

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERÍA
CARRERA INGENIERÍA ELECTRÓNICA**



PROYECTO DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UNA RED DE DISPENSADORES AUTOMÁTICOS DE AGUA, BASADA EN INTERNET DE LAS COSAS, PARA EL ANÁLISIS DE VENTA DE AGUA POR PARTE DE LA DIVISIÓN DE PURIFICACIÓN DE AGUA DEL GRUPO GUNDLACH.

PARA OPTAR AL TÍTULO DE:

INGENIERO ELECTRÓNICO

POSTULANTE: NELSON CRISTHIAN URUCHI QUISPE

ASESOR: ING. RICARDO GOTTRET RIOS

LA PAZ - BOLIVIA

NOVIEMBRE, 2020



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERIA**



LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICITAMENTE 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 cita o referencia correspondiente en apego a las 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 CONTEMPLADAS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria.

*Dedicado a mis hijos Matías y Nicolás,
mis principales motivaciones profesionales.*

Pronto nos pondremos a jugar mis niños.

Agradecimientos.

A Dios, porque nada sería posible sin su guía.

*A mis padres Nelson y Cristina,
a mi esposa Merybeth y mis hijos Mattías y Nicolás,
por su paciencia y atenciones mientras este trabajo era desarrollado,
a mis hermanos Rodrigo y Winston
y a todas las personas que me brindaron su colaboración gentilmente.*

*A mi asesor Ricardo Gottret Ríos, por permitirme
contar con un profesional tan comprometido con su trabajo,
por su apoyo y amistad que me permitieron aprender mucho más que
lo estudiado en este proyecto.*

¡Es un honor ser asesorado por un profesional de su calidad!

Resumen

La contaminación es una de las problemáticas más latentes y preocupantes que se abordan en nuestra época, debido a las consecuencias inminentes de la misma en la calidad de vida de nuestro planeta.

En la gestión 2018, India fue elegida por la Organización de las Naciones Unidas (ONU), como la sede para el evento del Día Mundial de Medio Ambiente (5 de junio), bajo el lema “Un planeta #SinContaminación por plásticos”. Con el objetivo de instar a gobiernos, sector industrial y a cada persona a unirse para reducir urgentemente la producción y el uso excesivo de plásticos desechables.

Justamente ésta es la problemática que la división de purificación de agua del grupo Gundlach (encargada de la producción y distribución de botellones de agua) pretende mitigar, mediante una solución que efectivamente el consumo de agua, evitando el manejo de las botellas de plástico de 500ml denominadas “botellas de un solo uso” y en su lugar fomentar el uso de botellas reciclables y elementos que principalmente puedan ser reutilizados.

El uso de las Tecnologías de la Información (TI) para mitigar la contaminación ambiental, y en este caso reducir la utilización de plástico, juega un papel importante, de acuerdo a la solución que se plantea para este problema.

En efecto, se pretende diseñar un dispensador de agua automático microcontrolado. Bajo el concepto de Internet de las Cosas (IoT), el dispensador podrá comunicarse con un servidor, al cual informará sobre el consumo de agua.

Con el objetivo de generar estadísticas de consumo y ventas de agua para la empresa, no solo se realizará un dispensador, sino una red IoT de dispensadores.

Este dispensador ofrecerá agua de la misma calidad que una botella de agua de mesa, pero no proporcionará una botella, sino solicitará al usuario la utilización de un envase retornable, fomentando la reutilización de botellas por parte del usuario.

Abstract

Pollution is one of the most latent and worrying problems addressed in our time, due to its imminent consequences on the quality of life on our planet.

In the 2018 administration, India was chosen by the United Nations (UN) as the venue for the World Environment Day event (June 5), under the slogan “A planet # Without Plastic Pollution”. With the aim of urging governments, the industrial sector and each person to join together to urgently reduce the production and excessive use of disposable plastics.

This is precisely the problem that the water purification division of the Gundlach group (in charge of the production and distribution of bottles of water) intends to mitigate, through a solution that makes water consumption more effective, avoiding the handling of 500ml plastic bottles. called “single-use bottles” and instead promote the use of recyclable bottles and items that can mainly be reused.

The use of Information Technology (IT) to mitigate environmental pollution, and in this case reduce the use of plastic, plays an important role, according to the solution that is proposed for this problem.

Indeed, it is intended to design a micro-controlled automatic water dispenser. Under the concept of Internet of Things (IoT), the dispenser will be able to communicate with a server, to which it will inform about water consumption.

With the aim of generating water consumption and sales statistics for the company, not only will a dispenser be built, but an IoT network of dispensers will be created.

This dispenser will offer water of the same quality as a tabletop water bottle, but it will not provide a bottle, but will ask the user to use a returnable container, promoting the reuse of bottles by the user.

ÍNDICE GENERAL

CAPÍTULO 1. INTRODUCCIÓN	1
1.1. ANTECEDENTES.....	2
1.1.1. Soluciones TI a la contaminación por plásticos	2
1.1.2. Dispensadores automáticos de líquidos	3
1.2. SITUACIÓN ACTUAL	3
1.3. PLANTEAMIENTO DEL PROBLEMA.....	3
1.4. OBJETIVOS.....	4
1.4.1. Objetivo General	4
1.4.2. Objetivos Específicos.....	4
1.5. JUSTIFICACIÓN.....	4
1.6. ALCANCES Y LIMITACIONES.....	5
1.7. METODOLOGÍA	5
CAPÍTULO 2. MARCO TEÓRICO	6
2.1. POLUCIÓN POR PLÁSTICO	6
2.1.1. Alternativas a la botella de uso único.....	7
2.1.2. Futuros trabajos	7
2.1.2.1. Biopolímeros (PHA)	7
2.2. MICROCONTROLADORES	9
2.2.1. Características principales de un microcontrolador	9
2.2.2. Principales fabricantes de semiconductores	11
2.3. SENSORES Y ACTUADORES	12
2.3.1. Módulo SIM900	12
2.3.1.1. Comandos AT	14
2.3.1.2. Tecnología GPRS	15
2.3.1.3. Protocolos de transmisión TCP y UDP.....	16
2.3.2. Botones, interruptores y/o pulsadores	17
2.3.2.1. Resistencias de polarización	18
2.3.2.2. Supresión de rebotes por hardware	19
2.3.2.3. Supresión de rebotes por software	20
2.3.2.4. El microswitch del botón	20

