



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIMODAL
QUE PERMITA MONITOREAR Y CONTROLAR LA
RECOLECCIÓN DE DESECHOS SÓLIDOS EN TIEMPO REAL
PARA SMART CITIES.”

Trabajo de titulación presentado para optar al grado académico de:
INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y
REDES

AUTORES: VANESSA ELENA MOROCHO SAILEMA
JONATHAN ISMAEL QUINAPANTA PÁEZ
TUTOR: ING. OSWALDO MARTÍNEZ

Riobamba-Ecuador

2016

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

El Tribunal del Trabajo de Titulación certifica que el proyecto técnico: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIMODAL QUE PERMITA MONITOREAR Y CONTROLAR LA RECOLECCIÓN DE DESECHOS SÓLIDOS EN TIEMPO REAL PARA SMART CITIES, de responsabilidad de la señorita Vanessa Elena Morocho Sailema y el señor Jonathan Ismael Quinapanta Páez, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Dr. Miguel Tasambay Ph.D DECANO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Franklin Moreno DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA, TELECOMUNICACIONES Y REDES	_____	_____
Ing. Oswaldo Martínez DIRECTOR TRABAJO DE TITULACIÓN	_____	_____
Ing. Wilson Baldeón MIEMBRO TRIBUNAL	_____	_____

RESPONSABILIDAD DE AUTORES

Nosotros, Vanessa Elena Morocho Sailema y Jonathan Ismael Quinapanta Páez declaramos ser los autores del presente trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIMODAL QUE PERMITA MONITOREAR Y CONTROLAR LA RECOLECCIÓN DE DESECHOS SÓLIDOS EN TIEMPO REAL PARA SMART CITIES”, que fue elaborado en su totalidad por nosotros, bajo la dirección del Ingeniero Oswaldo Martínez, haciéndonos totalmente responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

Vanessa Elena Morocho Sailema

Jonathan Ismael Quinapanta Páez

DEDICATORIA

A Dios por darme un día más de vida, brindándome la paz y fuerzas necesarias para seguir adelante. A mis padres Rosa y Jesús que han sido mi apoyo incondicional día a día y sin ellos no podría haber alcanzado este logro académico. A mi hermano que siempre ha estado atento a mi avance en el trabajo noche a noche. A todos ellos dedico el presente Trabajo Técnico que con sacrificio y esmero lo he culminado. Un versículo que siempre me ha acompañado: La esperanza que se demora es tormento del corazón, pero árbol de vida es el deseo cumplido. Proverbios 13:12

Jonathan

A mi padre del cielo, por haberme llenado de la fuerza espiritual que necesitaba para terminar tan arduo recorrido estudiantil. Eduardo y Carmita mis queridos padres, sabiendo que no hay forma de agradecerles por todo el esfuerzo y sacrificio depositado, pues este triunfo es también de ustedes. A mis hermanos Marlon y María de los Ángeles son mi razón de vivir y les prometo seguir adelante que serán muchos éxitos más por ustedes. A mi hermano Israel (+), hoy con orgullo puedo decirte misión más que cumplida, gracias por ser mi inspiración, apoyo y compañía.

Vanessa

AGRADECIMIENTO

Agradecimiento fraterno a mi compañera Vanessa Morocho con quien hemos sacrificado malas noches y al mismo tiempo nos hemos sabido valorar nuestras virtudes las cuales nos permitieron culminar el presente trabajo. Así mismo a los Ingenieros Franklin Moreno, Oswaldo Martínez, Wilson Baldeón y Edwin Altamirano quienes con su paciencia, experiencia y consejos nos han guiado desde el inicio hasta la culminación del presente trabajo. Y a todas las personas que estuvieron presente en el desarrollo de la misma, de todo corazón Gracias.

Jonathan

Mis más sinceros agradecimientos a nuestro tutor Ing. Oswaldo Martínez, gracias por ser nuestra guía y alentarnos en los momentos difíciles. A todos nuestros profesores que fueron nuestro apoyo para poder culminar esta carrera. A mi compañero de trabajo de titulación Jonathan por su paciencia y comprensión. Infinitas gracias a todos quienes fueron testigos de tanto esfuerzo y sacrificio.

Vanessa

TABLA DE CONTENIDO

PORTADA	I
FIRMAS DE RESPONSABILIDADES	II
RESPONSABILIDAD DE AUTORES	III
DEDICATORIA	IV
AGRADECIMIENTO	V
ÍNDICE DE TABLAS	VI
ÍNDICE DE FIGURAS	VII
RESUMEN	XIV
SUMMARY	XV
INTRODUCCION	XVI
CAPITULO I	
1. ESTADO DEL ARTE	1
1.1 Internet de las Cosas.....	1
1.1.1 Definición.....	1
1.1.2 Características y Avances	2
1.1.3 Seguridad en las IoT.....	3
1.1.4 Aplicaciones.....	4
1.1.4.1 Casas Inteligentes.....	4
1.1.4.2 Transporte y Logística.....	4
1.1.4.3 Salud	5
1.1.4.4 Ciudades Inteligentes	5
1.2 Smart Cities.....	6
1.2.1 Relación entre IoT y Smart Cities.....	6
1.2.2 Ecuador y las Smart Cities.	7

1.2.3	Smart Cities y soluciones sostenibles.....	8
1.2.4	Solución Medio Ambiental.	9
1.3	Tecnologías para la recolección de datos en una Smart City.	10
1.3.1	Sensores	10
1.3.1.1	Concepto y Características.....	10
1.3.2	Los Sensores y las Smart Cities	11
1.3.3	Tipos de sensores que se utilizan en una Smart City	12
1.3.3.1	Sensores de Aparcamiento	12
1.3.3.2	Sensores de Humedad	12
1.3.3.3	Sensores de Tráfico	12
1.3.3.4	Sensores de control de consumo de agua y electricidad	13
1.3.4	Sensores de recogida y tratamiento de residuos urbanos.	13
1.4	Tecnologías Para La Transmisión de Datos en una Smart City	14
1.4.1	Como ayudan las tecnologías inalámbricas en proyectos para Smart Cities.....	14
1.4.2	Cuadro comparativo entre posibles tecnologías inalámbricas.	14
1.5	Almacenamiento de Datos	16
1.6	Plataformas de hardware libre para proyectos de Smart Cities.....	16
1.6.1	Plataformas disponibles en el mercado	16
1.6.1.1	Arduino	16
1.6.1.2	Waspote.....	17
1.6.1.3	Spark	18
1.6.1.4	INTEL Galileo	18
1.7	Recolección de Desechos Sólidos en el Ecuador	20
1.7.1	Norma de calidad ambiental para el manejo y disposición final de desechos sólidos no peligrosos.....	20
1.7.1.1	Introducción	20

1.7.1.2	Objetivos	20
1.7.1.3	De las responsabilidades en el manejo de los desechos sólidos.....	20
1.7.1.4	De las prohibiciones en el manejo de desechos sólidos	21
1.7.1.5	Normas generales para la recolección y transporte de desechos sólidos no peligrosos.	21
1.7.2	Recolección de desechos sólidos: Tradicional vs. Inteligente.	21
1.7.2.1	Recolección Tradicional de Desechos Solidos.....	22
1.7.2.2	Desarrollo de la recolección.....	22
1.7.2.3	Desventajas de este Método	23
1.7.2.4	Recolección Inteligente de Desechos Solidos.....	23

CAPÍTULO II

2.	DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO PARA MONITOREAR LA RECOLECCIÓN DE DESECHOS EN TIEMPO REAL	24
2.1	Introducción	24
2.2	Diseño de la Arquitectura del Prototipo.....	24
2.3	Requerimientos para el diseño del prototipo.....	25
2.3.1	Bloque de Entrada	26
2.3.1.1	Eventos.....	26
2.3.1.2	Sensores	27
2.3.1.3	Sensor Ultrasónico	27
2.3.1.4	Sensores de Localización	29
2.3.1.5	Sensores de Humo.....	31
2.3.1.6	Sensores Acelerómetro/Giroscopio.....	33
2.3.2	Bloque de procesamiento	34
2.3.2.1	Hardware Libre	35
2.3.2.2	Plataforma Arduino.....	35
2.3.2.3	Arduino Mega	36

2.3.2.4	Esquema de Pines.....	37
2.3.3	Bloque de Salida	41
2.3.3.1	GPRS	41
2.3.3.2	Arquitectura GPRS.....	41
2.3.3.3	Asignación IP.....	42
2.3.3.4	Protocolos e Interfaces GPRS	43
2.3.3.5	Burst GPRS	43
2.3.3.6	Tipos de Terminales.....	44
2.3.3.7	Esquema de Codificación.....	44
2.3.3.8	Módulo GPRS	45
2.3.3.9	Shield GSM/GPRS SIM900.....	46
2.3.4	Bloque de Almacenamiento	46
2.3.4.1	Hosting Web	47
2.3.4.2	Dominio Web.....	48
2.3.4.3	Protocolo FTP	49
2.3.4.4	Trama FTP	49
2.3.5	Interfaz de usuario.....	50
2.4	Fuente de Alimentación	51
2.5	Implementación.....	52
2.5.1	Configuración de los dispositivos.....	52
2.5.1.1	Bloque de entrada	52
2.5.1.2	Bloque de procesamiento y Salida.....	54
2.5.1.3	Bloque de almacenamiento.....	61
2.5.1.4	Programación de la Interfaz de Usuario	62
2.5.2	Protección Física.....	63

CAPITULO III

3.	EVALUACIÓN Y COMPARACION DE RESULTADOS	64
3.1	Introducción	64
3.2	Estabilidad del Sistema	64
3.3	Pruebas de Transmisión	68
3.3.1	Esquema de transmisión	70
3.3.2	Recepción de variables	70
3.3.3	Tiempo de transmisión y tráfico de datos	72
3.4	Pruebas de Campo.....	72
3.4.1	Características del contenedor.....	72
3.4.2	Ubicación del módulo en el contenedor	73
3.4.3	Ubicación de los contenedores.....	74
3.4.3.1	Consideraciones para considerar cada evento	76
3.4.4	Ruta Óptima.....	77
3.5	Eficiencia Energética	79
3.6	Presupuesto	79
	CONCLUSIONES	80
	RECOMENDACIONES	81
	BIBLIOGRAFÍA	
	ANEXOS	

INDICE DE TABLAS

Tabla 1-1:	Ejemplos De Objetos IoT	3
Tabla 1-2:	Comparación entre tecnologías para transmitir datos en una Smart City	14
Tabla 1-3:	Comparación entre plataformas más usadas de hardware libre	19
Tabla 1-4:	Elementos básicos para la recolección tradicional de desechos	22
Tabla 2-1:	Descripción de los Bloques de Proceso	25
Tabla 2-2:	Descripción de los eventos y sensores.....	26
Tabla 2-3:	Cuadro Comparativo Sensores Ultrasónicos	28
Tabla 2-4:	Cuadro Comparativo Sensores de Localización.....	30
Tabla 2-5:	Cuadro Comparativo Sensores de Gases	32
Tabla 2-6:	Cuadro Comparativo Sensores Acelerómetro/Giroscopio.....	33
Tabla 2-7:	Cuadro Comparativo de Modelos Arduino	36
Tabla 2-8:	Tipos de Terminales	44
Tabla 2-9:	Esquema de Codificación	44
Tabla 2-10:	Modelos GPRS para Arduino	45
Tabla 2-11:	Hosting de Pago y Gratuito	48
Tabla 2-12:	Ventajas y Desventajas de las baterías para alimentar proyectos con arduino.	51
Tabla 2-13:	Conexión de sensores con arduino Mega.	53
Tabla 2-14:	Distribución de pines Shield GSM/GPRS	56
Tabla 2-15:	APN, usuario y contraseña de operadoras celulares en Ecuador.....	57
Tabla 2-16:	Comandos AT para conectarse al internet.....	58
Tabla 2-17:	Comandos para Autenticacion FTP.....	59
Tabla 2-18:	Comandos para subir los datos	60
Tabla 3-1:	Variables recibidas del módulo con nivel de llenado vacío.	65
Tabla 3-2:	Repetitividad del módulo en estado contenedor medio.....	66
Tabla 3-3:	Repetitividad del módulo en estado contenedor lleno.....	67
Tabla 3-4:	Características del contenedor.....	73
Tabla 3-5:	Consideraciones de nivel de llenado	76
Tabla 3-6:	Consideraciones para alertar posibles incendios.	76
Tabla 3-7:	Presupuesto.....	79

INDICE DE FIGURAS

Figura 1-1:	IoT Definition	1
Figura 1-2:	Libelium-Logistics.....	5
Figura 1-3:	Smart City! Las ciudades del presente.....	6
Figura 1-4:	Smart City.....	6
Figura 1-5:	La experiencia japonesa en ‘Smart Cities’ se compartió en Quito	8
Figura 1-6:	Smart Cities, gestión eficiente, sostenible y transparente	9
Figura 1-7:	Smart city Sostenibilidad y Medio Ambiente	9
Figura 1-8:	Transducers: sensors and actuators.	11
Figura 1-9:	Sensores inteligentes de parking en la Feria Intertraffic 2016.....	11
Figura 1-10:	Here’s an intelligent garbage collection system.....	13
Figura 1-11:	Arquitectura de alto nivel de tecnologías M2M en el marco de las Smart Cities ...	15
Figura 1-12:	Placa Arduino.....	17
Figura 1-13:	Waspote-Wireless Sensor Networks.....	17
Figura 1-14:	Placa Spark.....	18
Figura 1-15:	Placa INTEL Galileo.....	19
Figura 2-1:	Bloques de Proceso	25
Figura 2-2:	Sensor Ultrasónico SRF05	28
Figura 2-3:	Sensor Ultrasónico con vista abajo	29
Figura 2-4:	Módulo GPS NEO 6M.....	30
Figura 2-5:	Sensor GPS con vista al cielo	31
Figura 2-6:	Módulo MQ-135.....	32
Figura 2-7:	Módulo MPU-6050.....	34
Figura 2-8:	Arduino Mega.	37
Figura 2-9:	Pines Power.....	38
Figura 2-10:	Pines Digitales.	39
Figura 2-11:	Pines PWM.	39
Figura 2-12:	Pines de Comunicación.....	40
Figura 2-13:	Pines Analógicos.....	40
Figura 2-14:	Arquitectura GPRS.	42
Figura 2-15:	Protocolos GPRS.	43

Figura 2-16:	Figura 2-15. Burst GPRS.	43
Figura 2-17:	Shield GSM/GPRS SIM900.	46
Figura 2-18:	Recursos que forman parte de un Hosting Web.....	47
Figura 2-19:	Buscador de Dominios de Go Daddy.....	49
Figura 2-20:	Trama FTP	50
Figura 2-21:	Bloque de entrada	53
Figura 2-22:	IDE Arduino.....	54
Figura 2-23:	Tarjeta SIM para Shield GSM/GPRS, SIM 900	54
Figura 2-24:	Botón encendido.	55
Figura 2-25:	Puntos para soldar	55
Figura 2-26:	Selector Serial	56
Figura 2-27:	Diagrama de conexiones de los tres bloques iniciales.	61
Figura 2-28:	Almacenamiento de archivos online	62
Figura 2-29:	Programación Interfaz de usuario	62
Figura 2-30:	Caja de protección.....	63
Figura 3-1:	Módulo ubicado en la Facultad de Informática y Electrónica de la ESPOCH	68
Figura 3-2:	Ubicación en el Centro Integral para la Salud de la ESPOCH	69
Figura 3-3:	Ubicación del módulo en el modular de Electrónica	69
Figura 3-4:	Esquema de transmisión	70
Figura 3-5:	Visualización en tiempo real de los datos recibidos por uno de los módulos.	71
Figura 3-6:	Trafico PHP	71
Figura 3-7:	Ubicación del módulo	73
Figura 3-8:	Contenedor 1	74
Figura 3-9:	Contenedor 2	75
Figura 3-10:	Contenedor 3	75
Figura 3-11:	Información del estado de cada contenedor en tiempo real	77
Figura 3-12:	Ruta óptima para la recolección de desechos en contenedor 1 y 3.	78
Figura 3-13:	Distribución tentativa, contenedores en el centro de Riobamba	78

RESUMEN

Para el presente trabajo de titulación se diseñó e implementó un sistema multimodal para monitorear y controlar la recolección de desechos sólidos en tiempo real para Smart Cities, realizado en la ciudad de Riobamba. El sistema permite monitorear de forma remota el nivel de llenado, localización, alerta de incendio y manipulación de un contenedor de basura, en tiempo real, optimizando con eficiencia y eficacia la recolección de los desechos en una ciudad, mejorando los parámetros ambientales. A partir del método analítico, consideramos cada parte del diagrama por bloques, interrelacionando todos los elementos para formar un solo patrón del sistema de monitoreo. El sistema de monitoreo comprende, un bloque de entrada, procesamiento, salida de datos, un canal de comunicación celular, y la recepción mediante servidor web, mostrando los eventos de cada contenedor en tiempo real. El bloque de entrada contiene un grupo de sensores: ultrasónico HY-SRF05, GPS NEO-6M, detector de humo MQ-135, acelerómetro MPU 6050, cada uno se encargó de la adquisición de datos específicos del sistema. Para el bloque de procesamiento de información y señales se utilizó una placa ARDUINO-MEGA 2560, encargado de evaluar los datos obtenidos por los sensores. En el bloque de salida se utilizó un transmisor Shield GSM/GPRS SIM 900, permitiendo enviar todos los valores obtenidos desde el bloque de entrada hacia un servidor web. El servidor web presentó los datos medidos en un intervalo de 1 a 3 minutos, mostrando un nivel de estabilidad adecuado para transmisiones en tiempo real. Las pruebas realizadas en la ciudad de Riobamba mostraron niveles aceptables de potencia con respecto a la cobertura celular. Para un sistema en tiempo real con GSM/GPRS recomendamos utilizar operadoras que brinden coberturas amplias con costos económicos para el uso de las redes en Smart Cities.

Palabras Claves: <TELECOMUNICACIONES Y REDES>, <CIUDADES INTELIGENTES (SMART CITIES)>, <PARÁMETROS AMBIENTALES>, <CONTROL DE DESECHOS>, <DESECHOS SÓLIDOS>, <COMUNICACIÓN CELULAR>, <TIEMPO REAL>.

SUMMARY

This research work was carried out to design and implement a multimodal system in order to monitor and control collection of solid wastes in real time for Smart Cities in the city of Riobamba. This system allows monitoring filling, localization, and fire warning level; and manipulation of garbage trash in real time by optimizing with effectiveness and efficacy the collecting of solid wastes in a city and then environmental parameters will be bettered. Every single block diagram was considered through an analytic method by inter-relating all elements to form a whole pattern of monitoring system. The monitoring system has an input, processing and output block, a communication mobile channel, and reception by means of a web server showing the events of containers in real time. The input block contains a group of sensors like: ultrasonic HY-SRF05, GPS NEO-6M, smoking detector MQ-135, and accelerometer MPU-6050, each was in charge of getting specific information of the system. An ARDUINO-MEGA 2560 was used for processing information and signs in order to assess data obtained by sensors. A transmitter Shield GSM/GPRS SIM900 was used in the output block to allow sending all the data obtained from input block to the web server. The web server presented data measured in a range of 1 to 3 minutes showing an appropriate stability level for transmissions in real time. Tests applied in the City of Riobamba showed acceptable levels of power related to mobile signal. It is recommended that a system in real time with GSM/GPRS must consider good mobile signal providers which cover large areas with economic costs to use networks in Smart Cities.

Key words: <TELECOMMUNICATIONS AND NETWORKS>, < SMART CITIES>, < ENVIRONMENTAL PARAMETERS>, <WASTE CONTROL>, <SOLID WASTE>, <MOBILE COMMUNICATION>, <REAL TIME>.

INTRODUCCIÓN

Actualmente en las ciudades del Ecuador hay un número considerable de personas que habitan en zonas urbanas, quienes generan una gran cantidad de residuos. Para llevar a cabo el servicio de recolección, la municipalidad de cada ciudad, cuenta con camiones especializados y personal calificado para realizar esta tarea. Cuando los recolectores realizan su recorrido, pueden encontrarse con contenedores a niveles variables de desechos, sin embargo deben revisar cada uno de los mismos, produciendo pérdida de tiempo, provocando mayor cantidad de viajes, mayor gasto en combustible y por lo tanto mayores emisiones de CO₂. (C.I.T.I.C., 2015 p. 3)

En el presente proyecto se, diseñó e implementó un sistema multimodal para monitorear y controlar la recolección de desechos sólidos en tiempo real para Smart Cities, para lograrlo se procedió a investigar sobre el avance de las IoT y las ciudades inteligentes, los diversos sensores que monitorean niveles de llenado y otros eventos en tiempo real, un microcontrolador que procesó los datos obtenidos por el sensor, una tecnología de comunicación que ofreció la cobertura idónea para una Smart City y se evaluó la fiabilidad de cada evento en cada contenedor a través de una página web.

Para lograr el objetivo planteado se utilizó una metodología analítica llevando a cabo las siguientes actividades: investigación bibliográfica, diseño del sistema, implementación de un prototipo, evaluación de resultados y documentación. El proyecto es un aporte académico dirigido para todos los estudiantes de carreras tecnológicas que se interesen por el monitoreo de sistemas de manera remota a través de internet, utilizando tecnologías de comunicación inalámbrica y por otro lado ayudará a cualquier ciudad que desee optimizar el proceso de recolección de desechos sólidos.

ANTECEDENTES

A partir del año 1926 Nikola Tesla, desarrolló trabajos teóricos que conformaron la base de las comunicaciones inalámbricas, para el año de 1966 el departamento de defensa de los EEUU creó la primera red de computadoras llamada ARPANET, seguidamente aparecieron los protocolos TCP/IP basados en internet y transmisión de datos. Después de varios años Berners-Lee logro la comunicación entre cliente-servidor (http), que en años más tarde sería lo que ahora se conoce como world wide web o WWW, a partir de esto comienza la revolución de Internet. (SorayapaNiagua, 2012)

El concepto de Internet de las cosas aparece en el año de 1999 por Kevin Ashton, que impartió en una conferencia de Procter & Gamble. Luego la agencia de las Naciones Unidas publicó el primer estudio sobre el tema, a partir de este momento el internet de las cosas alcanza otro nivel. De este modo el internet de las cosas se ha convertido en la segunda revolución digital que cambiará los estilos de vida. (SorayapaNiagua, 2012)

El internet de las cosas se aprovecha para medir parámetros externos como: temperatura, energía, luz, humedad, etc., de forma automática y sin la interacción del ser humano, el ejemplo más cercano de su aplicación está en el propio hogar y otro gran ámbito de acción es supervisar ciudades avanzadas o Smart Cities. Las ciudades inteligentes ayudan a solucionar los principales desafíos de: infraestructura, seguridad, administración pública y servicios para los ciudadanos. (ENDESA EDUCA, 2014)

Existen diversas aplicaciones de Smart Cities, en este proyecto nos enfocaremos al cuidado del medio ambiente, se trata de contenedores de basura inteligentes, que desencadenen una alarma cuando se encuentren cercanos al límite de su capacidad, además que puedan captar alertas de incendio en su interior dando así una señal de emergencia. De este modo se puede organizar rutas de recogida que permitan mayor eficiencia, y lograr reducir la contaminación del ambiente que se ha visto afectada en los últimos años.

FORMULACIÓN DEL PROBLEMA

La recolección inteligente de desechos sólidos se ha implementado en ciertos países europeos como Holanda, conocida por ser uno de los primeros países que respeta el Medio Ambiente. El sistema que utiliza se basa en sensores colocados en el basurero que envían el estado de llenado a la empresa de recolección de basura.

En Ecuador, el sistema de recolección de basura está enfocado en una revisión manual de cada basurero, ocupando tiempo y combustible; recursos que se podrían optimizar. En el presente proyecto se presenta un prototipo para la ciudad de Riobamba, pensando en el cuidado del ambiente y la calidad de vida de la ciudadanía. Se trata de un sistema que informará, que contenedores están llenos y así trazar rutas óptimas que ayudarán a reducir el consumo de gasolina, las emisiones de CO₂ y tiempo de recorrido de los vehículos recolectores.

JUSTIFICACIÓN TEÓRICA

La vinculación de objetos inteligentes hacia el internet, permite un intercambio de datos y llevar la información a los usuarios con mayor seguridad. El internet de las cosas ha permitido administrar algunos de los principales servicios como el control de alumbrado público, el control del tránsito y los estacionamientos a través de una red Inteligente.

La tecnología del internet de las cosas ofrece a los países la oportunidad de satisfacer sus necesidades básicas. Se proyecta que para el 2050, el 66% de la población vivirá en ciudades. Las redes de banda ancha, las cosas conectadas y los datos abiertos ayudaran a impulsar la competitividad, la sostenibilidad y la habitabilidad.

Para lograr todo lo mencionado anteriormente se necesitan de tecnologías y plataformas de redes inteligentes, un sistema utilizado en los últimos tiempos son las llamadas Redes de sensores. Son un conjunto de motas que se coordinan para llevar a cabo un proceso de control y monitoreo para llegar a una aplicación específica y en tiempo real.

Con la ayuda de las diferentes tecnologías mencionadas se busca diseñar un sistema que permita monitorear la recolección de desechos sólidos en tiempo real. Todos los datos obtenidos en el sistema

se almacenaron en un archivo online como son: ubicación de cada contenedor, notificaciones y alertas de posibles emergencias, ya sea: incendios, manipulación y límite de llenado de los mismos.

JUSTIFICACIÓN APLICATIVA

Con el auge de las IoT, las personas encontraron nuevos mecanismos para mejorar su estilo de vida y solucionar problemas a través de sensores que se conectaran a una red, permitiendo la supervisión automática de un evento.

Entre las mejoras del estilo de vida está el tema de las ciudades inteligentes, escenario que administra, gestiona, controla los recursos públicos como luz, agua, teléfono, seguridad y limpieza dentro de una ciudad.

