



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN
División de Estudios Profesionales

Trabajo de Titulación
Opción TI. Tesis

Proyecto: “Red de comunicación por tecnología de radiofrecuencia entre un VANT y una computadora”

ALUMNO(S):	Eduardo Iván Rodríguez Vitela
No. CONTROL:	12480496
CARRERA:	Ingeniería en Sistemas Computacionales
ASESOR DE RESIDENCIA:	M.C Isidro Hernandez Vega
REVISORES:	

AGRADECIMIENTOS.

Primeramente le agradezco a mis padres por apoyarme a lo largo de mi carrera, así como también a mis maestros que tuve a lo largo de la carrera.

RESUMEN.

A lo largo de este documento se podrá apreciara como hacer un sistema de monitoreo de contaminantes y una aplicación web para desplegar la información obtenida de los escaneos. Para poder llevar acabo este proyecto se tendrá que realizar la instalación de un servidor LAMP, donde se almacenaran los datos de todos los sensores, posteriormente crear la interface web para que proceda a hacer la lectura de los registros de la base y muestre la información en forma de gráficas. Para la obtención de los datos se empleara una microcontroladora (raspberry) y atreves de sensores analógicos por el SPI realizara las lecturas con Python, cabe mencionar que la raspberry 3 no tiene puertos analógicos por lo cual se empleara un circuito MPC3008 el cual cumplía con la función de transformar los datos analógicos a digitales para su posterior interpretación por la microcontroladora, realizara a la vez por un módulo xbee la información por el protocolo Zigbee y mediante una red de comunicación inalámbrica de radio frecuencia se leerá la información desde el servidor y se procederá a insertar en la base de la información censada. De esta forma la información desplegada en la gráfica, atreves del navegador, será información en tiempo real.

Contenido

Descripción de la empresa	2
Misión:	2
Visión:	2
PLANTEAMIENTO DEL PROBLEMA A RESOLVER.....	3
OBJETIVOS GENERALES Y ESPECIFICOS.	3
JUSTIFICACIÓN.....	3
ALCANCES Y LIMITACIONES.....	4
PRODUCTO – BENEFICIO	4
MARCO TEORICO.....	5
Tecnología de radiofrecuencia.....	5
Redes de sensores inalámbrico WSN.....	10
Alianza ZigBee	11
Qué es XBee.	21
X-CTU	29
Convertor analógico – digital (A/D).	30
Sistemas embebidos.	31
Rasspberry Pi.....	32
Sistema operativo.	33
Servidor LAMP.	35
Linux.....	35
Apache (Servidor HTTP).	36
MySQL.....	37
PHP.....	38
Python.....	39
PROCEDIMIENTO Y DESCRIPCIÓN DE ACTIVIDADES.....	40
RESULTADOS.	41
EXPRIMENTACION.....	50
CONCLUSIÓN.....	53
COMPETENCIAS APLICADAS Y/O DESARROLLADAS.	54
FUENTES DE INFORMACIÓN.....	54
ANEXOS.	56
Códigos.....	56
Oficios.	77

Descripción de la empresa.

Instituto Tecnológico de Nuevo León

Dirección Av. Eloy Cavazos #2001, col. Tolteca Guadalupe. Nuevo León, México.

Cp.67170.

Teléfonos: 81570500 fax: 81570500 ext.306.

Área de posgrado del Instituto Tecnológico de Nuevo León

M.C Isidro Hernandez Vega

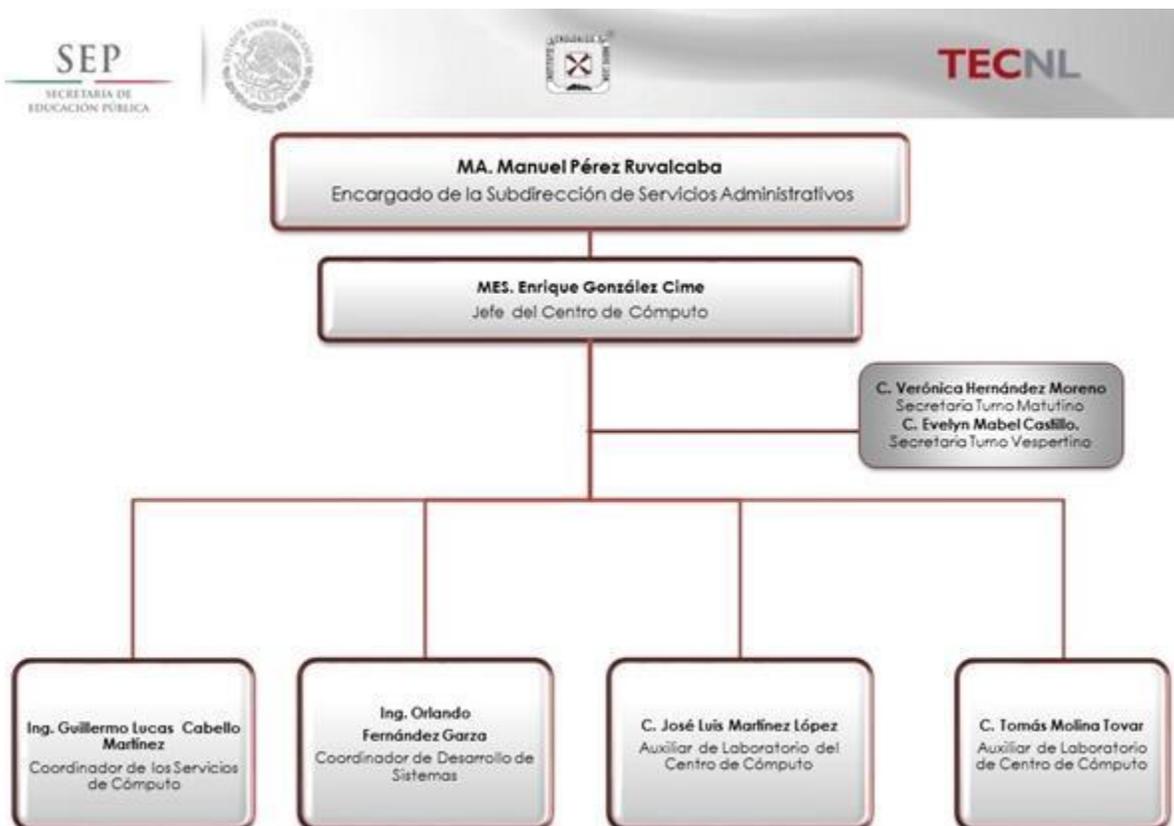
Edificio # 19

Misión:

Ofrecer una amplia variedad de servicios satisfaciendo de esta manera las múltiples necesidades de información y automatización de funciones administrativas y académicas formando así, una base firme para la concesión de los objetivos generales de nuestra institución.

Visión:

Consolidar a nuestro instituto como un organismo líder y pionero en los avances tecnológicos acordes a las necesidades de cambio que nos exige la globalización manteniendo altos niveles de calidad, solidez y eficacia en los diversos planes de acción tanto internos como externos orientados a dar servicio a la comunidad de nuestro instituto.



PLANTEAMIENTO DEL PROBLEMA A RESOLVER.

Este proyecto nace del complemento del trabajo de investigación, desarrollado en la división de estudios de posgrado, contaminantes CRISA, dado a que este proyecto necesitaba la particularidad de poder establecer una red de comunicación entre los diferentes Vehículos Aéreos No Tripulados (VANT) y la terminal en tierra, de manera que se pudieran censar los niveles de contaminantes del medio ambiente, estos estando a una determinada altura o para su fácil escaneo de áreas un poco accesibles, además de entablar la comunicación es necesario el análisis de los datos y la interpretación de estos, punto del cual en el proyecto anterior se necesitaba poder interpretar los datos de manera correcta, para poder crear información que sea de utilidad para un usuario a partir de una interfaz, de manera que sea muy fácil el tener en cuenta los datos que captan los VANT y verlos reflejados a través de gráficas o esquemas. Es a partir de esto que se establece que la utilización de una red de comunicación por medio de radiofrecuencia sería una idea viable dado al alcance que tienen estos dispositivos.

OBJETIVOS GENERALES Y ESPECIFICOS.

Este proyecto de investigación se centra en la implementación de una red de comunicación por medio de la tecnología de radiofrecuencia; basada, en este caso, en la configuración y uso de los módulos xbee que utilizan el protocolo de comunicación Zigbee. Dicha red establecerá conexión entre los distintos dispositivos hasta poder enviar la información a la terminal en tierra. Dado a la distancia que existe entre el VANT y la terminal, y la necesidad de poder tener un censo en cualquier área se implementará la red de comunicación con los módulos xbee.

Los objetivos específicos son los siguientes:

- Desarrollo e implementación de una red de comunicación de radiofrecuencia mediante el protocolo Zigbee.
- Comunicación entre el VANT y la terminal mediante la red.
- Desarrollo de la obtención de datos para enviar por la red.

JUSTIFICACIÓN

Este proyecto es de mucha utilidad ya que actualmente se utilizan este tipo de tecnologías para la detección y control del medio ambiente, grandes instituciones como lo son la UNEP (Programa de las Naciones Unidas para el Medio Ambiente) y la NASA utilizan este tipo de tecnologías para el censo y monitorear la cantidad de gases que son desprendidos por el volcán Turrialba en San José, Costa Rica, esto es una muestra de lo útil que son este tipo de proyectos y de los grandes resultados que nos dejan para el desarrollo y cuidado del medio ambiente. Tomando estos puntos en cuenta, es muy claro que enfocados en el área ambiental, el presente proyecto desarrollado, es

de mucha utilidad y más aún porque no es una propuesta cara de implementación, en comparación con los productos existentes en el mercado especializados en ese tipo de trabajos.

Claro está que de dicho proyecto desarrollado es una opción viable pero no estará al nivel de los proyectos desarrollados por la NASA, u organizaciones de renombre similar, más aun es útil para el conocimiento de los valores de contaminación en algunas áreas, y los niveles de interpretación de la información serán mucho más claros o inconsistentes dependiendo de los diversos componentes electrónicos, dígitos sensores, que se utilizan en el desarrollo del proyecto.

Algunos detalles que representa la justificación de este proyecto es lo referente a el área de innovación, dado a que como se mencionó es una propuesta buena dado a que los actuales fabricantes de dispositivos similares no ofrecen algo tal cual como el presente proyecto. Lo cual nos permite elegir los componentes que se ajusten a las necesidades del proyecto y en presupuesto.

ALCANCES Y LIMITACIONES.

Como se ha mencionado dado, hasta ahora, el proyecto se enfocará en el desarrollo de los componentes para la detección de los niveles de gases en el medio ambiente, los cuales serán principalmente nitrógeno, dióxido de carbono, monóxido de carbono e hidrógeno, la interpretación y procesamiento de estos datos en una tarjeta de código embebido y enviados a través de la red de comunicación para la interpretación en un servidor central.

Algunos de las posibles limitaciones que se podrían presentar durante la realización del proyecto podrían ser:

- La obtención de los materiales para poder trabajar con ellos.
- La incompatibilidad entre las herramientas para el desarrollo del software necesario.
- La experiencia de uso de los diversos lenguajes de programación y la integración de estos.

PRODUCTO – BENEFICIO

Durante el proceso de desarrollo, aplicación e integración de todos los recursos se busca poder dar continuidad y completar el proyecto de los contaminantes, dado a que este proyecto ya tenía un desarrollo en cuanto a el desarrollo del dron y el uso de la telemetría para la programación de los planes de vuelo autónomos, con la integración del actual proyecto las funciones anteriores se ampliarán.

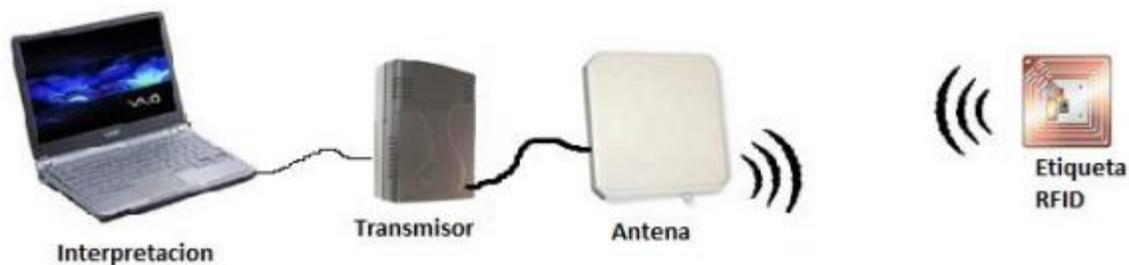
De la misma manera se busca dar a conocer que el ITNL se interesa en el desarrollo de proyectos de innovación tecnológica para el control del medio ambiente, temas que han tomado un gran auge por las acciones del calentamiento global. Así mismo es una gran área de oportunidad para aprender a integrar los conocimientos desarrollados, a lo largo de la carrera de sistemas, con otras áreas como lo son la electrónica y tratar de darles a estos conocimientos un enfoque de resolución de problemas, no solo de la industria, tales como lo son los ambientales.

MARCO TEORICO.

Tecnología de radiofrecuencia.

RFID significa Radio Frequency Identification, en español Identificación por Radio Frecuencia. Se trata de un sistema de almacenamiento y recuperación de datos remotos que utiliza dispositivos llamados etiquetas. Urbina (2011, p.25).

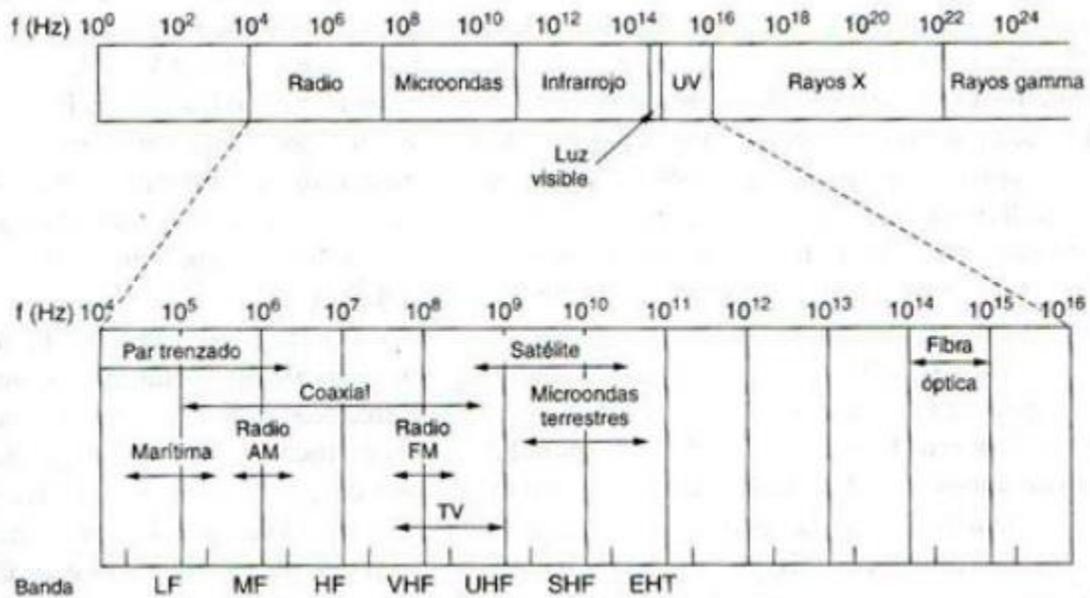
Urbina (2011) nos menciona que la radiofrecuencia suele ser similar a los códigos de barras ya que cuentan con etiquetas o identificadores para poder enviar información a otros dispositivos. En este tipo de tecnología se suele hacer uso de 3 partes, una antena un transmisor y un receptor, la cual se encarga de leer los datos contenidos dentro de la etiqueta y obtener la información de manera inalámbrica. En la siguiente imagen se puede apreciar con se relacional los dispositivos mencionados anteriormente. Donde una terminal intenta obtener información, de otros dispositivos, haciendo lecturas mediante un transmisor que tendrá conectado una antena, está variando según el tipo de comunicación a entablar con los otros dispositivos y mandando la señal a la etiqueta de RFID la cual retorna la información a la terminal para su posterior interpretación.



Cabe destacar que la antena es la que envía la señal, la cual no es más que una radiación electromagnética, que viaja por el aire hasta toparse con la etiqueta RFID la cual al entrar dentro de campo electromagnético de la señal enviada por la antena se activara y realizara la función que tenga activada. Este tipo de radiación electromagnética puede variar dependiendo de la banda de frecuencia. Es muy sabido que todos los aparatos electrónicos que trabajan a distancias son manejados a través de ondas mandadas por aire. Y estas bandas de frecuencia pueden ser las siguientes:

	Rango de operación	Rango de lectura	Consumo de energía	Rango de transferencia de datos	Tamaño de longitud de onda
Baja LF	30 y 300 KHz	10 cm	Bajo	Alto	10 y 1 Km
Media MF	300KHz y 3 MHz	20 cm	Bajo	Alto	1000 y 100 mts
Alta HF	3 y 30 MHz	1 metro	Moderado	Alto	100 y 10 mts
Ultra Alta UHF	300 MHz y 3GHz	hasta 3 metros	Moderado	Rápido	1 metro y un decímetro
Súper Alta Frecuencia SHF	3 y 30GHz	Hasta 10 metros	Moderado	Súper rápido	1 decímetro y 1 centímetro

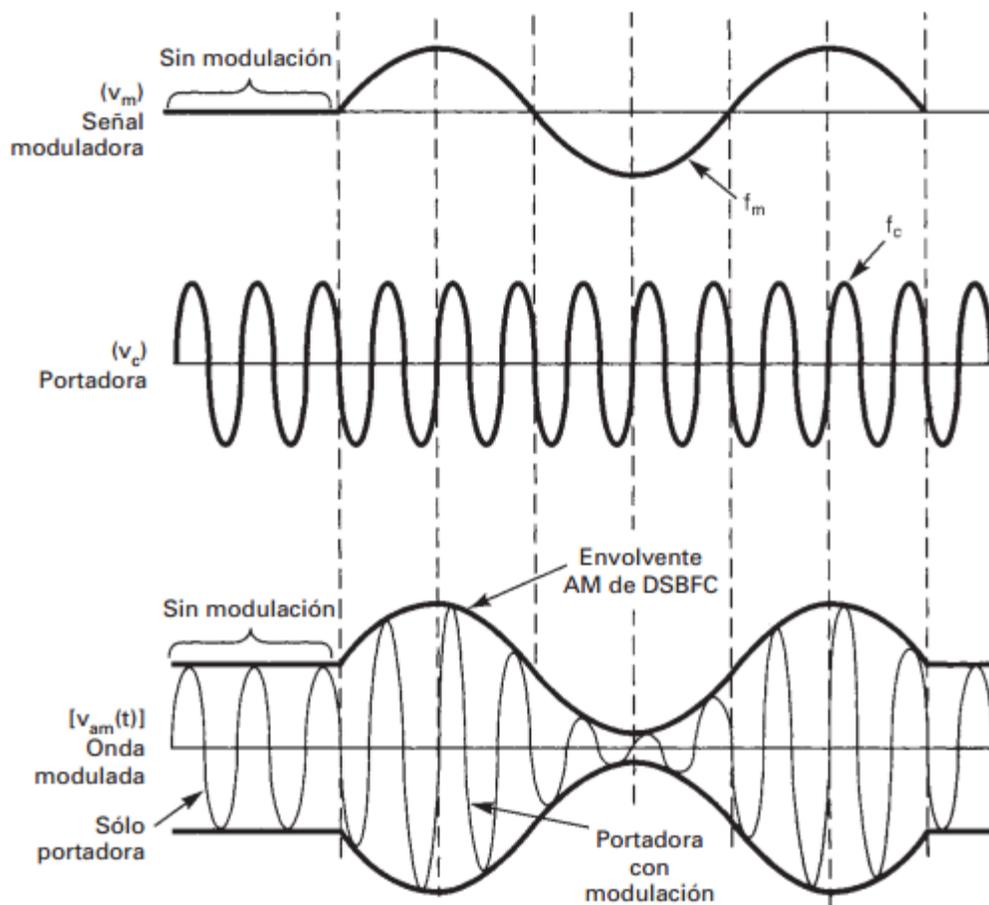
Y de estas, para nuestro caso de la tecnología de radiofrecuencia, se suelen usar las siguientes:



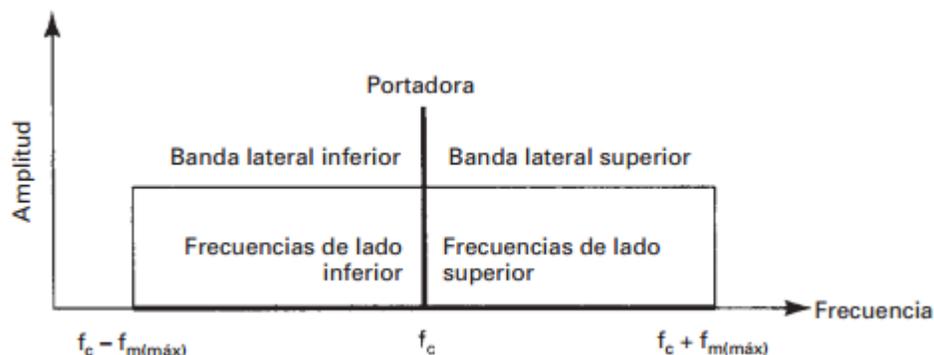
Los datos que se envían por radiofrecuencia pueden ser enviados por distintas formas, ya sea por el tipo de señal de onda amplitud modulada (AM) o modulación de frecuencia (FM).

Espectro de frecuencia y ancho de banda de AM.

Un modulador de AM es un dispositivo no lineal. En consecuencia, hay mezclado no lineal, y la envolvente de salida es una onda compleja formada por un voltaje de cd, la frecuencia de la portadora y la suma ($f_c + f_m$) y diferencia ($f_c - f_m$) de las frecuencias, es decir, los productos cruzados. Las frecuencias de suma y diferencia están desplazadas respecto a la frecuencia de la portadora una cantidad igual a la frecuencia de la señal moduladora. Por consiguiente, un espectro de señal de AM contiene los componentes de frecuencia apartados f_m Hz a ambos lados de la portadora. Sin embargo, se debe hacer notar que la onda modulada no contiene un componente de frecuencia que sea igual a la de la señal moduladora. El efecto de la modulación es trasladar la señal moduladora en el dominio de la frecuencia, de modo que se refleje simétricamente respecto a la frecuencia de la portadora. (Tomasi, 2003, p.101)



La siguiente imagen muestra el espectro de frecuencias para una onda de AM. Este espectro se extiende desde $f_c - f_m(\text{max})$, siendo f_c la frecuencia de la portadora y $f_m(\text{max})$ la frecuencia máxima de la señal moduladora. La banda de frecuencia entre $f_c - f_m(\text{max})$ y f_c se llama banda lateral inferior (LSB, de lower sideband) y las frecuencias dentro de esta banda se llaman frecuencias de lado superior (USF, de upper side frequency).



Por consiguiente, el ancho de banda (B) de una onda DSBFC de AM es igual a la diferencia entre la frecuencia máxima de lado superior y la mínima del lado inferior, o también, igual a dos veces la frecuencia máxima de la señal modulante, es decir, $B = 2f_m(\text{max})$. Para propagación de ondas de radio, la portadora y todas las frecuencias dentro de las bandas laterales superior e inferior deben ser lo bastante elevadas como para poder propagarse lo suficiente a través de la atmósfera terrestre. (Tomasi, 2003, p.102)

Modulación angular o de frecuencia (FM).

La modulación angular se introdujo por primera vez en 1931, como alternativa a la modulación de amplitud. Se sugería que una onda con una modulación angular es menos susceptible al ruido que una onda de AM y, en consecuencia, podría mejorar el desempeño de las radiocomunicaciones. El mayo E.H. Armstrong (quien también desarrolló el receptor superheterodino) desarrolló el primer sistema de radio FM funcional en 1936, y en julio de 1939 comenzaron las primeras emisiones con programa regular, e Alpine, New Jersey. Hoy, la modulación angular se usa en formas extensa para radioemisión comercial, televisión y transmisión de sonido, radioteléfonos, radios celulares y sistemas de comunicaciones por microondas y satélites.

La modulación angular se produce siempre que se varía el ángulo de fase, Θ , de una onda senoidal con respecto al tiempo. Una onda con modulación angular se describe matemáticamente como:

$$m(t) = V_c \cos[\omega_c t + \theta(t)] \quad (6-1)$$

$m(t)$ = onda con modulación angular

V_c = amplitud máxima de portadora (volts)

ω_c = frecuencia de la portadora en radianes, es decir, velocidad angular, $2\pi f_c$, en radianes por segundo

$\theta(t)$ = desviación instantánea de fase (radianes)

La diferencia entre modulación de frecuencia y modulación de fase es, en esencia, cuál propiedad de la portadora se hace variar en forma directa como la señal moduladora: la frecuencia o la fase, y cuál se hace variar en forma indirecta. Siempre que se varía la frecuencia de una portadora también varía la fase, y viceversa. Por consiguiente, la FM y la PM deben estar presentes al mismo tiempo siempre que se hace cualquier forma de modulación angular. Si la frecuencia de la portadora se hace variar directamente de acuerdo con la señal moduladora se obtiene la FM. Si se varía la fase de la portadora de forma directa por la señal moduladora, resulta la PM. Por consiguiente, la FM directa es PM indirecta, y la PM directa es FM indirecta. Se pueden definir las modulaciones de frecuencia y fase como sigue.