2.3.3.	Monedero Electrónico	21
2.3.4.	Sensor ultrasónico HC-SR04	24
2.3.5.	Bomba de Agua RS385	26
2.3.6.	Pantalla LCD 16x2	27
2.3.6.1.	Comunicación I2C	28
2.3.6.2.	Módulo conversor I2C LCD	29
2.4.	CONCEPTOS DE INGENIERÍA DE SOFTWARE.....	31
2.4.1.	Postulados del manifiesto Ágil.....	32
2.4.2.	Doce Principios del manifiesto Ágil	32
2.4.3.	Metodologías Ágiles de desarrollo.....	33
2.4.4.	Metodología SCRUM	35
2.4.4.1.	Roles en Scrum: quién y qué	36
2.4.4.2.	El Sprint: dónde y cuándo	36
2.4.4.3.	Herramientas Scrum: por qué y cómo	38
2.5.	INTERNET DE LAS COSAS (IOT)	40
2.5.1.	Dominio de Sensores.....	41
2.5.2.	Dominio de Red	42
2.5.3.	Dominio de Aplicación	43
2.5.4.	Empresas dedicadas al IoT	44
2.6.	TECNOLOGÍAS DE DESARROLLO WEB.....	45
2.6.1.	Stack WAMP.....	45
2.6.1.1.	Apache	46
2.6.1.2.	MySQL	47
2.6.1.3.	PHP	48
2.6.2.	Stack MEAN	48
2.6.2.1.	MongoDB	50
2.6.2.2.	Express Framework	52
2.6.2.3.	Angular Framework	53
2.6.2.4.	Node	55
	CAPÍTULO 3. INGENIERÍA DEL PROYECTO.....	57

3.1. ESTABLECIMIENTO DE FASES DEL PROYECTO	57
3.1.1. Recolección de necesidades de usuario.....	57
3.1.2. Elección de tecnologías a utilizar.....	61
3.1.2.1. Plataforma de Hardware	61
3.1.2.2. Plataforma de Software.....	64
3.1.3. Realización de Sprints.....	65
3.1.3.1. Sprint 1.....	65
3.1.3.2. Sprint 2.....	66
3.1.3.3. Sprint 3.....	67
3.1.3.4. Sprint 4.....	70
3.1.4. Entrega del prototipo.....	70
3.2. DESCRIPCIÓN TÉCNICA DEL PROTOTIPO	71
3.2.1. Circuito Base	71
3.2.1.1. Botones	72
3.2.1.2. Monedero electrónico	73
3.2.1.3. Sensor Ultrasónico	73
3.2.1.4. Bombas de Agua	74
3.2.1.5. Pantalla LCD.....	75
3.2.2. Canales de recepción de información en ThingSpeak	77
3.2.3. Comunicación con el servidor en la nube	78
3.2.3.1. Registro en la red	79
3.2.3.2. Conexión con el Circuito de control	80
3.2.4. Transmisión de información al servidor en la nube	80
3.2.5. Cálculo del flujo de agua por segundo	83
3.2.6. Diagrama de Flujo de la Solución final.....	83
3.2.7. Desarrollo del Tablero de Control.....	85
3.2.8. Acceso vía dispositivo Móvil.....	85
3.2.9. Indicadores técnicos importantes	86
3.3. IMPLEMENTACIÓN PRÁCTICA	87
3.4. PRUEBAS PREVIAS A LA IMPLEMENTACIÓN EN SERIE	87

CAPÍTULO 4. ANALISIS DE FACTIBILIDAD	89
4.1. FACTIBILIDAD TECNICA.....	89
4.1.1 Redes IoT	90
4.2. FACTIBILIDAD ECONOMICA.....	90
CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES	95
5.1. Conclusiones	95
5.2. Recomendaciones.....	96
Bibliografía	98
Material Anexo	101
Anexo a. código fuente a nivel de hardware	102
Anexo b. código fuente a nivel de software (DashBoard)	109
Anexo c. Set de instrucciones de comandos AT	120
Anexo d. Resolución SENAPI.....	122

INDICE DE FIGURAS

<i>Figura 1: Polución por plástico en una de las ciudades del eje troncal de Bolivia</i>	8
<i>Figura 2: Diversos campos de aplicación de los microcontroladores</i>	9
<i>Figura 3: Placa de comunicaciones SIM900.....</i>	13
<i>Figura 4: Red GPRS completa, la SIM900 actuaría como un celular más.</i>	15
<i>Figura 5: Estructura de un datagrama TCP</i>	17
<i>Figura 6: Estructura de un paquete UDP</i>	17
<i>Figura 7: Modelo de botón o pulsador básico.....</i>	18
<i>Figura 8: Rebotes que aparecen al presionar un pulsador</i>	18
<i>Figura 9: Polarizaciones PullDown y PullUp.....</i>	19
<i>Figura 10: Circuitos RC para reducir el rebote con 7404 (izquierda) y 7414 (derecha)</i>	20
<i>Figura 11: MicroSwitch incorporado en los botones del prototipo</i>	21
<i>Figura 12: Monedero electrónico simple, de una moneda</i>	22
<i>Figura 13: Espacio donde reside la moneda patrón de comparación.....</i>	22
<i>Figura 14: Monedero electrónico multimoneda</i>	24
<i>Figura 15: Sensor HC-SR04.....</i>	24
<i>Figura 16: Bomba RS385</i>	26
<i>Figura 17: Pantalla LCD 16x2</i>	27
<i>Figura 18: Forma de comunicación I2C</i>	29
<i>Figura 19: Conversor I2C para LCD</i>	30
<i>Figura 20: Conversor I2C conectado a la pantalla LCD</i>	30
<i>Figura 21: Etapas de la metodología SCRUM</i>	36
<i>Figura 22: Relación entre dispositivos y personas conectadas a internet.....</i>	40
<i>Figura 23: Dominios del concepto del IoT</i>	41
<i>Figura 24: Dominios disgregados de la arquitectura IoT</i>	45
<i>Figura 25: Componentes de un servidor WAMP</i>	46
<i>Figura 26: Funcionamiento del Stack MEAN.....</i>	49
<i>Figura 27: Ejemplo de un documento JSON</i>	50
<i>Figura 28: Comparación de términos entre una BD relacional y una BD orientada a documentos: MongoDB.....</i>	50
<i>Figura 29: Funcionamiento de Express Framework.</i>	53
<i>Figura 30: Funcionamiento de Angular Framework.....</i>	55
<i>Figura 31: Funcionamiento de un servidor node</i>	56
<i>Figura 32: Resumen del sistema IoT en funcionamiento.</i>	58

