



*“Responsabilidad con pensamiento positivo”*

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:  
INGENIERO EN ELECTRÓNICA DIGITAL Y  
TELECOMUNICACIONES**

TEMA:

“PROTOTIPO DE UN SISTEMA AUTOMÁTICO PARA EL CONTROL DE LLENADO DE DEPÓSITOS DE BASURA CON MONITOREO REMOTO EN EL MERCADO MAYORISTA DE LA CIUDAD DE QUITO UTILIZANDO UNA APLICACIÓN EN ANDROID”.

AUTOR:

José Luis Castillo Tipantasi

TUTOR:

Ing. Millard Escalona, PhD.

QUITO, ECUADOR 2018

## DECLARACIÓN

Yo, José Luis Castillo Tipantasi, declaro bajo juramento que el trabajo aquí descrito es de mi autoría.



**José Luis Castillo Tipantasi**

**CERTIFICACIÓN TUTOR**  
**UNIVERSIDAD TECNOLÓGICA ISRAEL**  
**APROBACIÓN DEL TUTOR**

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “Prototipo de un sistema automático para el control de llenado de depósitos de basura con monitoreo remoto en el Mercado Mayorista de la ciudad de Quito utilizando una aplicación en Android”, presentado por la Sr. José Luis Castillo Tipantasi, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito, 23 de Febrero del 2018

TUTOR

.....  
Ing. Millard Escalona Hernandez



## CONTENIDO

<b>UNIVERSIDAD TECNOLÓGICA ISRAEL</b> .....	
<b>CERTIFICACIÓN TUTOR</b> .....	
<b>INDICE</b> .....	
RESUMEN .....	5
ABSTRACT .....	6
DEDICATORIA Y AGRADECIMIENTO .....	7
<b>I. INTRODUCCIÓN</b> .....	8
<b>I.I. ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO</b> .....	8
<b>I.II. PLANTEAMIENTO DEL PROBLEMA</b> .....	9-10
<b>I.III. JUSTIFICACIÓN DEL PROBLEMA</b> .....	10-12
<b>1.4. OBJETIVOS DE LA INVESTIGACIÓN</b> .....	12
<b>1.5. DESCRIPCIÓN DE LOS CAPITULOS</b> .....	13
<b>CAPÍTULO I</b> .....	14
<b>1. FUNDAMENTACIÓN TEÓRICA</b> .....	14
<b>1.1 FUNDAMENTACIÓN TEÓRICA</b> .....	14
<b>1.1.1 RECOLECCIÓN DE BASURA</b> .....	14
<b>1.1.2 CONTENEDORES</b> .....	15
<b>1.1.3 SISTEMA DE CONTROL</b> .....	15
<b>1.1.4 CARACTERÍSTICAS DE UN SISTEMA DE CONTROL</b> .....	16
<b>1.1.4.1 CLASIFICACIÓN DE LOS SISTEMA DE CONTROL</b> .....	17
<b>1.1.5 SISTEMA DE MONITOREO</b> .....	18
<b>1.1.6 SENSORES</b> .....	18
<b>1.1.6.1 CLASIFICACIÓN DE LOS SENSORES</b> .....	19-21
<b>1.1.7 SOFTWARE</b> .....	21
<b>1.1.7.1 ANDROID STUDIO</b> .....	22
<b>1.1.7.2 AMAZON WEB SERVICE (AWS)</b> .....	25
<b>1.1.7.3 PHPMYADMIN</b> .....	26
<b>1.1.8 ARQUITECTURA DE LA APLICACIÓN</b> .....	27
<b>1.1.9 HARDWARE LIBRE</b> .....	29
<b>1.1.9.1 MÓDULO GSM/GPRS SIM 900</b> .....	29
<b>1.1.9.2 CARACTERÍSTICAS DEL SHIELD GSM/GPRS SIM 90</b> .....	30
<b>1.1.9.3 DIAGRAMA DE PINES SIM 900</b> .....	30
<b>1.1.9.4 MÓDULO GPS neo 6M Ublox</b> .....	31
<b>1.1.9.5 ARDUINO MEGA 2560</b> .....	31
<b>1.1.9.6 ARDUINO MEGA</b> .....	36
<b>1.1.9.7 MODULO GSM GPRS SIM900</b> .....	35
<b>1.1.9.8 MODULO GPS UBBLOX NEO 6M</b> .....	36

<b><u>CAPÍTULO II.</u></b> .....	38
<b><u>2. PROPUESTA</u></b> .....	38
<u>2.1 PROPUESTA DE SOLUCIÓN</u> .....	38
<u>2.2 PROPUESTA DE DISEÑO</u> .....	38
<u>2.3 MODALIDAD DE LA INVESTIGACIÓN</u> .....	38
<u>2.4 POBLACIÓN Y MUESTRA</u> .....	39
<u>2.5 RECOLECCIÓN DE INFORMACIÓN</u> .....	39
<u>2.6 PROCESAMIENTO Y ANALISIS DE DATOS</u> .....	39
<u>2.7 DESARROLLO DEL PROYECTO</u> .....	40
<u>2.8 ANALISIS DE FACTIBILIDAD</u> .....	41
<u>2.9. INFORMACIÓN DE ACTUAL SISTEMA DE RECOLECCIÓN</u> .....	41
<u>2.10. CONTENEDORES METALICOS CMR-2400/300</u> .....	42-43
<u>2.11. RUTAS DE RECOLECCIÓN DE BASURA</u> .....	44
<u>2.12. DIAGRAMA DE BLOQUES DE LA PROPUESTA</u> .....	45-48
<u>2.13. ELEMENTOS Y COSTOS DEL PROTOTIPO</u> .....	49
<u>2.14. VENTAJAS DEL PROTOTIPO</u> .....	50
<u>2.15. CRONOGRAMA DE ACTIVIDADES</u> .....	51
<b><u>CAPÍTULO III.</u></b> .....	51
<b><u>3. IMPLEMENTACIÓN</u></b> .....	52
<u>3.1 DESARROLLO</u> .....	52
<u>3.2 IMPLEMENTACIÓN</u> .....	52
<u>3.2.1 REQUISITOS DE SOFTWARE</u> .....	53
<u>3.2.2 DISEÑO</u> .....	53
<u>3.2.3 MONTAJE DEL PROYECTO</u> .....	54-60
<u>3.2.4 ANDROID STUDIO</u> .....	60
<u>3.2.5 VENTAJAS DE HERRAMIENTAS</u> .....	62
<u>3.2.6 DISEÑO DEL DISPOSITIVO</u> .....	63
<u>3.3 PRUEBAS DE FUNCIONAMIENTO</u> .....	65
<u>3.3.1 LISTA DE VERIFICACIÓN DE PRUEBAS</u> .....	65-66
<u>3.4. ANALISIS DE RESULTADOS</u> .....	67
<b><u>CONCLUSIONES</u></b> .....	68
<b><u>RECOMENDACIONES</u></b> .....	69
<b><u>BIBLIOGRAFÍA</u></b> .....	70
<b><u>ANEXOS</u></b> .....	71-100
<b><u>DECLARACIÓN Y AUTORIZACIÓN</u></b> .....	101

## **RESUMEN**

En la investigación se presenta el desarrollo de un prototipo automático para el control, llenado y monitoreo de depósitos de basura de forma remota mediante una aplicación Android, para así lograr mejorar el sistema actual de recolección de los desechos sólidos.

El diseño, funcionamiento del prototipo y su aplicación realiza la función de recibir la información proveniente de los sensores, permite conocer el estado actual en tiempo real las variables tanto en nivel como en peso de los desechos que se encuentran dentro del contenedor, mediante una red GSM/GPRS, además permite mantener un historial del llenado del contenedor.

La sistemática está orientada a establecer las rutas más eficientes, para que los vehículos de carga lateral encargados de la recolección de los desechos sólidos sean enviados solo cuando el contenedor haya superado el 80% de su capacidad ya sea en peso o en volumen, evitando de esta manera que la basura este a la intemperie generando diversos problemas y malestares a la sociedad en general, el 20% restante de la capacidad del contenedor tiene un tiempo estimado para que el operador pueda llegar al sitio de recolección de 60 minutos.

**PALABRAS CLAVES:** Arduino, Monitoreo Remoto, GPRS, Android, Java.

## **ABSTRACT**

The research presents the development of an automatic prototype for the control, filling and monitoring of garbage deposits remotely using an Android application, in order to improve the current solid waste collection system.

The design, operation of the prototype and its application performs the function of receiving information from the sensors, allows to know the current status in real time variables both in level and weight of the waste that is inside the container, through a network GSM / GPRS, also allows to maintain a history of filling the container.

The systematics is oriented to establish the most efficient routes, so that lateral loading vehicles responsible for the collection of solid waste are sent only when the container has exceeded 80% of its capacity either by weight or volume, avoiding this way that the garbage is out in the open and generates various problems and discomforts to society in general, the remaining 20% of the capacity of the container has an estimated time for the operator to reach the collection site of 60 minutes.

**KEY WORDS:** Arduino, Remote Monitoring, GPRS, Android, Java

## **DEDICATORIA**

El presente proyecto va dedicado a todas esas personas que supieron encaminarme y contribuyeron de una manera u otra para su realización. A todas esas personas que mediante sus ideas e incentivos permitieron se materialice este proyecto, del cual me siento totalmente satisfecho y muy orgulloso en presentar. Para todos ellos hago esta dedicatoria con mucho cariño.

## **AGRADECIMIENTO**

A DIOS, a mis Padres, a un ser muy especial Mi Abuelita que desde el cielo me ha guiado por buen camino, a quienes, a lo largo de mi vida, me han apoyado y motivado mi formación académica, poniendo toda su confianza en mí en todo momento y no dudaron de mis habilidades. A mis queridos profesores a quienes les debo la mayor parte de mis conocimientos, muy agradecido por su paciencia y enseñanza, y finalmente un eterno agradecimiento a esta respetuosa y prestigiosa universidad la cual me abrió sus puertas, preparándome para un futuro competitivo y formándome como un buen profesional.

## **I.- INTRODUCCIÓN**

El Prototipo para el Control y Monitoreo de los contenedores de basura de la ciudad de Quito , permite la optimización de los recursos tanto económicos como humanos al momento de realizar la recolección de los desechos sólidos, de tal forma que se conozca de manera inmediata mediante un sistema central de monitoreo web el estado de los contenedores de basura; además permite que el personal de EMASEO tenga un conocimiento oportuno del estado de los contenedores tanto en volumen como en peso y con el análisis del operario a través de la aplicación se envíen los vehículos de carga lateral solo cuando sea necesaria la recolección, evitando así el malestar en los moradores por el desbordamiento de la basura.

## **II.- ANTECEDENTES DE LA SITUACIÓN OBJETO DE ESTUDIO**

Realizado el proceso de investigación de los diferentes repositorios digitales de las Universidades y Escuelas Politécnicas, se encontraron los siguientes resultados:

Mauricio Renato Pozo García, en su tesis de grado presenta Análisis de los beneficios de una adecuada gestión de manejo de residuos sólidos en el Distrito Metropolitano de Quito.

Cesar Augusto Jácome Comina de la Escuela Politécnica del Ejército extensión Latacunga, en su tesis de grado presenta el diseño y la implementación de una red tipo MESH para el monitoreo de una amplia gama de sectores y cultivos agrícolas y florícolas del cantón Pujili; utiliza el estándar ZigBee, el cual va a permitir gestionar los equipos y dispositivos de manera más eficiente. Para la obtención de los datos cuenta con un dispositivo terminal con sensores para evaluar las variables físicas como la temperatura y humedad del suelo en una zona de cultivo para posteriormente ser enviados al dispositivo coordinador, el cual va a ser el encargado de procesar los datos de entrada y salida además permite el reconocimiento y comunicación con los demás nodos presentes en la red MESH para evitar pérdidas de información, se ha tomado en cuenta mucho esta tesis ya que abarca tecnologías de comunicación que utilizaremos en nuestro proyecto.

Y otro proyecto de graduación interesante es el del Ingeniero Gonzalo Daniel Ruiz Mesías de la Universidad Técnica de Ambato, quien desarrollo un Sistema de control y monitoreo para

mejorar los procesos de administración de los laboratorios de las Carreras de Sistemas, Electrónica e Industrial en la FISEI-UTA, el cual está encaminado a minimizar y optimizar los tiempos que emplean los ayudantes al momento de abrir y cerrar las puertas de cada uno de los laboratorios de la facultad, mediante un sistema que permita accionar dichas cerraduras, al mismo tiempo que permite observar mediante cámaras IP cada una de las actividades que se están generando en los mismos, para todo esto se empleó una tarjeta Arduino como elemento central.

### **III.- PLANTEAMIENTO DEL PROBLEMA**

La recolección de basura a nivel mundial es una de las principales preocupaciones de las entidades administrativas de cada ciudad y localidad; de acuerdo a trabajos técnicos del Departamento del Medio Ambiente del Banco Mundial realizado en el año 2011, la recolección de basura consume un 30% a 60% de las rentas municipales, lo que ocasiona que se reduzcan recursos financieros para las demás necesidades urbanas como lo son: la educación, alcantarillado, servicios de emergencia, alumbrado, pavimentación de la vía pública, etc.

El diario el País en el año 2012, en una de sus investigaciones revela que la gran cantidad de basura generada día a día en América Latina y el Caribe supera las 436.000 toneladas de residuos sólidos urbanos; además este problema se agrava por la deficiente planificación de rutas para la recolección del mismo al igual que técnicas de selección para conocer el estado de los contenedores previa su recolección y el envío de los respectivos recolectores. Los malos sistemas de recolección de basura tienen efectos nocivos en la ciudadanía, debido a la tardanza para recoger la misma, se generan enfermedades, un ejemplo claro de esto son las infecciones nicóticas ocasionada por la acumulación de basura que propicia hongos y bacterias que al estar en contacto con la piel provoca irritaciones e infecciones; pero no solo la población se ve afectada de esta manera, el medio ambiente también ya que se da una contaminación tanto física, química biológica y visual.

En Ecuador los sistemas para la recolección de basura han tenido una evolución poco significativa y de lento avance, dado que en los últimos diez años ha ocurrido una transición de los camiones de basura cuyo trabajo se realizaba de manera manual con un grupo aproximado

de cuatro a cinco trabajadores por unidad, a camiones con un sistema automatizado para el alcance y lavado de los contenedores, pero que solo ha permitido mejorar el almacenamiento de basura, más no el control del nivel de basura que puede ingresar en el contenedor; el Instituto Nacional de Estadísticas y Censos en su Módulo de Información Ambiental para hogares, de acuerdo al censo del año 2014, revela que la acumulación de basura a nivel nacional constituye un 25.99% entre los principales problemas ambientales.

En la ciudad de Quito la recolección de basura de los diferentes contenedores se la realiza cada cierto número de días, siguiendo una zona de recolección establecida, haya o no la necesidad, es decir los contenedores tenga o no basura; esta falta de orden en el sistema ocasiona que en muchas zonas exista una acumulación de basura generando malos olores y enfermedades a las personas tales como enfermedades gastrointestinales, cutáneas, respiratorias, entre otras, esta poca organización tiene consecuencias a gran escala, pues las pérdidas en cuanto a recursos económicos y humanos son cuantiosas y se ven afectadas en otras áreas sociales.

La recolección de basura no se limita solo a este ámbito, la basura es un problema que va aumentando a medida que transcurren los años y la ausencia de un adecuado manejo y control de la misma es una necesidad imperiosa tanto para las zonas urbanas como rurales y para las entidades públicas o privadas encargadas de su manejo ya que se convertiría en un mecanismo para evitar la pérdida de recursos monetarios y personales innecesarios. Además, esto impediría la mala manipulación de la basura, esta realidad se ve reflejada a las afueras de la ciudad debido a que las personas dejan los desechos sólidos a la intemperie, tanto en laderas como ríos.

#### **IV.- JUSTIFICACIÓN DEL PROBLEMA**

- **Área Académica:**  
Comunicaciones
  
- **Líneas de Investigación:** Tecnologías de Comunicación

- **Sublínea de Investigación:** Comunicaciones Inalámbricas
- **Delimitación Espacial:**  
La presente investigación estará orientada a realizarse en la ciudad de Quito mediante un prototipo.
- **Delimitación Temporal:**  
La investigación se desarrollará en los seis meses posteriores a la aprobación del proyecto de investigación por parte del Honorable Consejo Directivo de la Facultad de Ingeniería en Electrónica Digital y telecomunicaciones.

## V.- JUSTIFICACIÓN

Los contenedores de basura tienen como principal función mantener los sectores y la ciudad en general, de una manera limpia y libre de desechos sólidos en las aceras y en los alrededores de los contenedores, evitando que esta quede a la intemperie cuando el recolector de basura no la ha ido a recoger o a la espera que según el horario de recolección sea el tiempo de dicho sector.

La ruta de recolección de basura para cada sector está establecida por días según un estimado de llenado de cada contenedor, un ejemplo de esto es la denominada zona crítica del Mercado mayorista al Sur de la Ciudad, cuya recolección se realiza los días lunes, miércoles y viernes, por la cantidad de basura que se genera en dicho lugar, pero aun así no es suficiente ya que el tiempo en el que se llena dicho contenedor es impredecible; para este y los demás contenedores que posee la Empresa Pública EMASEO, se cuenta con seis vehículos de carga lateral que, según el día y sector, salen a recoger la basura de los contenedores de cada uno de los sectores de acuerdo a la planificación .

Debido a que la recolección de basura se lo realiza por medio de una estimación y en base a esto se encuentra la sectorización, no se puede conocer de manera exacta el estado de llenado del contenedor y esto ocasiona que muchas de las veces los vehículos de carga lateral se dirijan hacia una ruta establecida sin ningún fin, provocando pérdidas, tanto de tiempo como

económicas para EMASEO. Este hecho se ve agravado en zonas donde el lapso de llenado de un contenedor no es constante, como es el caso de sectores aledaños a mercados y lugares de recreación tal es el caso del Mercado Mayorista.

El desarrollo del presente proyecto es de gran importancia debido a la necesidad de EMASEO de poseer un óptimo control y monitoreo de los contenedores de basura del sector del Mercado Mayorista con el fin de mejorar los recursos financieros, humanos y logísticos de los vehículos de carga lateral de tal manera que posean rutas de recolección y el personal necesario de acuerdo al nivel de llenado de los contenedores y no como se lo ha venido haciendo en la actualidad.

El impacto social definitivamente será significativo ya que se evitará la presencia de desechos sólidos fuera del contenedor generando una mala imagen a la ciudad, permite reducir costos en cuanto a vehículos de carga lateral que no logran, según su ruta y día recolectar basura ocasionando una pérdida en cuanto al presupuesto que podría ser direccionado hacia otras actividades o áreas que requieren mayor atención y, por ende los moradores del sector se encontrarán más satisfechos por el servicio prestado por la entidad.

En cuanto a la factibilidad, la implementación de este sistema es totalmente realizable ya que los diferentes componentes electrónicos que se necesitan para su ejecución pueden ser adquiridos sin dificultad, por otro lado el software con el cual se desea trabajar puede ser desarrollado en base a cada uno de los conocimientos adquiridos durante la carrera y al poder trabajar con hardware libre el costo del dispositivo será muy conveniente para la empresa encargada de los desechos sólidos.

## **VI.- OBJETIVOS DE LA INVESTIGACIÓN**

### **OBJETIVO GENERAL**

- Desarrollo de un prototipo a escala 1:10 para el control y monitoreo de los contenedores de basura en el Mercado Mayorista en el sector Sur de la ciudad de Quito.

## **OBJETIVOS ESPECÍFICOS**

- Examinar el actual sistema de gestión de rutas utilizado por la Empresa Pública EMASEO de la ciudad de Quito.
- Determinar las características del sistema de control y monitoreo de contenedores de basura para la Empresa Pública EMASEO de la ciudad de Quito.
- Diseñar un prototipo para el control y monitoreo de los contenedores de basura en el Mercado Mayorista de la ciudad de Quito.
- Implementar la base de datos para el almacenamiento de información de los diferentes contenedores de basura.
- Validar funcionamiento.

## **VII.- DESCRIPCIÓN DE LOS CAPITULOS**

A continuación, se hace una breve descripción de los capítulos de este proyecto de investigación:

**Capítulo 1:** Consta de los antecedentes sobre investigaciones de Sistemas de control y monitoreo, al igual que investigaciones sobre comunicaciones inalámbricas. Además, se desarrolla el marco teórico basado en el tema de investigación.

**Capítulo 2:** Se detalla el problema que posee el actual sistema de recolección de basura en la ciudad de Quito además de la justificación junto con los objetivos propuestos para este proyecto de investigación.

**Capítulo 3:** La propuesta la cual describe las diferentes técnicas de investigación a las que se debió recurrir, la forma de obtención y análisis de la información, además de las pautas para el desarrollo del proyecto.

Se muestran las conclusiones y recomendaciones obtenidas durante el desarrollo y construcción del Prototipo para el Control y Monitoreo de los contenedores de basura del Mercado mayorista al Sur de Quito.