Modulación directa de frecuencia (FM): Varía la frecuencia de una portadora de amplitud constante en proporción directa a la amplitud de la señal moduladora, con una rapidez igual a la frecuencia de la señal moduladora.

Modulación directa de fase (PM): Varía la fase de una portadora de amplitud constante en proporción directa a la amplitud de la señal moduladora, con una rapidez igual a la frecuencia de la señal moduladora. (Tomasi, 2003, p.229)

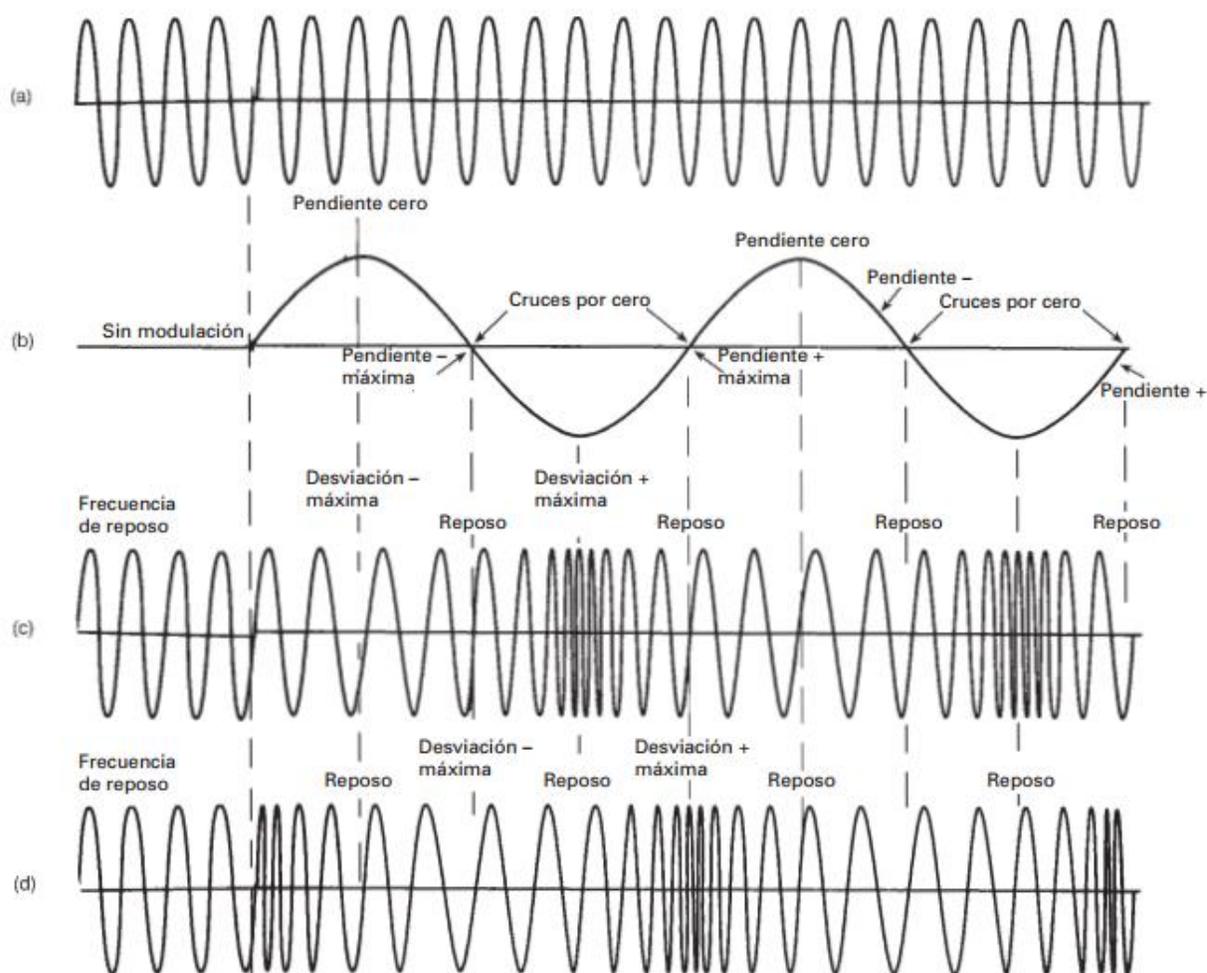
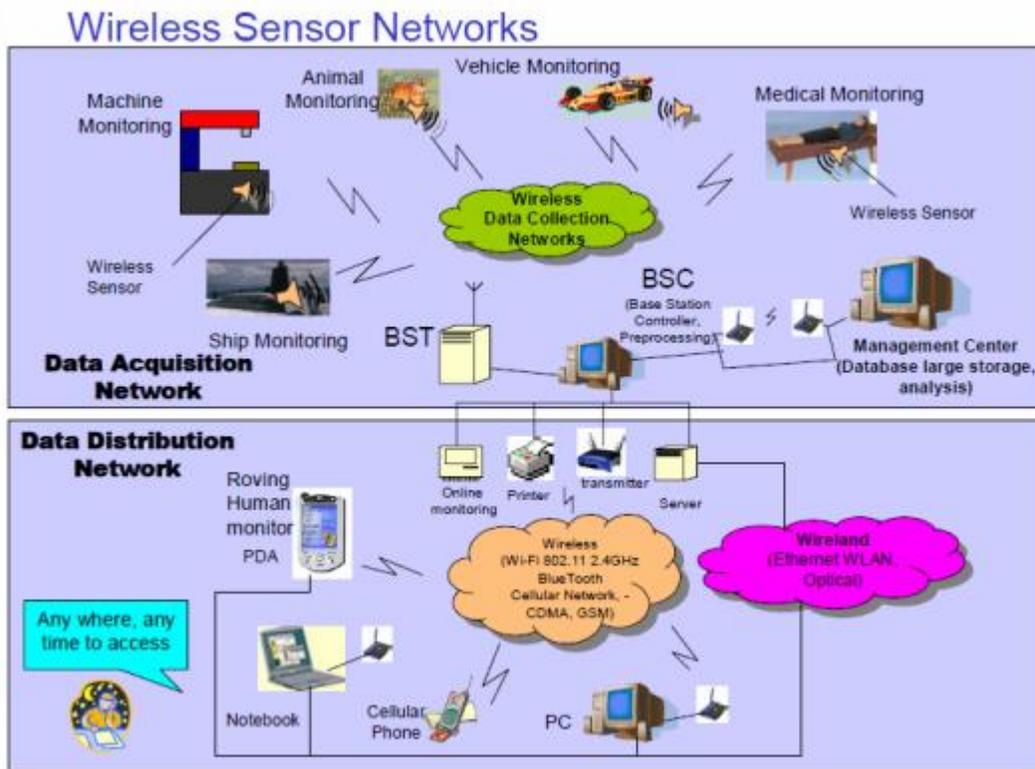


FIGURA 6-3 Modulación de fase y de frecuencia de una onda portadora senoidal por una señal de onda senoidal: (a) portadora no modulada; (b) señal moduladora; (c) onda con frecuencia modulada; (d) onda con fase modulada

La figura. 6-3 ilustra la modulación de frecuencia y de fase de una portadora senoidal, mediante una señal moduladora de frecuencia única. Se ve que las formas de onda de FM y de PM son idénticas, a excepción de su relación temporal (fase). Así, es imposible distinguir una forma de onda de FM de una de PM, sin conocer las características dinámicas de señal moduladora. Con FM, la desviación máxima de frecuencia (cambio de frecuencia de la portadora) se efectúa durante los picos máximos positivos y negativos de la señal moduladora; es decir, la desviación de frecuencia es proporcional a la amplitud de la señal moduladora. Con la PM, la desviación máxima de frecuencia se efectúa durante los cruces de la señal moduladora por cero; es decir, la desviación de frecuencias es proporcional a la pendiente de la rapidez con que cambia la frecuencia es igual a la frecuencia de la señal moduladora. (Tomasi, 2003, p.223-224)

Redes de sensores inalámbrico WSN

Las redes de sensores inalámbricas (Wireless Sensor Networks, o simplemente WSN), constituyen actualmente una herramienta tecnológica muy valiosa para recolectar información del entorno. Estas redes están formadas por plataformas de hardware con recursos limitados de memoria, cómputo y energía, motivo por el cual las WSN trabajan con protocolos especialmente diseñados para funcionar sobre estas plataformas (IEEE 802.15.4, ZigBee, etc.). Iacono, Godoy, Marianetti, García, Párraga (2012, p.1).



Mediante este tipo de redes podemos realizar la obtención de datos de distintos dispositivos conectados a estos mismos tipos de redes, de esta manera, dado a su portabilidad, es posible obtener datos en cualquier zona geográfica; para el caso del desarrollado de actual proyecto se aplicara sobre una red para poder obtener los valores de contaminación de cualquier área sobre la que se esté aplicando la red, esto a través de un protocolo de Zigbee que será descrito más adelante.

El principal inconveniente de este tipo de redes es el uso de la batería, dado a que suelen emplearse en lugares de una gran amplitud, es decir con distancias considerables entre cada dispositivo, lo cual demandara una cantidad considerable de energía más aun sumada a esta la cantidad de energía gastada por los distintos dispositivos usados para realizar las obtenciones de valores.

Este tipo de redes se componen de nodos, los cuales a su vez están constituidos por alguna microcontroladora; que realizara la lectura a través de sensores, la cual deberá de tener algún tipo de conexión hacia la terminal.

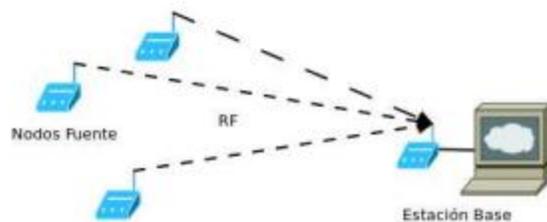


Figura 1.- Elementos de una WSN.

En la imagen anterior podemos apreciar los dispositivos de la red, los cuales la terminal o estación base tiene un módulo de comunicación hacia los demás dispositivos, esta tiene la función de hacer la lectura de los demás módulos de la red y una vez con esa información se puede ser manejada a la conveniencia. Además de los nodos anteriormente mencionados se suele hacer uso de los routers, los cuales tiene la función de repetir la señal, en pocas palabras amplían la red y permiten a los dispositivos más lejanos al servidor poder llegar a mandar la información requerida; de manera que los nodos podrán escoger, automáticamente, la ruta más cercana para el envío de los datos.

Cabe mencionar que los nodos utilizados están limitados en cuanto a recursos de procesamiento y memoria, por lo cual no puede utilizar los protocolos de comunicación tradicionales, como lo son TCP/IP, de tal manera que suele utilizar protocolos distintos (dependiendo de los tipos de nodos) y esto ayuda a reducir un poco de la energía.

La mayoría de los fabricantes de estos tipos de dispositivos o módulos suele construir su hardware bajo el estándar IEEE 802.15.4. Los cuales permiten la integración de algunas topologías de red como un punto a punto, estrella, o tipo Mesh.

Alianza ZigBee

La alianza ZigBee es un consorcio no lucrativo de más de 70 compañías, incluyendo Invensys, Mitsubishi Electric y Motorola. Estas compañías trabajan juntas para crear un estándar inalámbrico de bajo costo y de bajo consumo de energía.

Las redes ZigBee comenzaron a ser concebidas por el año 1998, cuando muchos ingenieros al crear WiFi y Bluetooth vieron que estos dos estándares iban a ser inadecuados para muchos usos. En particular, el afán de la alianza Zigbee fue diseñar redes ad hoc de auto organización para radios digitales y además, desarrollar perfiles de uso, programas de certificación, insignias y estrategias de comercialización.

El estándar IEEE 802.15.4n en el que se basa ZigBee, fue terminado en mayo del 2003, pero para ese verano, Philips, el promotor más importante cesó su inversión. Sin embargo, el impulso en las investigaciones sigue y Philips pasó a ser un miembro promotor de la junta directiva de la alianza ZigBee.

La alianza ZigBee tiene dos clases de socios. Los promotores son los jefes de la alianza que representan un cruce de proveedores de semiconductores, de software y de sistemas de industria, como es el caso de Freescale u promotor subsidiado de Motorola.



Los participantes tienen un interés menos comprometido en la alianza ZigBee. Pueden asistir a las reuniones de la alianza y tener acceso a todas las especificaciones preliminares. Igual que los promotores, los participantes de la alianza ZigBee representan una gran variedad de empresas, tanto fabricantes de semiconductores como fabricantes de equipos. Origen del nombre ZigBee. Se le puso el nombre de ZigBee por una especie de abejas, en las cuales es sabido que trabajan unidos por el bien de su colonia, de manera que siempre tienen que estar en constante comunicación, estas para comunicarse suele hacer una especie de danza en forma de zig-zag de un lado para otro. Es de aquí que parte el termino Zig, que utilizan las abejas para comunicarse, y bee por abeja en inglés.

Estándar ZigBee.

El término ZigBee describe un protocolo inalámbrico normalizado para la conexión de una red de área personal inalámbrico o WPAN. ZigBee es diferente de los otros estándares inalámbricos, ha

sido diseñado para soportar un diverso mercado de aplicaciones con una conectividad más sofisticada que los anteriores sistemas inalámbricos.

Como se ha mencionado anteriormente la mayoría de los fabricantes de hardware para nodos de redes WSN utilizan el estándar IEEE 802.15.4, para el caso de la alianza de Zigbee contemplan las capas PHY (Physical Layer) y MAC (Application Layer), que especifica el estándar, y la alianza Zigbee agrega y especifica las capas superiores de red y seguridad.

Esta alianza promueve el estándar para poder crear una unificación entre distintos fabricantes, tratando de minimizar los riesgos de compatibilidad.

Zigbee desde el punto hardware y técnico.

Si pensamos en el módulo de Zigbee desde su punto de hardware no podremos dar cuenta que no pose, en realidad, muchas partes analógicas y trata de utilizar siempre que pueda circuitos digitales. Esto con el fin de optimizar lo más posible su funcionamiento, así como el desgaste de energía que requiere para su uso.

Desde el punto de vista técnico el estándar de Zigbee utiliza protocolos para el envío de datos de baja velocidad, de corto alcance, y estos regularmente suelen operar a 868 MHz, 915 MHz y 2.4GHz.



	Wi-Fi	Bluetooth	ZigBee
Velocidad	<50 Mbps	1 Mbps	<250 kbps
Núm. nodos	32	8	255 / 65535
Duración batería	Horas	Días	Años
Consumo transm.	400 ma	40 ma	30 ma
Consumo reposo	20 ma	0.2 ma	3 µa
Precio	Caro	Medio	Barato
Configuración	Compleja	Compleja	Simple
Aplicaciones	Internet en edificios	Informática y móviles	Domótica y monitorización

Por lo tanto, podemos catalogar a Zigbee como un protocolo:

- Basado en el estándar 802.15.4.
- Con aplicaciones WPAN (Wireless Personal Area Network).
- Con gran soporte desde la industria: Honeywell, Mitsubishi Electric, Motorola, Puhilips, Samsung, etc.
- Bajo coste.

- Bajo consumo.
- Instalación fácil y barata.
- Redes flexibles y extensibles.

(Fortuño, 2012, p.18)

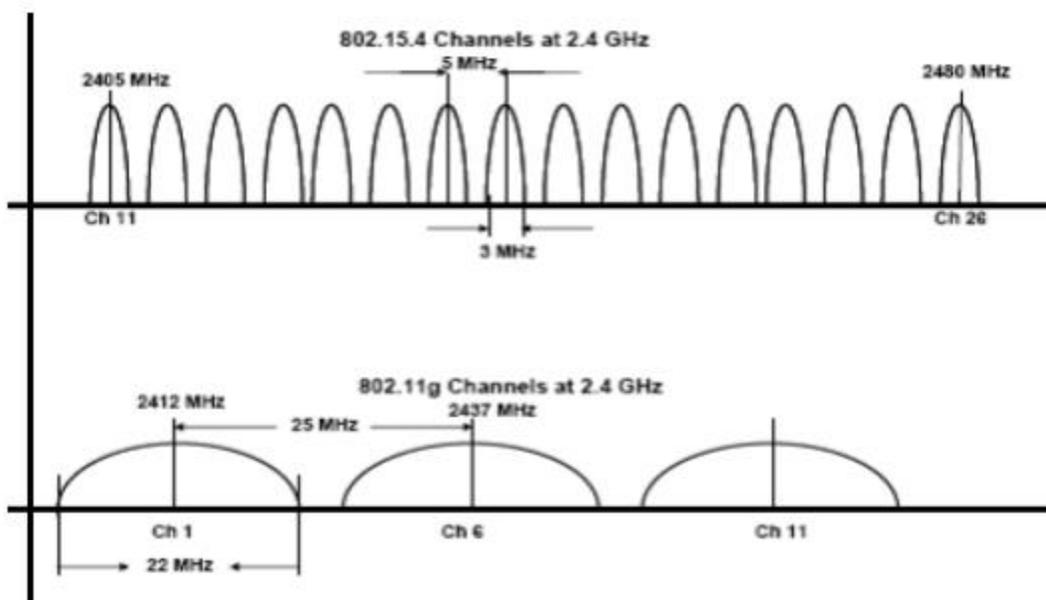
Frecuencia de trabajo de Zigbee.

Como se ha mencionado, anteriormente, Zigbee tiene tres frecuencias distintas sobre las que opera, las cuales son:

- 868 MHz
- 915 MHz
- 2.4GHz

Fortuño (2013, p.19) dice que “La banda de 2.4 GHz es usada en el mundo entero mientras que las de 868 MHz y 915 MHz se usan en Europa y América del Norte respectivamente.”

Por lo cual se tiene la libertad de trabajar con los Zigbee en la frecuencia que gustemos pero aun así debemos de tomar en cuenta los factores que afectaran la banda que utilizemos para llevar a cabo las comunicaciones, dado a que la banda a 2.4 GHz puede verse interferida por el WiFi o Bluetooth o viceversa.

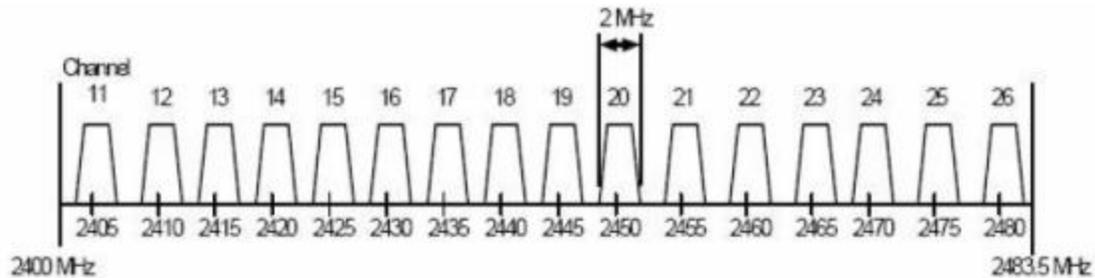


En la imagen anterior podemos apreciar la similitud en la forma y amplitud de las ondas de una señal de Zigbee (en la parte superior) y una de WiFi (en la parte inferior) ambas operando a 2.4

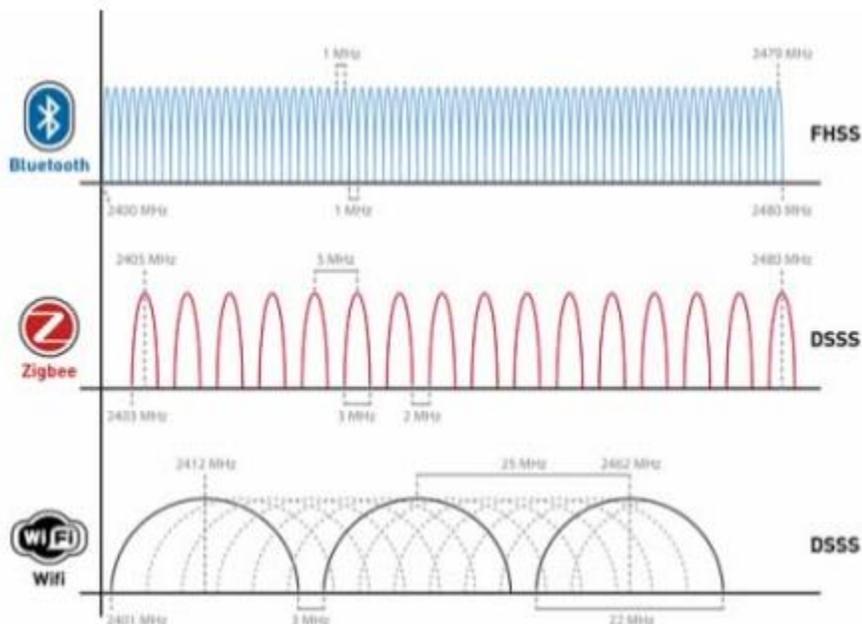
MHz, es por esto que se debe de tomar en cuenta su uso si se emplea con otras tecnologías, ya antes mencionadas. Aunque actualmente es más usada la banda a 2.4 GHz dado a que permite un mucho mayor ancho de banda y mayor cantidad de canales.

Para ampliar más el tema podemos citar a Fortuño que dice:

El estándar IEEE 802.15.4 divide la banda de 2.4 GHz e 16 canales de 2 MHz de ancho y 5MHz de separación, tal como se muestra en la figura.



Por otro lado, Bluetooth usa 79 canales de 1 MHz cada uno, de forma que tampoco hay solapamiento entre sus canales y la señal que modulan en cada uno de ellos.



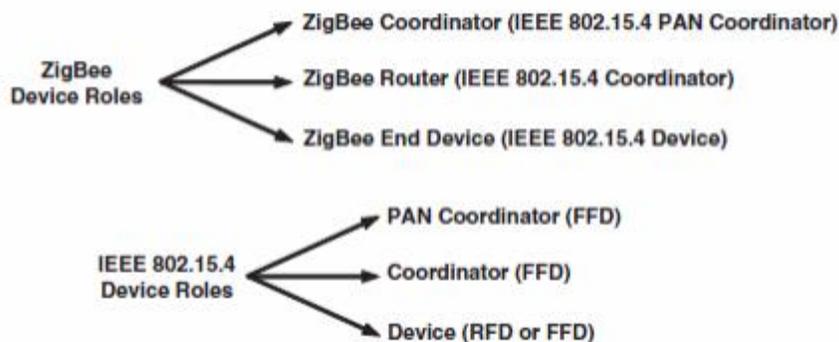
Una vía para evitar interferencias puede ser la de seleccionar canales que no se superpongan. En concreto los canales 15, 16, 21 y 22 del estándar IEEE 802.15.4 no se superponen con los canales 1, 7 y 13 del estándar 802.11b. Por lo tanto, es en estos canales dónde tendríamos la menor interferencia posible con los canales de WiFi. (Fortuño, 2012, p.18)

	Frequency (MHz)	Number of Channels	Modulation	Chip Rate (Kchip/s)	Bit Rate (Kb/s)	Symbol Rate (Ksymbol/s)	Spreading Method
	868-868.6	1	BPSK	300	20	20	Binary DSSS
	902-928	10	BPSK	600	40	40	Binary DSSS
Optional	868-868.6	1	ASK	400	250	12.5	20-bit PSSS
	902-928	10	ASK	1600	250	50	5-bit PSSS
Optional	868-868.6	1	O-QPSK	400	100	25	16-array orthogonal
	902-928	10	O-QPSK	1000	250	62.5	16-array orthogonal
	2400-2483.5	16	O-QPSK	2000	250	62.5	16-array orthogonal

En la tabla anterior podemos visualizar las características con las que se puede contar en las distintas bandas de frecuencias y en distintos tipos de modulación, claro está hay que tenerlas muy en cuenta al momento de elegir una para usar en un proyecto.

Tipos de dispositivo Zigbee.

La IEEE 802.15.4 en sus estándares habla acerca de tres roles que pueden tomar los dispositivos de una red, mientras que Zigbee les da otra analogía a las funciones de los dispositivos, lo podemos apreciar más claro en la siguiente imagen.



A la hora de configurar hay que tomar en cuenta los dispositivos que usaremos y que rol jugaran en el desarrollo de un proyecto, de la imagen anterior podemos observar y detallar los diferentes roles que usan los módulos de Zigbee.

Zigbee Coordinator.

Este es dispositivo, o configuración, mas importante su función es la de coordina los router y end device para poder obtener la información, básicamente es el que ira conectado a una terminal para recibir los datos de la red.

En este estado, el coordinador ZigBee debe permitir que otros dispositivos se unan a la red basándose en sus parámetros de configuración; como pudieran ser la duración de la incorporación del dispositivo a la red o el número máximo de elementos que se pueden unir.

El coordinador ZigBee debe responder a cualquier dispositivo u operaciones del servicio de descubrimiento de su propio dispositivo o de cualquier dispositivo que tenga asociado y que esté dormido. La aplicación del dispositivo debe asegurarse de que el número de entradas de enlace no excede de los indicados en los parámetros de configuración. Por tanto, el coordinador ZigBee tiene que soportar el control del proceso de incorporación a la red de cualquier dispositivo.

El coordinador tiene que mantener una lista de los dispositivos asociados y facilitar el soporte para elementos huérfanos, permitiendo que se vuelvan a unir a la red, permitiendo que los dispositivos se incorporen directamente en la red.

Zigbee Router.

Este, como su nombre lo dice, será el encargado del encaminamiento y enrutamiento de la información, es decir, dirigirá la información a su correspondiente destino, el coordinador, o en su defecto a otro router para llegar a alcanzar el coordinador.