<i>Figura 33: Diagrama de Flujo general del proceso del Dispensador.</i>	59
<i>Figura 34: Diagrama de control de Dispensador</i>	61
<i>Figura 35: Modelo de Dashboard para el sistema IoT de Dispensadores.</i>	65
<i>Figura 36: Placa de desarrollo SIM900</i>	67
<i>Figura 37: Información que muestra ThingSpeak, en relación al proyecto</i>	67
<i>Figura 38: Tablero de control del dispensador</i>	68
<i>Figura 39: Interfaz de acceso para los usuarios.</i>	68
<i>Figura 40: Interfaz de gestión de usuarios.</i>	69
<i>Figura 41: Interfaz de creación / actualización de usuarios.</i>	69
<i>Figura 42: Estructura planteada al cliente para el dispensador de líquidos.</i>	70
<i>Figura 43: Prototipo del Dispensador en funcionamiento</i>	71
<i>Figura 44: Circuito previo al final, utilizando el microcontrolador ATMEGA328P</i>	71
<i>Figura 45: Botones utilizados para la construcción del prototipo</i>	72
<i>Figura 46: Mensaje de pantalla cuando la botella se encuentra lejos del dispensador</i>	74
<i>Figura 47: mitigación del ruido eléctrico del motor</i>	74
<i>Figura 48: Mensaje de pantalla cuando el dispensador se encuentra listo para operar</i>	75
<i>Figura 49: Mensaje de pantalla cuando el dispensador recibe la primera moneda</i>	75
<i>Figura 50: Mensajes de pantalla del dispensador para que el usuario se prepare</i>	76
<i>Figura 51: Mensaje de pantalla cuando el dispensador está en pleno proceso</i>	76
<i>Figura 52: Mensaje de pantalla cuando el dispensador se encuentra sin el líquido suficiente</i>	76
<i>Figura 53: Canales generados en Thingspeak</i>	77
<i>Figura 54: Descripción de la placa de comunicaciones SIM900</i>	78
<i>Figura 55: Pines a unir para habilitar el encendido por software de la SIM900</i>	79
<i>Figura 56: Pines a unir para habilitar el serial por software de la SIM900</i>	80
<i>Figura 57: Graficas que genera ThingSpeak</i>	82
<i>Figura 58: Vista de las interfaces en versión móvil</i>	86
<i>Figura 59: Reporte de consumo del Dispensador de prueba</i>	87

INDICE DE TABLAS

<i>Tabla 1: Tabla comparativa de tecnologías móviles básicas.</i>	15
<i>Tabla 2: Protocolos por capa y su función</i>	16
<i>Tabla 3: Formas generales de las sentencias básicas de definición del esquema en SQL y en MongoDB</i>	51
<i>Tabla 4: Ejemplos de las consultas en SQL y en MongoDB</i>	52
<i>Tabla 5: Tabla descriptiva de todo el proceso del Dispensador.</i>	60
<i>Tabla 6. Puertos a utilizar del microcontrolador</i>	62
<i>Tabla 7. Tabla comparativa de microcontroladores en relación a los requerimientos.</i>	63
<i>Tabla 8: Sensores y Actuadores del sistema de control del dispensador.</i>	72
<i>Tabla 9: Constantes definidas por el Cliente</i>	83
<i>Tabla 10: Indicadores del sistema completo del dispensador.</i>	87
<i>Tabla 11: Lugares estratégicos donde se instalaron los Dispensadores 2, 3, 4 y 5.</i>	88
<i>Tabla 12: Detalle de recursos técnicos para la elaboración del sistema completo.</i>	89
<i>Tabla 13: Detalle de inversión en recursos técnicos para la elaboración del sistema completo.</i>	90
<i>Tabla 14: Detalle de inversión en recurso humano para la elaboración del sistema completo.</i>	91
<i>Tabla 15: Detalle de inversión en recursos de tipo recurrente para el funcionamiento del dispensador.</i> 91	
<i>Tabla 16: Detalle de recaudación de un dispensador.</i>	92
<i>Tabla 17: Detalle de flujo de caja anual de un dispensador.</i>	92
<i>Tabla 18: Detalle del flujo de caja proyectado de un dispensador.</i>	93
<i>Tabla 19: Valores del VAN y TIR a diferentes años.</i>	93
<i>Tabla 20: Evaluación de costos para 5 dispensadores.</i>	93
<i>Tabla 21: Detalle del flujo de caja proyectado de 5 dispensadores.</i>	94
<i>Tabla 22: Valores del VAN y TIR a diferentes años.</i>	94

CAPÍTULO 1. INTRODUCCIÓN

La contaminación es una de las problemáticas más latentes y preocupantes que se abordan en nuestra época, debido a las consecuencias inminentes de la misma en la calidad de vida de nuestro planeta.

En la gestión 2018, India fue elegida por la Organización de las Naciones Unidas (ONU), como la sede para el evento del Día Mundial de Medio Ambiente (5 de junio), bajo el lema “Un planeta #SinContaminación por plásticos”. Con el objetivo de instar a gobiernos, sector industrial y a cada persona a unirse para reducir urgentemente la producción y el uso excesivo de plásticos desechables.

Justamente ésta es la problemática que la división de purificación de agua del grupo Gundlach (encargada de la producción y distribución de botellones de agua) pretende mitigar, mediante una solución que efectivice el consumo de agua, sin utilizar tantas botellas de plástico por parte del usuario final.

El uso de las Tecnologías de la Información (TI) para mitigar la contaminación ambiental, y en este caso reducir la utilización de plástico, juega un papel importante, de acuerdo a la solución que se plantea para este problema.

En efecto, se pretende diseñar un dispensador de agua automático microcontrolado. Bajo el concepto de Internet de las Cosas (IoT), el dispensador podrá comunicarse con un servidor, al cual informará sobre el consumo de agua.

Con el objetivo de generar estadísticas de consumo y ventas de agua para la empresa, no solo se realizará un dispensador, sino una red IoT de dispensadores.

Este dispensador ofrecerá agua de la misma calidad que una botella de agua de mesa, pero no proporcionará una botella, sino solicitará al usuario la utilización de un envase retornable, fomentando la reutilización de botellas que el usuario tenga.

1.1. ANTECEDENTES

1.1.1. Soluciones TI a la contaminación por plásticos

El **uso de bioplásticos** se ha convertido en una de las primeras soluciones amigables para el medio ambiente. Se trata de la producción de polímeros biodegradables mediante el uso de aditivos. Sin embargo, esto no logra resolver el problema de su origen petroquímico, lo que sigue implicando el uso de una fuente no renovable.

Entre los investigadores que experimentan con nuevos plásticos de base biológica y fácil degradación se encuentra un equipo del Instituto Italiano de Tecnología en Génova, dirigido por Ilker Bayer y Athanassia Athanassiou. Estos científicos trabajan en la producción de plásticos a partir de residuos vegetales comestibles, como el perejil, los tallos de espinacas y las cáscaras de arroz o cacao. La ventaja del método es que permite obtener una amplia gama de bioplásticos de celulosa, desde los más rígidos hasta los blandos y extensibles. “La comparación de sus propiedades mecánicas con las de varios polímeros sintéticos basados en el petróleo indica que estos bioplásticos tienen propiedades mecánicas equivalentes a las de los no degradables”, según sus estudios.