## **CAPÍTULO I.**

### **1. FUNDAMENTACIÓN TEÓRICA**

#### **1.1. FUNDAMENTACIÓN TEÓRICA**

##### **1.1.1. RECOLECCIÓN DE BASURA**

La recolección de basura es la actividad de recoger los desechos sólidos que han sido generados por una determinada población, para posteriormente ser transportados al sitio de almacenamiento o planta de procesamiento de basura.

##### **PARÁMETROS PARA LA RECOLECCIÓN DE BASURA**

Los métodos de recolección de basura se relacionan directamente con los siguientes parámetros.

- Características del lugar
- Características del equipamiento
- Hábitos de la población

##### **MÉTODOS DE RECOLECCIÓN**

Los métodos de recolección van de acuerdo al grado de tecnificación del equipo.

- Mecanizados
- Semi-Mecanizados.
- Manuales.

Por el tipo de operación existen:

- Contenedores
- Esquina o parada fija
- Acera
- Intra-domiciliario o llevar y traer

### 1.1.2. CONTENEDORES

Los contenedores de basura son recipientes con diferentes dimensiones, donde las personas arrojan desechos de todo tipo. Estos contenedores generalmente están fabricados a partir del metal o plástico y se encuentran ubicados en espacios públicos y dentro de los hogares. En la figura 1.1 se observa un contenedor de uso público.



Figura. 1.1. Contenedor de basura.

Fuente: Sodimac,2018

#### **Beneficios:**

Los beneficios que ofrecen los contenedores son:

- Higiénicos para el usuario y el medio-ambiente.
- A disposición permanente del usuario.
- Posee amortiguadores neumáticos que permiten un cierre suave de las tapas.

- Su diseño y estampado geométrico impide la filtración del agua lluvia y facilita un rápido deslizamiento de los residuos durante la descarga.
- Permite que la tapa se cierre de manera automática para que los olores y demás bacterias no se propaguen de esta manera se asegura hermeticidad del contenedor.

### 1.1.3. SISTEMAS DE CONTROL

Un sistema de control es un conjunto de elementos que trabajan de manera conjunta y ordenada para conseguir un objetivo de control, como se indica en la figura 1.2, el sistema está conformado por una variable de entrada, una variable de salida que va a ser la respuesta final o resultado y la perturbación que es una señal que afecta el valor de la variable de salida.



Figura. 1.2. Sistema de control

Fuente: Elaboración propia

### 1.1.4.- CARACTERÍSTICAS DE UN SISTEMA DE CONTROL

Los sistemas de control deben tener las siguientes características para cumplir con su objetivo de control:

- Ser estables en caso que exista alguna señal indeseada (perturbación).
- Ser seguros en cuanto a la información que entrega, evitando resultados bruscos e irreales.

#### 1.1.4.1.- CLASIFICACIÓN DE LOS SISTEMAS DE CONTROL

Los sistemas de control tienen la siguiente clasificación:

### Sistemas de control en lazo abierto

Un sistema de control de lazo abierto es aquel donde la variable de salida no tiene efecto sobre la variable de control, como se observa en la figura 1.3.



Figura. 1.3. Sistema de Control de lazo abierto

Fuente: Elaboración Propia

### Sistema de control en lazo cerrado

Los sistemas de control de lazo cerrado o también denominados sistemas de retroalimentación, son aquellos en donde la variable de salida tiene efecto sobre la variable de control, es decir la salida se compara con el valor de entrada para establecer el error, como se indica en la figura 1.4.

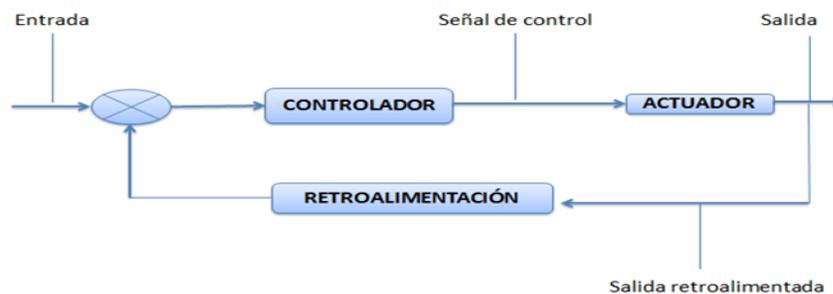


Figura. 1.4. Sistema de Control de lazo cerrado

Fuente: Elaboración Propia

### 1.1.5.- SISTEMAS DE MONITOREO

Los sistemas de monitoreo son aquellas herramientas que permiten tener una administración y supervisión de los dispositivos en la estación remota, para su posterior análisis, está compuesto por elementos tales como:

- Software o herramienta de monitoreo
- Indicadores a monitorear
- Variables a monitorear
- Gráficos de resultados
- Responsable de monitorear, etc.

### 1.1.6.- SENSORES

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas.

Los sensores tienen las siguientes características estáticas:

**Rango de medida:** El rango de medida son valores comprendidos entre el valor máximo y mínimo detectado por el sensor.

**Resolución:** La resolución permite conocer qué porcentaje mínimo de la señal que ingresa al sistema es posible detectar a su salida.

**Precisión:** La precisión es el valor promedio obtenido entre la medida de salida real y la medida de salida teórica.

**Repetitividad:** La repetitividad es la capacidad de repetir o reproducir una lectura con una precisión establecida.

**Linealidad:** Se tiene linealidad si los incrementos del valor de salida son directamente proporcional con los incrementos de la señal de entrada.

**Sensibilidad:** La sensibilidad es la razón de cambio de la salida frente a cambios en la entrada.

**Ruido:** El ruido es cualquier perturbación o variación ya sea externa o interna que afecte la señal que va a ser medida.

Los sensores tienen las siguientes características dinámicas:

**Velocidad de respuesta:** La velocidad de respuesta indica la capacidad del sensor para determinar el retraso de la señal de salida con respecto a la señal de entrada.

**Respuesta en frecuencia:** La respuesta de frecuencia mide la capacidad del sensor para seguir la señal de entrada según aumenta la frecuencia.

**Estabilidad:** La estabilidad muestra la diferencia en la salida del sensor con respecto al valor teórico dado.

### 1.1.6.1 CLASIFICACIÓN DE LOS SENSORES

La clasificación de los sensores es la siguiente:

#### 1.1.6.1.1 SEGÚN EL PRINCIPIO DE FUNCIONAMIENTO

Según el principio de funcionamiento los sensores se clasifican en:

- **SENSORES PASIVOS**

Los sensores pasivos son aquellos sensores que generan señales representativas de las magnitudes a medir por intermedio de una fuente auxiliar.

- **SENSORES ACTIVOS**

Los sensores activos son aquellos que generan señales representativas de las magnitudes a medir en forma autónoma, sin requerir alguna fuente de alimentación, un ejemplo de este tipo de sensor es una termocupla como se indica en la figura 1.5.



Figura. 1.5. Termocupla.

Fuente: Arian, 2015

#### 1.1.6.1.2 SEGÚN EL TIPO DE SEÑAL QUE GENERAN

Según el tipo de señal que los sensores generan se clasifican en:

- **SENSORES DIGITALES**

Los sensores digitales son los que por medio de una señal pasan de un estado a otro y los valores de voltaje que se obtienen son de 5V y 0V.

- **SENSORES ANÁLOGOS**

Los sensores análogos a la salida, retransmite una señal comprendida por un campo de valores rápidos que cambian en el tiempo y son proporcionales a los resultados que se están midiendo, un ejemplo de esto es el sensor analógico de temperatura que se muestra en la figura 1.6.



Figura. 1.6. Sensor analógico temperatura

Fuente: Electrónica, 2018

### 1.1.6.1.3.- SEGÚN EL RANGO DE VALORES DE SALIDA

Según el rango de los valores de salida los sensores se clasifican en:

- **SENSORES On/Off**

Los sensores On/Off son conocidos como sensores si-no, sensores 0-1 o sensores binarios, son dispositivos mecánicos entre los más importantes están:

- Interruptores Reed, que se enlazan por la proximidad de un imán.
- Interruptor de péndulo, consiste en un peso que se suspende de un hilo conductor dentro de un anillo metálico y el movimiento del anillo ocasiona el cierre del circuito.

- **SENSORES DE MEDIDA**

Los sensores de medida o sensores de nivel son dispositivos electrónicos que miden la altura del material; es decir, se tiene una salida proporcional a la señal de entrada. Un ejemplo de un sensor de nivel es: el sensor ultrasónico sin contacto.

#### 1.1.6.1.4.- SEGÚN EL TIPO DE VARIABLE FÍSICA MEDIDA

Según el tipo de variable física que los sensores son capaces de medir se clasifican en:

- **SENSORES MECÁNICOS**

Los sensores mecánicos son instrumentos que modifican su comportamiento por la acción de un valor físico que pueden directa o indirectamente afectarlos.

Los sensores mecánicos son utilizados para medir:

Desplazamiento, posición, tensión, movimiento, presión. En la figura 1.7, se puede observar un módulo de potencia. (Sensor mecánico)



Figura. 1.7. Sensor mecánico

Fuente: Blogspot, 2012

- **SENSORES DE PESO**

Los sensores de peso son aquellos dispositivos eléctricos que tienen la finalidad de reconocer o detectar las variaciones eléctricas ocasionadas por una variante en la intensidad de un peso aplicado. Un ejemplo de este tipo de sensor se puede observar en la figura 1.8, que es una celda de carga WLC-SB.



Figura. 1.8. Celda de carga WLC-SB.

Fuente: IPC, 2018

#### 1.1.7. SOFTWARE

El Software es un programa o conjunto de programas que permiten realizar varias tareas en un sistema informático, el software posee datos, cálculos, procedimientos y reglas para obtener el objetivo deseado.

Para el sistema a realizarse la interfaz gráfica de monitoreo y control que va a ser utilizada se lo hará mediante el desarrollo de una aplicación Web y su IDE oficial de Android llamado Android Studio ya que permite el desarrollo de aplicaciones de una manera intuitiva.

#### **1.1.7.1.- Android Studio**

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA . Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ; Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android, como las siguientes:

- Un sistema de compilación basado en Gradle flexible
- Un emulador rápido con varias funciones
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android
- Instant Run para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código
- Gran cantidad de herramientas y frameworks de prueba
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

Cada proyecto en Android Studio contiene uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos se incluyen los siguientes:

- módulos de apps para Android
- módulos de bibliotecas
- módulos de Google App Engine

De manera predeterminada, Android Studio muestra los archivos de tu proyecto en la vista de proyectos de Android, como se muestra en la figura 1.9. Esta vista se organiza en módulos para proporcionar un rápido acceso a los archivos de origen clave de tu proyecto.

Todos los archivos de compilación son visibles en el nivel superior de Secuencias de comando de Gradle y cada módulo de la aplicación contiene las siguientes carpetas:

- manifests: contiene el archivo AndroidManifest.xml.
- java: contiene los archivos de código fuente de Java, incluido el código de prueba JUnit.
- res: Contiene todos los recursos, como diseños XML, cadenas de IU e imágenes de mapa de bits.

La estructura del proyecto para Android en el disco difiere de esta representación plana. Para ver la estructura de archivos real del proyecto, selecciona Project en la lista desplegable Project.

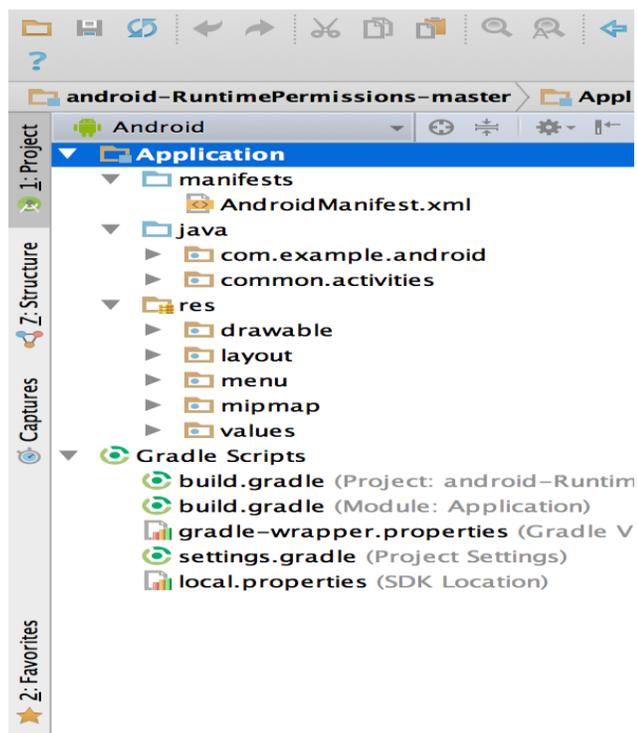


Figura. 1.9. Archivos del proyecto en la vista de Android.

Fuente: Developer, 2017

También se puede personalizar la vista de los archivos del proyecto para concentrarse en aspectos específicos del desarrollo de tu App. Por ejemplo, al seleccionar la vista Problems de tu proyecto, aparecerán enlaces a los archivos de origen que contengan errores conocidos de codificación y sintaxis, como una etiqueta de cierre faltante para un elemento XML en un archivo de diseño mostrado en la figura 1.10.

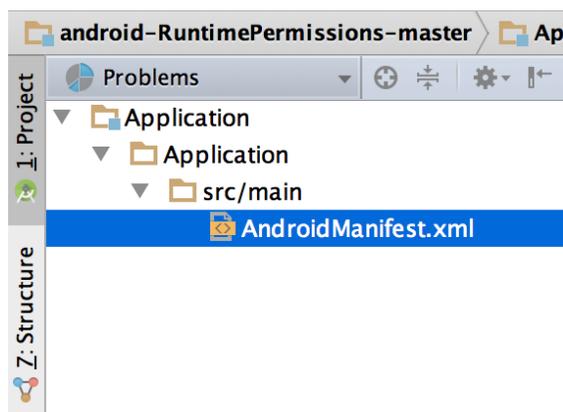


Figura. 1.10. Archivos del proyecto en la vista Problems, en la que se muestra un archivo de diseño.

Fuente: Developer, 2017

## INTERFAZ DE USUARIO

La ventana principal de Android Studio consta de varias áreas lógicas que se identifican en la figura 1.11.

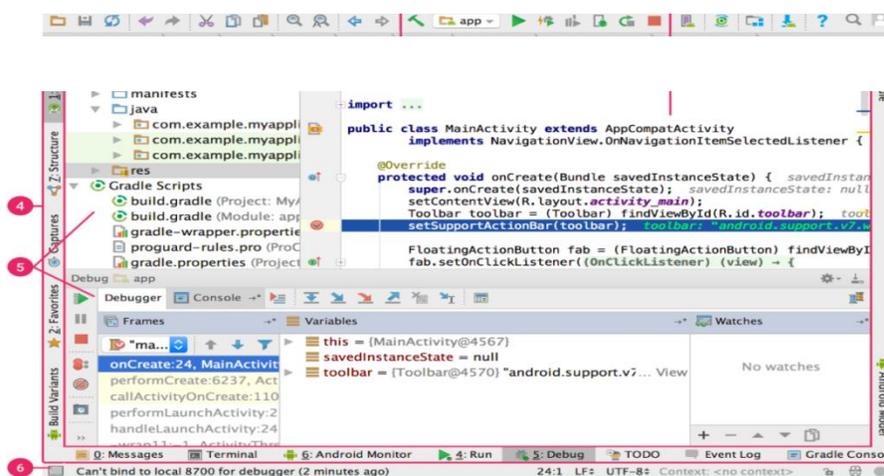


Figura. 1.11. Ventana principal de Android Studio.

Fuente: Developer, 2017

1. La barra de herramientas te permite realizar una gran variedad de acciones, como la ejecución de tu App y el inicio de herramientas de Android.
2. La barra de navegación te ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana Project.
3. La ventana del editor es el área donde puedes crear y modificar código. Según el tipo de archivo actual, el editor puede cambiar. Por ejemplo, cuando se visualiza un archivo de diseño, el editor muestra el editor de diseño.
4. La barra de la ventana de herramientas se extiende alrededor de la parte externa de la ventana del IDE y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.
5. Las ventanas de herramientas te permiten acceder a tareas específicas, como la administración de proyectos, las búsquedas, los controles de versión, etc. Puedes expandirlas y contraerlas.
6. En la barra de estado, se muestra el estado de tu proyecto y del IDE en sí, como también cualquier advertencia o mensaje.

Puedes organizar la ventana principal para tener más espacio en pantalla ocultando o desplazando barras y ventanas de herramientas. También puedes usar combinaciones de teclas para acceder a la mayoría de las funciones del IDE.

En cualquier momento, puedes realizar búsquedas en tu código fuente, bases de datos, acciones, elementos de la interfaz de usuario, etc., presionando dos veces la tecla Shift o haciendo clic en la lupa que se encuentra en la esquina superior derecha de la ventana de Android Studio. Esto puede ser muy útil, por ejemplo, si intentas localizar una acción específica del IDE que olvidaste cómo activar.

#### **1.1.7.2.- Amazon Web Service (AWS)**

Amazon Web Services (AWS) es una plataforma de servicios de nube que ofrece potencia de cómputo, almacenamiento de bases de datos, entrega de contenido y otra funcionalidad para ayudar a las empresas a escalar y crecer. Explore cómo millones de clientes aprovechan

los productos y soluciones de la nube de AWS para crear aplicaciones sofisticadas y cada vez más flexibles, escalables y fiables.

Amazon Web Services (AWS) ofrece soluciones de confianza basadas en la nube para atender las necesidades de su negocio. Ejecutar sus soluciones en la nube de AWS puede ayudarle a poner en marcha y ejecutar sus aplicaciones en menos tiempo, al tiempo que podrá disfrutar de la misma seguridad en la que confían entidades como Pfizer, Intuit y la armada estadounidense (US Navy). AWS también ofrece recursos en todo el mundo para que usted pueda implementar sus soluciones donde estén sus clientes. Con la nube de AWS, es fácil disponer de una amplia gama de servicios, socios y opciones de soporte para que usted pueda centrarse en conseguir que su solución sea todo un éxito, como se muestra en la figura 1.12

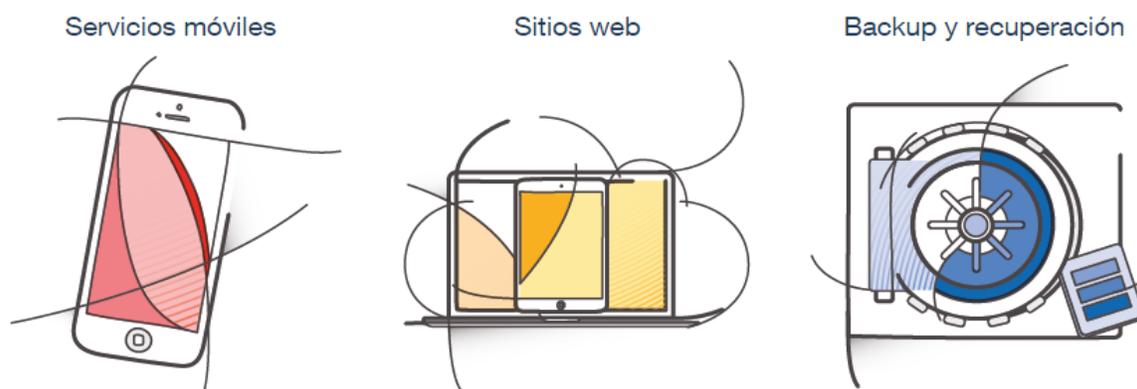


Figura. 1.12. Servicios en la nube amazon

Fuente: usb.org, 2017

### 1.1.7.3.- PHPMYAdmin

A través de Internet. phpMyAdmin es compatible con una amplia gama de operaciones de MySQL y MariaDB. (Operaciones de gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc) se puede realizar a través de la interfaz de usuario que se utiliza con frecuencia, mientras que todavía tiene la capacidad de ejecutar directamente cualquier sentencia SQL.

phpMyAdmin viene con una amplia gama de documentación y los usuarios son bienvenidos para actualizar nuestras páginas wiki para compartir ideas y howtos para diversas operaciones. El equipo de phpMyAdmin tratará de ayudarle si frente a cualquier problema; se puede utilizar una variedad de canales de soporte para obtener ayuda.

phpMyAdmin es también muy profundamente documentado en un libro escrito por uno de los desarrolladores - El dominio de phpMyAdmin para la eficaz gestión de MySQL , que está disponible en Inglés y Español .

Para facilitar el uso de una amplia gama de personas, phpMyAdmin está siendo traducido a 72 idiomas y es compatible con los dos idiomas LTR y RTL, phpMyAdmin es un proyecto de dieciocho años de edad, con una base de código estable y flexible; usted puede encontrar más información sobre el proyecto y su historia y los premios que ganó. Cuando el proyecto cumplió los 15 años, se publicó una página de celebración .

El proyecto phpMyAdmin es un miembro del Software Freedom Conservancy . SFC es una organización sin fines de lucro que ayuda a promover, mejorar, desarrollar y defender proyectos de software de código abierto (FLOSS) gratuito, libre.

### 1.1.8. ARQUITECTURA DE LA APLICACIÓN

El prototipo realizado se divide en 2 partes. La primera parte es la aplicación para el desarrollo en Android por lo que para su funcionamiento necesitamos de un dispositivo móvil Android, una vez instalada y con acceso a internet se podrá consultar desde cualquier parte del mundo, la misma se conecta al servicio Web (WebService publicado en Amazon) como lo muestra la figura 1.13, la misma a su vez se comunica con la de datos en donde es almacenada la información.