Para la limpieza, las ciudades inteligentes se benefician con una supervisión automática del estado de los contenedores de basura, permitiendo saber si estos están copados de desperdicios, si se encuentran en buen estado, su geolocalización o si han sufrido alguna manipulación.

A través de sensores colocados estratégicamente en el contenedor, se tiene datos en tiempo real los cuales informarán la situación de los contenedores en una aplicación Web.

Actualmente en la ciudad de Riobamba, no se ha implementado IoT en el sector de recolección de desechos sólidos, el presente proyecto propone una supervisión y control de estos contenedores.

OBJETIVOS

OBJETIVO GENERAL

Diseñar e implementar un sistema multimodal para monitorear y controlar el sistema de recolección de desechos sólidos en tiempo real.

OBJETIVOS ESPECIFICOS

- Investigar sobre el avance de las IoT y redes de sensores orientadas a las ciudades inteligentes.
- Analizar los diferentes sensores que permiten supervisar una recolección de desechos sólidos en tiempo real.
- Escoger una plataforma de hardware libre que tenga las características adecuadas para el prototipo que se ha planteado.
- Determinar el tipo de tecnología de comunicación inalámbrica que ofrezca la cobertura idónea para una Smart City.
- Diseñar el prototipo utilizando todos los elementos necesarios.
- Implementar el prototipo y validar la fiabilidad de los datos obtenidos.
- Crear una página web que permita visualizar en tiempo real todos los eventos que se van a producir en los contenedores de desechos sólidos.

CAPITULO 1

1. ESTADO DEL ARTE

1.1 Internet de las Cosas

1.1.1 Definición

Internet de las cosas o bien conocido con las siglas IoT se define por la UIT y IERC como una “dynamic global network infrastructure” (VERMESAN, 2009 p. 15).

Es una nueva tendencia que hace posible la conexión de cada elemento que nos rodea hacia la red global internet, a través de varias tecnologías de comunicación e información, la idea de las IoT es controlar de manera remota cada elemento desde un Smartphone o algún servidor que tenga acceso a internet. En la *Figura 1-1* se muestra un ideal general de la definición de IoT.

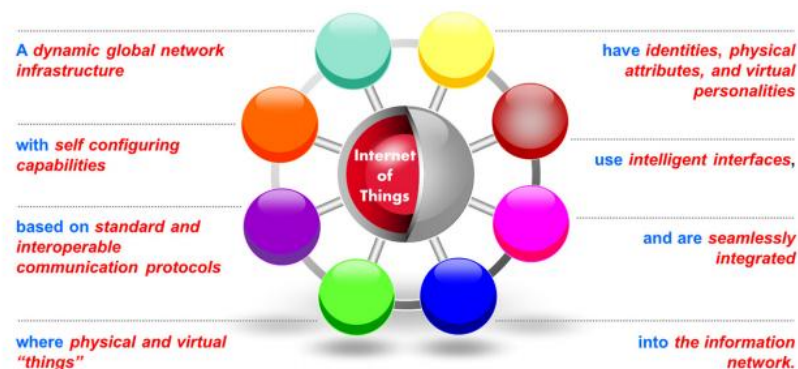


Figura 1-1: IoT Definition

Fuente: (VERMESAN, 2009 p. 17)

Como lo menciona (Cisco, 2011): IoT conocido como Internet of things, ha venido a causar impactos en el ámbito educativo, en la comunicación, negocios, ciencia, gobierno y la humanidad entera. Siendo IoT una representación de la siguiente evolución de internet, se está presentando varios casos de estudio de proyectos relacionados con este nuevo avance, dichos proyectos han venido a solventar las necesidades de la sociedad, mejorando la distribución de los recursos, entre las escalas sociales.

La idea de esta nueva evolución es la convergencia de múltiples factores como son: hardware libre, sensores de medición, comunicaciones inalámbricas o móviles, plataformas IoT, con la finalidad de conectar varios objetos de manera independiente en la red global. Se prevé que el número de objetos que estarán conectados al internet será de 50.000 millones de dispositivos en el 2020. (EVERLET, 2013 p. 5)

Iniciando con la posibilidad de ofrecer servicios a la sociedad con un resultado de la monitorización de nuestras actividades cotidianas ofreciendo mejorar los servicios para una mejor eficiencia.

1.1.2 Características y Avances

Entre las principales características tenemos:

- **Inteligencia:** Los objetos que se conectan a internet deben tener un grado de inteligencia para interrelacionarse entre sí.
- **Arquitectura:** Como se menciona en: (RINCON B., 2015) el sistema IoT será probablemente un ejemplo de “arquitectura orientada a eventos”, esta arquitectura estará basada en procesos de tiempo real, considerando la complejidad debido a la inexistencia de reglas determinadas para su control.
- **Tipo de Sistema:** Internet of things estará basado en sistemas complejos, porque utilizará semi-bucles, tanto abiertos como cerrados para su desarrollo, además se considera un sistema TI en forma paralela masiva.

Gracias al internet de las cosas, hoy en día el alcance a internet es más accesible. Dicho de otra manera se ha convertido en un servicio básico. En la *Tabla 1-1* se visualiza los avances que se han venido surgiendo en los últimos tiempos.

Tabla 1-1: Ejemplos De Objetos IoT

AVANCE	DESCRIPCIÓN
Kolibree	Cepillo inteligente detecta si los dientes se han cepillado bien.
iGrill Mini	Termómetro que determina la cocción de la carne
Ok-i-dokeys	Cerradura que se activa con una app en los Smartphones
WeMo	Permite encender y apagar los dispositivos eléctricos con un Smartphone
Vaso Vessyl	Vaso inteligente calcula la composición nutricional del líquido interior.
Vitality GlowCaps	Pastillero que ayuda a su dueño a tomar la medicación a los horarios exactos
Smart Tennis Sensor	Ayuda a mejorar el rendimiento en el tenis.

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016.

Fuente: (VELASCO R., 2015 p. 3)

1.1.3 Seguridad en las IoT.

Las plataformas IoT, consideran asegurar las plataformas de posibles ataques por terceras personas, para ello es necesario imponer ciertas soluciones en donde se pretende proteger las aplicaciones y los dispositivos, buscar anomalías en todo el sistema y corregir errores de manera remota. (SYMANTEC, 1995)

Para lograr establecer un algoritmo de seguridad en el sistema IoT se deber considerar los siguientes parámetros:

- Seguridad en los dispositivos: bloquear el acceso a personas o dispositivos no autorizados.
- Autenticación de los dispositivos: plantear claves encriptadas y algoritmos de autenticación.
- Autorización de códigos: verificar los códigos que se establecen en cada dispositivo con el método de autenticación.

Una de las visiones que se tiene para futuro es la actualización del software de cada dispositivo IoT de manera remota, así como también, se pretende realizar un algoritmo que permita el análisis más sofisticado de todo el sistema IoT.

1.1.4 Aplicaciones

IoT da uso a la tecnología para crear plataformas, aplicaciones y dispositivos que contribuyen al estilo de vida, seguridad, protección del medio ambiente, ahorro de energía entre otros. (MAROTO P., 2013). Ya se cuenta con varios dispositivos o prototipos que pretenden cumplir estas tareas. A continuación se presentan los más relevantes escenarios de aplicaciones IoT.

1.1.4.1 Casas Inteligentes

El objetivo de tener una casa inteligente es brindar un aporte a las viviendas de nuestra sociedad en varios ámbitos como el ambiente de vida y confort. (MAROTO P., 2013)

Para esto se debe empezar con una solución de mejora en el uso del agua y la energía, que, si bien es cierto, son necesarias para la humanidad, pero se ha detectado en los últimos tiempos el abuso de los mismos. Otra de las mejoras que se puede dar a un hogar es la seguridad, como: alerta de incendio, entrada no autorizada, etc.

1.1.4.2 Transporte y Logística

IoT proporciona varias ventajas en cuanto a la gestión de ventas y suministro de operaciones en su manera global así como también lograr identificar de manera automática las mercancías que se están transportando, hoy en día gracias a las plataformas IoT podemos contar con la comprobación automática de la entrada de bienes en tiempo real. (Debasis Bandyopadhyay, 2011 pp. 19-20).

En la *Figura 1-2* se puede observar el proceso de logística inteligente que maneja la empresa Libelium.



Figura 1-2: Libelium-Logistics

Fuente: (ESPAÑA. Libelium, 2015)

1.1.4.3 Salud

En el sector de la salud las plataformas IoT desempeñan aplicaciones como: monitorización de los parámetros médicos, administración de fármacos, prevención y control de ciertas enfermedades que se puedan producir en las personas, mejorar los horarios de atención médica, ya que los pacientes tendrán un nivel de privilegio dependiendo del tipo de emergencia como personas con enfermedades del corazón, derrame cerebral, alteraciones cognitivas etc. (BANDYOPADHYAY, et al., 2011 pp. 16-17)

1.1.4.4 Ciudades Inteligentes

Se trata de una zona urbana que pretende mejorar la calidad de vida y distribuir los recursos que posee una ciudad de manera eficiente e inteligente. Las ciudades inteligentes se valen de la ayuda de las redes integradas, sensores y sistemas inteligentes. (Cisco, 2011)

Para poder establecer aplicaciones como estacionamiento inteligente, controlar la cantidad de tráfico en una ciudad con la finalidad de buscar rutas alternas, contaminación acústica, recolección de desechos inteligente entre otros. Como se puede ver en la *Figura 1-3*.



Figura 1-3: Smart City! Las ciudades del presente.

Fuente: (ESCUELA DE ORGANIZACIÓN INDUSTRIAL, 2015)

1.2 Smart Cities

1.2.1 Relación entre IoT y Smart Cities



Figura 1-4: Smart City.

Fuente: (ESCUELA DE ORGANIZACIÓN INDUSTRIAL, 2015)

Ciudades Inteligentes o Smart Cities es un tema que no se puede conceptualizar ya que su significado cubre muchas aplicaciones. Una de las principales aplicaciones de una Smart City es la interacción entre cosas y la calidad de vida de las personas que viven en una ciudad, con el fin de dar soluciones sostenibles y rentables, mejorando la calidad de vida de las personas que habitan en ella.

Como lo mencionan (MUÑOZ, 2015) Actualmente hay 13 millones de elementos conectados a la Web. Es por eso que las Ciudades Inteligentes son las que más influyen para el crecimiento de la IoT. Un ejemplo son los países de Europa como Reino Unido y España que han empezado con un sinnúmero de proyectos que han solucionado necesidades básicas como alumbrado y recogida de residuos.

Para el 2050, la ONU espera que más del 70% de la población este concentrada en las ciudades, permitiendo que en los centros poblados se desarrollen un sinnúmero de sistemas IOT que beneficien a sus ciudadanos. (enigmedia, 2016)

El crecimiento de una ciudad exige la capacidad de prever soluciones, con un plan que cubra las necesidades del mañana. Cada ciudad es un sistema único por lo que la solución aplicada en cada una de estas variara según los parámetros que se desee trabajar. Para el desarrollo de una ciudad sostenible se necesita varios elementos como son: Tecnología, Sociedad, Iniciativa Privada y pública y demás organizaciones que deseen formar parte del cambio. (SAPIENSA, 2015)

1.2.2 Ecuador y las Smart Cities.

Empresas internacionales han identificado en nuestro país oportunidades de negocio para el sector de Smart Cities, encontrando como potenciales clientes a las instituciones públicas y entes administradores del territorio ecuatoriano. Las actividades prometedoras son potabilización de agua, alcantarillado, tratamiento de la basura, turismo y prevención de desastres. (J, CEVALLOS, 2015)

El crecimiento económico de Ecuador en los últimos años ha sido prospero sin dejar de lado sus desequilibrios como la depreciación del dólar y petróleo que lo han azotado. En esta etapa de crecimiento se ha dado las condiciones adecuadas para que las instituciones públicas puedan utilizar tecnologías IoT para suplir servicios de carácter básico. (EL UNIVERSO, 2015)

Para darse a conocer las ofertas y demandas de Smart City, en la ciudad de Quito se organizaron las Jornadas Españolas de Smart Cities en Octubre del 2015, como se puede ver en la *Figura 1-5* en donde se presentaron algunas empresas españolas dedicadas a ofrecer soluciones inteligentes. Como participantes fueron los ayuntamientos de algunos municipios de las principales ciudades del Ecuador, como son Quito, Guayaquil, Cuenca y de algunas provincias del Ecuador. (SMARTCITIES IDOM, 2015)



Figura 1-5: La experiencia japonesa en ‘Smart Cities’ se compartió en Quito
Fuente: (ECUADOR. Líderes, 2015)

1.2.3 Smart Cities y soluciones sostenibles

Las ciudades al tener un número importante de ciudadanos forman parte de la sostenibilidad como se puede ver en la *Figura 1-6* por la transformación de agujeros de amenaza y a su vez pueden ser escenarios de la industria de la sostenibilidad. Se consideran agujeros de amenaza ya que su concentración de personas acelera el cambio climático, incrementando las emisiones de CO₂.

A su vez para su desarrollo sostenible se considera que crear una ciudad inteligente funciona como antídoto que reduce el impacto de los agentes contaminantes. Para buscar la sostenibilidad las empresas tecnológicas lanzan soluciones a problemas que los ciudadanos no las han considerado como tales. (OpenMind, 2015)

Según la ITU una ciudad inteligente es sostenible cuando es innovadora y aprovecha las TICS y otros medios para mejorar la calidad de vida. (UIT, 2016)



Figura 1-6: Smart Cities, gestión eficiente, sostenible y transparente

Fuente: (ESPAÑA. HIBERUS, 2014)

1.2.4 Solución Medio Ambiental.

Dentro de las soluciones que se tiene con Smart Cities y el medio Ambiente se encuentra el campo de la recolección de residuos urbanos con sensores que informan si está lleno el contenedor y ayudan a coordinar su respectivo despacho de basura. Además dentro de la ciudad se puede realizar mediciones para monitorear la calidad de aire, contaminación acústica, etc.

Estas soluciones como se puede ver en la *Figura 1-7* permiten tener una idea más clara de los tipos de contaminantes que amenazan a la ciudad, como su respectivo monitoreo y control en tiempo real, de esa manera se realiza una reducción considerable de la contaminación, ya que al detectarse una anomalía, se realizara los respectivos correctivos para reducir los impactos.



Figura 1-7: Smart city Sostenibilidad y Medio Ambiente

Fuente: (ExpoBarcelona, 2015)

1.3 Tecnologías para la recolección de datos en una Smart City.

La recolección de información en una ciudad se da mediante el uso de sensores que captan todos los eventos que se vayan dando dentro de la misma.

1.3.1 Sensores

1.3.1.1 Concepto y Características

Un sensor se trata de un dispositivo que recibe y da una respuesta a una acción que se produce en el medio físico. (ROUSE M, 2012).

Las señales que un sensor puede captar del medio pueden ser: luz, calor, movimiento, humedad, radiación, presión, peso, etc... La detección que realizan los sensores se dan debido a las características que presentamos a continuación:

- Rango de medida: tipo de dominio en la magnitud que mida el sensor.
- Precisión: exactitud de los datos recibidos en el sensor.
- Resolución: variación de la medida con la que se reciben los datos.
- Tiempo de Respuesta: varía dependiendo de la variación de magnitud a medir.
- Utiliza circuitos de acondicionamiento para generar señales normalizadas.

Los sensores pueden captar varias señales del medio externo como se puede ver en la *Figura 1-8*, esto permite realizar sistemas que controlen y monitoreen varios eventos en tiempo real y con un bajo margen de error.

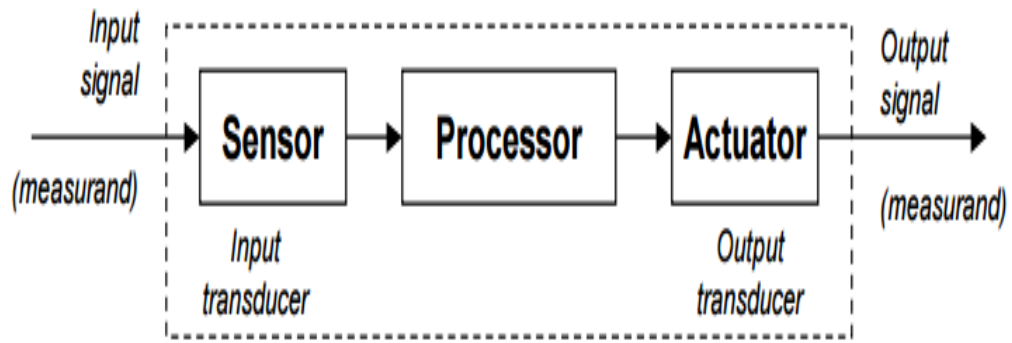


Figura 1-8: Transducers: sensors and actuators.

Fuente: (GUTIERREZ R. pp. 2-3)

1.3.2 Los Sensores y las Smart Cities

En una Smart City los recolectores de datos o más bien llamados, sensores, están repartidos en puntos estratégicos de la misma como se puede ver en la *Figura 1-9*, estos con el fin de captar eventos para controlar y monitorear tomando el inicio de las redes de sensores inalámbricos se puede obtener: disponibilidad de estacionamientos, servicios públicos, medio ambiente, cantidad de tráfico, etc..

Todos los sensores se conectan a la nube a través de tecnologías inalámbricas o móviles, de este modo los usuarios pueden observar los eventos que se vayan dando desde una PC o móvil.

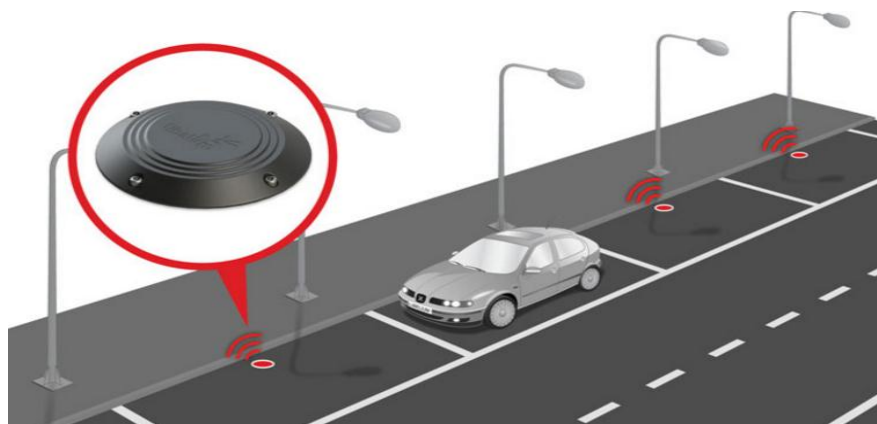


Figura 1-9: Sensores inteligentes de parking en la Feria Intertraffic 2016.

Fuente: (eSMARTCITY, 2016 p. 1)

1.3.3 Tipos de sensores que se utilizan en una Smart City

Existen diversos sensores en el mercado que ya se pueden implementar dentro de una Smart City, la principal función de estos sensores es la captación de información o eventos que se vayan dando en ámbitos específicos, así por ejemplo en un sistema de monitoreo de calidad de aire se puede dar avisos de contaminación en tiempo real y de este modo tomar medidas de precaución para evitar posibles enfermedades respiratorias.

Entre los sensores más relevantes dentro de una ciudad inteligente tenemos los siguientes:

1.3.3.1 Sensores de Aparcamiento

Estos sensores ayudan a verificar si hay espacio disponible para un parqueadero mejorando la fluidez del tráfico dentro de una ciudad, estos sensores pueden aportar actualmente en nuestro país, específicamente en las horas pico, se puede ver la cantidad de personas que desean estacionar su vehículo, provocando así un bloqueo de tráfico en las calles, con este sistema se puede informar a la ciudadanía los posibles lugares de aparcamiento disponibles.

1.3.3.2 Sensores de Humedad

Su fin es detectar la humedad de los parques y jardines públicos con el fin de que sean regados cuando sea necesario, de este modo se logra proteger las áreas verdes de las ciudades, que son lugares de recreación para las personas, además, se contribuye al ahorro del agua.

1.3.3.3 Sensores de Tráfico

Ayudan alertar a la ciudadanía sobre la cantidad de tráfico en las vías, con el fin de buscar rutas alternas para llegar a su destino, estos sensores pueden contribuir a las grandes ciudades metrópolis del país.

1.3.3.4 Sensores de control de consumo de agua y electricidad

Ayudan al ciudadano a controlar la cantidad de agua y energía que están consumiendo ya sea diaria, semanal, mensual o anualmente, el propósito de los sensores es llevar a la población a concientizar y a la vez incentivar al ahorro de estos dos recursos importantes.

Existen algunos sensores que se utilizan para el tratamiento de residuos sólidos en una ciudad, en este capítulo se menciona de manera general cual es el trabajo de estos sensores y para que pueden ser útiles.

1.3.4 Sensores de recogida y tratamiento de residuos urbanos.

Este tipo de sensores ayudan a controlar y monitorear el nivel de llenado de los contenedores de basura como se puede ver en la *Figura 1-10* que se encuentran distribuidos en la ciudad, de este modo las rutas de los recolectores de basura son más eficientes, además se puede prever algún tipo de alerta, golpe, robo del contenedor.



Figura 1-10: Here's an intelligent garbage collection system

Fuente: (ESTADOS UNIDOS. The internet of all things, 2014)

1.4 Tecnologías Para La Transmisión de Datos en una Smart City

1.4.1 Como ayudan las tecnologías inalámbricas en proyectos para Smart Cities.

Las redes de comunicación en una Smart City son de vital importancia para el desarrollo de los dispositivos de medición, permite la comunicación entre los mismos y las personas. La interoperabilidad entre los dispositivos también es importante para desarrollar un sistema eficiente y con mayor facilidad.




Cuando hablamos de Smart Cities no se piensa en redes fijas, ya que, por sus características no es óptima para usarla en una ciudad, por ello se procede a enfocarse en redes inalámbricas. (ESPAÑA. Telefónica, 2011 p. 58)

No se puede mencionar específicamente que cierta tecnología inalámbrica es la más acorde para una Smart City ya que la tecnología para utilizar dependerá de las características del medio en donde se vaya a desarrollar el sistema.

1.4.2 Cuadro comparativo entre posibles tecnologías inalámbricas.


En la *Tabla 1-2*, se presenta un cuadro comparativo de las posibles tecnologías inalámbricas y móviles más adecuadas para una finalidad y un entorno determinado, que se puede utilizar en una Smart City.

Tabla 1-2: Comparación entre tecnologías para transmitir datos en una Smart City

	Rango	Velocidad de Datos	Frecuencia de operación	Topología de Red	Modulación	Estándar
	10-100m	20-250kbps	868 Mhz, 902-928Mhz, 2.4 Ghz	Árbol, estrella, malla	OQPSK	802.15.4
	30-100m	54 Mbps	2.4 Ghz	AD-HOC, Mesh	PSK, QPSK, OFDM	802.11
	1-15 Km	100 Mbps	2.3-3.5 Ghz	Punto-Punto, Punto-Multipunto	OFDM	802.16

Continuará...

Continúa...

GSM/GPRS 	30 km	114 Kbps	800 Mhz 1800 Mhz 1900 Mhz	Punto-Punto, Punto-Multipunt	GMSK	GPRS
---	-------	----------	---------------------------------	---------------------------------	------	------

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Actualmente las ciudades inteligentes, ya cuentan con el servicio M2M o más conocido como: comunicaciones dispositivo a dispositivo, máquina a máquina, etc... Lo que permite que dos dispositivos se puedan comunicar de manera remota, inclusive se puede decir, independientemente del lugar en donde se encuentren.

Uno de los elementos fundamentales que están dentro de un entorno M2M son las redes de comunicación mediante los cuales se puede establecer la comunicación entre dos dispositivos, pueden ser: con cable, o inalámbricos como: GSM/GPRS/UMTS/HSDPA/Wifi Zigbee, etc., este servicio M2M se encuentran en un proceso de desarrollo y se espera que en los próximos años tenga una considerable demanda en las ciudades inteligentes.

En la *Figura 1-11* se puede ver la arquitectura de un entorno M2M y como se conectan los dispositivos a la red global.

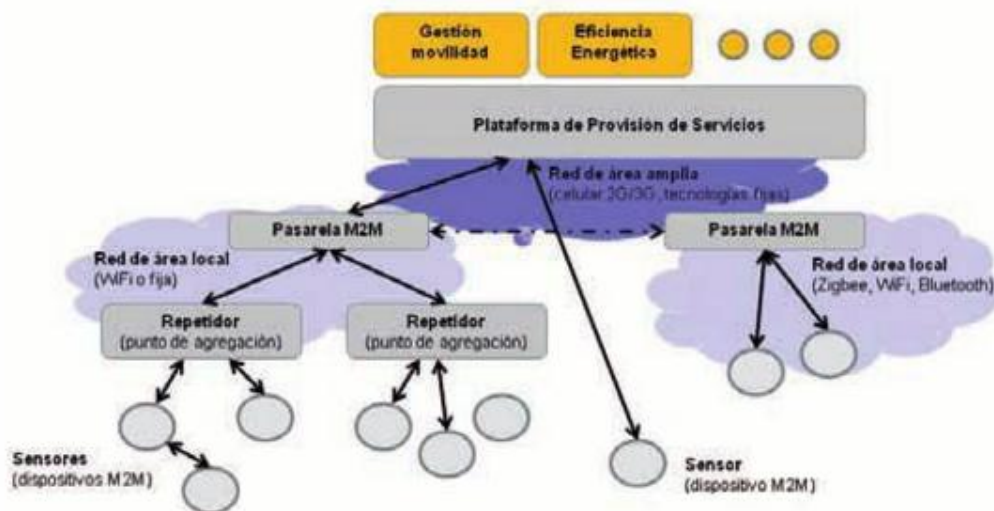


Figura 1-11: Arquitectura de alto nivel de tecnologías M2M de las Smart Cities

Fuente: (ESPAÑA. Telefónica, 2011 p. 61)

1.5 Almacenamiento de Datos

Los datos recolectados en una Smart City deben almacenarse en una plataforma para el tratamiento y análisis de los mismos. Debido a la cantidad de datos que se pueden recolectar en una ciudad, los proyectos realizados para Smart Cities cuentan con el apoyo de Big Date.