Otra alternativa que se estudia es el denominado **reciclaje hacia atrás**. Dado que el plástico se produce a partir del petróleo, **¿por qué no convertirlo de vuelta en un combustible líquido?** El método diseñado por el químico Achyut Kumar Panda, de la Universidad Centurión de Tecnología y Gestión en Odisha, y el ingeniero químico Raghubansh Kumar Singh, del Instituto Nacional de Tecnología en Orissa, utiliza un calentamiento a 450 grados centígrados en presencia de un catalizador para convertir el polietileno de las bolsas en un combustible líquido similar a la gasolina, el queroseno y el carburante diésel. Por cada kilo de plástico se producen 700 gramos de combustible. En su estudio, los investigadores subrayan que el procedimiento ayudaría a “reducir el problema de los residuos”, lo que sería especialmente interesante en los países en desarrollo.

1.1.2. Dispensadores automáticos de líquidos

Existen algunas soluciones automáticas de dispensación de agua, pero la mayoría son para uso doméstico, tal es el caso del “AquaMATIC”, que es un emprendimiento colombiano que surge en 2017, con baja aceptación hasta el momento.

Sin embargo, en relación a otros líquidos, existen dispensadores automáticos para refrescos, jaboncillo líquido o para café por ejemplo.

1.2. SITUACIÓN ACTUAL

1.3. PLANTEAMIENTO DEL PROBLEMA

La división de purificación de agua del grupo Gundlach (encargada de la producción y distribución de botellones de agua) pretende reducir la utilización de botellas de plástico por parte del usuario final, para el consumo de agua, como una medida de cuidado ambiental.

Por lo tanto, para dar una solución a esta problemática, es que se plantea la posibilidad de desarrollar un dispensador de agua automático microcontrolado.

Este dispensador ofrecerá agua de la misma calidad que una botella de agua de mesa, pero no proporcionará una botella, sino solicitará al usuario la utilización de un envase retornable, propio del usuario, fomentando la reutilización de botellas que el usuario tenga.

Utilizando el concepto de Internet de las Cosas (IoT), el dispensador podrá comunicarse con un servidor IoT, el cual registrará la información del consumo de agua.

Con el objetivo de generar estadísticas de consumo de agua para la empresa, no solo se realizará un dispensador, sino una red de dispensadores.

Finalmente, este servidor IoT podrá ser accedido mediante un sistema de información, el cual presentará un panel de control con las estadísticas y alertas necesarias para el posterior análisis y toma de decisiones de la empresa.

El sistema podrá ser revisado con cualquier dispositivo (equipo computador, móvil, etc.) y desde cualquier lugar del mundo.

1.4. OBJETIVOS

1.4.1. Objetivo General

- Diseñar e implementar una red de dispensadores automáticos de agua, basada en IoT, para el análisis de consumo de agua por parte de la división de purificación de agua del grupo Gundlach.

1.4.2. Objetivos Específicos

Para lograr este objetivo será necesario:

- Diseñar el sistema de control del dispensador de agua.
- Diseñar la red de dispensadores, basado en las características y estándares definidos sobre el concepto de IoT.
- Diseñar el sistema de información que servirá como tablero de control, para el análisis y toma de decisiones de la empresa.

1.5. JUSTIFICACIÓN

- **Social:**

El problema de las botellas plásticas radica en su único uso en cada toma. La gente se encuentra acostumbrada a tomar el contenido y desechar el envase. El dispensador, pretende brindar agua de la misma calidad que una botella, pero contribuyendo a la reutilización de envases por parte del usuario

- **Económico:**

Se ha podido encontrar que el uso de botellas plásticas representa un porcentaje del costo del producto a vender. Es decir, una botella de agua, tiene incluido en el precio, el costo de la botella.

Se pretende evitar el uso de la botella en cada toma, de esa forma reducir el costo para brindar más cantidad de agua por un precio competitivo en comparación al de una botella de agua

- **Académico:**

Durante la investigación previa a este documento, no se han encontrado desarrollos académicos que abarquen el tema a nivel Bolivia. Por lo que se pretende elaborar un proyecto que sirva como un precedente para referencias futuras.

Se realizará la aplicación de los conocimientos adquiridos a lo largo de la carrera y la investigación de nuevas áreas, para alcanzar los objetivos que se buscan en el presente proyecto, motivando a la profundización en el área del control, sistemas de información y telemática.

1.6. ALCANCES Y LIMITACIONES

Se trabajará con equipos, circuitos y redes que se encuentren presentes en la ciudad de La Paz, de preferencia de licencia libre (open source), con el propósito de proponer una solución económicamente viable.

1.7. METODOLOGÍA

Utilizamos el Método Científico, como método de estudio sistemático de la naturaleza. Este incluye las técnicas de observación, reglas para el razonamiento y la predicción, ideas sobre la experimentación planificada y las formas de comunicar los resultados experimentales y teóricos.

Los aspectos de la investigación que se tomarán en cuenta son:

Investigación exploratoria: Considerada como el primer acercamiento científico a un problema, debido a que este aún no ha sido abordado o no ha sido suficientemente estudiado y las condiciones existentes no son aún determinantes; posteriormente se pasará a la

Investigación Descriptiva: para representar, en todos sus componentes principales, una realidad.

CAPÍTULO 2. MARCO TEÓRICO

2.1. POLUCIÓN POR PLÁSTICO

La palabra plástico se refiere a un grupo de materiales sintéticos, derivados del petróleo principalmente, que se han convertido en uno de los mayores materiales utilizados en el mundo, debido a su facilidad de moldeo, impermeabilidad, baja densidad, su baja conductividad eléctrica, su resistencia a la corrosión, a la intemperie, a factores químicos y biológicos y, en buena medida, su bajo costo (Segura, Noguez, & Espín).

El plástico está presente en todos los productos que se puedan encontrar: desde el vaso de plastoformo que mantiene el calor del café, hasta la corbata que el burócrata lleva puesta.

El problema surge en que existen productos de plástico que solo se utilizan una sola vez, como las bolsas, platillos, vasos, que comúnmente llevan un adjetivo calificativo: Desechables.

Lastimosamente esa característica hace que su consumo sea demasiado elevado. De acuerdo al periódico La Razón (Pérez, 2019), en Bolivia, cada habitante utiliza 372 bolsas de plástico al año. Lo que supone que los 11 millones de bolivianos, usan alrededor de 4 mil millones de bolsas al año.

Asimismo, frente a esta situación, el 19 de febrero del presente, la Unión de Periodistas Ambientales de Bolivia – UPAB presentó un proyecto de Ley a la presidenta de la Cámara de Senadores, Adriana Salvatierra.

En la misma, se trata de la reducción del uso de bolsas plásticas, mas no se habla de otros plásticos que contaminan de igual o mayor forma, como las botellas de un solo uso.

De acuerdo a la infografía de Los Tiempos (Figura Nro. 2.1) (Cahuaya, 2019), se recogen 30 Toneladas de Botellas plásticas en un solo día en Cochabamba. Se puede deducir que similar comportamiento se produce tanto en La Paz, como en Santa Cruz de la Sierra, por ser ciudades con bastante movimiento económico en el país.

