

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



TESIS DE GRADO

“CONTROL DE DISPENSADORES DE ODORIZACIÓN BASADO EN ARDUINO”

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: NATHALY YALY ALFARO ORTEGA

TUTOR METODOLÓGICO: M.Sc. ALDO RAMIRO VALDEZ ALVARADO

ASESOR: LIC. GROVER ALEX RODRIGUEZ RAMIREZ

ASESOR ADJUNTO: M.Sc. REYNALDO JAVIER ZEBALLOS DAZA

LA PAZ – BOLIVIA

2015



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

El presente trabajo lo dedico al motor de mi vida, Dios.

Que ha guiado mis pasos y me ha fortalecido para lograr los objetivos que Él puso en mi corazón.

Isaías 41:10

“No temas, porque yo estoy contigo; no desmayes, porque yo soy tu Dios que te esfuerzo; siempre te ayudaré, siempre te sustentaré con la diestra de mi justicia.”

AGRADECIMIENTOS

A Dios que ha estado conmigo en todo momento de la etapa universitaria, bendiciéndome y cuidándome.

Agradecer a la Universidad Mayor De San Andrés, carrera de informática que me dio la oportunidad de empaparme de sus conocimientos.

A mi tutor metodológico M.Sc. Aldo Ramiro Valdez Alvarado por encaminarme a elaborar y estructurar la tesis de grado, por los consejos constantes, por las ideas y sugerencias dispensadas a mi persona. Por impartirme el rigor académico en todas las asignaturas dictadas que me han servido de mucho para superar cualquier dificultad, por su paciencia, dedicación, apoyo y profesionalismo.

Un sincero agradecimiento al Licenciado Grover Alex Rodríguez Ramírez, por su colaboración en la estructura de la tesis de grado, sus ideas, la información dispensada, la guía y consejos oportunos.

A mi asesor adjunto M.Sc. Reynaldo Javier Zeballos Daza por sus valiosos aportes y por su incondicional apoyo a lo largo de todo este proyecto y etapa universitaria, principalmente por transmitirme el conocimiento académico en este camino y el apoyo incondicional de todo un profesional.

A todos los docentes de la carrera de informática que en determinado momento de mi vida académica, formaron parte de mi educación y crecimiento profesional, que impartieron con profesionalismo todos sus amplios conocimientos en favor de nuestra formación académica.

Quiero agradecer a mi mamá Yaly Ortega Vega y mi tío Johnny Ortega Vega que me impulsaron y apoyaron, a ser mejor persona y superarme, dándome la confianza y fortaleza, para lograr todo lo que me proponga.

Un agradecimiento especial también a todos mis amigos que en determinado tiempo me impulsaron a seguir adelante y “no tirar la toalla”.

RESUMEN

En esta tesis, se ha desarrollado un sistema de control de los dispensadores de odorización, mediante el monitoreo de la cantidad de líquido existente, logrando la oportuna atención de los clientes que usan este servicio, de igual manera la reposición del producto en el momento preciso.

La empresa proveedora, es informada de la situación de los dispensadores de odorización en tiempo real, mediante un SMS que el prototipo emite, de esta manera la empresa proveedora pueda dar la oportuna atención a sus clientes.

Para lograr optimizar el control de los dispensadores de odorización, en tiempo real, se elabora un prototipo, en el cual se halla un módulo que maneja sensores, cuya función es obtener la distancia y de esta manera poder determinar la cantidad de líquido en el dispensador de odorización, como forma de salida de la información y con el objetivo de mandar una alerta temprana, se instala el sim900, esta placa es la encargada de mandar un SMS de acuerdo a la situación de los dispensadores de odorización. Todo esto basado en Arduino, Sensores ultrasónicos y la placa sim900.

En esta tesis se han descrito los conocimientos básicos para entender que es y cómo funciona un sistema de control automático, con el uso del hardware y software libre.

La presente tesis se divide en cinco partes.

En la introducción se va a poder dar un primer paso en el mundo de los dispensadores de odorización y conocer el porqué de la utilización de Arduino.

En la etapa del marco teórico se conoce, sobre el hardware y software libre, las placas Arduino y el entorno de trabajo que se dispone para poder programar.

En la etapa del marco aplicativo, se incorpora la metodología en V de esta manera efectuamos la funcionalidad del control de los dispensadores de odorización.

Acto seguido se expone el prototipo, en el que se aplicaron los conocimientos adquiridos.

Palabras clave: Control, sensor, software libre, hardware libre, Arduino, SMS, dispensador de odorización.

ABSTRACT

In this thesis, a system has been developed for control of the dispensers of odorization, by monitoring the amount of fluid, reaching the timely attention of the customers who use this service, in the same way the replacement of the product at the right moment.

The supplying company is informed of the situation of the dispensers of odorization in real time, through an SMS that the prototype emits, in this way the supplying company can give an appropriate attention to their customers.

For achieving optimum control of the dispensers of odorization, in real time, it develops a prototype, in which there is a module that handles sensors, whose function is to obtain the distance and being able to determine the amount of fluid in the dispenser of odorization, as a form of output of the information and with the goal of sending a early warning, it installs the sim900, this board is responsible for sending an SMS according to the situation of the dispensers of odorization.

In this thesis, we have described the basic knowledge to understand what it is and how it operates a system of automatic control, with the use of the hardware and free software.

This thesis is divided into five parts.

In the introduction is going to be able to take a first step in the world of the dispensers of odorization and know the because of the use of Arduino. In the stage of the theoretical framework is known, on the hardware and free software, the Arduino boards and working environment that is available for you to be able to Schedule.

In the stage of the application framework, incorporates the methodology in V in this way we the functionality of the control of the dispensers of odorization. He then exposes the prototype, in which were applied the knowledge gained.

Key Words: Control, sensor, free software, free hardware, Arduino, SMS, dispenser of odorization.

ÍNDICE

CAPITULO I MARCO REFERENCIAL	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES	2
1.2.1 CONTROL AUTOMÁTICO	2
1.2.1.1 CONTROL AUTOMÁTICO SIGLO III a. C.	2
1.2.1.2 CONTROL AUTOMÁTICO ÉPOCA MODERNA, EUROPA	3
1.2.1.3 AVANCES DURANTE LA SEGUNDA GUERRA MUNDIAL	4
1.2.1.4 SISTEMAS DE CONTROL EN LA ACTUALIDAD.....	5
1.2.2 DESCRIPCIÓN DEL OBJETO BAJO ESTUDIO DISPENSADOR DE ODORIZACIÓN.	7
1.3 PLANTEAMIENTO DEL PROBLEMA.....	8
1.3.1 PROBLEMA CENTRAL.....	8
1.3.2 PROBLEMAS SECUNDARIOS	9
1.4 DEFINICIÓN DE OBJETIVOS.....	9
1.4.1 OBJETIVO GENERAL	9
1.4.2 OBJETIVOS ESPECÍFICOS	9
1.5 HIPÓTESIS	10
1.5.1. OPERACIONALIZACIÓN DE VARIABLES.....	10
1.6 JUSTIFICACIÓN.....	10
1.6.1 JUSTIFICACIÓN ECONÓMICA.....	10
1.6.2 JUSTIFICACIÓN SOCIAL	11
1.6.3 JUSTIFICACIÓN CIENTÍFICA.....	11
1.7 ALCANCES Y LÍMITES	11
1.7.1. ALCANCES	11

1.7.2. LÍMITES	12
1.8 APORTES	12
1.8.1. PRÁCTICO	12
1.8.2 TEÓRICO	12
1.9 METODOLOGÍA.....	13
1.9.1 METODOLOGÍA CIENTÍFICA	13
1.9.2 METODOLOGÍA DE DESARROLLO DEL SOFTWARE LIBRE Y HADWARE.	14
CAPITULO II MARCO TEÓRICO	15
2.1 SOFTWARE LIBRE	15
2.1.1 SOFTWARE LIBRE EN BOLIVIA	16
2.1.1.1 DECRETO SUPREMO N° 1793 DE 13 DE NOVIEMBRE DE 2013, REGLAMENTO PARA EL DESARROLLO DE LAS TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN.	16
2.2 HARDWARE LIBRE	20
2.3 MÉTODO EN V	21
2.3.1 NIVELES DE LA METODOLOGÍA EN V	22
2.4 DISPENSADORES DE ODORIZACIÓN.....	24
2.5 ARDUINO.....	25
2.5.1 QUÉ ES ARDUINO?	25
2.5.2 SOFTWARE PARA ARDUINO	25
2.5.2.1 ENTORNO ARDUINO.....	26
2.5.3 BENEFICIOS DE ARDUINO	29
2.5.4 ARDUINO UNO	29
2.5.4.1 PLACA ARDUINO Y SUS PARTES	30
2.5.4.2 ENTRADAS Y SALIDAS DE ARDUINO UNO.....	31
2.5.4.3 UNIVERSAL SERIAL BUS USB.....	32

2.5.4.4 COMUNICACIÓN.....	33
2.5.4.5 PROTECCIÓN CONTRA SOBRE TENSIONES EN EL USB	33
2.5.4.6 SHIELDS ARDUINO	34
2.6 SENSORES	35
2.6.1 SENSOR ULTRASÓNICO.....	35
2.7 PLACA DE PRUEBAS O PROTOBOARD.....	36
2.7.1 ZONAS DEL PROTOBOARD.....	37
2.8 INTERNET DE LAS COSAS.....	38
2.9 SISTEMAS AUTOMÁTICOS Y DE CONTROL.....	39
2.9.1 NECESIDAD Y APLICACIONES DE LOS SISTEMAS AUTOMÁTICOS DE CONTROL	39
CAPITULO III MARCO APLICATIVO	41
3.1 INTRODUCCIÓN.....	41
3.2 INGENIERÍA DE REQUERIMIENTOS.....	42
3.2.1 REQUERIMIENTOS DEL USUARIO:	42
3.2.2 REQUERIMIENTOS DEL HADWARE.....	42
3.2.3 REQUERIMIENTOS DEL SOFTWARE.....	44
3.3 DISEÑO DEL SISTEMA.....	44
3.3.1 IMPLEMENTACIÓN DEL PROTOTIPO	45
3.4 VERIFICACIÓN DEL SISTEMA	50
3.4.1 VERIFICACIÓN DE LA INSTALACIÓN DEL SENSOR ULTRASÓNICO EN LA PLACA ARDUINO.....	50
3.4.2 VERIFICACIÓN DE LA INSTALACIÓN DEL MÓDULO SIM900.....	51
3.4.3 VERIFICACIÓN DE LA INSTALACIÓN DEL SISTEMA EN EL EQUIPO DE ODORIZACIÓN	52
3.5 DISEÑO DEL SOFTWARE	53

3.5.1 DESARROLLO DEL ALGORITMO PARA EL SENSOR ULTRASÓNICO.....	53
3.5.2 DESARROLLO DEL ALGORITMO PARA EL SIM 900	55
3.6 VERIFICACIÓN DEL SOFTWARE.....	58
3.6.1 ALGORITMO PARA EL SENSOR ULTRASÓNICO	58
3.6.1.1 ENVASE LLENO	58
3.6.1.2 ENVASE SEMI/LLENO.....	58
3.6.1.3 ENVASE VACÍO.....	59
3.6.2 VERIFICACIÓN DEL ALGORITMO PARA EL ENVÍO DE SMS	60
3.6.2.1 VERIFICACIÓN DEL DISPENSADOR LLENO	60
3.6.2.2 VERIFICACIÓN DEL DISPENSADOR CON LA MITAD DEL PRODUCTO ...	60
3.6.2.3 VERIFICACIÓN DEL DISPENSADOR VACÍO.	61
3.7 ENTRELAZAMIENTO DE DISPOSITIVOS.....	62
CAPITULO IV PRUEBA DE HIPOTESIS	63
4.1 INTRODUCCIÓN.....	63
4.1.1 INVESTIGACIÓN CUALITATIVA	63
4.1.1.1 CALIDAD DE LAS ACTIVIDADES Y CASOS ENCONTRADOS.....	63
4.1.1.2 FUNCIONAMIENTO DE CADA CASO.....	65
4.1.1.3 FUNCIONAMIENTO DEL PROTOTIPO	66
4.1.1.4 MATERIALES O INSTRUMENTOS:	67
CAPITULO V CONCLUSIONES Y RECOMENDACIONES	69
5.1 CONCLUSIONES GENERALES	69
5.2 ESTADO DE OBJETIVOS ESPECIFICOS	69
5.3 ESTADO DEL OBJETIVO GENERAL.....	70
5.4 RECOMENDACIONES	70
CAPITULO VI REFERENCIAS BIBLIOGRÁFICAS.....	71
6.1 REFERENCIAS BIBLIOGRAFÍCAS	71

6.1.1 LIBROS Y TESIS	71
6.1.2 SITIOS WEB.....	72

GLOSARIO

ANEXOS

ÍNDICE DE TABLAS

CAPITULO III

Tabla 3.1 Elementos del prototipo.....	44
--	----

CAPITULO IV

Tabla 4.1 Casos encontrados	64
-----------------------------------	----

ÍNDICE DE FIGURAS

CAPITULO I

Figura 1.1 Dispensador de Odorización Patentado en el año 1969	8
--	---

CAPITULO II

Figura 2.1 Definición gráfica del software libre.....	15
---	----

Figura 2.2 Metodología en V.....	22
----------------------------------	----

Figura 2.3 Niveles de la metodología en V	23
---	----

Figura 2.4 Dispensador de Odorización	24
---	----

Figura 2.5 Partes del dispensador de Odorización	25
--	----

Figura 2.6 IDE para Arduino.....	27
----------------------------------	----

Figura 2.7 Sim900	34
-------------------------	----

Figura 2.8 Sensor ultrasónico	35
-------------------------------------	----

Figura 2.9 Protoboard.....	36
----------------------------	----

Figura 2.10 Zonas del protoboard.....	37
---------------------------------------	----

CAPITULO III

Figura 3.1 Estructura del prototipo.....	45
--	----

Figura 3.2 Sensor instalado en el protoboard	46
Figura 3.3 Conexión del Sensor ultrasónico con Arduino.....	47
Figura 3.4 Preparación del módulo GPRS	47
Figura 3.5 Conexión del chip a la tarjeta sim900 Fuente: Elaboración Propia	48
Figura 3.6 Instalación sim900 y Arduino	48
Figura 3.7 Conexión de la tarjeta sim900 al Arduino.....	49
Figura 3.8 Elaboración de la cavidad en la botella.....	50
Figura 3.9 Verificación del sensor ultrasónico en la placa Arduino	51
Figura 3.10 Verificación de la correcta instalación.....	52
Figura 3.11 Instalación del sensor ultrasónico en la botella.....	52
Figura 3.12 Interfaz de programación	53
Figura 3.13 Interfaz de programación	54
Figura 3.14 Interfaz de programación	55
Figura 3.15 Interfaz de programación	56
Figura 3.16 Interfaz de programación	57
Figura 3.17 Verificación del software líquido lleno.....	58
Figura 3.18 Verificación del software con el líquido a la mitad	59
Figura 3.19 Verificación del software botella vacía.....	59
Figura 3.20 Verificación del envase lleno	60
Figura 3.21 Verificación de envase Semi/LLeno vía SMS	61
Figura 3.22 Verificación del envase vacío vía SMS	61
Figura 3.23 Entrelazamiento de dispositivos.....	62
CAPITULO IV	
Figura 4.1 Funcionamiento del prototipo	67
Figura 4.2 software Arduino.....	68

CAPITULO I

MARCO REFERENCIAL

1.1 INTRODUCCIÓN

Desde el año 2012, en la ciudad de La Paz, Bolivia nos hemos visto inundados en diferentes baños de restaurantes, oficinas, centros de shopping, cines, entre otros, por los dispensadores de odorización.

Los dispensadores de odorización son llenados con un líquido llamado bactericida cuya función es que el baño se mantenga fresco, desinfectado y libre de malos olores, los dispensadores tienen el objetivo de que el producto gotee hacia el sanitario de manera automática las 24 horas al día, los 30 días del mes. Pero lamentablemente no siempre se logra este cometido, encontramos el problema de que no existe un buen goteo o que el producto se acaba antes del tiempo establecido. El producto es recargado mensualmente, la empresa hace el control una vez al mes y no existe ninguna manera de ver en qué fecha o cuando se acabó el producto.

Los dispensadores son entregados en calidad de préstamo e instalados de manera gratuita, la empresa que da el servicio cobra de manera mensual, siendo así su deber, que al cliente nunca le falte el producto y que el equipo tenga un buen funcionamiento.

En Bolivia existen empresas como Prohigiene, OpenClean, Clean, entre otras reconocidas en funda empresa las cuales brindan este servicio, los equipos que usan estas empresas son iguales en cuanto a funcionamiento se refiere, la diferencia entre una y otra, está en el producto.

De esta manera el presente trabajo busca coadyuvar en el objetivo de las empresas de mencionado servicio, implementando un sistema de control constante para los dispensadores de odorización, basado en Arduino, GPS y SMS; para que este de una alerta temprana mediante tecnología móvil, el cual mantendrá a la empresa que ofrece el servicio al tanto de sus clientes y equipos. Este control es conocido como control automático de procesos, el cual es parte del progreso industrial desarrollado durante la segunda revolución

industrial. La ciencia de control automático ha ido creciendo y evolucionando, gracias a las técnicas de medición y control.