El uso de Big date ayuda al tratamiento de datos masivos en la red. Más adelante se describe la plataforma que se utilizara para el tratamiento de los datos del presente trabajo de titulación.

1.6 Plataformas de hardware libre para proyectos de Smart Cities

Existen diversas plataformas de hardware libre existentes en el mercado, cada uno con sus ventajas y desventajas. Todo depende del tipo de proyecto que se vaya a realizar y las características del mismo, además se debe tomar en consideración el entorno en el que se desarrollará el proyecto y la cantidad de datos que se va a recolectar.

1.6.1 Plataformas disponibles en el mercado

1.6.1.1 Arduino

Arduino es una plataforma electrónica como se puede ver en la *Figura 1-12*, se trata de una placa de código abierto basado en hardware y software fácil de usar, su trabajo es ayudar a las personas a realizar proyectos interactivos. (Arduino, 2016)

Arduino se ayuda de varios dispositivos que están disponibles en el mercado como sensores, actuadores, motores que con una programación previa hace posible desarrollar diversos proyectos útiles para la humanidad. (Arduino, 2016).

Las placas Arduino son de bajo costo y funcionan con una alimentación de 5v, para la programación de algún proyecto se utiliza el software de Arduino IDE que en su mayoría se limitan a Windows. Su programación es fácil y clara ideal para principiantes y grandes

desarrolladores. Además Arduino cuenta con una serie de modelos con diversas características acordes al proyecto que se vaya a realizar.



Figura 1-12: Placa Arduino

Fuente: (Arduino, 2016)

1.6.1.2 Wasmote

Es una placa de hardware libre como se puede ver en la *Figura 1-13* creada por la empresa Libelium, ésta placa se desarrolló específicamente para proyectos con sensores inalámbricos, su principal objetivo es el tiempo de vida del dispositivo que está valorado entre 1 y 5 años dependiendo del trabajo que se haya asignado a la placa.

Este dispositivo ya se encuentra disponible en el mercado pero tiene una desventaja, su costo es elevado, aunque ya existe una variedad de sensores en el mercado que son compatibles con esta placa, no son suficientes para la gran cantidad de sensores que se demandan y que Arduino si dispone. (Cooking hacks, 2012).



Figura 1-13: Wasmote-Wireless Sensor Networks

Fuente: (Cooking hacks, 2012)

1.6.1.3 Spark

Su principal característica es la facilidad de instalación y programación, su tamaño es reducido y cuenta con una placa incorporada de Wifi, lo que hace más fácil la conexión a internet. Las características inalámbricas de la placa se basan en el estándar 802.11, la desventaja de esta placa es que no tiene variedad de su producto. En la *Figura 1-14* se puede divisar el tamaño de la placa Spark.



Figura 1-14: Placa Spark

Fuente: (GARRIDO R., 2015 p. 30)

1.6.1.4 INTEL Galileo

Una placa que es compatible con Raspberry Pi y Arduino, basada en arquitectura INTEL como se puede ver en la *Figura 1-15*, cuenta con un microcontrolador IQuart SoCX1000 que funciona a 400Mhz, se puede desarrollar en plataformas como Linux, Mac, Windows. Su precio es relativamente elevado y por sus características se prefiere usar plataformas de más bajo costo.



Figura 1-15: Placa INTEL Galileo

Fuente: (GARRIDO R., 2015 p. 31)

A continuación en la *Tabla 1-3* se muestra una comparación entre las plataformas mencionadas anteriormente:

Tabla 1-3: Comparación entre plataformas más usadas de hardware libre

Plataforma	Ventajas	Desventajas	Micro	Memoria	Voltaje
Waspnote	Gran cantidad de sensores disponibles en el mercado.	Solo funciona para requerimientos específicos. Alto costo	ATMega 1281	128kb	3.3-4.2 V
Arduino	Bajo costo. Disponibilidad de sensores. Variedad de modelos.	Para proyectos más grandes talvez sea necesario acoplar a ciertas características.	ATMega38 2	32-256kb	5-9 V
Spark	Fácil de configurar Bajo costo	No hay mucha variedad en su marca.	ARM-32-Cortex	128kb	3-6 V
Intel Galileo	Compatible con Arduino	Alto costo. Por su similitud la mayoría prefiere Arduino.	Intel Quark.S0 Cx1000	512 Kb	7-15 V

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

1.7 Recolección de Desechos Sólidos en el Ecuador

1.7.1 Norma de calidad ambiental para el manejo y disposición final de desechos sólidos no peligrosos.

1.7.1.1 Introducción

Tomando en consideración la Ley de Gestión Ambiental y el Reglamento a la Ley de Gestión Ambiental para prevención y control de la contaminación ambiental. Las normas que se han tomado en consideración para el presente trabajo de titulación, son las más relevantes.

Esta norma establece criterios para manejar los desechos sólidos no peligrosos, desde que se originan hasta su disposición final. (ECUADOR. ESPOL, 2009)

1.7.1.2 Objetivos

- ✓ Prevenir y controlar la contaminación ambiental en lo que se refiere al recurso aire, agua y suelo.
- ✓ Conservar la integridad de las personas, ecosistemas y el ambiente en general.
- ✓ El manejo de los desechos sólidos no peligrosos se deberán realizarse bajo esta norma técnica.

1.7.1.3 De las responsabilidades en el manejo de los desechos sólidos

El Manejo de los desechos sólidos en todo el país será responsabilidad de las municipalidades, de acuerdo a la Ley de Régimen Municipal y el Código de Salud. Las municipalidades o personas responsables del servicio de aseo, de conformidad con las normas administrativas

correspondientes podrán contratar o conceder a otras entidades las actividades de servicio. (ECUADOR. ESPOL, 2009)

Es por ello que el presente trabajo de titulación ayudara a todos los municipios de todas las ciudades a optimizar este proceso de recolección de desechos.

1.7.1.4 De las prohibiciones en el manejo de desechos sólidos

Se prohíbe la quema de desechos sólidos en los contenedores de almacenamiento de desechos sólidos.

1.7.1.5 Normas generales para la recolección y transporte de desechos sólidos no peligrosos.

Las entidades encargadas del servicio de aseo, deben establecer la frecuencia óptima para la recolección y transporte, por sectores, de tal forma que los desechos sólidos no se alteren o propicien condiciones adversas a la salud tanto en domicilios como en los sitios de recolección.

Además el horario y las rutas de recolección y transporte de los desechos sólidos contenidos en los recipientes de almacenamiento, deben ser establecidos por las entidades encargadas del servicio, basándose en los estudios técnicos correspondientes. (ECUADOR. ESPOL, 2009)

Los contenedores de residuos deben estar a una altura suficiente de tal manera que se impida el acceso a los niños y animales domésticos.

1.7.2 Recolección de desechos sólidos: Tradicional vs. Inteligente.

La recolección de desechos sólidos consiste en el proceso de recoger la basura en la vía pública y realizar el traslado de esta a una planta para su disposición, incineración o selección. Actualmente se tiene un sistema tradicional de recolección que no aprovecha los beneficios de la IoT; y el sistema inteligente que en cambio va de la mano con las Ciudades Inteligentes a partir de sensores que ayudan a optimizar la recolección en sí.

Para conocer los beneficios de una recolección Inteligente de desechos sólidos, empezaremos viendo cómo funciona la Recolección Tradicional.

1.7.2.1 Recolección Tradicional de Desechos Solidos

En una recolección Tradicional de desechos sólidos se tienen varios elementos que son: Usuario, Contenedor de Basura, Carro Recolector, Trayectoria y Botadero de Basura, como se puede ver en la *Tabla 1-4*.

Tabla 1-4: Elementos básicos para la recolección tradicional de desechos

ELEMENTO	DESCRIPCION
Usuario	Es la persona o conjunto de personas que generan desechos de basura a partir del consumo de productos alimenticios, industriales o derivados. Estos desechos los deja en el Contenedor más cercano a su localidad
Contenedor de desechos	Es un recipiente de dimensiones considerables diseñado para acaparar toda la basura que genera un sector. Dependiendo del recolector de basura puede tener diferentes diseños pero su funcionamiento será el mismo.
Carro Recolector	Vehículo diseñado para transportar la basura que ira recolectando de todos los contenedores que exista en su trayectoria.
Trayectoria	Cada carro recolector se le asigna una ruta la cual debe cumplir y de esa manera ir despachando la basura que haya en cada uno de los contenedores.
Depósitos de desechos	Lugar de acopio de basura en donde los Recolectores realizan su descarga. Dependiendo del sistema de Reciclaje se ira separando la basura para su pronto reutilización.

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

1.7.2.2 Desarrollo de la recolección

Simple, al recolector le asignan la ruta y este realiza su recorrido desde un inicio hasta el final, de ahí todo el contenido lo despacha en el Botadero. En este caso el sistema presenta algunas falencias como son no saber que contenedores están vacíos y de esa manera evitar ir a ese sitio o caso contrario contenedores que están a máxima su capacidad y donde es necesario acudir continuamente.

1.7.2.3 Desventajas de este Método

- No existe una base de datos que me informe sobre la cantidad de basura que existe en el contenedor.
- Para comprobar el estado del contenedor se tiene que ir personalmente a donde se encuentra.
- No se dispone de un localizador que me indique Latitud y Longitud del contenedor en caso de pérdida.
- Habrá un consumo constante de gasolina y no se puede optimizar.

1.7.2.4 Recolección Inteligente de Desechos Solidos

La recolección inteligente de desechos sólidos involucra el uso de IoT, donde se tendrá información que genera el contenedor y de esa manera gestionarla desde una base de datos. A partir de sensores ubicados dentro de cada contenedor se puede monitorear los contenedores de manera remota para optimizar los recursos que se dispone al momento de realizar la recolección por parte de los recolectores.

Entre los eventos que se puede monitorear aplicando IoT están:

- Ver el estado de llenado del contenedor y ubicación vía GPS.
- Transmitir los datos a través de un canal de comunicación.
- Saber el estado físico del contenedor y posibles accidentes.

Antes de que el recolector realice su recorrido la base de datos realizará un análisis de que contenedores están llenos y cuales están aún vacíos, de esa manera realizara un trazo de ruta con los contenedores que necesariamente necesitan ser vaciados.

CAPÍTULO 2

2. DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO PARA MONITOREAR LA RECOLECCIÓN DE DESECHOS EN TIEMPO REAL

2.1 Introducción

En el capítulo se muestra la arquitectura utilizada para el diseño del prototipo, la selección de los elementos que dan forma al sistema y los pasos para integrar cada uno de los elementos y crear el prototipo que monitorea y controla la recolección de desechos sólidos.

Entre los elementos están los sensores, que evalúan los eventos, hardware libre para procesar la información, transmisor inalámbrico para enviar datos hacia el internet, hosting web con almacenamiento en la nube y su respectiva visualización para el usuario en la aplicación web.

El prototipo se implementó en 3 contenedores de desechos sólidos y se optó por utilizar una tecnología inalámbrica idónea para transmitir dentro de Smart Cities, debido a la manipulación de los contenedores de basura no se consideró factible usar comunicación por cable.

2.2 Diseño de la Arquitectura del Prototipo

Para empezar con el diseño, se determinó la arquitectura que proporcione una visión global de cómo se conforma todo el sistema, por lo tanto se empezó por establecer en bloques los procesos desarrollados en el prototipo, teniendo como resultado 5 bloques como se muestra en la *Figura 2-1*:



Figura 2-1: Bloques de Proceso

Fuente: Creado por los autores

Cada bloque tiene a su disposición uno o varios elementos dependiendo el proceso que realice como se detalla en la *Tabla 2-1*.

Tabla 2-1: Descripción de los Bloques de Proceso

BLOQUE	DESCRIPCIÓN
ENTRADA	Son los elementos encargados de recolectar información a partir de eventos que se han definido.
PROCESAMIENTO	Conformado por una plataforma de hardware libre que recibe los datos enviados por el Bloque de Entrada e implementa un algoritmo para filtrar y/o manipular los mismos.
SALIDA	Elemento que ayuda a transmitir los datos finales hacia la red WAN de la operadora celular e internet a través del medio que es el aire.
ALMACENAMIENTO	Los datos llegan al Hosting Web para ser almacenados.
INTERFAZ DE USUARIO	A partir de la aplicación web se procesara la información para obtener como resultado una interfaz amigable que muestre en tiempo real los eventos que se han detectado en el Bloque de Entrada.

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.3 Requerimientos para el diseño del prototipo

Con los bloques definidos se determinó los elementos disponibles para solucionar y satisfacer dichos procesos. En el mercado existen algunos elementos que se describieron en el capítulo anterior y a continuación se detalla los sensores y placas idóneos que se empleó para cada uno de los bloques.

2.3.1 *Bloque de Entrada*

Es la primera etapa del sistema que consiste en adquirir información para saber el estado en el que se encuentra cada contenedor. Se utilizaron sensores que captan los eventos en cada contenedor y los mismos fueron procesados en el Bloque de Procesamiento.

2.3.1.1 *Eventos*

La información se toma de las mediciones que realizan los sensores a partir de los eventos producidos en cada contenedor, siendo estos:

- Nivel de llenado
- Ubicación del contenedor
- Alerta de incendio
- Orientación en que se encuentra el contenedor

Cada evento es de gran importancia porque se mejora la situación en la que se encuentra el contenedor, siendo un apoyo a la recolección tradicional de desechos sólidos.

A continuación se muestra en la *Tabla 2-2* las características de los eventos y el respectivo sensor para la medición correspondiente.

Tabla 2-2: Descripción de los eventos y sensores.

EVENTO	DESCRIPCION	SENSOR
Nivel de llenado	Información del contenedor: vacío, medio o lleno	Ultrasónico
Localización	Localización del contenedor en latitud y longitud	Localización
Medir nivel de Humo	Desechos que se queme, se envía una alerta de incendio	Calidad del Aire
Orientación del contenedor	Hay situaciones donde el contenedor está en una posición incorrecta impidiendo su buen funcionamiento.	Acelerómetro/ Giroscopio

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.3.1.2 Sensores

Para la medición de los eventos es necesario el uso de sensores, los cuales están diseñados para captar y arrojar valores del evento medido.

Los sensores escogidos para medir los eventos son:

- Ultrasónico
- Localización
- Humo
- Acelerómetro/ Giroscopio

A continuación se explica el funcionamiento de cada uno de los sensores.



2.3.1.3 Sensor Ultrasónico

Son dispositivos encargados de medir la distancia de un objeto a través del envío de pulsos de ultrasonido que rebotan al chocar en la superficie del objeto y regresan; se mide el tiempo de retorno de la señal obteniendo como respuesta la distancia que se encuentra dicho objeto.

El sensor ultrasónico se propuso para detectar la cantidad de desechos que se encuentra en el contenedor a partir de niveles que son: lleno, medio o vacío. El valor se envía al hosting web para ser visualizado en la aplicación web.

En el mercado existen modelos de sensores ultrasónicos los cuales se describen en la *Tabla 2-3*.

Tabla 2-3: Cuadro Comparativo Sensores Ultrasónicos

Modelo	HY-SRF05	HC-SR04	US-100
Alimentación	5 VDC	5 VDC	5 VDC
Corriente	4 mA	15 mA	2 mA
Frecuencia	40 kHz	40 kHz	40 Khz
Cobertura	15°	15°	15°
Rango de Medida	1cm - 4m	2cm - 4m	2 - 450cm
Pines	VCC, Trig, Echo, OUT, GND	VCC, Trig, Echo, GND	VCC, Trig, Echo, GND, GND
Dimensiones	43mm x 20mm x 17mm	45mm x 20 mm x 15mm	44mm x 26mm x 14mm
Imagen			

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- Sensor HY-SRF05

De la *Tabla 2-3* se escogió el modelo HY-SRF05 al ofrecer un rango de cobertura amplio y disponibilidad en el mercado, como se muestra en la *Figura 2-2*.



Figura 2-2: Sensor Ultrasónico SRF05

Fuente: (ITALIA. Troniqa, 2014)

El sensor tiene la ventaja de ser de bajo costo, con rango de cobertura amplia y de tamaño reducido. Funciona con una alimentación VCC de 5VDC y GND para poder trabajar. Posee 2 capsulas ultrasónicas encargadas de disparar y recibir el eco. Para el manejo de los pulsos dispone de 2 pines conformado por: Trigger (Input) y Echo (Output).

Cada contenedor de basura necesita de un sensor HY-SRF05 que se ubicara con vista hacia abajo para detectar la basura que se almacena como se muestra en la *Figura 2-3*.



Figura 2-3: Sensor Ultrasónico con vista abajo

Fuente: (ESTADOS UNIDOS. ENEVO, 2011)

2.3.1.4 Sensores de Localización


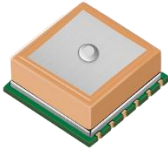

Para localizar un objeto en coordenadas de latitud y longitud se utilizó el módulo GPS, dispositivo que se encarga de mostrar la posición de un elemento a partir de un sensor incrustado que detecta los satélites que se encuentran alrededor del globo terráqueo.

Estos satélites envían información todo el tiempo y es procesada por el sensor para brindar información como datos, Hora GMT, Longitud, Latitud, Número de Satélites, Altitud, etc.

Se planteó este sensor para conocer la ubicación de cada contenedor en un mapa, permitiendo hacer un rastreo en tiempo real cuando el contenedor sea transportado a otro sitio.

En el mercado existen módulos que vienen integrados el sensor GPS, entre los modelos de sensores que se pueden adquirir lo detallamos en la *Tabla 2-4*.

Tabla 2-4: Cuadro Comparativo Sensores de Localización.

Modelo	NEO 6M	Quectel L80	EM-406
Alimentación Interna	2.7-3.6VDC	3.0 - 4.3 VDC	4.5 – 6.5 VDC
Corriente(Adquisición)	47mA	25mA	44mA
Señal recibida	GPS L1	Gps L1	GPS L1
Canales	50	66	20
Sensibilidad (Inicio, Readquisición y Tracking)	-156, -160, -161 dBm	-148, -160, -165 dBm	-159dBm
Velocidad Transmisión x defecto	9600 baudios	9600 baudios	4800 baudios
Pines Sensor	24	9	6
Presentación en	Modulo GPS Neo 6M	Módulo GPS3 CLICK Shield M2M	Solo
Pines Modulo	4	16 y 2	6
Dimensiones Sensor	16 x 12.2 mm	16 x 16 x 6.45 mm	30 x 30 x 10.5 mm
Dimensión Antena	22 x 22 mm	15 x 15 x 4 mm	25.20 x 25.20 mm
Imagen			

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- **Modulo GPS NEO 6M**

Se escogió el modulo GPS Neo 6M al disponer de número reducido de pines, adquisición inmediata en el mercado y su antena externa acoplable, como se muestra en la *Figura 2-4*.

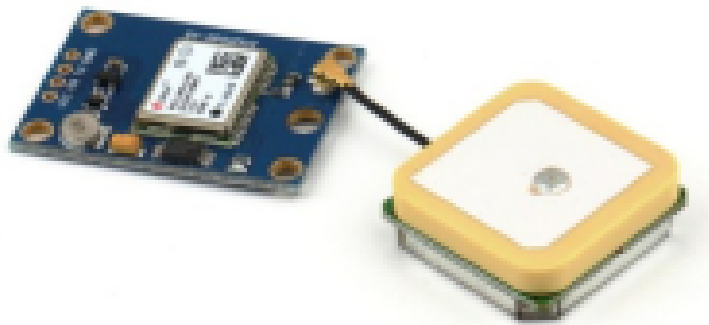


Figura 2-4: Módulo GPS NEO 6M.

Fuente: (GONZALES J.& MARIN C., 2015 pp. 40-41)

El módulo conocido como NEO 6M, está diseñado para operar con una alimentación de 3-5 voltios, posee dos pines TX y RX para transmitir y recibir datos.

Los beneficios que ofrece este módulo están su bajo costo, precisión de los datos al ser recibidos, tamaño reducido y compatibilidad con todo hardware libre del mercado. Cuando el modulo se sincroniza con los satélites y procesa la información recibida; un led indicador empieza a parpadear, indicando la disponibilidad de latitud y longitud.

La antena del módulo será colocada con vista hacia arriba para receptor la señal recibida por los satélites que lo rodean, como se ve en la *Figura 2-5*.



Figura 2-5: Sensor GPS con vista al cielo

Fuente: Creado por los autores

2.3.1.5 Sensores de Humo

Para medir los elementos presentes en el aire, existe un sensor especializado llamado Sensor de Gas, el cual mide la calidad que tiene el aire. Este sensor consiste en un pequeño bulbo llamado calentador electro-químico, sensible a una gama de gases.

El sensor es útil dentro del contenedor permitiendo detectar la presencia de humo y fuego que cause daños graves dentro del mismo.

El sensor de humo viene incrustado en un módulo para facilitar su manejo con hardware libre. En el mercado existe una familia de sensores que se encargan de detectar gases, los cuales se detalla en la *Tabla 2-5*.

Tabla 2-5: Cuadro Comparativo Sensores de Gases

Modelo	MQ-135	MQ137	MQ-3	MQ-9
Alimentación Interna	5 VDC	5 VDC	5VDC	5 – 15 VDC
Gases	Benceno, Alcohol, Humo	Amoniaco	Alcohol, etanol, fumo	CO, gases inflamables
Imagen				

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Módulo MQ-135

Se escogió el módulo MQ-135 al ofrecer la cualidad de detectar humo y presencia de fuego, como se ve en la *Figura 2-6*.



Figura 2-6: Módulo MQ-135.

Fuente: (THAOYU-ELECTRONICS, 2015)

Este sensor aloja salidas analógicas para ser interpretadas y monitoreadas con valores umbrales predeterminados, dependiendo del medio en donde se encuentre. Opera en un voltaje no mayor a 5V. Para cumplir su funcionamiento es necesario que este encendido en un rango de 12 a 24 horas para la estabilización de sus elementos con el medio. Se debe tener cuidado ya que el bulbo se mantiene encendido y puede causar quemaduras leves. El pin utilizado para trabajar es en modo Analógico o pin A0.

Su ubicación dentro del contenedor será con vista hacia abajo para detectar los gases que se produzcan en el fondo del mismo.




2.3.1.6 Sensores Acelerómetro/Giroscopio

Para detectar si un objeto está inclinado o ha sido movido se emplea el sensor Giroscópico y Acelerómetro. Un giroscopio está encargado de medir las velocidades angulares a partir de impulsos de rotación, mostrando los rangos en eje X, Y Z. Un acelerómetro en cambio mide la aceleración, inclinación o vibración. Emplea rangos que van de décimas hasta los miles de g.

Se propuso el sensor para saber cómo está orientado el contenedor de desechos, si se encuentra inclinado o en una posición inadecuada.

Entre los sensores de acelerómetro y giroscopio existentes, se tiene los siguientes en la *Tabla 2-6*.

Tabla 2-6: Cuadro Comparativo Sensores Acelerómetro/Giroscopio

Modelo	MPU-6050	MMA7361	ADXL335
Alimentación Interna	2.375 – 3.46 VDC	2.2 – 3.6 VDC	1.8 – 3.6 VDC
Corriente	140 uA	400 uA	350 uA
Temperatura	-40 - +85°C	-40 - +125°C	-40 - +85°C
Pines Sensor	24	14	16
Pines Modulo	8	10	7
Dimensión Sensor	4 x 4 x 0.9 mm	3 x 5 x 1 mm	4 x 4 x 1.45 mm
Imagen			

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- **Módulo MPU-6050**

Dada la tabla comparativa se escogió el modelo MPU-6050 al consumir niveles bajos de corriente y su disponibilidad en el mercado. Los ejes que usa el modulo se ve en la *Figura 2-7*.

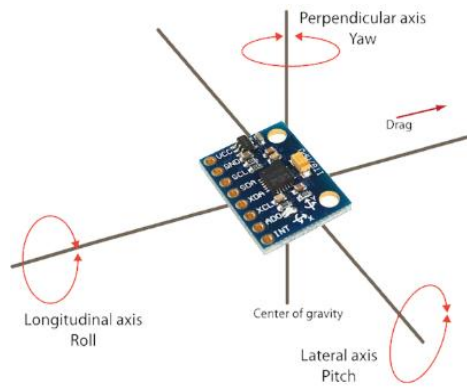


Figura 2-7: Módulo MPU-6050

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

El módulo MPU-6050 se caracteriza por cumplir las funciones de acelerómetro y giroscopio. El sensor es ideal para el diseño y control de robótica, medición de vibraciones, sistemas inerciales, medir caídas, etc.

El voltaje de alimentación va en un rango de 2.37 a 3.46 v. Ofrece 6 salidas digitales correspondientes a los ejes x, y z de acelerómetro y giroscopio. Los pines encargados de enviar la información son SCL, SDA e INT.

2.3.2 *Bloque de Procesamiento*

Corresponde a la segunda etapa del sistema en donde se empleó hardware libre encargado de recolectar la información que envían los sensores para luego procesarla, filtrarla y finalmente preparar los datos que se enviarán hacia el internet.

El hardware libre tiene la misión de ejecutar algoritmos que controlen cada uno de los sensores que recolectan la información.

Para esta etapa se escogió como hardware libre la plataforma Arduino por poseer las características como: bajo precio, disponibilidad en el mercado, compatibilidad con los sensores ya mencionados anteriormente, y una comunidad en línea presta a ayudar con información.

2.3.2.1 *Hardware Libre*

Constituyen aquellos dispositivos hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, adquiriéndose de manera gratuita o con algún costo.

Entre las plataformas que forman parte de este grupo están:

- **Arduino**
- **Waspote**
- **Intel Galileo**
- **Spark**






De las plataformas mencionadas nos centraremos en la plataforma Arduino, escogida para el diseño del sistema.