2.1.1. Alternativas a la botella de uso único

Una probable solución a este tipo de botellas sería el reciclarlas, es decir, utilizarlas más de una vez. Pero se ha comprobado que no son reciclables, debido a que poseen una sustancia denominada BPA (Bisfenol A): "...Se trata de un compuesto orgánico con cierta toxicidad, porque funciona como un disruptor endocrino, es decir: tiene la facultad de intervenir en nuestro metabolismo, desequilibrando el sistema hormonal y provocando malformaciones genéticas en los embriones de mujeres embarazadas, esterilidad en sus hijos y alteraciones en la dopamina: hiperactividad, agresividad, faltas de memoria y problemas cardiovasculares..." (Soria, 2019).

De la misma manera, en el portal Hola.com (Soria, 2019), indica que las botellas de acero son el perfecto reemplazo para el uso de botellas, debido a su facilidad de manejo y resistencia a caídas en comparación a las botellas de vidrio o de aluminio que también reemplazarían al plástico.

2.1.2. Futuros trabajos

2.1.2.1. Biopolímeros (PHA)

Un biopolímero es un plástico hecho en base a elementos biodegradables. Mantiene las características físicas del plástico contaminante, pero es amigable con el Medio Ambiente.

Los PHA son polímeros naturales producidos por bacterias (Segura, Noguez, & Espín). Son utilizados principalmente en aplicaciones médicas (para implantes, gasas), debido a su nula toxicidad.

Últimamente sea comenzado a utilizar para generar tejidos y con futuro órganos, que podrían formarse en una especie de moldes que le darían la forma de algún órgano en específico.

Si bien el PHA se produce de forma industrial, el futuro uso de este material es el del sustituto al plástico convencional. Lastimosamente resulta poco competitivo frente a la económica fabricación del plástico derivado del petróleo, por lo que actualmente se investiga métodos para reducir costos (mejora de la cepa bacteriana, tecnología de fermentación, etc).



FIGURA 1: POLUCIÓN POR PLÁSTICO EN UNA DE LAS CIUDADES DEL EJE TRONCAL DE BOLIVIA
Fuente: <https://www.lostiemplos.com.bo>

2.2. MICROCONTROLADORES

Un microcontrolador es un circuito integrado digital monolítico que contiene todos los elementos de un procesador digital secuencial síncrono programable de arquitectura Harvard o Princeton (Von Neumann). Se le suele denominar también como computador integrado o empotrado (Embedded processor) y está especialmente orientado a tareas de control y comunicaciones.

Por su pequeño tamaño los microprocesadores permiten empotrar un procesador programable en muchos productos industriales. Su costo reducido y consumo de energía y velocidad adaptables, resultan apropiados para numerosas aplicaciones. Los microcontroladores se utilizan para la realización de sistemas eléctricos empotrados en otros sistemas (eléctricos, mecánicos, etc.) (Figura 2.2) (Mandado Pérez & Mandado Rodríguez, 2008)

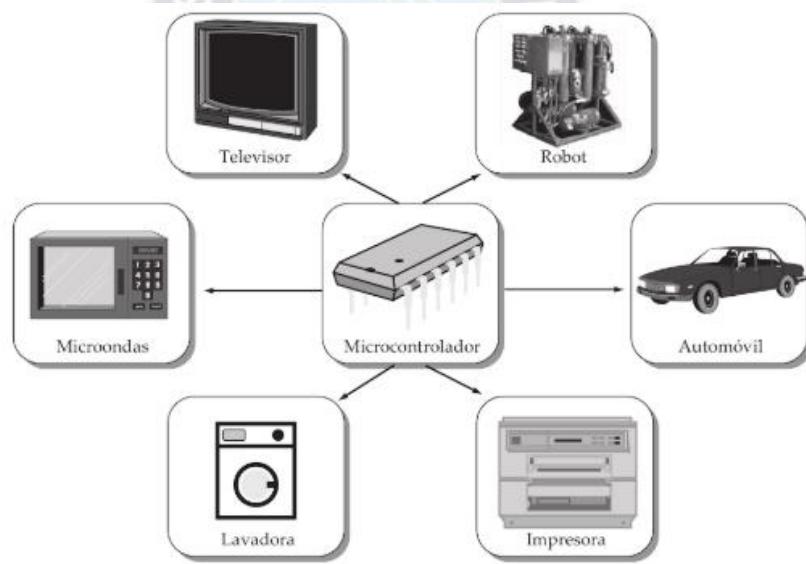


FIGURA 2: DIVERSOS CAMPOS DE APLICACIÓN DE LOS MICROCONTROLADORES

Fuente: (Mandado Pérez & Mandado Rodríguez, 2008)

2.2.1. Características principales de un microcontrolador

- **Recursos de entrada y salida.** Más que en la capacidad de cálculo del microcontrolador, muchas veces se requiere hacer énfasis en los recursos de entrada y de salida del dispositivo, tales como el manejo individual de líneas de entrada y salida, el manejo de interrupciones, señales analógicas, etc.

- **Espacio optimizado.** Se trata de tener en el menor espacio posible, y a un coste razonable, los elementos esenciales para desarrollar una aplicación. Dado que el número de terminales que puede tener un circuito integrado viene limitado por las dimensiones de su encapsulado, el espacio se puede optimizar haciendo que unos mismos terminales realicen funciones diferentes.
- **El microcontrolador idóneo para una aplicación.** Se procura que el diseñador disponga del microcontrolador hecho a la medida de su aplicación. Por esto los fabricantes ofrecen familias de microcontroladores, compuestas por miembros que ejecutan el mismo repertorio de instrucciones pero que difieren en sus componentes de hardware (más o menos memoria, más o menos dispositivos de entrada y salida, etc.), permitiendo así que el diseñador de aplicaciones pueda elegir el microcontrolador idóneo para cada aplicación.
- **Seguridad en el funcionamiento del microcontrolador.** Una medida de seguridad elemental es garantizar que el programa que esté ejecutando el microcontrolador sea el que corresponde, es decir, que si el microcontrolador se “pierde”, esto pueda ser rápidamente advertido y se tome alguna acción para corregir la situación. Un componente común en los microcontroladores y que contribuye a una operación segura es el perro guardián (WDT: Watchdog Timer), dispositivo que no existe en los ordenadores personales.
- **Bajo consumo.** Dado que hay muchas aplicaciones donde se desea utilizar baterías como fuente de alimentación, es altamente deseable que el microcontrolador consuma poca energía. También interesa que el microcontrolador consuma muy poco cuando no está realizando ninguna acción. Por ejemplo, si está a la espera de que se pulse una tecla, el microcontrolador debería consumir muy poco durante la espera; para ello conviene paralizar total o parcialmente al microcontrolador, poniéndolo “a dormir” hasta que ocurra la acción esperada.
- **Protección de los programas frente a copias.** Se trata de proteger la información almacenada en la memoria, es decir, el programa de la aplicación, contra lecturas furtivas de la memoria del microcontrolador. Los microcontroladores disponen de mecanismos que les protegen de estas acciones. (Valdez Pérez & Pallás Areny, 2007)

- **Precio:** Es importante agregar esta característica, debido a que, el factor económico puede ser el preponderante en cuanto a un posible empate en las prestaciones técnicas descritas anteriormente.