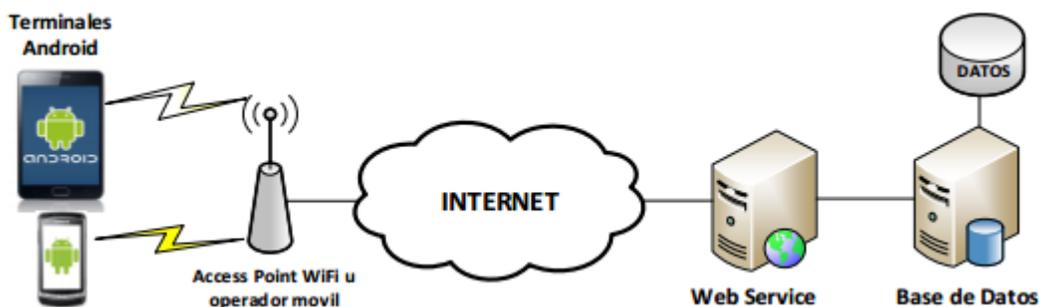


Figura 1.13. Diseño Mecánico

Fuente: (Elaboración Propia)

La Segunda parte y como un plus adicional consta de un módulo Web de Monitoreo mostrado en la figura 1.14, publicado en un servidor de Amazon, y publicada en un servidor Web Apache mediante peticiones PHP, interactúa con las peticiones del usuario desde la aplicación ya que está publicado en Internet, el agente de despacho usara un navegador (Internet Explorer) para el monitoreo de los contenedores de basura en la ciudad de Quito.

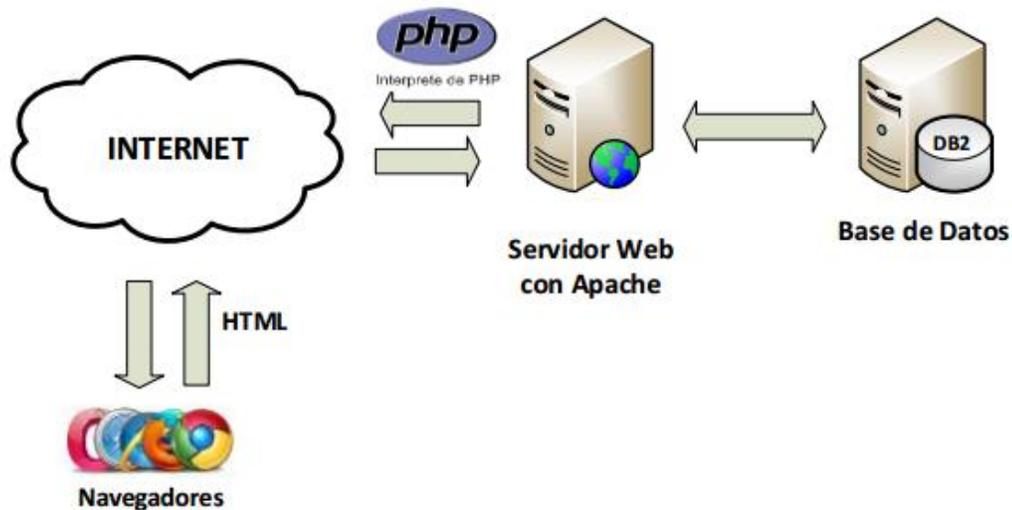


Figura. 1.14. Arquitectura de la intranet

Fuente: (Elaboración Propia)

Como se observa en la figura 1.15, la base de datos del servidor donde se almacena la información levantada sobre un servidor Linux en Amazon.

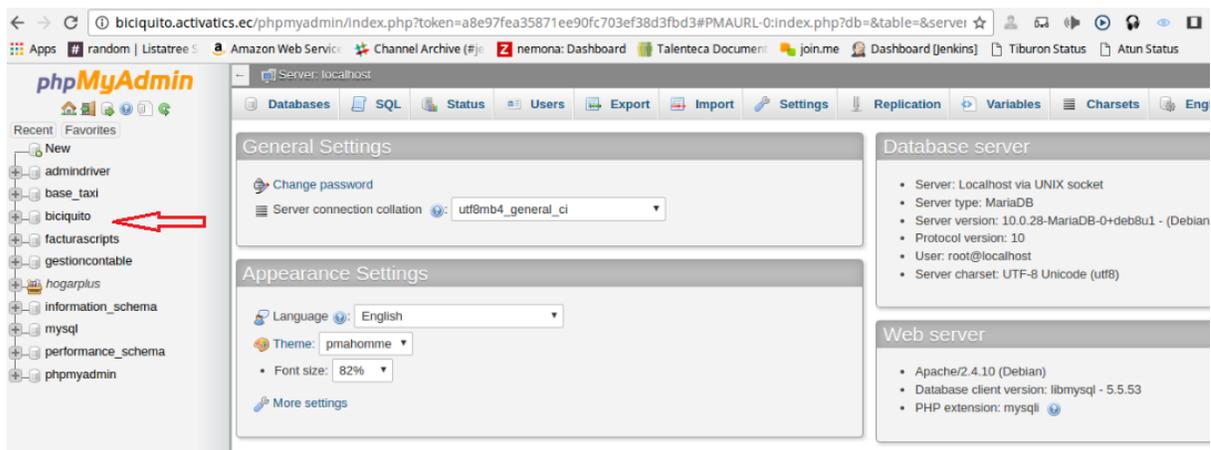


Figura. 1.15. Base de datos del servidor Linux

Fuente: (Elaboración Propia)

## WEB SERVICE (SERVICIOS WEB)

Los servicios Web son aplicaciones de cliente y servidor que se comunican sobre (www) Protocolo de transferencia de hipertexto de la World Wide Web (HTTP), proporcionan un medio estándar de interoperabilidad entre aplicaciones de software que se ejecutan en una variedad de plataformas y marcos.

Los servicios Web se caracterizan por su gran interoperabilidad y extensibilidad, así como sus descripciones procesables por máquina, gracias al uso de XML. Servicios Web se pueden combinar de una manera imprecisa para lograr operaciones complejas. Programas que proporcionan servicios simples pueden interactuar entre sí para ofrecer servicios de valor añadido sofisticado. ( docs.oracle.com 2006), como se muestra en la Fig. 1.16.

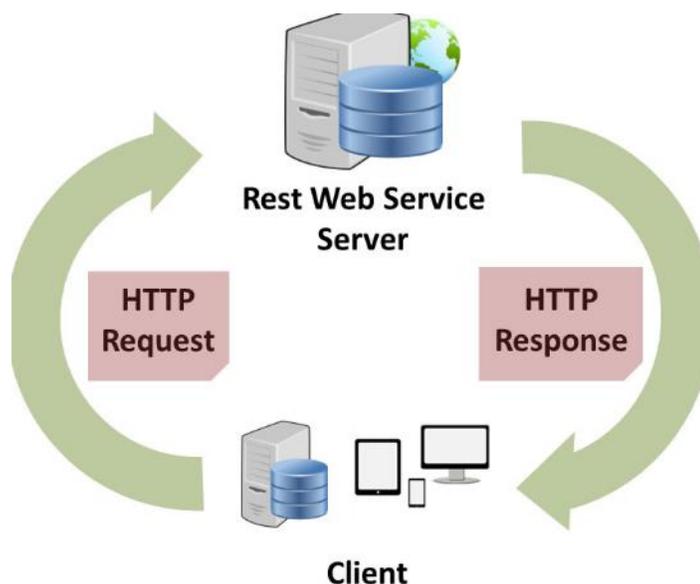


Figura. 1.16. Diagrama Web Service

Fuente: Chemaxon,2016

### 1.1.9.- HARDWARE LIBRE

#### 1.1.9.1.- Módulo GSM/GPRS SIM 900.

Esta shield convierte la placa Arduino en una plataforma capaz de ofrecer conectividad GPRS/GSM. Integra un módulo SIM900 que permite realizar llamadas con otros dispositivos

móviles, envío de SMS y la comunicación de datos mediante protocolos TCP, UDP, HTTP o FTP.

En la figura 1.17, se presenta una imagen del módulo SIM 900.



Figura. 1.17. Módulo GPRS/GSM Quadband para arduino (Sim900)

Fuente: Simcom, 2017

### 1.1.9.2.- Características del Shield GSM/GPRS SIM 900.

Las características principales del Shield GSM/GPRS SIM 900 son:

- Operación global y con cualquier proveedor (multibanda).
- Se controla mediante comandos AT y comandos AT extendidos.
- Incluye stack TCP/IP soporta TCP, HTTP, FTP mediante comandos AT.
- Ideal para transmisión de datos sobre GPRS.
- Ancho de banda de descarga 85.6 kbps.
- Ancho de banda de subida 42.8 kbps.
- SMS en modo PDU y Texto.
- La interfaz de control del módem GSM se puede asociar con varios pines del Arduino no solamente los del UART, permitiendo usar librerías como SoftSerial.
- Incluye súper capacitor para almacenamiento persistente de datos.
- Librería para funcionar con Arduino.

### 1.1.9.3.- Diagrama de Pines SIM 900

Se puede observar en la Figura 1.18 el diagrama de pines SIM 900.

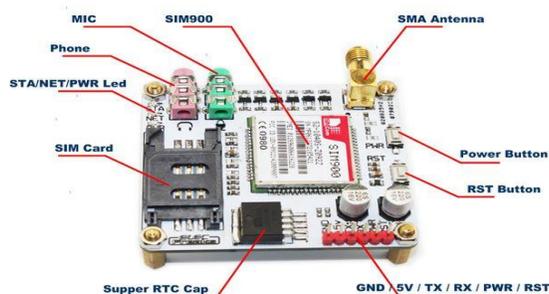


Figura. 1.18. Diagrama de pines (Sim900)

Fuente: TUXTI, 2014

#### 1.1.9.4.- Módulo GPS neo 6M Ublox

El módulo es de cerámica que incluye antena, la señal EEPROM permite guardar la configuración de los parámetros de datos. Este módulo utiliza la última tecnología para dar información de posición óptima, lo que le permite un mejor rendimiento. (GPS), se puede observar en la Figura 1.19.



Figura. 1.19. Módulo Gps

Fuente: Geekfactory, 2017

#### 1.1.9.5.- Arduino Mega 2560

Es una lámina electrónica con un procesador Atmega2560, cuenta con 54 pines digitales de entrada / salida.

Contiene una conexión USB de alimentación, o la corriente con un adaptador de AC a DC., se muestra en la Figura 1.20.

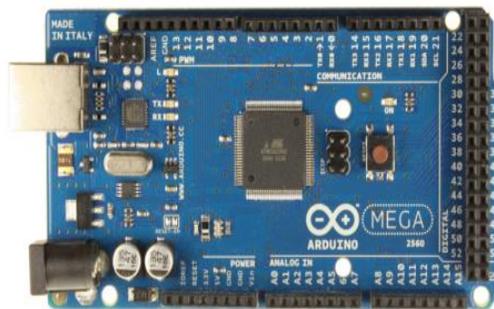


Figura. 1.20. Arduino mega 2560

Fuente: Arduino ,2017

Para la construcción de este prototipo se utilizó un dispositivo que sea compacto, ligero y que se acople al prototipo en base a sus funciones y características que se detalla en la tabla 1.2.

<b>DENOMINACIÓN</b>	<b>DESCRIPCIÓN</b>
<b>Microcontrolador</b>	Atmega2560
<b>Tensión de funcionamiento</b>	5V
<b>Voltaje de entrada (recomendado)</b>	7-12V
<b>Voltaje de entrada (límite)</b>	6-20V
<b>E / S digitales prendedores</b>	54 (de los cuales 15 proporcionan salida PWM)
<b>Pines de entrada analógica</b>	Dieciséis
<b>Corriente continua para Pin I / O</b>	20 mA8
<b>Corriente CC para Pin 3.3V</b>	50 Ma
<b>Memoria flash</b>	256 KB, 8 KB utilizado por el gestor de arranque
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Velocidad de reloj</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Longitud</b>	101.52 mm
<b>Anchura</b>	53,3 mm
<b>Peso</b>	37 g

Tabla. 1.1. Especificaciones y características técnicas del Arduino Mega 2560

Fuente: (Módulo Arduino, 2017)

En la figura 1.21, se detalla la distribución de pines en el Arduino MEGA 2560.

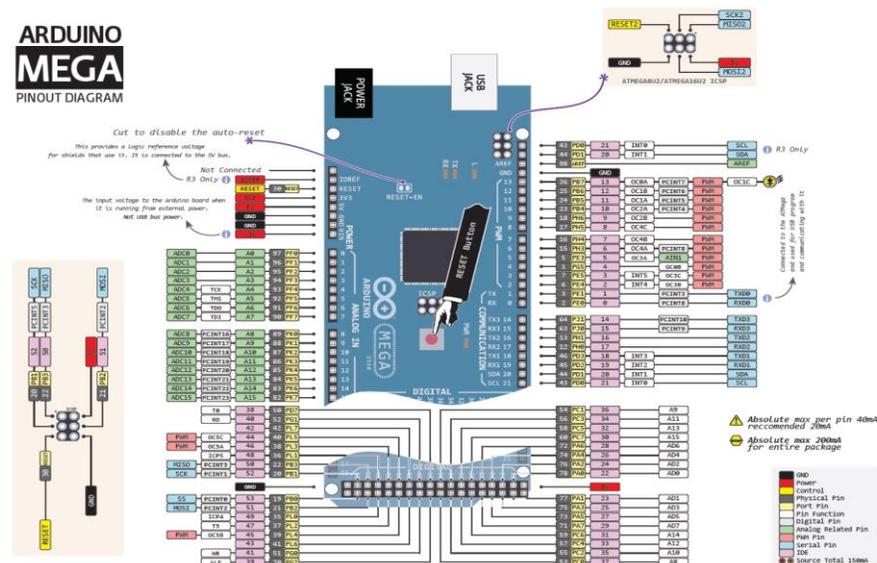


Figura. 1.21. Diagrama de pines arduino mega 2560

Fuente: Geekfactory, 2017

La Tabla 1.3 se muestra la comparación de las características técnicas de 3 módulos GPS de la familia NEO 6M ublox.

Nombre del parámetro	GPS ublox neo-06M	GPS ublox neo 6Q	GPS ublox neo 6G
Red	GPS	GPS	GPS
Tamaño	23mm*30mm	23mm*30mm	23mm*30mm
Rango de voltaje	3V a 5V	2.75V a 3.76V	1.75V a 2V
Margen de error	5m	10m	10m

Tabla. 1.2. Tabla especificaciones de los Módulos GPS

Fuente: (Morales, 2015)

## Alimentación

Puede ser alimentado vía conexión USB o con una fuente de alimentación externa DC. El origen de la alimentación se selecciona automáticamente. Las fuentes de alimentación externas no USB pueden ser tanto una batería o un adaptador de pared. (Arduino, Arduino Uno, 2003)

## Comunicación

Una de las diferencias principales de la tarjeta Arduino MEGA 2560 es que no utiliza el convertidor USB-serie de la firma FTDI. Por lo contrario, emplea un microcontrolador Atmega8U2 programado como actuar convertidor USB a serie. (Patagoniatect,2016)

Dispone de 54 entradas/salidas digitales, 14 de las cuales se pueden utilizar como salidas PWM (modulación de anchura de pulso). Además, dispone de 16 entradas analógicas, 4 UARTs (puertas series), un oscilador de 16MHz, una conexión USB, un conector de alimentación, un conector ICSP y un pulsador para el reset. Para empezar a utilizar la placa sólo es necesario conectarla al ordenador a través de un cable USB, o alimentarla con un adaptador de corriente AC/DC. También, para empezar, puede alimentarse mediante una batería, como se muestra en la Tabla 1.4.

Característica de Arduino	UNO	Mega 2560	Leonardo	DUE	
Tipo de microcontrolador	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E	
Velocidad de reloj	16 MHz	16 MHz	16 MHz	84 MHz	
Pines digitales de E/S	14	54	20	54	
Entradas analógicas	6	16	12	12	
Salidas analógicas	0	0	0	2 (DAC)	
Memoria de programa (Flash)	32 Kb	256 Kb	32 Kb	512 Kb	
Memoria de datos (SRAM)	2 Kb	8 Kb	2.5 Kb	96 Kb	
Memoria auxiliar (EEPROM)	1 Kb	4 Kb	1 Kb	0 Kb	

Tabla. 1.3. Tabla comparativa de Arduinos.

Fuente: (Sabetecnologia), 2013

### 1.1.9.6.- ARDUINO MEGA

El Arduino Mega está basado en el microcontrolador ATmega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16

entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable USB o con una fuente de alimentación externa. El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO, como se muestra en la Figura 1.22.

### Características

- Microcontrolador ATmega2560.
- Voltaje de entrada de - 7-12V.
- 54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).
- 16 entradas análogas.
- 256k de memoria flash.
- Velocidad del reloj de 16Mhz

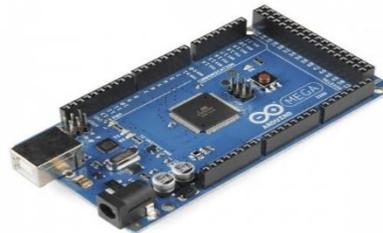


Figura. 1.22. Arduino mega

Fuente: Arduino,2017

### 1.1.9.7.- MÓDULO GSM GPRS SIM900

SIM900 tiene capacidades para voz, fax y datos, concretamente es un teléfono celular en un chip. Tal vez lo mejor del módulo es que toda la funcionalidad del mismo es accesible desde sencillos comandos AT facilitando enormemente su utilización.

Es posible también controlarlo desde las interfaces I2C o SPI, pero en este proyecto nos concentraremos en los comandos AT ya que son más fáciles de comprender y usados en la programación. Este sistema mínimo SIM900 incluye el regulador de voltaje y el zócalo para la

tarjeta SIM, por lo cual es bastante sencillo hacer que trabaje con cualquier microcontrolador, como se muestra en la Figura 1.23.

**Características:**

- Operación global y con cualquier proveedor (multibanda).
- Se controla mediante comandos AT y comandos AT extendidos.
- Incluye stack TCP/IP soporta TCP, HTTP, FTP mediante comandos AT.
- Ideal para transmisión de datos sobre GPRS.
- Ancho de banda de descarga 85.6 kbps.
- Ancho de banda de subida 42.8 kbps.
- SMS en modo PDU y Texto
- La interfaz de control del módem GSM se puede asociar con varios pines del Arduino no solamente los del UART, permitiendo usar librerías como SoftSerial
- Incluye súper capacitor para almacenamiento persistente de datos
- Librería para funcionar con Arduino.



Figura. 1.23. MODULO GSM GPRS SIM900

Fuente: HETPRO,2017

**1.1.9.8.- MÓDULO GPS UBBLOX NEO 6M**

El módulo NEO 6M de la empresa suiza UBlox es una excelente alternativa de precisión y costo beneficio, como se muestra en la Figura 1.24.

Es de un tamaño reducido por lo que es fácilmente portable en tus proyectos. Se comunica a través de puerto serial UART.

Esta presentación viene en modo modular para que se pueda usar en cualquier plataforma como Arduino, RaspberryPi, PCDuino o con tu Laptop.

Mide parámetros Latitud, Longitud, Velocidad, Altitud.

**Características:**

- Ultra sensibilidad: -165dBm
- 22 tracking/66 acquisition-channel receiver
- Soporta estándares WAAS/EGNOS/MSAS/GAGAN
- Frecuencia de actualización 5Hz
- Velocidad de desplazamiento máxima: 500m/seg
- Protocolo NMEA (a 9600bps)
- 01 puerto serial
- Antena incorporada de 18.2 x 18.2 x 4.0 mm
- Rango de temperatura: -40 to 85 C
- Cumple estándar RoHS
- Tamaño reducido 30mm x20mm x 11.4mm
- Batería lipo 7,4v 1500ma
- Buzer
- Pulsador
- Caja
- Baquelita
- Cables para conectar
- conectores
- Led
- Switch



Figura. 1.24. Módulo Gps ubblox neo 6m

Fuente: HETPRO, 2017

## **CAPÍTULO II**

### **2.- PROPUESTA**

#### **2.1. PROPUESTA DE SOLUCIÓN**

La propuesta de solución para el presente proyecto de investigación es el desarrollo de un prototipo que consiste en sensores ubicados en los contenedores de basura, encargados de detectar tanto el nivel como el peso de los desechos, una vez que estos han alcanzado el 80% en cuanto a nivel y 400 kg en peso, se emite una alarma mediante una red inalámbrica que será monitoreada a través de una estación remota con una interfaz hombre-máquina.

#### **2.2.- PROPUESTA DE DISEÑO**

En este capítulo se presenta el estudio diseño e implementación del prototipo electrónico, tanto en hardware, software planteado diagramas circuitales, simulaciones, diagramas de flujo y demás ítems que dan pasó al proceso de implementación del prototipo.