2.3.2.2 *Plataforma Arduino*

Arduino es una plataforma de código abierto construida para facilitar la creación de prototipos electrónicos, para ello se basa en el uso de un dispositivo hardware y un programador Software. El dispositivo Hardware consta de una placa que contiene un microcontrolador el cual se le envía una serie de instrucciones por medio del Programador de Arduino, más conocido como Arduino IDE a través del lenguaje de programación C.

Con el pasar de los años Arduino ha sido la cabeza de miles de proyectos siendo apoyado por una gran comunidad como estudiantes, aficionados, programadores, profesionales e inclusive empresas como son Intel, Telefónica, entre otras. Arduino para satisfacer las necesidades de los consumidores ha creado líneas de modelos los cuales se detalla en la *Tabla 2-7*.

Tabla 2-7: Cuadro Comparativo de Modelos Arduino

Modelo	MEGA 2560	UNO	101	NANO	YUN
Microcontrolador	ATmega2560	ATmega328P	Intel Curie	ATmega168 o ATmega328	ATmega32U4
Alimentación	7 – 12 VDC	7 - 12 VDC	7 – 12 VDC	7 – 12 VDC	5 VDC
Corriente	20 mA	20 mA	20 mA	40 mA	40 mA
Pines Digitales	54	14.	14	14	20
Pines PWM	15	6	4	0	7
Pines Analógicos	16	6	6	8	12
Puerto Serial	4	1	1	1	1
Observación	Conector USB, Power Jack ICSP Boton de Reset	Conector USB, Power Jack ICSP Boton de Reset	Conector USB, Power Jack ICSP Boton de Reset	Conector mini-USB, Boton de Reset	Microprocesador Linux, Ethernet, Wifi, USB, Lectora SD, RAM y Flash Memory, 3 Botones de Reset
Memoria	256 KB	32 KB	196 KB	16, 32 KB	32 KB
Dimensiones	101.52 x 53.3 mm	68.6 x 53.4 mm	68.6 x 53.4 mm	45 x 18 mm	73 x 53 mm
Peso	37 gr	25 gr		5 gr	32
Imagen					

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Dada la tabla y a partir de las especificaciones descritas se decidió adquirir el Arduino Mega 2560 como placa de facto para procesar los datos enviados desde los sensores.

2.3.2.3 Arduino Mega 2560

Se escogió la placa Arduino Mega 2560 al ofrecer mayor capacidad de memoria, ítem crucial por el espacio de programación que se demanda.

En la *Figura 2-8* observamos la placa Arduino Mega.



Figura 2-8: Arduino Mega.

Fuente: (Arduino, 2016)

La placa está basada en el Microcontrolador ATmega1280, que ofrece una memoria de 256 KB para guardar la programación, del cual 8KB se usa para el bootloader, así también 8KB para SRAM y 4KB de EEPROM.

La alimentación de la placa se lo realiza por fuente USB o conexión externa a través de un puerto Jack. El rango de voltaje que soporta va desde los 6 hasta 12 voltios gracias al regulador de voltaje en su interior. Si se le proporciona voltajes menores a 7V la placa se torna inestable, caso contrario si supera los 12V el regulador de voltaje se calienta causando daños en la placa.

El Arduino Mega 2560 está formado por 54 pines ubicados en el perímetro de la placa con un diseño fijo para acoplamiento con shields. Los pines trabajan de manera digital como entrada y salida aunque hay un número de pines que trabajan como entradas analógicas.

2.3.2.4 Esquema de Pines

Los pines disponibles para usar en la placa son en total 54 distribuidos en grupos que son:

- Power
- Digital
- PWM

- Comunicación
- Analógico
- Power

Corresponde a los pines que ofrecen alimentación para otros elementos o al propio Arduino, se encuentran ubicados como se muestra en la *Figura 2-9*.

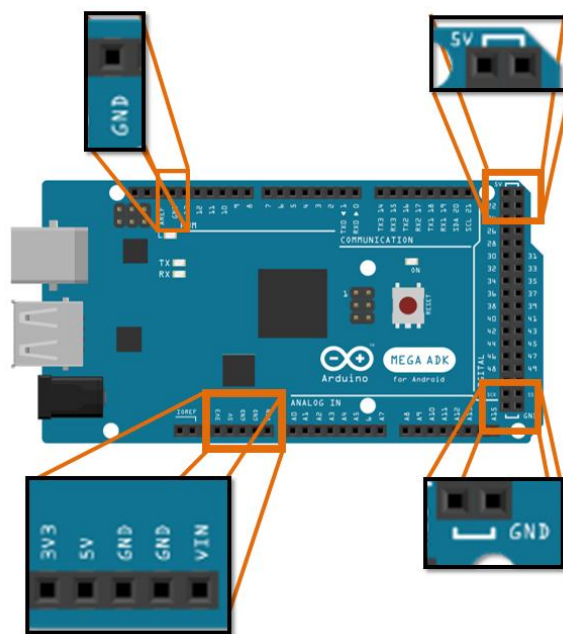


Figura 2-9: Pines Power.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- Digital

Pines destinados para funcionar como entrada o salida de información digital, y su ubicación se muestra en la *Figura 2-10*.

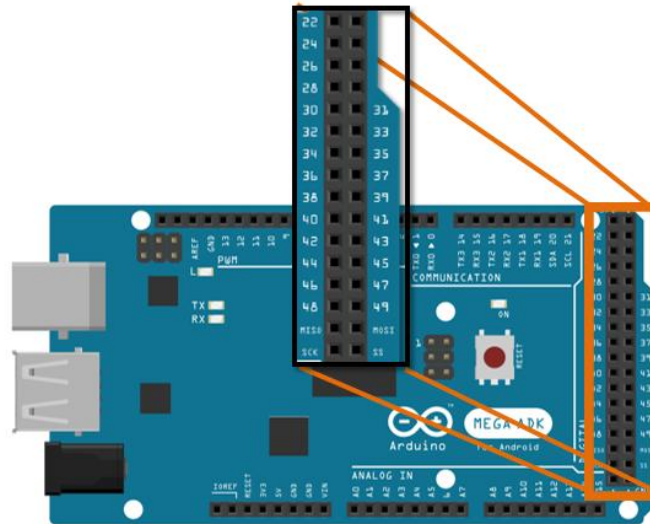


Figura 2-10: Pines Digitales.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- PWM

Pines digitales con la función específica de generar pulsos PWM de 8 bits como se muestra en la *Figura 2-11*.

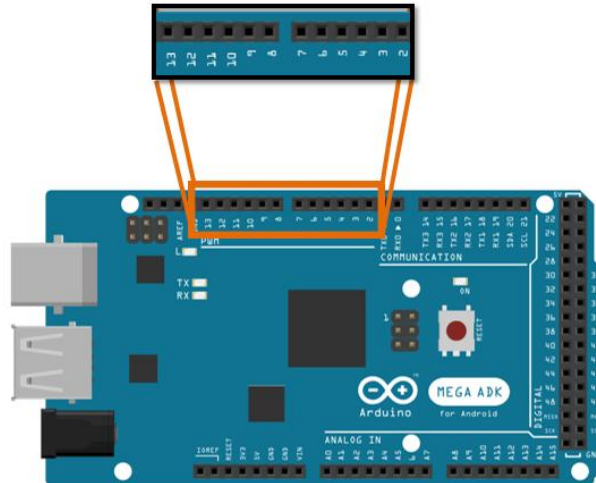


Figura 2-11: Pines PWM.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- Comunicación

Pines digitales destinados a la transmisión y recepción de datos con otros elementos, su ubicación está dada por la *Figura 2-12*.

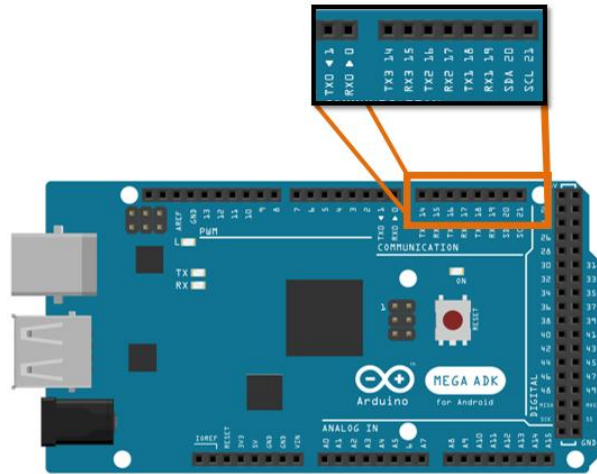


Figura 2-12: Pines de Comunicación.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- Analógico

Pines de entrada que detectan señales analógicas. Por el momento no se dispone de pines de salida analógica.

Se encuentran ubicados según muestra la *Figura 2-13*.

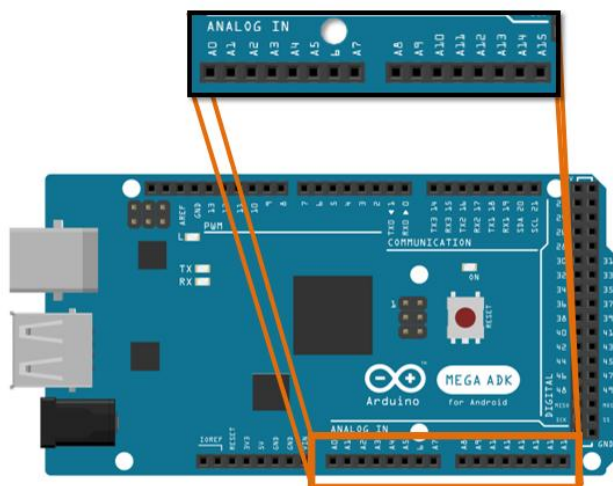


Figura 2-13: Pines Analógicos.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.3.3 *Bloque de Salida*

La tercera etapa de nuestro proyecto está bajo las manos del módulo que maneje una tecnología inalámbrica permitiendo transmitir los datos procesados de la placa Arduino hacia el Internet. Al enlazarse al internet, el dispositivo se convierte en un cliente más dentro del mundo de la WWW, es aquí donde se encaminan los datos hacia el Hosting Web, donde se dispone de los recursos para ser almacenado y luego visualizado en línea.

De las tecnologías inalámbricas para una Smart City mostradas en el capítulo anterior se optó por utilizar la tecnología GPRS. Se escogió esta tecnología dado que su cobertura cubre una extensa área, su infraestructura se encuentra implementada por la operadora celular y los datos a transmitir no demandan una velocidad alta, por lo que la velocidad usada en GPRS cubre los requisitos planteados.

2.3.3.1 *GPRS*

GPRS del inglés General Packet Radio Service, es una tecnología de transmisión de datos móviles perteneciente a 2.5 G. Se basa en la conmutación de paquetes en comparación a GSM que utiliza conmutación de circuitos, ofreciendo una mayor velocidad de datos con respecto a GSM.

GPRS nace como respuesta a la evolución de las telecomunicaciones hacia una comunicación móvil de internet. Esta tecnología utiliza la infraestructura ya desplegada de GSM con el plus de agregar nuevos elementos como soporte de Nodo GSN y actualización interna de Software. Con GPRS se dio lugar al boom de IPs y gracias al éxito que lo tuvo se ha ido mejorando y aumentando la velocidad de conexión llegando a la actualidad con la tecnología LTE Advanced.

2.3.3.2 *Arquitectura GPRS*

Al utilizar la misma infraestructura GSM solamente se aplicaron algunos cambios como actualización y adición de nuevos elementos, permitiendo una mejora en la trasmisión entre usuarios, red interna y el Internet, como se lo puede ver en la *Figura 2-14*.

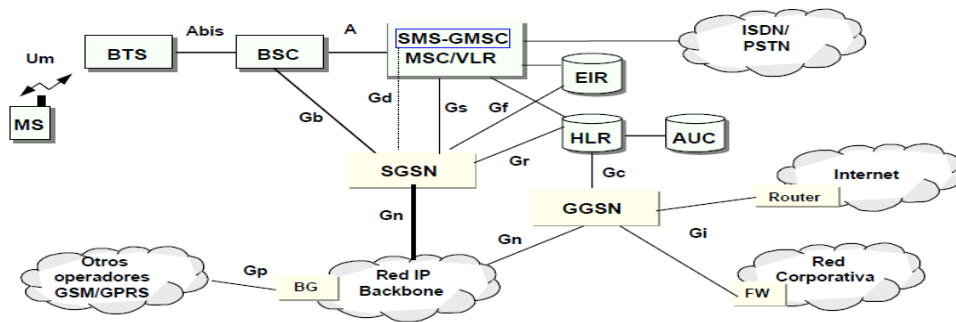


Figura 2-14: Arquitectura GPRS.

Fuente: (ESPAÑA. UPM p. 8)

Los elementos agregados a GPRS fueron:

- SGSN: Encamina paquetes IP entrantes y salientes, maneja el cifrado y autenticación como la facturación.
- GGSN: Router que encamina paquetes hacia el Internet.
- BG: Interconecta GSN de otros operadoras.

2.3.3.3 Asignación IP

La red GPRS crea una subred con todos los abonados que desean tener acceso al internet, los cuales se les configura los siguientes datos:

- Cada terminal móvil se les asigna una dirección IP
- La red interna utiliza un DNS local.
- Se requiere APN para poder entrar en la red celular.
- La asignación IP puede ser transparente o utilizando un pool específico de IPs según el nodo

2.3.3.4 Protocolos e Interfaces GPRS

GPRS maneja diferentes protocolos entre cada uno de sus elementos, los cuales se destacan en la *Figura 2-15*.

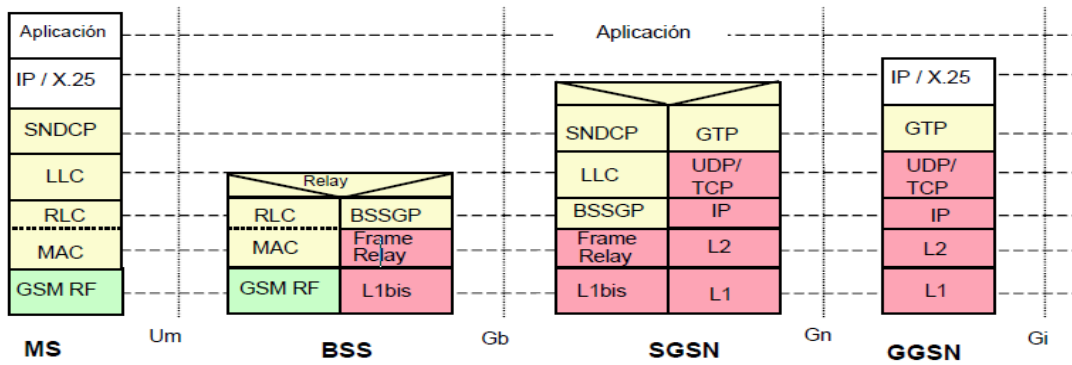


Figura 2-15: Protocolos GPRS.

Fuente: (ESPAÑA. UPM p. 16)

Como se puede observar en cada sección el modelo IP se mantiene, demostrando que es una tecnología útil dentro de IoT.

2.3.3.5 Burst GPRS

Para su operación en GSM, GPRS utiliza un canal físico exclusivo llamado PDCH, con el mismo perfil de potencia que GSM y la misma modulación.

Cada canal de portadora tiene 200KHz que transporta 8 canales TDMA, cada uno con 8 Timeslots. Un timeslot o burst consiste en la parte útil de datos a ser transmitidos.

El Burst o Timeslots de GPRS contiene 156.25 bits y está conformado por los siguientes elementos como se muestra en la *Figura 2-16*.

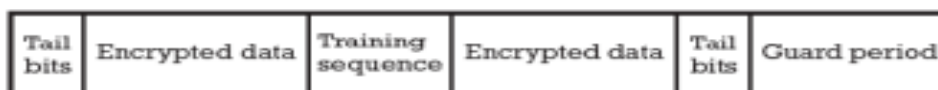


Figura 2-16: Figura 2-15. Burst GPRS.

Fuente: (BERNAL I., 2007 p. 24)

Cada uno de estos elementos cumplen una función específica, que consiste en:

Tail bits: Contiene 3 bits encargados de identificar al receptor que se envía el Burst.

Encrypted Data: Conformado por 116 Bits encriptados.

Training Sequence: 26 bits usados para la ecualización del canal de radiofrecuencia.

Guard Period: Contiene 8.25 bits y se usan para anunciar el fin del timeslot.

2.3.3.6 Tipos de Terminales

Al crearse GPRS se definieron los tipos de terminales con sus compatibilidades entre GSM y GPRS, las cuales se muestran en la *Tabla 2-8*.

Tabla 2-8: Tipos de Terminales

Clase	GSM y GPRS	Degradación
A	Simultaneo	No hay
B	No Simultaneo	GPRS sufre QoS
C	Manual	Si usa GPRS se desactiva llamadas GSM

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.3.3.7 Esquema de Codificación

En GPRS se tiene algunos esquemas que aplican los terminales para identificar su tasa de transmisión, como se muestra en la *Tabla 2-9*.

Tabla 2-9: Esquema de Codificación





Esquema	Velocidad (kbps)
CS1	9,05
CS2	13.4
CS3	15.6
CS4	21.4

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.3.3.8 Módulo GPRS

Con la tecnología GPRS explicada se procedió a comparar modelos de Shield GPRS que permitan interconectar el Bloque de Procesamiento con el Internet, entre los ejemplos existentes se detallan en la *Tabla 2-10*.

Tabla 2-10: Modelos GPRS para Arduino

Modelo	Shield GSM SIM900	ICOMSAT	Shield M2M	GSM Shield
Fabricante	SIMCOM	Itead Studio	MCI Electronics	Arduino
Procesador	Sim900	Sim900	Quectel M66	Quectel M10
Alimentación	4.8 – 5.2 VDC	4.5 – 5.5 VDC	5 VDC	5VDC
Corriente	50- 450 mA	500 – 2000 mA	40 – 1800 mA	700 – 1000 mA
Bandas	Quad-Band	Quad-Band	Quad-Band	Quad-Band
Frecuencias	850, 900 1800 y 1900 MHz	850, 900 1800 y 1900 MHz	850, 900 1800 y 1900 MHz	850, 900 1800 y 1900 MHz
Clase	4 y 1	4 y 1	4 y 1	
Velocidad Subida y Bajada	42.8/85.6 Kbps	42.8/85.6 Kbps	85.6 Kbps	85.6 Kbps
Funciones	SMS, Llamadas, Datos	SMS, Llamadas, Datos	SMS, Llamadas, Datos	SMS, Llamadas, Datos
Soporta	TCP/HTTP, FTP	TCP/HTTP, FTP	TCP/ UDP/ PPP/ FTP/ HTTP/ SMTP/ CMUX/ SSL	TCP/UDP, HTTP
Puerto de Antena	Si	Si	Si	No
Puerto de Micrófono y Altavoz	Si	Si	No	No
Selector de Serie	Si	Si	Si	No
Indicadores Led	PWR, Status, Net	PWR, Status y Net	PWR, Net	On, Status, Net
Dimensiones	110 x 58 x 19 mm	77.2 x 66 x 1.6 mm	68.6 x 53.3 mm	101.6 x 76.2 mm
Imagen				

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.3.3.9 *Shield GSM/GPRS SIM900*

De los productos mencionados se escogió el Shield GSM Sim900 como se muestra en la *Figura 2-17* por tener conexión TCP y FTP, útil para transmitir la información hacia el internet. También por su reducido consumo de corriente en comparación con la competencia.



Figura 2-17: Shield GSM/GPRS SIM900.






Fuente: (Zou W., 2013)

El shield utiliza el procesador SIM900 encargado de realizar las tareas encaminadas a GSM y GPRS, perteneciendo a la Clase B y esquema 1 y 4. El diseño de la placa esta para ser compatible con la Plataforma Arduino. El control de la placa se los realiza a través de la interfaz UART con el uso de comandos AT. Sus puntos fuertes son la comunicación M2M sin la intervención del usuario, manejo remoto de aplicaciones por SMS, o un rastreo en tiempo real de un objeto enviando los datos de GPS por internet.

2.3.4 *Bloque de Almacenamiento*

Para el cuarto bloque el cual se encuentra en el internet, se dispone de un lugar 24/7 donde se alojan los datos enviados por el Shield GPRS, es decir un Hosting Web donde puedan ser procesada la información por la Aplicación Web. Para ello se necesita disponer de un Dominio Web. El Dominio Web corresponde a la URL por la cual el cliente ingresara para observar los datos.

Tabla 2-11: Hosting de Pago y Gratuito

	GRATUITA		PAGO		
Sitio Web	Eshost	Hostinger	Eshost	GoDaddy	Hosting de Pago
Imagen					
Dominio	Controlada	Controlado	Libre	Libre	Libre
Dirección	http://eshost.com.ar/	http://www.hostinger.es/	http://eshost.com.ar/	https://es.godaddy.com/	https://www.hostingde pago.com/
Espacio de Disco Duro	1000 MB	2GB	5GB	100GB	5GB
Transferencia Mensual	10 GB	100GB	300GB	Ilimitado	Ilimitado
Soporta	FTP, PHP, MySQL, Correo	FTP, PHP, MySQL, Correo	PHP y MySQL	Correo, FTP, MySQL, PHP, Javascript, Apps	Correo, FTP, MySQL, PHP, Javascript, Apps
Publicidad	No	No	No	No	No
Costo Mensual	0 \$	0\$	2.50\$	1.00 \$	3.95\$

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

De los sitios mencionados se decidió escoger el servidor de Eshost al ser gratuito por motivo de costos y por las prestaciones ofrecidas como son compatibilidad con HTML y PHP, que son de gran importancia para tratar los datos enviados por la Shield GSM/GPRS.

2.3.4.2 Dominio Web

Corresponde al nombre único que se le asigna a la página web contenida en un hosting web. El dominio es una manera de hacer de la navegación en internet más sencilla y accesible, en comparación al estar manejando direcciones IP las cuales son complicadas al ser un conjunto de números para un sitio. El dominio funciona de la mano de los Servidores DNS que están

encargados de traducir internamente el nombre a su dirección IP, proceso transparente que no necesita de la intervención del usuario.

Para obtener un dominio se debe recurrir a las páginas web que ofrecen el servicio de Hosting, ellos vienen incluidos con un buscador de dominios como en la *Figura 2-19* que facilita comprobar si el nombre está disponible y las respectivas extensiones com, net, co, entre otras.

Estados Unidos - Español USD 24/7 Asistencia técnica (480) 463-8300 Ayuda Iniciar sesión Cupones GoDaddy Pro

Estados Unidos GoDaddy Todos los productos

Eres elegible para un dominio gratis.

Encuentra tu nombre de dominio perfecto

.com

Buscar

> No quiero mi dominio gratis.

Figura 2-19: Buscador de Dominios de Go Daddy

Fuente: (RUBEN A., 2014)

2.3.4.3 Protocolo FTP

Del inglés, File Transfer Protocol, es el protocolo más usado para intercambiar archivos en el mundo del internet, utiliza la interfaz TCP/IP y la arquitectura cliente-servidor para compartir la información. FTP se lo emplea para compartir archivos vía remota con total seguridad ya que se comprueba la llegada de los datos.

2.3.4.4 Trama FTP

El tren de bits utilizado por FTP consiste en 2 secciones q son:

- Header: Descriptor y Byte Count
- Marker

La sección Header está formada por el Descriptor y Byte Counter. El primero tiene 8 bits y es el código que identifica como se van a utilizar la información que se envía. En cambio Byte Counter de 16 bits indica cuantos bytes de información van a ser enviados. La sección Marker consiste en los Bytes a ser enviados cada Byte contiene 8 bits.

La representación de la información enviada por FTP está en la *Figura 2-20*.

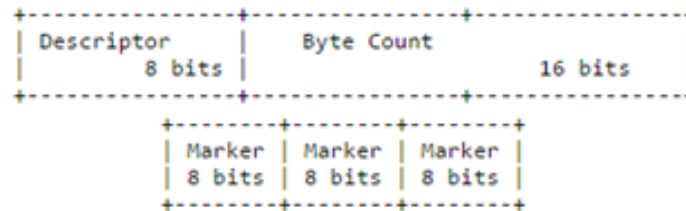


Figura 2-20: Trama FTP

Fuente: (RUBEN A., 2014)

2.3.5 Interfaz de usuario

Corresponde al último bloque de nuestro sistema, encargado de visualizar los datos enviados por los sensores con una interfaz amigable al usuario que es la Aplicación Web. Es decir aquí los datos que se tengan serán analizados en condiciones para un mejor entendimiento del cliente, es decir mostrando información adecuada a partir de los datos enviados por el Shield GSM/GPRS.

Para realizar este proceso se necesita usar algunos lenguajes de programación web como son: HTML, PHP y Javascript, los cuales permiten procesar la información y generar un registro de los cambios en tiempo real.

Cada lenguaje cumple una función dentro del sistema, que corresponde así:

- HTML: Lenguaje básico y responsable de mostrar los demás lenguajes en la página web.
- PHP: Inspecciona los datos que han llegado de los contenedores y realiza operaciones que HTML no puede hacer.
- JavaScript: Otorga interactividad a la página web en conjunto con PHP.

Estos lenguajes están escritos dentro de archivos o páginas web, cada uno con una función definida. Cuando el usuario ingrese a nuestro sistema con el Dominio Web, uno de estos archivos llamado index.php inicia con la visualización de la Aplicación Web.

2.4 Fuente de Alimentación



Con el prototipo ubicado dentro de un contenedor de desechos, se deduce que el sistema debe ser netamente autónomo. Se considera por lo tanto alimentar el sistema con una batería que cubra el rango de voltaje y corriente que se necesita para que funcione correctamente. En la *Tabla 2-12* se muestra las posibles formas de alimentar el sistema, disponibles en el mercado.

Tabla 2-12: Ventajas y Desventajas de las baterías para alimentar proyectos con arduino.