El control automático es el mantenimiento de un valor deseado dentro de una cantidad o condición, midiendo el valor existente, comparándolo con el valor deseado, y utilizando la diferencia para proceder a reducirla. En consecuencia, el control automático exige un lazo cerrado de acción y reacción que funcione sin intervención humana, logrando la eliminación de errores.

El uso de las computadoras analógicas y digitales ha posibilitado la aplicación de ideas de control automático a sistemas físicos que hace apenas pocos años eran imposibles de analizar o controlar.

1.2 ANTECEDENTES

1.2.1 CONTROL AUTOMÁTICO

El control automático, a través de la historia fue evolucionando, en esta parte realizaremos una descripción y la situación actual del mismo.

1.2.1.1 CONTROL AUTOMÁTICO SIGLO III a. C.

La historia del control automático se remonta desde los principios de la civilización. Las primeras aplicaciones de control automático, surgen en Grecia, describen a los mecanismos reguladores con flotador, cabe destacar la presencia de tres mecánicos: Ktesibios, Philon y Herón.

En el siglo III antes de Cristo Ktesibios, diseño un reloj de agua, conocido también como Clepsydra este consistía en un mecanismo cuyo objetivo era que el nivel de un depósito de agua subiera con una velocidad constante. Inspirándose en esta idea el filósofo Platón plantea un reloj de agua que funcionaba como un sistema de alarma basándose en una Clepsydra. En el vaso de la Clepsydra se ubicó un flotador, encima del cual se depositan

unas bolas. Durante la noche se llenaba el vaso y al amanecer alcanzaba su máximo nivel y las bolas caían, sobre un plato de cobre.

Philon de Bizancio, construyó un sistema de regulación de nivel de una lámpara de aceite. Al quemarse el aceite de la lámpara, el nivel del depósito de aceite bajaba, haciendo que entrara aire en otro depósito de forma que éste suministraba más aceite, al depósito de la lámpara.

Herón también construye un Odómetro, este es un instrumento que tenía el fin de medir la distancia recorrida por un vehículo. El sistema consistía en una transmisión, que cada vez que daba una vuelta la rueda final, caía una bola en un contenedor. Solo había que contar el número de bolas para conocer la distancia recorrida.

1.2.1.2 CONTROL AUTOMÁTICO ÉPOCA MODERNA, EUROPA

En esta época de la historia en el año 1200, Lansperg realizó un sistema de control de un molino de harina, de forma que la cantidad de grano suministrada al molino dependía de la fuerza del viento y la dureza del propio grano, permitiendo que el sistema funcionara en condiciones óptimas, no se pretendía moler a velocidad constante.

En el siglo XVII se presentaron diversos sistemas de regulación de temperatura, entre ellos los aplicados en el horno y la incubadora de Drebbel. El principio utilizado en la regulación de temperatura es el siguiente, si la temperatura del horno sube, se dilata el contenido de un depósito de alcohol, de forma que se desplaza un juego de palancas que abre un orificio de salida de gases.

En el año 1745, E. Lee inventa un sistema para controlar automáticamente la orientación e inclinación de las aspas de los molinos de viento, de modo que se aprovechara mejor la dirección del viento. Se trataba del primer servomecanismo de posición el cual fue patentado bajo el nombre de "Self-regulating Wind Machine". En esta patente se describen dos mecanismos:

- El molinillo de cola, el cual no gira si no se encuentra en la dirección normal al viento, y por lo tanto no hace girar la cúpula del molino.

- Un variador automático del ángulo de ataque de las aspas. Con el que se podía regular la velocidad de giro de las aspas del molino.

Este segundo mecanismo no se llegó a realizar debido a su complicación constructiva. El ingenio de Lee se implantó rápidamente en Inglaterra y en el norte de Alemania.

Cuando el grano de trigo es molido, la calidad de la harina producida depende de la distancia entre las dos ruedas, la móvil y la fija, y de la velocidad de rotación de la primera.

En las últimas décadas del siglo XVII, muchos investigadores se dedicaron a desarrollar dispositivos que consigan controlar estos dos factores, de esta manera, en 1787, Thomas Mead patenta un diseño que combinaba la solución de los dos problemas. El invento disponía de un regulador que aseguraba que la presión ejercida entre las piedras del molino fuera proporcional a la velocidad de rotación. Este se combinaba con otro ingenio que variaba el ángulo de ataque de las aspas del molino, de forma que se controlaba la velocidad del molino. Mead implemento como sensor de velocidad, un péndulo rotativo precursor de los reguladores centrífugos.

El control automático se caracterizó por ser intuitivo hasta finales del siglo XIX de esta manera se pretendió mejorar las respuestas transitorias y la exactitud de los sistemas de control, lo cual obligó a desarrollar la teoría de control, de esta manera en 1868 J.C. Maxwell (1831-1879), consideró una teoría matemática relacionada con la teoría de control usando el modelo de una ecuación diferencial. En 1976 I.A. Vyshnegradskii formuló, una teoría matemática de los reguladores de manera independiente a Maxwell, con posible influencia europea, planteando los diagramas de estabilidad.

1.2.1.3 AVANCES DURANTE LA SEGUNDA GUERRA MUNDIAL

Durante la Segunda Guerra Mundial frente a la necesidad que implicaba la guerra, se desarrolló sistemas de control aplicados, como los servos de los radares y el posicionamiento de cañones.

La Segunda Guerra Mundial creó una necesidad urgente para diseñar servomecanismos de altas prestaciones y condujo a grandes avances en la forma de construir sistemas de control

realimentados. Las exigencias de la guerra enfocaron la atención, sobre el problema de control de tiro, proporcionando una cadena automática de órdenes entre la detección del blanco, el apuntamiento del arma y el disparo.

Con el objetivo de investigar y avanzar en los problemas de control del radar y de control de tiro, en marzo de 1942, de una manera informal se conforma un grupo denominado "Servo-Panel". Su principal función consistió en organizar encuentros, proporcionar información y servir de nexo de comunicación entre diferentes grupos de investigación.

El gobierno americano al intentar desarrollar los sistemas de control automático de tiro, se enfrentó con el problema de que aunque había una considerable experiencia, en temas de control, ésta se encontraba dispersa entre muchas ramas de la ingeniería y faltaba el atributo unificador de una terminología en común. La reacción no se hizo esperar, con la formación en 1940 bajo la dirección del Dr. Vannevar Bush del Comité de Investigación de Defensa Nacional (NDRC).

Entre los muchos comités del NDRC estaba el de Control de Tiro que bajo el liderazgo de Warren Weaver coordinó el trabajo de los servicios, laboratorios industriales y universidades. El comité era responsable de la dirección de la investigación y de la circulación de informes reservados a los grupos apropiados.

1.2.1.4 SISTEMAS DE CONTROL EN LA ACTUALIDAD

Actualmente los sistemas de control se siguen estudiando y aplicando tal es el caso de los siguientes trabajos que veremos a continuación:

- TITULO: “Modelado, Análisis y Diseño del Sistema de Control de un Robot Gantry usando un Sistema Operativo de Tiempo Real Convencional”.
- AUTOR: JULIA GARIBALDI
- AÑO: 2009
- INSTITUCION: Universidad de Buenos Aires, Facultad de Ingeniería.

RESUMEN:

Este trabajo, se basa fundamentalmente, en el estudio de un robot industrial, de tipo Gantry o cartesiano. Estos son robots fijos y sus ejes principales son ejes de desplazamiento longitudinal o prismático, que forman entre sí ángulos rectos. Las tareas que desempeña todo robot industrial requieren que el extremo del brazo que sostiene la herramienta se mueva y pase por puntos determinados en el espacio. Para ello, el robot debe contar con un algoritmo generador o interpolador de trayectoria. Este algoritmo se encarga de tomar muestras sucesivas de la trayectoria que debe ejecutar el robot en función del requerimiento de la tarea. Estas muestras se entregan a un sistema de control, que garantice que los ejes se muevan en el espacio con precisión. Basándose en la cinemática del robot, para transformar los puntos calculados, en valores que acepta el controlador. Para lograr que los ejes del robot realicen el movimiento deseado se requiere un sistema de control apropiado.

- TITULO: “Sistema de control de riego automático mediante el monitoreo de humedad del suelo vía internet”.
- AÑO: 2013
- AUTOR: ROSARIO CAMARGO GARCIA
- INSTITUCION: Universidad Autónoma de Querétaro, Facultad de Ingeniería.

RESUMEN:

La tesis planteada tiene el fin de reponer la humedad necesaria, mediante el control y monitoreo de la variable de humedad, de manera que el riego sea de forma automática, para satisfacer tal necesidad, implementando una tarjeta Arduino, con un servidor web, de tal forma que se generen ahorros del 50% en agua.

- TITULO: “Diseño e implementación de un sistema de Control de Ruta Vehicular Mediante Comparación con el uso del GPS”.
- AÑO: 2012

- AUTOR: DIANA NATIVIDAD PAZMIÑO PINEDA
- INSTITUCION: Universidad Tecnológica Israel, carrera de electrónica.

RESUMEN:

Trata de un sistema llamado GPS LOCARD el cual funciona con software y hardware libre denominado Arduino UNO, con una interface totalmente amigable.

El GPS LOCARD adicionalmente cuenta con un GPS que capta las señales emitidas por el satélite captando la longitud, tiempo, velocidad, y la altura de los vehículos que porten el dispositivo, posteriormente se almacenan los datos en una MicroSD en formato de archivo CV.

Para poder visualizar la ruta establecida por los vehículos es necesario introducir la MicroSD en un computador, descargar el archivo obtenido y subirlo en Google Earth.

Pudiendo comparar la ruta establecida originalmente, con la ruta realizada por los vehículos, de esta manera obtener un control adecuado de los vehículos.

1.2.2 DESCRIPCIÓN DEL OBJETO BAJO ESTUDIO DISPENSADOR DE ODORIZACIÓN.

Muy poco o casi nada se sabe, sobre el origen de los dispensadores de odorización por goteo, sin embargo encontramos que el 16 de Julio de 1969, fue patentado un dispensador llamado “aparato desodorizador perfeccionado”, caracterizado esencialmente por el hecho de comprender un envase alargado, provisto de una tapa en forma de cazoleta que encaja a presión en la boca del envase y que presenta en su fondo unos orificios comunicantes con un cuerpo solidario al fondo de la tapa y que se proyecta axialmente en el interior del envase, estando revestido dicho cuerpo con un material absorbente apto para impregnarse del líquido desodorante previamente vertido en la cazoleta que forma la tapa y que pasa a través de los orificios de su fondo, presentando el envase una orejeta con orificio para su colgado y una serie de ventanas en su pared lateral para que pueda salir al exterior la emanación del producto desodorante.

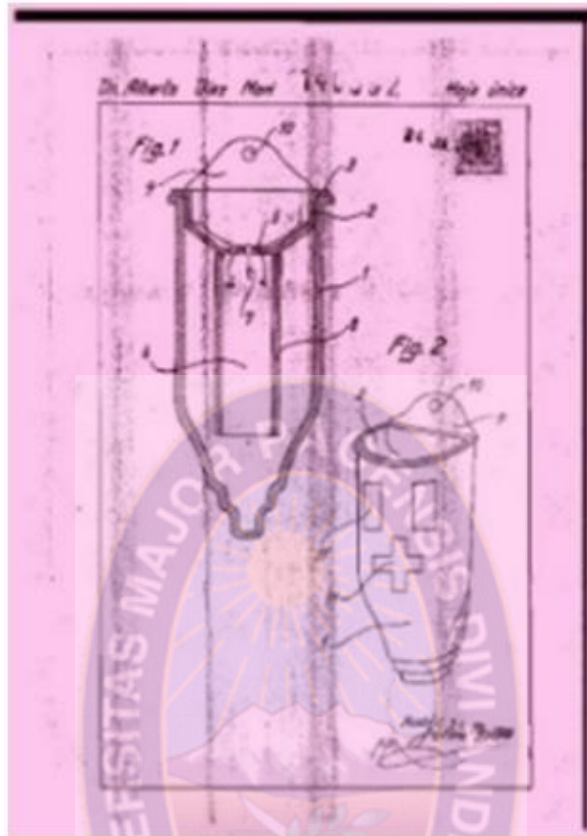


Figura 1.1 Dispensador de Odorización Patentado en el año 1969
Fuente: Patentados.com, 2005

1.3 PLANTEAMIENTO DEL PROBLEMA

En Bolivia y el mundo existen empresas de limpieza servicios, enfocadas a brindar servicios de odorización.

Estas empresas no cuentan con un sistema de control diario y constante, solo se cuentan con un control mensual y manual, que se efectúa con una revisión y mantenimiento, pero muchas veces se encuentran con el problema de que el producto se acaba antes de tiempo establecido o que no existió un buen goteo, dejando al cliente con un mal servicio.

1.3.1 PROBLEMA CENTRAL

¿Cómo se puede mejorar el control de los dispensadores de odorización?

1.3.2 PROBLEMAS SECUNDARIOS

- Existe un mal goteo, lo que lleva a que la empresa que ofrece el servicio no se muestre responsable.
- La empresa que da el servicio no tiene conocimiento de cuánta es su demanda mensual, lo que ocasiona que se produzca un desabastecimiento.
- El producto se acaba antes del tiempo establecido, lo que ocasiona que el cliente se quede desabastecido y sin servicio.
- Falta de un buen servicio lo que ocasiona una pérdida de clientes.
- La empresa que ofrece el servicio solo hace una revisión mensual lo que ocasiona que el dispensador de odorización no tenga un buen control.

1.4 DEFINICIÓN DE OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un sistema de control del funcionamiento y alerta temprana de los dispensadores de odorización, basados en hardware y software libre.

1.4.2 OBJETIVOS ESPECÍFICOS

- Mostrar una alerta temprana para evitar el mal goteo, de esta manera la empresa se muestra responsable y cumple con una buena funcionalidad en sus productos.
- Utilizar los recursos tecnológicos, para que exista una comunicación entre los dispensadores de odorización y la empresa que provee estos servicios, vía SMS.
- Dar una alerta temprana, para que el producto no se acabe antes de tiempo establecido, de esta manera el cliente siempre estará abastecido con el producto.
- Desarrollar un algoritmo que mida la cantidad de producto que gasta cada mes, de esta manera tener el dato exacto de cuanto es la demanda mensual.

- Construir un prototipo de control, que sea capaz de mantener informada a las empresas que ofrecen el servicio de odorización, sobre el estado de los dispensadores en tiempo real, de esta manera las empresas proveedoras, brindan una mejor atención a sus propios clientes manteniendo los que tienen y sumando nuevos.
- Mejorar el control, manipulando la información, que brinda el prototipo al emitir un SMS, cuando el equipo presente algún error de lectura.

1.5 HIPÓTESIS

El uso de un sistema de control basado en Arduino, sensores ultrasónicos y SMS permite asegurar la oportuna atención a los clientes y regular el buen funcionamiento de los dispensadores de odorización.

1.5.1. OPERACIONALIZACIÓN DE VARIABLES

1.5.1.1 VARIABLE INDEPENDIENTE

Sistema de control, Arduino, sensores ultrasónicos, GPS y SMS.

1.5.1.2 VARIABLE DEPENDIENTE

Regula el funcionamiento de los dispensadores de odorización.

1.6 JUSTIFICACIÓN

1.6.1 JUSTIFICACIÓN ECONÓMICA

La implementación del sistema de control basado en Arduino para los dispensadores de odorización generara mayores ingresos, al brindar un buen control y servicio, se podrán

generar nuevos clientes y mantener los propios aumentando de manera sustancial los ingresos empresariales.

La empresa proveedora del servicio reduce costos al eliminar la mano de obra pasiva, y así la contratación de empleados para el área de control, de esta manera se recupera la inversión de la implementación del sistema de control.

Al implementarlo en Arduino, GPS, SMS, los costos son mínimos, puesto que estamos hablando de hardware y software libre.

1.6.2 JUSTIFICACIÓN SOCIAL

Las empresas de servicios de odorización se benefician, al obtener un sistema de control automatizado brindando y teniendo al alcance de manera continua la información, de esta manera, una vinculación con el cliente y los dispensadores, sin necesidad de estar presentes de manera física.

1.6.3 JUSTIFICACIÓN CIENTÍFICA

Podremos adentrarnos al mundo del Arduino, sensores, Software Libre y Hardware Libre dejando un precedente para los futuros investigadores. Se pretende desarrollar un sistema con esta tecnología dejando los conocimientos adquiridos y la investigación de la conjunción de todos estos elementos, además de aplicar la ingeniería de control.

1.7 ALCANCES Y LÍMITES

1.7.1. ALCANCES

El “control de los dispensadores de odorización basado en Arduino” está orientado a:

- Medir la cantidad de líquido existente en el dispensador de odorización.
- Detectar si el dispensador de odorización se encuentra vacío, lleno, o semi/lleño.
- Emitir SMS, con la información del estado de los dispensadores de odorización.

- Mejorar el control eficiente y constante en tiempo real de los equipos de odorización.
- A una comunicación entre la empresa que ofrece el servicio y los equipos de odorización.
- Es capaz de enviar una alerta temprana vía SMS al celular de la empresa que ofrece el servicio.

1.7.2. LÍMITES

- No está a su alcance solucionar el problema de mal goteo de manera remota.
- No controla las partes externas del equipo (mangueras).
- No está a su alcance recargar el producto por sí solo.