#### **2.3.- MODALIDAD DE LA INVESTIGACIÓN**

El presente proyecto es una investigación aplicada ya que tiene como finalidad la aplicación de conocimientos ya existentes para la resolución de un determinado problema, en este proyecto de investigación las modalidades que se utilizaron son las siguientes:

Se recurrió a la modalidad bibliográfica, y se obtuvo la información y datos mediante fuentes secundarias como libros, revistas, periódicos, tesis, páginas web, para poder aumentar los conocimientos y de esta manera indagar, conocer, actualizar y verificar diferentes datos ya obtenidos previamente.

Se empleó la metodología de campo porque permitió al investigador adquirir información exacta y veraz del lugar en donde es necesaria la implementación del proyecto y principal

fuentes de problemas que necesitan ser solucionados.

#### **2.4.- POBLACIÓN Y MUESTRA**

Para el desarrollo de este proyecto de investigación se contó con la siguiente población:

- Personal técnico de EMASEO
- Moradores y comerciantes del sector del Mercado Mayorista, ubicado en el Barrio el Carmen las calles Teniente Hugo Ortiz y Ayapamba en la ciudad de Quito, la misma que cuenta con 10 contenedores de basura.

Este proyecto de investigación debido a las características técnico-científicas que presenta no requirió una muestra.

#### **2.5.- RECOLECCIÓN DE INFORMACIÓN**

Para la adquisición de información se obtuvo datos de personas capacitadas en este tema como lo es el personal que trabaja en EMASEO encargados del área de recolección de basura, quienes otorgaron información sobre las diferentes necesidades e inconvenientes que se presentan al momento de realizar su labor, de la misma manera se recopiló opiniones con el motivo de conocer la temática y situación actual de los contenedores de basura; así también documentos que avalaron como se maneja el actual sistema de acopio de basura para cada sector otorgado por los Ingenieros a cargo del área de mantenimiento.

Además, se recopiló información a través de textos que resultaron de gran utilidad para el correcto análisis de las soluciones más factibles y viables al problema planteado.

#### **2.6.- PROCESAMIENTO Y ANÁLISIS DE DATOS.**

Para la presente investigación se realizó el procesamiento y análisis de la información con los parámetros detallados a continuación:

Análisis y estudio de las fuentes bibliográficas como libros de textos, páginas web, tesis, así como datos fiables y verídicos otorgados por la EMASEO

1. Revisión del estado actual de los contenedores ubicados en el sector del Mercado Mayorista en la ciudad de Quito.
2. Análisis e interpretación de los Resultados.

## **2.7.- DESARROLLO DEL PROYECTO**

Para el correcto desarrollo del presente proyecto se detallan los siguientes aspectos básicos:

1. Determinar las características de funcionamiento y operación de trabajo, al momento de realizar la recolección de la basura en el Mercado Mayorista de la ciudad de Quito.
2. Analizar el actual sistema que utiliza EMASEO para la recolección de basura.
3. Realizar un estudio de campo del actual sistema de gestión de rutas de EMASEO.
4. Investigar los sistemas de control y monitoreo a través de libros, páginas web, tesis, etc.
5. Sintetizar los datos más relevantes obtenidos a partir de la investigación de los sistemas de control y monitoreo.
6. Analizar los dispositivos y equipos necesarios para establecer un prototipo para el control y monitoreo de los contenedores de basura.
7. Diseñar un prototipo para el control y monitoreo de los contenedores de basura.
8. Seleccionar el software de acuerdo a los requerimientos de EMASEO.
9. Diseñar la aplicación para el control y monitoreo de los contenedores de basura del Mercado Mayorista de la ciudad de Quito.
10. Implementar el Prototipo (diseño del Prototipo, control, monitoreo y aplicación móvil).
11. Realizar las pruebas de funcionamiento y corrección de errores para el ajuste del prototipo.

## **2.8.- ANÁLISIS DE FACTIBILIDAD**

El estudio de factibilidad permitió conocer y evaluar la necesidad de este proyecto de investigación y su viabilidad en cuanto a varios factores que intervienen para su realización, como son:

### **2.8.1.- Factibilidad Institucional**

Existe la factibilidad institucional para este proyecto de investigación, debido a la necesidad de EPM-GIDSA de poseer un sistema que ayude a mejorar el actual sistema de rutas a fin de optimizar la recolección de los desechos sólidos de los ecotachos ubicados en el Mercado Sur en el cantón Ambato.

### **2.8.2.- Factibilidad Técnica**

La factibilidad técnica para este proyecto de investigación existe, dado que se cuenta con todas las herramientas tecnológicas para el desarrollo del Prototipo para el control y monitoreo de los contenedores de basura en el sector del Mercado Mayorista de la ciudad de Quito.

### **2.8.3.- Factibilidad Bibliográfica**

Se tiene factibilidad bibliográfica debido a que se cuenta con la información necesaria obtenida de diferentes fuentes, tanto de la Facultad de Ingeniería en Sistemas, Electrónica al igual que EMASEO

### **2.8.4.- Factibilidad Económica**

La factibilidad económica de este proyecto de investigación es positiva, debido a que el valor total de este sistema es accesible para el investigador, en cuanto a hardware y software se refiere, incluyendo mano de obra y otros componentes electrónicos que se requieran para el desarrollo del sistema.

## **2.9.- INFORMACIÓN DEL ACTUAL SISTEMA DE RECOLECCIÓN DE BASURA**

El sistema para la recolección de basura que utiliza actualmente EMASEO está basado en una planificación de rutas que incluye vehículos, días y la distancia recorrida a cada destino. Esta planificación se ha hecho de acuerdo a un estimado del llenado de

los desechos sólidos de cada contenedor de basura de acuerdo al sector en que se encuentre, para su posterior recolección mediante los vehículos de carga lateral.

En la planificación se destacan las llamadas zonas críticas, debido a la cantidad de basura que se generan en estos contenedores al encontrarse ubicados en sectores de gran afluencia de personas, tal es el caso del Mercado Mayorista, que cuenta con 10 contenedores de basura ubicados de tal manera que los moradores puedan acceder a ellos desde cualquier punto del mercado.

Estos contenedores de basura están ubicados en toda la provincia de Pichincha, con las mismas características, son de tipo mecánico con capacidad para 2400 litros del modelo CMR 2400/300, como se detallara a continuación.

## 2.10.- CONTENEDORES METÁLICOS CMR-2400/300

Los contenedores metálicos Themac son elaborados con tecnología mecánica, enfocada hacia todas las necesidades y la atención de las peculiaridades del usuario latinoamericano. Las dimensiones de los contenedores han sido analizadas en conjunto con equipos de carga lateral para certificar una completa funcionalidad en las operaciones de recolección y lavado. Los contenedores metálicos brindan una forma higiénica y técnica de manipular los residuos sólidos tanto para los operadores, moradores y medio ambiente. Con capacidad para 2400 litros de acuerdo al modelo que se esté utilizando, en el caso del Mercado Mayorista son los CMR-2400/300, como se muestra en la figura 2.1. Estos contenedores se encuentran para el uso permanente de los moradores y tienen una vida útil promedio de 10 años y son color plomo. Estas características se detallan en la tabla 2.1.

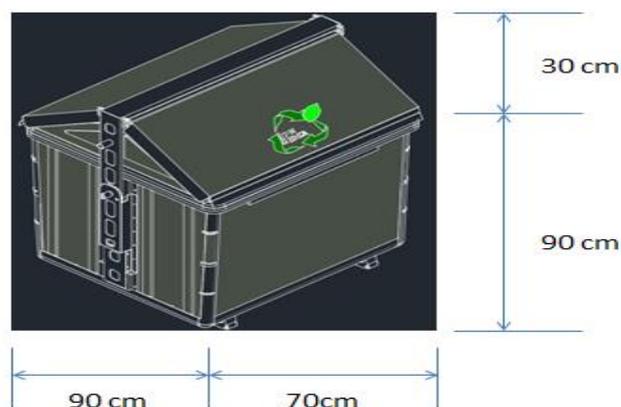


Figura. 2.1. Original Contenedor de basura

Fuente: Elaboración Propia

También se detalla en la figura 2.2, las medidas del prototipo del contenedor de basura.

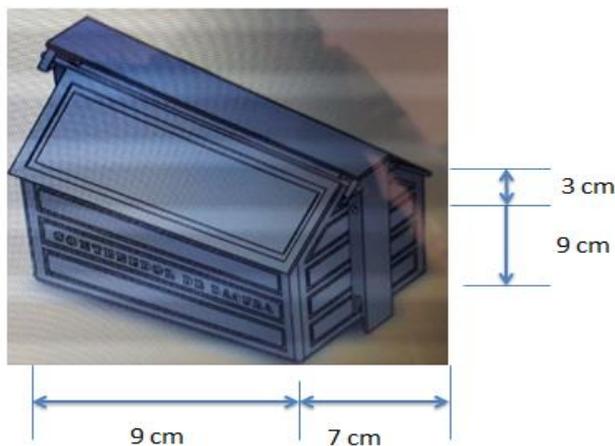


Figura 2.2. Prototipo Contenedor de basura

Fuente: Elaboración Propia

CONTENEDORES MÉTALICOS	
Volumen	<b>2400 Litros</b>
Carga	<b>500 Kg.</b>
Volumen (Prototipo)	<b>1.5 Litro</b>
Carga (Prototipo)	<b>3 Kg</b>
Peso del contenedor	<b>70 Kg.</b>
Peso del contenedor (Prototipo)	<b>0.5 Kg.</b>
Vida útil	<b>10 años</b>
Fecha de elaboración (Prototipo)	<b>15 Enero</b>
Modelo	<b>RSU- para residuos domiciliarios</b>
Modelo (Prototipo)	<b>Contenedor de basura</b>
Color	<b>Plomo</b>
Color (Prototipo)	<b>Azul</b>
Año de Compra	<b>2010</b>
Año de Elaboración (Prototipo)	<b>2018</b>
Material de Construcción	<b>Polietileno, estabilizado contra la radiación U.V.</b>
Material de Construcción (Prototipo)	<b>Termoplástico</b>

Tabla. 2.1. Características de los contenedores Prototipo y Original.

Fuente: (Elaboración propia)

## 2.11.- RUTAS DE RECOLECCIÓN DE BASURA

Las rutas de recolección de basura están establecidas según un estimado de llenado o la afluencia de gente que tenga dicho sector, esto se lo hace por días como se detalla en la tabla 2.2. En esta tabla se puede apreciar que está dividido por zonas, la 20, 21, 22 y 28 en total con un número de 235 contenedores a cargo de EMASEO en donde se destaca las zonas críticas, que corresponde a la Av. Teniente Hugo Ortiz, Mercado Mayorista, Hospital de San Bartolo, entre otras.

Además en la tabla 2.2 se observa la hora de salida de cada camión de carga lateral la zona y el número de viajes que realiza diariamente.

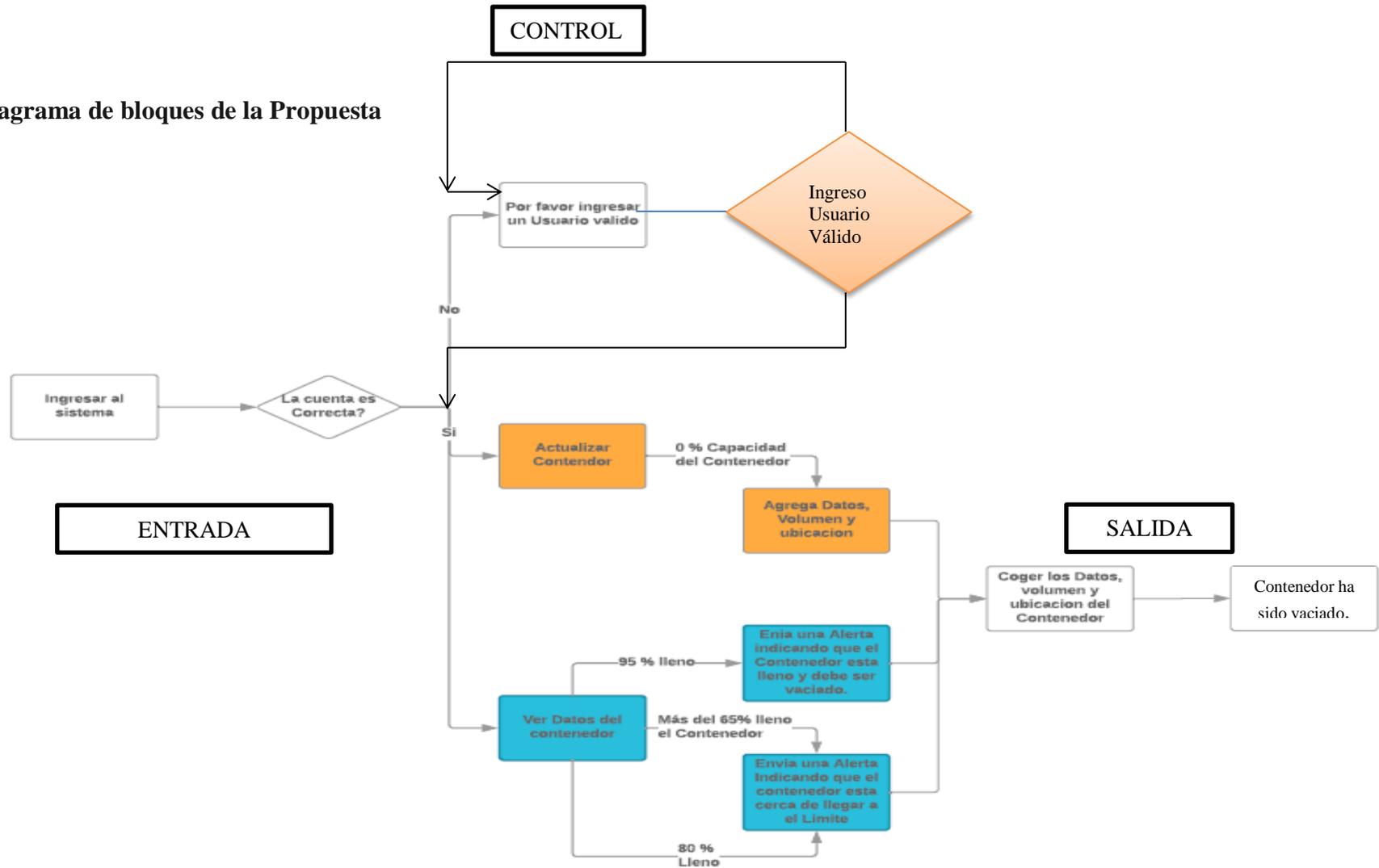


ZONAS	DESCRIPCIÓN	VEHÍCULOS	DÍAS	RECORRIDO DIARIO EN KM
20	EMASEO (PRIMER VIAJE) 17-01-38-REDONDEL DE LA ATAHUALPA-EL CALZADO – CARDENAL DE LA TORRES	MERCEDEZ BENZ #29	LUNES-MARTES-MIÉRCOLES-JUEVES VIERNES(MANTENIMIENTO)	70 KM
	(SEGUNDO VIAJE) 05-01-07 LAS CATILINARIAS--LA FLORIDA-NUEVA ESPERANZA-05-01-07-AV. INDOAMERICA-			
	VICTORIA-LA FERROVIARIA-SOLANDA 1- SOLANDA 2.			
21	EMASEO (PRIMER VIAJE) 16-01-36 AV. TENIENTE HUGO ORTIZ-MERCADO MAYORISTA-PARQUE ECOLOGICO--PUERTAS DEL SOL-URB. OSCUS-MIÑARICA I	MERCEDEZ BENZ # 29	LUNES-MIERCOLES-VIERNES- SABADOS	96 KM
	EMASEO (SEGUNDO VIAJE) 03-02-04 MIÑARICA II-LA ALBORADA-LA FLORESTA-HOSPITAL SAN BARTOLO		MARTES MANTENIMIENTO	
22	EMASEO (PRIMER VIAJE) 17-01-38 AV.ATAHUALPA-CDLAS SAN CA YETANO LA PRESIDENCIAL-LA PRADERA	MERCEDEZ BENZ # 29	LUNES-MIERCOLES-VIERNES- SABADOS	145 KM
	EMASEO (SEGUNDO VIAJE) 17-02-39 URBANIZACION - CONCEJO PROVINCIAL – AV. AJAVI		JUEVES MANTENIMIENTO	
28	EMASEO (PRIMER VIAJE) 02-02-09 –SAN BARTOLO-EL RECREO	MERCEDES BEMZ #25	LUNES-MIERCOLES-VIERNES- SABADOS	80 KM
	EMASEO (SEGUNDO VIAJE) 23-01-48 –TENIENTE HUGO ORTIZ-AV. MALDONADO			

Tabla. 2.2. Rutas de recolección de basura

Fuente: (Elaboración propia)

## 2.12.- Diagrama de bloques de la Propuesta



## USUARIOS FUNCIONALES

Figura 2.3. Esquema del Diagrama de Bloques de la Propuesta

Fuente: (Elaboración propia)

En la figura 2.3 se puede observar el esquema de la propuesta del diseño, el cual consta de las siguientes etapas que se detallan a continuación.

- 1. Entrada**
- 2. Control**
- 3. Salida**

#### **1. Etapa de Entrada.**

##### **a. Módulo GPS**

Este módulo a través del GPS recibe la información de los satélites que se conectan a cada momento que se encienda el dispositivo, esta información contiene las coordenadas de posiciones que son hora, longitud y latitud, este módulo está encargado de recibir y enviar la información almacenada al módulo de control.

#### **2. Dispositivo de Control**

Este módulo lo primero que hace es recibir los datos correspondientes al monitoreo del GPS, para luego transmitir información al micro controlador Atmega 2560, su función es controlar toda la información y los datos que envía el módulo GSM. A este módulo están interconectados todos los demás módulos que conforman este dispositivo de rastreo vehicular.

#### **3. Dispositivo de Salida**

##### **a.- Módulo GSM**

Consta del módulo SIM900 que va cumplir la función principal de enviar un SMS de texto al dispositivo de control, el mismo que se controla a través de comandos AT, este módulo esta interconectado a través de una comunicación serial 232 hacia el módulo de control el cual recibirá la información para después ser procesada.

##### **b.- Módulo alimentación de energía.**

Para proporcionar el voltaje y corriente requeridos se hará mediante la batería de litio que es de 5V.

Se realizará la interfaz gráfica para visualización de los parámetros en la aplicación Android, para las aplicaciones que se desarrollan habitualmente en el lenguaje Java sistema operativo.

Para la simulación del proyecto es necesaria tener una descripción mediante un esquema de la propuesta del diseño electrónico como se observa en la figura 2.4. Se muestra la comunicación que existe entre el hardware y el software, así de esa forma, se logra el envío y recepción de datos en forma bidireccional, los mismos que serán tomados para la ejecución de varios procesos, acciones y resultados.

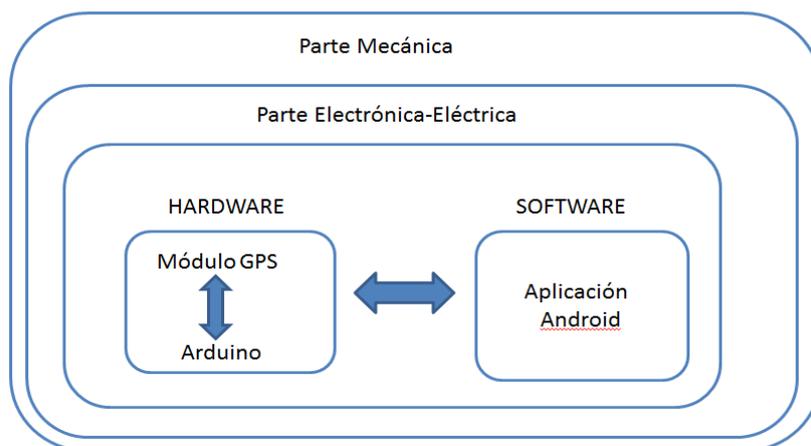


Figura. 2.4. Esquema de la propuesta del diseño electrónico

Fuente: (Elaboración Propia)

El diagrama de flujo se propone el siguiente proceso, como se muestra en la Figura 2.5.

1. Se enciente el dispositivo ubicado en el contenedor.
2. Se energiza los módulos
3. Mediante la comunicación GPRS existirá envío y recepción de datos.
4. El prototipo que se encuentra en el contenedor será manipulado por el aplicativo instalado en el Smartphone para realizar las distintas acciones.
5. Se envían los datos los cuales se mostrarán dentro del aplicativo.

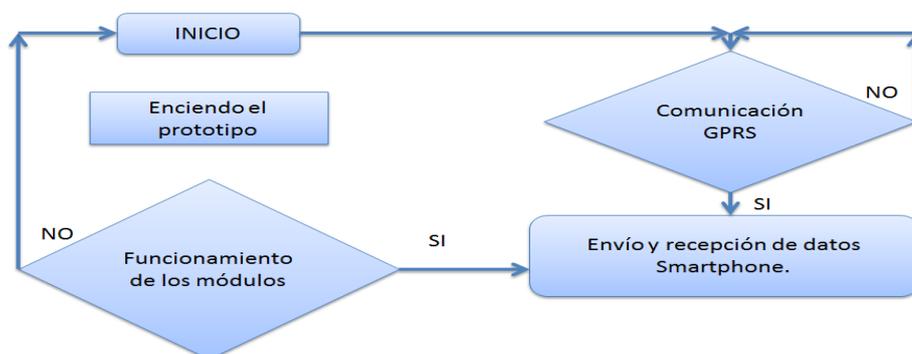


Figura. 2.5. Diagrama de Flujo Prototipo

Fuente: (Elaboración propia)

### Parte electrónica del dispositivo.

Es la etapa, la más significativa del sistema, ya que en base a esta etapa se trabajará todo el circuito y es el cerebro del prototipo.