Fuente de Alimentación	Ventajas	Desventajas
 Pila de 9v	Fáciles de encontrar en el mercado. Disposición de cables y portapilas.	Baja densidad energética. Corriente máxima muy baja. Útil para proyectos pequeños.
 4 pilas AA de 1.4 V	Fáciles de encontrar. Carga superior a las pilas de 9V. Intensidad hasta 1A. Bajo costo	Al no ser recargables a largo plazo no resulta económico, las pilas AA recargables a su contrario a largo plazo es económico pero no brindan la intensidad necesaria para proyectos grandes.
 2 baterías litio 18650 de 3.7 V	Alta capacidad de carga. Son recargables.	Costo un poco elevado. El uso incorrecto de estas baterías puede ser peligroso. Para motores y servos es un voltaje muy elevado, se debe usar reguladores de voltaje.
 Banco de baterías	Recargables. Se puede alimentar diversidad de dispositivos,	Costo elevado No apto para proyectos grandes.

Continuará...

Continúa...

	como: motores, sensores, servos, etc...	
 Lipo	Densidad de energía más alta. Proporcionan enormes intensidades.	Son más caras. Peligrosas usar
 Batería Seca	Buena densidad de energía. Voltajes variables.	No ideales para proyectos pequeños, por su peso físico.

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.5 Implementación

Con los bloques del sistema descritos se procede a configurar cada elemento para formar el modulo que recolectara la información y visualizara en la Aplicación Web.

2.5.1 Configuración de los dispositivos.

Las primeras pruebas de funcionamiento se realizaron en una Protoboard, para constatar el buen funcionamiento de los elementos.

2.5.1.1 Bloque de entrada

En la *Figura 2-21*, se muestra el conjunto de sensores que componen el bloque. Para su buen funcionamiento se consideró la cantidad de voltaje necesaria para cada uno de los sensores, permitiendo alargar la vida de los mismos y evitando posibles fallas del sistema.

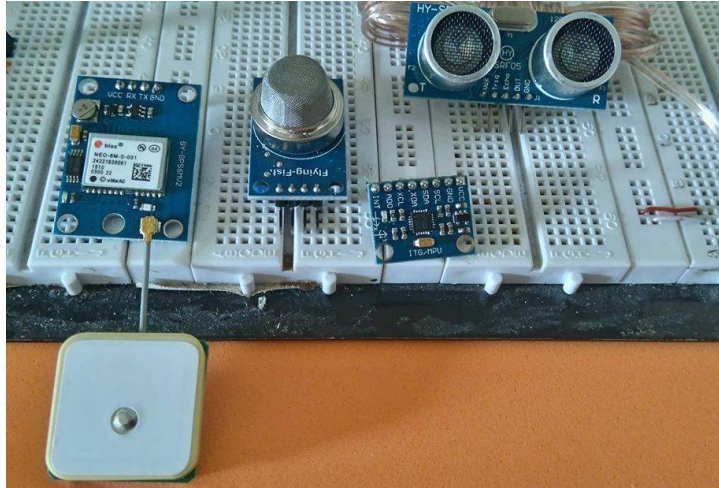


Figura 2-21: Bloque de entrada

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Cada de uno de los sensores está conectado al hardware libre Arduino Mega 2560, con la siguiente distribución de pines como se muestra en la *Tabla 2-13*:

Tabla 2-13: Conexión de sensores con arduino Mega.

	PINES SENSOR	PINES ARDUINO MEGA
Ultrasónico HY-SRF05	Vcc	5v
	Gnd	Gnd
	Trig	4
	Echo	5
MQ-135	Vcc	5v
	Gnd	Gnd
	A0	A0
GPS NEO 6M	Vcc	5v
	Rx	52
	Tx	53
	Gnd	Gnd
MPU-6050	Vcc	5v
	Gnd	Gnd
	SCL	21
	SDA	20
	INT	2

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.5.1.2 Bloque de procesamiento y Salida

Es la segunda etapa, aquí se compila todo el programa del sistema. Para la configuración de los sensores se trabaja en un entorno de desarrollo que utiliza la placa de hardware libre llamado Arduino IDE como se muestra en la *Figura 2-22*. El entorno de desarrollo está basado en lenguaje de programación.

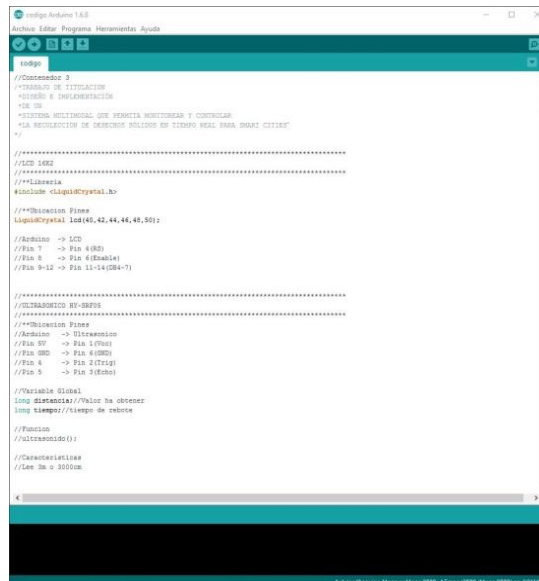


Figura 2-22: IDE Arduino

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

En el bloque de salida está el shield GSM/GPRS, la misma que transmite los datos hacia el bloque de almacenamiento. El shield dispone de una ranura SIM en donde se coloca un Chip, como se muestra en la *Figura 2-23*. Para lograr la transmisión hacia el internet se contrató un paquete de datos móviles económico en la operadora CNT.



Figura 2-23: Tarjeta SIM para Shield GSM/GPRS

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Para el buen funcionamiento del shield GSM/GPRS y la placa Arduino, se debe conectar la placa Arduino a una fuente externa debido a la demanda de corriente cuando se enciende por primera vez, si no recibe tal corriente el shield se apaga y no cumplirá las funciones requeridas.

El shield viene de fábrica para ser encendido manualmente a través del botón Power ubicado cerca de la salida de Micrófono y Altavoz como se ve en la *Figura 2-24*.



Figura 2-24: Botón encendido.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Para iniciar el encendido por código se procede a soldar 2 puntos del shield que se encuentra localizado cerca del Selector Serial como se muestra en la *Figura 2-25*.

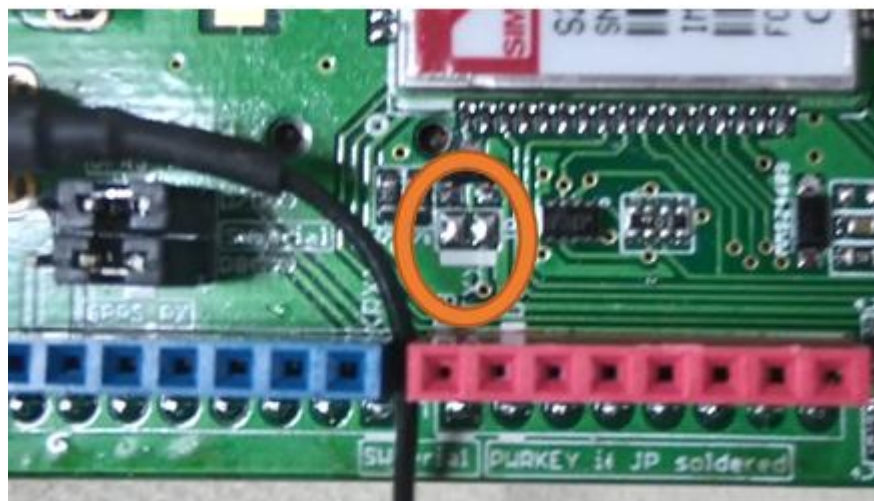


Figura 2-25: Puntos para soldar

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Para la comunicación entre el shield y Arduino se posee un selector serial según muestra la *Figura 2-26* para la comunicación vía software o hardware, cuyos pines son D7, D8 y D0, D1 respectivamente.



Figura 2-26: Selector Serial

Fuente: (GEEETECH, 2014)

Las conexiones del shield GSM/GPRS se realizó bajo la siguiente distribución de pines como se observa en la *Tabla 2-14*:

Tabla 2-14: Distribución de pines Shield GSM/GPRS

Arduino	GSM
5V	5V
GND	GND
Pin 0(RX)	Pin 0
Pin 1(TX)	Pin 1
Pin 9	Pin 9

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

- APN

Para que el shield se conecte al internet es necesario conocer el APN de la Operadora Celular. Un APN es el Punto de Acceso a la Red que le indica a un móvil que camino virtual usar para conectarse al internet. APN está formado por 3 datos que son:

- APN
- Usuario
- Contraseña

En Ecuador se tiene 4 operadoras con sus respectivos APN como se demuestra en la *Tabla 2-15*:

Tabla 2-15: APN, usuario y contraseña de operadoras celulares en Ecuador

OPERADOR	APN	Usuario	Contraseña
CNT	internet.cnt.net.ec	No tiene	No tiene
Movistar	internet.movistar.com.ec	No tiene	No tiene
Claro	internet.claro.com.ec	No tiene	No tiene
Tuenti	internet.tuenti.ec	No tiene	No tiene

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Para el sistema se utilizó los servicios de la operadora CNT, contratando un plan de datos móviles económico.

✓ **Comunicación con la Aplicación Web**

Para enviar los datos al internet el shield maneja el protocolo FTP

✓ **Comandos AT**

Con el shield conectado al Arduino se procedió a controlarlo a través de comandos AT los cuales se comunican por los puertos seriales escogidos previamente.

Los comandos AT consisten en ordenes que la placa Arduino envía al shield y este realizara las funciones que indique el comando. Así mismo es necesario un tiempo prudencial para que el shield responda con un mensaje de confirmación. Los comandos AT se han dividido en 3 partes, que son:

✓ **Configurar shield para conectarse al internet**

Los comandos que permiten conectarse al internet, se describen en orden en la siguiente *Tabla 2-16*:

Tabla 2-16: Comandos AT para conectarse al internet

COMANDOS AT	RESPUESTA	TIEMPO ESPERA (segundos)	DESCRIPCION
AT	OK	2.6	Comprobar que el dispositivo acepta comandos AT. OK indica que acepta comandos AT
AT+CPIN?	+CPIN: READY	2.6	Ver estado del PIN de la Tarjeta SIM. Ready indica que esta lista para ser usada
AT+CFUN?	+CFUN: 1	2.6	Consultar nivel de funcionalidad de Shield. 1 Indica que esta en Funcionalidad Completa
AT+CCALR?	+CCALR: 1	2.6	Consultar si vale hacer llamadas. 1 indica que las llamadas están activadas
AT+CIPSHUT	Shut OK	10.1	Cierra sesión GPRS. Shut OK indica que se desconectó de la red GPRS
AT+CGATT=0	OK	10.1	Desconecta de GPRS para liberar recursos OK indica que se liberó recursos que usaron para GPRS
AT+CGATT=1	OK	10.1	Conecta con GPRS. OK indica que se conectó a la red GPRS
AT+CSQ	+CSQ: 13,0	2.6	Muestra calidad de señal. 13 indica que se tiene -87 dBm que es una señal buena.
AT+CREG?	+CREG: 0,5	2.6	Muestra el estado de registro y acceso a la red celular 0 indica que no se ha registrado y 5 indica que encontró la red HSUPA.
AT+CSTT?	+CSTT: "CMNET", "", ""	2.6	Consultar APN actual. CMNET indica que no se ha registrado ninguna APN

Continuará...

Continúa...

AT+SAPBR=3,1,"C ONTYPE","GPRS"	OK	2.6	Primera configuración GPRS con APN, usuario y contraseña. OK indica que los datos han sido aceptados
AT+SAPBR=3,1,"A PN","internet.cnt.net .ec"	OK	2.6	
AT+SAPBR=3,1,"U SER",""	OK	2.6	
AT+SAPBR=3,1,"P WD",""	OK	2.6	
AT+SAPBR=1,1	OK	2.6	Enganchando a la red GPRS. OK indica que se enlazo correctamente
AT+CSTT="internet .cnt.net.ec", "", ""	OK	2.6	Segunda configuración GPRS. OK indica que los datos han sido aceptados
AT+CIPSRIP=1	OK	2.6	Permitir que el dispositivo tenga una IP y puerto cuando reciba datos. OK indica que se procesó correctamente el comando
AT+CIICR	OK	2.6	Registrando a la red. OK indica que se procesó correctamente el comando
AT+CIFSR	10.186.94.52	5.1	Mostrar IP actual. Se muestra la IP local

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

✓ Configurar shield para Autenticación FTP

Para subir los datos al hosting web se necesita definir la cuenta FTP como se muestra en la *Tabla 2-17*:

Tabla 2-17: Comandos para Autenticación FTP

COMANDOS AT	RESPUESTA	TIEMPO ESPERA (segundos)	DESCRIPCION
AT+FTPCID=1	OK	2.6	Perfil de FTP.
AT+FTPTYPE="A"	OK	2.6	Tipo de Sesión.
AT+FTPSERV="ftp.eshost.com.ar"	OK	2.6	Host FTP
AT+FTPPORT=21	OK	2.6	Puerto FTP

Continuará...

Continúa...

AT+FTPN="eshos_17784 473"	OK	2.6	Usuario FTP
AT+FTPPW="prueba1"	OK	2.6	Contraseña FTP
AT+FTPPUTNAME="prue ba.txt"	OK	2.6	Archivo con extensión
AT+FTPPUTPATH="/htdo cs/"	OK	2.6	Directorio

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

✓ Subida de datos al internet

Con la conexión establecida hacia el internet se procede a iniciar sesión FTP para enviar los datos, como se muestra en la *Tabla 2-18*:

Tabla 2-18: Comandos para subir los datos

COMANDOS AT	RESPUESTA	TIEMPO ESPERA (segundos)	DESCRIPCION
AT+FTPPUT=1	OK +FTPPUT:1,1,130 0	9.5	Indicando al servidor que se desea enviar datos. 1300 indica que se puede enviar máximo 1300 bytes
AT+FTPPUT=2,45	+FTPPUT:2,45	9	Iniciando transferencia Se indica que se envían solamente 45 bytes desde ahora
Texto	OK +FTPPUT:1,1,130 0	20	Enviando datos. OK indica que se subieron correctamente los datos.
AT+FTPPUT=2,0	OK +FTPPUT:1,0	9	Cerrando transferencia

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Fuente: Creado por los autores

Finalmente en la *Figura 2-27*, se muestra un diagrama general de las conexiones de los tres primeros bloques del sistema. El mismo que representa un módulo, encargado de monitorear la recolección de desechos sólidos en un contenedor.

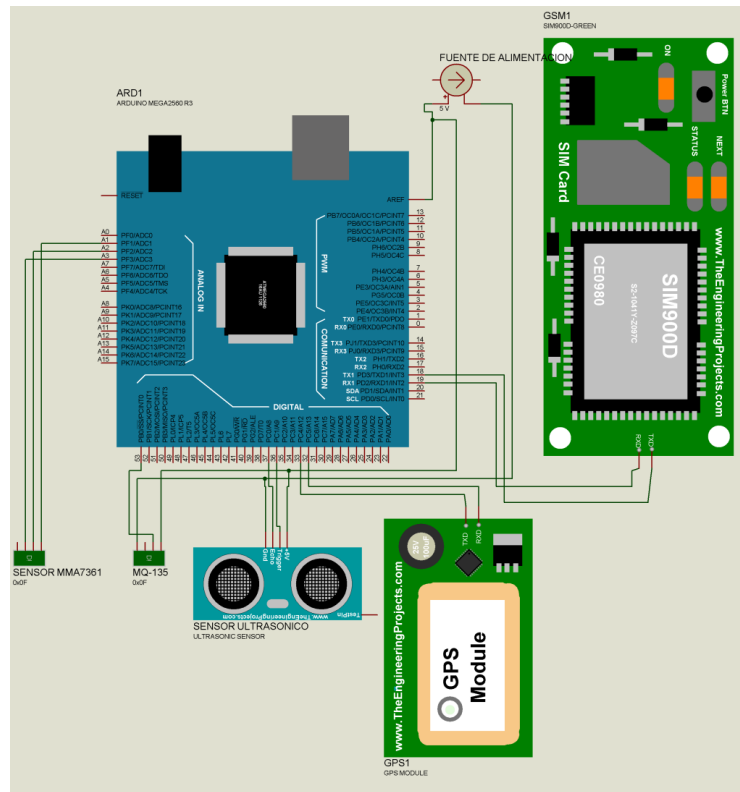


Figura 2-27: Diagrama de conexiones de los tres bloques iniciales.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.5.1.3 Bloque de almacenamiento

Los datos del sistema se alojan en un servidor de hosting gratuito, se contrató los servicios de almacenamiento de la empresa argentina ESHOST. Esta empresa dispone de un generador de dominios web, donde se creó la URL: con www.monitoreoarduino.eshost.com.ar.

El tiempo de espera para que el dominio sea reconocido por los DNS no superó los 5 minutos. Pasado el tiempo se comprobó que el dominio estaba disponible en línea con un mensaje de Bienvenida.

Se empezó a subir los archivos de la Aplicación Web a través del panel de control como se ve en la *Figura 2-28*, donde se observa cómo se va guardando los archivos en el administrador online.

All	Name	Type	Size	Owner	Group	Perms	Mod Time	Actions
	Up ..							
	Templates	Directory	4096	17784473	17784473	rwxt-xt-x	May 8 21:30	
	css	Directory	4096	17784473	17784473	rwxt-xt-x	May 8 21:29	
	images	Directory	4096	17784473	17784473	rwxt-xt-x	May 11 13:00	
	includes	Directory	4096	17784473	17784473	rwxt-xt-x	May 8 21:30	
	js	Directory	4096	17784473	17784473	rwxt-xt-x	May 8 21:30	
	detalles.php	PHP script	2781	17784473	17784473	rw-f--	May 10 02:28	Edit
	index.php	PHP script	4260	17784473	17784473	rw-f--	May 11 15:35	Edit
	mapa.php	PHP script	2293	17784473	17784473	rw-f--	May 10 02:28	Edit
	masinfo.php	PHP script	8779	17784473	17784473	rw-f--	May 11 12:59	Edit
	normas.php	PHP script	4948	17784473	17784473	rw-f--	May 10 02:28	Edit
	prueba.txt	Text file	45	17784473	17784473	rw-f--	May 10 02:28	Edit
	prueba2.txt	Text file	45	17784473	17784473	rw-f--	May 10 02:28	Edit
	prueba2t.txt	Text file	45	17784473	17784473	rw-f--	May 12 12:39	Edit
	prueba3.txt	Text file	45	17784473	17784473	rw-f--	May 12 13:25	Edit
	prueba3t.txt	Text file	45	17784473	17784473	rw-f--	May 12 12:24	Edit
	pruebat.txt	Text file	45	17784473	17784473	rw-f--	May 12 12:39	Edit
	registro.php	PHP script	7220	17784473	17784473	rw-f--	May 10 02:28	Edit

Figura 2-28: Almacenamiento de archivos online

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.5.1.4 Programación de la Interfaz de Usuario

Para visualizar, monitorear y controlar los datos almacenados en tiempo real, se procedió a la programación de la aplicación web. En la Figura 2-29, se muestra la programación de la misma apoyado del software Dreamweaver CS5 v 11.0 y el lenguaje utilizado PHP.

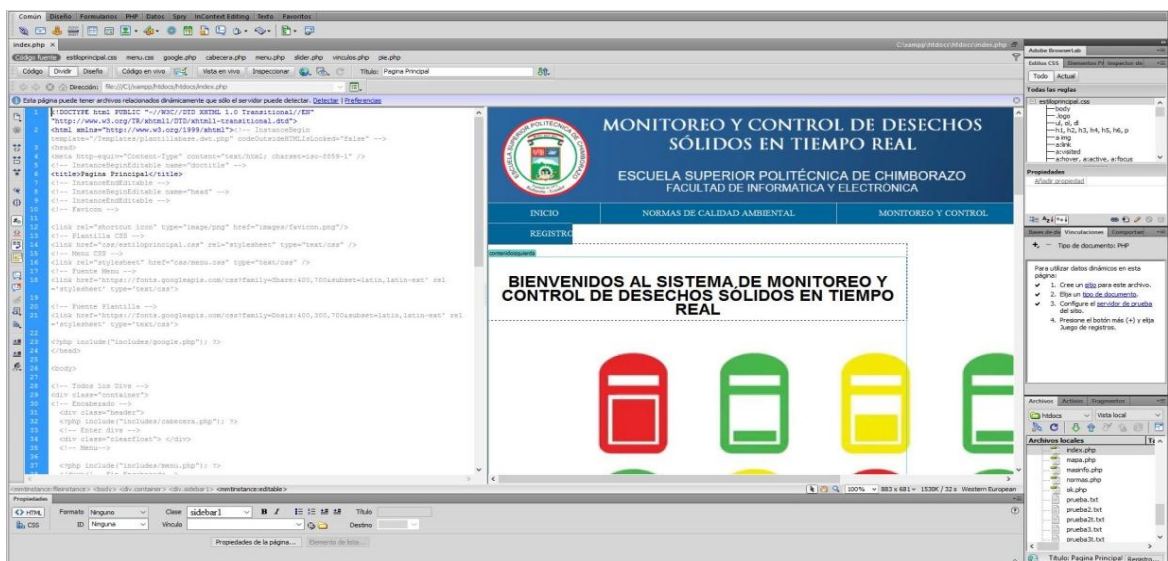


Figura 2-29: Programación Interfaz de usuario

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

2.5.2 *Protección Física.*

El módulo que comprende los tres primeros bloques se instaló dentro de un contenedor de basura, y para proteger los elementos de posibles golpes y maltratos se diseñó una caja protectora como se puede ver en la *Figura 2-30*, esta caja es de mucha utilidad para salvaguardar todos los dispositivos que componen el sistema.



Figura 2-30: Caja de protección

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

CAPITULO 3

3. EVALUACIÓN Y COMPARACION DE RESULTADOS

3.1 Introducción

En el capítulo se muestra las pruebas que se realizó en el sistema, empezando con estabilidad y luego transmisión, las mismas que permitieron observar cómo los sensores recolectaron los datos y enviaron la información a la aplicación web. Con los datos esperados se procedió a colocar los módulos en los contenedores para realizar pruebas de campo validando la eficiencia del sistema en tiempo real.

Para la recepción de información, se utilizó tres contenedores de desechos sólidos propios, porque la ubicación de cada contenedor depende de la cobertura de la operadora móvil, y de la recepción satelital.

3.2 Estabilidad del Sistema

Para este paso se procedió a la lectura de los 4 eventos (distancia, humo, posición y localización) con el fin de constatar que los módulos cuando sean ubicados en cada contenedor envíen datos verídicos y en tiempo real. Para determinar la estabilidad del sistema, se tomaron los valores que se recibieron en la aplicación de todos los sensores que se encuentran en el módulo.

Tomando los tres casos principales del evento distancia que son: vacío, medio y lleno, se muestreó los valores por un lapso de 30 minutos. Con la ayuda de un software estadístico se obtuvo el cálculo de la media, desviación estándar y coeficiente de variación, de los valores registrados por las 4 variables en la página Web. En la *Tabla 3-1* se observa las muestras recibidas en estado de contenedor vacío.

Tabla 3-1: Variables recibidas del módulo con nivel de llenado vacío.

#Muestra	Hora	SRF05	MQ-135	GPS		MPU-6050		
		Ultrasónico (cm)	Ppm	Latitud	Longitud	Eje x	Eje y	Eje z
1	16:20:17	54	0.02	-1,65769	-78,680439	500	600	14500
2	16:22:06	55	0.02	-1,65769	-78,680439	501	600	14500
3	16:23:29	55	0.02	-1,65769	-78,680439	498	600	14500
4	16:27:34	54	0.02	-1,65769	-78,680439	498	600	14500
5	16:28:50	54	0.02	-1,65769	-78,680439	498	600	14500
6	16:29:24	55	0.02	-1,65769	-78,680439	498	600	14500
7	16:30:14	55	0.02	-1,65769	-78,680439	498	600	14500
8	16:33:02	54	0.02	-1,65769	-78,680439	498	600	14500
9	16:34:52	55	0.02	-1,65769	-78,680439	498	600	14500
10	16:36:21	55	0.02	-1,65769	-78,680439	502	600	14500
11	16:38:30	54	0.02	-1,65769	-78,680439	502	600	14500
12	16:40:19	54	0.02	-1,65769	-78,680439	502	600	14500
13	16:42:09	55	0.02	-1,65769	-78,680439	500	600	14500
14	16:42:25	55	0.02	-1,65769	-78,680439	500	600	14500
15	16:43:58	54	0.02	-1,65769	-78,680439	501	600	14500
16	16:44:30	54	0.02	-1,65769	-78,680439	501	600	14500
17	16:45:47	54	0.02	-1,65769	-78,680439	502	600	14500
18	16:46:32	54	0.02	-1,65769	-78,680439	502	600	14500
19	16:47:36	55	0.02	-1,65769	-78,680439	502	600	14500
20	16:49:26	55	0.02	-1,65769	-78,680439	501	600	14500
21	16:51:15	54	0.02	-1,65769	-78,680439	500	600	14500
22	16:51:47	54	0.02	-1,65769	-78,680439	500	600	14500
Media		54	0.02	-1,65769	-78,680439	500	600	14500
Desviación Estándar		0,51	0	0	0	2	0	0
Coefficiente de Variación		0,94%	0%	0%	0%	0,33%	0,00%	0,00%

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Los datos obtenidos demostraron que, cuando el contenedor de desechos está vacío, el sensor ultrasónico tiene una variación de valores del 0,94%, en el caso del GPS y MQ-135 es 0%, finalmente en el caso del sensor MPU-6050 se puede ver una variación del 0.33% en el eje x, eje y e z, son 0%.

En la *Tabla 3-2* se observa las muestras recibidas en estado de contenedor medio.

Tabla 3-2: Repetitividad del módulo en estado contenedor medio.