1.8 APORTES

1.8.1. PRÁCTICO

El control automatizado de los dispensadores de odorización es capaz de dar una alerta temprana garantizando un buen control constante para el funcionamiento de los dispensadores de odorización de manera remota, de esta manera ayuda a ejercer una buena supervisión. Implementando Arduino, sensores ultrasónicos, GPS, y SMS.

1.8.2 TEÓRICO

Con el desarrollo de la metodología en V aplicada al desarrollo del hardware del control de los dispensadores de odorización y basándonos en los pasos de la metodología, la cual consiste en:

- **La corriente de especificación**, donde podemos analizar los conceptos de operaciones para ver qué debe hacer el sistema a grandes rasgos, los requisitos del sistema y arquitectura del mismo. Y el diseño detallado.

- **La corriente de pruebas**, donde se integran las distintas partes, la prueba y verificación de las mismas. La verificación y validación del sistema en conjunto y el mantenimiento del sistema.
- **La corriente de desarrollo** puede consistir en personalización, configuración o codificación.
- Gracias a la metodología en V se desarrolla el hardware basado en Arduino del presente trabajo.

1.9 METODOLOGÍA

1.9.1 METODOLOGÍA CIENTÍFICA

Para el desarrollo de la investigación se implementara la metodología de investigación científica y deductiva planteada por Mario Bunge identificamos las siguientes etapas:

1. **Observación:** Se realizara un análisis de los equipos de odorización en el cual analizaremos el funcionamiento mecánico actual para poder implementar el nuevo sistema de control.
2. **Planteamiento de la hipótesis:** Se propone implementar Arduino para lograr la automatización de los equipos de odorización.
3. **Diseño de la aplicación:** En esta etapa se hará la estructuración de la metodología para guiar la construcción del sistema basado en Arduino.
4. **Casos de prueba:** Se basara en los postulados teóricos de la ingeniería del software, para tal verificación se aplicara las pruebas de caja negra.
5. **Conclusiones:** En esta etapa final se mencionaran los resultados obtenidos y recomendaciones para nuevos temas de investigación, en base al conocimiento obtenido en el trabajo actual.

1.9.2 METODOLOGÍA DE DESARROLLO DEL SOFTWARE LIBRE Y HADWARE

Para el desarrollo del prototipo se implementó el modelo en v. El cual se desarrolló para terminar con algunos de los problemas que se vieron utilizando el enfoque de cascada tradicional.

Los defectos estaban siendo encontrados demasiado tarde en el ciclo de vida, ya que las pruebas no se introducían hasta el final del proyecto.

El modelo en v dice que las pruebas necesitan empezarse lo más pronto posible en el ciclo de vida. También muestra que las pruebas no son sólo una actividad basada en la ejecución. Hay una variedad de actividades que se han de realizar antes del fin de la fase de codificación. Estas actividades deberían ser llevadas a cabo en paralelo con las actividades de desarrollo, y los técnicos de pruebas necesitan trabajar con los desarrolladores y analistas de negocio de tal forma que puedan realizar estas actividades y tareas y producir una serie de entregables de pruebas.

Los productos de trabajo generados por los desarrolladores y analistas de negocio durante el desarrollo son las bases de las pruebas en uno o más niveles. El modelo en v es un modelo que ilustra cómo las actividades de prueba (verificación y validación) se pueden integrar en cada fase del ciclo de vida. Dentro del modelo en v, las pruebas de validación tienen lugar especialmente durante las etapas tempranas, por ejemplo, revisando los requisitos de usuario y después por ejemplo, durante las pruebas de aceptación de usuario. El modelo en v es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Describe las actividades y resultados que han de ser producidos durante el desarrollo del producto. La parte izquierda de la v representa la descomposición de los requisitos y la creación de las especificaciones del sistema. El lado derecho de la v representa la integración de partes y su verificación. V significa “Validación y Verificación” [INTECO, 2009].

CAPITULO II

MARCO TEÓRICO

2.1 SOFTWARE LIBRE

El software libre respeta la libertad de los usuarios y la comunidad. En grandes líneas significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir el software libre es una cuestión de libertad, no de precio [Free Software Foundation, 2014].

Ventajas

- Ahorro de costos.
- Adaptaciones a la medida del propósito.
- Total independencia de un proveedor.
- Facilidad para personalizar el software de acuerdo a las necesidades del usuario.

Desventajas

- Interfaces graficas poco amigables.
- No se brinda soporte y buena documentación [Richard Stallman, 2004].



Figura 2.1 Definición gráfica del software libre
Fuente: Stallman, 2004

2.1.1 SOFTWARE LIBRE EN BOLIVIA

2.1.1.1 DECRETO SUPREMO N° 1793 DE 13 DE NOVIEMBRE DE 2013, REGLAMENTO PARA EL DESARROLLO DE LAS TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN.

El Estado Boliviano promueve el uso del Software Libre como política de gobierno bajo los principios de soberanía y autonomía tecnológica, así como un ahorro de recursos públicos al prescindir de licencias de software propietario.

Bolivia: Decreto Supremo N° 1793, 13 de noviembre de 2013

Decreto Supremo N° 1793

EVO MORALES AYMA

**PRESIDENTE CONSTITUCIONAL DEL ESTADO PLURINACIONAL DE
BOLIVIA**

CONSIDERANDO:

- Que el Parágrafo II del Artículo 103 de la Constitución Política del Estado, determina que el Estado asumirá como política la implementación de estrategias para incorporar el conocimiento y aplicación de nuevas tecnologías de información y comunicación.
- Que el numeral 1 del Parágrafo I del Artículo 85 de la Ley N° 031, de 19 de julio de 2010, Marco de Autonomías y Descentralización “Andrés Ibáñez”, establece formular y aprobar el régimen general y las políticas de comunicaciones y telecomunicaciones del país, incluyendo las frecuencias electromagnéticas, los servicios de telefonía fija y móvil, radiodifusión, acceso a internet y demás

Tecnologías de Información y Comunicaciones - TIC, es una competencia exclusiva del nivel central del Estado.

- Que los numerales 2 y 5 del Artículo 2 de la Ley N° 164, de 8 de agosto de 2011, General de Telecomunicaciones, Tecnologías de Información y Comunicación, disponen como objetivos asegurar el ejercicio del derecho al acceso universal y equitativo a los servicios de telecomunicaciones, tecnologías de información y comunicación; y promover el uso de las tecnologías de información y comunicación para mejorar las condiciones de vida de las bolivianas y bolivianos.
- Que el Artículo 71 de la Ley N° 164, declara de prioridad nacional la promoción del uso de las tecnologías de información y comunicación para procurar el vivir bien de todas las bolivianas y bolivianos.
- Que la Disposición Transitoria Sexta de la Ley N° 164, señala que todos los aspectos que se requieran para la aplicación de la citada Ley serán reglamentados por el Órgano Ejecutivo y regulados por la Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes.
- Que la Disposición Transitoria Novena de la Ley N° 164, establece que la Autoridad de Fiscalización y Control Social de Telecomunicaciones y Transportes - ATT se denominará en adelante Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes - ATT y asumirá las atribuciones, competencias, derechos y obligaciones en materia de telecomunicaciones y tecnologías de la información y comunicación, transportes y del servicio postal, bajo tuición del Ministerio de Obras Públicas, Servicios y Vivienda.
- Que las TIC se han convertido en medios esenciales para el desarrollo social, cultural, económico y político de los pueblos. En virtud a lo cual, el Órgano Ejecutivo ha procedido a la concertación para el Reglamento a la Ley N° 164 para el Desarrollo de Tecnologías de Información y Comunicación, en cuyo proceso se recolectaron aportes y propuestas para el presente Decreto Supremo.

EN CONSEJO DE MINISTROS, DECRETA:

Artículo Único.-

- I. Se aprueba el Reglamento a la Ley N° 164, de 8 de agosto de 2011, para el Desarrollo de Tecnologías de Información y Comunicación, que en Anexo forma parte integrante del presente Decreto Supremo.
- II. Todos los aspectos complementarios referidos a la firma y certificado digital así como de las entidades certificadoras, que se requieran para la aplicación del presente Reglamento serán establecidos mediante Resolución Ministerial por el Ministerio de Obras Públicas, Servicios y Vivienda.

Disposiciones transitorias

Artículo transitorio 1°.- La Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes - ATT, en un plazo no mayor a:

- a. Ocho (8) meses a partir de la publicación del presente Decreto Supremo, elaborará y aprobará los estándares técnicos y otros lineamientos establecidos para el funcionamiento de las entidades certificadoras;
- b. Seis (6) meses a partir de la aprobación de los estándares técnicos, deberá implementar la infraestructura y procedimientos internos necesarios para la aplicación de la firma y certificación digital como Certificadora Raíz.

Artículo transitorio 2°.- La Agencia para el Desarrollo de la Sociedad de la Información en Bolivia - ADSIB, en un plazo no mayor a:

- a. Seis (6) meses a partir de la aprobación de los estándares técnicos, deberá implementar la infraestructura y procedimientos internos necesarios para la aplicación de la firma y certificación digital como Entidad Certificadora Pública;

- b. Seis (6) meses a partir de la publicación del presente Decreto Supremo, deberá implementar el Repositorio Estatal de Software Libre para las aplicaciones desarrolladas en el sector estatal.

Artículo transitorio 3°.- Todas las entidades públicas en todos los niveles en un plazo no mayor a:

- a. Seis (6) meses a partir de la implementación del Repositorio Estatal de Software Libre, tienen la obligación de registrar las aplicaciones desarrolladas de manera directa o a través de terceros en el repositorio, previa evaluación y validación a cargo de la ADSIB;
- b. Seis (6) meses desde la aprobación del plan de implementación de software libre, iniciarán la migración de sus sistemas informáticos a Software Libre y Estándares Abiertos.

Artículo transitorio 4°.- En un plazo no mayor a dieciocho (18) meses a partir de la publicación del presente Decreto Supremo, el Ministerio de Planificación del Desarrollo, en coordinación con el Ministerio de Obras Públicas, Servicios y Vivienda a través del Viceministerio de Telecomunicaciones y la ADSIB, elaborará el Plan de Implementación del Gobierno Electrónico y el Plan de Implementación de Software Libre y Estándares Abiertos que serán aprobados mediante Decreto Supremo.

Artículo transitorio 5°.- El plazo máximo para la migración de los sistemas de las entidades públicas a Software Libre y Estándares Abiertos será de siete (7) años desde el inicio de la migración.

Artículo transitorio 6°.- Lo establecido en el Artículo 23 del Reglamento para el Desarrollo de Tecnologías de Información y Comunicación, se hará efectivo a partir de la aprobación del Plan de Implementación de Software Libre y Estándares Abiertos.

Disposiciones abrogatorias y derogatorias

Se abrogan y derogan todas las disposiciones contrarias al presente Decreto Supremo.

Disposiciones finales

Artículo final 1°.- Se modifica el segundo párrafo del Artículo 101 del Reglamento a la Ley General de Aduanas, aprobado por Decreto Supremo N° 25870, de 11 de agosto de 2000, modificado por el Parágrafo II del Decreto Supremo N° 0784, de 2 de febrero de 2011, con el siguiente texto:

“La Aduana Nacional a través de Resolución de Directorio reglamentará el uso de la firma digital en la suscripción y presentación de la declaración de mercancías u otros documentos.”

Artículo final 2°.- En tanto se establezca la ADSIB como entidad Certificadora, las entidades públicas podrán optar por una Certificadora Extranjera para el uso de los servicios de Certificación digital [Decreto Supremo de Bolivia N° 1793, 2013].

2.2 HARDWARE LIBRE

Se llama hardware libre a los dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público.

El hardware libre toma las mismas ideas del software libre de acuerdo a las cuatro libertades:

1. Libertad de **usar** el programa, con cualquier propósito (Uso).
2. Libertad de **estudiar** cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (Estudio).
3. Libertad de **distribuir** copias del programa, con lo cual se puede ayudar a otros usuarios (Distribución).

4. Libertad de **mejorar** el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (Mejora).

Se debe recordar en todo momento que libre no es sinónimo de gratis. El hardware libre forma parte de la cultura libre.

Dado que el hardware tiene asociados a él costos variables directos, ninguna definición de software libre se puede aplicar directamente sin modificación. En cambio, el término hardware libre se ha usado principalmente para reflejar el uso del software libre con el hardware y el lanzamiento libre de la información con respecto al hardware, a menudo incluyendo el lanzamiento de los diagramas esquemáticos, diseños, tamaños y otra información acerca del hardware. De todos modos, incluye el diseño del hardware y la distribución de los elementos en la tarjeta madre.

Existen muchas comunidades que trabajan en el diseño, desarrollo y pruebas de hardware libre, además de brindan soporte. [Herrador, 2009].

2.3 MÉTODO EN V

El modelo en v se desarrolló con el objetivo de optimizar el método de la cascada, en el método en cascada, los defectos estaban siendo encontrados muy tarde en el ciclo de vida, ya que las pruebas se introducían al finalizar el proyecto. En el método en v las pruebas empiezan lo más pronto posible, es por esto que V significa “Validación y Verificación”.

El método en V, define un procedimiento uniforme para el desarrollo de sistemas, que tiene una serie de pasos en el ciclo de vida de un proyecto.

En la cual se describen las actividades y resultados que deben producirse durante el desarrollo del producto.

En el lado izquierdo de la V se descompone las necesidades y la creación de las especificaciones del producto.

El lado derecho de la V representa la verificación de las piezas y su integración [Blum, 2007].

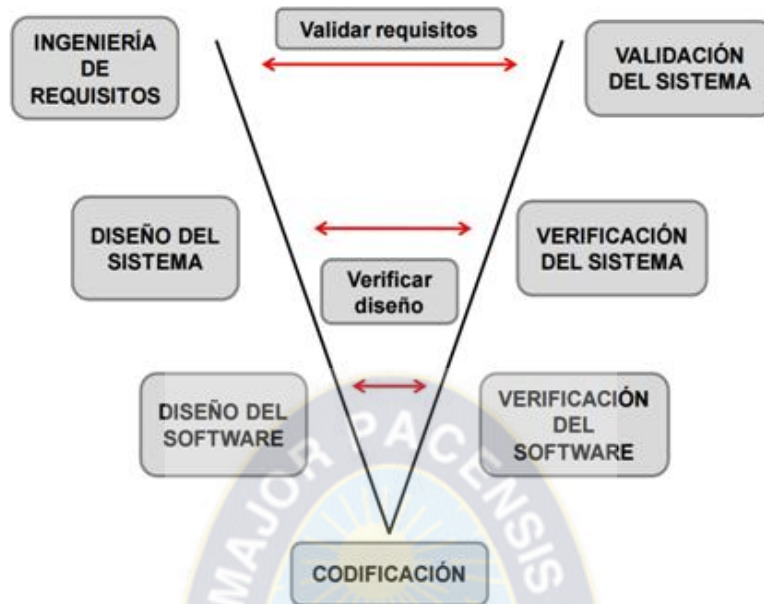


Figura 2.2 Metodología en V
Fuente: Blum, 2007

2.3.1 NIVELES DE LA METODOLOGÍA EN V

La metodología en V consta de 4 etapas, en las cuales las fases se enlazan es decir para cada fase existe una fase correspondiente o paralela de verificación o validación. Esta estructura obedece al principio de que para cada fase del desarrollo debe existir un resultado verificable.

En la misma estructura se advierte también que la proximidad entre una fase del desarrollo y su fase de verificación correspondiente va decreciendo a medida que aumenta el nivel dentro de la V. La longitud de esta separación intenta ser proporcional a la distancia en el tiempo entre una fase y su homóloga de verificación.

Nivel 1 Análisis de requerimientos y especificaciones

Orientado al “cliente”. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones.

Nivel 2 Análisis funcional

Se enfoca a las características funcionales del sistema propuesto. Puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.

Nivel 3 Arquitectura del sistema

Define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.

Nivel 4 Implementación

Es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa [Blum, 2007].

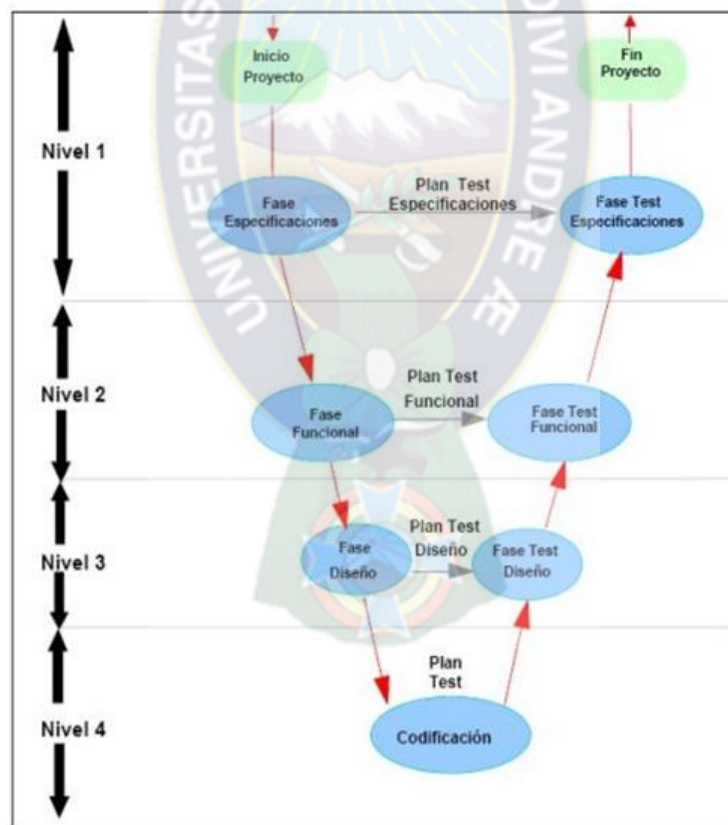


Figura 2.3 Niveles de la metodología en V
Fuente: Blum, 2007

2.4 DISPENSADORES DE ODORIZACIÓN

El dispensador de odorización, por goteo, es un equipo, que se instala en los baños, con el objetivo de desinfectar o dejar libre de bacterias a los inodoros, y de esta manera mantener los baños limpios. El producto gotea de manera constante a través de una manguera que llega al inodoro y posteriormente a las tuberías.