La etapa de control es la encargada de tomar las decisiones de activar: la alarma, la comunicación con el servidor, la información tanto de latitud como de longitud del dispositivo de localización.

Viendo la necesidad de trabajar en software y hardware libre se investigó entre varios tipos de tecnologías existentes en el mercado nacional para ser utilizada como etapa de control, y con esto determinar cuál cubre con las especificaciones requeridas por el prototipo, con este fin se tomó en cuenta la opción de Arduino.

Dentro de la tecnología Arduino existen diversos diseños, la placa electrónica Arduino Mega, por la capacidad es la que mejor se acopla a las necesidades del prototipo.

En el diagrama de la figura 2.6, se muestra que valor debe ingresar a los puertos serial del Arduino donde se conectará con el módulo GSM, se muestran 4 procesos establecidos de conexión repetitiva durante un periodo de tiempo hasta establecer la localización mediante las coordenadas enviadas.

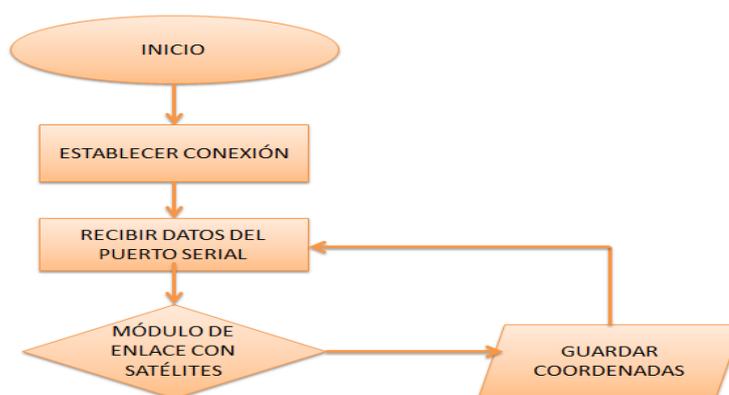


Figura. 2.6. Comunicación GPS

Fuente: (Elaboración propia)

A través del puerto serial 2, se da el paso de datos y conectividad con el módulo GSM/GPRS, definiendo la funcionalidad en el dispositivo electrónico, tomando las posiciones del dispositivo y sus coordenadas, se muestra un bosquejo del diagrama completo y su funcionalidad de conexión entre cada uno de los elementos que lo componen como se muestra en la fig. 2.7.

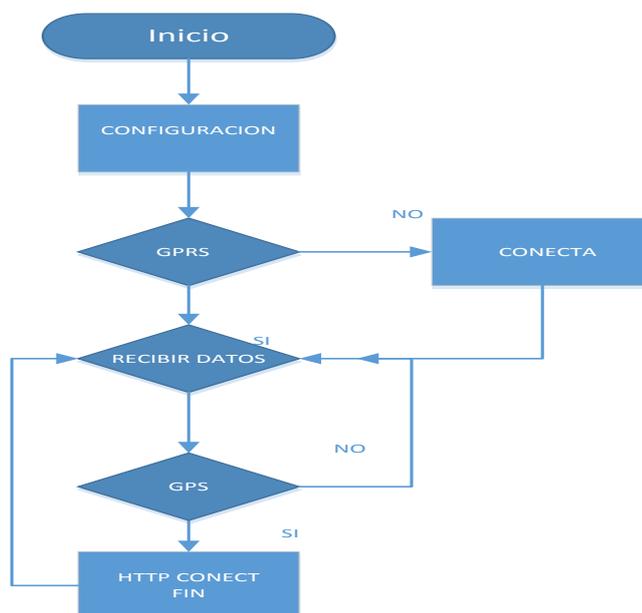


Figura. 2.7. Flujograma funcional del sistema

Fuente: (Elaboración propia)

### 2.13.- ELEMENTOS Y COSTOS DEL PROTOTIPO

Para realizar este proyecto serán necesarios los siguientes elementos y costos:

	<b>MATERIALES</b>	<b>CANTIDAD</b>	<b>COSTO UNITARIO</b>	<b>VALOR</b>
1	Arduino Mega 2560	1	\$ 25,00	\$ 25,00
2	Sensor de Distancia Ultrasónico HC-SR04	1	\$ 10,00	\$ 10,00
3	Módulo Gsm/Gprs Sim900	1	\$ 55,00	\$ 55,00
4	Módulo Gps Neo 6m Ublox	1	\$ 25,00	\$ 25,00
5	Sensor De Peso 5kg	1	\$ 15,00	\$ 15,00
6	Bateria Lipo 7.4 V 2 Celdas 4000mah	2	\$ 25,00	\$ 50,00
7	Diseño E Impresión De Prototipo De Contenedor			\$ 80,00
8	Mano De Obra	1 PERSONAS	100 HORAS	\$ 450,00
	<b>TOTAL</b>			\$ 710,00

Tabla. 2.3. Costos del Prototipo

Fuente: (Elaboración propia)

**2.14. VENTAJAS DEL PROTOTIPO:**

- Lograr un espacio sano, limpio, libre de desechos, bacterias o gérmenes, que sea agradable como paisaje y que se pueda habitar en él.
- Garantizar la adecuada recolección de los residuos en los contenedores cuando estén a un 80% de llenado, ya que emitirá un mensaje al teléfono y de esta manera se acercara el respectivo recolector de basura a vaciar los contenedores.
- Ayudará a minimizar la contaminación del medio ambiente.
- Beneficios ecológicos y sociales, ya que de esta manera todos los factores salen favorecidos, el medio ambiente porque se conserva; los individuos porque mejoran su entorno haciéndolo saludable y habitable.
- En cuanto al sector económico se implementará un sistema integrado al recopilar de una mejor manera la basura.

## 2.15. CRONOGRAMA DE ACTIVIDADES

### PROYECTO DE INVESTIGACIÓN “PROTOTIPO DE UN SISTEMA AUTOMÁTICO PARA EL CONTROL DE LLENADO DE DEPÓSITOS DE BASURA CON MONITOREO REMOTO EN LA CIUDAD DE QUITO UTILIZANDO UNA APLICACIÓN EN ANDROID”.

NOMBRE DE TAREA	TIEMPO DE DURACIÓN	COMIENZO						FINALIZACIÓN
		10-11 nov 2017	12-13 nov 2017	14-20 nov 2017	21-23 nov 2017	24 nov 2017 al 08 ene 2018	09 ene al 24 ene 2018	25 ene 11 feb 2018
INICIO	0 DÍAS							
ANALIZAR LOS ELEMENTOS	2 DÍAS	█						
ANALIZAR LOS TIPOS DE ARDUINO	2 DÍAS		█					
ALTERNATIVAS DE LENGUAJE DE PROGRAMACIÓN	7 DÍAS			█				
PRUEBAS DE BIDIRECCIONALIDAD	3 DÍAS				█			
<b>DISEÑO DEL SISTEMA DE LOCALIZACIÓN DEL DISPOSITIVO</b>	20 DÍAS					█		
SELECCIÓN DE LOS ELEMENTOS	4 DÍAS					█		
CONEXIÓN DEL DISPOSITIVO AL SMARTPHONE Y VICEVERSA	2 DÍAS					█		
INSTALACIÓN DE LOS ELEMENTOS	8 DÍAS					█		
IMPLEMENTACIÓN DEL PROTOTIPO	8 DÍAS					█		
PRUEBAS DE ENLACE DE LOS COMPONENTES	3 DÍAS					█		
<b>DISEÑO PARA LA APLICACIÓN SMARTPHONE</b>	25 días						█	
ELECCIÓN DE LA PLATAFORMA	3 DÍAS						█	
IMPLEMENTACIÓN DE LA APLICACIÓN EN PROGRAMACIÓN JAVA	15 DÍAS						█	
PRUEBAS DE CONECTIVIDAD	2 DÍAS						█	
<b>PRUEBAS Y RESULTADOS DE FUNCIONAMIENTO</b>	5 DÍAS							█
ESCENARIO PARA LAS PRUEBAS DE FUNCIONAMIENTO	3 DÍAS							█
PRUEBAS DE VISUALIZACIÓN	3 DÍAS							█
PRUEBAS DE CAMPO	3 DÍAS							█
ANÁLISIS DE RESULTADOS	3 DÍAS							█
FINALIZACIÓN								█

Tabla. 2.4. Cronograma de Actividades

Fuente: (Elaboración propia)

## **CAPITULO III**

### **3.- IMPLEMENTACIÓN**

#### **3.1 DESARROLLO**

La presente propuesta se basa en un sistema de reciclaje, implementando los códigos en el lenguaje PHP y JS, indicando el manual de usuario para su posterior implementación en el lugar de estudio, el compromiso de esta fase es ofrecer una solución viable al problema de la contaminación al medio ambiente, reduciendo los costos operativos, el tiempo de ejecución y aumentando la eficiencia laboral de EMASEO.

La metodología empleada es el Modelo de Cascada, la cual consiste es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada.

#### **3.2 IMPLEMENTACIÓN**

El modelo de desarrollo en cascada sigue una serie de etapas de forma sucesiva, la etapa siguiente empieza cuando termina la etapa anterior.

**LAS FASES QUE COMPONEN EL MODELO SON LAS SIGUIENTES:**

##### **3.2.1 REQUISITOS DEL SOFTWARE**

En esta fase se hace un análisis de las necesidades del cliente para determinar las características del software a desarrollar, y se especifica todo lo que debe hacer el sistema sin entrar en detalles técnicos. Hay que ser especialmente cuidadoso en esta primera fase, ya

que en este modelo no se pueden añadir nuevos requisitos en mitad del proceso de desarrollo.

Por lo tanto, esta es la etapa en la que se lleva a cabo una descripción de los requisitos del software, y se acuerda entre el cliente y la empresa desarrolladora lo que el producto deberá hacer. Disponer de una especificación de los requisitos permite estimar de forma rigurosa las necesidades del software antes de su diseño. Además, permite tener una base a partir de la cual estimar el costo del producto, los riesgos y los plazos.

En el documento en el que se especifican los requisitos, se establece una lista de los requerimientos acordados. Los desarrolladores deben comprender de forma clara el producto que van a desarrollar. Esto se consigue teniendo una lista detallada de los requisitos, y con una comunicación fluida con el cliente hasta que termine el tiempo de desarrollo.

### **3.2.2 DISEÑO**

En esta etapa se describe la estructura interna del software, y las relaciones entre las entidades que lo componen.

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

En esta fase se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior. La programación es el proceso que lleva de la formulación de un problema de computación, a un programa que se ejecute produciendo los

pasos necesarios para resolver dicho problema.

Al programar, tenemos que realizar actividades como el análisis de las condiciones, la creación de algoritmos, y la implementación de éstos en un lenguaje de programación específico.

### 3.2.3. MONTAJE DEL PROYECTO

#### MONTAJE DEL CIRCUITO EN PROTOBOARD

Se inicia el montaje del dispositivo electrónico que se va a implementar, previamente se desarrolló la simulación y luego se procede a conectar los elementos con el cableado respectivo en el protoboard, partimos del diagrama indicado en el circuito de la figura 3.1.

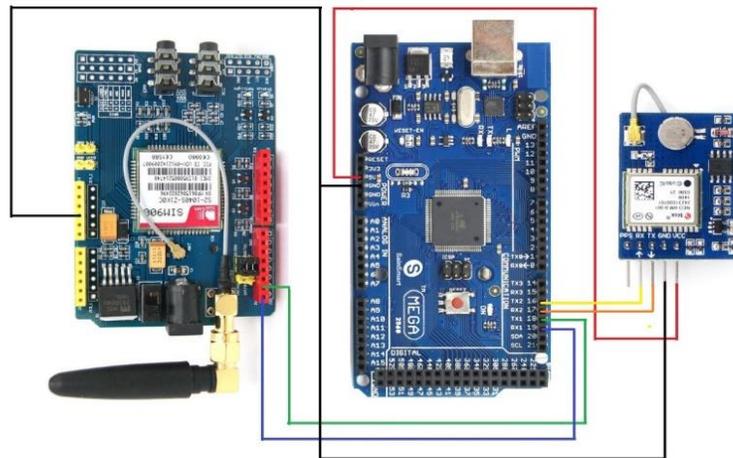


Figura. 3.1. Diagrama de conexiones

Fuente: (Elaboración Propia)

Mientras que la conexión para el Arduino es la misma que para el *Prototipo de Control y Monitoreo*, en la figura 3.2. de detalla tanto la polarización del sensor a 5V y GND como los pines digitales a los que va a estar conectado.

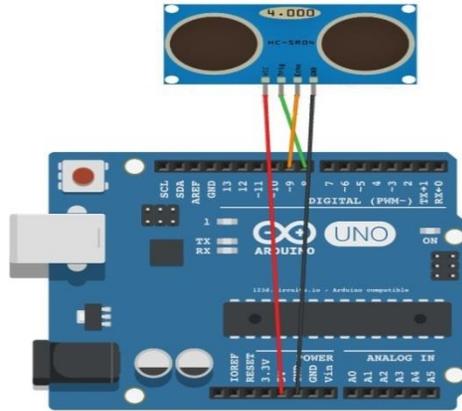


Figura. 3.2. Conexión del sensor de distancia HC-SR04 y Arduino

Fuente: (Elaboración Propia)

En la figura 3.3 (a) y (b), se indica la conexión realizada entre el módulo GPS y el módulo GPRS-GSM-SIM 900, y la inserción del chip de movistar en el módulo GSM.



(a)



(b)

Figura. 3.3. Conexión entre módulos GPS - GPRS GSM

Fuente: (Elaboración Propia)

Luego se procedió a conectar el módulo GPRS-GSM con el Arduino Mega, las conexiones se realizaron en un protoboard considerando todas seguridades del caso, para que no exista ningún cortocircuito y de tal manera algún daño de los elementos electrónicos, todas las conexiones se encuentran en el esquema de la figura 3.4.

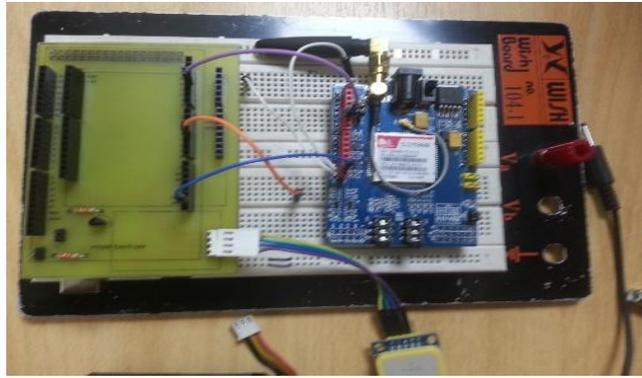


Figura. 3.4. Conexiones módulo GPRS GSM –ARDUINO MEGA

Fuente: (Elaboración Propia)

Para conectar al Arduino se tienen dos opciones y para ambas se necesitó 2 fuentes de alimentación externa, ya que con conectar el Arduino por USB será suficiente para encender el módulo, pero no para alimentar también la tarjeta. Si se acopla la alimentación externa al shield GPRS y se alimenta el Arduino con el USB o una fuente aparte, sólo se tendrá que unir los pines 7 y 8 para la comunicación serie y el GND entre ambas tarjetas, como se muestra en la Figura 3.5.

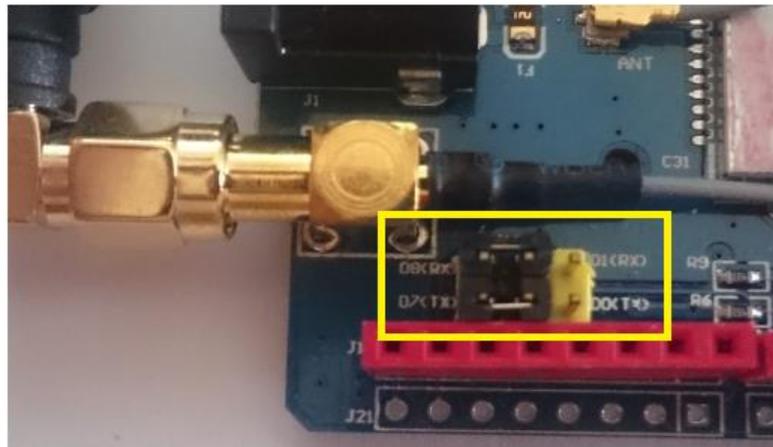


Figura. 3.5. Pines de conexión GPRS

Fuente: (Elaboración Propia)

Para encender el GPRS se tendrá que colocar el interruptor en la posición correcta, las posiciones corresponden a cada uno de los tipos de conexión.

Una vez hecho se encenderán dos LEDs, para activar la alimentación de la tarjeta SIM podemos hacerlo manualmente presionando el pulsador que hay en uno de los laterales durante uno o dos segundos y esto se notará que se enciende otro LED, y que uno de los

que se había encendido antes empieza a parpadear una vez por segundo.

Este parpadeo indica que está buscando la red, cuando la encuentre y se conecte cambiará la frecuencia de parpadeo y lo hará cada 3 o 4 segundos, como se muestra en la Figura 3.6.

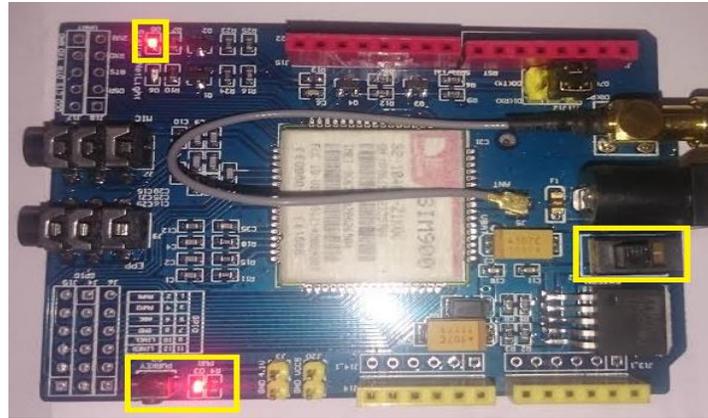


Figura. 3.6. Configuración del módulo GPRS-GSM SIM 900

Fuente: (Elaboración Propia)

La conexión final se indica en la Figura 3.7, donde la celda de carga se conecta al módulo HX711 y esta a su vez al Arduino Mega.

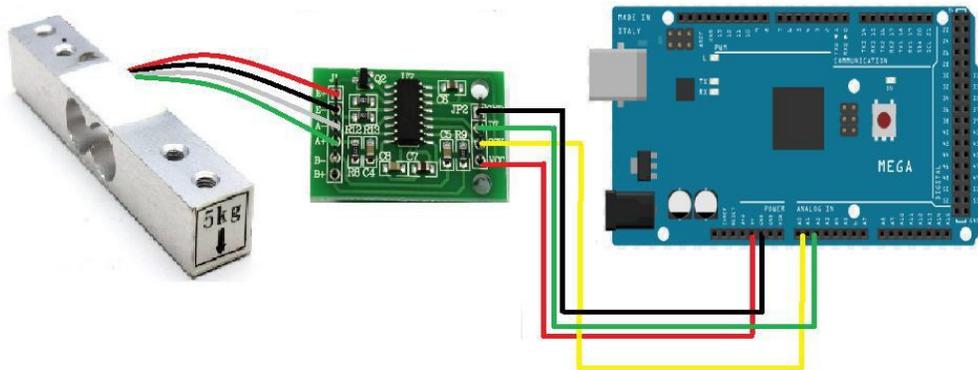


Figura. 3.7. Conexión entre la celda de carga, HX711 y Arduino Mega

Fuente: (Elaboración Propia)

En la siguiente Figura 3.8, se muestra la conectividad del dispositivo.

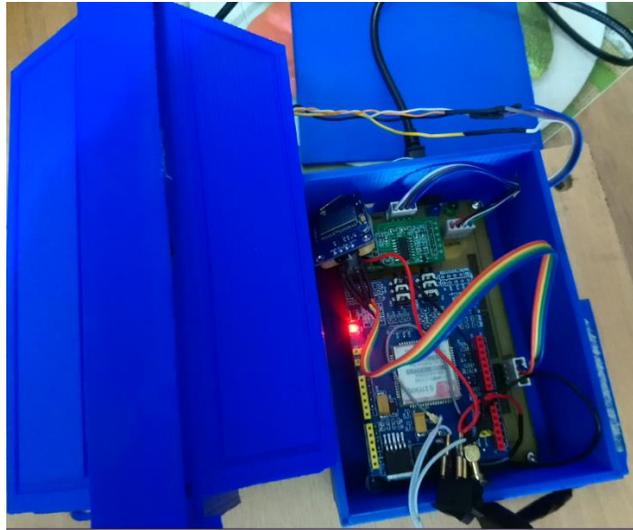


Figura. 3.8. Conectividad del dispositivo.