2.2.2. Principales fabricantes de semiconductores

- **Microchip:** Microchip Technology Inc. es una empresa fabricante de microcontroladores, memorias y semiconductores analógicos, situada en Chandler, Arizona, EE. UU. Su Producto más popular son los microcontroladores PIC de 8 bits.
- **Atmel Corporation:** Atmel es una compañía de semiconductores, fundada en 1984. Su línea de productos incluye microcontroladores (incluyendo derivados del 8051, el AT91SAM basados en ARM, y sus arquitecturas propias AVR y AVR32), dispositivos de radiofrecuencia, memorias EEPROM y Flash, ASICs, WiMAX, y muchas otras. También tiene capacidad de ofrecer soluciones del tipo system on chip (SoC).
- **Freescale semiconductor:** Freescale Semiconductor es una compañía global líder en la industria de semiconductores enfocada proveer procesamiento embebido y productos de conectividad. Actualmente, se enfoca al suministro de productos para la industria automotriz, de redes, comunicaciones inalámbricas, control industrial e industrias de consumo electrónico. Con se oferta de procesadores embebidos y de productos complementarios, proporciona una solución completa de semiconductores y software.
- **Texas Instruments:** Texas Instruments o TI, es una empresa norteamericana que desarrolla y comercializa semiconductores y tecnología para sistemas de cómputo. Igualmente, es el mayor productor de procesadores digitales de señal y semiconductores analógicos. TI es el tercer mayor fabricante de semiconductores del mundo tras Intel y Samsung y es el mayor suministrador de circuitos integrados para teléfonos móviles. Otras áreas de actividad incluyen circuitos integrados para módem de banda ancha, periféricos para ordenadores, dispositivos digitales de consumo y RFID.
- **ZiLOG Inc:** ZiLOG Inc, es un fabricante de microprocesadores y microcontroladores. Su producto más conocido es el Zilog Z80 de 8 bits.

- **Motorola:** Motorola Empresa dedicada a fabricar microprocesadores y microcontroladores entre otros productos, su mayor logro en la industria fue poner al Mercado un microprocesador de 8 bits, llamado 6800. Motorola fue la primera compañía en construir otros periféricos como el 6820 y el 6850.
- **Intel:** Intel empresa dedicada a la fabricación de microcontroladores y microprocesadores, aunque no trabajaba sola obtuvo un logro en abril de 1971 pone en el Mercado el microprocesador bajo el nombre 8080 con capacidad de direccionar 64kb de memoria, con 75 instrucciones y un precio de inicio de \$360 dólares. (Microcontroladores y sus aplicaciones)

Se tiene un apartado especial denominado “Elección de la plataforma de hardware” en la Ingeniería del Proyecto, donde se analiza el microcontrolador a ser utilizado de acuerdo a los requerimientos del Dispensador.

2.3. SENSORES Y ACTUADORES

2.3.1. Módulo SIM900

El módulo SIM900 es un Shield ultra compacto y confiable, basado en el chip SIM900, 100% compatible con Arduino Uno y Mega. Puede trabajar en frecuencias GSM/GPRS de 850/900/1800/1900 MHz, para realizar llamadas de voz, envío de SMS y fax. Tiene un muy bajo consumo y un diseño muy pequeño con grandes posibilidades. Se controla y configura mediante protocolo UART, usando comandos AT. Por default la velocidad UART está establecida en 19200 baudios. Para esta función posee un jumper, para poder seleccionar los pines digitales mediante los cuales queremos realizar la comunicación (D0-D3) hay un switch en la placa que nos permite seleccionar entre una conexión UART o un puerto de debug, que se puede setear para el arduino también, el SIM900 se puede conectar directamente a una pc vía un chip FTDI232. Una de las funciones más interesantes es que posee un súper capacitor con su propio circuito de alimentación, capaz de funcionar como RTC y dar información de fecha y hora. (Carrod.mx, 2019)

Características:

- Totalmente compatible con Arduino

- Conexión con el puerto serial
- Quad-Band 850/ 900/ 1800/ 1900 Mhz
- GPRS multi-slot clase 10/8GPRS mobile station clase B
- Compatible GSM fase 2/2+Clase 4 (2 W (AT) 850 / 900 MHz)
- Clase 1 (1 W (AT) 1800 / 1900MHz)TCP/UP embebido
- Soporta RTC
- Consumo de 1.5 mA (susp)
- Frecuencia 850/900/850/900

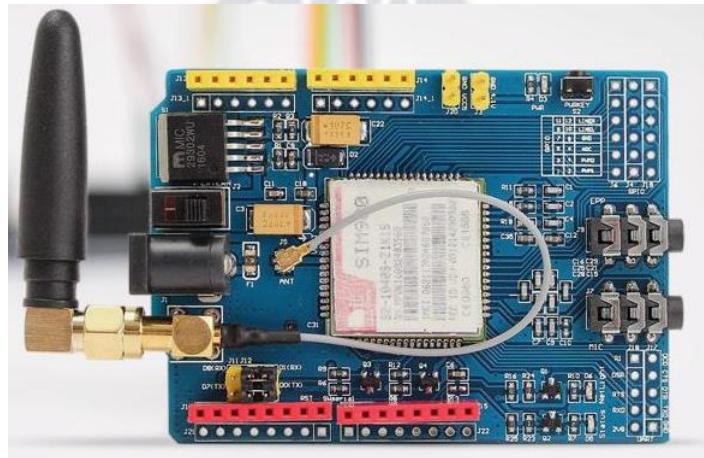


FIGURA 3: PLACA DE COMUNICACIONES SIM900.
Fuente: (<https://lastminuteengineers.com, 2019>)

Especificaciones técnicas:

- Voltaje de alimentación mínima: 9 V
- Voltaje de alimentación máxima: 20 V
- Corriente: 1.5 mA (modo descanso)
- Comunicación: UART
- Bandas de frecuencia: 850/900/1800/1900MHz
- Multi-GPRS Slot: clase 10/8
- Estación móvil: GPRS Clase B
- Fase: GSM 2/2+
- Clase 1 1W a: 1800/1900 MHz
- Clase 4 2W a: 850/900 MHz

- Servicio de mensajería (cortos): Envío de pequeñas cantidades de datos a través de la red (ascii o hexadecimal)
- Pila embebida TCP/UDP: Carga de datos a un servidor web
- Puerto serie: Libre selección
- Portabatería para batería: CR1220 (no incluida)
- Altavoz y tomas de auriculares: 2 conectores Jack 3.5 mm
- Temperatura de operación mínima: -40 °C
- Temperatura de operación máxima: 85 °C
- Dimensiones: 75 mm X 55 mm X 10 mm

2.3.1.1. Comandos AT

Los comandos AT (attention command) son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un Terminal MODEM. Los comandos AT son cadenas de caracteres ASCII que comienzan con AT y terminan con un retorno de carro (ASCII 13). Cada vez que el MODEM recibe un comando este lo procesa y emite su respuesta dependiendo como se lo haya configurado al MODEM.