Figura 2.4 Dispensador de Odorización
Fuente: Odoritec, 2013

El dispensador de odorización está compuesto por:

Una base.- La base es atornillada a la pared y es la encargada de sujetar el equipo.

Una botella.- En la botella se llena el producto, odorizante o digestante biológico.

Una tasa.- Es un recipiente en el cual existe un orificio

Una mecha.- (tejido de algodón) la cual se impregna del producto y se la instala en la tasa, para que posteriormente, el producto pase a la manguera, actuando como un conductor.

Una Manguera La manguera sirve para que el producto llegue al inodoro.

Una tapa.- La cual cubre el dispensador y ayuda a sostenerlo y protegerlo [Open Clean, 2013].



Figura 2.5 Partes del dispensador de Odorización
Fuente: Aromatiza, 2014

2.5 ARDUINO

2.5.1 QUÉ ES ARDUINO?

Arduino es una plataforma de electrónica abierta basada en el hardware y software libre. Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador de la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software [Herrador, 2009].

2.5.2 SOFTWARE PARA ARDUINO

El entorno de código abierto Arduino hace fácil escribir el código y cargarlo a la placa de E/S. Funciona en Windows, Mac OS X, Linux y Androide. El entorno está escrito en Java y basado en Processing, avr-gcc y otros programas también de código abierto serie [Anderson y Cervo, 2013].

2.5.2.1 ENTORNO ARDUINO

El entorno Arduino también es conocido como IDE para Arduino el cual permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos, constituido por:

- Editor de textos para escribir el código
- Área de mensajes,
- Consola de textos,
- Barra de herramientas
- Botones para las funciones comunes
- Serie de menús


Arduino utiliza para escribir el software lo que se denomina sketch. Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar, pegar y buscar/reemplazar texto. En el área de mensajes se muestra información mientras se carga el programa y también muestra errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie [Anderson y Cervo, 2013].


En el caso de Arduino, necesitamos un IDE que nos permita escribir y editar nuestro programa, que nos permita comprobar que no hayamos cometido ningún error y que además nos permita, cuando ya estemos seguros de que el sketch es correcto, grabarlo en la memoria del microcontrolador de la placa Arduino para que este se convierta a partir de entonces en el ejecutor autónomo de dicho programa [Torrente, 2013].




Figura 2.6 IDE para Arduino
Fuente: Torrente, 2013

a. Barra de Botones

 **Verify:** este botón realiza dos cosas: comprueba que no haya ningún error en el código de nuestro sketch, y si el código es correcto, entonces lo compila. Este es el primer botón que deberemos pulsar cada vez que deseemos probar cualquier modificación que hagamos en nuestro sketch. (Si no sabes lo que significa “compilar”, no te preocupes: por ahora tan solo hay que saber que es un paso imprescindible).

 **Upload:** este botón deberemos pulsarlo inmediatamente después del botón “Verify”. Su función es invocar internamente al comando “avrdude” para cargar en la memoria del microcontrolador de la placa Arduino el sketch recientemente verificado y compilado. “Avrdude” es capaz de realizar su tarea porque al hacer clic en este botón se activa automáticamente el bootloader del microcontrolador (el “auto-reset”), así que raras veces será necesario utilizar el método “tradicional” de activar el bootloader mediante la pulsación física del botón de reinicio que viene en la placa.

 **New:** crea un nuevo sketch vacío.



Open: presenta un menú con todos los sketches disponibles para abrir. Podremos abrir tanto nuestros propios sketches como gran cantidad de sketches de ejemplo listos para probar, clasificados por categorías dentro del menú. Estos sketches son muy útiles para aprender; de hecho, en este



Save: guarda el código de nuestro sketch en un fichero, el cual tendrá la extensión .ino, podemos guardar estos ficheros donde queramos, pero el IDE Arduino nos ofrece una carpeta específica para ello, la carpeta “sketchbook” [Torrente, 2013].

b. Barra de Menú

Archivo: El menú archivo nos ofrece una serie de acciones como crear un nuevo sketch, abrir uno existente, guardarlo, cerrarlo y cerrar el IDE.

También nos permite acceder a los sketches de ejemplo que vienen de serie con el IDE y gracias a la entrada “Sketchbook” podemos acceder a nuestros propios sketches guardados en las diferentes subcarpetas que hay dentro de la carpeta “sketchbook”.

Editar: El menú editar nos ofrece una serie de acciones como deshacer y rehacer, cortar, copiar y pegar texto, seleccionar todo el texto o buscar y reemplazar texto

Programa: En este menú se ofrece la acción de verificar/compilar nuestro sketch, y también podemos abrir la carpeta donde está guardado el fichero .ino que se esté editando.

Herramientas: en este menú se ofrecen diferentes herramientas variadas, como la posibilidad de auto formatear el código, la posibilidad de guardar una copia de todos los sketches del proyecto actual en formato zip, la posibilidad de abrir el monitor serie.

Ayuda: desde este menú podemos acceder a varias secciones de la página web oficial de Arduino que contienen diferentes artículos, tutoriales y ejemplos de ayuda. No se necesita Internet para consultar dichas secciones ya que esta documentación se descarga junto con el propio IDE, por lo que su acceso se realiza en local.

c. Área de Mensajes

La barra y la consola de mensajes informan en el momento de la compilación de los posibles errores cometidos en la escritura de nuestro sketch, además de indicar el estado en tiempo real de diferentes procesos.

2.5.3 BENEFICIOS DE ARDUINO

- **Económico.-** En relación a otros microcontroladores las placas Arduino son relativamente económicas.
- **Multiplataforma.-** El software de Arduino se ejecuta en los diferentes sistemas operativos Windows, Macintosh OSX y GNU/Linux. A diferencia de otros microcontroladores que están limitados para Windows.
- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero [Herrador, 2009].

2.5.4 ARDUINO UNO

Se basa en un microcontrolador ATMEGA328. Este microcontrolador dota a este modelo de 32 Kb de memoria para almacenar programas, además cuenta con 14 pines digitales de

los cuales los numerados como 0 y 1 se pueden utilizar como bus de comunicación para interactuar con otros dispositivos. Los pines suministran una corriente muy pequeña limitados a 50 mA, suficiente para iluminar un led y para alimentar circuitos digitales, también incorporan una resistencia interna que puede usarse a la hora de implementar botones. De los 14 pines digitales existen algunos (marcados por ~) que pueden ser usados para hacer una modulación por ancho de banda. Cuenta también con un conector USB para su programación y con una entrada de alimentación.

2.5.4.1 PLACA ARDUINO Y SUS PARTES

1. Conector USB para el cable Tipo AB.
2. Pulsador de Reset.
3. Pines de E/S digitales y PWM.
4. LED verde de placa encendida.
5. LED naranja conectado al pin13.
6. ATmega 16U2 encargado de la comunicación con el PC.
7. LED TX (Transmisor) y RX (Receptor) de la comunicación serial.
8. Puerto ICSP para programación serial Microcontrolador.
9. ATmega 328, cerebro del Arduino.
10. Cristal de cuarzo de 16Mhz.
11. Regulador de voltaje.
12. Conector hembra 2.1mm con centro positivo.
13. Pines de voltaje y tierra.
14. Entradas análogas.

[Herrador, 2009].

2.5.4.2 ENTRADAS Y SALIDAS DE ARDUINO UNO

Cada uno de los 14 pines digitales puede utilizarse como entradas o salidas, dependiendo de las `pinMode()`, `digitalWrite()` y `digitalRead()` que se vayan a programar.

Las E/S operan a 5 V, cada pin puede proporcionar o recibir una intensidad máxima de 40 mA y tienen una resistencia interna, de 20 K Ω .

Además, algunos pines tienen funciones especializadas:

- **Serie: pin 0 (RX) y pin 1 (TX):** Usado para recibir (RX) y transmitir (TX) datos a través del puerto serie TTL. Estos pines están conectados en paralelo a los pines correspondientes del Atmega8U2 y a los pines RXD y TXD del Atmega.
- **Interrupciones Externas:** Pin 2 y pin 3. Estos pines se pueden configurar para lanzar una interrupción en un valor LOW (0V), en flancos de subida o bajada (cambio de LOW a HIGH o viceversa), o en cambios de valor.
- **PWM:** Pines 3, 5, 6, 9, 10 y 11. Proporciona una salida PWM modulación por onda de pulso de 8 bits de resolución con valores de 0 a 255. Se los identifica por el símbolo ~ en la placa Arduino.
- **SPI:** Pines 10 (SS), 11 (MOSI), 12 (MISO) y 13 (SCK). Estos pines proporcionan comunicación SPI, que a pesar de que el hardware la proporcione actualmente no está incluido en el lenguaje Arduino.
- **LED:** Pin 13. Hay un led integrado en la placa conectado al pin digital 13, cuando este pin tiene un valor HIGH (5V) el led se enciende y cuando éste tiene un valor LOW (0V) este se apaga.
- La Uno tiene 6 entradas analógicas y cada una de ellas proporciona una resolución de 10 bits (1024 valores). Por defecto se mide de tierra a 5 V, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función `analogReference()`. Además algunos pines tienen funciones especializadas:
- **I²C:** Pin 4 (SDA) y pin 5 (SCL). Soporte del protocolo de comunicaciones I²C (TWI) usando la librería Wire.

- **AREF:** Voltaje de referencia para las entradas analógicas. Configura el voltaje de referencia usado por la entrada analógica. La función analogRead() devolverá un valor de 1023 para aquella tensión de entrada que sea igual a la tensión de referencia. El valor del voltaje debe estar en el rango de 0 a 5 V. Reset. Suministra un valor LOW (0V) para reiniciar el microcontrolador. Típicamente usado para añadir un botón de reset a los Shields que no permiten acceso a la placa. [Schmidt, 2011].

2.5.4.3 UNIVERSAL SERIAL BUS USB

El Universal Serial Bus, abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a un ordenador. Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- **Baja velocidad (1.0):** Tasa de transferencia de hasta 1,5Mbps (192kB/s). Utilizado en su mayor parte por dispositivos de interfaz humana.
- **Velocidad completa (1.1):** Tasa de transferencia de hasta 12Mbps según este estándar, pero se dice en fuentes independientes que habría que realizar nuevamente las mediciones. Ésta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de impedancias LIFO (Last In FirstOut - último en entrar, primero en salir).
- **Alta velocidad (2.0):** Tasa de transferencia de hasta 480Mbps (60MB/s) pero por lo general de hasta 125Mbps. Está presente casi en el 99% de los PC actuales. El cable USB 2.0 dispone de cuatro líneas, un par para datos, una de corriente y un cuarto que es el negativo o retorno.
- **Súper alta velocidad (3.0):** Tiene una tasa de transferencia de hasta 4.8Gbps.
- La velocidad del bus es diez veces más rápida que la del USB 2.0, debido a que han incluido 5 conectores extra, desechando el conector de fibra óptica propuesto

inicialmente, y será compatible con los estándares anteriores, usa un cable de 9 hilos [Schmidt, 2011].

2.5.4.4 COMUNICACIÓN

- El Arduino Uno facilita en varios aspectos la comunicación con el ordenador, otro Arduino u otros microcontroladores.
- El ATmega328 proporciona comunicación vía serie UART TTL (5V), disponible a través de los pines digitales 0(RX) y 1(TX).
- Un microcontrolador ATmega8U2 integrado en la placa que canaliza la comunicación serie a través del USB y proporcionan un puerto serie virtual en el ordenador.
- El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino.
- Los LEDs RX y TX de la placa parpadearán cuando se detecte comunicación transmitida a través del Atmega8U2 y la conexión USB.
- La librería Software Serial permite comunicación serie por cualquier par de pines digitales del Uno.
- El ATmega328 también soporta la comunicación I²C (TWI) y SPI. El software de Arduino incluye una librería Wire para simplificar el uso el bus I²C [Herrador, 2009].

2.5.4.5 PROTECCIÓN CONTRA SOBRE TENSIONES EN EL USB

El Arduino Uno tiene un multifusible reiniciable que protege la conexión USB del ordenador de cortocircuitos y sobre tensiones. A parte que la mayoría de ordenadores proporcionan su propia protección interna, el fusible proporciona una capa extra de protección. Si más de 500 mA son detectados en el puerto USB, el fusible automáticamente corta la conexión hasta que el cortocircuito o la sobre tensión desaparece [Anderson y Cervo, 2013].

2.5.4.6 SHIELDS ARDUINO

Un shield es una placa impresa que se pueden conectar en la parte superior de la placa Arduino para ampliar sus capacidades, pudiendo ser apilada una encima de la otra.

Las shields suelen ser diseños bastante simples y en general de código abierto y publicados libremente [Anderson y Cervo, 2013].

a. Shields Arduino Sim900

Es un módulo de 4 bandas (850, 900, 1800, 1900 Mhz), para el desarrollo de proyectos de comunicación y control remoto inalámbrico con microcontroladores, por vía red celular e internet, ya sea utilizando mensajes SMS o enlaces por vía GPRS.

El sim900 es el circuito base del minicore, e integra en su firmware el stack TCP-IP de tal manera es posible la comunicación GPRS entre un sistema microcontrolador con un servidor de internet.

El sim900 consta de las siguientes partes:

- Receptor estándar para chip SIM de cualquier operador de red de telefonía celular.
- Un conector para entrada de alimentación externa
- Un circuito MAX232 con la interfaz RS232 completa
- Antena GSM.



Figura 2.7 Sim900
Fuente: Corona, 2015

2.6 SENSORES

Un sensor es un dispositivo capaz de detectar acciones o estímulos externos y responder a los mismos. Los sensores pueden transformar las magnitudes físicas o químicas, en variables eléctricas.

Los sensores siempre que estén activados estarán tomando continuamente la situación actual de una habitación y es el servidor o la placa Arduino quien leerá esta información y decidirá la manera de actuar de acuerdo a la programación. Capturando los datos que nosotros deseemos.

De acuerdo al software los sensores tienen que ser inicializados como pin de salida con el método `pinMode` (número de pin, OUTPUT), para poder obtener una lectura de los datos se utiliza el método `digitalRead` (número de pin, INPUT).

2.6.1 SENSOR ULTRASÓNICO

Los sensores ultrasónicos son capaces de detectar la proximidad, que trabajan libres de roces mecánicos, los cuales detectan objetos, a distancias que van desde pocos centímetros hasta varios metros.

El sensor emite un sonido y mide el tiempo que la señal tarda en regresar. Estos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas.

Estos sensores trabajan solamente en el aire, y pueden detectar objetos con diferentes formas, colores, superficies y materiales sólidos, líquidos o polvorientos.

Los sensores trabajan según el tiempo de transcurso del eco, es decir, la distancia temporal entre el impulso de emisión y el impulso del eco. Este tipo de sensor emite una señal, de tipo ultrasónica, y esta señal es recibida por un receptor [Corona, 2015].



Figura 2.8 Sensor ultrasónico
Fuente: Corona, 2015

2.7 PLACA DE PRUEBAS O PROTOBOARD

Es un tablero con orificios los cuales se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipo de circuitos electrónicos y sistemas similares.

Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí. Uno de sus usos principales es la creación y comprobación de prototipos de circuitos electrónicos antes de llegar a la impresión mecánica del circuito en sistemas de producción comercial.

La placa de pruebas evita tener que soldar u otro método para realizar la unión física y eléctrica entre los distintos elementos que componen un proyecto. Esta placa tiene una serie de conexiones internas de forma que todos los elementos conectados en la misma serie quedan conectados eléctricamente y se puede avanzar muy rápido en el montaje de los experimentos [Schmidt, 2011].



