFiClient Cclient; //Creamos nuestro cliente
Cclient.connect(host,port); //Conectamos nuestro cliente a la i del
//servidor

Cclient.print(request); //Enviamos nuestro mensaje al servidor
//Serial.println(request);

```

```
String req= Cclient.readStringUntil('\r');           //Lee el mensaje
if(Cclient.readStringUntil('\r'))
{
Serial.println(req);
//ESP.restart();
//Cclient.stop();
WiFi.disconnect();

if(Cclient.connected() !=1 || WiFi.status() != WL_CONNECTED)
{
ESP.reset();
}
}
}
```

Anexo 13. Código nodo central Módulo Bluetooth

```
#include <SoftwareSerial.h> // Incluimos la librería SoftwareSerial
SoftwareSerial BT(10,11); // Definimos los pines RX y TX del Arduino conectados al
Bluetooth
```

```
void setup()
{
  BT.begin(9600); // Inicializamos el puerto serie BT (Para Modo AT 2)
  Serial.begin(9600); // Inicializamos el puerto serie
}

void loop()
{
  if(BT.available()) // Si llega un dato por el puerto BT se envía al monitor serial
  {
    Serial.write(BT.read());
  }

  if(Serial.available()) // Si llega un dato por el monitor serial se envía al puerto BT
  {
    BT.write(Serial.read());
  }
}
```