#Muestra	Hora	SRF05	MQ-135	GPS		MPU-6050		
		Ultrasónico(cm)	Ppm	Latitud	Longitud	Eje x	Eje y	Eje z
1	19:32:20	35	0.02	-1,657693	-78,680439	500	600	14500
2	19:34:09	36	0.02	-1,657693	-78,680439	501	600	14500
3	19:34:52	36	0.02	-1,657693	-78,680439	498	600	14500
4	19:35:58	36	0.02	-1,657693	-78,680439	498	600	14500
5	19:37:47	38	0.02	-1,657693	-78,680439	498	600	14500
6	19:38:56	38	0.02	-1,657693	-78,680439	498	600	14500
7	19:41:26	36	0.02	-1,657693	-78,680439	498	600	14500
8	19:45:35	36	0.02	-1,657693	-78,680439	498	600	14500
9	19:46:57	36	0.02	-1,657693	-78,680439	498	600	14500
10	19:48:43	36	0.02	-1,657693	-78,680439	502	600	14500
11	19:50:05	36	0.02	-1,657693	-78,680439	502	600	14500
12	19:52:22	35	0.02	-1,657693	-78,680439	502	600	14500
13	19:53:13	35	0.02	-1,657693	-78,680439	500	600	14500
14	19:54:11	36	0.02	-1,657693	-78,680439	500	600	14500
15	19:56:06	36	0.02	-1,657693	-78,680439	501	600	14500
16	19:56:15	36	0.02	-1,657693	-78,680439	501	600	14500
17	19:57:53	36	0.02	-1,657693	-78,680439	502	600	14500
18	19:58:20	36	0.02	-1,657693	-78,680439	502	600	14500
19	19:59:39	37	0.02	-1,657693	-78,680439	502	600	14500
20	20:00:23	37	0.02	-1,657693	-78,680439	501	600	14500
Media		36	0.02	-1,657693	-78,680439	500	600	14500
Desviación Estándar		0,81	0	0	0	2	0	0
Coefficiente de Variación		2,25%	0%	0%	0%	0,34%	0,00%	0,00%

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Los datos obtenidos en este evento demostraron que cuando el contenedor de desechos está en nivel medio de llenado, el sensor ultrasónico tiene una variación de valores del 2,25%, en el caso del GPS y MQ-135 es 0%, para el sensor MPU-6050 se puede ver una variación del 0.34% en el eje x, eje y e z, son 0%.

En la *Tabla 3-3* se observa las muestras recibidas en estado de contenedor lleno.

Tabla 3-3: Repetitividad del módulo en estado contenedor lleno.

#Muestra	Hora	SRF05	MQ-135	GPS		MPU-6050		
		Ultrasónico(cm)	Ppm	Latitud	Longitud	Eje x	Eje y	Eje z
1	20:40:03	5	0.02	-1,657693	-78,680439	500	600	14500
2	20:40:35	5	0.02	-1,657693	-78,680439	501	600	14500
3	20:41:53	5	0.02	-1,657693	-78,680439	498	600	14500
4	20:43:42	5	0.02	-1,657693	-78,680439	498	600	14500
5	20:45:32	6	0.02	-1,657693	-78,680439	498	600	14500
6	20:46:02	6	0.02	-1,657693	-78,680439	498	600	14500
7	20:47:21	6	0.02	-1,657693	-78,680439	498	600	14500
8	20:48:09	6	0.02	-1,657693	-78,680439	498	600	14500
9	20:52:50	6	0.02	-1,657693	-78,680439	498	600	14500
10	20:53:57	6	0.02	-1,657693	-78,680439	502	600	14500
11	20:54:39	6	0.02	-1,657693	-78,680439	502	600	14500
12	20:55:05	6	0.02	-1,657693	-78,680439	502	600	14500
13	20:56:02	6	0.02	-1,657693	-78,680439	500	600	14500
14	20:56:35	6	0.02	-1,657693	-78,680439	500	600	14500
15	20:57:34	6	0.02	-1,657693	-78,680439	501	600	14500
16	20:58:05	6	0.02	-1,657693	-78,680439	501	600	14500
17	20:59:15	6	0.02	-1,657693	-78,680439	502	600	14500
18	21:02:42	6	0.02	-1,657693	-78,680439	502	600	14500
19	21:03:15	6	0.02	-1,657693	-78,680439	502	600	14500
20	21:05:39	6	0.02	-1,657693	-78,680439	501	600	14500
Media		6	0.02	-1,657693	-78,680439	500	600	14500
Desviación Estándar		0,41	0	0	0	2	0	0
Coefficiente de Variación		7,11%	0%	0%	0%	0,34%	0,00%	0,00%

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Finalmente, cuando el contenedor de desechos está lleno, el sensor ultrasónico tiene una variación de valores del 7,11%, en el caso del GPS y MQ-135 es 0%, finalmente en el caso del sensor MPU-6050 se puede ver una variación del 0.34% en el eje x, eje y e z, son 0%.

Como lo menciona en (DANE, 2008 pág. 5) y con la ayuda de las tres tablas anteriores se puede verificar que el sistema presenta un nivel de variabilidad no mayor a un 8%, por ende se considera un sistema preciso.

3.3 Pruebas de Transmisión

Para las pruebas de transmisión se ubicó cada módulo en tres puntos del campus de la Escuela Superior Politécnica de Chimborazo. Con la ayuda de la aplicación GMON, se conoció la potencia recibida por las radio bases de la operadora móvil. De ésta manera se determinó las ubicaciones idóneas para colocar los módulos de transmisión.

El primer módulo se ubicó en la salida del edificio de la Facultad de Informática y Electrónica, ya que la potencia recibida por la aplicación fue de RSSI= -63dBm, como se muestra en la *Figura 3-1*, potencia acorde para que el dispositivo GSM/GPRS, pueda transmitir datos sin problema.

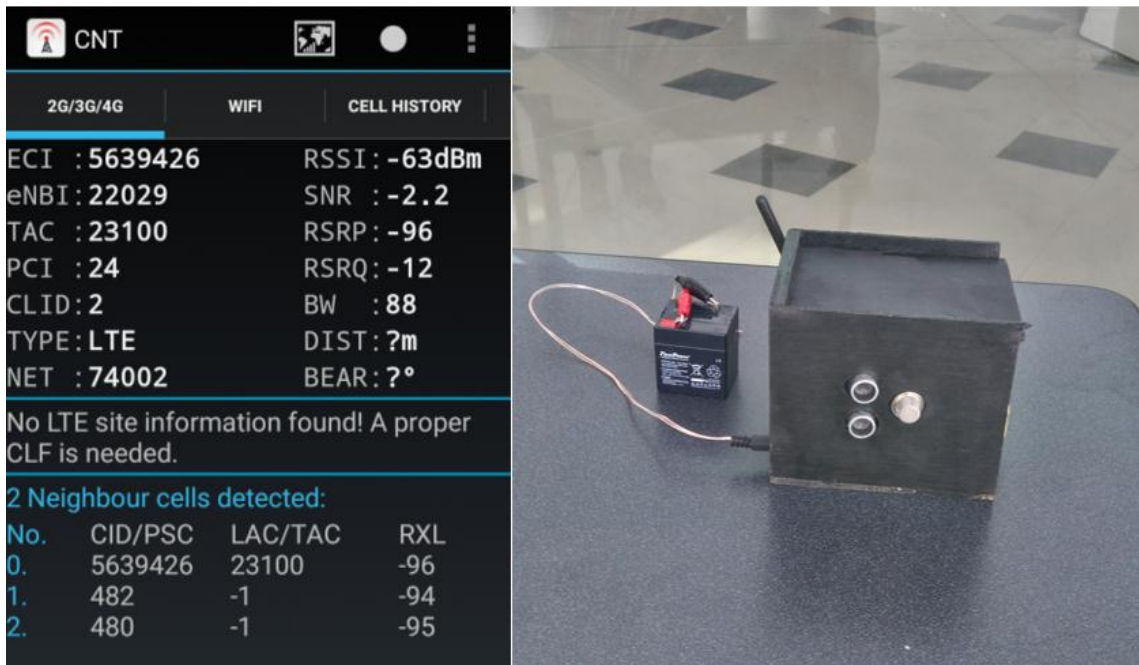


Figura 3-1: Módulo ubicado en la Facultad de Informática y Electrónica de la ESPOCH

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

El segundo módulo se ubicó en el centro de Atención Integral de la Salud, de la misma forma se obtuvo un valor de RSSI=-67dBm, como se puede ver en la *Figura 3-2*.



Figura 3-2: Ubicación en el Centro Integral para la Salud de la ESPOCH

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

El tercer y último módulo, se ubicó en el modular de la Escuela de Electrónica, como se ve en la *Figura 3-3*, el valor de RSSI=-67dBm.



Figura 3-3: Ubicación del módulo en el modular de Electrónica

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.3.1 Esquema de transmisión

En la *Figura 3-4*, se puede observar los tres módulos que fueron ubicados en el campus de la ESPOCH, los cuales se conectaron por el shield GPRS a la radiobase más cercana, permitiendo que los datos leídos por cada sensor sean subidos a la nube y sean visualizados por un dispositivo final.



Figura 3-4: Esquema de transmisión

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.3.2 Recepción de variables

Para comprobar la recepción de los datos se utilizó una laptop SONY VAIO VPCF1. Para el monitoreo del sistema se puede usar cualquier dispositivo siempre y cuando tenga acceso a internet, a través del URL: <http://monitoreoarduino.eshost.com.ar/ver.php>.

En la *Figura 3-5* se observa las variables recibidas desde el módulo que estuvo ubicado en el edificio de la FIE. Los valores que se observaron fueron: distancia, humo, latitud, longitud, Eje x, Eje y, Eje z.



Figura 3-5: Visualización en tiempo real de los datos recibidos por uno de los módulos.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Para observar el tráfico PHP entre la laptop SONY y el internet se utilizó el software Fiddler Web Debugger como se observa en la *Figura 3-6*, este programa captó todo el tráfico.

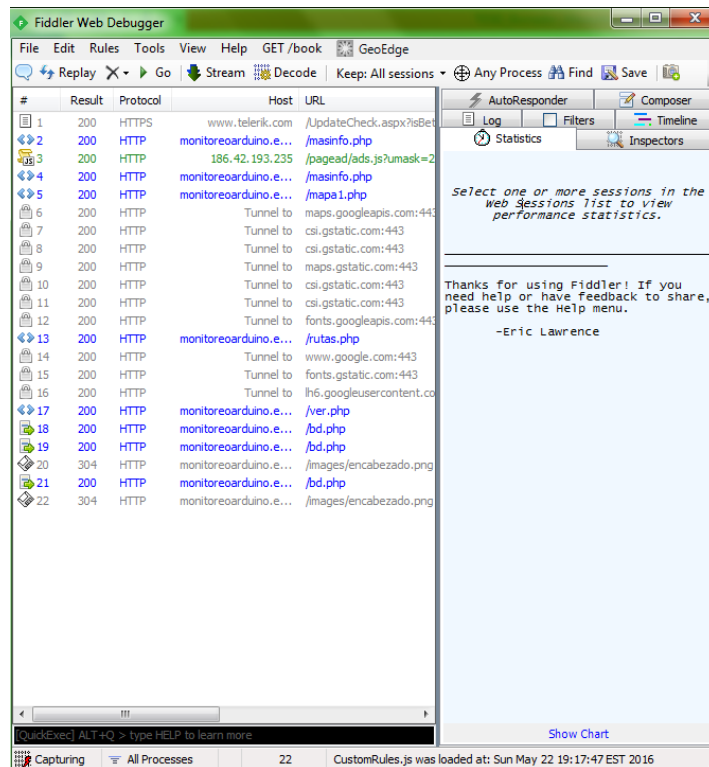


Figura 3-6: Trafico PHP

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.3.3 *Tiempo de transmisión y tráfico de datos*

Debido a que las operadoras celulares tienen estrictamente prohibido que terceras personas puedan ingresar a su infraestructura de red, no se pudo tomar los tiempos de transmisión y tráfico de datos.

Para medir los tiempos se consideró, el tiempo de encendido del módulo con la configuración inicial que es de aproximadamente 3 minutos y 5 segundos, esto se da porque el shield GPRS buscó las radiobases más cercanas para iniciar la transmisión de los datos a la nube.

Empíricamente se tomó el tiempo de envío de datos de cada módulo teniendo así una demora de 1 a 3 min. Considerando las pausas que se realizaron para la aceptación de los comandos AT dentro de la programación del Arduino Mega 2560.

Haciendo un análisis de los datos mostrados anteriormente en la *Figura 3-6* se observó un tiempo de hasta 3 minutos en recibir los datos y no se registraron pérdidas. Lo que se consideró un tiempo aceptable y un sistema óptimo.

3.4 Pruebas de Campo

Luego de haber verificado que la transmisión de los datos de cada módulo es óptima, se procedió a realizar una prueba en el campo real, para esto se ubicó tres contenedores en la ciudadela “Los Olivos” de la ciudad de Riobamba.

3.4.1 *Características del contenedor*

Para la prueba del prototipo se escogió un contenedor de las características que se muestran en la Tabla 3-4.

Tabla 3-4: Características del contenedor

Contenedor	Características
Forma	
Material	PVC
Medidas	Altura: 92.5cm Tapa: 48.5x49.5 Fondo: 40x37

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.4.2 Ubicación del módulo en el contenedor

La ubicación del módulo en el contenedor se basó a partir del sensor ultrasónico, el cual es el encargado de medir el nivel de llenado del contenedor, se llegó a la decisión de ubicarlo en la parte superior central del tacho para protegerlo de posibles golpes y para un mejor sensado de la acumulación de basura, como se ve en la *Figura 3-7*.



Figura 3-7: Ubicación del módulo

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.4.3 Ubicación de los contenedores

Cada contenedor se ubicó de la misma forma con la ayuda del software GMON para verificar que la potencia de la operadora CNT sea óptima. El primer contenedor fue ubicado en las calles Antonio de Alcedo y Jaime Roldós Aguilera como se puede ver en la *Figura 3-8*.



Figura 3-8: Contenedor 1

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

El segundo contenedor fue ubicado en las calles Pedro Franco y Díaz de la Madrid como se ve en la *Figura 3-9*.



Figura 3-9: Contenedor 2

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

El tercer y último contenedor en las calles Antonio de Alcedo y Díaz de la Madrid como se observa en la *Figura 3-10*.

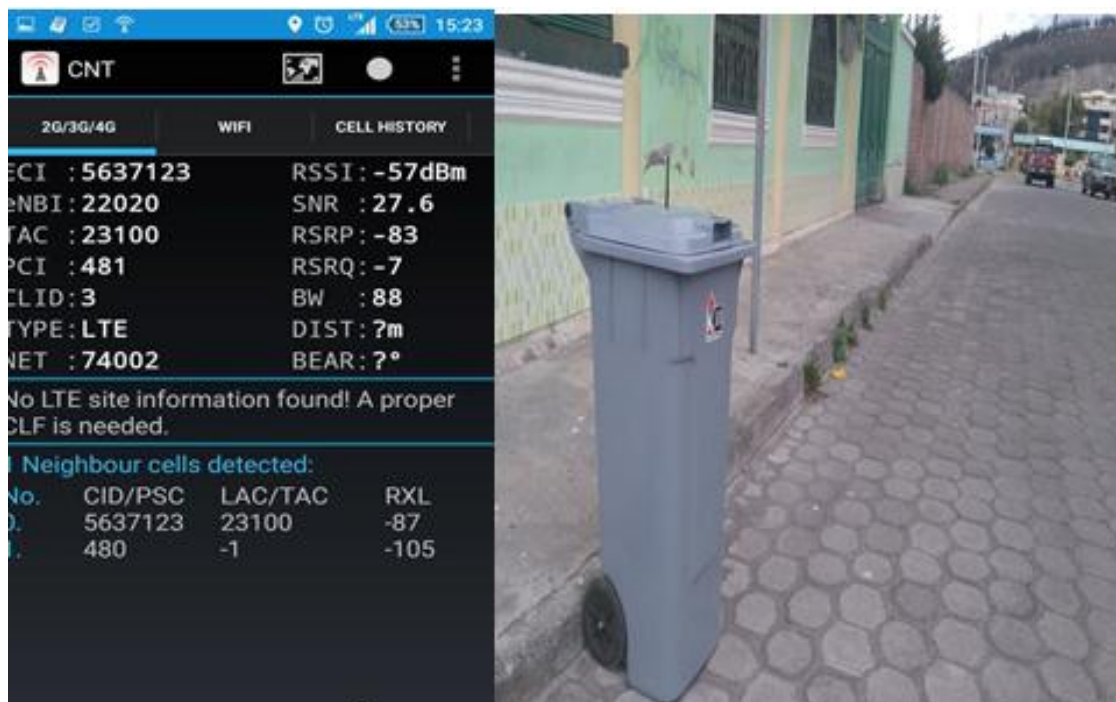


Figura 3-10: Contenedor 3

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.4.3.1 Consideraciones para considerar cada evento

Para monitorear los eventos de nivel de llenado y alerta de incendio se tomó las siguientes consideraciones como se muestra en la *Tabla 3-5* y *Tabla 3-6*.

Tabla 3-5: Consideraciones de nivel de llenado

MEDICIONES DEL SENSOR ULTRASÓNICO	ESTADO
<20cm	Lleno
<=20cm<=50cm	Medio
>50	Vacío

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Tabla 3-6: Consideraciones para alertar posibles incendios.

MEDICIONES DEL SENSOR MQ-135	ESTADO
>1.5 ppm	Alerta de Incendio

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Los eventos que se fueron dando en cada contenedor se muestran en la *Figura 3-11*. En el caso del contenedor 1 y contenedor 3 se detectó un nivel lleno, por consiguiente que se reportó con una alarma cambiando el color del contenedor a rojo.

En ninguno de los contenedores se reportó alertas de incendios o posibles movimientos. Y para el caso del contenedor 2 se reportó un nivel medio de llenado, por lo mismo no fue necesario tomar decisiones con respecto a ese contenedor.



Figura 3-11: Información del estado de cada contenedor en tiempo real

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.4.4 Ruta Óptima

De este modo se puede trazar una ruta óptima para que los recolectores puedan ir hacia los contenedores que necesitan ser vaciados. En la *Figura 3-12* se muestra la ruta que se trazó para los dos contenedores que se reportaron llenos.

Para calcular la ruta más óptima se siguió un algoritmo especial conocido como el del camino más corto, previsto por la API de Google. Se pudo calcular el tiempo estimado de recolección desde la central de monitoreo hasta el contenedor, aplicando la fórmula básica: $\text{tiempo} = \frac{\text{distancia}}{\text{velocidad}}$, para la velocidad se tomó en consideración las normas de la Agencia Nacional de Tránsito, dentro de una ciudad se permite hasta un máximo de 30km/h.



Figura 3-12: Ruta óptima para la recolección de desechos en contenedor 1 y 3.

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

Luego de haber finalizado las pruebas de campo, se planteó un mapa del centro de la ciudad de Riobamba donde se muestra una distribución tentativa de los contenedores de desechos como se puede ver en la *Figura 3-13* para este prototipo, la distancia entre cada contenedor será entre 100 y 200 metros de distancia, tomando en cuenta la cobertura de la operadora que se utilizó, en este caso CNT.

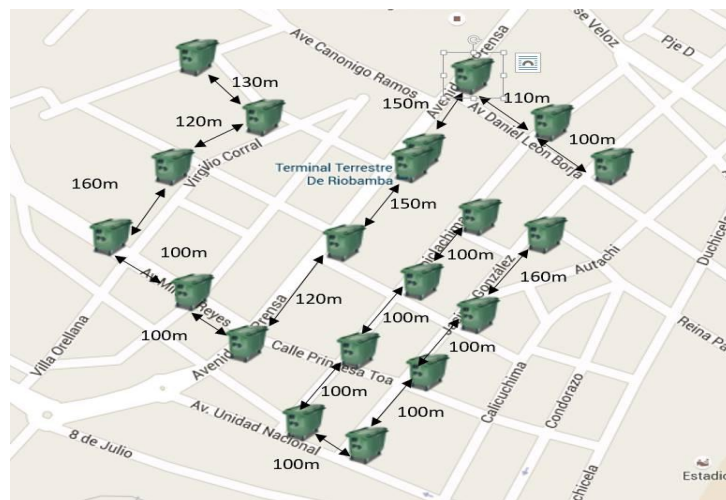


Figura 3-13: Distribución tentativa, contenedores en el centro de Riobamba

Fuente: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

3.5 Eficiencia Energética

En las pruebas realizadas se utilizó baterías secas recargables de 6v- 1.3A, Las mismas que tuvieron una duración de tres horas máximo. Para una proyección futura se prevé alcanzar una mayor duración y se recarguen las baterías con paneles solares debido a la potencia que transmite el sistema en cada una de las estaciones y tener una mejor eficiencia energética.

3.6 Presupuesto

Para la realización del sistema se utilizaron los siguientes elementos que se detallan a continuación con sus respectivos precios, como se muestra en la *Tabla 3-7*.

Tabla 3-7: Presupuesto

N°	MATERIAL	CANTIDAD	PVU	PRECIO FINAL
1	Sensor Ultrasónico HY-SRF05	3	9,5	28,5
2	Sensor Humo MQ-135	3	7	21
3	Modulo GPS Neo 6M	3	30	90
4	Módulo Acelerómetro MPU6050	3	13	39
5	Shield GSM/GPRS SIM900	3	50	150
6	Arduino Mega 2560	3	20	60
7	Cable Arduino		13,5	13,5
8	Tarjeta Sim CNT	3	3	9
9	Plan de Datos CNT	3	1	3
10	Caja Protectora	3	5	15
11	Batería seca	3	6,9	20,7
12	Cinta Doble Faz	3	3,75	11,25
13	Pintura y Lijas	1	3,35	3,35
14	Otras		10	10
	TOTAL			474,3

Realizado por: MOROCHO Vanessa & QUINAPANTA Jonathan, 2016

CONCLUSIONES

Cumpliendo con los objetivos propuestos y las respectivas evaluaciones de los resultados obtenidos se concluye lo siguiente:

1. En este trabajo se diseñó e implementó un sistema multimodal que permitió monitorear y controlar la recolección de desechos sólidos en tiempo real, lo que permite ingresar al Ecuador dentro del auge tecnológico que ofrecen las plataformas IoT y las Smart Cities para alcanzar un desarrollo sostenible siguiendo el plan del Buen Vivir.
2. En la valoración de los sensores se obtuvo una eficiencia en cuanto a los parámetros propuestos siendo estos: nivel de llenado, localización, alerta de incendio y manipulación de los contenedores, que permitieron recolectar datos fiables y confiables en el sistema de recolección de desechos en tiempo real.
3. La plataforma Arduino Mega 2560 permitió el procesamiento de la información obtenida por los sensores, y la transmisión de la misma mediante shield GSM/GPRS hacia el internet, haciendo de la tarjeta un dispositivo óptimo para el desarrollo de prototipos relacionados con el monitoreo y transmisión de datos requeridos en esta investigación.
4. Las pruebas realizadas en el barrio “Los Olivos”, y en el campus de la ESPOCH se demostró un nivel aceptable de RSSI con respecto a la cobertura celular, lo que permite que los módulos ubicados en cada contenedor de desechos sólidos envíen en tiempo real la información para ser visualizada en la red WAN y en la aplicación diseñada para el sistema de recolección de desechos para Smart Cities.
5. En base a los resultados obtenidos en las pruebas de campo, se concluye que la ubicación de los contenedores de basura debe estar a una distancia no mayor a 200 metros, para que no afecte a la transmisión de información por procesos de handoff.
6. Al trazar rutas óptimas para el recorrido de los recolectores de desechos sólidos se prevé el ahorro de recursos económicos al reducir el consumo de combustible siendo el tiempo de recorrido menor al tiempo tradicional fijo, además se minimiza las emisiones de CO₂ en el medio ambiente mejorando la calidad de vida de las personas.

RECOMENDACIONES

1. Para el buen funcionamiento del sensor ultrasónico, se debe utilizar la librería PING, la cual otorga una mejor sensibilidad del sensor al momento de adquirir datos y de este modo mejorar la estabilidad de todo el sistema.
2. Se recomienda que para comprobar la cobertura de la operadora celular se debe usar el programa de apoyo GMON el cual nos muestra el tipo de tecnología celular disponible y RSSI que se recibe en esa zona.
3. Para la utilización de un sistema en tiempo real con GSM/GPRS recomendamos utilizar operadoras que brinden coberturas amplias y a costos muy económicos para el uso de las redes WSN.
4. Se prevé que para implementaciones futuras se utilice una plataforma hardware libre más robusta, para que aumente el tiempo de vida de cada módulo
5. Se recomienda para mejorar la eficiencia energética en el sistemas utilizar nuevas tecnologías como los paneles solares en la alimentación con voltaje en el sistema.

BIBLIOGRAFIA

Arduino. "*Que es arduino*". [En línea] 2016.

[Citado el: 2 de Mayo de 2016.]

<https://www.arduino.cc/en/Guide/Introduction>

BANDYOPADHYAY, Debasis y SEN, Jaydip. "Internet of things – Applications and Challenges in Technology and Standardization". [pdf] 2011.

[Citado el: 20 de Abril de 2016.]

<http://arxiv.org/pdf/1105.1693.pdf>

BERNAL I. "*Comunicaciones Inalámbricas*". [pdf] 2007.

[Citado el: 3 de Mayo de 2016.]

http://clusterfie.epn.edu.ec/ibernal/html/CURSOS/2008_Marzo_Sept/ComInalam/ClasesNuevas/GPRS.pdf

CHINA. THAOYU-ELECTRONICS. "*MQ135 Gas Sensor for Air Qaulity*". [En línea] 2015.

[Citado el: 28 de Abril de 2016.]

<http://www.hotmcu.com/mq135-gas-sensor-for-air-qaulity-p-165.html>

CISCO. "*Internet of things-La próxima evolución de internet lo esta cambiando todo*". [pdf] 2011.