Figura 2.9 Protoboard
Fuente: Torrente, 2013

Para poder conectar correctamente nuestros componentes al protoboard, hemos de conocer primero cómo se estructuran sus propias conexiones internas. En este sentido, si observáramos su interior oculto bajo la superficie perforada, podríamos comprobar que está compuesto de muchas tiras de metal, normalmente cobre, dispuestas de la siguiente manera:

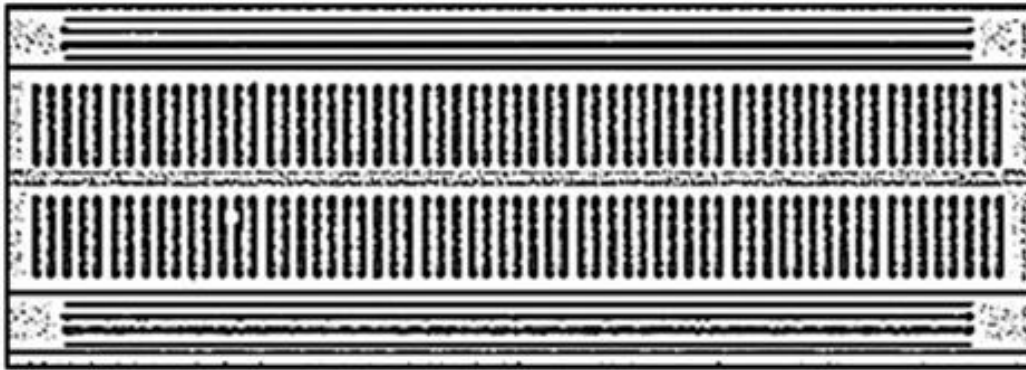


Figura 2.10 Zonas del protoboard
Fuente: Torrente, 2013

2.7.1 ZONAS DEL PROTOBOARD

Buses: Los buses se localizan en uno o ambos lados del protoboard. Allí se conectarán (en cualquiera de sus puntos) las fuentes de alimentación externas. Normalmente aparece pintada una línea roja que se suele utilizar para indicar el bus sometido al voltaje de entrada (es decir, donde insertaremos el borne positivo de la fuente) y una línea azul que representa el bus conectado a tierra (es decir, donde normalmente insertaremos el borne negativo). Todos los puntos del bus marcado con la línea roja son equivalentes porque están conectados entre sí y todos los puntos del bus marcado con la línea azul también lo son entre sí, pero ambos buses están aislados eléctricamente uno del otro.

Nodos: en la parte central del protoboard aparecen gran cantidad de agujeros. Su cantidad puede ser mayor o menor dependiendo del modelo (de hecho, el tamaño de la breadboard se indica por el número de filas y de columnas de agujeros que contiene: un tamaño típico es 10x64). Estos agujeros se usan para colocar los componentes y realizar las conexiones entre ellos. Tal como se puede observar en la figura anterior, las conexiones internas entre los agujeros están dispuestas en vertical. Lo más importante es comprender que cualquier agujero es completamente equivalente a otro que pertenezca a la misma conexión interna. Esto significa que al insertar una patilla de algún componente en un agujero, disponemos del resto de agujeros de su misma conexión interna para poder insertar en ellos una patilla

de cualquier otro componente que queramos poner en contacto entre sí, tal como si los uniéramos directamente por un cable. A todos esos agujeros equivalentes conectados entre sí se les da el nombre en conjunto de “nodo”. La manera más habitual de conectar dos nodos diferentes es enchufando los extremos de un cable en un agujero de cada nodo a unir.

Canal central: es la región localizada en el medio del protoboard, que separa la zona superior de la inferior. Se suele utilizar para colocar los circuitos integrados (esos componentes con forma de “cucarachas negras con patitas” también llamados “chips” o IC –del inglés “integrated circuits” –) de manera que pongamos la mitad de patitas en un lado del canal y la otra mitad en el otro lado. De esta manera, además de disponer así de varios agujeros de conexión por cada patita, una mitad del chip estará aislada eléctricamente de la otra (tal como debe ser) [Torrente, 2013].

2.8 INTERNET DE LAS COSAS

Este concepto se refiere a la interconexión digital de objetos cotidianos con internet. Si los objetos de la vida cotidiana tuvieran incorporadas etiquetas de radio, podrían ser identificados y gestionados por otros equipos, de la misma manera que si lo fuesen por seres humanos. Esta tecnología nos permite medir parámetros externos como ser temperatura, energía, actividad, luz, humedad entre otros, de forma automática y que esos datos viajen a un centro de procesamiento para que se tomen las decisiones adecuadas en tiempo real.

El concepto de internet de las cosas lo propuso Kevin Ashton en el Instituto de Tecnología de Massachusetts (MIT) en el año 1999, Auto-ID, donde se realizaban investigaciones en el campo de la identificación por radiofrecuencia en red RFID y tecnologías de sensores. Gracias a esta tecnología en los próximos años, bastará con integrar un chip de pocos milímetros en cualquier objeto del hogar, del trabajo o de la ciudad para poder procesar y transmitir información con el beneficio de que las señales de este tipo de tecnología RFID no necesitan Wi-Fi ni Bluetooth.

Por supuesto esta es la revolución, en las relaciones entre los objetos y las personas, incluso entre los objetos directamente, ofreciendo la posibilidad de obtener datos en tiempo real y permitiéndonos acercarnos a la digitalización del mundo físico.

Actualmente, el término internet de las cosas se usa con una denotación de conexión avanzada de dispositivos, sistemas y servicios que va más allá del tradicional “máquina a máquina” que cubre una amplia variedad de protocolos, dominios y aplicaciones.

Siendo así Arduino una de las tecnologías que coadyuva y satisface las necesidades del internet de las cosas, tales como: sensores para el hogar, circuitos con control y gestión de cámaras de video vigilancia o incluso termostatos. Y su futuro puede ser todavía más relevante en el mercado del Internet de las Cosas por su acercamiento a Windows 10 [Doukas, 2013].

2.9 SISTEMAS AUTOMÁTICOS Y DE CONTROL

Un sistema automático de control es un conjunto de elementos físicos relacionados entre sí, de tal forma que son capaces de gobernar su actuación por sí mismos, sin necesidad de la intervención de agentes externos, incluyendo al ser humano, anulando los posibles errores que puedan surgir a lo largo de su funcionamiento debido a perturbaciones no previstas.

Cualquier sistema automático está constituido por un sistema físico que realiza la acción, y un sistema de mando, que genera las órdenes precisas para que se ejecuten las acciones.

En los sistemas de regulación y control automáticos, se sustituye el componente humano por un mecanismo, circuito eléctrico, electrónico o, un ordenador. En este caso, el sistema de control sería automático.

2.9.1 NECESIDAD Y APLICACIONES DE LOS SISTEMAS AUTOMÁTICOS DE CONTROL

La implantación y el desarrollo de los sistemas de regulación están presentes en infinidad de sectores, en el ámbito doméstico, en los procesos industriales, en el desarrollo tecnológico y científicos, provocando avances significativos en todos los campos.

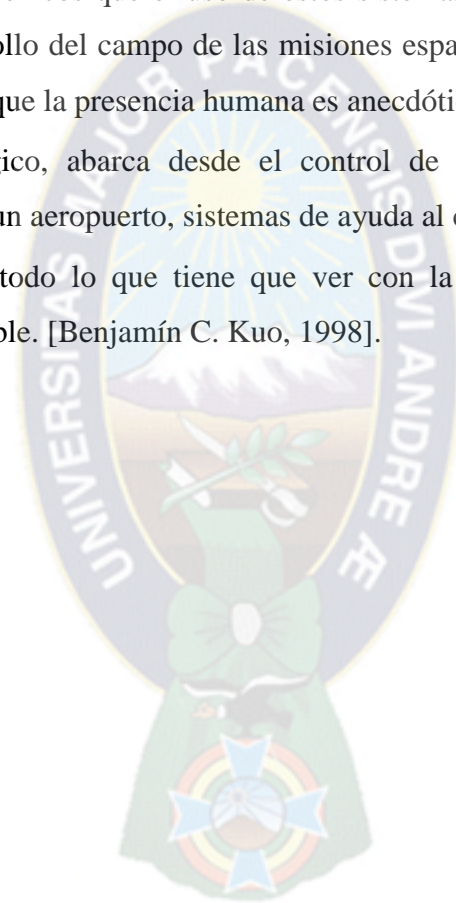
En la producción industrial su utilización permite:

- Aumentar la calidad y la cantidad del producto fabricado.
- Mejorar los sistemas de seguridad del proceso industrial.
- Ejecutar operaciones cuya realización sería impensable con la única participación del hombre.
- Reducir enormemente los costes productivos.

Dentro de los avances científicos que el uso de estos sistemas ha posibilitado tenemos un ejemplo claro en el desarrollo del campo de las misiones espaciales, que son realizadas de modo automático y en las que la presencia humana es anecdótica.

En el desarrollo tecnológico, abarca desde el control de robots, como la regulación centralizada del tráfico en un aeropuerto, sistemas de ayuda al conductor de un vehículo,...

En el ámbito doméstico, todo lo que tiene que ver con la domótica que provoca una habitabilidad más confortable. [Benjamín C. Kuo, 1998].



CAPITULO III

MARCO APLICATIVO

3.1 INTRODUCCIÓN

Para el diseño e implementación del control de dispensadores de Odorización basado en Arduino, y con el uso de los sensores ultrasónicos, Módulo sim900 y Arduino Uno se tomó en cuenta los siguientes parámetros:

1. Para desarrollar el prototipo en Arduino se hará la implementación de la metodología en V. Bajo los siguientes términos:
 - Ingeniería de requerimientos.
 - Requerimientos del usuario
 - Requerimientos del hardware
 - Requerimientos del software
 - Diseño del sistema.
 - Diseño del software.
 - Corriente de pruebas:
 - Validación del sistema.
 - Verificación del sistema.
 - Verificación del software.
 - Entrelazamiento de dispositivos
2. La utilización de un elemento electrónico, que permita medir la distancia del líquido en la botella del equipo de odorización.
3. Realizar experimentos reales a través de pruebas con monitoreo de equipos instalados.
4. Visualizar posteriormente los datos obtenidos por el Sensor ultrasónico, a través de un SMS enviado a otro teléfono celular, perteneciente a la empresa proveedora de los servicios de odorización.



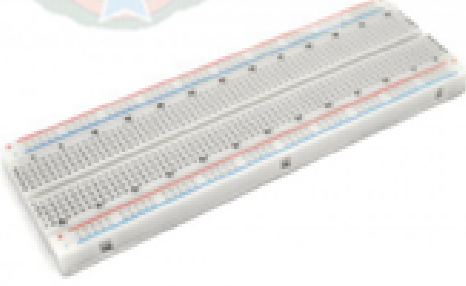
3.2 INGENIERÍA DE REQUERIMIENTOS

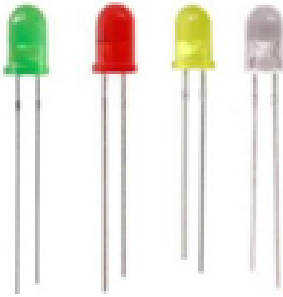




3.2.1 REQUERIMIENTOS DEL USUARIO:

- Mejorar el control de los dispensadores de odorización.
- Contar con un control constante.

3.2.2 REQUERIMIENTOS DEL HADWARE

Los elementos a usar en la construcción del control automatizado de los equipos de odorización se detallan a continuación en la tabla 3.1.

Cantidad	Elemento	Gráfico	Voltaje
1	Placa Arduino		5V- 500 mA
1	Sensor ultrasónico		4.5V-10 mA 5V- 15 mA
1	Protoboard		

3	Focos led		5V-400 mA
1	Cable de alimentación		5V-100 mA 9V-3300 mA
1	Módem sim900		3.2V-100 mA 4.8v- 50 mA
	Cables de conexión macho-macho		Transmisores de corriente eléctrica.
	Cables de conexión hembra-macho		Transmisores de corriente eléctrica.

	<p>Fuente de corriente para alimentar el prototipo</p>		<p>5V-5000mA</p>
--	---	--	------------------

Tabla 3.1 Elementos del prototipo
Fuente: Elaboración Propia

3.2.3 REQUERIMIENTOS DEL SOFTWARE

Se estableció en la ingeniería de requerimientos que el software a usar es Arduino o conocido también como IDE para Arduino. Utilizada para escribir el software, lo que se denomina sketch, ya que este software se puede emplear en todos los sistemas operativos, además de ser un software libre.

3.3 DISEÑO DEL SISTEMA

La estructura del control de dispensadores de Odorización basado en Arduino contempla como sus partes principales a los sensores ultrasónicos, Arduino uno, sim900 y dispensadores de odorización.

La placa Arduino UNO se conecta con el sensor ultrasónico, el sensor ultrasónico se instala en el dispensador de odorización.

La placa Arduino al ser el cerebro del prototipo, a su vez, se conecta con el módulo sim900, se instala un chip de cualquier empresa de telecomunicaciones en el módulo sim900 y a partir de este se manda un SMS al celular de la empresa que brinda el servicio. El diseño del modem sim900 se visualiza en la figura 3.1, que vemos a continuación.

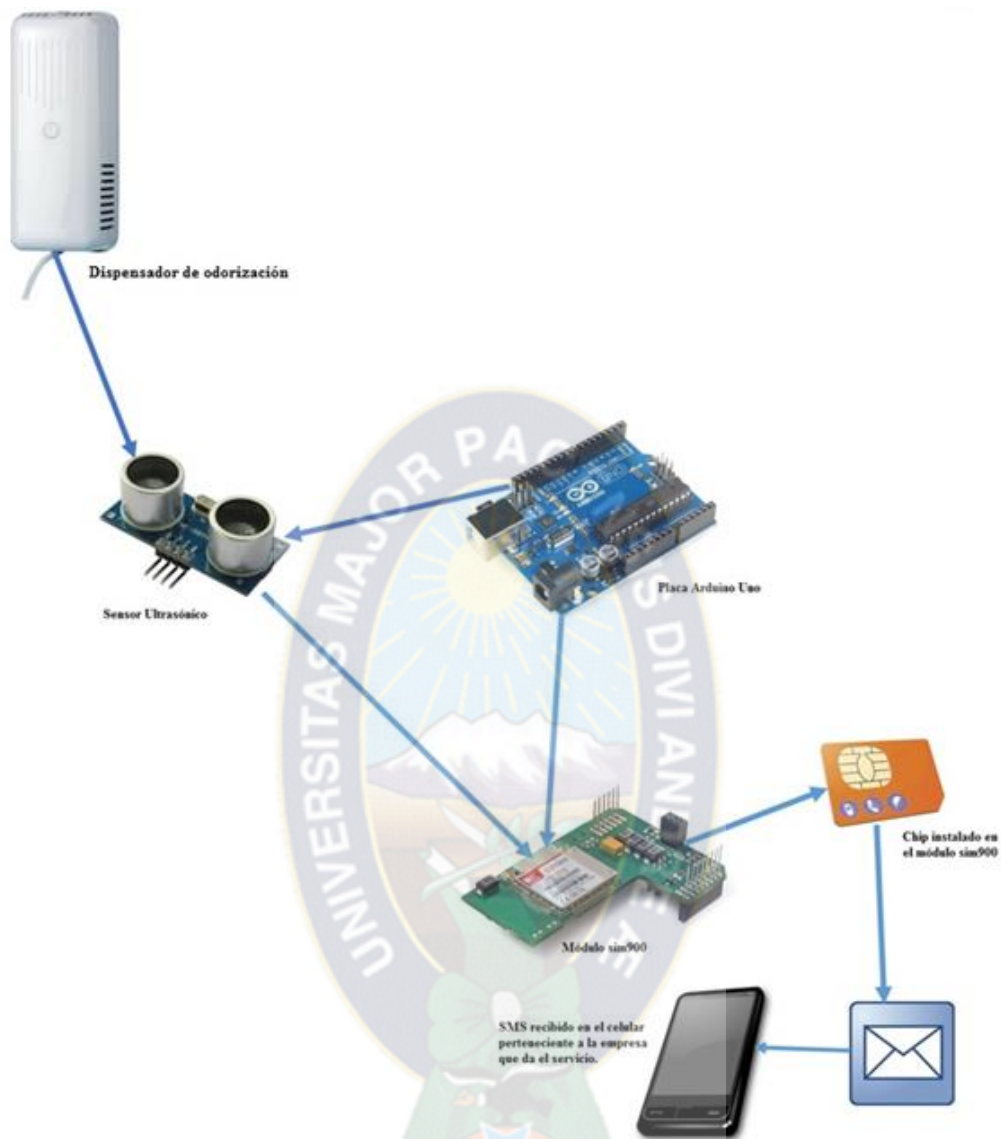


Figura 3.1 Estructura del prototipo
Fuente: Elaboración Propia

3.3.1 IMPLEMENTACIÓN DEL PROTOTIPO

Paso 1.- Instalación del sensor ultrasónico en el Protoboard

Se procede a instalar el sensor ultrasónico en el protoboard, utilizando el mapa de conexión interna entre las filas y las columnas.

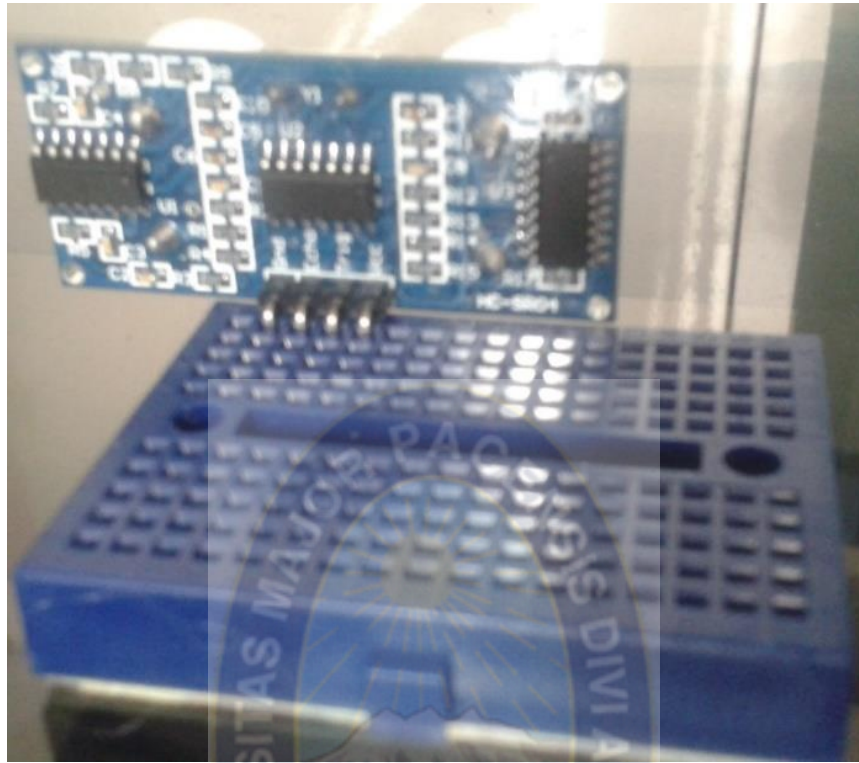


Figura 3.2 Sensor instalado en el protoboard
Fuente: Elaboración Propia

Paso 2.- Conexión de la placa Arduino UNO con el sensor ultrasónico

Se hace la conexión de la siguiente manera, se conecta un cable que sale del puerto GND del sensor ultrasonico, instalado en el protoboard, al puerto GND de la placa arduino, como se muestra en la imagen.