La aplicación del dispositivo utiliza el ChannelList y sus parámetros de configuración para buscar o escanear los canales que se le indiquen. El resultado permite obtener la lista de red con los PAN activos en la red. Entonces se realizan varias peticiones de descubrimiento para obtener cuales son realmente los elementos que existen en la red y asociar los enlace en la capa de red. La aplicación del dispositivo compara el ChannelList con la NetworkList para seleccionar los PAN existentes que se deben unir. Una vez que el PAN al que unirse se ha identificado, la aplicación del dispositivo debe realizar una petición para asociar el PAN en el canal. Después debe chequear el estado de verificación de la asociación en el coordinador u otros routers seleccionados en ese PAN.

Si la red tiene la seguridad activada, el dispositivo tiene que esperar a que el centro de validación le proporcione la clave maestra y establecer con éste la clave de enlace. Una vez establecido espera a que el centro de validación de pase la clave de red. Ahora ya que está autenticado puede funcionar como un router de la red.

El router debe permitir que otros dispositivos se unan a la red basándose en los parámetros de configuración que tiene, como el número de elementos máximos o el tiempo en el que puede estar un elemento en la red. Cuando un dispositivo nuevo se une a la red, la aplicación del dispositivo debe ser informada. Cuando se haya admitido en el PAN, el router debe indicarle la confirmación de la conexión. Si la seguridad está habilitada, el dispositivo debe informar al centro de validación.

El router ZigBee debe responder a cualquier dispositivo descubierto o a operaciones del servicio de descubrimiento, tanto de su propio dispositivo como de cualquier otro asociado que pudiera

estar dormido. Si la seguridad está activada, el router debe utilizar la clave maestra para establecer los procedimientos para la gestión de la clave de enlace (Link Keys). El router debe soportar el establecimiento de una clave maestra con el dispositivo remoto y establecer entonces la clave de enlace. El router tiene que poder almacenar y eliminar las claves de enlace para destinos conocidos que requieran que la comunicación sea segura con lo que debe poder recibir las claves del centro de validación.

Zigbee End Device.

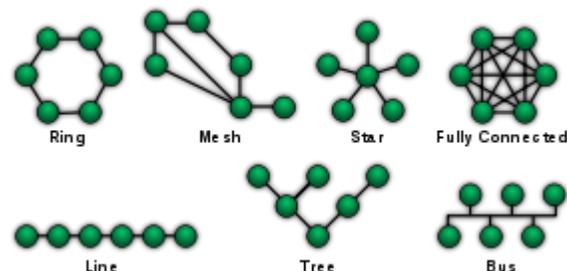
Es un elemento pasivo de la red, ya que no transmitirá información de manera autónoma; simplemente dispone de la funcionalidad mínima para ser capaz de responder a peticiones de dispositivos superiores, ya sea coordinador o router. Debido a esto la demanda energética es baja.

La aplicación del dispositivo debe obtener de la lista de canales la configuración para escanear los canales especificados. El resultado debe contener una lista de red (Network List) detallando los PAN activos en la red. Al igual que el router, se realizan varias peticiones de descubrimiento para saber cuántos elementos son los que hay en la red. La aplicación del dispositivo debe comparar la lista de canales con la lista de red para deducir a qué red debe unirse. En el algoritmo debe indicarse entre otras cosas: el modo de operación de la red, identificación del router o coordinador de la red, capacidad del router o coordinador, coste de enrutamiento, etc. Una vez hecho, debe chequear la asociación con el router o el coordinador ZigBee en el PAN.

Topologías de red de Zigbee.

Que es una topología.

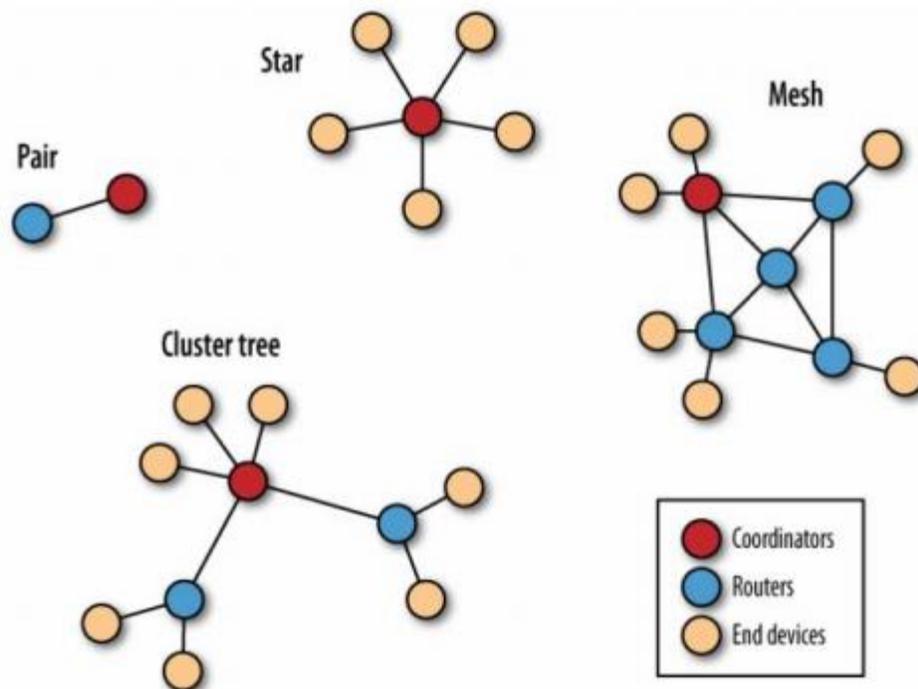
Una topología de red es un mapa de como varios dispositivos se comunican los unos con los otros. En esta forma de conexión intervienen varios elementos como lo son los nodos; que son dispositivos que realizan peticiones (computadoras, teléfonos, modems, router, etc.) mediante una red de comunicación, que suelen ser cables o por medio de señales de ondas.



- En los sistemas de redes tradicionales se conocen algunas topologías de redes como lo son: Punto a punto (point to point, PtP) o peer-to-peer (P2P)
- En bus (“conductor común” o bus) o lineal (line)
- En estrella (star)

- En anillo (ring) o circular
- En malla (mesh)
- En árbol (tree) o jerárquica
- Topología híbrida, combinada o mixta, por ej. circular de estrella, bus de estrella
- Cadena margarita (daisy chain)

Zigbee maneja, para el uso de sus protocolos de comunicación, cuatro tipos de redes, punto a punto (pair), estrella (star), maya (mesh), grupo de árbol (cluster tree). En la siguiente imagen podemos apreciar estas topologías, así como las tres formas de configurar los dispositivos, anteriormente mencionados.



Red Pair.

Este tipo de configuración de red es la más simple de todas, ya que solo se envía la información de un módulo a otro, en esta red solo se debe configurar un coordinador, el cual se encontrara conectado a la terminal, y un router o end device, cualquiera de los dos cumple con la función de enviar la información.

Red Star.

En la configuración de estrella, como su nombre lo indica y se observa en la imagen, se configura la terminal con el coordinador y solo en device se conectan a él, esto en forma centralizada ya que los end device no se podrán comunicar entre ellos solo podrán enviar información al concentrador.

Red Mesh.

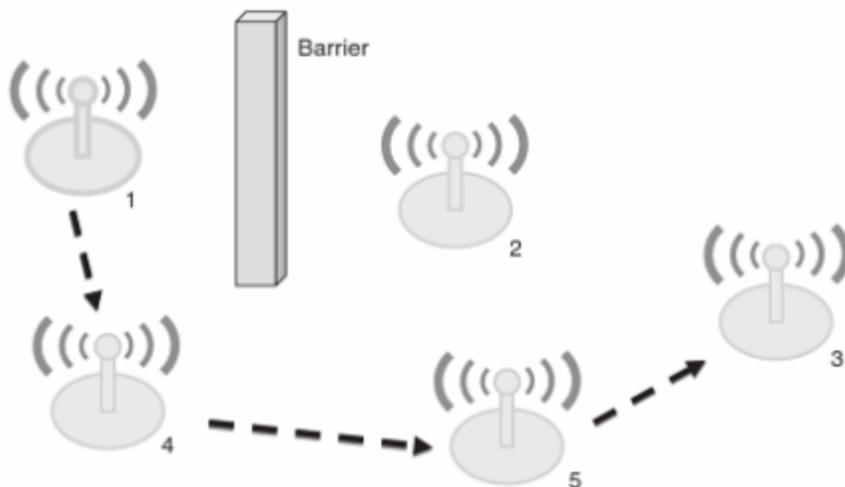
Este tipo de red cuenta con un coordinador, varios routers y varios en device; la particular más importante de esta red es que cuenta con los routers, los cuales permiten crear caminos de conexión entre los end device y el coordinador o entre otros routers, dependiendo de la posición en la que se encuentren los end device con respecto al coordinador. La ventaja de este tipo de red es que si en algún momento un dispositivo se aleja mucho del coordinador siempre puede usar los demás dispositivos para llevar, y en dado caso que alguno falle siempre buscaran el camino a su objetivo.

Red Clouster Tree.

Es una variación de la topología malla, por lo que no se le considera una cuarta topología como tal. En este diseño, los routers formaran una columna vertebral con los dispositivos finales, que están agrupados en torno a los routers.

Fortuño (2012, p.21) menciona: “De todas las topologías antes analizadas, la más ventajosa y característica del protocolo Zigbee es la Mesh. El poder hacer routing dinámico permite a este tipo de topología ganar sobretodo en fiabilidad en las comunicaciones.”

Y es cierto, es la que mayor utilidad tiene ya que, como se ha mencionado, permite hacer comunicación entre los dispositivos más alejados creando un camino buscando un camino de comunicación automáticamente. Además que en dado caso que algún dispositivo falle siempre puede buscar otro camino para entablar la comunicación, mas no siempre tiene que fallar algo para buscar un camino sino que también cuando hay obstáculos en la red se puede hacer la búsqueda de un camino no obstaculizado, físicamente.



Qué es XBee.

Xbee es un módulo de Digi para comunicación inalámbrica basado en el protocolo Zigbee, , estos proporcionan conectividad inalámbrica con otros dispositivos de la misma serie por un precio relativamente económico comparado con el de otros fabricantes.

Funcionan con la pila 802.15.4 (la base de ZigBee) y funcionan mediante un simple protocolo serie TTL. Permiten una comunicación bidireccional entre microcontroladores, ordenadores o prácticamente cualquier cosa que disponga de un puerto serie o Serie/USB a nivel TTL.

Soporta funciones de comunicación punto a punto o en red con varios módulos XBee, también llamado Mesh Network.

Características:

- Alimentación: 3.3V
- Velocidad de transferencia: 250kbps Max
- Potencia de salida: 1mW o 60mW (+18dBm)
- Alcance: 90metros o 1500 metros aprox.
- Certificado FCC
- 6 pines ADC de 10-bit
- 8 pines digitales IO
- Encriptación 128-bit
- Configuración local o de forma inalámbrica
- Comandos AT o API

Xbee Serie 1 (S1).

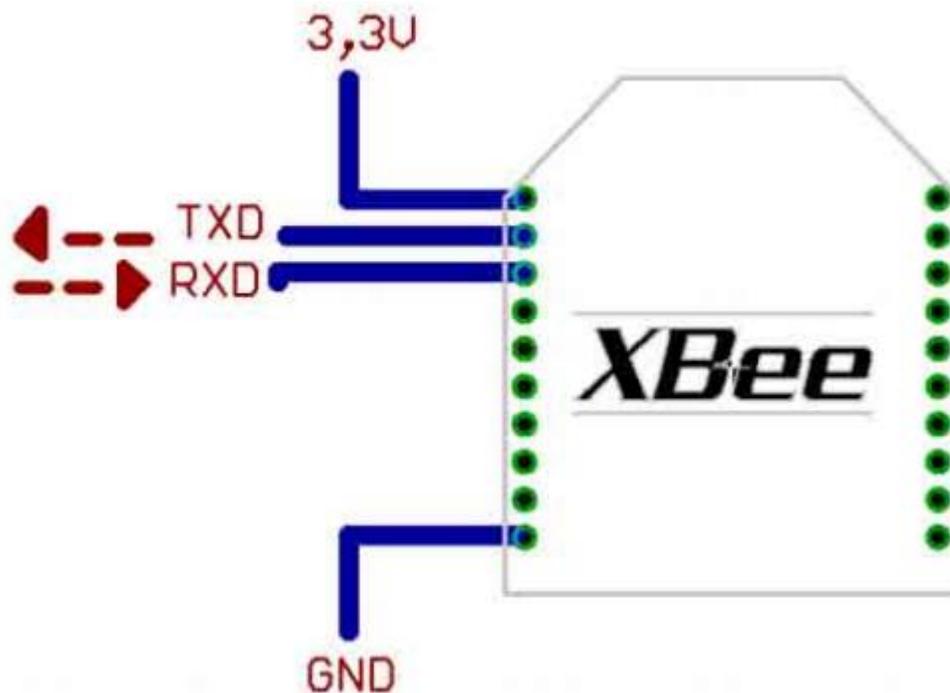
Los módulos RF XBee proporcionan a los fabricantes de equipos originales una superficie común compartida por múltiples plataformas, incluidas las topologías multipunto y ZigBee / Mesh, y las soluciones de 2,4 GHz y 900 MHz. Los OEM que despliegan el XBee pueden sustituir un XBee por otro, dependiendo de las necesidades dinámicas de las aplicaciones, con un desarrollo mínimo, un menor riesgo y un menor tiempo de lanzamiento al mercado. Los módulos XBee 802.15.4 RF son ideales para aplicaciones que requieren latencia baja y sincronización de comunicación predecible. Proporcionando una comunicación rápida y robusta en configuraciones de punto a punto, peer-to-peer y multipunto / estrella, los productos XBee 802.15.4 permiten conectividad de punto final robusta con facilidad. Ya sea que se despliegue como un reemplazo de cable puro para la comunicación serial simple, o como parte de una red de concentrador y radio más compleja, los módulos XBee 802.15.4 RF maximizan el rendimiento y la facilidad de desarrollo. Los módulos XBee 802.15.4 se conectan perfectamente con gateways, adaptadores de dispositivos y extensores compatibles, proporcionando a los desarrolladores una verdadera conectividad fuera del horizonte. El módulo actualizado XBee S2C 802.15.4 está construido con el SiliconLabs EM357 SoC y ofrece un consumo de energía mejorado, soporte para actualizaciones de firmware por aire y proporciona una ruta de actualización a los protocolos de malla DigiMesh® o ZigBee® si se desea.

Xbee Serie 2.

Los módulos XBee y XBee-PRO ZigBee RF proporcionan una conectividad inalámbrica rentable a dispositivos electrónicos. Son interoperables con otros dispositivos de conjunto de funciones ZigBee PRO, incluidos dispositivos de otros proveedores *. Los kits de desarrollo Digi ZigBee son la forma perfecta de comenzar el desarrollo de aplicaciones ZigBee. Los módulos XBee y XBee-PRO ZigBee son ideales para aplicaciones en los mercados de energía y control donde la eficiencia de fabricación es crítica. La Interfaz Periférica Serial (SPI) proporciona una interfaz de alta velocidad y optimiza la integración con microcontroladores integrados, reduciendo los costos de desarrollo y reduciendo el tiempo de lanzamiento al mercado. Los productos de la familia XBee requieren poca o ninguna configuración o desarrollo adicional. Las versiones programables del módulo XBee y XBee-PRO ZigBee facilitan la personalización de las aplicaciones. La programación directamente en el módulo elimina la necesidad de un procesador separado. Debido a que el software inalámbrico está aislado, las aplicaciones pueden desarrollarse sin riesgo para el rendimiento de RF o la seguridad. El módulo compatible con ZigBee de Digi se basa en los circuitos de radio Ember EM35x (EM357 y EM3587) en chip (SoC) de SiliconLabs, utilizando el procesador ARM Cortex™ M3 de 32 bits. La versión S2D EM3587 tiene una mayor huella de memoria para los clientes que deseen actualizar a Thread, una pila de redes basadas en IPv6.

Arquitectura Xbee.

La siguiente imagen muestra las conexiones mínimas del XBee.



Este módulo está configurado para operar de 2.8 a 3.3 Volts, cuenta con la conexión a tierra y los pines TXD y RXD, que son los que utiliza para realizar la comunicación con los demás dispositivos.

Modulo Xbee

Descripción de hardware:
La mochila permite conectar un modulo Xbee estandar y a su vez deja libre varios puertos de Arduino para uso de sensores, etc...

Pines del modulo Xbee Accesibles externamente

Jumpers de configuracion Modo USB/Xbee (con o sin AtMega)

Pines del modulo Xbee

1	VCC	AD0/DIO0	20
2	DOUT	AD1/DIO1	19
3	DIN/CONFIG	AD2/DIO2	18
4	DC0	AD3/DIO3	17
5	RESET	RTS/AD6/DIO6	16
6	PWM/RSSI	AD5/DIO5	15
7	PWM1	VREF	14
8	[reservado]	ONSLEEP	13
9	CTS/SLEEP_RD/D08	CTS/DIO7	12
10	GND	AD4/DIO4	11

Conexiones y funcionalidades:

- Xbee funciona a 3.3V y se comunica por un puerto serie con Arduino.
- 3 modelos de antenas: integrada, wire o externa.
- Puertos integrados de entrada y salida digitales y analogicos

Circuito Basico

Los parámetros en la configuración de los módulos XBee-Pro® S2B. Módulo XBee-Pro® S2B ZB soporta tanto modo transparente como API (Application Programming Interface) interfaces seriales (Digi®, 2009).

Funcionamiento transparente: los módulos actúan como una línea serie de reemplazo. Todos los datos recibidos a través del UART (Universal Asynchronous Receiver-Transmitter) pin DIN se ponen en cola para la transmisión por radiofrecuencia (RF). Cuando los datos de RF se reciben, los datos se envían a través del pin DOUT.

Operación de la API: es una alternativa a la operación transparente. La API basada en tramas extiende el nivel al de una aplicación host, de modo que puede interactuar con las capacidades de red del módulo.

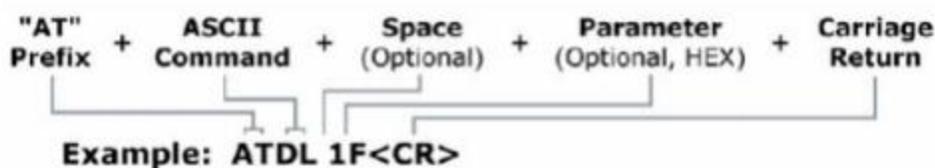
Cuando está en modo API, todos los datos que entran y salen del módulo están contenidos en las tramas que definen las operaciones o eventos dentro del módulo (Yawit, 2011).

(Tecnura, 2015)

Como lo menciona Tecnura, los xbee soportan varios modos para ser configurados. Para entenderlos mejor, procederemos a abordar en ellos más.

Modo Comando.

En este modo se permitirá ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otros aspectos. Para poder ingresar los comandos AT es necesario utilizar el Hyperteminal de Windows, el programa X-CTU o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.



Modo transparente.

Este modo está destinado principalmente a establecer una comunicación básica, es decir de punto a punto, donde no sea requerido ningún módulo de enrutamiento. También se usará para reemplazar alguna conexión serie por cable, ya que es la configuración que viene por defecto y es la más sencilla forma de configurar el módem. Básicamente todo lo que pasa por el puerto UART (DIN, pin 3), es enviado al módulo deseado, y lo que llega al módulo, es enviado de vuelta por el mismo puerto UART (DOUT, pin 2). Básicamente existen cuatro tipos de conexión transparente. La diferencia que existe entre estas es el número de nodos o puntos de acceso, y la forma en cómo éstos interactúan entre sí.

Modo de operación API.

Este modo es más complejo, pero permite el uso de tramas con cabeceras que aseguran la entrega de los datos, al estilo TCP. Extiende el nivel en el cual la aplicación del cliente, puede interactuar con las capacidades de red del módulo.

Cuando el módulo Xbee se encuentra en este modo, toda la información que entra y sale, es empaquetada en tramas, que definen operaciones y eventos dentro del módulo.

Así, una trama de Transmisión de información (información recibida por el pin 3 o DIN) incluye:

- Trama de información RF transmitida.
- Trama de comandos (equivalente a comandos AT).

Mientras que una trama de Recepción de información incluye:

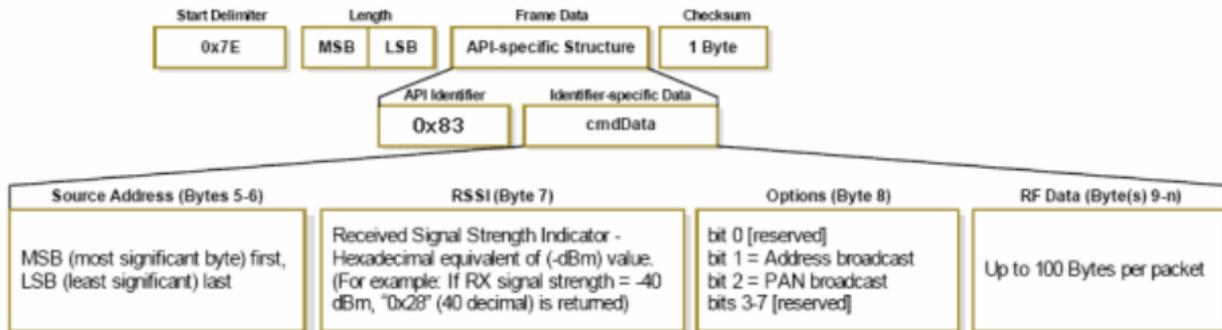
- Trama de información RF recibida.
- Comando de respuesta.
- Notificaciones de eventos como Reset, Associate, Dissociate, etc.

Entre las opciones que permite la API, se tienen:

- Transmitir información a múltiples destinatarios, sin entrar al modo de comandos
- Recibir estados de éxito/falla de cada paquete RF transmitido.
- Identificar la dirección de origen de cada paquete recibido.



El primer byte 0x7E indica el comienzo de la trama. Los dos bytes siguientes indican solamente la longitud de la trama de Datos (Data Frame) (es decir, toda la trama sin contar ni el byte 0x7E, ni el largo-Length, ni el byte Checksum). La estructura API que viene después se compone según veamos en la figura:



Parámetros de configuración de Xbee.

Xbee cuenta con una gran cantidad de parámetros, los cuales podemos configura dado el tipo de tipo de trabajo que realizaremos para que se ajuste a nuestras necesidades. A continuación se hablara un poco de ellos y de cómo poder configurarlos.

Parámetro CH: Permite asignar el canal que será utilizado para transmitir o recibir datos entre los módulos.

Rango	0x0B-0x1A
Valor seleccionado	0x0c

Parámetro ID	
Rango	0-0xFFFF
Valor seleccionado	123

Parámetro ID: Permite identificar a una Red de Área Personal en particular. Para enviar un mensaje a todas las Redes de Área Personal el valor de ID debe ser 0xFFFF (se supone la convención de Lenguaje C para representar números hexadecimales).

Parámetro DH: Corresponde a los últimos 32 bits (más significativos) de los 64 bits

correspondientes a la dirección de destino. Este valor, en conjunto con el parámetro DL, forma la dirección de destino usada para la transmisión. Para transmitir utilizando direcciones de 16 bits, el valor de DH debe ser cero, y el valor de DL debe ser inferior a 0xFFFF. Parámetro DL: Corresponde a los primeros 32 bits (menos significativos) de los 64 bits que forman la dirección de destino. El valor de DL, en conjunto con el valor de DH, define la dirección de destino. Para emplear direcciones de 16 bits, el valor de DH debe ser 0 y el valor de DL debe ser menor a 0xFFFF. Parámetro MY: Define una dirección de origen de 16 bits. Para deshabilitar la dirección de 16 bits y habilitar la dirección de 64 bits, se debe poner MY=0xFFFF. La dirección de origen de 64 bits siempre está habilitada.

Rango	0-0xFFFFFFFF				
Valor seleccionado	COOR	MOD 1	MOD 2	MOD 3	MOD 4
	0	DH Coor	DH Coor	DH Coor	DH Coor

Parámetro DH: Corresponde a los últimos 32 bits (más significativos) de los 64 bits correspondientes a la dirección de destino. Este valor, en conjunto con el parámetro DL, forma la dirección de destino usada para la transmisión. Para transmitir utilizando direcciones de 16 bits, el valor de DH debe ser cero, y el valor de DL debe ser inferior a 0xFFFF. Parámetro DL: Corresponde a los primeros 32 bits (menos significativos) de los 64 bits que forman la dirección de destino. El valor de DL, en conjunto con el valor de DH, define la dirección de destino. Para emplear direcciones de 16 bits, el valor de DH debe ser 0 y el valor de DL debe ser menor a

Rango	0-0xFFFFFFFF				
Valor seleccionado	COOR	MOD 1	MOD 2	MOD 3	MOD 4
	0	DL Coor	DL Coor	DL Coor	DL Coor

0xFFFF. Parámetro MY: Define una dirección de origen de 16 bits. Para deshabilitar la dirección de 16 bits y habilitar la dirección de 64 bits, se debe poner MY=0xFFFF. La dirección de origen de 64 bits siempre está habilitada.

Parámetro DL: Corresponde a los primeros 32 bits (menos significativos) de los 64 bits que forman la dirección de destino. El valor de DL, en conjunto con el valor de DH, define la dirección de destino. Para emplear direcciones de 16 bits, el valor de DH debe ser 0 y el valor de DL debe ser menor a 0xFFFF. Parámetro MY: Define una dirección de origen de 16 bits. Para deshabilitar la dirección de 16 bits y habilitar la dirección de 64 bits, se debe poner MY=0xFFFF. La dirección de origen de 64 bits siempre está habilitada.