Fuente: (Elaboración Propia)

Se carga el software implementado en el Arduino Mega y se verifica la conectividad entre los elementos en la figura 3.9, se indica el programa y el proceso de carga del programa en el microcontrolador.

### **MONTAJE DEL SOFTWARE**

Para el desarrollo de la aplicación Android, se ha estudiado una herramienta muy versátil en el mercado como lo es Android Studio, siendo un potente editor de códigos que aumenta la productividad durante su compilación.

Estructura del Proyecto en Android Studio.

Cada proyecto en Android Studio contiene uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos se incluyen los siguientes:

- Módulos de apps para Android
- Módulos de bibliotecas
- Módulos de Google App Engine

De forma predeterminada, en Android Studio se muestran los archivos de tu proyecto en la vista de proyectos de Android.

Esta vista está organizada en módulos para que puedas acceder rápidamente a los archivos de origen claves de tu proyecto.

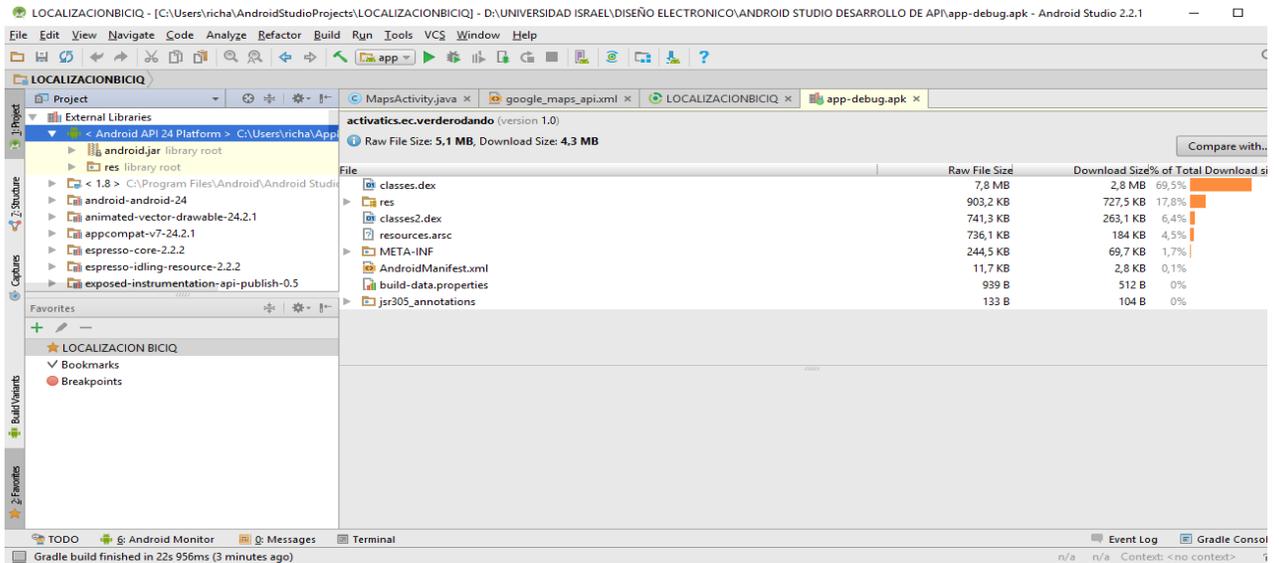


Figura. 3.9. Archivos del proyecto en la vista de Android.

Fuente: (Elaboración Propia)

## INTERFAZ DEL USUARIO

La ventana principal de Android Studio consta de varias áreas lógicas que se identifican en la figura 3.10.

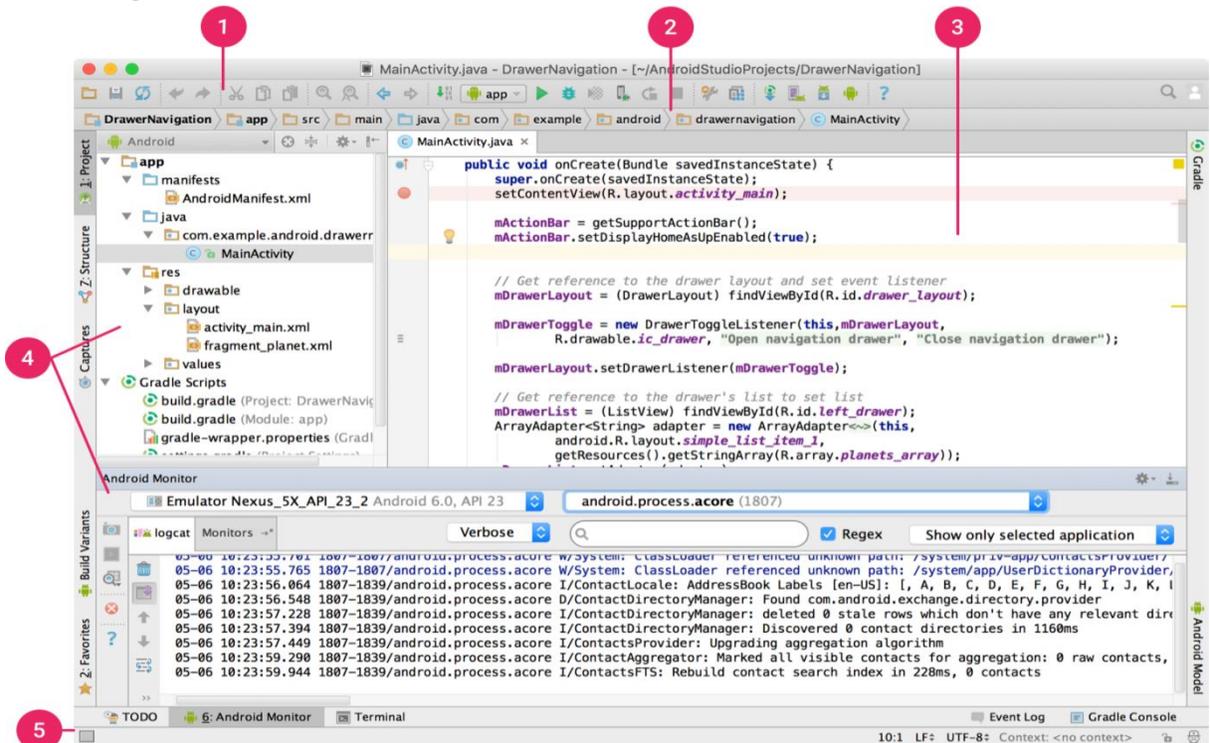


Figura. 3.10. Ventana principal de Android Studio.

Fuente: (developer.android.com)

La **barra de herramientas** te permite realizar una gran variedad de acciones, como la ejecución de tu app y el inicio de herramientas de Android.

La **barra de navegación** te ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana **Project**.

La **ventana del editor** es el área en la que puedes crear y modificar código. Según el tipo de archivo actual, el editor puede cambiar. Al visualizar un archivo de diseño, por ejemplo, el editor muestra el Editor de diseño.

Las **ventanas de herramientas** te permiten acceder a tareas específicas, como la administración de proyectos, la búsqueda y los controles de versión, entre otras. Puedes expandirlas y contraerlas.

En la **barra de estado** se muestra el estado de tu proyecto y el IDE, además de advertencias o mensajes.

Puedes organizar la ventana principal para tener más espacio en pantalla ocultando o desplazando barras y ventanas de herramientas. También puedes usar combinaciones de teclas para acceder a la mayoría de las funciones del IDE.

En cualquier momento, puedes realizar búsquedas en tu código fuente, bases de datos, acciones, elementos de la interfaz de usuario, etc., presionando dos veces la tecla Shift o haciendo clic en la lupa que se encuentra en la esquina superior derecha de la ventana de Android Studio. Esto puede ser muy útil, por ejemplo, si intentas localizar una acción específica del IDE que olvidaste cómo activar.

### 3.2.4 ANDROID STUDIO.

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA .



Figura. 3.11. Plataforma Android Studio

Fuente: (developer.android.com)



Figura. 3.12. Entorno de desarrollo Android Studio

Fuente: (developer.android.com)

### 3.2.5 VENTANAS DE HERRAMIENTAS

En lugar de usar perspectivas preestablecidas, Android Studio sigue el contexto personalizado y ofrece automáticamente ventanas de herramientas relevantes mientras realiza los trabajos. De forma predeterminada, las ventanas de herramientas usadas con mayor frecuencia se fijan en la barra de ventanas de herramientas en los bordes de la ventana de la aplicación.

Para expandir o contraer una ventana de herramientas, se debe hacer clic en el nombre de la herramienta en la barra de la ventana de herramientas. También se puede arrastrar, anclar, desanclar, adjuntar y ocultar ventanas de herramientas.

Para volver al diseño predeterminado actual de ventanas de herramientas, se debe hacer clic en **Window > Restore Default Layout** o personaliza el diseño predeterminado haciendo clic en **Window > Store Current Layout as Default**.

Para mostrar u ocultar la barra de ventanas de herramientas completa, Clic en el ícono de ventana en la esquina inferior izquierda de la ventana de Android Studio.

Para localizar una ventana de herramientas específica, posicionar el puntero sobre el ícono

de ventana y selecciona la ventana de herramientas en el menú.

También se puede usar combinaciones de teclas para abrir ventanas de herramientas. En la Tabla 3.1. Se muestran las combinaciones de teclas para las ventanas más comunes.

Ventana de herramienta	Windows y Linux	Mac
Proyecto	<b>Alt+1</b>	<b>Comando+1</b>
Control de versión	<b>Alt+9</b>	<b>Comando+9</b>
Ejecutar	<b>Shift+F10</b>	<b>Control+R</b>
Depurar	<b>Shift+F9</b>	<b>Control+D</b>
Android Monitor	<b>Alt+6</b>	<b>Comando+6</b>
Volver al editor	<b>Esc</b>	<b>Esc</b>
Ocultar todas las ventanas de herramientas	<b>Control+Shift+F12</b>	<b>Comando+Shift+F12</b>

Tabla. 3.1. Combinaciones de teclas para algunas ventanas de herramientas útiles.

Fuente: (Elaboración Propia)

### 3.2.6.- DISEÑO DEL DISPOSITIVO

En la figura 3.13 se observa el diseño de la carcasa en 3D que se realizó en el software solidworks, esta carcasa se utiliza para proteger los elementos del circuito final y así también tener una presentación excelente del prototipo al cliente.



Figura. 3.13. Diseño de la carcasa

Fuente: (Elaboración propia)

### 3.2.7.- Plataforma Arduino

La Figura 3.14 nos indica la iniciación y carga de datos del programa Arduino Uno.

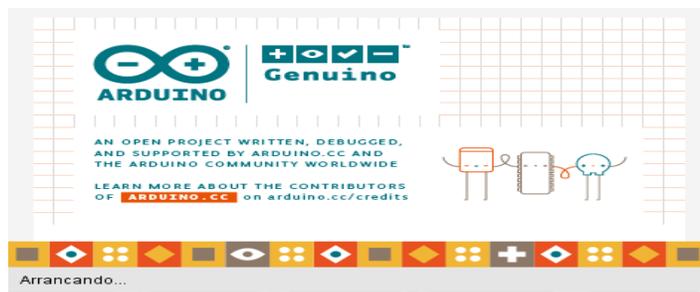


Figura. 3.14. Plataforma Arduino.

Fuente: (Arduino,2017)

#### 3.2.7.1.- PROGRAMACIÓN PLATAFORMA ARDUINO.

Para diseñar el sistema de localización se utilizará los dispositivos y tecnologías disponibles en la actualidad y en el mercado nacional.

En la figura 3.15, se muestra la programación utilizada en la plataforma de Arduino, dentro de los anexos se extiende la programación de la aplicación a detalle.

```

recolectorgprs Arduino 1.6.7
Archivo Editar Programa Herramientas Ayuda
recolectorgprs
#include <TinyGPS++.h>
#define BUFFSIZE 1024
#include "HX711.h"

HX711 scale(A0, A1);

char buffer1[BUFFSIZE];
int bufferSize = 0;
volatile byte chr;
float peso;
int enviar=0;
int dato;
int gprsconnect=1;
int postcol=0;
int httperror=0;
int gprserror=0;
static const uint32_t GPSBaud = 9600;

// The TinyGPS++ object
TinyGPSPlus gps;

int alerta=0;
int led0K=9;
int led2=10;
int buzzer=7;
int trigFin = 11;
int echoFin = 12;
long duration=0;

```

Figura. 3.15. Plataforma con el lenguaje de programación.

Fuente: (Elaboración Propia)

### 3.2.7.2 COMPILACIÓN DEL PROGRAMA.

En la figura 3.16, muestra que dentro del programa se selecciona la opción compilar para revisar si tiene errores.

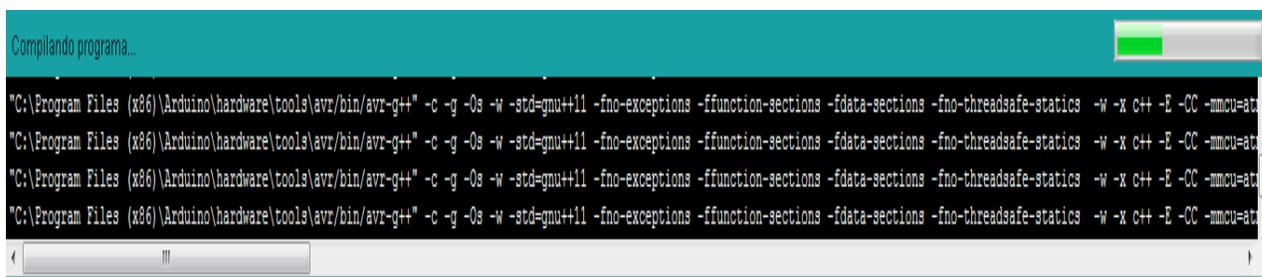


Figura. 3.16. Compilación de programas.

Fuente: (Elaboración Propia)

### 3.2.7.3. ERRORES DE COMPILACIÓN.

En la figura 3.17, ya compilado el programa se revisa los errores, se corrige manualmente revisando la lógica de programación para saber que error existe.



Figura. 3.17. Error de compilación.

Fuente: (Elaboración Propia)

## 3.3 PRUEBAS DE FUNCIONAMIENTO

### 3.3.1. LISTA DE VERIFICACIÓN DE PRUEBAS.

Se realizaron las respectivas pruebas de funcionamientos, que se muestra a continuación, en la tabla 3.2, para esto se fue revisando uno por uno los elementos del dispositivo, con el fin de evitar fallas posteriores.

ITEM	ELEMENTOS	FUNCIONALIDAD	
		SI	NO
1	Módulo GPRS-GSM SIM 900	X	
2	Módulo GPS NEO-6M-0-001	X	
3	Módulo Arduino Mega 250	X	
4	Led	X	
5	Pulsador	X	
6	Aplicación Android	X	
7	Módulo HX-711	X	

Tabla. 3.2. Lista de verificación de pruebas de los elementos del prototipo

Fuente: (Elaboración Propia)

### PRUEBAS DE FUNCIONAMIENTO DE ELEMENTOS

No	Prueba (Eco-Basureros)	Pb 1	Pb 2	Pb 3	Pb 4	Pb 5	Pb 6	Pb 7	Pb 8	Pb 9	Pb 10	Porcentaje
1	¿Conecta desconecta GPS-GPRS SIM 900?	√	√	√	√	√	√	√	√	√	√	100
2	¿Sensor distancia?	√	√	√	√	√	√	√	√	√	√	100
3	¿Sensor peso?	√	√	√	√	√	√	√	√	√	√	100
4	¿Notificación?	√	√	√	√	√	√	√	√	√	√	100
5	¿Mensaje texto?	√	√	√	√	√	√	√	√	√	√	100
6	¿Batería?	√	√	√	√	√	√	√	√	√	√	100
7	¿Modulo HX-711?	√	√	√	√	√	√	√	√	√	√	100
8	¿Modulo GPS?	√	√	√	√	√	√	√	√	√	√	100

Tabla. 3.3. Prueba de Funcionamiento de Elementos

Fuente: (Elaboración Propia)

### 3.4 ANÁLISIS DE RESULTADOS

Una vez que se realizaron las pruebas de funcionamiento en los contenedores de basura del Mercado Mayorista de la ciudad de Quito, se validó que el dispositivo funciona perfectamente a un 100%, de esta manera se concluye que nuestro proyecto es viable ya que los usuarios quedaron satisfechos con esta nueva implementación, se redujo el olor y el ahorro de tiempo para los recolectores de basura, según las condiciones que arrojó el diseño para la colocación de los sensores.

PRUEBA DE FUNCIONAMIENTO	SI	NO
Se visualiza el contenedor mediante la aplicación web tanto las latitudes como las longitudes y su funcionamiento	X	
Se visualiza la información en la web de los datos del contenedor	X	
Se válida el envío del mensaje a la aplicación del despachador y viceversa del operador cuando el contenedor es desalojado del material y residuos sólidos.	X	

Tabla. 3.4. Pruebas en el contenedor de basura

Fuente: (Elaboración Propia)

## CONCLUSIONES

- Se examinó el diseño del Prototipo de control y monitoreo, que permite obtener la información de una manera eficaz y exacta, dada la ubicación estratégica de los sensores de peso y distancia en el ecobasurero y de la misma manera evitando que estos puedan sufrir algún tipo de daño debido a las adversidades atmosféricas.
- Se determinó y desarrolló una programación para Smartphone, que facilita y sirve de manera intuitiva al operario analizar los datos de peso y nivel, donde se visualizan en los indicadores, además de mantener seguridad en la información al poder tener un historial de llenado del ecobasurero.
- Se diseñó un aplicativo Web para el monitoreo y control de los diferentes tachos de basura para un manejo eficiente de los Ecobasureros en la ciudad de Quito
- Se implementó una base de datos para un almacenamiento histórico de la información de los basureros para su posterior utilización en aplicativo Web, así como estadísticas.
- Se validó que la conexión del GPRS fue exitosa.

## RECOMENDACIONES

- Se recomienda a los técnicos, que se mantenga una base de datos tangible de la información que genera el Ecobasurero ubicado en el Mercado Mayorista, de tal manera que en un futuro EMASEO pueda tener un control más eficiente de los vehículos de carga lateral al momento de realizar sus rutas de recolección de basura.
- Se sugiere el uso de baterías recargables es indispensable para así no incurrir en gastos innecesarios al adquirir baterías alcalinas cada vez que se agote su autonomía. Las cuales se cargarán mediante un inversor de energía.
- Se recomienda entrenamiento de manejo y funcionalidad del dispositivo, al personal de EMASEO, encargado de la estación remota, que estos subprogramas deben encontrarse sincronizados en cuanto a tiempos de trabajos para evitar un desfase en la notificación que se recibe.
- Realizar pruebas técnicas en un futuro para comprobar el funcionamiento del prototipo y seguir mejorando el diseño.
- Se sugiere que el Módulo GPS siempre se conecte en lugares abiertos.