El software del teléfono se comunica con el MODEM por medio de comandos AT. Este software le permite al teléfono en si comunicarse por medio de menús y el programa de comunicaciones transmite estas selecciones al MODEM en el formato que este requiere. De esta manera el MODEM realiza la tarea que se le ha comunicado. Para el uso de aplicaciones más específicas se necesita el uso de aplicaciones como Hyperterminal en el caso de Windows, y Minicom en el caso de Linux.

La implementación de los comandos AT corre a cuenta del dispositivo GSM y no depende del canal de comunicación a través del cual estos comandos son enviados, ya sea cable serial, canal infrarrojo, Bluetooth, etc.

Para una descripción completa de los comandos y las respuestas, es recomendable visitar el sitio web siguiente:

http://biblioteca.geekfactory.mx/Modulo%20Simcom%20SIM900%20GSM/DOC_SIM900_AT_Command_Manual.pdf

2.3.1.2.Tecnología GPRS

GPRS (General Packet Radio Service) es una nueva tecnología inalámbrica que comparte el rango de frecuencias de la red celular GSM (Global System for Mobile) como muestra la figura siguiente, utilizando una transmisión de datos por medio de paquetes. La conmutación de paquetes es un procedimiento más adecuado tanto para transmitir datos como para la transmisión de voz, en lugar de la conmutación de circuitos que se venía utilizando al transmitirse datos antiguamente.

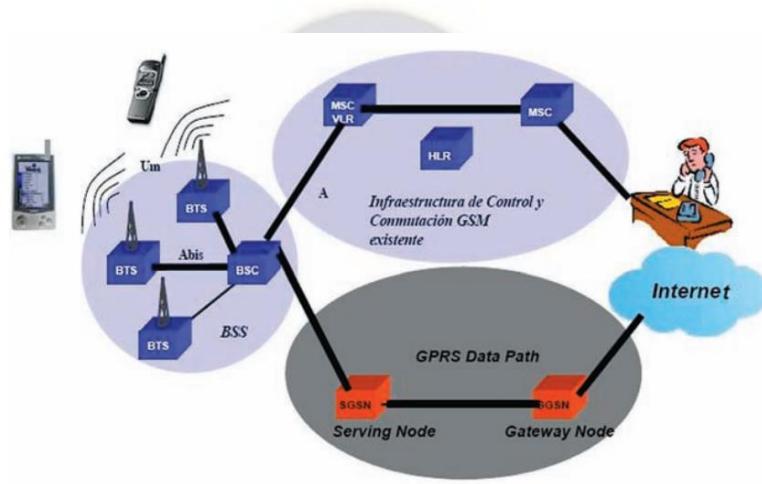


FIGURA 4: RED GPRS COMPLETA, LA SIM900 ACTUARÍA COMO UN CELULAR MÁS.

Fuente: (Viloria Núñez, Cardona Peña, & Lozano Garzón, 2009)

La siguiente tabla muestra una comparación entre GPRS y otras tecnologías celulares que trabajan de manera similar, como son EDGE (Enhanced Data Rates for GSM Evolution) y 3G (Tecnología Inalámbrica de Tercera Generación), teniendo en cuenta algunas de sus características (Viloria Núñez, Cardona Peña, & Lozano Garzón, 2009):

Tecnología	GPRS	EDGE	3G
Frecuencia de operación	0,8/1,7/1,8 GHz	0,8/1,7/1,8 GHz	1,7/2,1 GHz
Licenciado	Sí	Sí	Sí
Factor de canalización	200 kHz	200 kHz	5 MHz
Velocidad de transmisión de datos	Hasta 114 kbps	Hasta 473 kbps	Hasta 2 Mbps
Rango de cobertura	5 km o menos	5 km o menos	5 km o menos

TABLA 1: TABLA COMPARATIVA DE TECNOLOGÍAS MÓVILES BÁSICAS.

Fuente: (Viloria Núñez, Cardona Peña, & Lozano Garzón, 2009)

2.3.1.3. Protocolos de transmisión TCP y UDP

Estos dos protocolos son parte del modelo TCP/IP, que trabajan en la capa de transporte. Son protocolos que, prácticamente son contrarios en su orientación a conexión principalmente.

Capa	Protocolos Principales	Función Principal
Capa de Aplicación	SSH, DNS, FTP, RIP, SMTP, SNMP, DHCP, HTTP	Consiste de aplicaciones y procesos que utiliza la red
Capa de Transporte	TCP, UDP, DCCP, ICMP, UTP, FCP	Brinda servicios de entrega de datos entre los equipos finales
Capa de Internet	IP, ICMP, IPSEC, IGMP	Define el datagrama y la manera de intercambio de datos
Capa de Acceso a la Red	ARP, NDP, L2TP, ETHERNET	Consiste de rutinas para el acceso a la red física

TABLA 2: PROTOCOLOS POR CAPA Y SU FUNCIÓN

Fuente: Elaboración Propia

Orientación a conexión: Es la acción de establecer el camino de envío de paquetes entre dos equipos. Esto asegura la transmisión y como es conocido el camino, existe la posibilidad de reenviar algún paquete perdido. Esto significa que se debe tomar más tiempo para la conexión y el reenvío si el paquete se pierde. Además, para guardar la información del destinatario, se necesita más espacio en la cabecera del protocolo, lo que repercute en el uso de menos espacio para la carga útil.

TCP (Protocolo de Control de Transmisión) trabaja exactamente así. Es un protocolo altamente fiable, pero lento en comparación a UDP. Utiliza una cabecera de 20 Bytes, 12 Bytes más grande que la cabecera de UDP.

	0 1 2 3 4 5 6 7 8 9 10 11 1 2 1 14 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 3 4 2 5 2 6 2 7 2 8 2 9 0 1	
1	Dirección IP de Origen	Dirección IP de destino 4
2	Número de Secuencia 8	
3	Número de Acuse de Recibo 1	2
4	Long cabecera TCP Reservad o C W R E C E U R G A C K P S H R S T S N F Y N Ventana 1 6	Octetos
5	Suma de comprobación de la cabecera 2	0
...	Opciones (longitud variable)	Relleno (longitud variable) ...
1	Datos 6	
5		0

FIGURA 5: ESTRUCTURA DE UN DATAGRAMA TCP

Fuente: Elaboración propia

Debido a que UDP no establece una conexión con el destinatario, es un protocolo más rápido pero al no poder reenviar paquetes corruptos, es menos fiable. Esto permite que se utilice para envío multicast o el envío broadcast, información que no necesita un reenvío de paquetes.