Rango	0-0xFFFF				
Valor seleccionado	COOR	MOD 1	MOD 2	MOD 3	MOD 4
	0	1	2	3	4

Parámetro SC: Este parámetro define la lista de los canales que son examinados para determinar si están activos y el nivel de energía detectada. Este parámetro afecta cuando inicia el modo de comandos, durante la asociación de dispositivos finales y durante el establecimiento de un coordinador. Un bit representa el canal (desde el 11 hasta el 26). Con el valor por defecto se examinan los canales 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23. El valor asignado ha sido

SC=0x1FFE.

bit 0-0x0B bit 4-0x0F bit 8 - 0x13 bit12 - 0x17bit 1-0x0C bit 5-0x10 bit 9 - 0x14 bit13 - 0x18 bit 2-0x0D bit 6-0x11 bit 10- 0x15 bit14 - 0x19 bit 3-0x0E bit 7-0x12 bit 11- 0x16 bit15 - 0x1A

Parámetro SD: establece el tiempo para inspeccionar el canal. Para un dispositivo final, establece el tiempo durante el cual se examina el canal en un proceso de asociación. En el caso de un coordinador, si la opción de reasignación de identificador de PAN está activada, establece el periodo de tiempo durante el cual se examina el canal para localizar redes existentes. Asimismo, si la opción para reasignar el canal está activada, determina la cantidad de tiempo durante el cual se examinará el nivel de energía del canal, para determinar sobre cuál canal se operará. SD varía entre 0 a 15. El tiempo es dado por: tiempo = (número de canales)*(2SD)*(15,36 ms).

Rango	0-0x0F
Valor seleccionado	4

Número de bit	Opción de Asociación
0 – reasignación de identificador de PAN (ID PAN)	0 – El dispositivo final solo podrá asociarse con un coordinador que tenga el mismo identificador de red PAN.
	1 – El dispositivo final debe asociarse con un coordinador que opere con cualquier identificador de PAN.
1 – Reasignación de canal	0 – Solo podrá asociarse con un coordinador que opere dentro de su mismo canal
	1 – Debe asociarse a un coordinador que opere en cualquier canal.
2–Auto asociación	0 – El dispositivo final no intentará asociarse.
	1–El dispositivo final debe intentar asociarse hasta lograrlo.
3 – Sondeo del coordinador	0 – El dispositivo no sondeará al coordinador para requerir mensajes pendientes.
	1 – El dispositivo enviará un requerimiento.
4-7	Reservados
Valor seleccionado	0

Fuente: Digi® (2009)

Este valor permite establecer cómo un dispositivo final se puede asociar, como se indica en las siguientes opciones: Define cómo el coordinador se puede asociar.

Define cómo el coordinador se puede asociar.

Número de bit	Opción de asociación
0 – reasignación de identificador de PAN(ID PAN)	0 – El coordinador no examinará los canales para encontrar identificadores de PAN. Este operará con su identificador de PAN. 1 – El coordinador examinará los canales para determinar identificadores de PAN disponibles. Si un identificador de PAN encontrado crea conflicto, el identificador será cambiado.
1 – Reasignación de canal	0 – El coordinador no examinará el nivel de energía para encontrar un canal que se encuentre libre. El coordinador operará en el canal establecido por el parámetro CH. 1 – El coordinador examinará la energía de los canales para encontrar un canal que se encuentre libre y operará en este.
2–Permitir Asociación	0 – El coordinador no permitirá que cualquier dispositivo se asocie con él. 1 – El coordinador permitirá que un dispositivo se asocie con él.
4-7	Reservados
Valor seleccionado	0

Fuente: Digi® (2009).

Parámetro AS: Inspecciona un canal. Se envía un requerimiento a todas las redes en cada canal para solicitar información. AS determina el tiempo durante el cual se escuchará la respuesta al requerimiento. El mensaje recibido en respuesta al requerimiento contiene información como la dirección del coordinador, el identificador de PAN del coordinador, el tipo de dirección que usa el coordinador (16 o 64 bits), el canal en el cual opera, entre otros.

Parámetro ED: Comprueba el nivel de energía máximo que se detectó en cada canal.

Parámetro PL: Define el nivel de potencia con el cual el módulo de RF transmitirá.

Valor	Nivel de potencia
0	-10dBm
1	-6dBm
2	-4dBm
3	-2dBm
4	0dBm
Valor seleccionado	4

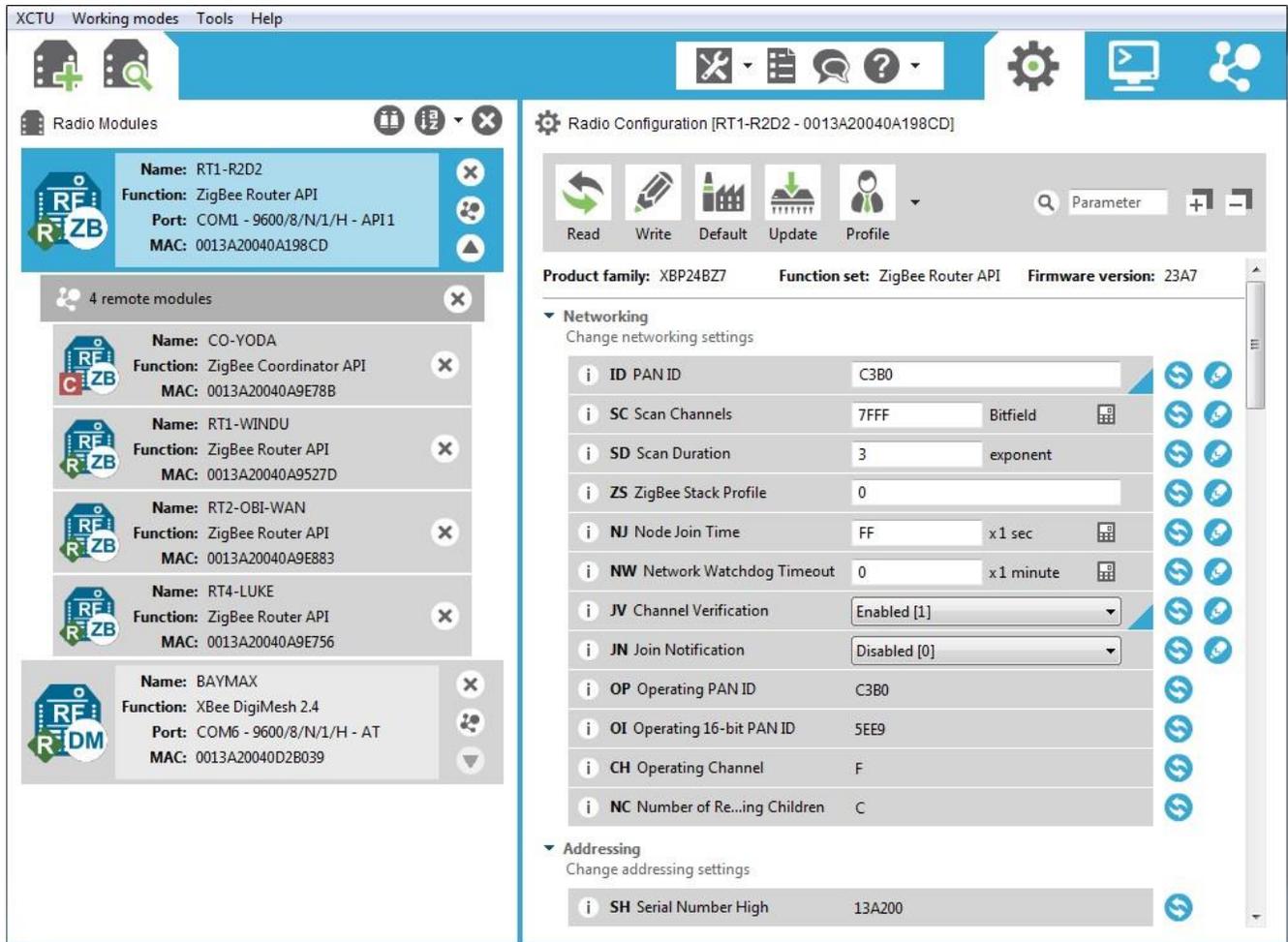
Fuente: Digi® (2009).

En resumen los parámetros que más nos ayudaran serán los siguientes:

Descripción	Parámetro	Módulo 0	Módulo 1	Módulo 2	Módulo 3	Módulo 4
Canal	CH	12	12	12	12	12
Identificador PAN	ID	1234	1234	1234	1234	1234
Dirección de Destino	DH	0	0	0	0	0
Dirección de Destino	DL	1 ó 2	0	0	0	0
Dirección de Origen (16 bits)	MY	0	1	2	3	3
Tiempo antes de transmitir	RN	0	0	0	0	0
Función (Coordinador/Dispositivo Final)	CE	1	0	0	0	0
Lista de canales para examinar	SC	0x1FFE	0x1FFE	0x1FFE	0x1FFE	0x1FFE
Tiempo para examinar el canal	SD	4	4	4	4	4
Asociación Dispositivo Final	A1	-	0	0	0	0
Asociación Coordinador	A2	0	-	-	-	-
Nivel de Potencia	PL	4(0dBm)	4(0dBm)	4(0dBm)	4(0dBm)	4(0dBm)
Umbral de decisión para transmitir	CA	0X2C (-44dBm)	0X2C (-44dBm)	0X2C (-44dBm)	0X2C (-44dBm)	0X2C (-44dBm)
Velocidad de Transmisión	BD	3(9600bps)	3(9600bps)	3(9600bps)	3(9600bps)	3(9600bps)
Modo de operación	AP	1	1	1	1	1
Modo de sueño	SM	1	0	0	0	0

X-CTU

XCTU es una aplicación multi-plataforma libre diseñada para permitir a los desarrolladores interactuar con módulos Digi RF a través de una interfaz gráfica fácil de usar. Incluye nuevas herramientas que hacen más fácil de configurar, configurar y probar módulos RF XBee®.



XCTU incluye todas las herramientas que un desarrollador necesita para obtener rápidamente en marcha y funcionando con XBee. Características únicas como el punto de vista gráfico de la red, lo que representa gráficamente la red XBee junto con la intensidad de la señal de cada conexión, y el formador de tramas API XBee, que intuitivamente ayuda a construir e interpretar tramas API para XBees está utilizando en modo API, se combinan para hacer que el desarrollo en la plataforma XBee más fácil que nunca.

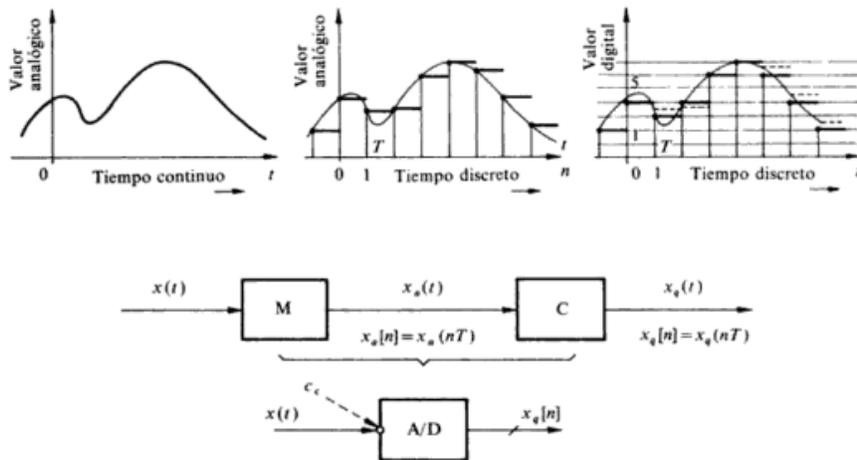
Otros aspectos destacados de XCTU incluyen las siguientes características:

- Puede administrar y configurar varios dispositivos de RF, incluso de forma remota los dispositivos conectados (over-the-air).
- La actualización del firmware proceso sin problemas restaura la configuración del módulo, la manipulación automática de modo y la velocidad de transmisión cambios.
- Dos específica API y AT consolas, se han diseñado desde cero para comunicarse con sus dispositivos de radio.
- Ahora puede guardar sus sesiones de consola y cargarlos en un PC que ejecuta XCTU diferente.
- XCTU incluye un conjunto de herramientas embebidas que se pueden ejecutar sin tener ningún módulo de RF conectado:
 - Generador de tramas: generar fácilmente cualquier tipo de marco de API para salvar su valor.
 - Intérprete de marcos: descodificar un marco de principios activos y de ver sus valores específicos del marco.
 - Recuperación: Recupera módulos de radio que han dañado firmware o están en modo de programación.
 - Carga de la sesión de la consola: Cargar una sesión de consola guardada en cualquier XCTU PC en funcionamiento.
 - Prueba de rango: Realizar una prueba de rango entre 2 módulos de radio de la misma red.
 - Explorador de firmware: Navegar a través de la biblioteca de firmware XCTU.
- Un proceso de actualización le permite actualizar automáticamente la aplicación en sí y el firmware de radio biblioteca sin necesidad de descargar los archivos adicionales.
- XCTU contiene documentación completa y exhaustiva que se puede acceder en cualquier momento.

Convertor analógico – digital (A/D).

Un sistema digital es pues un procesador, general o especializado, cuya entrada es una sucesión de números que designaremos por señal discreta. Po otro lado, en general, las señales a tratar son funciones del tiempo que pueden, en principio, tomar cualquier valor, las designaremos por señales analógicas o en tiempo continuo. Por consiguiente, un sistema digital debe contener como paso previo un dispositivo para convertir una señal analógica en una señal discreta. A este dispositivo se le denomina convertor analógico-digital (Convertor A/D) y existen como circuito integrado en varias modalidades, velocidades de conversión y coste.

La entrada $x(t)$ de un conversor A/D es una señal analógica que puede variar en cualquier instante t ; se le llama muchas veces analógica en tiempo continuo para distinguirla de otras señales analógicas que intervienen en el proceso de conversión.



Como primera parte está el sistema M que tomara muestras de la señal analógica $x(t)$ a intervalos de tiempo constantes T . Cada muestra $x[n] = x(nT)$ permanece inalterada a su salida durante el intervalo de tiempo T correspondiente. Se forma así la señal analógica en escalera $x(t)$ que muestra la figura. Esta será una señal analógicamente en tiempo discreto, ya que puede tomar cualquier valor pero cambia solamente en instantes discretos múltiplos de T . (Papoulis, 1968, p10)

Sistemas embebidos.

Un sistema embebido o empotrado es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas, frecuentemente en un sistema de computación en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas. En un sistema embebido la mayoría de los componentes se encuentran incluidos en la placa base (tarjeta de vídeo, audio, módem, etc.) y muchas veces los dispositivos resultantes no tienen el aspecto de lo que se suele asociar a una computadora. Algunos ejemplos de sistemas embebidos podrían ser dispositivos como un taxímetro, un sistema de control de acceso, la electrónica que controla una máquina expendedora o el sistema de control de una fotocopiadora entre otras múltiples aplicaciones.

Por lo general los sistemas embebidos se pueden programar directamente en el lenguaje ensamblador del microcontrolador o microprocesador incorporado sobre el mismo, o también, utilizando los compiladores específicos, pueden utilizarse lenguajes como C o C++; en algunos casos, cuando el tiempo de respuesta de la aplicación no es un factor crítico, también pueden usarse lenguajes interpretados como JAVA.

Puesto que los sistemas embebidos se pueden fabricar por decenas de millares o por millones de unidades, una de las principales preocupaciones es reducir los costes. Los sistemas embebidos suelen usar un procesador relativamente pequeño y una memoria pequeña para ello. Los primeros equipos embebidos que se desarrollaron fueron elaborados por IBM en los años 1980.

Los programas de sistemas embebidos se enfrentan normalmente a tareas de procesamiento en tiempo real.

Raspberry Pi.

Raspberry Pi es un computador de placa reducida, computador de placa única o computador de placa simple (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

Aunque no se indica expresamente si es hardware libre o con derechos de marca, en su sección de preguntas y respuestas frecuentes (FAQs) explican que disponen de contratos de distribución y venta con dos empresas, pero al mismo tiempo cualquiera puede convertirse en revendedor o redistribuidor de las tarjetas RaspBerry Pi: por lo que se entiende que es un producto con propiedad registrada pero de uso libre. De esa forma mantienen el control de la plataforma pero permitiendo su uso libre tanto a nivel educativo como particular. Tampoco deja claro si es posible utilizarlo a nivel empresarial u obtener beneficios con su uso, asunto que se debe consultar con la fundación.

En cambio el software sí es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite otros sistemas operativos, incluido una versión de Windows 10. El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos "Turbo" para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa. El 29 de febrero de 2012 la fundación empezó a aceptar órdenes de compra del modelo B, y el 4 de febrero de 2013 del modelo A.

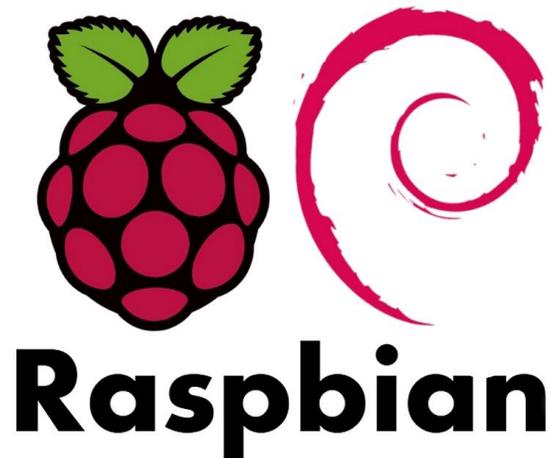
La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora); y promueve principalmente el aprendizaje del lenguaje de programación Python. Otros lenguajes también soportados son Tiny BASIC, C, Perl y Ruby.

Especificaciones técnicas.

	Raspberry Pi 1 Modelo A	Raspberry Pi 1 Modelo B	Raspberry Pi 1 Modelo B+	Raspberry Pi 2 Modelo B	Raspberry Pi 3 Modelo B
SoC: ⁵	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB) ³			Broadcom BCM2836 (CPU + GPU + DSP + SDRAM + Puerto USB)	Broadcom BCM2837 (CPU + GPU + DSP + SDRAM + Puerto USB)
CPU:	ARM 1176JZF-S a 700 MHz (familia ARM11) ³			900 MHz quad-core ARM Cortex A7	1.2GHz 64-bit quad-core ARMv8
Juego de instrucciones:	RISC de 32 bits				
GPU:	Broadcom VideoCore IV, ⁶¹ OpenGL ES 2.0, MPEG-2 y VC-1 (con licencia), ⁵⁹ 1080p30 H.264/MPEG-4 AVC ³				
Memoria (SDRAM):	256 MiB (compartidos con la GPU)	512 MiB (compartidos con la GPU) ⁴ desde el 15 de octubre de 2012		1 GB (compartidos con la GPU)	
Puertos USB 2.0: ⁵⁵	1	2 (vía hub USB integrado) ⁵⁴	4		
Entradas de video: ⁶²	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la RPF				
Salidas de video: ⁵	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), ⁶³ Interfaz DSI para panel LCD ^{64 65}				
Salidas de audio: ⁵	Conector de 3.5 mm, HDMI				
Almacenamiento integrado:	SD / MMC / ranura para SDIO		MicroSD		
Conectividad de red: ⁵	Ninguna	10/100 Ethernet (RJ-45) vía hub USB ⁵⁴			10/100 Ethernet (RJ-45) vía hub USB ⁶⁶ , Wifi 802.11n, Bluetooth 4.1
Periféricos de bajo nivel:	8 x GPIO, SPI, I2C, UART ⁶¹			17 x GPIO y un bus HAT ID	
Reloj en tiempo real: ⁵	Ninguno				
Consumo energético:	500 mA, (3.5 W) ⁵	700 mA, (3.5 W)	600 mA, (3.0 W)	800 mA, (4.0 W)	
Fuente de alimentación: ⁵	5 V vía Micro USB o GPIO header				
Dimensiones:	85.60mm x 53.98mm ⁶⁷ (3.370 x 2.125 inch)				
Sistemas operativos soportados:	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux, SUSE ⁶⁸ Linux Enterprise Server for ARM. RISC OS ²				

Sistema operativo.

El Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse.



Slackware ARM (también llamada ARMedslack) versión 13.37 y posteriores arranca sin ninguna modificación. Los 128-496 MiB de memoria RAM disponible en la Raspberry Pi, cubren los necesarios 64 MiB de RAM para arrancar esta distribución en sistemas ARM y i386 sin usar interfaz gráfica (el administrador de ventanas Fluxbox que funciona bajo X Window System requiere 48 MiB de memoria RAM adicional). Por otro lado, se están creando distribuciones más específicas y ligeras como IPfire (distribución para ser usada como firewall),⁸⁵ u OpenELEC y OSMC (distribuciones con el centro multimedia Kodi).

A la GPU se accede mediante una imagen del firmware de código cerrado, llamado blob binario, que se carga dentro de la GPU al arrancar desde la tarjeta SD. El blob binario está asociado a los drivers Linux que también son de código cerrado. Las aplicaciones hacen llamadas a las librerías de tiempo de ejecución que son de código abierto, y estas librerías hacen llamadas a unos drivers de código abierto en el kernel de Linux. La API del driver del kernel es específica para estas librerías. Las aplicaciones que usan vídeo hacen uso de OpenMAX, las aplicaciones 3D usan OpenGL ES y las aplicaciones 2D usan OpenVG; OpenGL ES y OpenVG hacen uso de EGL y este último, del driver de código abierto del kernel.

El 19 de febrero de 2012, la fundación lanzó un prototipo de imagen de tarjeta SD que almacenaba un sistema operativo y que podía ser cargado en una tarjeta SD. La imagen se basaba en Debian 6.0 (Squeeze), con el escritorio LXDE y el navegador Midori, más algunas herramientas de programación. La imagen funcionaba bajo QEMU permitiendo que el Raspberry Pi pudiera ser emulado en otros sistemas.

El 8 de marzo de 2012, la fundación lanzó Raspberry Pi Fedora Remix (actualmente llamada Pidora), que en el momento de era la distribución recomendada por la fundación, y fue desarrollada en la universidad de Séneca, en Canadá. También se propuso crear una tienda de aplicaciones para que la gente intercambiara programas.

El 24 de octubre de 2012, Alex Bradbury, director de desarrollo Linux de la fundación, anunció que todo el código del driver de la GPU Videocore que se ejecuta en ARM sería de código abierto,

mediante licencia BSD modificada de 3 cláusulas. El código fuente está disponible en un repositorio de la fundación en GitHub.

El 5 de noviembre de 2012, Eben Upton anunció el lanzamiento del sistema operativo RISC OS 5 para Raspberry Pi a la comunidad, pudiéndose descargar la imagen de forma gratuita desde la web de la fundación. Su relación con la comunidad RISC OS se remontaba a julio de 2011, cuando habló en ella de una hipotética versión. El sistema operativo incluye una gran cantidad de aplicaciones como !NetSurf para la navegación web, !StrongED para editar texto, !Maestro para editar música, !Packman para la gestión de paquetes o una tienda de aplicaciones llamada !Store donde se puede encontrar aplicaciones gratuitas o de pago. Además se incluyen manuales para crear aplicaciones en BASIC para el sistema operativo.

El 24 de noviembre de 2012, se anunció en la Minecon de París, el juego Minecraft: Pi Edition para Raspberry Pi, basado en la versión Minecraft: Pocket Edition para teléfonos inteligentes y tabletas. La descarga se hizo disponible de forma oficial y gratuita por primera vez el 12 de febrero de 2013 desde el blog del juego, como versión 0.1.1 alpha, junto a instrucciones para ejecutarlo en Raspbian Wheezy. Una de las características principales de este lanzamiento es poder interactuar con el juego mediante programación, con la intención de motivar a los niños a aprender a programar.

El 25 de mayo de 2013, la fundación informó de que se estaba trabajando en una versión del servidor gráfico Wayland para Raspberry Pi, para sustituir al sistema de ventanas X. Con este cambio se lograría suavidad al usar la interfaz gráfica del escritorio, ya que el procesamiento lo realizaría la GPU Videocore y no la CPU, sin interferir en el renderizado 3D.

El 3 de junio de 2013, fue lanzado en la web de la fundación para su descarga la aplicación NOOBS (New Out of Box Software), utilidad que facilita la instalación de diferentes sistemas operativos para Raspberry Pi. Se distribuye en forma de archivo zip que se copia descomprimido a una tarjeta SD de 4 GB o superior, y una vez arrancada la placa con la tarjeta por primera vez, aparece un menú en que se da la opción de instalar una de las diferentes distribuciones en el espacio libre de la tarjeta de memoria, o acceder a internet con el navegador Arora integrado. Más adelante si se desea, es posible acceder a este menú apretando la tecla shift durante el arranque para reinstalar el sistema operativo, elegir otro, o editar el archivo config.txt. NOOBS contiene las distribuciones Linux de carácter general Raspbian, Arch Linux ARM y Pidora; las distribuciones Linux para mediacenter con Kodi Openelec y RaspBMC; y el sistema operativo Risc OS 5.

El 26 de septiembre de 2013, se añadió a los repositorios de Raspbian una versión oficial de Oracle Java JDK ARM con soporte para coma flotante por hardware, que ofrece bastante más rendimiento que la versión OpenJDK ARM ya existente hasta ese momento y más compatibilidad con aplicaciones. También se anunció que esta versión de Oracle Java JDK se incluiría dentro de la distribución en futuras versiones de Raspbian.