## BIBLIOGRAFÍA

- (REYES C., 2008: 156)
- (TOMASI W., 2003: 35)
- (FAICÁN R., 2015: 56)
- (TORRENTE A., 2010: 80)
- (HUERTA E., 2005: 35)
- <https://es.aliexpress.com/item/free-shipping-1pc-ublox-neo6mv2-gps-module-aircraft-flight-controller-for-arduino-new/1092839175.html>
- ANTENA, G. (2012). AT-65.
- <https://www1.elfa.se/data1/wwwroot/assets/datasheets/07840051.PDF>
- ARDUINO. (2017). ARDUINO UNO. <http://www.arduino.cc/en/main/arduinomega2560>
- COMANDOSAT. (2005). <http://alarmagsm.google.com/fils/comandos%20at.doc>
- CURSO DE MICROCONTROLADORES. (MARZO DE 2008). <http://cursodemicro.blogspot.com/>
- CURSOSMICRO. (2009). BASCOM AV R. <http://www.cursomicros.com/avr/compiladores/tutorial-de-bascom-avr.html>
- MAQUINAS INTELIGENTES SICOM .EE. <http://simcom.ee/?gsm-gprs>
- TIENDAS VIRTUALES DE VENTA Y CUROS DE ARDUINO. <http://www.geekfactory.mx/tutoriales/tutoriales-arduino/diagrama-de-pines-arduino/>
- INGENIERA Y DISEÑO ELECTRÓNICO. <https://electronilab.co/tienda/modulo-gps-ublox-neo-6m-v2-con-memoria-eprom/>
- MEDIAWIKI. [http://www.tuxti.com.br/wiki/index.php/arduino\\_-\\_m%c3%b3dulo\\_gsm\\_gprs\\_sim900](http://www.tuxti.com.br/wiki/index.php/arduino_-_m%c3%b3dulo_gsm_gprs_sim900)
- COBAC, UNIDADES DE ESTUDIO EN REDES Y TELECOMUNICACIONES. <http://tecnoredes.mx.tripod.com/page33.html>

# ANEXOS

## ANEXO 1.

### PROGRAMACIÓN EN ARDUINO.

```

#include <TinyGPS++.h>
#define BUFSIZE 1024
#include "HX711.h"

HX711 scale(A0, A1);

char buffer1[BUFSIZE];
int buffSize = 0;
volatile byte chr;
float peso;
int enviar=0;
int dato;
int gprsconnect=1;
int postok=0;
int httperror=0;
int gprerror=0;
static const uint32_t GPSBaud = 9600;

// The TinyGPS++ object
TinyGPSPlus gps;

int alerta=0;
int ledOK=9;
int led2=10;
int buzer=7;
int trigPin = 11;
int echoPin = 12;
long duration=0;
int distance=0;

void setup() {
  Serial.begin(19200);
  Serial.println("Inicio");
  Serial1.begin(19200);
  Serial2.begin(GPSBaud);
  delay(2000);
  pinMode(ledOK, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(buzer, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

```

```

detectar();

if(gprserror==1){
  Serial.println("POR FAVOR REVISE EL MODULO");
  Serial.println("Y REINICIE EL ARDUINO");
  digitalWrite(ledOK, HIGH);
}else{
  parpadeo(50);
}

while(gprsconnect==0){
  Netconnect();
}

scale.set_scale(270.8);
scale.tare();

}

void loop() {
  if (Serial.available()) {
    Serial1.write(Serial.read());
  }
  if (Serial1.available()) {
    Serial.write(Serial1.read());
  }
  // if (Serial2.available()) {
  //Serial.write(Serial2.read());
  //}

  while (Serial2.available() > 0)
    if (gps.encode(Serial2.read()))
      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
    while(true);
  }

  peso=scale.get_units(1); // LECTURA DEL PESO

```

```

if (peso<1){peso=0;}
Serial.println(peso,0);
distancia();

if(distance<6 && enviar==0){
  enviar=1;
}
if(peso>300 && enviar==0){
  enviar=1;
}

if(distance>6 && peso<100 && enviar==1){
  enviar=0;
}

}

void distancia(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;
  Serial.print("Distance: ");
  Serial.println(distance);
}

// Limpia el buffer de datos
void clearBuffer() {
  for(int i=0;i<BUFSIZE;i++) {
    buffer1[i] = 0;
  }
  buffSize = 0;
}

//Agrega un caracter al bufer para comparar luego
void appendToBuffer(char c) {
  if(buffSize<BUFSIZE) {
    buffer1[buffSize++] = c;
  } else {
    Serial.print(buffer1);

```

```

    clearBuffer();
    buffer1[bufferSize++] = c;
}
}

// Espera por una cadena de texto sobre el serial
bool waitForString(char* string1, long waitForMilliseconds) {
    clearBuffer();
    long startTime = millis();
    boolean found = false;
    Serial.print("Waiting for string: ");
    Serial.print(string1);
    Serial.print("\r\n");
    while((millis()-startTime)<waitForMilliseconds) {
        if (Serial1.available()) {
            chr = Serial1.read();
            Serial.write((char)chr);
            addToBuffer((char)chr);
            if((char)chr=="\n") {
                if(strstr(buffer1,string1)!=NULL) {
                    found = true;
                    break;
                } else if(strstr(buffer1,"ERROR")!=NULL) {
                    found = false;
                    break;
                }
            }
            clearBuffer();
        }
    }
    if (Serial1.available())
        Serial1.write(Serial.read());
}
return found;
}

//Espera cualquier tipo de datos
void waitForData(int wait_delay) {
    //Espera por datos en el puerto serial por
    // el tiempo especificado
    long startTime = millis();
    while((millis()-startTime)<wait_delay) {
        if (Serial1.available())
            Serial.write(Serial1.read());
        if (Serial1.available()) {
            Serial1.write(Serial.read());
        }
    }
}

```

```

}
}

```

```

void Netconnect(){
  Serial1.println("AT");
  if(!waitForString("OK",2000)) {
    Serial.println("MODULO DESCONECTADO O DESACTIVADO");
    gprerror=1;
    return;
  }
  Serial1.println("AT");
  if(!waitForString("OK",2000)) {
    Serial.println("NADA");
    gprerror=1;
    return;
  }
  Serial1.println("AT+SAPBR=3,1,\"Contype\",\"GPRS\"");
  if(!waitForString("OK",2000)) {
    Serial.println("NADA");
    gprerror=2;
    return;
  }
  Serial1.println("AT+SAPBR=3,1,\"APN\",\"internet.movistar.com.ec\"");
  if(!waitForString("OK",2000)) {
    Serial.println("NADA");
    gprerror=3;
    return;
  }
  if(!waitForString("OK",20000)) {
    Serial.println("NADA");
    gprerror=4;
    return;
  }
  Serial1.println("AT+SAPBR=2,1");
  if(!waitForString("OK",2000)) {
    Serial.println("NADA");
    gprerror=5;
    return;
  }
  gprsconnect=1;
}

```

```

void Serverconnect(){
  Serial1.println("AT+HTTPINIT");
  if(!waitForString("OK",2000)) {
    Serial.println("NADA");
  }
}

```

```

    httperror=1;
    return;
}

```

```

Serial1.println("AT+HTTTPARA=\"CID\",1");
if(!waitForString("OK",2000)) {
    Serial.println("NADA");
    httperror=2;
    return;
}

```

```

Serial1.print("AT+HTTTPARA=\"URL\", \"ecobasureros.ec/locations?lat=");
Serial1.print(gps.location.lat(), 6);
//Serial1.print("-0.196932");
Serial1.print("&lng=");
Serial1.print(gps.location.lng(), 6);
//Serial1.print("78.498222");
Serial1.print("&id=");
Serial1.print("reco001");
Serial1.print("&alerta");
Serial.print("&");
Serial.println(distance);
Serial.print("&");
Serial.print(peso);
Serial1.println("");
if(!waitForString("OK",20000)) {
    Serial.println("NADA");
    httperror=3;
    return;
}

```

```

Serial1.println("AT+HTTPACTION=1");
if(!waitForString("+HTTPACTION:1,200",20000)) {
    Serial.println("NADA");
    httperror=4;
}
Serial1.println("AT+HTTPTERM");
if(!waitForString("OK",20000)) {
    Serial.println("NADA");
    httperror=5;
    return;
}

```

```

httperror=0;
postok=1;

```

```
}

```

```
void displayInfo()
{
  //Serverconect();
  Serial.print(F("Location: "));
  if (gps.location.isValid())
  {
    Serial.print(gps.location.lat(), 6);
    Serial.println(gps.location.lng(), 6);
    if (enviar==1){
      Serverconect();
    }
    if(httperror>0){
      ManejoErrorHttp();
      Serial.println("error http");
      digitalWrite(ledOK, HIGH);
    }else{
      parpadeo(500);
    }
    delay(1000);
  }
  else
  {
    Serial.print(F("INVALID"));
  }

  Serial.println();
}

```

```
void detectar(){
  Serial1.println("AT");
  if(!waitForString("OK",2000)) {
    Serial.println("MODULO DESCONECTADO O DESACTIVADO");
    gprsconnect=1;
    postok=1;
    gprserror=1;
    return;
  }

```

```
Serial1.println("AT+SAPBR =2,1");
if(waitForString("0.0.0.0",2000)) {

```

```

Serial.println("CONECTANDO A LA RED");
gprsconnect=0;
delay(1000);
}
Serial1.println("AT+HTTPTERM");
if(!waitForString("OK",2000)) {
  Serial.println("NADA");

}
Serial1.println("ATZ");
if(!waitForString("OK",2000)) {
  Serial.println("NADA");
  gprserror=1;
}
}

```

```

void ManejoErrorHttp(){

```

```

  Serial1.println("AT+HTTPTERM");
  if(!waitForString("OK",2000)) {
    Serial.println("NADA");
  }

```

```

  httperror=0;
}

```

```

void parpadeo(int vel){
  for(int x=0;x<5;x++){
    digitalWrite(ledOK, HIGH);
    delay(vel);
    digitalWrite(ledOK, LOW);
    delay(vel);
  }
}

```

```

void mensaje_sms1()

```

```

//Funcion para mandar mensaje de texto

```

```

{

```

```

  Serial1.print("AT+CMGF=1\r");

```

```

// AT command to send SMS

```

```

  message

```

```

  waitForData(500);

```

```

  Serial1.print("AT+CMGS=\"0989313096");

```

```

// recipient's mobile number, in

```

```

  international forma

```

```

  Serial1.println("");

```

```
waitForData(500);
Serial1.print("ALERTA "); // message to send
Serial1.print(gps.location.lat(), 6);
Serial1.print("LONG ");
Serial1.print(gps.location.lng(), 6);
Serial1.println(" Bici Nro: PBT-3886");
waitForData(500);
Serial1.println((char)26); // End AT command with a ^Z, ASCII
code 26 //Comando de finalizacion
waitForData(500);
Serial1.println();
waitForData(500); // Tiempo para que se envie el mensaje
Serial.println("SMS enviado");

}
```

## ANEXO 2.

### CODIGO SQL

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-----
-- Schema contenedordb
-----
DROP SCHEMA IF EXISTS `contenedordb` ;

-----
-- Schema contenedordb
-----
CREATE SCHEMA IF NOT EXISTS `contenedordb` DEFAULT CHARACTER SET utf8 ;
USE `contenedordb` ;

-----
-- Table `contenedordb`.`contenedor`
-----
DROP TABLE IF EXISTS `contenedordb`.`contenedor` ;

CREATE TABLE IF NOT EXISTS `contenedordb`.`contenedor` (
  `idcontenedor` INT NOT NULL AUTO_INCREMENT,
  `peso` DOUBLE NOT NULL,
  `distancia` DOUBLE NOT NULL,
  `longitud` DOUBLE NOT NULL,
  `latitud` DOUBLE NOT NULL,
  PRIMARY KEY (`idcontenedor`))
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

### Codigo Android .java

```
package com.example.david.controlcontenedoresdebasura;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
```

```
public class Principal extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_principal);
    }

    public void Contenedo1(View view){
        //Toast.makeText(this, "Contenedor1", Toast.LENGTH_SHORT).show();
        Intent con1 = new Intent(this, Contenedor_1.class);
        startActivity(con1);
    }

    public void Contenedo2(View view){
        //Toast.makeText(this, "Contenedor2", Toast.LENGTH_SHORT).show();
        Intent con2 = new Intent(this, Contenedor_2.class);
        startActivity(con2);
    }

    public void Contenedo3(View view){
        Toast.makeText(this, "Contenedor3", Toast.LENGTH_SHORT).show();
    }

    public void Contenedo4(View view){
        Toast.makeText(this, "Contenedor4", Toast.LENGTH_SHORT).show();
    }

    public void Contenedo5(View view){
        Toast.makeText(this, "Contenedor5", Toast.LENGTH_SHORT).show();
    }

    public void Contenedo6(View view){
        Toast.makeText(this, "Contenedor6", Toast.LENGTH_SHORT).show();
    }

    public void Contenedo7(View view){
        Toast.makeText(this, "Contenedor7", Toast.LENGTH_SHORT).show();
    }

    public void Contenedo8(View view){
        Toast.makeText(this, "Contenedor8", Toast.LENGTH_SHORT).show();
    }

    public void Contenedo9(View view){
        Toast.makeText(this, "Contenedor10", Toast.LENGTH_SHORT).show();
    }

    public void Contenedo10(View view){
        Toast.makeText(this, "Contenedor10", Toast.LENGTH_SHORT).show();
    }

}
```

# Ficha técnica microcontrolador Atmega2560.

## 1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560

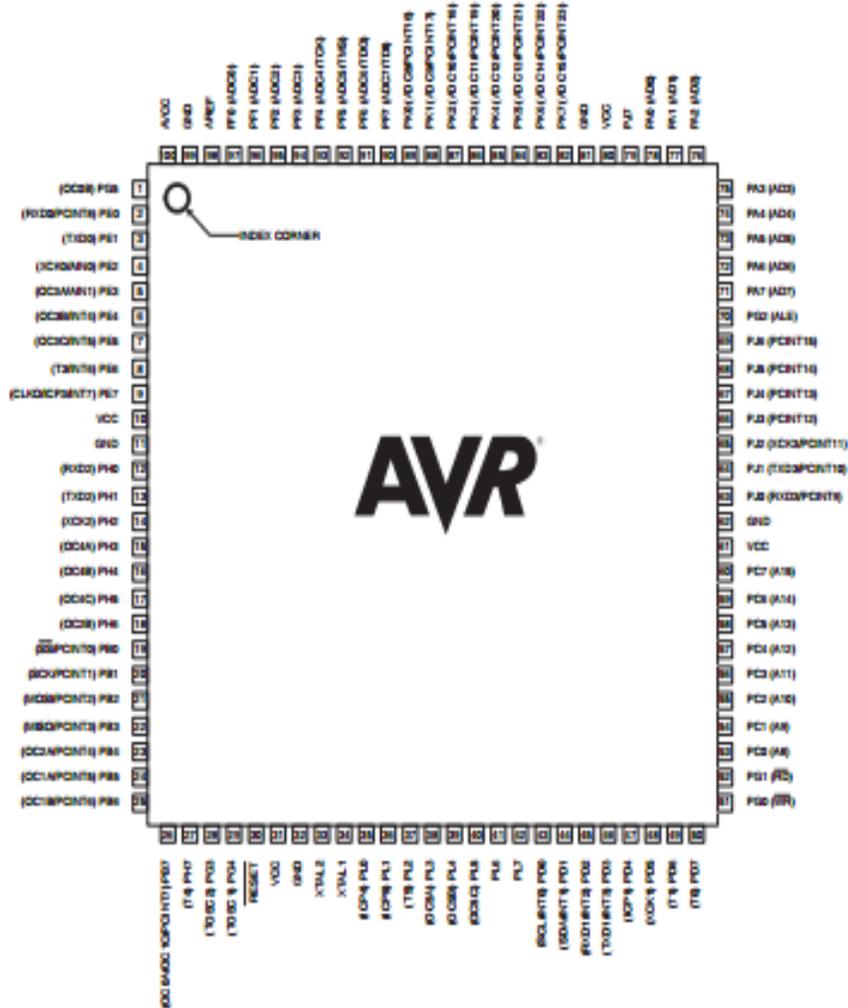


Figure 1-2. CBGA-pinout ATmega640/1280/2560

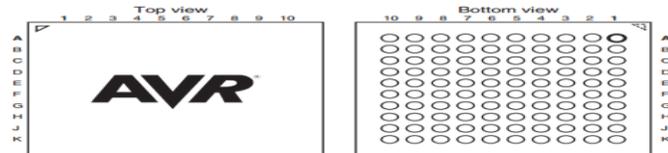


Table 1-1. CBGA-pinout ATmega640/1280/2560

	1	2	3	4	5	6	7	8	9	10
A	GND	AREF	PF0	PF2	PF5	PF0	PF3	PF6	GND	VCC
B	AVCC	PC5	PF1	PF3	PF6	PK1	PK4	PK7	PA0	PA3
C	PE0	PE0	PE1	PE4	PE7	PK2	PK5	PK7	PA1	PA3
D	PE3	PE4	PE5	PE6	PE6	PK4	PK5	PK6	PA7	PA2
E	PE7	PH0	PH1	PH3	PH5	PK6	PK6	PK4	PA3	PA2
F	VCC	PH4	PH5	PH0	PL4	PK1	PK1	PK0	PC7	GH0
G	GND	PB1	PB2	PB5	PL2	PK0	PK0	PK5	PC6	VCC
H	PC3	PC4	RESET	PL1	PL3	PL7	PK4	PK4	PC3	PC2
J	PH7	PC3	PB5	PL0	XTAL2	PL5	PK3	PK1	PC0	PC1
K	PH7	PC4	VCC	GND	XTAL1	PL5	PK2	PK2	PC6	PC0

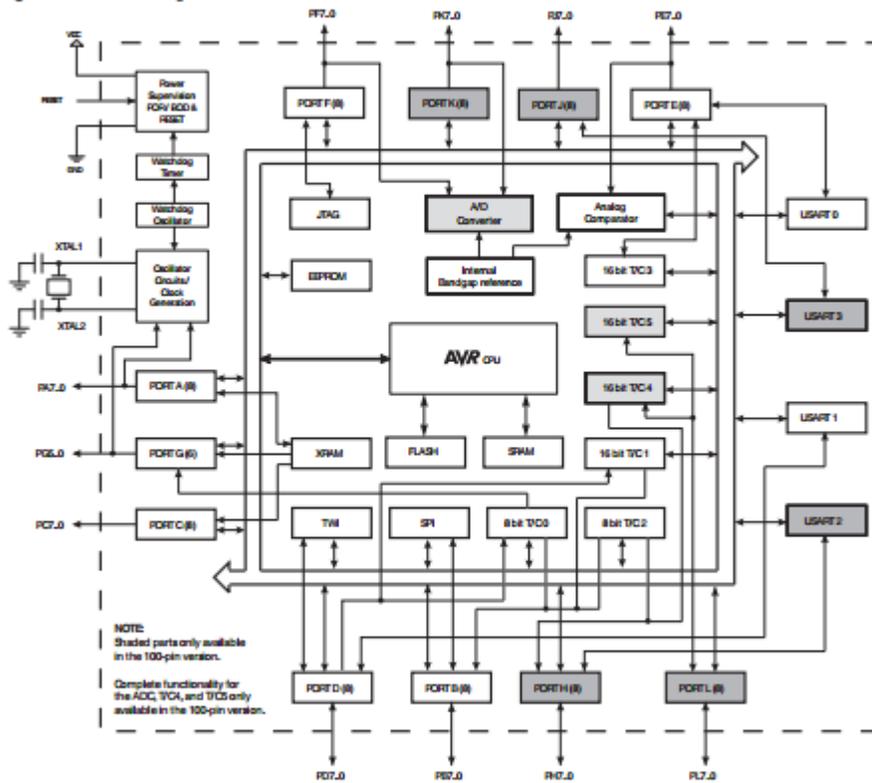
Note: The functions for each pin is the same as for the 100-pin packages shown in Figure 1-1 on page 2.

## 2. Overview

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### 2.1 Block Diagram

Figure 2-1. Block Diagram



The Atmel AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

## 2.2 Comparison Between ATmega1281/2561 and ATmega640/1280/2560

Each device in the ATmega640/1280/1281/2560/2561 family differs only in memory size and number of pins. [Table 2-1](#) summarizes the different configurations for the six devices.

**Table 2-1.** Configuration Summary

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

## 2.3 Pin Descriptions

### 2.3.1 VCC

Digital supply voltage.

### 2.3.2 GND

Ground.

### 2.3.3 Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on [page 75](#).

### 2.3.4 Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on [page 76](#).

### 2.3.5 Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the ATmega640/1280/1281/2560/2561 as listed on [page 79](#).

## ANEXO 4

## Ficha técnica del módulo SIM900



Smart Machine Smart Decision

## 2.3 SIM900 Functional Diagram

The following figure shows a functional diagram of SIM900:

- The GSM baseband engine
- Flash
- The GSM radio frequency part
- The antenna interface
- The other interfaces

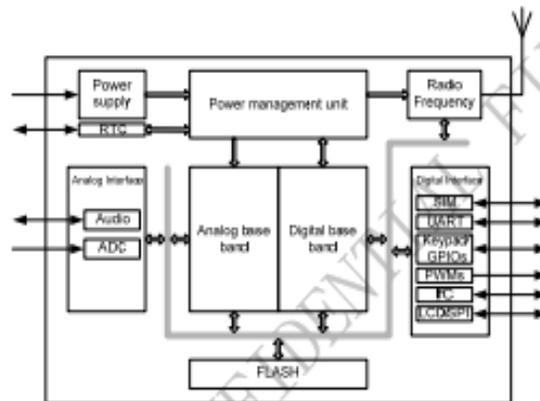


Figure 1: SIM900 functional diagram





### 3.2 Pin Description

Table 4: Pin description

Pin name	Pin number	I/O	Description	Comment
<b>Power supply</b>				
VBAT	55,56,57	I	Power supply	
VRTC	26	I/O	Power supply for RTC	It is recommended to connect with a battery or a capacitor (e.g. 4.7uF).
VDD_EXT	15	O	2.8V output power supply	If it is unused, keep open.
GND	17,18,29,39, 45,46,53,54, 58,59,61,62, 63,64,65		Ground	
<b>Power on/down</b>				
PWRKEY	1	I	PWRKEY should be pulled low at least 1 second and then released to power on/down the module.	Pulled up internally.
<b>Audio interfaces</b>				
MIC_P	19	I	Differential audio input	If these pins are unused, keep open.
MIC_N	20			
SPK_P	21	O	Differential audio output	
SPK_N	22			
LINEIN_R	23	I	Line-in input	
LINEIN_L	24			
<b>Status</b>				
STATUS	66	O	Power on status	If these pins are unused, keep open.
NETLIGHT	52	O	Network status	
<b>LCD interface</b>				
DISP_CLK	11	O	Display interface	If these pins are unused, keep open.
DISP_DATA	12	I/O		
DISP_DWC	13	O		
DISP_CS	14	O		
<b>I<sup>2</sup>C interface</b>				
SDA	37	O	I <sup>2</sup> C serial bus data	If these pins are unused, keep open.
SCL	38	I/O	I <sup>2</sup> C serial bus clock	
<b>Keypad interface / GPIOs</b>				
GPIO5/KBR0	44	I/O	GPIO5/keypad row 0	If these pins are unused, keep open.
GPIO4/KBR1	43		GPIO4/keypad row 1	
GPIO3/KBR2	42		GPIO3/keypad row 2	
GPIO2/KBR3	41		GPIO2/keypad row 3	



### 3.3 Package Dimensions

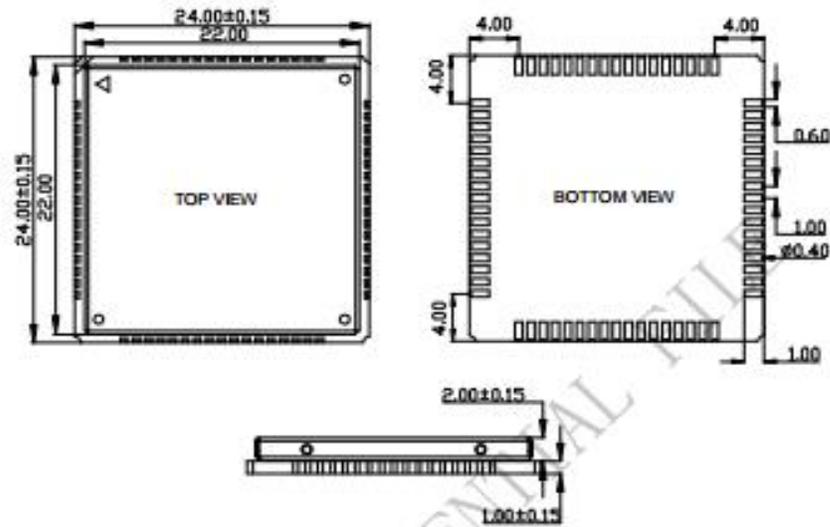


Figure 3: Dimensions of SIM900 (Unit: mm)

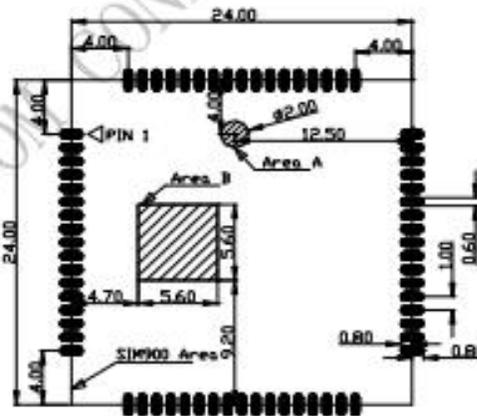


Figure 4: Recommended PCB footprint outline (Unit: mm)

*Note: Keep copper out of area A. Do not place via in area B to avoid short circuit between the via on customer board and the test points on the bottom side of the module.*

## Ficha técnica chip NEO-6 u-blox 6 GPS



### 1 Functional description

#### 1.1 Overview

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.