[Citado el: 19 de Abril de 2016.]

https://www.cisco.com/web/ES/assets/executives/pdf/Internet_of_Things_IoT_IBSG_0411FINAL.pdf

COOKING HACKS. "*Wasmote: Plataforma del sensor d redes inalambricas de código abierto*". [En línea] 20 de Junio de 2012.

[Citado el: 2 de Mayo de 2016.]

<https://www.cooking-hacks.com/blog/wasmote-wireless-sensor-networks-open-source-platform/>

COSTA RICA. ITNOW. *¿Cómo aumentar la seguridad en el internet de las cosas?*. [En línea] 2014.

[Citado el: 20 de Abril de 2016.]

<https://revistaitnow.com/como-puede-aumentar-la-seguridad-en-el-internet-de-las-cosas/>

DANE. *"Estimación e interpretación del coeficiente de variación de la encuesta concesal"*. [pdf] 2008.

[Citado el: 20 de Mayo de 2016.]

http://www.dane.gov.co/files/investigaciones/boletines/censo/est_interp_coefvariacion.pdf

ECUADOR. EL UNIVERSO. "El mercado de Smart Cities en Ecuador". [En línea]. 2011.

[Citado el: 20 de Abril de 2016.]

<http://www.eluniverso.com/noticias/2015/12/21/nota/5311224/espana-detecta-ecuador-oportunidades-negocio-campo-smart-cities>

ECUADOR. ESPOL. *"Norma de calidad ambiental para el manejo y disposicion "*. [pdf] 2009.

[Citado el: 2 de Mayo de 2016.]

<https://www.dspace.espol.edu.ec/bitstream/123456789/6078/55/LIBRO%20VI%20Anexo%206%20Manejo%20desechos%20solido%20no%20peligrosos.pdf>

ECUADOR. Líderes. *"La experiencia japonesa en 'Smart Cities' se compartió en Quito"*. [En línea] 11 de Noviembre de 2015.

[Citado el: 20 de Abril de 2016.]

<http://www.revistalideres.ec/lideres/experiencia-japonesa-smart-cities-quito.html>

ENDESA EDUCA. *"Smart Cities"*. [En línea] 2014.

[Citado el: 15 de Abril de 2016.]

http://www.endsaeduca.com/Endesa_educa/recursos-interactivos/smart-city/

ESTADOS UNIDOS. ENEVO. *"Optimising Waste Collection"*. [En línea] 2011.

[Citado el: 28 de Abril de 2016.]

<https://www.enevo.com/>

ESPAÑA. ENIGMEDIA. *"Ciudades Inteligentes con el IoT"*. [En línea] 2016.

[Citado el: 20 de Abril de 2016.]

<https://enigmedia.es/ciudades-inteligentes-iot/>

ESCUELA DE ORGANIZACIÓN INDUSTRIAL. "*Smart City! Las ciudades del presente.*".

[En línea] 3 de Octubre de 2015.

[Citado el: 20 de Abril de 2016.]

<http://www.eoi.es/blogs/redinnovacionEOI/2015/10/03/smart-city-las-ciudades-del-presente/>

ESPAÑA. eSMARTCITY. "*Sensores inteligentes de parking en la Feria Intertraffic 2016.*". [En línea] 13 de Abril de 2016.

[Citado el: 21 de Abril de 2016.]

[http://www.libelium.com/esmartcity-es-sensores-inteligentes-de-parking-en-la-feria-intertraffic-2016/#!prettyPhoto-img\[24182\]/0/](http://www.libelium.com/esmartcity-es-sensores-inteligentes-de-parking-en-la-feria-intertraffic-2016/#!prettyPhoto-img[24182]/0/)

EVERLET, Álvaro&PASTOR,Javier. "*Introduccion al internet de las cosas - Contruyendo un proyecto IOT*". [pdf] 2013.

[Citado el: 19 de Abril de 2016.]

https://www.carriots.com/newFrontend/img-carriots/press_room/Construyendo_un_proyecto_de_IOT.pdf

ESPAÑA. ExpoBarcelona. "Smart city Sostenibilidad y Medio Ambiente". [video] 2015.

[Citado el: 21 de Abril de 2016.]

<https://www.youtube.com/watch?v=59nsDqt2IdI>

ESPAÑA. TELEFÓNICA. "*Smart Cities: un paso hacia la internet de las cosas*". [En línea] 2011.

[Citado el: 29 de Abril de 2016.]

<https://app.box.com/s/wxe8yf7uawpcdj3xbacj>

ESPAÑA. Libelium. "*Transporte y Logística*". [En línea] 2015.

[Citado el: 20 de Abril de 2016.]

<http://www.libelium.com/>

ESPAÑA. HIBERUS. "*Smart Cities, ciudades inteligentes para la gestión eficiente, sostenible y transparente*". [blog] 2 de Diciembre de 2014.

[Citado el: 21 de Abril de 2016.]

<http://www.hiberus.com/blog/smart-cities-ciudades-inteligentes-para-la-gestion-eficiente-sostenible-transparente>

ESPAÑA. UPM. "*Comunicaciones móviles. GPRS*". [pdf] 30 de Marzo 2010.

[Citado el: 2 de Mayo de 2016.]

<http://ocw.upm.es/teoria-de-la-senal-y-comunicaciones-1/comunicaciones-moviles-digitales/contenidos/Presentaciones/GPRS-07.pdf>

ESTADOS UNIDOS. THE INTERNET OF ALL THINGS. "*Here's an intelligent garbage collection system that tells you when the bin is to be emptied*". [En línea] 2 de Enero de 2014.

[Citado el: 22 de Abril de 2016.]

<http://theinternetofallthings.com/iot-product-comes-intelligent-garbage-bin-tells-emptied-2014-01-02/>

ESPAÑA. SMARTCITIES IDOM. "*Jornadas modelo español Smart City*". [En línea] 14 de Octubre de 2015.

[Citado el: 20 de Abril de 2016.]

<http://www.smartcities.es/tag/ecuador/>

GARRIDO R. "*Estudio de Plataformas IoT*". [En línea] 12 de Junio de 2015.

[Citado el: 2 de Mayo de 2016.]

<https://openaccess.uoc.edu/webapps/o2/bitstream/10609/42812/6/rloureiroTFC0615memoria.pdf>

GEEETECH. "*GPRS Shield*". [blog] 8 de Julio de 2014.

[Citado el: 3 de Mayo de 2016.]

http://www.geeetech.com/wiki/index.php/Arduino_GPRS_Shield

GONZALES J.& MARIN C. "*Prototipo De Sistema De Seguridad En Vehículos De transporte publico que permite la captura de imagen fotografica, pisicionamiento global y almacenamiento en base de datos*". [pdf] 2015.

[Citado el: 28 de Abril de 2016.]

<http://repository.udistrital.edu.co/bitstream/11349/2249/1/MarinMendezCarlosArturo2015.pdf>

GUTIERREZ R. "*Sensor Characteristics*". [pdf] 2 de Agosto de 2002.

[Citado el: 21 de Abril de 2016.]

http://research.cs.tamu.edu/prism/lectures/iss/iss_12.pdf

INTERNATIONAL COMMUNICATION UNION. 2005. "*Internet of things*". [pdf] 2005.

[Citado el: 15 de Abril de 2016.]

<https://www.itu.int/net/wsis/tunis/newsroom/stats/The-Internet-of-Things-2005.pdf>

INTERNATIONAL COMMUNICATION UNION. "*Grupo Temático sobre ciudades sostenibles e inteligentes*". [En línea] 2016.

[Citado el: 20 de Abril de 2016.]

<http://www.itu.int/es/ITU-T/focusgroups/ssc/Pages/default.aspx>

ITALIA. Troniqa. "*Sensor HY-SRF05*". [En línea] 2014.

[Citado el: 27 de Abril de 2016.]

<http://www.troniqa.com/prossimita/81-hc-sr05-sensore-distanza-prossimit-ultrasuoni-sonar-robot-arduino-hy-srf05-111578757910.html>

J, CEVALLOS. "España detecta en Ecuador oportunidades de negocio en el campo de las 'smart cities'". *El Universo*. [En Línea]2015.

[Citado el: 20 de Abril de 2016]

<http://www.eluniverso.com/noticias/2015/12/21/nota/5311224/espana-detecta-ecuador-oportunidades-negocio-campo-smart-cities>

MAROTO P. "*Introduccion al Internet de las Cosas*". [blog] 2013.

[Citado el: 20 de Abril de 2016.]

<https://pacomaroto.wordpress.com/about/introduccion-a-la-internet-de-las-cosas/>

MEXICO. C.I.T.I.C. "*Ciudades Inteligentes: optimización en la Recoleccion de contenedores de Residuos Domiciliarios*". [En línea] 2015.

[Citado el: 19 de Abril de 2016.]

http://sedici.unlp.edu.ar/bitstream/handle/10915/45298/Documento_completo.pdf?sequence=1

MEXICO. SAPIENSA. "*Ciudades Inteligentes: Industria y el IoT garantizarán una eficiencia energética permanente*". [En línea] 15 de Octubre de 2015.

[Citado el: 20 de Abril de 2016.]

<http://www.sapiensa.org/tecnologia/ciudades-inteligentes-industria-y-el-iot-garantizaran-una-eficiencia-energetica-permanente/>

MUÑOZ, Marta y CATILLO, Laura. "*Las ciudades inteligentes como impulsoras de Internet de las cosas*". [En línea] 10 de Noviembre de 2015.

[Citado el: 20 de Abril de 2016.]

<http://toyoutome.es/blog/las-ciudades-inteligentes-como-impulsoras-de-internet-de-las-cosas/37659>

OpenMind. "*Sostenibilidad, Smart Cities y "ciudadanos sensores" en España*". [En línea] 2015.

[Citado el: 20 de Abril de 2016.]

<https://www.bbvaopenmind.com/sostenibilidad-smart-cities-y-ciudadanos-sensores-en-espana/>

RINCON B. "*Características del Internet de las cosas (IoT)*". [En línea] 2015.

[Citado el: 19 de abril de 2016]

<http://bsi1151136.blogspot.com/2015/10/caracteristicas-del-internet-de-las.html>

ROUSE M. "*Sensor*". [En línea] 2012.

[Citado el: 21 de Abril de 2016.]

<http://whatis.techtarget.com/definicion/sensor>

RUBEN A. "*¿Qué es y para qué sirve el dominio de tu página web?*". [En línea] 16 de Diciembre de 2014.

[Citado el: 3 de Mayo de 2016.]

<http://computerhoy.com/noticias/internet/que-es-que-sirve-dominio-tu-pagina-web-22007>

SorayapaNiagua. "*Un poco de historia sobre Internet de las Cosas*". [En línea] 15 de Abril de 2012.

[Citado el: 15 de Abril de 2016.]

<http://www.sorayapaniagua.com/2012/04/15/un-poco-de-historia-sobre-internet-de-las-cosas/>

Sticky Web Media. "*Web hosting*". [En línea] 2014.

[Citado el: 3 de Mayo de 2016.]

<http://www.stickywebmedia.com/web-hosting-services/>

SYMANTEC. "*Internet de las cosas-Protección de dispositivos y sistemas de Internet de las cosas*". [En línea] 1995.

[Citado el: 20 de Abril de 2016.]

<https://www.symantec.com/es/mx/iot/>

VELASCO R. "*Introducción al internet de las cosas*". [En línea] 29 de Enero de 2015.

[Citado el: 20 de Abril de 2016.]

<http://es.slideshare.net/siriuselearning/introduccion-al-internet-de-las-cosas>

VERMESAN, Ovidiu & FRIESS, Peter. "*Internet of things-From Research and Innovation to Market Deployment*".[pdf] 2009.

[Citado el: 22 de Mayo de 2016]

[http://www.internet-of-things-research.eu/pdf/Converging Technologies for Smart Environments and Integrated Ecosystems IERC Book Open Access 2013.pdf](http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf)

Zou W. "*GPRS Shield*". [En línea] 20 de Noviembre de 2013.

[Citado el: 03 de Mayo de 2016.]

<https://developer.mbed.org/users/lawliet/notebook/gprs/>

ANEXOS

DATASHEET SENSOR HY-SRF05

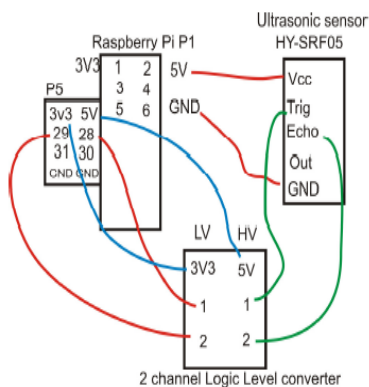
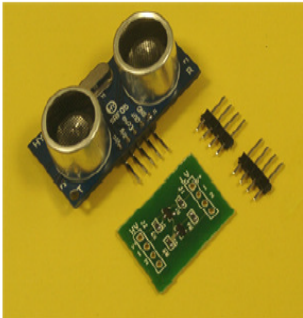
Ultrasonic distance sensor (HY-SRF05) with 2channel Logic Level converter

Use two GPIO Trig(T) Echo(E)

Update --> check distance and update

T28 --> Trig use GPIO28

E29 --> Echo use GPIO29



Ultrasonic sensor module Description :

- Working Voltage : 5V(DC)
- Static current: Less than 2mA.
- Output signal: Electric frequency signal, high level 5V, low level 0V.
- Sensor angle: Not more than 15 degrees.

- Detection distance: 2cm-450cm.
- High precision: Up to 0.2cm
- Input trigger signal: 10us TTL impulse
- Echo signal : output TTL PWL signal

• Mode of connection:

- 1.VCC 2.trig(T) 3.echo(R) 4.OUT 5.GND

• Use method:

- Supply module with 5V, the output will be 5V while obstacle in range, or 0V if not. The out pin of this module is used as a switching output when anti-theft module, and without the feet when ranging modules.

2 channel Logic Level Converter Description :

The logic level converter is a small PCB which safely allows you to interface devices which steps down 5V signals to 3.3V signals and step up 3.3V signals to 5V.

The converter has 2, inputs and outputs and can be used with I2C, SPI, UART, etc.

The board needs to be powered via two sources, 5V and 3.3V and your data lines are linked via pins 1 to 2 on each side so an input on pin 1 will be converted to the lower or higher output on the adjoining pin 1

Download GPIO library

<https://pypi.python.org/pypi/RPi.GPIO> GPIO library

GPIO library - RPi.GPIO-0.5.3a.tar.gz

Install python , library and run the test program

```
# sudo apt-get install python-dev
```

```
# wget http://www.pridopia.co.uk/pi-pgm/RPi.GPIO-0.5.3a.tar.gz
```

```
# gunzip RPi.GPIO-0.5.3a.tar.gz
```

```
# tar -xvf RPi.GPIO-0.5.3a.tar
```

```
# cd RPi.GPIO-0.5.3a
```

```
# sudo python setup.py install
```

TECHNICAL DATA

MQ-135 GAS SENSOR

FEATURES

- Wide detecting scope
- Fast response and High sensitivity
- Stable and long life
- Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH₃, NO_x, alcohol, Benzene, smoke, CO₂, etc.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33Ω±5%	Room Tem
P _H	Heating consumption	less than 800mw	

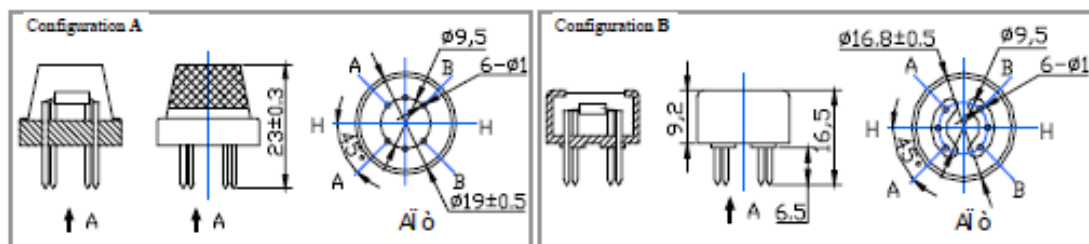
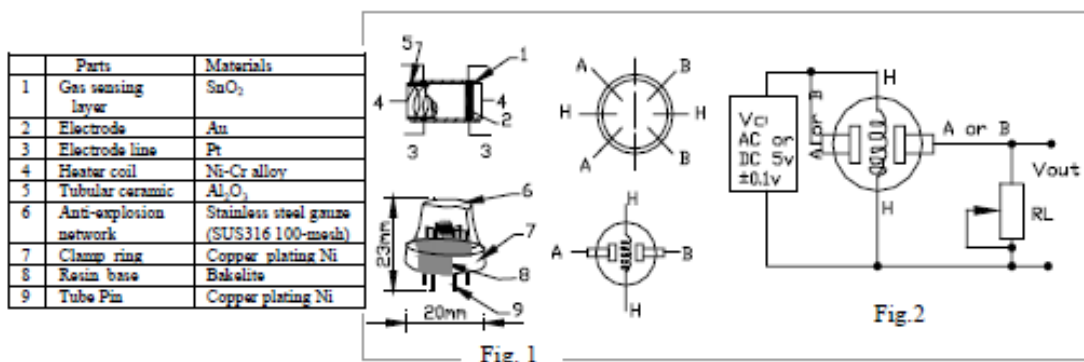
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
T _{ao}	Using Tem	-10~45℃	
T _{as}	Storage Tem	-20~70℃	
R _{rh}	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R _s	Sensing Resistance	30KΩ-200KΩ (100ppm NH ₃)	Detecting concentration scopell 10ppm-300ppm NH ₃ 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20±2℃ V _c :5V±0.1 Humidity: 65%±5% V _H : 5V±0.1		
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive

NEO-6 series

Versatile u-blox 6 GPS modules

Highlights

- UART, USB, DDC (iPC compliant) and SPI interfaces
- Available in Crystal and TCXO versions
- Onboard RTC crystal for faster warm and hot starts
- 1.8 V and 3.0 V variants

Features

- u-blox 6 position engine:
 - Navigate down to -162 dBm and -148 dBm coldstart
 - Faster acquisition with AssistNow Autonomous
 - Configurable power management
 - Hybrid GPS/SBAS engine (WAAS, EGNOS, MSAS)
 - Anti-jamming technology
- Simple integration with u-blox wireless modules
- A-GPS: AssistNow Online and AssistNow Offline services, OMA SUPL compliant
- Backward compatible (hardware and firmware); easy migration from NEO-5 family or NEO-4S
- LCC package for reliable and cost effective manufacturing
- Compatible with u-blox GPS Solution for Android
- Based on GPS chips qualified according to AEC-Q100
- Manufactured in ISO/TS 16949 certified production sites
- Qualified according to ISO 16750



NEO-6:
12.2 x 16.0 x 2.4 mm

Product description

The NEO-6 module series brings the high performance of the u-blox 6 position engine to the miniature NEO form factor. u-blox 6 has been designed with low power consumption and low costs in mind. Intelligent power management is a breakthrough for low-power applications. These receivers combine a high level of integration capability with flexible connectivity options in a small package. This makes them perfectly suited for mass-market end products with strict size and cost requirements. The DDC interface provides connectivity and enables synergies with u-blox LEON and LISA wireless modules.

All NEO-6 modules are manufactured in ISO/TS 16949 certified sites. Each module is tested and inspected during production. The modules are qualified according to ISO 16750 - Environmental conditions and electrical testing for electrical and electronic equipment for road vehicles.

Product selector

Model	Type	Supply	Interfaces	Features
	Standbone GPS Standbone GLONASS Timing & Raw Data Dead Reckoning	1.75 V - 2.0 V 2.7 V - 3.6 V	UART USB SPI DDC (iPC compliant)	Programmable (Flash) FW update Oscillator RTC crystal Antenna supply and supervisor Configuration pins Timepulse External interrupt / Wakeup
NEO-6G	•	•	• • • •	T • 0 3 1 •
NEO-6Q	•	•	• • • •	T • 0 3 1 •
NEO-6M	•	•	• • • •	C • 0 3 1 •

◦ = requires external components and integration on application processor

C = Crystal / T = TCXO



Receiver performance data

Receiver type	50-channel u-blox 6 engine GPS L1 C/A code SBAS: WAAS, EGNOS, MSAS	
Navigation update rate	up to 5 Hz	
Accuracy ¹	Position	2.5 m CEP
	SBAS	2.0 m CEP
Acquisition ¹		NEO-6G/Q NEO-6M
	Cold starts:	26 s 27 s
	Aided starts ² :	1 s < 3 s
	Hot starts:	1 s 1 s
Sensitivity ³		NEO-6G/Q NEO-6M
	Tracking:	-162 dBm -161 dBm
	Cold starts:	-148 dBm -147 dBm
	Hot starts:	-157 dBm -156 dBm

¹ All SV @ -130 dBm

² Dependent on aiding data connection speed and latency

³ Demonstrated with a good active antenna

Electrical data

Power supply	2.7 V – 3.6 V (NEO-6Q/6M) 1.75 V – 2.0 V (NEO-6G)
Power consumption	111 mW @ 3.0V (continuous) 33 mW @ 3.0V Power Save Mode (1 Hz) 68 mW @ 1.8V (continuous) 22 mW @ 1.8V Power Save Mode (1 Hz)
Backup power	1.4 V – 3.6V, 22 μ A
Supported antennas	Active and passive

Interfaces

Serial interfaces	1 UART 1 USB V2.0 full speed 12 Mbit/s 1 DDC (I ² C compliant) 1 SPI
Digital I/O	Configurable timepulse 1 EXTINT input for Wakeup
Serial and I/O	Voltages 2.7 – 3.6 V (NEO-6Q/6M) 1.75 – 2.0 V (NEO-6G)
Timepulse	Configurable 0.25 Hz to 1 kHz
Protocols	NMEA, UBX binary, RTCM

Legal Notice

u-blox reserves all rights to this document and the information contained herein. Products, names, logos and design described herein may in whole or in part be subject to intellectual property rights. Reproduction, use, modification or disclosure to third parties of this document or any part thereof without the express permission of u-blox is strictly prohibited.

The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or context of this document. This document may be revised by u-blox at any time. For most recent documents, please visit www.u-blox.com.

Copyright © 2011, u-blox AG

Specification applies to FW 7

Package

24 pin LCC (Leadless Chip Carrier): 12.2 x 16.0 x 2.4 mm, 1.6 g

Pinout



Environmental data, quality & reliability

Operating temp. -40° C to 85° C

Storage temp. -40° C to 85° C

RoHS compliant (lead-free)

Qualification according to ISO 16750

Manufactured in ISO/TS 16949 certified production sites

Support products

u-blox 6 Evaluation Kits:

Easy-to-use kits to get familiar with u-blox 6 positioning technology, evaluate functionality, and visualize GPS performance.