Servidor LAMP.

Se trata de una combinación muy popular en los servidores de Linux, que obedece al acrónimo de Linux, Apache, MySQL y PHP.



LAMP es el acrónimo usado para describir un sistema de infraestructura de internet que usa las siguientes herramientas:

- Linux, el sistema operativo; En algunos casos también se refiere a LDAP.
- Apache, el servidor web.
- MySQL/MariaDB, el gestor de bases de datos.
- Perl, PHP, o Python, los lenguajes de programación.

La combinación de estas tecnologías es usada principalmente para definir la infraestructura de un servidor web, utilizando un paradigma de programación para el desarrollo.

A pesar de que el origen de estos programas de código abierto no ha sido específicamente diseñado para trabajar entre sí, la combinación se popularizó debido a su bajo coste de adquisición y ubicuidad de sus componentes (ya que vienen pre-instalados en la mayoría de las distribuciones linux). Cuando son combinados, representan un conjunto de soluciones que soportan servidores de aplicaciones.

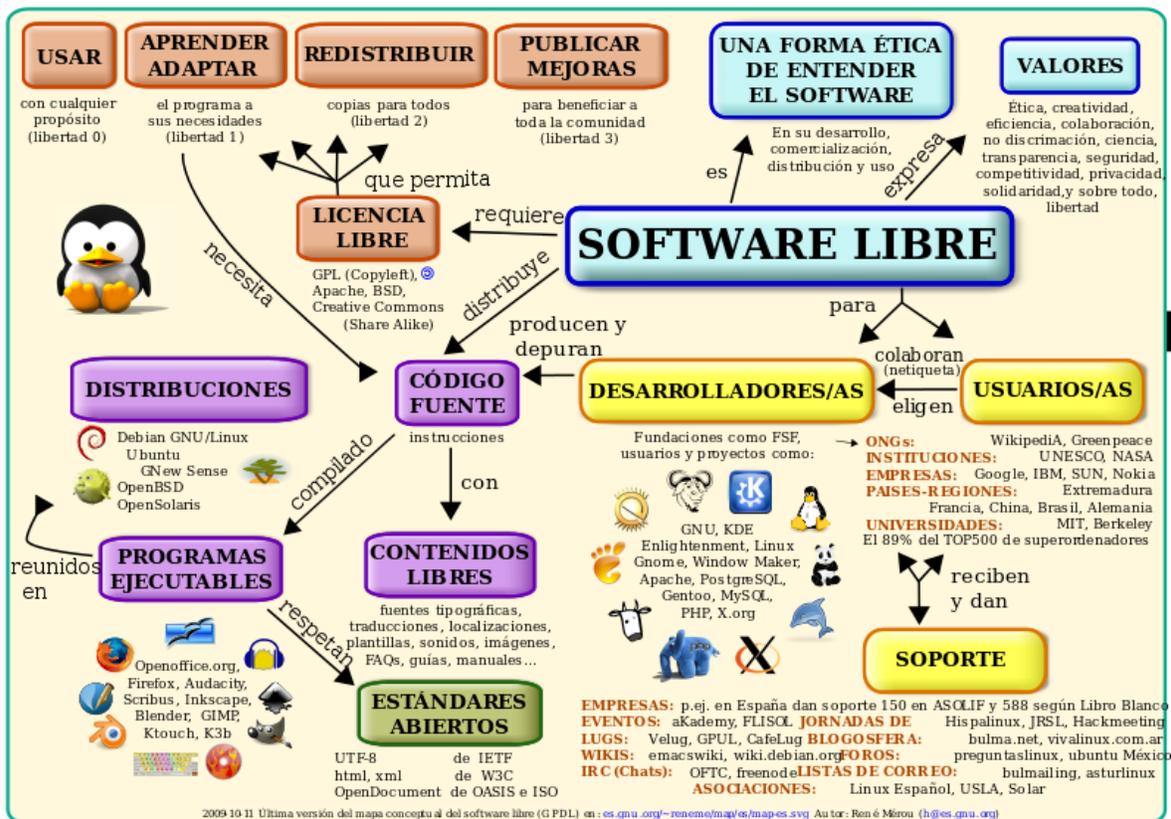
Ampliamente promocionado por el editor de la editorial O'Reilly, Dale Dougherty, a sugerencia de David Axmark y Monty Widenius desarrolladores de MySQL, la influencia de la editorial O'Reilly en el mundo del software libre hizo que el término se popularizara rápidamente en todo el mundo.

Linux.

Linux es un sistema operativo de termino de uso libre, el término software libre refiere el conjunto de software (programa informático) que por elección manifiesta de su autor, puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin y redistribuido con o sin cambios o mejoras. Su definición está asociada al nacimiento del movimiento de software libre, encabezado por Richard Stallman y la consecuente fundación en 1985 de la Free Software Foundation, que coloca la libertad del usuario informático como propósito ético³ fundamental. Proviene del término en inglés free software, que presenta ambigüedad entre los significados «libre» y «gratis» asociados a la palabra free. Por esto que suele ser considerado como software gratuito y no como software que puede ser modificado sin restricciones de licencia. En este sentido es necesario resaltar que la libertad tiene que ver con el uso y no con la gratuidad.

Un programa informático es software libre si otorga a los usuarios todas estas libertades de manera adecuada. De lo contrario no es libre. Existen diversos esquemas de distribución que no

son libres, y si bien podemos distinguirlos sobre la base de cuánto les falta para llegar a ser libres, su uso bien puede ser considerado contrario a la ética en todos los casos por igual.



Apache (Servidor HTTP).



El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (httpd).

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Jugó un papel fundamental en el desarrollo fundamental de la World Wide Web y alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. (Estadísticas históricas y de uso diario proporcionadas por Netcraft3). En 2009 se convirtió en el primer servidor web que alojó más de 100 millones de sitios web4 .

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

MySQL.

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.



MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL A.B. fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto

es lo que posibilita el esquema de doble licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una Community, distribuida bajo la Licencia pública general de GNU, versión 2, y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos. Las versiones Enterprise incluyen productos o servicios adicionales tales como herramientas de monitorización y soporte oficial. En 2009 se creó un fork denominado MariaDB por algunos desarrolladores (incluido algunos desarrolladores originales de MySQL) descontentos con el modelo de desarrollo y el hecho de que una misma empresa controle a la vez los productos MySQL y Oracle Database.

Está desarrollado en su mayor parte en ANSI C y C++ . Tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo LAMP y WAMP.

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr, y YouTube.

PHP.



PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP

Características:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Python.



Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa escritura dinámica y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa escrito dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostiles a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.

PROCEDIMIENTO Y DESCRIPCIÓN DE ACTIVIDADES.

Con el presente proyecto se plantean varias fases de desarrollo, dado a que es un proyecto complejo se ha decidido avanzar su desarrollo, utilizando una metodología águila como lo es scrum, mediante la cual se tendrá la posibilidad de desarrollar un prototipo y comprender cuáles serán las áreas de oportunidades o de mejora de la creación de tanto de la red como del senados de los datos.

Después de un largo análisis, respecto a la estructuración de las conexiones, se logró determinar cómo se llevara a cabo y resultado en lo siguiente:

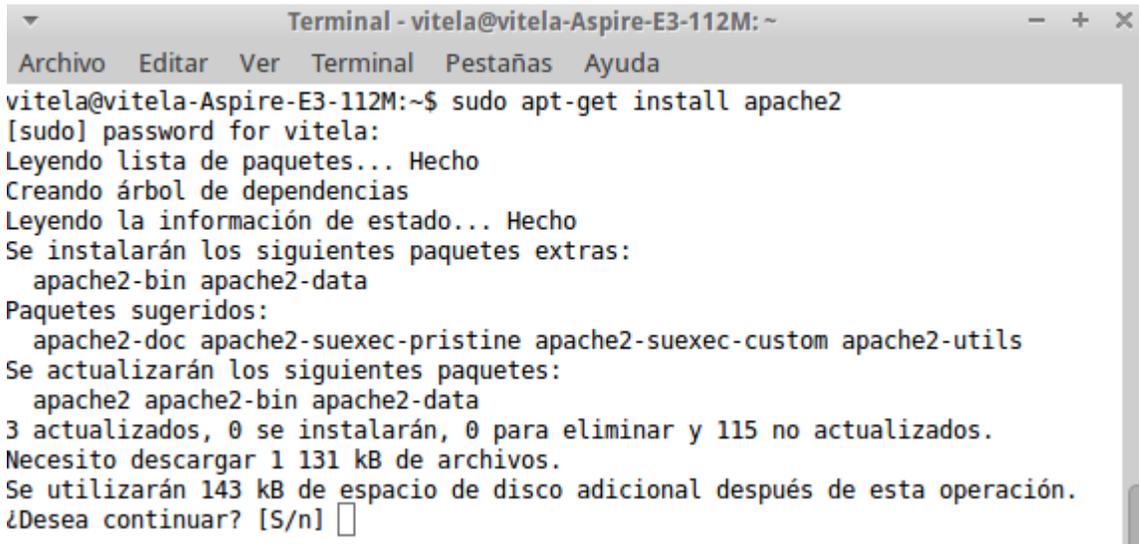
- Configurar una red con el protocolo Zigbee, para esto se montara en cada dron un módulo xbee para poder realizar él envío de datos.
- Configurar un servidor LAMP para la gestión de la base de datos y las páginas web.
- Crear una interfaz web para poder convertir la información de los módulos xbee en gráficas.
- Configurar el raspberry para poder realizar las lecturas de los sensores.
- Configurar al servidor un sistema de nombres de domino (DNS).
- Configurar con python las lecturas de los xbee del lado del servidor (dispositivo coordinador).
- Configurar las librerías GPIO para realizar lecturas de datos de los sensores mediante python en la raspberry.



RESULTADOS.

Los primeros pasos realizados en el desarrollo de este proyecto fue la instalación de un servidor LAMP para poder montar la base de datos y tener un lugar donde almacenar la información, para esto se realizó la instalación de los componentes sobre una distribución Linux, el cual es el primer paso de montar un servidor LAMP.

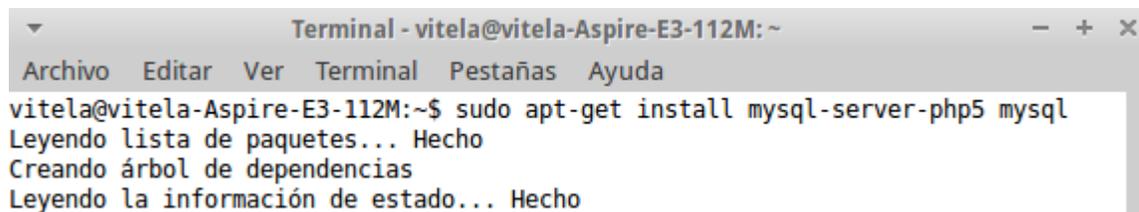
Una vez con el sistema Linux instalado procedemos a instalar apache, tecleamos en una consola lo siguiente y presionamos que sí.

A terminal window titled "Terminal - vitela@vitela-Aspire-E3-112M: ~" with a menu bar containing "Archivo", "Editar", "Ver", "Terminal", "Pestañas", and "Ayuda". The terminal output shows the command "sudo apt-get install apache2" being executed. The output includes: "[sudo] password for vitela:", "Leyendo lista de paquetes... Hecho", "Creando árbol de dependencias", "Leyendo la información de estado... Hecho", "Se instalarán los siguientes paquetes extras: apache2-bin apache2-data", "Paquetes sugeridos: apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils", "Se actualizarán los siguientes paquetes: apache2 apache2-bin apache2-data", "3 actualizados, 0 se instalarán, 0 para eliminar y 115 no actualizados.", "Necesito descargar 1 131 kB de archivos.", "Se utilizarán 143 kB de espacio de disco adicional después de esta operación.", and "¿Desea continuar? [S/n]".

```
Terminal - vitela@vitela-Aspire-E3-112M: ~
Archivo  Editar  Ver    Terminal  Pestañas  Ayuda
vitela@vitela-Aspire-E3-112M:~$ sudo apt-get install apache2
[sudo] password for vitela:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-bin apache2-data
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom apache2-utils
Se actualizarán los siguientes paquetes:
  apache2 apache2-bin apache2-data
3 actualizados, 0 se instalarán, 0 para eliminar y 115 no actualizados.
Necesito descargar 1 131 kB de archivos.
Se utilizarán 143 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Después procedemos a instalar mysql, para eso escribimos el siguiente comando y aceptamos de nuevo.

Durante la instalación, el servidor te pedirá que selecciones y confirmes una contraseña para el usuario "root" de MySQL. Esta es una cuenta administrativa en MySQL que ha aumentado privilegios. Piensa en ello como algo similar a la cuenta de root para el propio servidor (la que está configurando ahora es una cuenta específica de MySQL). Después procedemos a instalar PHP.

A terminal window titled "Terminal - vitela@vitela-Aspire-E3-112M: ~" with a menu bar containing "Archivo", "Editar", "Ver", "Terminal", "Pestañas", and "Ayuda". The terminal output shows the command "sudo apt-get install mysql-server-php5 mysql" being executed. The output includes: "Leyendo lista de paquetes... Hecho", "Creando árbol de dependencias", and "Leyendo la información de estado... Hecho".

```
Terminal - vitela@vitela-Aspire-E3-112M: ~
Archivo  Editar  Ver    Terminal  Pestañas  Ayuda
vitela@vitela-Aspire-E3-112M:~$ sudo apt-get install mysql-server-php5 mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Durante la instalación, el servidor te pedirá que selecciones y confirmes una contraseña para el usuario "root" de MySQL. Esta es una cuenta administrativa en MySQL que ha aumentado privilegios. Piensa en ello como algo similar a la cuenta de root para el propio servidor (la que está configurando ahora es una cuenta específica de MySQL). Después procedemos a instalar PHP.

Después procedemos a instalar PHP.

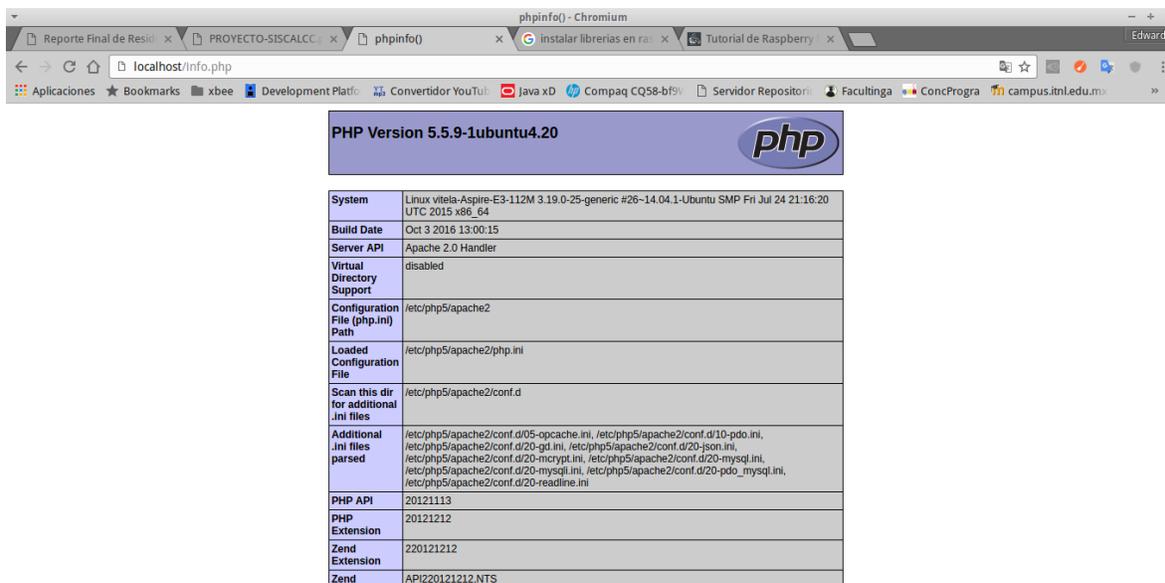
```

Terminal - vitela@vitela-Aspire-E3-112M: ~
Archivo Editar Ver Terminal Pestañas Ayuda
vitela@vitela-Aspire-E3-112M:~$ sudo apt-get install libapache2-mod-php5 php5 p
p5-mcrypt
[sudo] password for vitela:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
php5-mcrypt ya está en su versión más reciente.
libapache2-mod-php5 ya está en su versión más reciente.
Se actualizarán los siguientes paquetes:
  php5
1 actualizados, 0 se instalarán, 0 para eliminar y 117 no actualizados.
Necesito descargar 1 306 B de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] 

```

Por ultimo accedamos, por medio de un navegador web, a la siguiente dirección para verificar que nuestro servidor LAMP está completamente configurado y funcionando.

<http://localhost/info.php>



En esta página podemos visualizar la correcta instalación y los medios con los cuales php se relaciona.

Seguido esto viendo desde el lado del servidor procedimos a crear las tablas que serían requeridas para la base de datos, estas tablas no tienen relaciones dado a que son completamente distintas, solo almacenamos el valor de la lectura de cada sensor, la fecha de medición, etc.

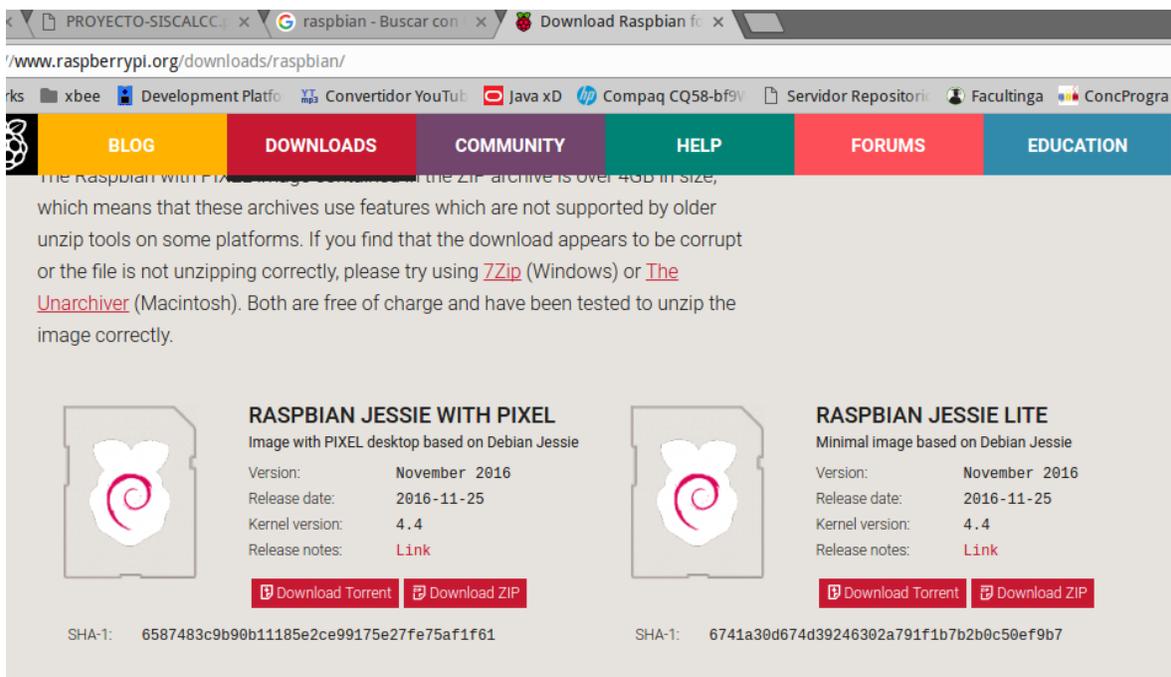
```
residencia.sensorNitrogeno
# id : int(11)
ppm : varchar(30)
```

```
residencia.sensorDioxidoCarbono
# id : int(11)
ppm : varchar(30)
```

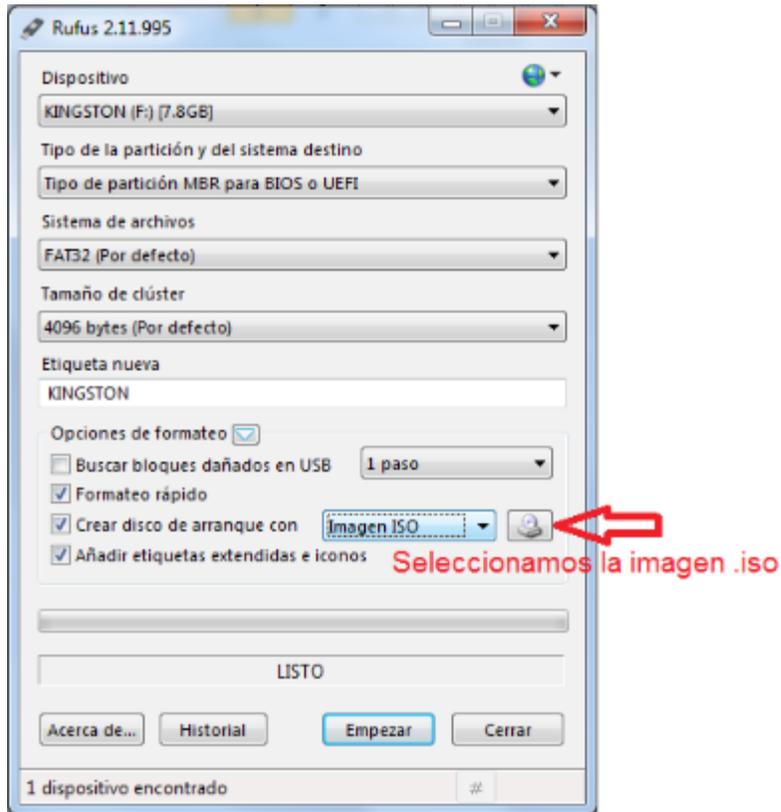
```
residencia.sensorMonoxido
# id : int(11)
ppm : varchar(30)
```

```
residencia.sensorHidrogeno
# id : int(11)
ppm : varchar(30)
```

Una vez hecho esto procedemos a instalar el sistema raspian en el raspberry pi, para esto procedemos a descarga la imagen ISO desde la página oficial.



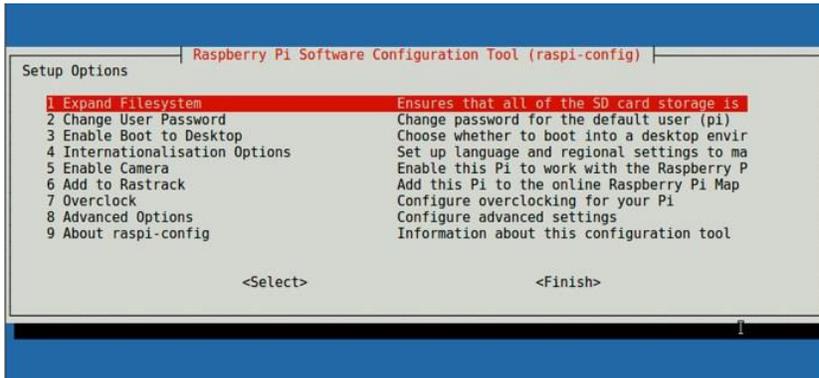
Una vez descargada solo quedara hacerla booteable sobre una tarjeta de datos (microSD) e insertarla en el raspberry.



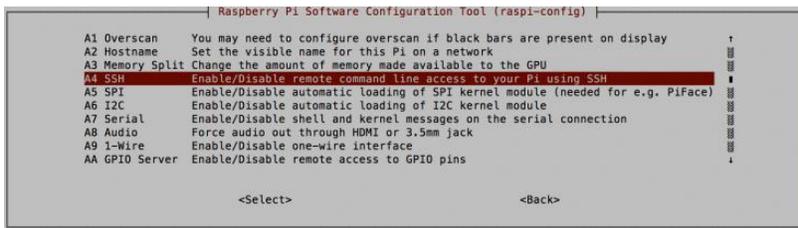
1. Montaremos la imagen en una memoria microSD con el programa Rufus y seleccionaremos empezar y al finalizar tendremos listo la usb para configurar nuestro sistema.
Insertar la microSD en la tarjeta Raspberry.



2. Conectar la tarjeta a una fuente de alimentación de 5 volts, por medio de un cable Hdmí conectaremos la tarjeta a una pantalla/monitor y conectar el teclado.
3. En el siguiente menú seleccionar la opción 1 para expandir el sistema y poder utilizar toda la capacidad de almacenamiento de la memoria.