De la misma manera se concatena un cable del puerto VCC al puerto 5v de la placa Arduino. Por ultimo la conexión del puerto Echo y Trig pertenecientes al sensor ultrasonico, se la hace por la eleccion del programador, en este caso el Trig se instala al puerto 2 de la placa Arduino y el echo al puerto 1 de la placa arduino.

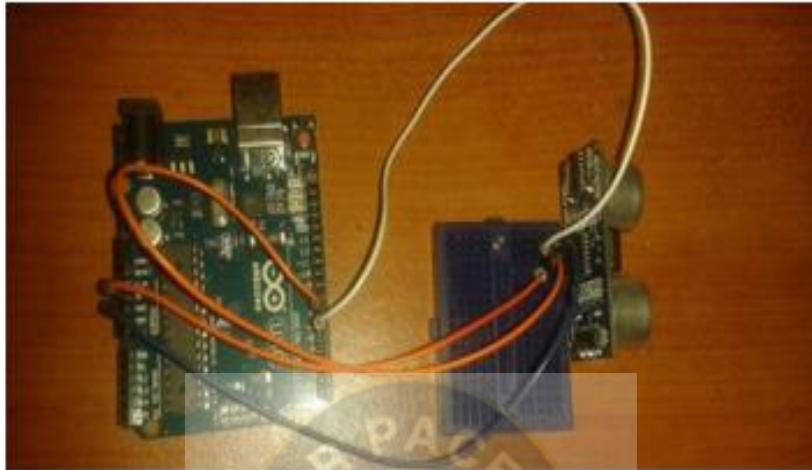


Figura 3.3 Conexión del Sensor ultrasónico con Arduino
Fuente: Elaboración Propia

Paso 3.- Instalación del módulo sim900

a) Preparación del sim900

Antes de hacer uso de este, es necesario registrarlo según el Decreto Supremo 353, al igual que se lo haría con cualquier teléfono celular con el código IMEI es decir el Código Identidad Internacional de Equipo Móvil, por sus siglas en inglés.



Figura 3.4 Preparación del módulo GPRS
Fuente: Elaboración Propia

b) Instalación del chip en el modem sim900

Existe una franja donde se puede instalar el chip de telefonía celular, una vez ubicado el lugar, se abre la tapa y se introduce el chip, posteriormente cerramos la tapa. Y el sim900 queda listo para ser programado.

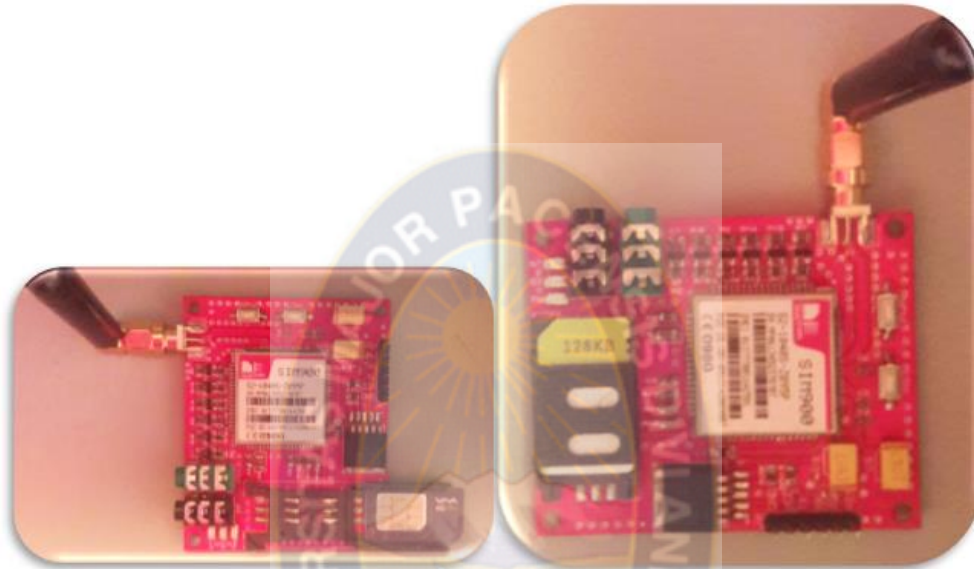


Figura 3.5 Conexión del chip a la tarjeta sim900
Fuente: Elaboración Propia

c) Se hace la conexión de la tarjeta sim900, al Arduino:

La conexión entre la placa Arduino y el modem sim900 se la desarrolla de la siguiente manera:



Figura 3.6 Instalación sim900 y Arduino
Fuente: Elaboración propia

El modem sim900 contiene un receptor RX y transmisor TX, de la misma manera la placa Arduino tiene un receptor RX y un transmisor TX.

Es así que la conexión, se hace cruzada con cables macho-hembra. Utilizando el cable rojo para la conexión positiva representada por VCC y el cable negro para la conexión negativa representada por GND.

El voltaje apropiado es de 3v para el sim900, y para la placa Arduino es de 5V, es así que se introduce una resistencia en el protoboard, para convertir el voltaje y el modem sim900 no se vea afectado.

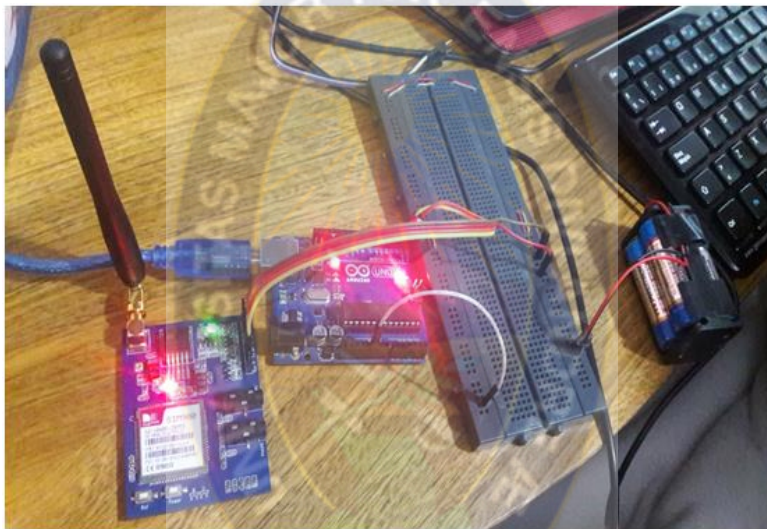


Figura 3.7 Conexión de la tarjeta sim900 al Arduino
Fuente: Elaboración Propia

Paso 4.- Instalación del sistema en el equipo de odorización.

Una vez hechas las pruebas respectivas del funcionamiento del hardware se procede a la instalación del mismo en el equipo.

Elaboramos una cavidad en la botella esto con el fin de posteriormente poder instalar el sensor ultrasónico, para que el orificio sea perfecto, se lo hace con un taladro fijo o taladro de banco, hallando un punto central y determinando el radio de cada orificio procedemos.



Figura 3.8 Elaboración de la cavidad en la botella
Fuente: Elaboración Propia

3.4 VERIFICACIÓN DEL SISTEMA

3.4.1 VERIFICACIÓN DE LA INSTALACIÓN DEL SENSOR ULTRASÓNICO EN LA PLACA ARDUINO

Una vez instalado el sensor ultrasónico en el protoboard y la conexión mediante los cables conectores con la placa Arduino UNO, se procede a verificar, que la instalación sea efectiva. Como se muestra en la figura 3.10, podemos visualizar que el foco led se enciende y comunicándonos que existe una buena conexión.



Figura 3.9 Verificación del sensor ultrasónico en la placa Arduino
Fuente: Elaboración Propia

3.4.2 VERIFICACIÓN DE LA INSTALACIÓN DEL MÓDULO SIM900

Se procede a verificar que la conexión del cableado, tenga el funcionamiento adecuado y se halle bien instalado.

Mediante los comandos AT, se puede verificar si existe comunicación entre el Arduino y el sim900.

Primero se instala la híper terminal, una vez instalada, ejecutamos el programa, en cual introducimos los comandos AT.

Si existe una respuesta, significa que es correcta la instalación entre el sim900 y Arduino. Para tal verificación se envía el comando AT, para comprobar la disponibilidad del dispositivo.

Como se puede visualizar en la figura 3.11 enviamos el comando AT y como respuesta obtenemos OK, por lo tanto existe una respuesta entre el transmisor y el receptor.

Si el mensaje seria ERROR entonces procedemos a realizar una revisión. De tal manera queda realizada la verificación, corroborando el correcto funcionamiento del sim900.

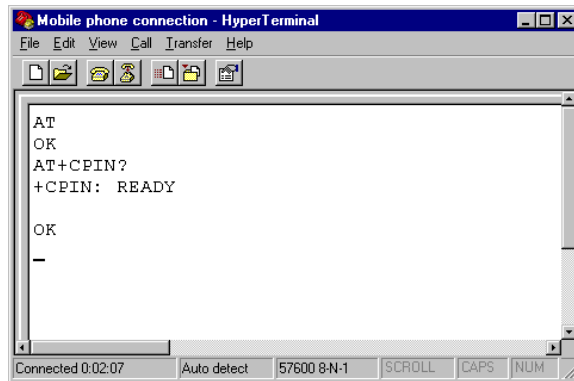


Figura 3.10 Verificación de la correcta instalación
Fuente: Elaboración Propia

3.4.3 VERIFICACIÓN DE LA INSTALACIÓN DEL SISTEMA EN EL EQUIPO DE ODORIZACIÓN

Se coloca en las cavidades realizadas de la botella el sensor ultrasónico, posicionándolo en la parte superior de esta, para la medición de la distancia del líquido

Si el líquido permanece en el envase significa que la medida del hoyo es la adecuada, si empieza a salir más líquido de lo normal, es necesario cubrir el orificio con silicona para que el momento de la instalación no entre aire a la botella y afecte el goteo normal del dispensador.



Figura 3.11 Instalación del sensor ultrasónico en la botella
Fuente: Elaboración Propia

3.5 DISEÑO DEL SOFTWARE

Para el diseño del software se tomaron en cuenta dos algoritmos, uno controla la cantidad de líquido existente en el dispensador de odorización y el segundo algoritmo controla el sim900 el cual manda un SMS, de acuerdo a los casos encontrados.

3.5.1 DESARROLLO DEL ALGORITMO PARA EL SENSOR ULTRASÓNICO

Se desarrolla un algoritmo que mide la cantidad de líquido, en la botella del dispensador de odorización.



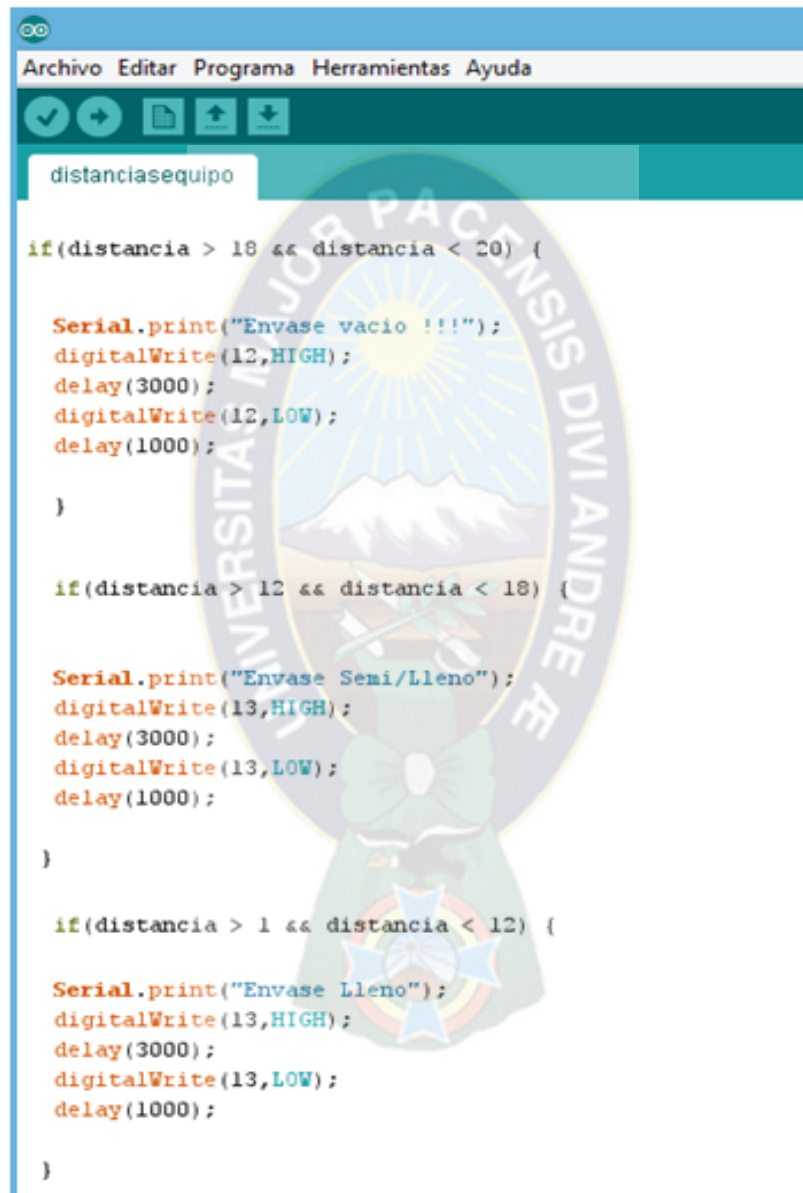
```
distanciasequipo
#define pinEcho 6
#define pinEcho 7
long duracion,distancia;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("inicializacion");
  pinMode(6, INPUT);
  pinMode(7, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(3000);
  Serial.println("nueva lectura");
  digitalWrite(7, LOW);
  delayMicroseconds(2);
  digitalWrite(7, HIGH);
  delayMicroseconds(10);
  digitalWrite(7, LOW);
  long leido = pulseIn(6, HIGH);
  long distancia = ((leido)/ 2) / 29;
  Serial.print(distancia);
  Serial.println(" cms.");
```

Figura 3.12 Interfaz de programación
Fuente: Elaboración propia

El código plasmado en el lenguaje de programación Arduino, consta de dos partes, en la primera se hace la declaración de variables, en la segunda entramos en un loop, este nos permite hacer una lectura constante de la distancia y analizar la información que ingresa.