The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

#### 1.2 Product features

Model	Type					Supply		Interfaces					Features					
	IS	IP	Timing	Serial Data	Dual Locking	1.8V - 2.0V	2.7V - 3.6V	UART	I2C	USB	BT	BLE (BT compliant)	Programmable (Boot/ FW update)	TCOD	REC OVM	Address inputs and outputs	Configuration pins	Threshold
NEO-6G	*					*		*	*	*	*		*	*	0	3	1	*
NEO-6Q	*					*		*	*	*	*		*	*	0	3	1	*
NEO-6M	*					*		*	*	*	*		*	*	0	3	1	*
NEO-6P	*	*		*		*		*	*	*	*		*	*	0	3	1	*
NEO-6V	*				*	*		*	*	*	*		*	*	0	3	1	*
NEO-6T	*		*	*		*		*	*	*	*		*	*	0	3	1	*

0 = Requires external components and integration on application processor

Table 1: Features of the NEO-6 Series



All NEO-6 modules are based on GPS chips qualified according to AEC-Q100. See Chapter 5.1 for further information.



### 1.3 GPS performance

Parameter	Specification			
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix <sup>1</sup>		ISO-80001	ISO-8000	ISO-8P
	Cold Start <sup>2</sup>	26 s	27 s	32 s
	Warm Start <sup>2</sup>	26 s	27 s	32 s
	Hot Start <sup>2</sup>	1 s	1 s	1 s
Sensitivity <sup>4</sup>		ISO-80001	ISO-8000	ISO-8P
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition <sup>3</sup>	-160 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-157 dBm	-156 dBm	-155 dBm
Maximum Navigation update rate		ISO-80001	ISO-8000	
		5Hz	1 Hz	
Horizontal position accuracy <sup>5</sup>	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP <sup>6</sup>	< 1 m (2D, R50) <sup>7</sup>		
	SBAS + PPP <sup>6</sup>	< 2 m (3D, R50) <sup>7</sup>		
Configurable Timepulse frequency range		ISO-80001	ISO-8P	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated <sup>8</sup>	15 ns		
Velocity accuracy <sup>9</sup>		0.1m/s		
Heading accuracy <sup>9</sup>		0.5 degrees		
Operational Limits	Dynamics	≤ 4 g		
	Altitude <sup>10</sup>	50,000 m		
	Velocity <sup>10</sup>	500 m/s		

Table 2: NEO-6 GPS performance

<sup>1</sup> All satellites at -130 dBm

<sup>2</sup> Without aiding

<sup>3</sup> Dependent on aiding data connection speed and latency

<sup>4</sup> Demonstrated with a good active antenna

<sup>5</sup> For an outage duration <10s

<sup>6</sup> CEP, 50%, 24 hours static, -130dBm, SEP: <3.5m

<sup>7</sup> NEO-6P only

<sup>8</sup> Demonstrated under following conditions: 24 hours, stationary, first 600 seconds of data discarded, HDOP < 1.5 during measurement period, strong signals. Continuous availability of valid SBAS correction data during full test period.

<sup>9</sup> Quantization error information can be used with NEO-6T to compensate the granularity related error of the timepulse signal

<sup>10</sup> Assuming Airborne <4g platform

## 2 Pin Definition

### 2.1 Pin assignment

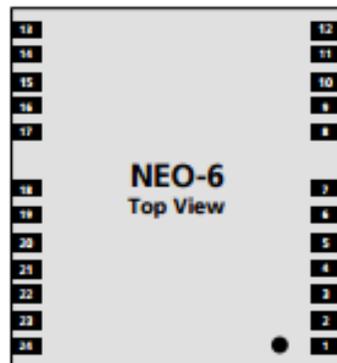


Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SPi Slave Select
3	All	TIMEPULSE	O	Timepulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	IO	USB Data
6	All	USB_DP	IO	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage RF section Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MOSI/CFG_COM0	O/I	SPi MOSI / Configuration Pin. Leave open if not used.
15	All	MISO/CFG_COM1	I	SPi MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPSDISCK	I	Power Mode Configuration Pin / SPi Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	IO	DDC Data
19	All	SCL2	IO	DDC Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1



## ANEXO 7

En el anexo 5 se indica las especificaciones técnicas de los componentes utilizados para obtener los datos de peso y nivel.



Tech Support: [services@elecfreaks.com](mailto:services@elecfreaks.com)

## Ultrasonic Ranging Module HC - SR04

### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The module includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time × velocity of sound (340M/S)) / 2,

### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

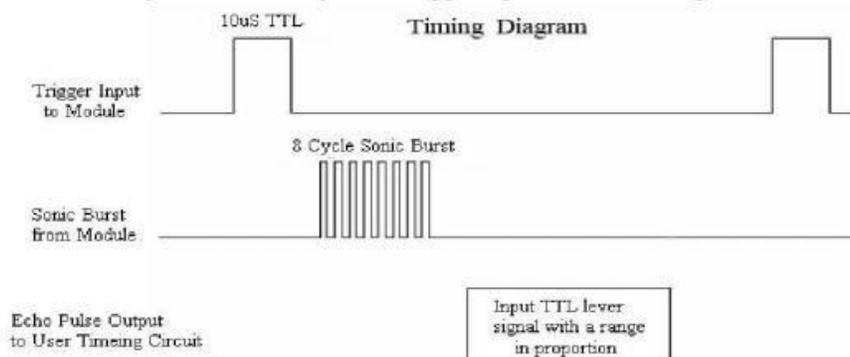
### Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm



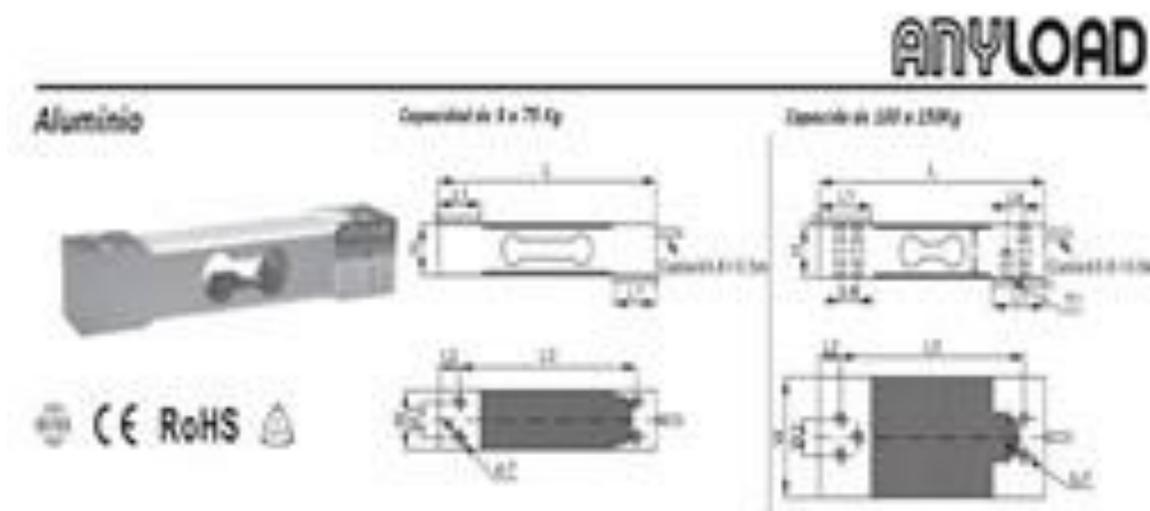
## Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 $\mu$ S pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula:  $\mu\text{S} / 58 = \text{centimeters}$  or  $\mu\text{S} / 148 = \text{inch}$ ; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



## CELDA DE CARGA DE 5 KG

Especificaciones técnicas de la celda de carga de 5 kg utilizada para conocer el peso de los desechos sólidos para el Prototipo de Control y Monitoreo



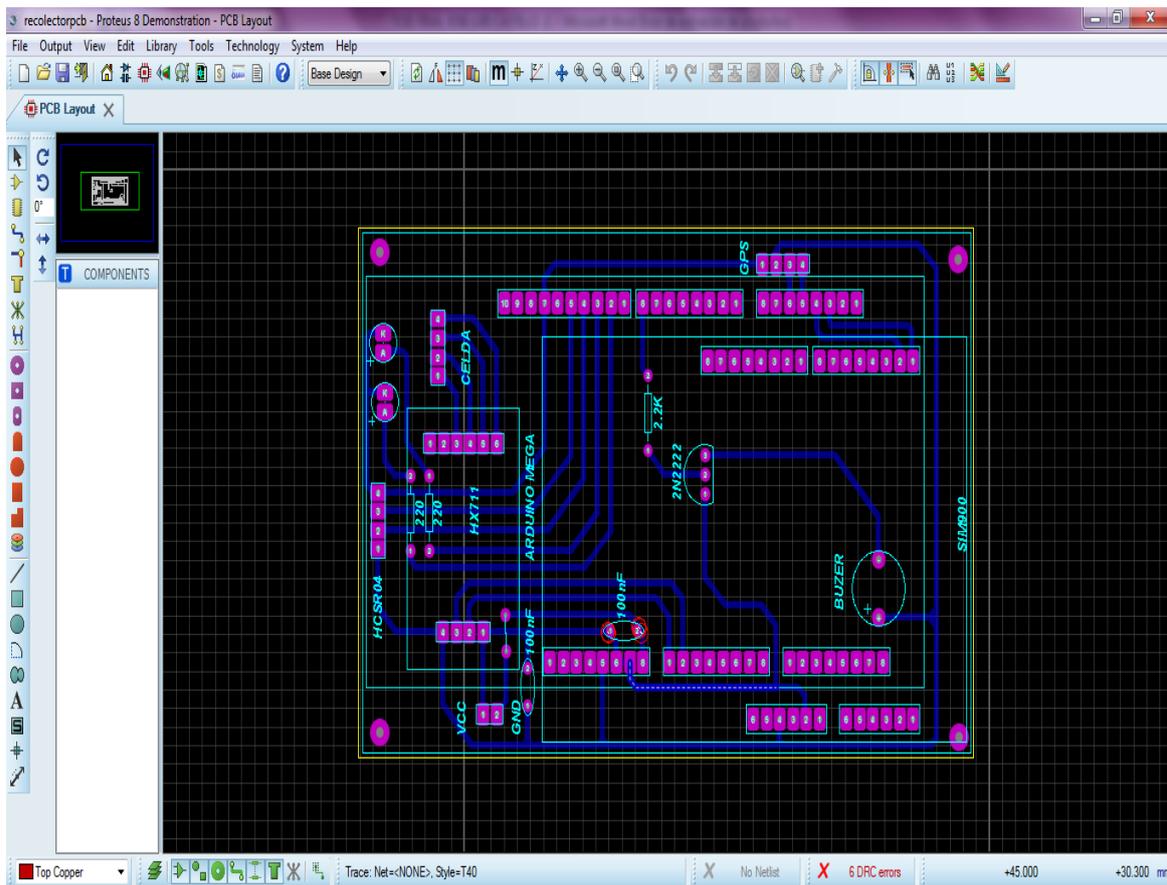
### DIMENSIONES

Capacidad Nominal	H	L	L1	L2	L3	L4	W	W1	T	T1
<b>kg/mm</b>										
3-30	22.0	130.0	25.0	12.0	106.0	-	25.4	15.0	M6	-
35	22.0	130.0	25.0	12.0	106.0	-	30.0	15.0	M6	-
40; 50; 60; 75	22.0	130.0	30.0	12.0	106.0	-	38.5	15.0	M6	-
100; 150	22.0	130.0	30.0	12.0	106.0	10.0	50.0	15.0	M6	M6 Depth10
<i>lb/pulgadas. conversion de dimensiones desde arriba</i>										
6.61-66.14	0.87	5.12	0.98	0.47	4.17	-	1.00	0.59	M6	-
77.16	0.87	5.12	0.98	0.47	4.17	-	1.18	0.59	M6	-
88.18; 110.23; 132.28; 165.35	0.87	5.12	1.18	0.47	4.17	-	1.52	0.59	M6	-
220.46; 330.69	0.87	5.12	1.18	0.47	4.17	0.39	1.97	0.59	M6	M6 Depth0.39

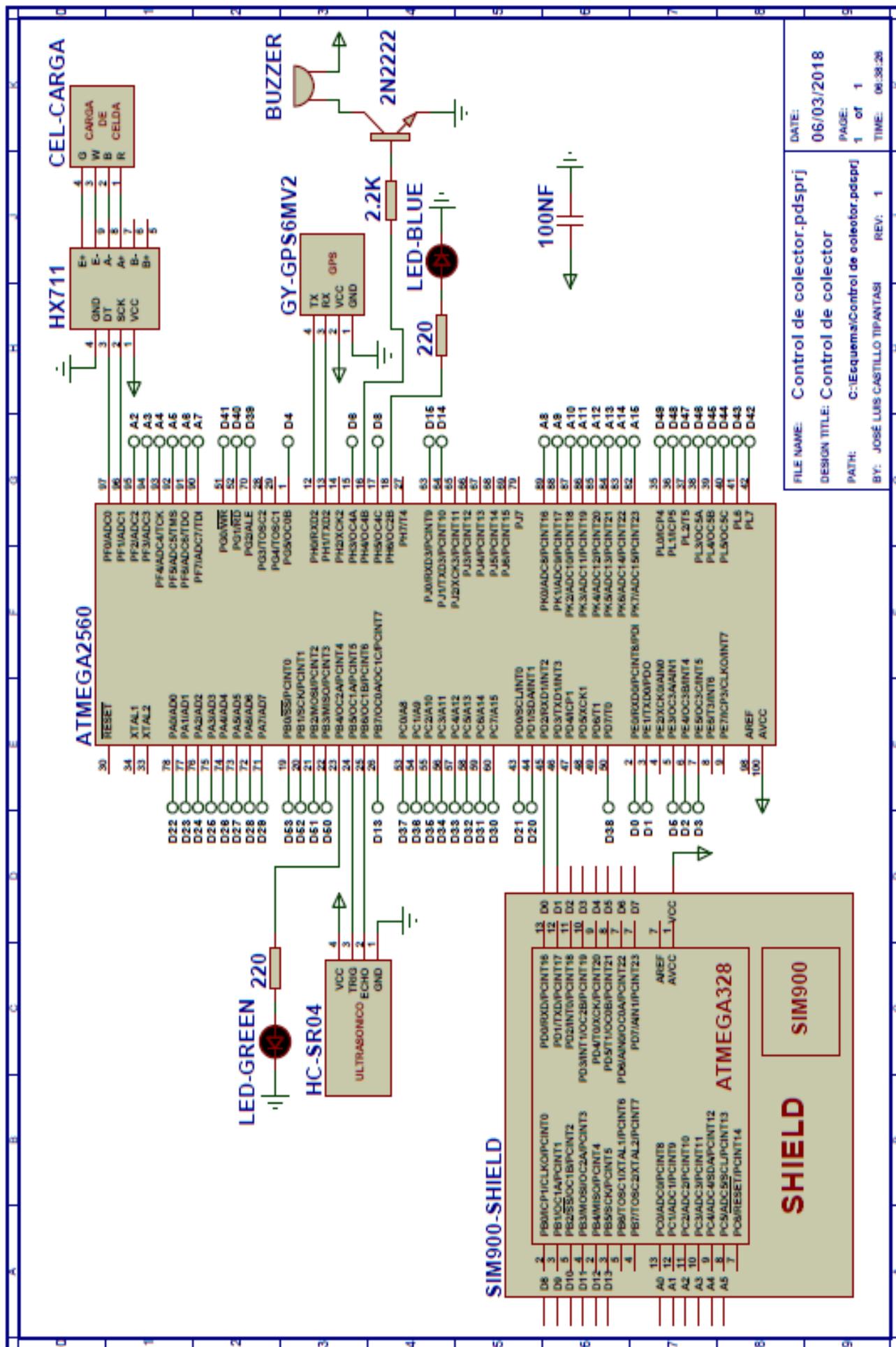
Capacidad nominal	3kg, 6kg, 7 kg, 10kg, 15kg, 20kg, 30kg, 35kg, 40kg, 45kg, 50kg, 60kg, 75kg, 100kg, 150kg
Referencia Max. Tamaño de la plataforma	400mm x 400mm
Escala de salida completa	2 mV / V $\pm$ 10%
Balance Cero	$\pm$ 0.02mV / V
No linealidad	$<\pm$ 0,017%
Repetibilidad	$<\pm$ 0,017%
Histéresis Error	$<\pm$ 0,017%
Fluencia en 30 minutos	$<\pm$ 0,023%
Resistencia de entrada	415 $\Omega$ $\pm$ 15 $\Omega$
Resistencia de salida	350 $\Omega$ $\pm$ 3 $\Omega$
Excitación recomendada	10V (15V Máximo)
Resistencia de aislamiento	$>$ 2G $\Omega$ (50V DC)
Efecto de la temperatura sobre la sensibilidad	$<\pm$ 0,0012% de Cn / k
Efecto de la temperatura sobre el Balance Cero	$<\pm$ 0,0040% de Cn / k
Rango de temperatura nominal	-10 ° C a 40 ° C / 14 ° F a 104 ° F
Servicio Rango de temperatura	-18 ° C a 65 ° C / 0 ° F a 149 ° F
Temperatura de almacenamiento	-50 ° C a 85 ° C / -58 ° F a 185 ° F
Sobrecarga segura	150% de la escala completa
Rotura de Sobrecarga	300% de la escala completa
Tipo de Seal	ambientalmente sellado

# ANEXOS 8.

## SIMULACIÓN EN PCB



ANEXO 9.  
PLANO ELECTRÓNICO



FILE NAME: Control de colector.pdsprj  
 DESIGN TITLE: Control de colector  
 PATH: C:\Esquemas\Control de colector.pdsprj  
 BY: JOSÉ LUIS CASTILLO TIPANTASI REV: 1  
 DATE: 06/03/2018  
 PAGE: 1 of 1  
 TIME: 06:38:28

## DECLARACIÓN Y AUTORIZACIÓN

Yo, José Luis Castillo Tipantasi, CI 1722228796 autor del trabajo de graduación: **PROTOTIPO DE UN SISTEMA AUTOMÁTICO PARA EL CONTROL DE LLENADO DE DEPÓSITOS DE BASURA CON MONITOREO REMOTO EN EL MERCADO MAYORISTA DE LA CIUDAD DE QUITO UTILIZANDO UNA APLICACIÓN EN ANDROID**, previo a la obtención del título de **Ingeniería en EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES** en la UNIVERSIDAD TECNOLÓGICA ISRAEL.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de difundir el respectivo trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de graduación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, Febrero del 2018

**Atentamente.**



**José Luis Castillo Tipantasi.**

**C.I. 1722228796**