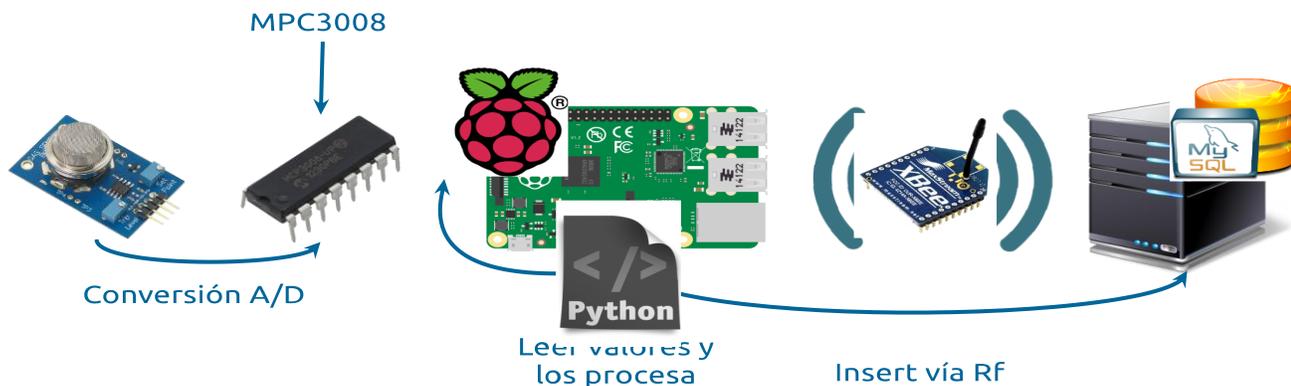


4. Seleccionar la opción 8 para ir a opciones avanzadas y allí habilitar la opción SSH, SPI y GPIO.

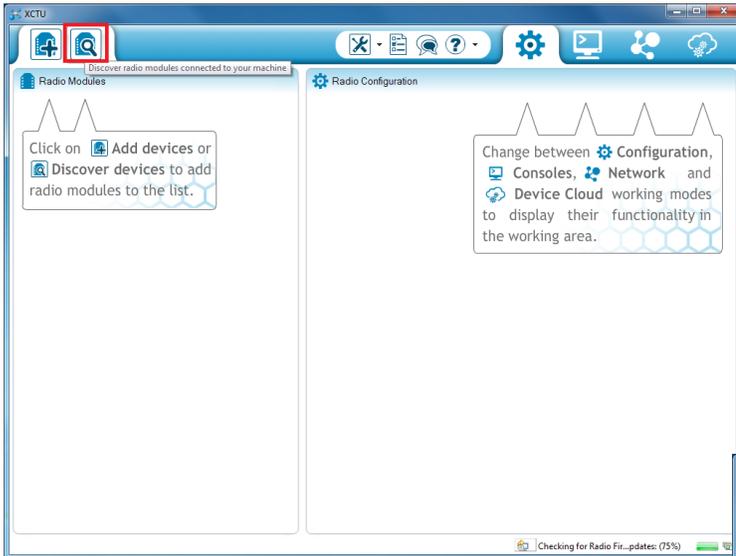


5. Dirigirse al menú anterior y finalizar

Ya instalado y configurado todo lo anterior procedemos configurar con python la lectura y envío de datos desde la raspberry a la terminal, como se describe en la siguiente imagen.



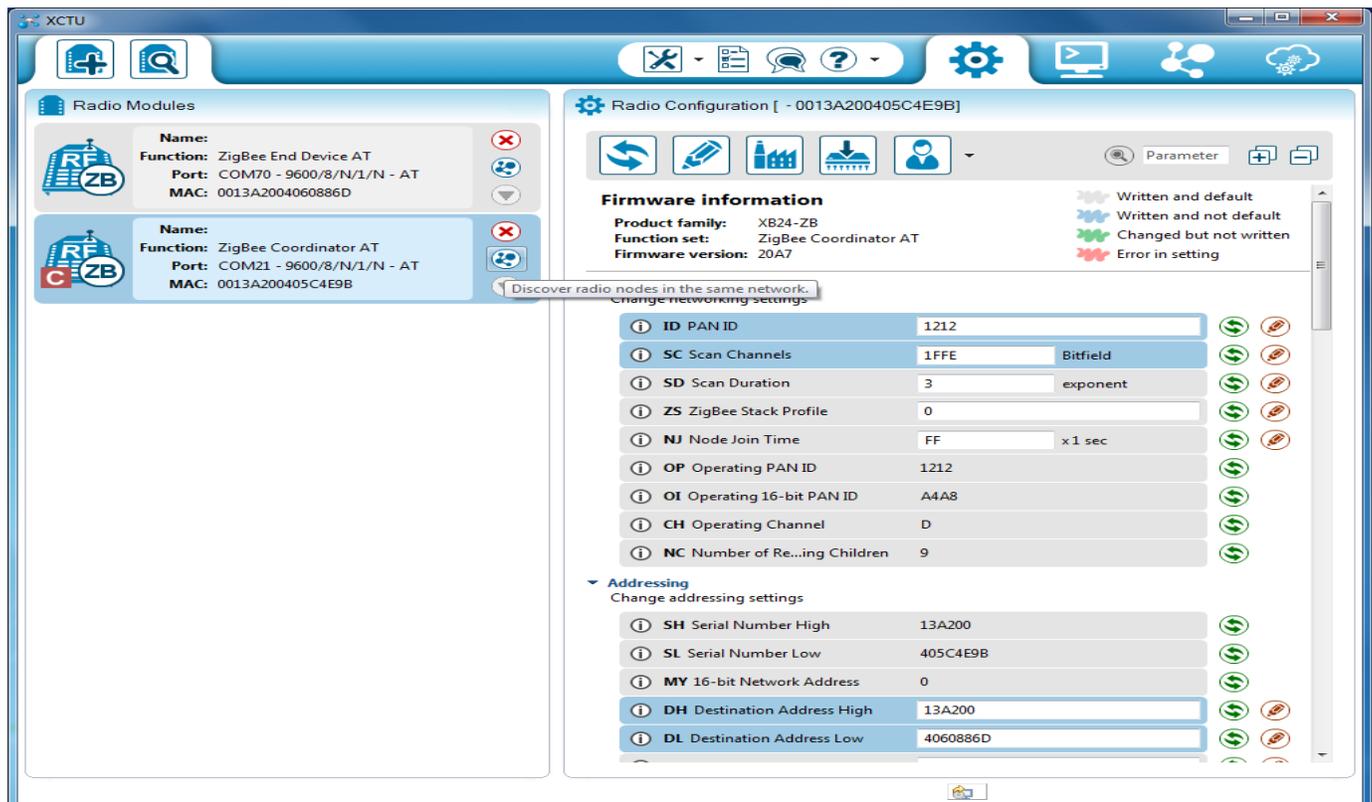
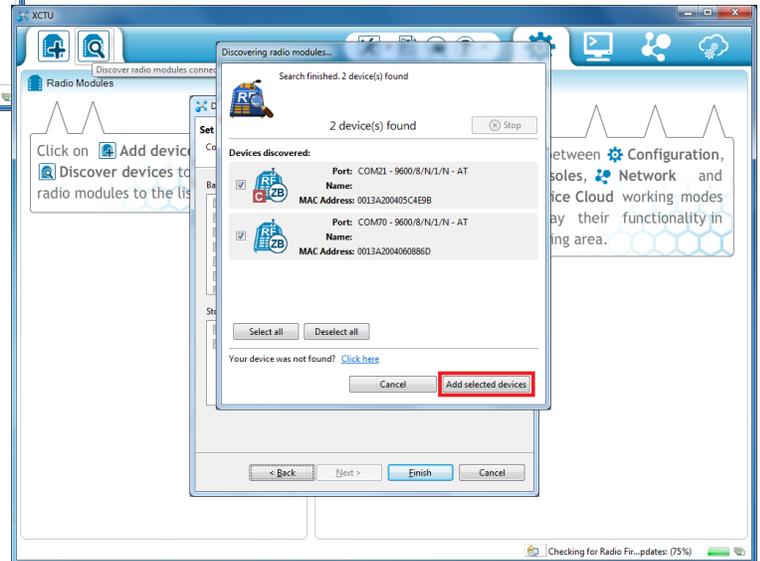
En esta imagen podemos apreciar como utilizaremos el MPC3008 el cual es un convertidor análogo digital, esto debido a que el raspberry no cuenta con entradas análogas requeridas por los sensores, por medio de python, en la raspberry, obtendremos los datos de los sensores y por el mismo archivo procederemos a enviar los valores utilizando el módulo xbee, el cual está configurado como un router, y ya por ultimo en el servidor tendremos corriendo un segundo script de python para, con otro módulo xbee un coordinador, realizar las lecturas de los módulos que envían información y una vez obtenida esta hacer la inserción en la base de datos mediante un plugin de python hacia MySQL.



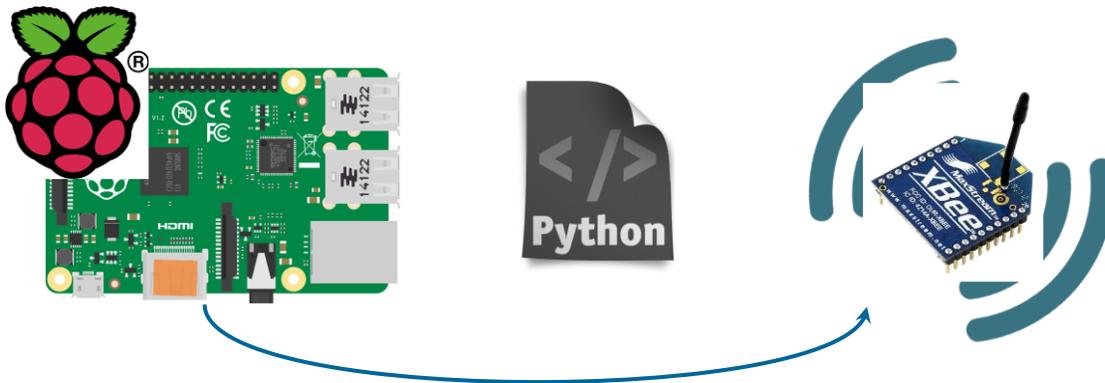
Usaremos el programa XCTU para proceder a la configuración de los módulos Xbee para poder enviar y recibir los datos de los diversos puntos hacia el terminal en tierra.

Una vez abierto el programa procedemos a escanear los módulos conectados a la computadora, presionamos el botón de escanear y dejamos los parámetros por default para que nos despliegue la lista de los dispositivos conectados y agregarlos.

En este punto nos aparecerán los dispositivos que tengamos conectados, como podemos apreciar ya cuentan con una configuración la cual cambiaremos a nuestro gusto.



En la imagen anterior podemos apreciar los parámetros de configuración del módulo de xbee, los parámetros más importantes en la configuración son el PAN ID que es el identificador de la red, si se quiere que todos los dispositivos trabajen juntos y tengan comunicación entre ellos es necesario que se configuren con el mismo PAN ID, otro factor muy importante es el DH y DL que son las direcciones con las cuales se dirá que módulo se conectará con cual otro, mientras que el SH y SL indican las direcciones del módulo actual que se configura. Usualmente la dirección la podemos encontrar al reverso del xbee en la etiqueta que tienen.



Envia datos por Rf a cualquier modulo cerca

Como se mencionaba él envío de los datos será desde la raspberry y python, esto dado a que como no podemos hacer uso del programa X-CTU, antes mencionado, dado a que necesitamos que sea autónomo y en tiempo real él envío de los datos.

```
def writeXbee(valor):
    try:
        ser.open()
    except Exception, e:
        print "error open serial port: " + str(e)
        exit()

    if ser.isOpen():
        try:
            ser.flushInput() #flush input buffer, discarding all its contents
            ser.flushOutput()#flush output buffer, aborting current output

            #write data
            ser.write("insert into sensorHidrogeno (ppm) values (" + valor + ")")

            ser.close()
        except Exception, e1:
            print "error communicating...: " + str(e1)

    else:
        print "cannot open serial port "
```

Mediante el código anterior se realizara él envío de los datos por medio de una comunicación serial, para esto antes debimos de haber configurado el puerto serial y los parámetros necesarios para enviar los datos. Para más información acerca de los códigos ver los anexos de este documento.



Para introducir los datos en la base de datos se cuenta con otro script de python el cual realizara la lectura.

```
def readXbee():
    try:
        ser.open()
    except Exception, e:
        print "error open serial port: " + str(e)
        exit()
    if ser.isOpen():
        try:
            ser.flushInput() #flush input buffer, discarding all its contents
            ser.flushOutput()#flush output buffer, aborting current output

            r = ""
            while True:
                response = ser.readline().strip()
                if response != "":
                    r = response
                    break

            ser.close()
            return r

        except Exception, e1:
            print "error communicating...: " + str(e1)

    else:
        print "cannot open serial port "
```

El código anterior, instalado en el lado del servidor LAMP, ejecutara constantes lecturas por medio de radio frecuencia y regresara el valor leído, que en este caso será una instrucción de sql, como las clases se desarrolló en un modelo basado en objetos solo se creó una clase para regresar el valor de la conexión donde más adelante mediante el uso de otra clase se ejecutara la sentencia de sql en el servidor.

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

import MySQLdb
DB_HOST = 'localhost'
DB_USER = 'root'
DB_PASS = '1234'
DB_NAME = 'residencia'

def conectar():
    datos = [DB_HOST, DB_USER, DB_PASS, DB_NAME]
    conn = MySQLdb.connect(*datos) # Conectar a la base de datos
    return conn

def insert(sql):
    conn = conectar()
    cursor = conn.cursor() # Crear un cursor
    cursor.execute(sql)

    conn.commit() # Hacer efectiva la escritura de datos'
    cursor.close() # Cerrar el cursor
    conn.close() # Cerrar la conexión

```

La imagen anterior hace la conexión con la base de datos de MySQL, la cual es local, y mediante

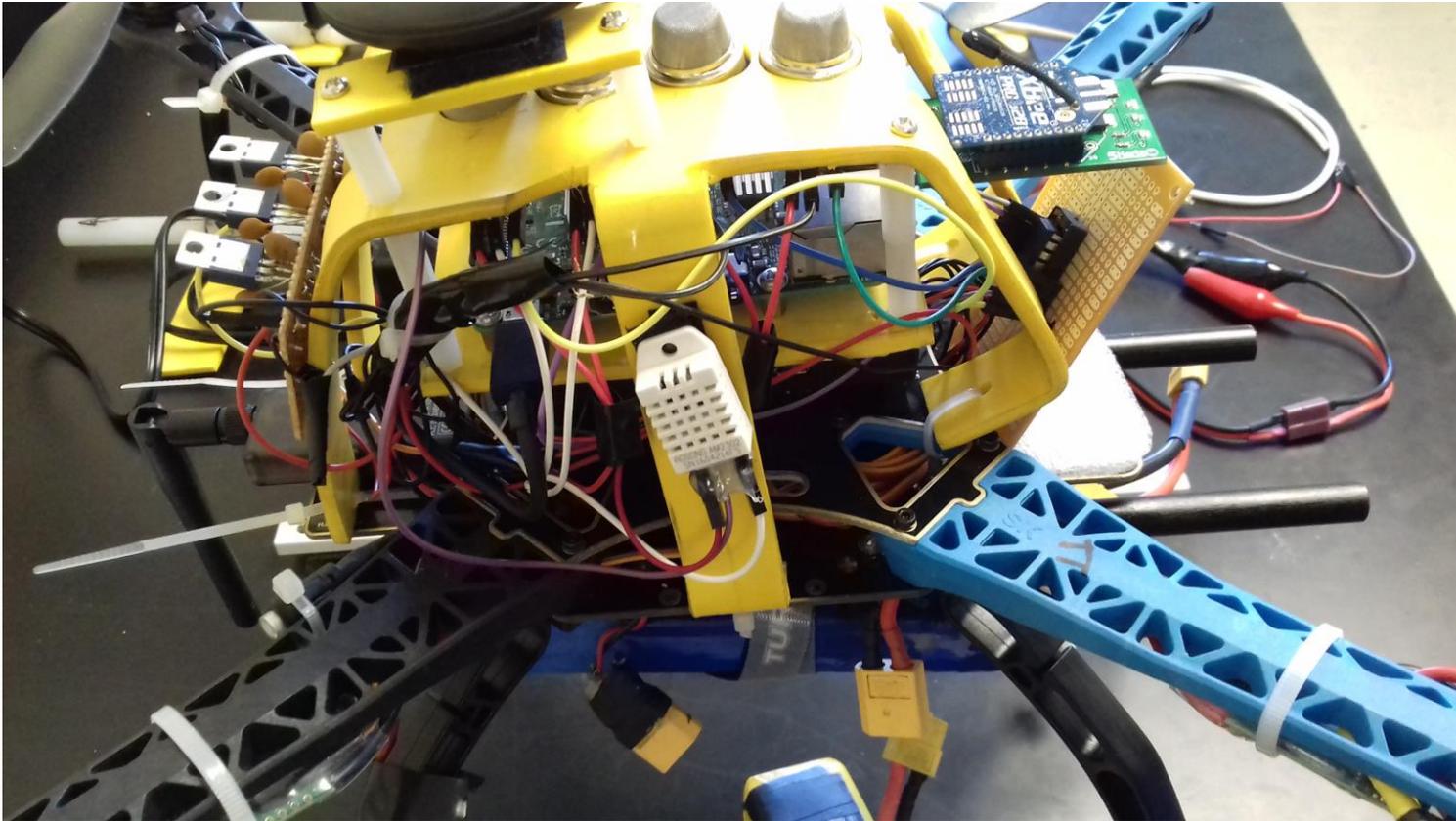


la llamada del método insertar se ejecuta la consulta, obtenida del xbee, en la base de datos.

Ya por último se procedió a la creación de la página web, la cual trabaja bajo php y javascript para poder hacer las consultas y mostrar las gráficas en tiempo real de los datos.

EXPRIMENTACION.

Para esta fase se procederá a montar la tarjeta con los sensores sobre un dron y realizar las pruebas correspondientes. El primer paso fue adaptarle al dron un protector para poder montar el dron, acoplarle los sensores y el módulo de GPS del dron.



Podemos apreciar la raspberry dentro de la estructura montada sobre el dron, así como los sensores, una placa preada para la alimentación, desde una batería, la cual fue creada para poder reducir la tensión y regular la corriente hacia los sensores y el raspberry, y así como también se aprecia el módulo xbee para la comunicación.

Para facilitar la comunicación desde la raspberry hacia la tierra, se creó un script para poder programar que una vez que se encienda el dron se corra automáticamente el Python para el envío de los datos de manera autónoma; el cual fue el siguiente:

```
#!/bin/sh
# /etc/init.d/initXbee

### BEGIN INIT INFO
# Provides:          initXbee
# Required-Start:    $all
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Script de arranque automático
# Description:       Script para arrancar el initXbee
### END INIT INFO
```

```

case "$1" in
start)
    echo "Arrancando initXbee"
    /usr/bin/python /var/www/http/xbee/ini.py
    ;;
stop)
    echo "Deteniendo initXbee"

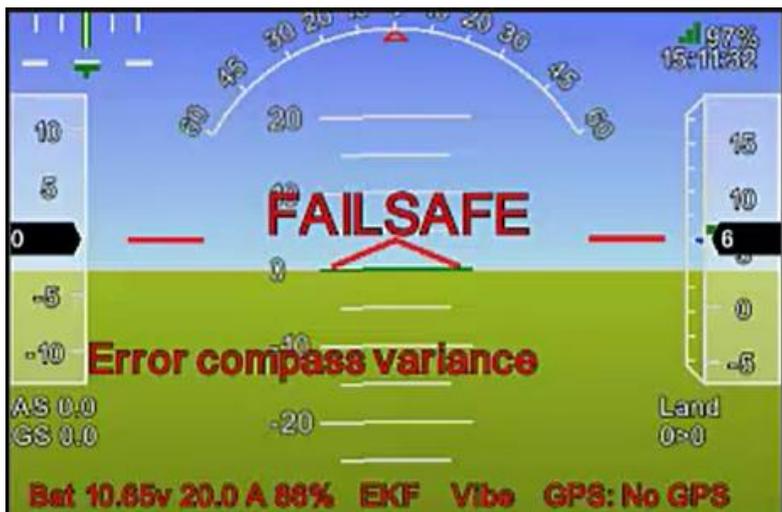
    ;;
*)
    echo "Modo de uso: /etc/init.d/initXbee {start|stop}"
    exit 1
    ;;
esac