En esta parte del código se imprime un resultado, y se vuelve a repetir la lectura.



```
Archivo Editar Programa Herramientas Ayuda
distanciasequipo

if(distancia > 18 && distancia < 20) {
  Serial.print("Envase vacio !!!");
  digitalWrite(12,HIGH);
  delay(3000);
  digitalWrite(12,LOW);
  delay(1000);
}

if(distancia > 12 && distancia < 18) {
  Serial.print("Envase Semi/Lleno");
  digitalWrite(13,HIGH);
  delay(3000);
  digitalWrite(13,LOW);
  delay(1000);
}

if(distancia > 1 && distancia < 12) {
  Serial.print("Envase Lleno");
  digitalWrite(13,HIGH);
  delay(3000);
  digitalWrite(13,LOW);
  delay(1000);
}
```

Figura 3 13 Interfaz de programación

Fuente: Elaboración Propia

3.5.2 DESARROLLO DEL ALGORITMO PARA EL SIM 900

Una vez desarrollado el algoritmo que mide la distancia, se procede a hacer la conexión del sim900 con el sensor ultrasónico vía software.



```
Archivo Editar Programa Herramientas Ayuda
distanciasequipo$
#define pinECHO 6
#define pinTrig 7
long duracion,distancia;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("inicializacion");
  pinMode(6, INPUT);
  pinMode(7, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  Serial.print("CONTROL DE DISPENSADORES DE ODORIZACION BASADO EN ARDUINO ");
  char remoteNum[71207967];
  readSerial(remoteNum);
  Serial.println(remoteNum);
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(3000);
  Serial.println("nueva lectura");
  digitalWrite(7, LOW);
  delayMicroseconds(2);
  digitalWrite(7, HIGH);
  delayMicroseconds(10);
  digitalWrite(7, LOW);
  long leido = pulseIn(6, HIGH);
  long distancia = ((leido) / 2) / 29;
  Serial.print(distancia);
  Serial.println(" cms.");

  if(distancia > 18 && distancia < 20) {

    digitalWrite(12,HIGH);
    delay(3000);
    digitalWrite(12,LOW);
    delay(1000);
  }
}
```

Figura 3.14 Interfaz de programación
Fuente: Elaboración Propia



```
Archivo  Editor  Programa  Herramientas  Ayuda

distanciaequipo$

Serial.print("CONTROL DE DISPENSADORES DE ODORIZACIÓN: ");
char txtMsg[200];
readSerial(txtMsg);
Serial.println("ENTIENDO...");
Serial.println();
Serial.println("Pasarse a hacer la recarga con suma urgencia, dispensador vacio");
Serial.println(txtMsg);

}

if(distancia > 12 || distancia < 10) {

Serial.print("Envase Semi/Lleno");
digitalWrite(13,HIGH);
delay(3000);
digitalWrite(13,LOW);
delay(1000);
Serial.print("CONTROL DE DISPENSADORES DE ODORIZACIÓN: ");
char txtMsg[200];
readSerial(txtMsg);
Serial.println("ENTIENDO...");
Serial.println();
Serial.println("Pasarse a hacer la recarga en 15 días, dispensador Semi/Lleno");
Serial.println(txtMsg);
}

if(distancia > 11 || distancia < 12) {

digitalWrite(13,HIGH);
delay(3000);
digitalWrite(13,LOW);
delay(1000);
Serial.print("CONTROL DE DISPENSADORES DE ODORIZACIÓN: ");
char txtMsg[200];
readSerial(txtMsg);
```

Figura 3.15 Interfaz de programación

Fuente: Elaboración propia

```
Archivo Editor Programa Herramientas Ayuda
distanciaequipo$

if(distancia > 1 as distancia < 12) {
    digitalWrite(13,HIGH);
    delay(3000);
    digitalWrite(13,LOW);
    delay(1000);
    Serial.print("CONTROL DE DISCENSADORES DE ODRIZACION: ");
    char txtMsg(200);
    readSerial(txtMsg);
    Serial.println("ENVIANDO...");
    Serial.println();
    Serial.println("El equipo se halla bien recargado");
    Serial.println(txtMsg);
}

int readSerial(char result[])
{
    int i = 0;
    while (1)
    {
        while (Serial.available() > 0)
        {
            char inChar = Serial.read();
            if (inChar == '\n')
            {
                result[i] = '\0';
                Serial.flush();
                return 0;
            }
            if (inChar != '\r')
            {
                result[i] = inChar;
                i++;
            }
        }
    }
}
}
```

Figura 3.16 Interfaz de programación
Fuente: elaboración propia

3.6 VERIFICACIÓN DEL SOFTWARE

Una vez desarrollado el algoritmo y plasmado en el lenguaje de programación Arduino procedemos a hacer las pruebas respectivas.

3.6.1 ALGORITMO PARA EL SENSOR ULTRASÓNICO

En esta etapa del ciclo de la metodología, se procede a elaborar las pruebas de activación del sensor ultrasónico en el Arduino Uno del control de dispensadores de odorización. Con lo cual tomamos en cuenta tres casos.

3.6.1.1 ENVASE LLENO

El sensor ultrasónico instalado en la placa Arduino detecta que la botella se halla llena. Al detectar esta actividad, el sistema muestra una alerta pudiendo visualizar la distancia entre el líquido y el sensor, de la misma manera se emite un mensaje de que la botella se halla con el líquido lleno.

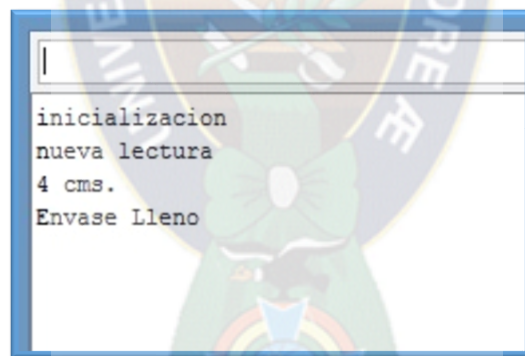


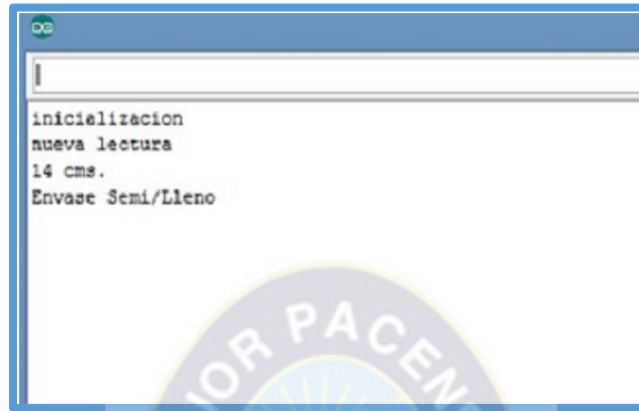
Figura 3.17 Verificación del software líquido lleno

Fuente: Elaboración propia

3.6.1.2 ENVASE SEMI/LLENO

El sensor ultrasónico instalado en la placa Arduino detecta que la botella se halla Semi/Llena. Al detectar esta actividad, el sistema muestra una alerta pudiendo visualizar la

distancia entre el líquido y el sensor, de la misma manera se emite un mensaje de que la botella se halla con el líquido Semi/llena.



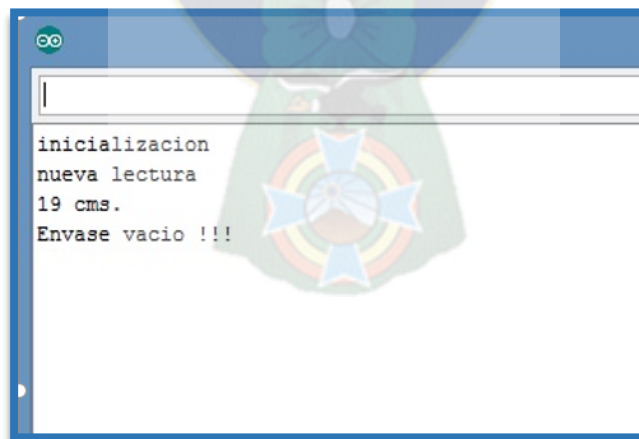
```
inicializacion
nueva lectura
14 cms.
Envase Semi/Lleno
```

Figura 3.18 Verificación del software con el líquido a la mitad

Fuente: elaboración propia

3.6.1.3 ENVASE VACÍO

El sensor ultrasónico instalado en la placa Arduino detecta que la botella se halla vacía. Al detectar esta actividad, el sistema muestra una alerta, pudiendo visualizar la distancia entre el líquido y el sensor, de la misma manera se emite un mensaje de que la botella se halla sin líquido.



```
inicializacion
nueva lectura
19 cms.
Envase vacio !!!
```

Figura 3.19 Verificación del software botella vacía

Fuente: elaboración Propia

3.6.2 VERIFICACIÓN DEL ALGORITMO PARA EL ENVÍO DE SMS

El algoritmo elaborado para la placa sim900, debe ser capaz de enviar SMS, como un mecanismo de salida, en esta etapa del ciclo, verificamos que el algoritmo funcione.

3.6.2.1 VERIFICACIÓN DEL DISPENSADOR LLENO

Cuando el sensor ultrasónico detecta que la distancia es menor a 5 cm entonces manda una señal a la placa sim900, y esta a su vez decide mandar un SMS, al celular que está programado.



Figura 3.20 Verificación del envase lleno
Fuente: Elaboración propia

3.6.2.2 VERIFICACIÓN DEL DISPENSADOR CON LA MITAD DEL PRODUCTO

Cuando el sensor ultrasónico detecta que la distancia se halla entre 18 Y 12 cm, entonces manda una señal a la placa sim900, y esta a su vez decide mandar un SMS, al celular que está programado.

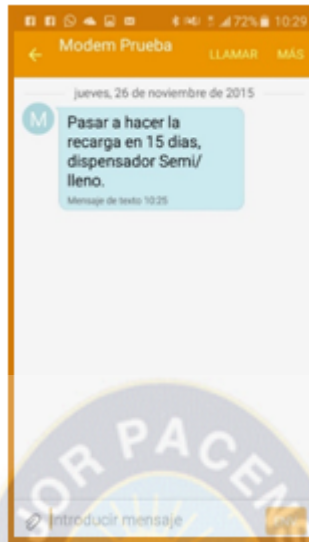


Figura 3.21 Verificación de envase Semi/Lleño vía SMS
Fuente: Elaboración Propia

3.6.2.3 VERIFICACIÓN DEL DISPENSADOR VACÍO

El sensor ultrasónico detecta ausencia de líquido en el dispensador de odorización, de esta manera se decide mandar un SMS con la información.

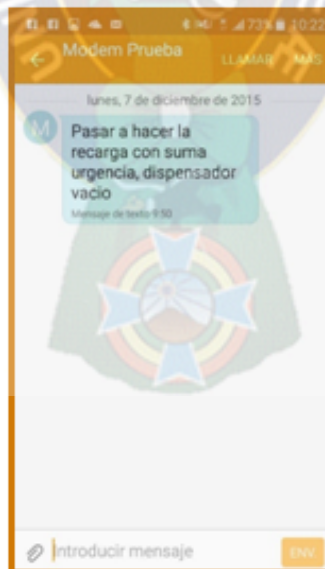


Figura 3.22 Verificación del envase vacío vía SMS
Fuente: Elaboración Propia

3.7 ENTRELAZAMIENTO DE DISPOSITIVOS

Una vez instalado el sensor ultrasónico en los dispensadores se hace la conexión con la placa Arduino. Con el entrelazamiento de dispositivos, para optimizar el prototipo.

Tomando en cuenta que el Arduino es el cerebro y el receptor de la información, como salida de información contamos con el modem sim900.

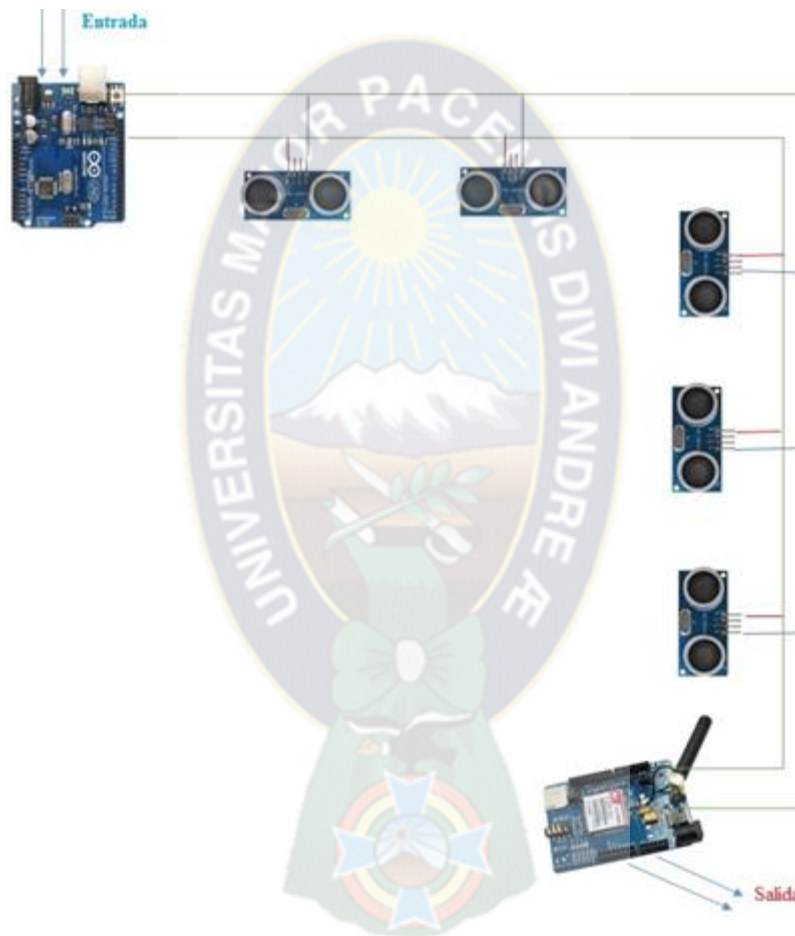


Figura 3.23 Entrelazamiento de dispositivos
Fuente: Elaboración propia

CAPITULO IV

PRUEBA DE HIPOTESIS

4.1 INTRODUCCIÓN

La prueba de hipótesis es una suposición, la cual en este capítulo verificaremos, cabe destacar que la investigación gira alrededor de un diseño cualitativo.

4.1.1 INVESTIGACIÓN CUALITATIVA

Las hipótesis de investigación cualitativas son formuladas y ajustadas durante el proceso de investigación, no se formulan antes del proceso de recolección de datos y más bien se modifican contrastándolas con la evidencia de los casos negativos que no las soportan, no se prueban estadísticamente [Hernández et al., 2010].

Es aquella donde se estudia:

- Calidad de las actividades
- Relaciones
- Asuntos
- Medios
- materiales o instrumentos

La misma analiza a detalle, un asunto o actividad en particular. A diferencia de los estudios descriptivos, correlacionales o experimentales, más que determinar la relación de causa y efectos entre dos o más variables, la investigación cualitativa se interesa más en saber cómo se da la dinámica o cómo ocurre el proceso de en qué se da el asunto o problema a tratar.

4.1.1.1 CALIDAD DE LAS ACTIVIDADES Y CASOS ENCONTRADOS

Se vio por conveniente tomar en cuenta tres casos, cuando el dispensador se halla sin producto, cuando el dispensador contiene la mitad del producto, y cuando el dispensador está lleno. Los casos son representados en la siguiente tabla.


N°	Especificación del caso	Representación visual
1.	Cuando el dispensador contiene el producto totalmente lleno.	
2.	Cuando el dispensador contiene producto a la mitad.	
3.	Cuando el dispensador no contiene producto.	

Tabla 4.1 Casos encontrados
Fuente: Elaboración Propia

4.1.1.2 FUNCIONAMIENTO DE CADA CASO

Se determinaron tres casos específicos, de los cuales el funcionamiento del prototipo en cada caso es el siguiente:

Caso n° 1

Botella llena

Cuando la botella del dispensador de odorización se encuentra llena, el sensor ultrasónico detecta que la distancia está entre 1 y 12 centímetros, el software desarrollado en Arduino determina que no hay ningún problema, puesto que lo encuentra lleno.

De igual manera se inicia un contador de días, si el contador es mayor a 5 y este mantiene la misma distancia, por 5 días, el software decide mandar un SMS, el cual determina que la empresa proveedora, necesita pasar a hacer una revisión puesto que el dispensador no está goteando ni ejerciendo el funcionamiento adecuado.

Caso n° 2

Botella con producto a la mitad

Cuando la botella del dispensador de odorización se encuentra con la botella Semi-llena, el sensor ultrasónico detecta que la distancia del producto entre el sensor y el líquido se halla entre 12 y 18 centímetros, el software desarrollado en Arduino determina que faltan 15 días para pasar a hacer la recarga, aun el cliente esta abastecido, por lo tanto no es necesario que la empresa proveedora pase a hacer la recarga.

Para la verificación del buen goteo, se inicia un contador de días, si el contador es mayor a 5 y este mantiene la misma distancia, el software decide mandar un SMS, el cual determina que la empresa proveedora necesita pasar a hacer una revisión puesto que el dispensador no está goteando ni ejerciendo el funcionamiento adecuado.

Caso n° 3

Botella vacía

Cuando la botella del dispensador de odorización se encuentra con la botella vacía, el sensor ultrasónico detecta que la distancia del producto entre el sensor y el líquido se halla entre 18 y 20 centímetros, el software desarrollado en Arduino determina que la empresa debe pasar a hacer la recarga lo más pronto posible.

Es así que mediante el módulo GPRS, sim900, se emite un SMS al celular de la empresa proveedora.

4.1.1.3 FUNCIONAMIENTO DEL PROTOTIPO

El prototipo mide la cantidad de líquido existente en cada dispensador de odorización con el uso del sensor ultrasónico logramos medir la distancia entre el líquido y el sensor ultrasónico instalado en cada botella de los dispensadores de odorización, una vez detectada dicha distancia, y dependiendo de la distancia que se determine, el software decide emitir un SMS al celular de la empresa proveedora.

Con lo cual verificamos que el presente prototipo cumple con la hipótesis planteada en el capítulo I, “El uso de un sistema de control basado en Arduino, sensores ultrasónicos y SMS permite asegurar la oportuna atención a los clientes y regula el buen funcionamiento de los dispensadores de odorización”. Puesto que se logró una comunicación, vía SMS con el equipo de odorización y la empresa proveedora.



Figura 4.1 Funcionamiento del prototipo
Fuente. Elaboración Propia

4.1.1.4 MATERIALES O INSTRUMENTOS:

Hardware:

- Arduino Uno
- Sensor ultrasónico
- SHIELDS ARDUINO

- Sim900
- Protoboard
- Cables de conexión
- Resistencias
- Focos Led

Software:

- Arduino 1.6.4

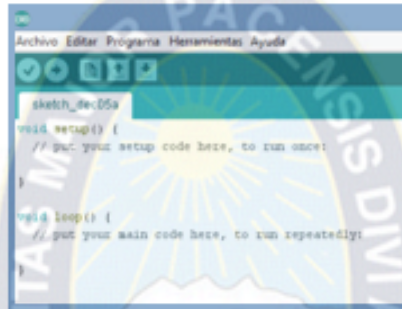


Figura 4.2 software Arduino
Fuente: elaboración Propia

Dispensador de Odorización

- Botella
- Tasa
- Mecha
- Manguera
- Líquido Bactericida

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES GENERALES

La implementación del control de dispensadores de odorización basado en Arduino, nos lleva a tener información precisa y continua del funcionamiento de los dispensadores de odorización.