EVK-6H: u-blox 6 Evaluation Kit with TCXO, suitable for NEO-6G, NEO-6Q

EVK-6P: u-blox 6 Evaluation Kit with crystal, suitable for NEO-6M

Ordering information

NEO-6G-0	u-blox 6 GPS Module, 1.8V, TCXO, 12x16mm, 250 pcs/reel
NEO-6M-0	u-blox 6 GPS Module, 12x16mm, 250 pcs/reel
NEO-6Q-0	u-blox 6 GPS Module, TCXO, 12x16mm, 250 pcs/reel

Available as samples and tape on reel (250 pieces)

Contact us

HQ Switzerland
+41 44 722 7444
info@u-blox.com

EMEA
+41 44 722 7444
info@u-blox.com

Americas
+1 703 483 3180
info_us@u-blox.com

APAC – Singapore
+65 6734 3811
info_ap@u-blox.com

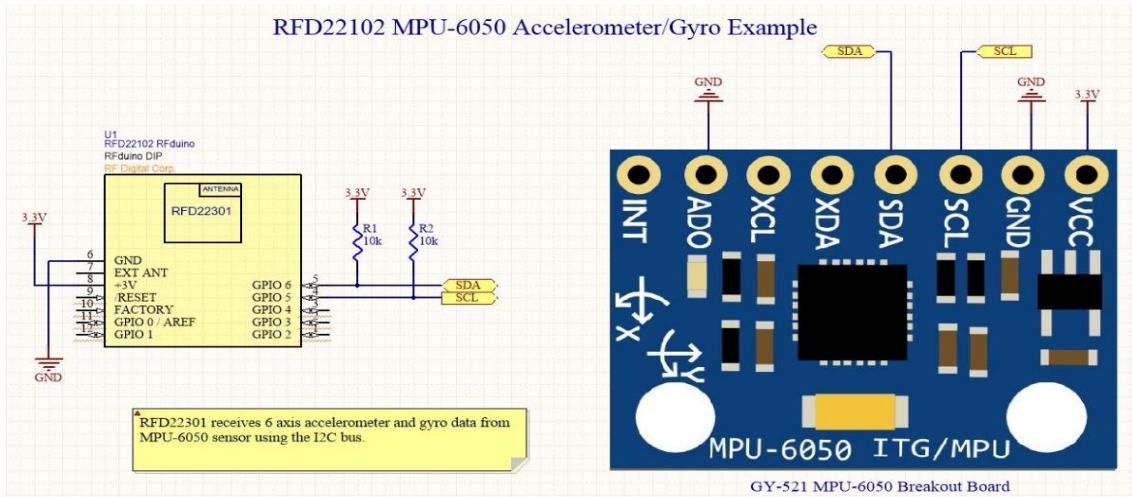
China
+86 10 68 133 545
info_cn@u-blox.com

Japan
+81 3 5775 3850
info_jp@u-blox.com

Korea
+82 2 542 0861
info_kr@u-blox.com

Taiwan
+886 2 2657 1090
info_tw@u-blox.com

RFD22102 MPU-6050 Accelerometer/Gyro Example

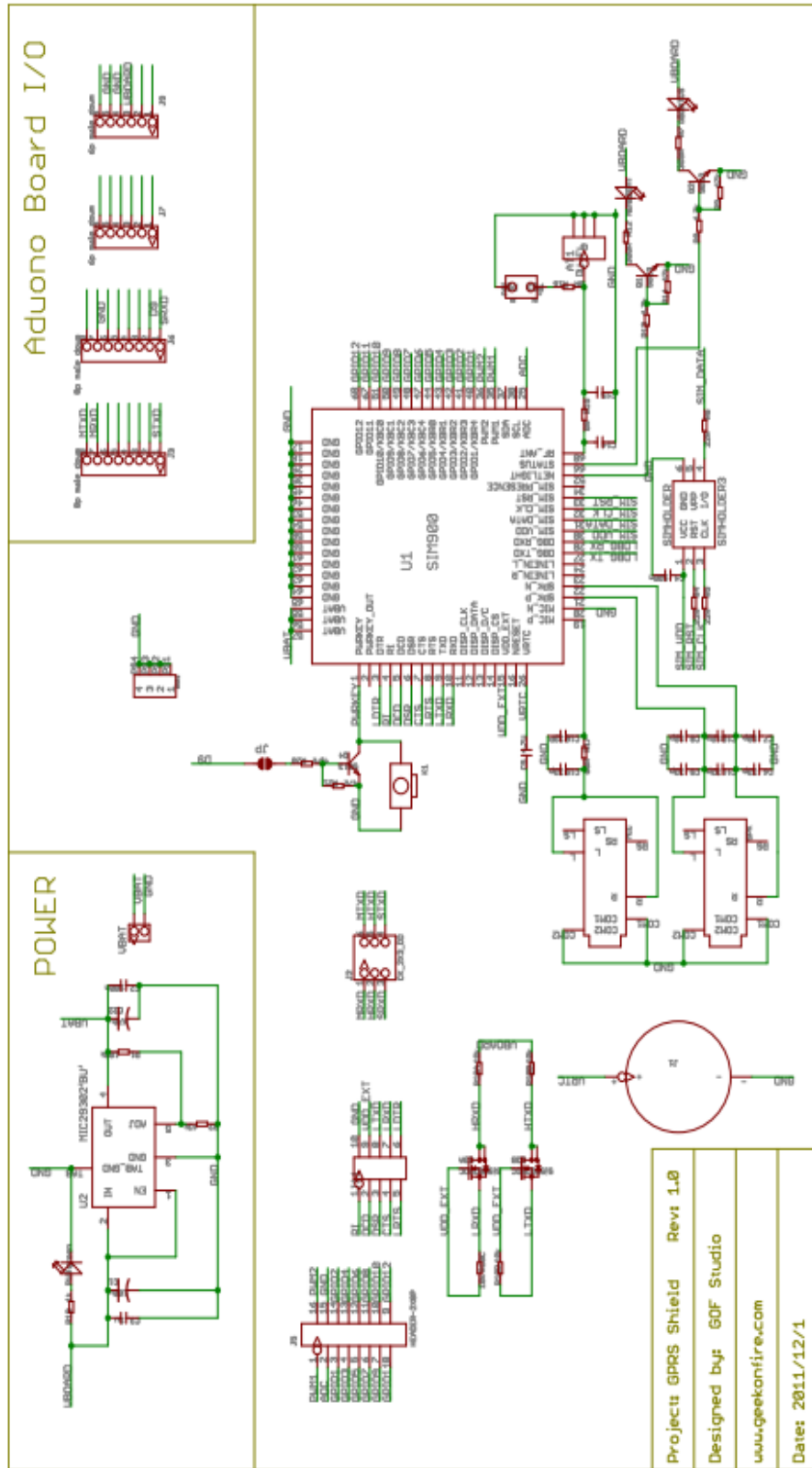


6.5 Electrical Specifications, Continued

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

Parameters	Conditions	Typical	Units	Notes
Primary I²C I/O (SCL, SDA)				
V _{IL} , LOW-Level Input Voltage	MPU-6000	-0.5 to 0.3*VDD	V	
V _{IH} , HIGH-Level Input Voltage	MPU-6000	0.7*VDD to VDD + 0.5V	V	
V _{hys} , Hysteresis	MPU-6000	0.1*VDD	V	
V _{IL} , LOW Level Input Voltage	MPU-6050	-0.5V to 0.3*VLOGIC	V	
V _{IH} , HIGH-Level Input Voltage	MPU-6050	0.7*VLOGIC to VLOGIC + 0.5V	V	
V _{hys} , Hysteresis	MPU-6050	0.1*VLOGIC	V	
V _{OL1} , LOW-Level Output Voltage	3mA sink current	0 to 0.4	V	
I _{OL} , LOW-Level Output Current	V _{OL} = 0.4V	3	mA	
	V _{OL} = 0.6V	5	mA	
Output Leakage Current		100	nA	
t _{of} , Output Fall Time from V _{IHmax} to V _{ILmax}	C _b bus capacitance in pF	20+0.1C _b to 250	ns	
C _i , Capacitance for Each I/O pin		< 10	pF	
Auxiliary I²C I/O (AUX_CL, AUX_DA)				
V _{IL} , LOW-Level Input Voltage	MPU-6050: AUX_VDDIO=0	-0.5V to 0.3*VLOGIC	V	
V _{IH} , HIGH-Level Input Voltage		0.7*VLOGIC to VLOGIC + 0.5V	V	
V _{hys} , Hysteresis		0.1*VLOGIC	V	
V _{OL1} , LOW-Level Output Voltage	VLOGIC > 2V; 1mA sink current	0 to 0.4	V	
V _{OL3} , LOW-Level Output Voltage	VLOGIC < 2V; 1mA sink current	0 to 0.2*VLOGIC	V	
I _{OL} , LOW-Level Output Current	V _{OL} = 0.4V	1	mA	
	V _{OL} = 0.6V	1	mA	
Output Leakage Current		100	nA	
t _{of} , Output Fall Time from V _{IHmax} to V _{ILmax}	C _b bus capacitance in pF	20+0.1C _b to 250	ns	
C _i , Capacitance for Each I/O pin		< 10	pF	
Auxiliary I²C I/O (AUX_CL, AUX_DA)				
V _{IL} , LOW-Level Input Voltage	MPU-6050: AUX_VDDIO=1; MPU-6000	-0.5 to 0.3*VDD	V	
V _{IH} , HIGH-Level Input Voltage		0.7*VDD to VDD+0.5V	V	
V _{hys} , Hysteresis		0.1*VDD	V	
V _{OL1} , LOW-Level Output Voltage	1mA sink current	0 to 0.4	V	
I _{OL} , LOW-Level Output Current	V _{OL} = 0.4V	1	mA	
	V _{OL} = 0.6V	1	mA	
Output Leakage Current		100	nA	
t _{of} , Output Fall Time from V _{IHmax} to V _{ILmax}	C _b bus cap. in pF	20+0.1C _b to 250	ns	
C _i , Capacitance for Each I/O pin		< 10	pF	

GPRS SHIELD



CODIGO FUENTE DEL SISTEMA

```
/*TRABAJO DE TITULACION
 *DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA MULTIMODAL QUE PERMITA MONITOREAR Y CONTROLAR LA
 RECOLECCION DE DESECHOS SÓLIDOS EN TIEMPO REAL PARA SMART CITIES”
*/
//*****
//ULTRASONICO HY-SRF05
//*****
//Libreria

#include <NewPing.h>
#define TRIGGER_PIN 4 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 5 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in centimeters). Maximum
sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum
distance.
/**Ubicacion Pines
//Arduino -> Ultrasonico
//Pin 5V -> Pin 1(Vcc)
//Pin GND -> Pin 6(GND)
//Pin 4 -> Pin 2(Trig)
//Pin 5 -> Pin 3(Echo)

//Variable Global
int distancia;
//Funcion
//ultrasonido();

//Caracteristicas
//Lee 3m o 3000cm

//*****
//SHIELD GSM SIM900
//*****
/**Libreria
#include <SoftwareSerial.h>

/**Ubicacion Pines
SoftwareSerial mySerial(0,1);//RX y TX
//Arduino -> GSM
//Pin 5V -> Pin 5V
//Pin GND -> Pin GND
//Pin 0(RX)-> Pin 0
//Pin 1(TX)-> Pin 1
//Pin 9 -> Pin 9
//Jumper en Hardware Serial(Xduino)
//Acoplar shield al arduino
```

```

//APN
//APN: internet.cnt.net.ec
//Usuario y Password: vacio

//*****
//FTP
//*****
//Caracteristicas
//WEB: monitoreoarduino.eshost.com.ar
//FTP: ftp.eshost.com.ar
//Panel FTP: cpanel.eshost.com.ar
//Usuario: eshos_17784473
//Contraseña: prueba1
//Directorio: /htdocs/

//*****
//MODULO GPS NEO 6M
//*****
/**Libreria
#include <TinyGPS.h>
TinyGPS gps;

/**Ubicacion Pines
SoftwareSerial ss(53,52); //TX y RX
//Arduino -> GPS
//Pin 3.3V -> Pin 3.3V
//Pin GND -> Pin GND
//Pin 52 -> Pin RX
//Pin 53 -> Pin TX
//Arduino Mega debe usarse pines digitales

//Variable Global
float flat, flon;

//Funcion
//gps2();

//*****
//CALIDAD DEL AIRE MQ-135
//*****
//Libreria
#include "MQ135.h"

MQ135 gasSensor = MQ135(0);
#define RZERO 76.63

/**Ubicacion Pines
//Arduino -> Calidad de Aire
//Pin A0 -> Pin 1(A0)
//Pin GND -> Pin 3(GND)

```

```

//Pin 5V    -> Pin 4(Vcc)
//Variable Global
float rzero;
float ppm;

//Funcion
//aire();

//*****
//ACELEROMETRO MPU-6050
//*****
/**Libreria
#include "I2Cdev.h"
#include "MPU6050.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
MPU6050 accelgyro;
#define OUTPUT_READABLE_ACCELGYRO
bool blinkState = false;

/**Ubicacion Pines
//Arduino    -> Acelerometro
//Pin Vcc    -> Pin 1(Vcc)
//Pin 5v     -> Pin 2(GND)
//Pin 21(SCL) -> Pin 3(SCL)
//Pin 20(SDA) -> Pin 3(SDA)
//Pin 2      -> Pin 3(INT)

//Variable Global
int16_t ax, ay, az;
int16_t gx, gy, gz;

//Funcion
//aire();

//Caracteristicas
//Lee 3m o 3000cm

//*****
//SUBIR A LA NUBE
//*****
//Variables Globales
int valor1=100; //Ultrasonido
float valor2=200; //Latitud
float valor3=300; //Longitud
float valor4=400; //aire
int valor5=500; //ejex
int valor6=600; //eje y
int valor7=700; //eje z

```

```

String valor = ""; //Valor acumulado

//PRIMERA VEZ
void setup()
{
    delay(100);//Esperando que se active GSM
    //Power GSM
    powerUp();
    delay(3000);
    //mySerial.begin(19200);//Baudios Radio

    //ULTRASONICO HY-SRF05
    //Activando
    pinMode(4, OUTPUT); /*trigger activación del pin 4 como salida: para el pulso ultrasónico*/
    pinMode(5, INPUT); /*echo activación del pin 5 como entrada: tiempo del rebote del
ultrasonido*/
    //Acelerometro
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
        Wire.begin();
    #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
        Fastwire::setup(400, true);
    #endif

    accelgyro.initialize();

    //Seccion Baudios en Puertos Seriales
    mySerial.begin(19200); //GPRS
    //Serial.begin(19200); //Serial PC
    ss.begin(9600); //GPS

    //Esperando que arranque Interfaz AT
    delay(9000);

    //GPRS
    Gprs();

    //FTP
    FTP();
}

//SECCION VOIDS PRIMERA VEZ

void powerUp()
{
    pinMode(9, OUTPUT);
    //digitalWrite(3,LOW);
    //delay(1000);
    digitalWrite(9,HIGH);
    delay(1000);
    digitalWrite(9,LOW);
}

```

```
    delay(5000);
}
void Gprs()
{
    mySerial.println("AT");
    delay(100);
    mySerial.println();
    delay(2500);

    mySerial.println("AT+CPIN?");
    delay(100);
    mySerial.println();
    delay(2500);

    mySerial.println("AT+CFUN?");
    delay(100);
    mySerial.println();
    delay(2500);

    mySerial.println("AT+CCALR?");
    delay(100);
    mySerial.println();
    delay(2500);

    mySerial.println("AT+CIPSHUT");
    delay(100);
    mySerial.println();
    delay(10000);

    mySerial.println("AT+CGATT=0");
    delay(100);
    mySerial.println();
    delay(10000);

    mySerial.println("AT+CGATT=1");
    delay(100);
    mySerial.println();
    delay(10000);

    mySerial.println("AT+CSQ");
    delay(100);
    mySerial.println();
    delay(2500);

    mySerial.println("AT+CREG?");
    delay(100);
    mySerial.println();
    delay(2500);

    mySerial.println("AT+CSTT?");
```

```
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"); //setting the SAPBR, the connection type
is using gprs
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+SAPBR=3,1,\"APN\",\"internet.cnt.net.ec\"); //setting the APN, the second
need you fill in your local apn server
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+SAPBR=3,1,\"USER\",\"\"); //
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+SAPBR=3,1,\"PWD\",\"\"); //setting the APN, the second need you fill in
your local apn server
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+SAPBR=1,1"); //
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+CSTT=\"internet.cnt.net.ec\", \"\", \"\"); //
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+CIPSRIP=1");
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+CIICR");
delay(100);
mySerial.println();
delay(2500);

mySerial.println("AT+CIFSR"); // read the IP
delay(100);
mySerial.println();
```



```
    delay(5000);
}
void FTP()
{
    mySerial.println("AT+FTPCID=1"); //
    delay(100);
    mySerial.println();
    delay(2500);

    //Tipo de Sesion FTP
    mySerial.println("AT+FTPTYPE=\"A\""); //
    delay(100);
    mySerial.println();
    delay(2500);

    //Direccion del Servidor FTP
    mySerial.println("AT+FTPSERV=\"ftp.eshost.com.ar\""); //
    delay(100);
    mySerial.println();
    delay(2500);

    //Puerto FTP
    mySerial.println("AT+FTPPORT=21"); //
    delay(100);
    mySerial.println();
    delay(2500);

    //Usuario FTP
    mySerial.println("AT+FTPUN=\"eshos_17784473\""); //
    delay(100);
    mySerial.println();
    delay(2500);

    //Password FTP
    mySerial.println("AT+FTPPW=\"prueba1\""); //
    delay(100);
    mySerial.println();
    delay(2500);

    //Nombre del fichero
    mySerial.println("AT+FTPPUTNAME=\"prueba3.txt\""); //
    delay(100);
    delay(2500);

    //Directorio
    mySerial.println("AT+FTPPUTPATH=\"/htdocs/\""); //
    delay(100);
    mySerial.println();
    delay(2500);
}
```

```

//SECCION REPITUTI
void loop()
{
  //Bahia de Importaciones
  ultrasonido();
  gps2();
  aire();
  acelerometro();
  //Bahia de Transferencia
  valor1= distancia; //int
  valor2= flat; //float
  valor3= flon; //float
  valor4= ppm; //float
  valor5= ax; //int
  valor6= ay; //int
  valor7= az; //int

  //Bahia de carga
  valor="";
  valor += valor1;
  valor += ':';
  valor += (valor2*10000);
  valor += ':';
  valor += (valor3*10000);
  valor += ':';
  valor += valor4;
  valor += ':';
  valor += valor5;
  valor += ':';
  valor += valor6;
  valor += ':';
  valor += valor7;
  valor += ':';
  //: Adicionales en el caso que no se cumpla la subida de bytes acordados, si se cumple estos
  seran dropeados
  valor += ':';
  valor += ':';
  valor += ':';
  valor += ':';

  SubirNube();
}

//SECCION VOID LOOP

void ultrasonido()
{
  distancia=sonar.ping_cm();
  while(distancia==0)

```

```

    {
        delay(50);                // Wait 50ms between pings (about 20 pings/sec). 29ms should
be the shortest delay between pings.
        distancia=sonar.ping_cm();
    }
    delay(500);

}

void SubirNube()
{
    //Iniciando la sesion
    mySerial.println("AT+FTPPUT=1"); //
    delay(9000);
    mySerial.println();
    delay(500);

    //Definiendo numero bytes de datos a subir. Cada byte corresponde a un caracter sea este
"A", "5", ".", "-",
    mySerial.println("AT+FTPPUT=2,45");
    delay(9000);

    //Subiendo datos a FTP
    mySerial.println(valor); //
    delay(20000);

    //Cierra Sesion FTP
    mySerial.println("AT+FTPPUT=2,0"); //
    delay(9000);
    mySerial.println();

    //Tiempo de muestreo
    delay(90000);
}

void gps2()
{
    bool newData = false;

    // For one second we parse GPS data and report some key values
    for (unsigned long start = millis(); millis() - start < 1000;)
    {
        while (ss.available())
        {
            {
                char c = ss.read();
                // Serial.write(c); // uncomment this line if you want to see the GPS data flowing
                if (gps.encode(c)) // Did a new valid sentence come in?
                    newData = true;
            }
        }
    }
}

```

```

if (newData)
{

    //muestreo
    for(int i=1;i<=5;i++)
    {
        gps.f_get_position(&flat, &flon);

        //Tiempo de Muestreo
        delay(2000);
    }
}
}
void aire() {

    for(int i=1;i<=5;i++)
    {
        rzero = gasSensor.getRZero();
        ppm = gasSensor.getPPM();

        delay(2000);
    }
}

void acelerometro()
{
    for(int i=1;i<=5;i++)
    {
        // read raw accel/gyro measurements from device
        accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

        delay(2000);
    }
}

```

CODIGO FUENTE PAGINA WEB

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/plantillabase.dwt.php" codeOutsideHTMLOIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Mas Informacion</title>
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<!-- InstanceEndEditable -->
<!-- Favicon -->

```



```
<th scope="col"><center>&nbsp;&nbsp;&nbsp;Observacion&nbsp;&nbsp;&nbsp;</center></th>
</tr>
```

```
<?php
```

```
//Numero contenedor
```

```
$numerocont=0;
```

```
//Leyendo fichero local
```

```
//fichero: nombre normal
```

```
//ficherot: respaldo
```

```
//Aqui debe colocar los ficheros que enviaron los arduinos en orden
```

```
$ficheros="prueba.txt prueba2.txt prueba3.txt";
```

```
$ficherost="pruebat.txt prueba2t.txt prueba3t.txt";
```

```
//Separando nombres
```

```
$nombre=explode(' ', $ficheros);
```

```
$nombret=explode(' ', $ficherost);
```

```
//Lmites
```

```
$limite=count($nombre);
```

```
$limitet=count($nombret);
```

```
//echo $limite." ".$limitet."<br>";
```

```
//For para todos los ficheros
```

```
for($i=0;$i<$limite;$i++)
```

```
{
```

```
    $file=fopen($nombre[$i], "r");
```

```
//Comprobando si el fichero principal no esta en blanco
```

```
if(filesize($nombre[$i])>0)
```

```
{
```

```
    //Fecha Creacion Archivo
```

```
    $fecha=date("d/m/Y", filectime($nombre[$i]));
```

```
    $hora=date("H:i:s", filectime($nombre[$i]));
```

```
    //Guardando variable para evitar perdidas de datos por malas subidas temporal
```

```
        $original=$nombre[$i];
```

```
        $copia=$nombret[$i];
```

```
        copy($original,$copia);
```

```
    }
```

```
    //SI esta en Blanco
```

```
    else
```

```
    {
```

```
        $file=fopen($nombret[$i], "r"); //Se procede a leer archivo temporal
```

```
        //Fecha Creacion Archivo
```

```
        $fecha=date("d/m/Y", filectime($nombret[$i]));
```

```
        $hora=date("H:i:s", filectime($nombret[$i]));
```

```

}
$numerocont=$numerocont+1; //Numero de Contenedor

//Comprobando que no este vacio
while(!feof($file))
{
    $afile=fgets($file);//Leer archivo
    $bfile=explode(':', $afile);//Hacer vector a partir de separaciones :
    //Leyendo Acelerometro
    $ax=$bfile[4];
    $ay=$bfile[5];
    $az=$bfile[6];
    ?><tr>

    <!--Imagen tiene condiciones de distancia humo y acelerometro-->
    <td><center><?php
    if($bfile[0]<20 || $bfile[3]>1.5 || (($ax<-1500 || $ax>1500)&&($ay<-1500 ||
$ay>1500)&&($az<=13500)))
    {
        ?><?php
    }
    else
    {
        ?><?php
    }
    ?></center></td>

    <!--Estado tiene condiciones de distancia humo y acelerometro-->
    <td><center><?php
    if($bfile[0]<20 || $bfile[3]>1.5 || (($ax<-1500 || $ax>1500)&&($ay<-1500 ||
$ay>1500)&&($az<=13500)))
    {
        echo "Atencion";
    }
    else
    {
        echo "Excelente";
    }
    ?></center></td>

    <!--Numero Contenedor -->
    <td><center><?php
    echo "$numerocont";
    ?></center></td>

    <!--Fecha y Hora -->
    <td><center><?php
    echo $fecha;?>

```

```
</center></td>
```

```
<td><center><?php  
echo $hora;?>  
</center></td>
```

```
<!--Nivel de LLenado -->
```

```
<td><center><?php  
if($bfile[0]<20)  
{  
    echo "Lleno";  
}  
else  
{  
    if($bfile[0]>=20 && $bfile[0]<=50)  
    {  
        echo "Medio";  
    }  
    else  
    {  
        echo "Vacio";  
    }  
}  
?></center></td>
```

```
<!--Latitud y Longitud -->
```

```
<td><center><?php  
echo $bfile[1]/10000;  
$latitud[$i]=$bfile[1]/10000;  
?></center></td>
```

```
<td><center><?php  
echo $bfile[2]/10000;  
$longitud[$i]=$bfile[2]/10000;  
?></center></td>
```

```
<!--Humo-->
```

```
<td><center><?php  
if($bfile[3]<=1.5)  
{  
    echo "Normal";  
}  
else  
{  
    echo "Alerta de Incendio";  
}  
?></center></td>
```

```
<!-- Posicion -->
```

```
<td><center><?php
```



```

//Boca arriba
if(($ax>=-1500 && $ax<=1500)&&($ay>=-1500 && $ay<=1500)&&($az>13500))//1//4//7//10
{
    echo "Tapa Cerrada";
}
//Vcc Yendo Arriba
else if(($ax>=-1500 && $ax<=1500)&&($ay>1500 && $ay<=13500)&&($az>1500 &&
$az<=13500))//2
{
    echo "Funcionamiento Anormal 1";
}
//Vcc Arriba
else if (($ax>=-1500 && $ax<=1500)&&($ay>13500)&&($az>=-1500 && $az<=1500))//3/13
{
    echo "Funcionamiento Anormal 2";
}
//Vcc Yendo Abajo
else if(($ax>=-1500 && $ax<=1500)&&($ay>=-13500 && $ay<=-1500)&&($az>1500 &&
$az<=13500))//5
{
    echo "Tapa Semiabierta";
}
//Vcc Abajo
else if (($ax>=-1500 && $ax<=1500)&&($ay<-13500)&&($az>=-1500 && $az<=1500))//6/16
{
    echo "Tapa Abierta";
}

//ITG yendo arriba
else if((( $ax>1500)&&($ax<=13500))&&($ay>=-1500 && $ay<=1500)&&($az>1500 &&
$az<=13500))//8
{
    echo "Tacho Inclinado 1";
}

//ITG Arriba
else if (($ax>13500)&&($ay>=-1500 && $ay<=1500)&&($az>=-1500 && $az<=1500))//9/19
{
    echo "Tacho de lado 1";
}

//IYG yendo abajo
else if((( $ax>=-13500)&&($ax<-1500))&&($ay>=-1500 && $ay<=1500)&&($az>1500 &&
$az<=13500))//11
{
    echo "Tacho Inclinado 2";
}

//ITG Abajo
else if (($ax<-13500)&&($ay>=-1500 && $ay<=1500)&&($az>=-1500 && $az<=1500))//12/22

```

```

{
    echo "Tacho De lado 2";
}

//Boca Abajo
    else if(($ax>=-1500 && $ax<=1500)&&($ay>=-1500 && $ay<=1500)&&($az<-
13500))//15/18/21/24
    {
        echo "Funcionamiento Anormal 3";
    }
    //Vcc yendo arriba con boca abajo
    else if(($ax>=-1500 && $ax<=1500)&&($ay>1500 && $ay<=13500)&&(($az>=-13500)&&($az<-
1500))//14
    {
        echo "Funcionamiento Anormal 4";
    }

    //Vcc yendo abajo en boca abajo
    else if(($ax>=-1500 && $ax<=1500)&&($ay>=-13500 && $ay<-1500)&&(($az>=-
13500)&&($az<-1500))//17
    {
        echo "Funcionamiento Anormal 5";
    }

    //ITG arriba yendo a boca abajo
    else if(($ax>1500 && $ax<=13500)&&($ay>=-1500 && $ay<=1500)&&(($az>=-13500)&&($az<-
1500))//20
    {
        echo "Tacho Inclinado 3";
    }

    //ITG abajo yendo boca abajo
    else if(($ax>=-13500 && $ax<-1500)&&($ay>=-1500 && $ay<=1500)&&(($az>=-13500)&&($az<-
1500))//23
    {
        echo "Tacho Inclinado 4";
    }

    else
    {
        echo "Tapa Cerrada 1";
    }
    ?></center></td></tr><?php
}

```

```
        //Fclose de los ficheros prueba.txts
        fclose($file);
    }

?>
</table></center>

    <p>&nbsp;</p> <!-- Enter drastico -->
<!-- InstanceEndEditable --></div>
<!-- Fin Lateral -->

    <!-- Pie Final -->
<div class="footer">
    <?php include("includes/pie.php"); ?>
    </div> <!-- Fin pie -->
</div><!-- Fin Divs -->
</body>
<!-- InstanceEnd --></html>

<!--/refrescar Toda la Pagina >
<script type="text/javascript">
    //<![CDATA[
    <!--
    setTimeout("location.reload()", 60000);
    //-->
    //]]>
</script>
```