exit 0

```

Prosiguiendo con las pruebas, el dron fue llevado al aire libre en el campo del ITNL.

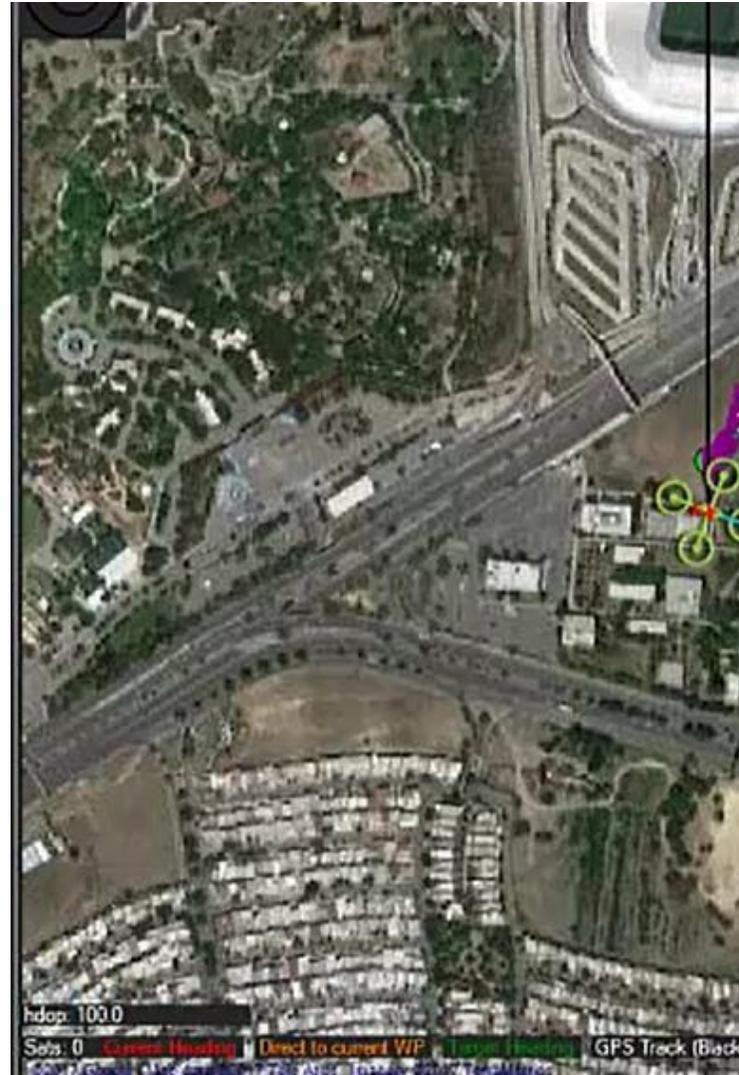
En esta fase se tuvieron algunos problemas relacionado con el funcionamiento del dron, dando a que algunos sensores no funcionaban bien y se tuvo que hacer una calibración manual de los sensores propios del dron.



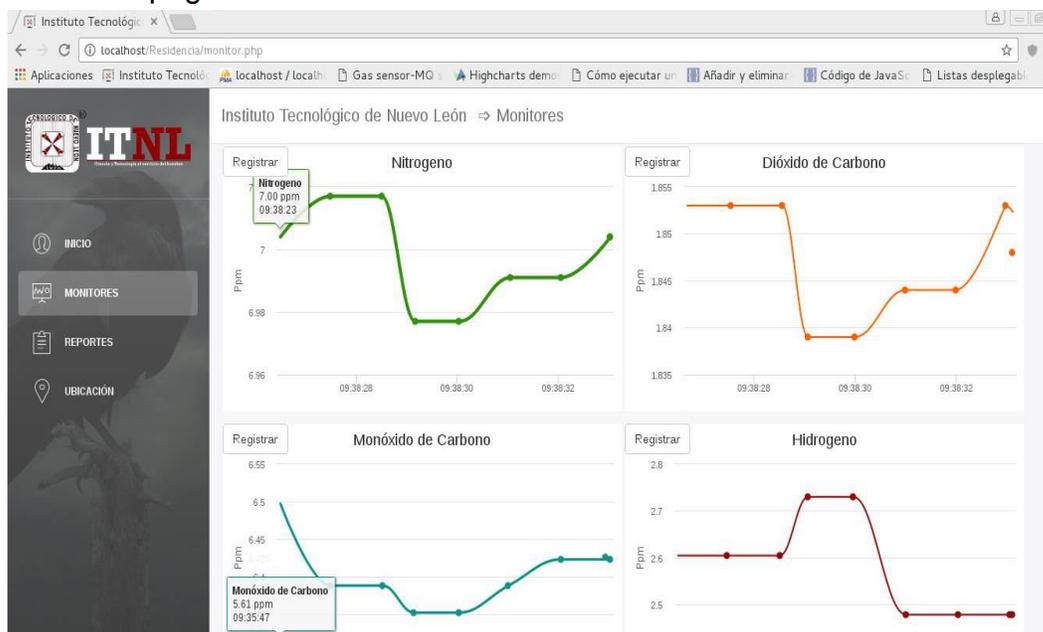
Con la ayuda de un programa llamado mission planner se pudo monitorizar el vuelo del dron y se pudo obtener la trayectoria que siguió el dron desde su



despegue (línea morada dentro del mapa).

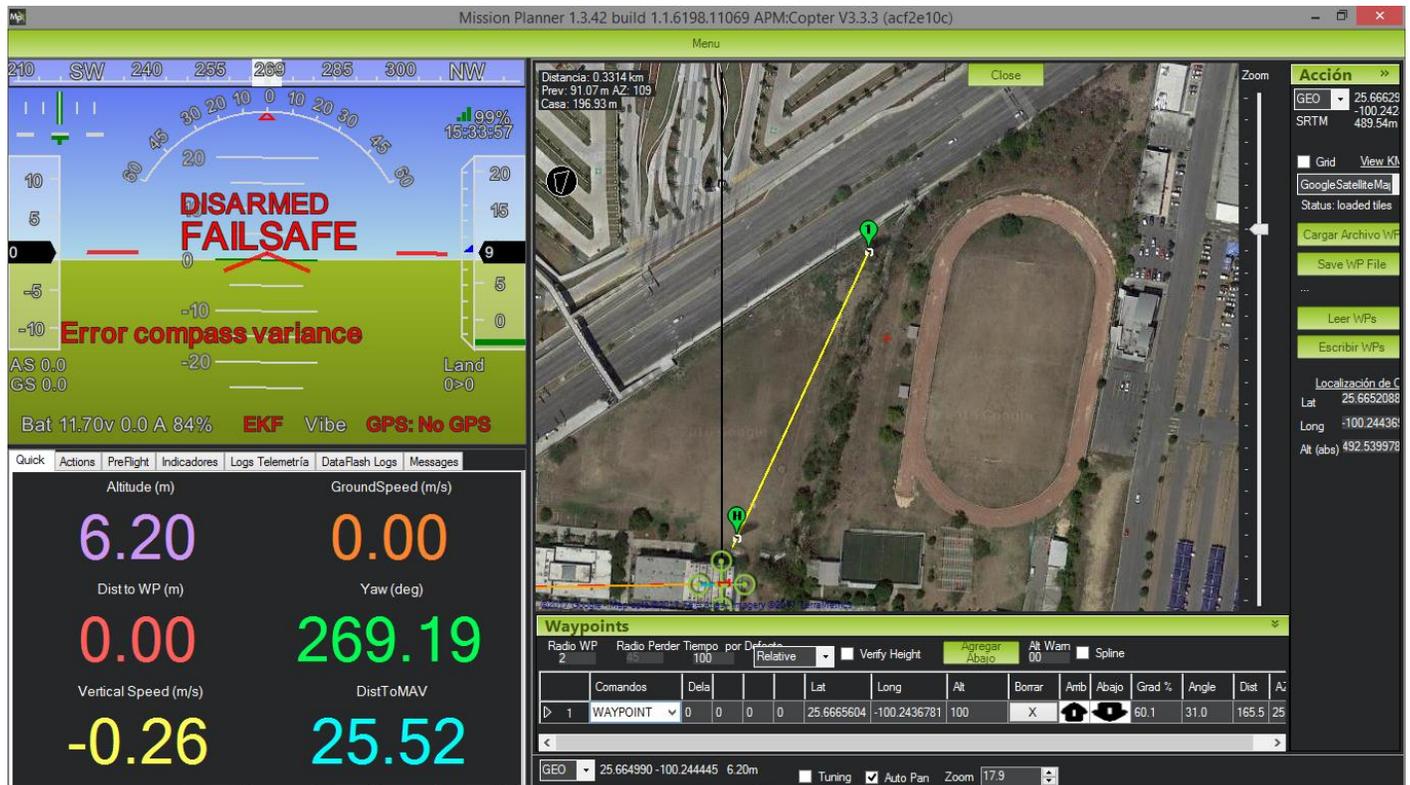


Tomando en cuenta los factores de la des calibración de los sensores del dron, por motivos de seguridad y de la propia integridad del dron, no se sacó fuera de las instalaciones de la institución, aun así con el ron volando se recibieron datos en la terminal y se comenzaron a mostrar los datos en las gráficas de la página web.



En esta imagen se puede apreciar el vuelo del dron junto con el raspberry pi y los sensores enviado los datos por el módulo de xbee.

En la imagen de abajo podemos apreciar la distancia que recorrió el dron, la cual fue de 331m y se voló a una altura de 33m. A esta altura y distancia el módulo de xbee seguía transmitiendo información, esto dado a que como ya se vio anteriormente trabajan a una frecuencia de 2.5GHz y su alcance de 1500m por lo cual el envío y lectura de datos fue satisfactorio.



CONCLUSIÓN.

Con el desarrollo de este proyecto se pudo demostrar que con la integración de varios conocimientos se puede llegar a crear una aplicación sin limitarse a solo un lenguaje o concepto. Además de que personalmente aprendí sobre otros lenguajes de programación y la integración con la comunicación a una base de datos. Hay algunas cosas que puede ser retomadas para el proyecto y mejorarse o incluso ser integradas a otros proyectos con similitud esto gracias a la programación orientada a objetos, por lo cual este proyecto puede aceptar más funciones para

ampliar su capacidad ya sea tanto como agregar más sensores o módulos de comunicación. Existieron algunos problemas desde el inicio del proyecto ya que había incompatibilidad entre algunos lenguajes con la maquina servidor, desde el principio que se desarrolló la base de datos no se podía hacer conexión con php esto dado a que no se había creado un usuario para la conexión y mucho menos tenía los permisos necesarios para el manejo de la base de dato, siguiendo con eso es importante recalcar que siempre hay que tener presente la incompatibilidad. Toda esta aplicación se hizo con recursos accesibles en cuanto precio e información para poder programarlos por lo que se puede llegar a comparar con productos en el mercado con una similitud semejante en el ámbito de trabajo.

COMPETENCIAS APLICADAS Y/O DESARROLLADAS.

Para la realización de este proyecto se tuvieron que hacer uso de los conocimientos adquiridos a lo largo de la carrera de Ingeniería en Sistemas Computacionales, desde los conocimientos de base de datos, los usos de los lenguajes de programación y lo más importante la lógica que se emplea para el desarrollo en cualquier lenguaje de programación. Aun así, con todos los conocimientos que ya tenía, aprendía a usar php, el cual nunca lo había usado pero era requerido para la conexión a la base de datos, y el uso de ajax, para la misma función, además de lo referente a la parte de la microcontroladora no tenía mucho noción del funcionamiento de ello y menos de obtener datos de una forma que no fuera con ventanas, a como este proyecto lo hace mediante la ejecución de comandos en un script de python.

FUENTES DE INFORMACIÓN

Lindsey, H. & Joseph, M.. (2013). *A new eye in the sky: Eco-drones*. Noviembre 20, 2016, de United Nations Environment Programme Sitio web:

https://na.unep.net/geas/getUNEPPageWithArticleIDScript.php?article_id=100

Roberto U.. (2011). *Tutorial sobre circuitos RFID*. Noviembre 20, 2016, de Colección de Tesis Digitales Universidad de las Américas Puebla Sitio web:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/urbina_r_rd/capitulo2.pdf

Lucas I., Pablo G., Osvaldo M., Carlos G. & Cristina P.. (2012). *Estudio de la Integración entre WSN y redes TCP/IP*. Noviembre 20, 2016, de Universidad de Montemorelos Sitio web:
http://www.um.edu.uy/docs/5_estudio_de_la_integracion_entre_WSN_redes%20TCP_IP.pdf

Acosta P. & María C.. (2006). *Estudio del estándar IEEE 802.15.4 ZIGBEE para comunicaciones inalámbricas de área personal de bajo consumo de energía y su comparación en el estándar IEEE 802.15.1 BLUETOOTH*. 2 Noviembre 20, 2016 006, de Escuela Politécnica Nacional Sitio web:
<http://bibdigital.epn.edu.ec/handle/15000/55>

TOMASI W. (2003). *TRANSMISIÓN POR MODULACIÓN DE AMPLITUD*. Sistemas de Comunicaciones Electrónicas, 4, 51-136. 2016, Noviembre 20, De hellsingge Base de datos.

Antón G. (2012). *Desarrollo e implementación de una red de sensores Zigbee mediante el dispositivo Xbee de Digi*. Noviembre 20, 2016, de DEEEA Sitio web:
<http://deeea.urv.cat/public/PROPOSTES/pub/pdf/1957pub.pdf>

Hidalgo M. & Alejandro H.. (2011). *Diseño y construcción de un nodo concentrador en una red de alarmas para hogares utilizando tecnología inalámbrica con el estándar IEEE 802.15.4 ZIGBEE*. Noviembre 20, 2016, de Escuela Politécnica Nacional Sitio web:
<http://bibdigital.epn.edu.ec/handle/15000/3925>

Víctor M.. (2012). *IMPLEMENTACIÓN DE RTU INALÁMBRICA USANDO ZIGBEE*. Noviembre 20, 2016, de Repositorio institucional PIRHUA – Universidad de Piura Sitio web:
https://pirhua.udep.edu.pe/bitstream/handle/123456789/1732/IME_162.pdf?sequence=1&isAllowed=y

Esteban R.. (2008). *Diseño e implementación de un prototipo para una red de domótica y seguridad para un hogar utilizando el estándar IEEE 802.15.4 "ZIGBEE"*. Noviembre 20, 2016, de Escuela Politécnica Nacional Sitio web:
<http://bibdigital.epn.edu.ec/bitstream/15000/938/1/CD-1838%282009-01-21-12-25-09%29.pdf>

ZigBee. (2016). *ZigBee 3.0 The foundation for the internet of things: Launched and open for certifications*. ZigBee, 1, 40. 2016, noviembre 20, De www.ZigBeeResourceGuide.com Base de datos.

Jorge D.. (2011). *ANÁLISIS DEL PROTOCOLO ZIGBEE*. noviembre 20, 2016, de UNLP - Facultad de Informática Sitio web:
http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Redes_y_Seguridad/Trabajos_Finales/Dignanni_Jorge_Pablo.pdf

David G. (2016). *Digi XBee 802.15.4*. noviembre 20, 2016, de Digi XBee Sitio web:
<https://www.digi.com/products/xbee-rf-solutions/modules/xbee-802-15-4#overview>

Jordi O. & Carles B.. (1989). *Sistemas y circuitos digitales y analógicos*. noviembre 20, 2016, de Polytechnic University of New York Sitio web:
https://books.google.com.mx/books?hl=es&lr=lang_es&id=U9bsO5zPmagC&oi=fnd&pg=PA1&dq=conversor+analogo+digital&ots=LWZyn6Wd1y&sig=LAAgraj3Qus7GOz0oyh21bqFnWY#v=onepage&q&f=false

Philip C. (2016). *WELCOME TO THE RASPBERRY PI COMMUNITY*. noviembre 20, 2016, de The official Raspberry Pi magazine Sitio web: <https://www.raspberrypi.org/magpi-issues/MagPi-EduEdition02.pdf>

ANEXOS.

Códigos.

Clase de lectura y envió de datos en el cliente.

```

def writeXbee(valor):
    try:
        ser.open()
    except Exception, e:
        print "error open serial port: " + str(e)
        exit()

    if ser.isOpen():

        try:
            ser.flushInput() #flush input buffer, discarding all its contents
            ser.flushOutput()#flush output buffer, aborting current output

            #write data
            ser.write("insert into sensorHidrogeno (ppm) values (" + valor + ")")

            ser.close()
        except Exception, e1:
            print "error communicating...: " + str(e1)

    else:
        print "cannot open serial port "

```

Clase que realiza la incursión en el servidor.

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

import MySQLdb
DB_HOST = 'localhost'
DB_USER = 'root'
DB_PASS = '1234'
DB_NAME = 'residencia'

def conectar():
    datos = [DB_HOST, DB_USER, DB_PASS, DB_NAME]
    conn = MySQLdb.connect(*datos) # Conectar a la base de datos
    return conn

def insert(sql):
    conn = conectar()
    cursor = conn.cursor() # Crear un cursor
    cursor.execute(sql)

    conn.commit() # Hacer efectiva la escritura de datos'
    cursor.close() # Cerrar el cursor
    conn.close() # Cerrar la conexión

```

Clase ini.py

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  import sensores
5  import mysqlcon
6  import datetime
7  import time
8
9  def fecha():
10     x = datetime.datetime.now()
11     year = ""+ "%s"%x.year
12     month = "%s"%x.month
13     day = "%s"%x.day+""
14     fecha = year+"-"+month+"-"+day
15     return fecha
16
17 def tiempo():
18     x = datetime.datetime.now()
19     hour = ""+ "%s"%x.hour
20     min = "%s"%x.minute
21     sec = "%s"%x.second+""
22     tiempo = hour+":"+min+":"+sec
23     return tiempo
24
25 delay = 1
26
27 while True:
28
29     sensores.humTemp()
30     mysqlcon.insertMonoxidoC(str(sensores.SensorMonoxidoC()))
31     mysqlcon.insertHidrogeno(str(sensores.SensorHidrogeno()))
32     mysqlcon.insertDioxidoC(str(sensores.SensorDioxidoC()))
33     mysqlcon.insertNitrogeno(str(sensores.SensorNitrogeno()))
34
35 # Print out results
36 print ("-----")
37 print ("Lectura mq7: {} ppm".format(sensores.SensorMonoxidoC()))
38 print ("Lectura mq8: {} ppm".format(sensores.SensorHidrogeno()))
39 print ("Lectura mq9: {} ppm".format(sensores.SensorDioxidoC()))
40 print ("Lectura mq135: {} ppm".format(sensores.SensorNitrogeno()))
41
42
43 # Wait before repeating loop
44
45     time.sleep(delay)

```

Programación Servidor

Clase /datosMonitor/dioxidoca.php

```
1 <?php
2 header('Content-Type: application/json');
3 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
4 switch($_GET['Consultar']){
5     // Buscar Último Dato
6     case 1 :
7         $statement=$pdo->prepare("SELECT id as x, ppm as y FROM sensorDioxidoCarbono ORDER BY id DESC LIMIT 0,1");
8         $statement->execute();
9         $results=$statement->fetchAll(PDO::FETCH_ASSOC);
10        $json=json_encode($results);
11        echo $json;
12        break;
13    // Buscar Todos los datos
14    default:
15
16        $statement=$pdo->prepare("SELECT id as x, ppm as y FROM sensorDioxidoCarbono ORDER BY id ASC");
17        $statement->execute();
18        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
19        $json=json_encode($results);
20        echo $json;
21        break;
22    }
23 }
24 ?>
```

Clase /datosMonitor/hidrogeno.php

```
1 <?php
2 header('Content-Type: application/json');
3 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
4 switch($_GET['Consultar']){
5     // Buscar Último Dato
6     case 1 :
7         $statement=$pdo->prepare("SELECT id as x, ppm as y FROM sensorHidrogeno ORDER BY id DESC LIMIT 0,1");
8         $statement->execute();
9         $results=$statement->fetchAll(PDO::FETCH_ASSOC);
10        $json=json_encode($results);
11        echo $json;
12        break;
13    // Buscar Todos los datos
14    default:
15
16        $statement=$pdo->prepare("SELECT id as x, ppm as y FROM sensorHidrogeno ORDER BY id ASC");
17        $statement->execute();
18        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
19        $json=json_encode($results);
20        echo $json;
21        break;
22    }
23 }
24 ?>
```

Clase /datosMonitor/monóxido.php

```
1
2 <?php
3 header('Content-Type: application/json');
4 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
5 switch($_GET['Consultar']){
6     // Buscar Último Dato
7     case 1 :
8         $statement=$pdo->prepare("SELECT ppm as y FROM sensorMonoxido ORDER BY id DESC LIMIT 0,1");
9         $statement->execute();
10        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
11        $json=json_encode($results);
12        echo $json;
13        break;
14        // Buscar Todos los datos
15        default:
16
17        $statement=$pdo->prepare("SELECT id as x, ppm as y FROM sensorMonoxido ORDER BY id ASC");
18        $statement->execute();
19        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
20        $json=json_encode($results);
21        echo $json;
22        break;
23    }
24 }>
```

Clase /datosMonitor/nitrógeno.php

```
1
2 <?php
3 header('Content-Type: application/json');
4 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
5 switch($_GET['Consultar']){
6     // Buscar Último Dato
7     case 1 :
8         $statement=$pdo->prepare("SELECT ppm as y FROM sensorNitrogeno ORDER BY id DESC LIMIT 0,1");
9         $statement->execute();
10        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
11        $json=json_encode($results);
12        echo $json;
13        break;
14        // Buscar Todos los datos
15        default:
16
17        $statement=$pdo->prepare("SELECT id as x, ppm as y FROM sensorNitrogeno ORDER BY id ASC");
18        $statement->execute();
19        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
20        $json=json_encode($results);
21        echo $json;
22        break;
23    }
24 }>
```

Clase /datosMonitor/js/dioxidoca.js

```
1 $(function () {
2   $(document).ready(function () {
3     //tomar hora del servidor local
4     Highcharts.setOptions({
5       global: {
6         useUTC: false
7       }
8     });
9
10    var ultimoy;
11
12    $.ajax({
13      url: "/Residencia/datosMonitor/dioxidoca.php",
14      type: 'get',
15      success: function(datos) {
16        $.each(datos, function(i,o){
17          if (o.y) {datos[i].y = parseFloat(o.y);}
18        });
19        //sety(datos[(datos.length)-1].y);
20        sety(((datos.length !=0)?datos[(datos.length)-1].y: 0));
21
22        $('#dioxidoca').highcharts({
23          chart: {
24            type: 'spline',
25            animation: Highcharts.svg,
26            marginRight: 10,
27            events: {load: function () {
28              series = this.series;
29            }
30            },
31          credits: {
32            enabled: false
33          },
34          title: {text: 'Dióxido de Carbono'},
35          xAxis: {type: 'datetime', tickPixelInterval: 150},
36          yAxis: {title: {text: 'Ppm'},
37            plotLines: [{value: 0, width: 1, color: '#808080'}]
38          },
39          tooltip: {
40            formatter: function () {
41              return '<b>' + this.series.name + '</b><br/>' +
42                Highcharts.numberFormat(this.y, 2) + ' ppm<br/>' +
43                Highcharts.dateFormat('%H:%M:%S', this.x);
44            }
45          },
46          legend: {
47            enabled: false
48          },
49          exporting: {
50            enabled: true
51          },
52          series: [{ name: 'Dióxido de Carbono', color: '#ff5f01', data: (function () {
53            // generate an array of random data
54            var data = [],
55              time = (new Date()).getTime(),
56              i;
57            for (i = -7; i <= 0; i += 1) {
58              data.push({
59                x: time + i * 1000,
60                y: 0
61              });
62            }
63            return data;
64          })()
65          }
66        ]
67      });
68    });
69  });
70
71  setInterval(function () {
72    $.get( "/Residencia/datosMonitor/dioxidoca.php?Consultar=1", function( UltimosDatos ) {
73      var varlocaly=parseFloat((UltimosDatos.length!=0)?UltimosDatos[0].y:0);
74      series[0].addPoint([(new Date()).getTime(), varlocaly], true, true);
75      sety(varlocaly);
76    });
77  }, 1000);
78  function gety(){return ultimoy;}
79  function sety(y){ultimoy=y;}
80
81 });
82
83
84 });
```

Clase /datosMonitor/js/hidrogeno.js

```

1  $(function () {
2  $(document).ready(function () {
3      //tomar hora del servidor local
4      Highcharts.setOptions({
5          global: {
6              useUTC: false
7          }
8      });
9      var ultimoy;
10
11     $.ajax({
12         url: "/Residencia/datosMonitor/hidrogeno.php",
13         type: 'get',
14         success: function(datos) {
15             $.each(datos, function(i,o){
16                 if (o.y) {datos[i].y = parseFloat(o.y);}
17             });
18             //sety(datos[(datos.length)-1].y);
19             sety(((datos.length !=0)?datos[(datos.length)-1].y: 0));
20
21             $('#hidrogeno').highcharts({
22                 chart: {
23                     type: 'spline',
24                     animation: Highcharts.svg,
25                     marginRight: 10,
26                     events: {load: function () {
27                         series = this.series;
28                     }
29                 },
30                 credits: {
31                     enabled: false
32                 },
33                 title: {text: 'Hidrogeno'},
34                 xAxis: {type: 'datetime', tickPixelInterval: 150},
35                 yAxis: {softMax: 1000, title: {text: 'Ppm'},
36                     plotlines: [{value: 0, width: 1, color: '#808080'}]}
37                 },
38                 tooltip: {
39                     formatter: function () {
40                         return '<b>' + this.series.name + '</b><br/>' +
41                             Highcharts.numberFormat(this.y, 2) + ' ppm<br/>' +
42                             Highcharts.dateFormat('%H:%M:%S', this.x);
43                     },
44                 },
45                 legend: {
46                     enabled: false
47                 },
48                 exporting: {
49                     enabled: true
50                 },
51             },
52             series: [{ name: 'Hidrogeno', color: '#930707', data: (function () {
53                 // generate an array of random data
54                 var data = [],
55                     time = (new Date()).getTime(),
56                     i;
57                 for (i = -7; i <= 0; i += 1) {
58                     data.push({
59                         x: time + i * 1000,
60                         y: 0
61                     });
62                 }
63                 return data;
64             })()
65             }]);
66         });
67     });
68 });
69
70     setInterval(function () {
71         $.get( "/Residencia/datosMonitor/hidrogeno.php?Consultar=1", function( UltimosDatos ) {
72             var varlocaly=parseFloat((UltimosDatos.length!=0)?UltimosDatos[0].y:0);
73             series[0].addPoint([(new Date()).getTime(), varlocaly], true, true);
74             sety(varlocaly);
75         });
76     }, 1000);
77
78     function gety(){return ultimoy;}
79     function sety(y){ultimoy=y;}
80
81 });
82
83
84

```

Clase /datosMonitor/js/monóxido.js

```
1 $(function () {
2   $(document).ready(function () {
3     //tomar hora del servidor local
4     Highcharts.setOptions({
5       global: {
6         useUTC: false
7       }
8     });
9
10    var ultimoy;
11
12    $.ajax({
13      url: "/Residencia/datosMonitor/monoxido.php",
14      type: 'get',
15      success: function(datosM) {
16        $.each(datosM, function(i,o){
17          if (o.y) {datosM[i].y = parseFloat(o.y);}
18        });
19
20        //sety(datos[(datos.length)-1].y);
21        sety(((datosM.length !=0)?datosM[(datosM.length)-1].y: 0));
22
23        $('#monoxido').highcharts({
24          chart: {
25            type: 'spline',
26            animation: Highcharts.svg,
27            marginRight: 10,
28            events: {load: function () {
29              series = this.series;
30            }
31            },
32          },
33          credits: {
34            enabled: false
35          },
36          title: {text: 'Monóxido de Carbono'},
37          xAxis: {type: 'datetime', tickPixelInterval: 150},
38          yAxis: {title: {text: 'Ppm'},
39            plotlines: [{value: 0,width: 1,color: '#808080'}]
40          },
41          tooltip: {
42            formatter: function () {
43              return '<b>' + this.series.name + '</b><br/>' +
44                Highcharts.numberFormat(this.y, 2) + ' ppm<br/>' +
45                Highcharts.dateFormat('%H:%M:%S', this.x);
46            }
47          },
48          legend: {
49            enabled: false
50          },
51          exporting: {
52            enabled: true
53          },
54          series: [{ name:'Monóxido de Carbono', color: '#0E928A', data: (function () {
55            // generate an array of random data
56            var data = [],
57                time = (new Date()).getTime(),
58                i;
59
60            for (i = -7; i <= 0; i += 1) {
61              data.push({
62                x: time + i * 1000,
63                y: 0
64              });
65            }
66            return data;
67          })()
68          ]
69        });
70      }
71    });
72
73    setInterval(function () {
74      $.get( "/Residencia/datosMonitor/monoxido.php?Consultar=1", function( UltimosDatos ) {
75        var varlocaly=parseFloat((UltimosDatos.length!=0)?UltimosDatos[0].y:0);
76        series[0].addPoint([(new Date()).getTime(), varlocaly], true, true);
77        sety(varlocaly);
78      });
79    }, 1000);
80
81    function setx(x){ultimox=x;}
82    function sety(y){ultimoy=y;}
83  });
84
85
86 });
```

Clase /datosMonitor/js/nitrógeno.js

```
1 $(function () {
2     $(document).ready(function () {
3         //tomar hora del servidor local
4         Highcharts.setOptions({
5             global: {
6                 useUTC: false
7             }
8         });
9
10
11     var ultimoy;
12
13     $.ajax({
14         url: "/Residencia/datosMonitor/nitrogeno.php",
15         type: 'get',
16         success: function(datos) {
17             $.each(datos, function(i,o){
18                 if (o.y) {datos[i].y = parseFloat(o.y);}
19             });
20
21             sety(((datos.length !=0)?datos[(datos.length)-1].y:0));
22
23             $('#nitrogeno').highcharts({
24                 chart: {
25                     type: 'spline',
26                     animation: Highcharts.svg,
27                     marginRight: 10,
28                     events: {load: function () {
29                         series = this.series;
30                     }
31                 },
32                 credits: {
33                     enabled: false
34                 },
35                 title: {text: 'Nitrogeno'},
36                 xAxis: {type: 'datetime', tickPixelInterval: 150},
37                 yAxis: {title: {text: 'Ppm'},
38                     plotLines: [{value: 0,width: 1,color: '#808080'}]
39                 },
40                 tooltip: {
41                     formatter: function () {
42                         return '<b>' + this.series.name + '</b><br/>' +
43                             Highcharts.numberFormat(this.y, 2) + ' ppm<br/>' +
44                             Highcharts.dateFormat('%H:%M:%S', this.x);
45                     }
46                 },
47                 legend: {
48                     enabled: false
49                 },
50                 // ...
51                 // ...
52                 // ...
53                 // ...
54                 //llenar 6 pts
55                 series: [{ name: 'Nitrogeno', color: '#32920E', lineWidth: 3,data: (function () {
56                     // generate an array of random data
57                     var data = [],
58                         time = (new Date()).getTime(),
59                         i;
60
61                     for (i = -7; i <= 0; i += 1) {
62                         data.push({
63                             x: time + i * 1000,
64                             y: 0
65                         });
66                     }
67                     return data;
68                 })()
69             }
70         });
71     });
72
73     setInterval(function () {
74         $.get( "/Residencia/datosMonitor/nitrogeno.php?consultar=1", function( ultimosDatos ) {
75             var varlocaly=parseFloat((ultimosDatos.length!=0)?ultimosDatos[0].y:0);
76             series[0].addPoint([(new Date()).getTime(), varlocaly], true, true);
77             sety(varlocaly);
78         });
79     }, 1000);
80
81     function gety(){return ultimoy;}
82     function sety(y){ultimoy=y;}
83
84 });
85
```

Class /datosMonitor/js/general.js