Con la aplicación de la programación en los sensores ultrasónicos y el uso de la tarjeta sim900, se pudo lograr una comunicación entre los dispensadores y la empresa proveedora vía SMS. Es así que se puede concluir que el planteamiento de nuestros objetivos e hipótesis son aceptados.

5.2 ESTADO DE OBJETIVOS ESPECIFICOS

- El objetivo uno se cumple, porque se utilizó la tarjeta sim900 y así lograr una alerta temprana para evitar el mal goteo de los dispensadores de odorización.
- El objetivo dos se cumple porque logramos que exista una comunicación entre los dispensadores de odorización y la empresa que provee estos servicios, vía SMS.
- El objetivo dos se cumple porque logramos una alerta temprana, de esta manera el producto no se acaba antes de tiempo establecido, con lo cual el cliente siempre esta abastecido con el producto.
- El objetivo cuatro se cumple ya que se desarrolló un algoritmo, que mide la cantidad de producto que gasta cada mes, de esta manera tener el dato exacto de cuanto es la demanda mensual.
- El objetivo cinco se cumple puesto que se construyó un prototipo de control, que es capaz de mantener informada a las empresas que ofrecen el servicio de odorización, sobre el estado de los dispensadores en tiempo real.

- El objetivo seis se cumple puesto que mejoramos el control de los dispensadores de odorización, manipulando la información, que brinda el prototipo al emitir un SMS.

5.3 ESTADO DEL OBJETIVO GENERAL

En el capítulo I se planteó como objetivo general lo siguiente “Desarrollar un sistema de control del funcionamiento y alerta temprana de los dispensadores de odorización, basados en hardware y software libre” del objetivo general y la prueba de hipótesis, con el desglose de los casos se puede concluir que el uso de Arduino en los dispensadores de odorización efectivamente nos da una alerta temprana, con lo cual la empresa proveedora asegura la oportuna atención a los clientes y regula el buen funcionamiento de los dispensadores de odorización.

5.4 RECOMENDACIONES

- Se recomienda contar con un administrador de clientes y usuarios para la instalación del control de dispensadores de odorización.
- Extender el circuito de control de los sensores y actuadores para abarcar una mayor cantidad de dispositivos.
- Se recomienda elaborar un administrador de los mensajes que emana el control de dispensadores de odorización.
- Proteger el prototipo, mediante plomo para evitar el ruido y frecuencias, en el microcontrolador.
- El momento de programar en Arduino UNO es necesario tomar en cuenta los puertos de comunicación a ser empleados, debido a que si no reconoce el puerto serial, será necesario buscar los instaladores necesarios para cargar el programa en la placa de Arduino UNO.

CAPITULO VI

REFERENCIAS BIBLIOGRÁFICAS

6.1 REFERENCIAS BIBLIOGRÁFICAS

6.1.1 LIBROS Y TESIS

Benjamín c. Kuo, b, 1996, “sistemas de control automático”

Bruce Blum, 2007 “software engineering: a holistic view”

Brian w. Evans, 2007, “arduino programming notebook”

Bruce Blum, 2007, “software engineering: a holistic view”

Culebro Juárez m, Gómez herrero w, torres Sánchez, s., 2006, software libre

Dorothy graham, Erik van veenendaal, Isabel evans y rex black, 2007 “foundations of software testing”

Dorothy graham, Erik van veenendaal, Isabel evans y rex black, 2007 “foundations of software testing”

González Barahona j, seoane pascual j, robles g, 2003, introducción al software libre

Hernández j, 2005 software libre: técnicamente viable y económicamente sostenible y socialmente justo.

Ing. Virgilio Rosendo Pérez Pérez, v, 2010, tesis “contribución al diseño de sistemas demóticos”

Julia Garibaldi, j, 2009, “modelado, análisis y diseño del sistema de control de un robot gantry usando un sistema operativo de tiempo real convencional”

Katsuhiko ogata, k, 1996, sistemas de control en tiempo discreto

Katsuhiko ogata, k, 1996, sistemas de control moderna.

Óscar torrente artero, 2013, “arduino”

Rosario Camargo García, 2013, tesis “sistema de control de riego automático mediante el monitoreo de humedad del suelo vía internet”, universidad autónoma de Querétaro, facultad de ingeniería

Richard m. Stallman, 2004, “software libre para una sociedad”

Rafael Enríquez herrador, 2009, “guía de usuario de arduino”

Stallman rm, 2004, software libre para una sociedad libre. Ed. Traficantes de sueños. Madrid.

6.1.2 SITIOS WEB

Arduino [En línea] <<http://arduino.cc/es/Main/ArduinoBoardDuemilanove>> [consultado el 15 de junio de 2015]

Arduino [En línea] <<https://es.wikipedia.org/wiki/Arduino>> [consultado el 15 de junio de 2015].

Arduino uno [En línea] <<http://arduino.cc/en/Main/ArduinoBoardUno>> [consultado el 15 de junio de 2015]

ExtremeProgramming [En línea] <<http://www.extremeprogramming.org>> [consultado el 12 mayo del 2015].

ExtremeProgramming [En línea] <<http://www.xp123.com>> [consultado el 12 de mayo del 2015].

GPS [En línea] <https://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global> [consultados el 15 de junio de 2015].

GSM [En línea] <https://es.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones> [Consultado el 15 de junio de 2015].

Hardware libre [En línea] <http://es.wikipedia.org/wiki/Hardware_libre> [consultado el 15 de junio de 2015]

Microcontrolador [En línea]<<http://es.wikipedia.org/wiki/Microcontrolador>> [consultado el 15 de junio de 2015].

Microcontrolador[En línea]<<http://es.wikipedia.org/wiki/Microcontrolador>> [consultado el 15 de junio de 2015].

Sensor de agua [En línea] <https://es.wikipedia.org/wiki/Sensor_de_humedad> [consultado el 15 de junio de 2015].

Un aparato desodorizado perfeccionado [En línea] <<http://www.Patentados.com>> [consultado en 29 de mayo del 2015].



GLOSARIO DE TERMINOS

Algoritmo de programación

Estructuras que permiten secuencialmente generar instrucciones.

Software

Interfaz computacional en la que interactúa un usuario con un computador, una maquina o un robot.

Prototipo

Modelo de prueba y testeo que se realiza previo a la elaboración de un producto final.

Shield

Son placas impresas que se pueden conectar en la parte superior de la placa Arduino para ampliar sus capacidades, pudiendo ser apiladas una encima de otra, son fáciles de montar, y barato de producir.

GSM

Sistema Global para Comunicaciones Móviles

SMS

Servicio de mensajes cortos

USB

Bus Universal en serie es un estándar industrial desarrollado en los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores, periféricos y dispositivos electrónicos.

GND

Tierra o polo negativo

HDL

Hardware descripción lenguaje, es un lenguaje definido por el IEEE (Institute of Electrical and Electronics Engineers) usado por ingenieros para describir circuitos digitales.

RX

En la Placa GPS EM-406A es el principal canal para recibir comandos de software.

TX

En GPS EM-406A es el canal de transmisión principal para la salida de datos de navegación y medición para usuario.

RESET

Se conoce como reset a la puesta en condiciones iniciales de un sistema. Este puede ser mecánico, electrónico o de otro tipo. Normalmente se realiza al conectar el mismo, aunque, habitualmente, existe un mecanismo, normalmente un pulsador, que sirve para realzar la puesta en condiciones iniciales manualmente.

Amperio: Unidad de intensidad de corriente eléctrica.

Culombio: Se define como la cantidad de carga transportada en un segundo por una corriente de un amperio de intensidad de corriente eléctrica.

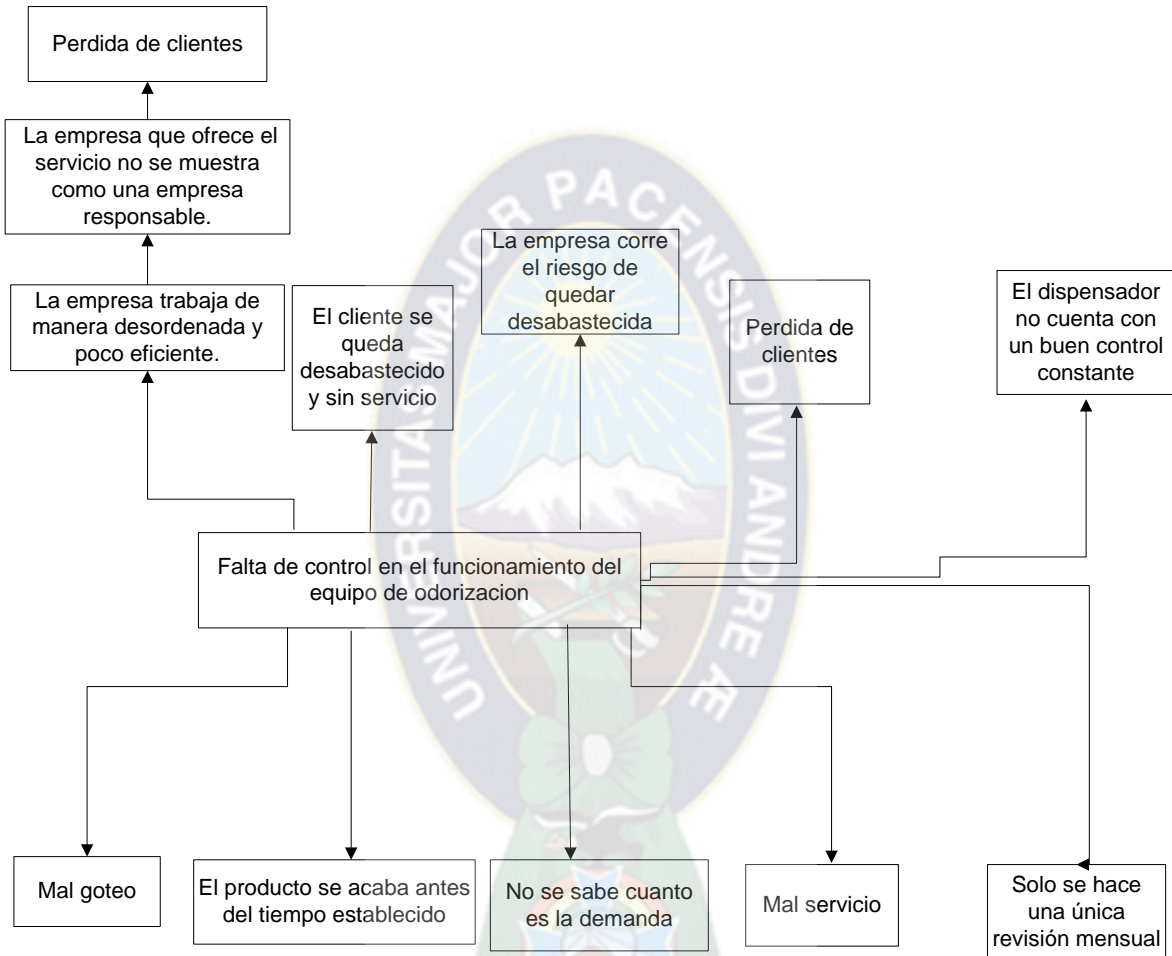
Voltaje: Es la unidad derivada del Sistema Internacional para el potencial eléctrico, la fuerza electromotriz y la tensión.

Corriente alterna: Corriente eléctrica en la que la magnitud y el sentido varían cíclicamente.

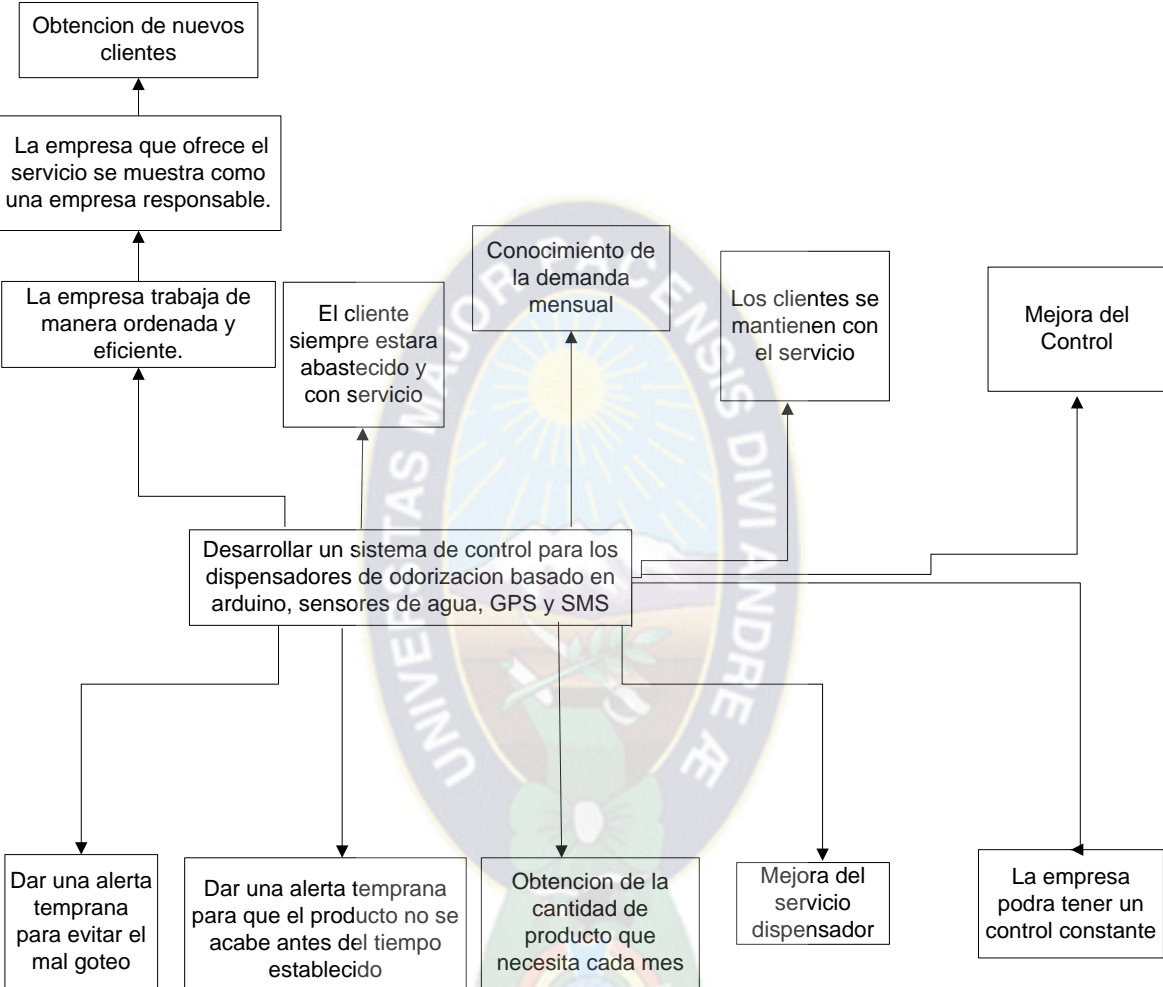
Corriente continua: cuando los electrones se mueven siempre en el mismo sentido, el flujo se denomina corriente continua y va del polo positivo al negativo

ANEXOS

ANEXO A – ÁRBOL DE PROBLEMAS



ANEXO B – ÁRBOL DE OBJETIVOS



ANEXO C - MATRIZ DE PLANIFICACIÓN (MARCO LÓGICO)

	RESUMEN NARRATIVO	INDICADORES	MEDIOS O FUENTES DE VERIFICACIÓN	SITUACIONES IMPORTANTES
OBJETIVO GENERAL	Mejorar el sistema de control de los equipos de odorización.	Metodología de Investigación Científica.	Metodología en V, Análisis de Requerimientos y Especificaciones, Análisis Funcional, Arquitectura del Sistema, implementación.	El uso de un sistema de control basado en Arduino, sensores ultrasónicos y SMS permite asegurar la oportuna atención a los clientes y regula el buen funcionamiento de los dispensadores de odorización.
OBJETIVO DEL PROYECTO	Plantear un Sistema de control basado en arduino, sensores ultrasónicos, GPS Y SMS.	Componentes del Sistema de control Arduino, sensores ultrasónicos, GPS y SMS.	Software Libre y Hardware Libre	La tecnología arduino junto con el software libre permitirá dar alerta temprana al funcionamiento de los dispensadores de odorización.
RESULTADOS	El sistema de control será capaz de dar una alerta temprana y este determinará si el equipo de odorización está en buen funcionamiento.	Verificación mediante el software utilizando el sensor ultrasónico una vez leída la lectura.	Software Arduino 1.6.4	El sistema de control mejorará el funcionamiento de los dispensadores de odorización y mediante sensores verificará la funcionalidad.
ACTIVIDADES	El componente arduino podrá realizar resultados donde en diferentes casos se verá si el dispositivo funciona con éxito o marca error.	Casos de análisis: envase lleno, envase semi lleno, envase vacío.	Equipo de Odorización, Circuito de Control Arduino, Dispositivo Móvil Receptor de GPS y SMS	Cuando el sistema de control de una alerta al detectar un error inmediatamente se realizará la alerta, enviando un mensaje a un dispositivo móvil donde indique el detalle de la falla.