```
1 $(function () {
2 $(document).ready(function () {
3 //tomar hora del servidor local
4 Highcharts.setOptions({
5     global: {
6         useUTC: false
7     }
8 });
9
10 var ultimoyD;
11 var ultimoyH;
12 var ultimoyM;
13 var ultimoyN;
14
15 $.ajax({
16 url: "/Residencia/datosMonitor/dioxidoca.php",
17 type: 'get',
18 success:function(datos) {
19     $.each(datos, function(i,o){
20         if (o.y) {datos[i].y = parseFloat(o.y);}
21     });
22     //setyH(datos[(datos.length)-1].y);
23     setyD(((datos.length !=0)?datos[(datos.length)-1].y: 0));
24     setyH(((datos.length !=0)?datos[(datos.length)-1].y: 0));
25     setyM(((datos.length !=0)?datos[(datos.length)-1].y: 0));
26     setyN(((datos.length !=0)?datos[(datos.length)-1].y: 0));
27
28     $('#general').highcharts({
29         chart:{
30             type: 'column',
31             animation: Highcharts.svg,
32             marginRight: 10,
33             events: {load: function () {
34                 series = this.series;
35             }
36             },
37         },
38         credits: {
39             enabled: false
40         },
41         title:{text: 'sensores'},
42         xAxis:{
43             categories: [
44             ],
45             crosshair: true
46         },
47     });
```

```

48     yAxis:{min:0,
49     title: {text:'Ppm'},dioxidoca.js
50     plotLines: [{value: 0,width: 1,color: '#808080'}]
51     },
52     tooltip: {
53     headerFormat: '<span style="font-size:10px"></span><table>',
54     pointFormat: '<tr><td style="color:{series.color};padding:0">{series.name}: </td>' +
55     '<td style="padding:0"><b>{point.y:.1f} ppm</b></td></tr>',
56     footerFormat: '</table>',
57     useHTML: true
58     },
59     legend: {
60     enabled: true
61     },
62     exporting: {
63     enabled: true
64     },
65     series: [
66     { name:'Nitrogeno', color: '#32920E', data: [getyN()] },
67     { name:'Dióxido de Carbono', color: '#ff5f01', data: [getyD()] },
68     { name:'Monóxido de Carbono', color: '#0E928A', data: [getyM()] },
69     { name:'Hidrógeno', color: '#930707', data:[getyH()] }
70     ]
71     },
72     });
73     }
74     });
75     });
76     });
77     });
78     setInterval(function () {
79     $.get( "/Residencia/datosMonitor/nitrogeno.php?Consultar=1", function( MultimosDatos ) {
80     var varlocalyN=parseFloat((MultimosDatos.length!=0)?MultimosDatos[0].y:0);
81     series[0].addPoint([0, varlocalyN], true, true);
82     setyN(varlocalyN);
83     });
84     $.get( "/Residencia/datosMonitor/dioxidoca.php?Consultar=1", function( DultimosDatos ) {
85     var varlocalyD=parseFloat((DultimosDatos.length!=0)?DultimosDatos[0].y:0);
86     series[1].addPoint([0, varlocalyD], true, true);
87     setyD(varlocalyD);
88     });
89     $.get( "/Residencia/datosMonitor/monoxido.php?Consultar=1", function( MultimosDatos ) {
90     var varlocalyM=parseFloat((MultimosDatos.length!=0)?MultimosDatos[0].y:0);
91     series[2].addPoint([0, varlocalyM], true, true);
92     setyM(varlocalyM);
93     });
94     $.get( "/Residencia/datosMonitor/hidrogeno.php?Consultar=1", function( MultimosDatos ) {
95     var varlocalyH=parseFloat((HultimosDatos.length!=0)?MultimosDatos[0].y:0);
96     series[3].addPoint([0, varlocalyH], true, true);
97     setyH(varlocalyH);
98     });
99     });
100     });
101     }, 1000);
102     function getyD(){return ultimoyD;}
103     function setyD(y){ultimoyD=y;}
104     function getyH(){return ultimoyH;}
105     function setyH(y){ultimoyH=y;}
106     function getyM(){return ultimoyM;}
107     function setyM(y){ultimoyM=y;}
108     function getyN(){return ultimoyN;}
109     function setyN(y){ultimoyN=y;}
110     });
111     });
112     });
113     });
114     });

```

Clase /grafica/dioxidoca.php

```
1 <?php
2 $fecha = date("-Y-m-d-g:ia");
3 $parte1 = "select * from sensorDioxidoCarbono into outfile ";
4 $parte2 = ".txt' FIELDS TERMINATED BY ',' LINES TERMINATED BY '$\n'";
5 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
6
7 if(isset($_POST["dioxido"])){
8     $value = "Diox";
9     $sql = "$parte1$value$parte2";
10    $statement=$pdo->prepare($sql);
11    $statement->execute();
12    $ssss = ("cp /home/redes/Escritorio/Residencia/index.html /home/redes/Escritorio/");
13    $salida = exec($ssss);
14    echo $ssss;
15    echo "<pre>$salida</pre>";
16    $statement2=$pdo->prepare("TRUNCATE TABLE sensorDioxidoCarbono");
17    $statement2->execute();
18 }
19 ?>
20
21 <!DOCTYPE html>
22 <html>
23 <head>
24 <meta charset="utf-8">
25 <title></title>
26 <link href="../assets/css/bootstrap.min.css" rel="stylesheet" />
27
28 </head>
29 <body style="margin:0">
30 <form class="" action="<?=$_SERVER['PHP_SELF']?>" method="post">
31 | | <input type="submit" name="dioxido" class="btn btn-default" value="Registrar" style="position:absolute;z-index:1;">
32 </form>
33 <div id="dioxidoca" style="height:290px"></div>
34 <script src="../assets/js/jquery-2.1.4.js"></script>
35 <script src="../highcharts/js/highcharts.js"></script>
36 <script src="../highcharts/js/modules/data.js"></script>
37 <script src="../datosMonitor/js/dioxidoca.js"></script>
38 </body>
39 </html>
```

Clase /grafica/hidrogeno.php

```
1 <?php
2 $time = time();
3 $fecha = date("-Y-m-d g:ia");
4 $parte1 = "select * from sensorDioxidoCarbono into outfile ";
5 $parte2 = ".txt' FIELDS TERMINATED BY ',' LINES TERMINATED BY '$\n'";
6 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
7
8 if(isset($_POST["hidrogeno"])){
9     $value = "H";
10    $sql = "$parte1$value$fecha$parte2";
11    $statement=$pdo->prepare($sql);
12    $statement->execute();
13    $statement2=$pdo->prepare("TRUNCATE TABLE sensorHidrogeno");
14    $statement2->execute();
15 }
16 ?>
17
18 <!DOCTYPE html>
19 <html>
20 <head>
21 <meta charset="utf-8">
22 <title></title>
23 <link href="../assets/css/bootstrap.min.css" rel="stylesheet" />
24
25 </head>
26 <body style="margin:0">
27 <form class="" action="<?=$_SERVER['PHP_SELF']?>" method="post">
28 | | <input type="submit" name="hidrogeno" class="btn btn-default" value="Registrar" style="position:absolute;z-index:1;">
29 </form>
30 <div id="hidrogeno" style="height:290px"></div>
31 <script src="../assets/js/jquery-2.1.4.js"></script>
32 <script src="../highcharts/js/highcharts.js"></script>
33 <script src="../highcharts/js/modules/data.js"></script>
34 <script src="../datosMonitor/js/hidrogeno.js"></script>
35 </body>
36 </html>
```

Clase /grafica/monóxido.php

```
1 <?php
2 $time = time();
3 $fecha = date("-Y-m-d g:ia");
4 $parte1 = "select * from sensorDioxidoCarbono into outfile ";
5 $parte2 = ".txt' FIELDS TERMINATED BY ',' LINES TERMINATED BY '$\n'";
6 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
7
8 if(isset($_POST["monoxido"])){
9     $value = "M";
10    $sql = "$parte1$value$fecha$parte2";
11    $statement=$pdo->prepare($sql);
12    $statement->execute();
13    $statement2=$pdo->prepare("TRUNCATE TABLE sensorMonoxido");
14    $statement2->execute();
15 }
16 ?>
17
18 <!DOCTYPE html>
19 <html>
20 <head>
21     <meta charset="utf-8">
22     <title></title>
23     <link href="../assets/css/bootstrap.min.css" rel="stylesheet" />
24
25 </head>
26 <body style="margin:0">
27     .....
28     <form class="" action="<?=$_SERVER['PHP_SELF']>" method="post">
29         <input type="submit" name="monoxido" class="btn btn-default" value="Registrar" style="position:absolute;z-index:1;">
30     </form>
31
32     <div id="monoxido" style="height:290px"></div>
33     <script src="../assets/js/jquery-2.1.4.js"></script>
34     <script src="../highcharts/js/highcharts.js"></script>
35     <script src="../highcharts/js/modules/data.js"></script>
36     <script src="../datosMonitor/js/monoxido.js"></script>
37 </body>
38 </html>
```

Clase /grafica/nitrogeno.php

```
1 <?php
2 $time = time();
3 $fecha = date("-Y-m-d g:ia");
4 $parte1 = "select * from sensorDioxidoCarbono into outfile ";
5 $parte2 = ".txt' FIELDS TERMINATED BY ',' LINES TERMINATED BY '$\n'";
6 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
7
8 if(isset($_POST["nitrogeno"])){
9     $value = "N";
10    $sql = "$parte1$value$fecha$parte2";
11    $statement=$pdo->prepare($sql);
12    $statement->execute();
13    $statement2=$pdo->prepare("TRUNCATE TABLE sensorNitrogeno");
14    $statement2->execute();
15 }
16 ?>
17
18 <!DOCTYPE html>
19 <html>
20 <head>
21     <meta charset="utf-8">
22     <title></title>
23     <link href="../assets/css/bootstrap.min.css" rel="stylesheet" />
24
25 </head>
26 <body style="margin:0">
27     <form class="" action="<?=$_SERVER['PHP_SELF']>" method="post">
28         <input type="submit" name="nitrogeno" class="btn btn-default" value="Registrar" style="position:absolute;z-index:1;">
29     </form>
30
31     <div id="nitrogeno" style="height:290px"></div>
32     <script src="../assets/js/jquery-2.1.4.js"></script>
33     <script src="../highcharts/js/highcharts.js"></script>
34     <script src="../highcharts/js/modules/data.js"></script>
35     <script src="../datosMonitor/js/nitrogeno.js"></script>
36 </body>
37 </html>
```

Class index.html

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8" />
5   <link rel="icon" type="image/png" href="assets/img/itnl.ico">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
7   <title>Instituto Tecnológico de Nuevo León</title>
8   <meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0' name='viewport' />
9   <meta name="viewport" content="width=device-width" />
10  <link href="assets/css/bootstrap.min.css" rel="stylesheet" />
11  <link href="assets/css/animate.min.css" rel="stylesheet"/>
12  <link href="assets/css/light-bootstrap-dashboard.css" rel="stylesheet"/>
13  <link href="assets/css/pe-icon-7-stroke.css" rel="stylesheet" />
14  <link href="assets/css/estilo.css" rel="stylesheet" />
15 </head>
16 <body>
17
18   <div class="wrapper">
19     <div class="row">
20       .....
21     </div>
22
23     <div class="sidebar" data-color="gray" data-image="assets/img/fondo.jpg">
24
25       <div class="sidebar-wrapper">
26         <div class="logo">
27           <a href="index.php" class="simple-text">
28             ..... 
29           </a>
30         </div>
31
32         <ul class="nav">
33           <li class="active">
34             <a href="index.php">
35               <i class="pe-7s-user"></i>
36               <p>Inicio</p>
37             </a>
38           </li>
39           <li>
40             <a href="monitor.php">
41               <i class="pe-7s-display1"></i>
42               <p>Monitores</p>
43             </a>
44           </li>
45           <li style="display:none">
46             <a href="reporte.php">
47               <i class="pe-7s-note2"></i>
48               <p>Reportes</p>
49             </a>
50           </li>
51           <li>
52             <a href="ubicacion.php">
53               <i class="pe-7s-map-marker"></i>
54               <p>Ubicación</p>
55             </a>
56           </li>

```

```

56         </li>
57     </ul>
58 </div>
59 </div>
60
61 <div class="main-panel">
62     <nav class="navbar navbar-default navbar-fixed">
63         <div class="container-fluid">
64             <div class="navbar-header">
65                 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navigation-example-2">
66                     <span class="sr-only">Toggle navigation</span>
67                     <span class="icon-bar"></span>
68                     <span class="icon-bar"></span>
69                     <span class="icon-bar"></span>
70                 </button>
71                 <a class="navbar-brand" href="index.php">Instituto Tecnológico de Nuevo León</a>
72             </div>
73             <div class="collapse navbar-collapse">
74             </div>
75         </div>
76     </nav>
77     <div class="content">
78         <div class="container-fluid">
79             <div class="row">
80                 <div class="titulo col-md-12">
81                     <h4>Proyecto: Red de Comunicación por tecnología WIFI entre un Vant y una Computadora.</h4>
82                 </div>
83             </div>
84             <div class="row">
85                 <div class="objetivo col-md-6">
86                     <h3>Objetivo</h3>
87                     <p class="parrafo">
88                         Diseño e implementación de una red mediante tecnología Wifi entre un VANT y una computadora
89                         para el proyecto de investigación "Monitoreo de la calidad del aire por VANTS" desarrollado
90                         en el Instituto Tecnológico de Nuevo León.
91                     </p>
92                 </div>
93                 <div class="justificacion col-md-6">
94                     <h3>Justificación</h3>
95                     <p class="parrafo">
96                         Este proyecto se desarrolla debido a la necesidad de establecer una comunicación entre un módulo
97                         de telecomunicación que se integrará a un VANT que le permita enviar datos que se estan monitoreando
98                         en tiempo real mediante su tarjeta de adquisición de datos. <br>
99                         Con esta Red de comunicación se generará una plataforma de experimentación para la prueba de envío,
100                         recepción de datos, tratamiento de la información en una computadora de propósito general.<br>
101                         La integración entre sistemas embebidos con aplicaciones de alto nivel permite procesar y generar
102                         información con mayor grado de interpretación entre el usuario de este tipo de aplicaciones.
103                     </p>
104                 </div>
105             </div>
106         </div>
107     </div>
108 </div>
109 </div>
110
111
112 <script src="assets/js/jquery-1.10.2.js" type="text/javascript"></script>
113 <script src="assets/js/bootstrap.min.js" type="text/javascript"></script>
114 <script src="assets/js/bootstrap-checkbox-radio-switch.js"></script>
115 <script src="assets/js/chartist.min.js"></script>
116 <script src="assets/js/bootstrap-notify.js"></script>
117 <script src="assets/js/light-bootstrap-dashboard.js"></script>
118
119 </body>
120
121 <!-- Core JS Files -->
122
123
124 </html>

```

Clase monitor.php

```
1
2 <!doctype html>
3 <html lang="en">
4 <head>
5   <meta charset="utf-8" />
6   <link rel="icon" type="image/png" href="assets/img/itnl.ico">
7   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
8   <title>Instituto Tecnológico de Nuevo León</title>
9   <meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0' name='viewport' />
10  <meta name="viewport" content="width=device-width" />
11  <link href="assets/css/bootstrap.min.css" rel="stylesheet" />
12  <link href="assets/css/animate.min.css" rel="stylesheet"/>
13  <link href="assets/css/light-bootstrap-dashboard.css" rel="stylesheet"/>
14  <link href="assets/css/pe-icon-7-stroke.css" rel="stylesheet" />
15  <link href="assets/css/estilo.css" rel="stylesheet" />
16 </head>
17 <body scrolling="no">
18
19   <div class="wrapper">
20     <div class="row">
21       .....
22     </div>
23
24   <div class="sidebar" data-color="gray" data-image="assets/img/fondo.jpg">
25
26     <div class="sidebar-wrapper">
27       <div class="logo">
28         <a href="index.php" class="simple-text">
29           ..... 
30         </a>
31       </div>
32
33       <ul class="nav">
34         <li>
35           <a href="index.php">
36             <i class="pe-7s-user"></i>
37             <p>Inicio</p>
38           </a>
39         </li>
40         <li class="active">
41           <a href="monitor.php">
42             <i class="pe-7s-display1"></i>
43             <p>Monitores</p>
44           </a>
45         </li>
46         <li style="display:none">
47           <a href="reporte.php">
48             <i class="pe-7s-note2"></i>
49             <p>Reportes</p>
50           </a>
51         </li>
52         <li>
53           <a href="ubicacion.php">
54             <i class="pe-7s-map-marker"></i>
55             <p>Ubicación</p>
56           </a>

```

```

57     </li>
58 </ul>
59 </div>
60 </div>
61
62 <div class="main-panel" style="overflow-y:hidden">
63 <nav class="navbar navbar-default navbar-fixed" style="height:10%">
64 <div class="container-fluid">
65 <div class="navbar-header">
66 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navigation-example-2">
67 <span class="sr-only">Toggle navigation</span>
68 <span class="icon-bar"></span>
69 <span class="icon-bar"></span>
70 <span class="icon-bar"></span>
71 </button>
72 <a class="navbar-brand" href="index.php">Instituto Tecnológico de Nuevo León</a>
73 <a class="navbar-brand" href="monitor.php"> Monitores</a>
74 </div>
75 <div class="collapse navbar-collapse">
76 </div>
77 </div>
78 </nav>
79
80
81 <div class="col-md-12" style="max-height:90%; overflow-y: auto;width:100%;">
82 <div class="col-md-12">
83 <div class="">
84 </div>
85 </div>
86 <div class="col-md-12 row">
87 <div class="col-md-6 alturaR" style="padding:0;height:100%">
88 <iframe scrolling="no" src="grafica/nitrogeno.php" height="100%" width="100%"></iframe>
89 </div>
90 <div class="col-md-6 alturaR" style="padding:0;height:100%">
91 <iframe scrolling="no" src="grafica/dioxidoca.php" height="100%" width="100%"></iframe>
92 </div>
93 </div>
94 <div class="col-md-12 row">
95 <div class="col-md-6 alturaR" style="padding:0;height:100%">
96 <iframe scrolling="no" src="grafica/monoxido.php" height="100%" width="100%"></iframe>
97 </div>
98 <div class="col-md-6 alturaR" style="padding:0;height:100%">
99 <iframe scrolling="no" src="grafica/hidrogeno.php" height="100%" width="100%"></iframe>
100 </div>
101 </div>
102
103 <div class="col-md-12 row">
104 <div class="col-md-12 alturaR" style="padding:0;height:100%">
105 <iframe scrolling="no" src="grafica/general.php" height="100%" width="100%"></iframe>
106 </div>
107 </div>
108
109 </div>
110
111 </div>
112
113 </div>
114
115 <script src="assets/js/jquery-2.1.4.js"></script>
116 <script src="assets/js/bootstrap.min.js" type="text/javascript"></script>
117 <script src="assets/js/bootstrap-checkbox-radio-switch.js"></script>
118 <script src="assets/js/chartist.min.js"></script>
119 <script src="assets/js/bootstrap-notify.js"></script>
120 <script src="assets/js/light-bootstrap-dashboard.js"></script>
121 <script src="highcharts/js/highcharts.js"></script>
122 <script src="highcharts/js/modules/data.js"></script>
123 <script src="highcharts/js/modules/drilldown.js"></script>
124 </body>
125
126 </html>

```

Clase ubicación.php

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8" />
5 <link rel="icon" type="image/png" href="assets/img/itnl.ico">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
7 <title>Instituto Tecnológico de Nuevo León</title>
8 <meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0' name='viewport' />
9 <meta name="viewport" content="width=device-width" />
10 <link href="assets/css/bootstrap.min.css" rel="stylesheet" />
11 <link href="assets/css/animate.min.css" rel="stylesheet"/>
12 <link href="assets/css/light-bootstrap-dashboard.css" rel="stylesheet"/>
13 <link href="assets/css/pe-icon-7-stroke.css" rel="stylesheet" />
14 <link href="assets/css/estilo.css" rel="stylesheet" />
15 </head>
16 <body>
17 <div class="wrapper">
18 <div class="row">
19 </div>
20 </div>
21 <div class="sidebar" data-color="gray" data-image="assets/img/fondo.jpg">
22 <div class="sidebar-wrapper">
23 <div class="logo">
24 <a href="index.php" class="simple-text">
25 
26 </a>
27 </div>
28 <ul class="nav">
29 <li>
30 <a href="index.php">
31 <i class="pe-7s-user"></i>
32 <p>Inicio</p>
33 </a>
34 </li>
35 <li>
36 <a href="monitor.php">
37 <i class="pe-7s-display1"></i>
38 <p>Monitores</p>
39 </a>
40 </li>
41 <li class="active">
42 <a href="ubicacion.php">
43 <i class="pe-7s-map-marker"></i>
44 <p>Ubicación</p>
45 </a>
46 </li>
47 </ul>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div>
```

```

56 <div class="main-panel" style="overflow-y:hidden">
57 <nav class="navbar navbar-default navbar-fixed" style="height:10%">
58 <div class="container-fluid">
59 <div class="navbar-header">
60 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navigation-example-2">
61 <span class="sr-only">Toggle navigation</span>
62 <span class="icon-bar"></span>
63 <span class="icon-bar"></span>
64 <span class="icon-bar"></span>
65 </button>
66 <a class="navbar-brand" href="index.php">Instituto Tecnológico de Nuevo León</a>
67 <a class="navbar-brand" href="monitor.php"> Monitores</a>
68 </div>
69 <div class="collapse navbar-collapse">
70 </div>
71 </div>
72 </nav>
73
74
75 <div class="col-md-12" style="max-height:90%;width:100%;">
76 <div class="col-md-12 row">
77 <div class="col-md-12" style="padding:0; height:490px">
78 <iframe scrolling="no" src="googleMap/map.html" height="100%" width="100%"></iframe>
79 </div>
80 </div>
81 </div>
82
83 </div>
84
85 </div>
86
87 <script src="assets/js/jquery-2.1.4.js"></script>
88 <script src="assets/js/bootstrap.min.js" type="text/javascript"></script>
89 <script src="assets/js/bootstrap-checkbox-radio-switch.js"></script>
90 <script src="assets/js/light-bootstrap-dashboard.js"></script>
91 </body>
92
93
94 </html>

```

Class /googleMap/map.html

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <div id="map" style="width:100%;height:500px"></div>
5
6 <script src="../assets/js/jquery-2.1.4.js"></script>
7 <script src="map.js"></script>
8 <script async defer src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDDxR608vDvhaM1c4JTv1HzJd6Wcoz9iGU&signed_in=true&callback=myMap"></script>
9
10 </body>
11 </html>
```

Class /googleMap/map.js

```
1 function myMap() {
2   var mapCanvas = document.getElementById("map");
3   var pos = new google.maps.LatLng(25.6646376, -100.2444697);
4   var mapOptions = {
5     center: new google.maps.LatLng(25.6646376, -100.2444697),
6     zoom: 18
7   }
8   map = new google.maps.Map(mapCanvas, mapOptions);
9   var image = 'dron1.png';
10  marker = new google.maps.Marker({
11    position: pos,
12    map: map,
13    title: "Dron WIFI",
14    icon: image,
15    animation: google.maps.Animation.DROP
16  });
17  //move(map, marker);
18 }
19 function prueba(lat, Long)
20 {
21   var geo_lng = Long; //longg+0.0001;
22   var geo_lat = lat; //latt+0.0001;
23   longg = geo_lng;
24   latt = geo_lat;
25
26   var myLatLng = {lat: geo_lat, lng: geo_lng};
27
28   var newLatLng = new google.maps.LatLng(geo_lat, geo_lng);
29   marker.setPosition(newLatLng);
30
31   map.setCenter(marker.getPosition());
32 }
33 $.ajax({
34   url: "map.php",
35   type: 'get',
36   success: function(datos) {
37     $.each(datos, function(i,o){
38       if (o.lat) {datos[i].lat = parseFloat(o.lat);}
39       if (o.lon) {datos[i].lon = parseFloat(o.lon);}
40     });
41   }
42 });
43 setInterval(function () {
44   $.get( "map.php?Consultar=1", function( UltimosDatos ) {
45     var varlocalat=parseFloat((UltimosDatos.length!=0)?UltimosDatos[0].lat:25.6646376);
46     var varlocalon=parseFloat((UltimosDatos.length!=0)?UltimosDatos[0].lon:-100.2444697);
47     prueba(varlocalat,varlocalon);
48   });
49 }, 1000);
```

Clase /googleMap/map.php

```
1
2 <?php
3 header('Content-Type: application/json');
4 $pdo=new PDO("mysql:dbname=residencia;host=127.0.0.1","root","1234");
5 switch($_GET['Consultar']){
6     // Buscar Último Dato
7     case 1 :
8         $statement=$pdo->prepare("SELECT * FROM gps ORDER BY id DESC LIMIT 0,1");
9         $statement->execute();
10        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
11        $json=json_encode($results);
12        echo $json;
13    break;
14    // Buscar Todos los datos
15    default:
16
17        $statement=$pdo->prepare("SELECT * FROM gps ORDER BY id ASC");
18        $statement->execute();
19        $results=$statement->fetchAll(PDO::FETCH_ASSOC);
20        $json=json_encode($results);
21        echo $json;
22    break;
23 }
24 ?>
```