	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 3 1	
1	Dirección IP de Origen	Dirección IP de destino 4
2	Longitud de Carga Útil 8	Suma de comprobación de la cabecera
4	Datos 16	

FIGURA 6: ESTRUCTURA DE UN PAQUETE UDP

Fuente: Elaboración propia

Otra utilización es la transmisión de información en tiempo real, como sería una video conferencia, en el que si se utilizara un control sobre la recepción y el orden de paquetes, se tendría que detener la transmisión únicamente para reenviar los paquetes perdidos, y poder retomarla después. Esto generaría una conexión lenta y con mucho retardo. Es preferible la pérdida de una parte de la información para poder garantizar su emisión en tiempo real. (Andalucía, 2009)

2.3.2. Botones, interruptores y/o pulsadores

Un interruptor es un dispositivo simple con dos posiciones, EN y AP (ENcendido y APagado). Una clase de interruptor que usted usa cada día es el interruptor de la luz. Cuando conecta, dentro del interruptor, dos cables son unidos, lo que permite fluir a la corriente que enciende la

luz o la tostadora se caliente. Cuando lo desconecta, los dos cables son desunidos y corta el flujo de la corriente.

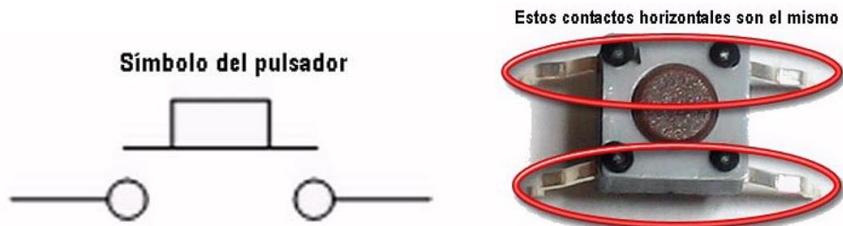


FIGURA 7: MODELO DE BOTÓN O PULSADOR BÁSICO
Fuente: (www.diarioelectronicohoy.com, 2019)

En definitiva, se trata de un mecanismo simple (los hay muy sofisticados), constituido por un par de contactos eléctricos que se unen o separan por medios mecánicos. Los falsos contactos que se producen al ser utilizados normalmente, en algunos casos produce una chispa debido a la corriente que atraviesa los contactos, provocando que los contactos eléctricos vayan llenándose de hollín, lo que a la larga acaba deteriorando dichos contactos. La chispa se produce siempre al separar los contactos (desconectar), en ocasiones parece que también salta al conectarlos, eso es debido a los rebotes mecánicos que se producen al cambiar de estado.

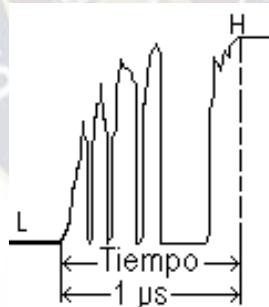


FIGURA 8: REBOTES QUE APARECEN AL PRESIONAR UN PULSADOR
Fuente: (www.diarioelectronicohoy.com, 2019)

Esto se considera un verdadero nido de problemas, debido a dichos falsos contactos. Por su propia naturaleza, al cambiar de posición un interruptor, los contactos chocan entre sí y esto significa una serie de falsos contactos que se reproducen de un modo sin control, por lo que se generan los temidos rebotes (debounce en inglés), estos rebotes, se producen incluso cuando unimos dos cables desnudos, simulando un interruptor o pulsador.

2.3.2.1. Resistencias de polarización

Existen dos tipos de polarización:

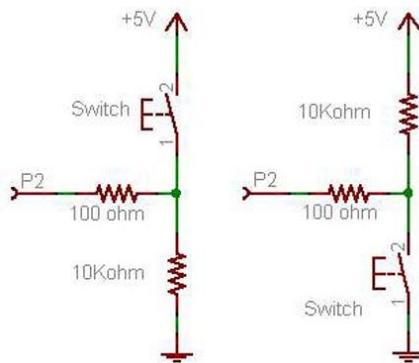


FIGURA 9: POLARIZACIONES PULLDOWN Y PULLUP
Fuente: (www.diarioelectronicohoy.com, 2019)

- Polarización alta (término inglés **Pullup**): la resistencia va conectada a + (5V)
- Polarización baja (término inglés **Pulldown**): la resistencia va conectada a tierra – (0V).

El efecto de rebote mecánico que produce inevitablemente el pulsador al ser presionado o al soltarlo, es inapreciable por el ojo humano. El ojo humano tiene una persistencia de alrededor de 0,1 segundo. (www.diarioelectronicohoy.com, 2019)

2.3.2.2. Supresión de rebotes por hardware

Una de las formas más habituales de suprimir el rebote anteriormente mencionado, es utilizando capacitores, es decir, el tiempo de carga del capacitor será el tiempo de oscilación del circuito RC que se encuentra a la entrada de un circuito integrado digital, que puede ser un negador (7404) o un Smitch Trigger (7414).

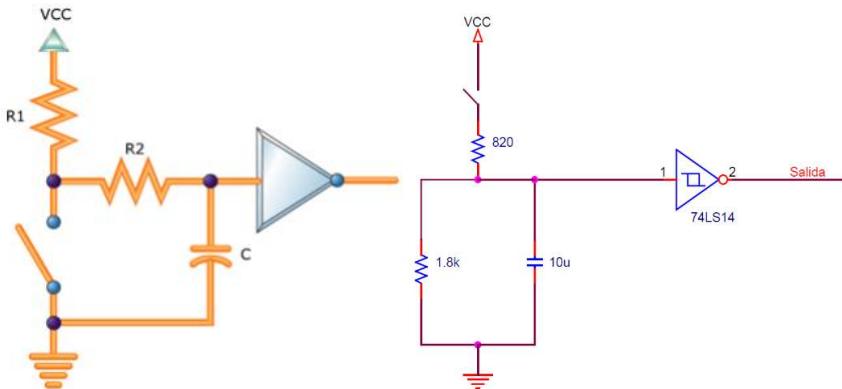


FIGURA 10: CIRCUITOS RC PARA REDUCIR EL REBOTE CON 7404 (IZQUIERDA) Y 7414 (DERECHA)
 Fuente: (www.retrowiki.es, 2019)

2.3.2.3. Supresión de rebotes por software

Para asegurar una correcta lectura de un puerto, se puede también utilizar la eliminación del rebote por software, es decir, dentro del código de lectura del puerto del microcontrolador.

La forma más sencilla de eliminar los rebotes por software es comprobar el tiempo entre disparos de la interrupción. Si el tiempo es inferior a un determinado umbral de tiempo (threshold) simplemente ignoramos la interrupción, es decir, se habrá definido una "zona muerta" en la que ignoramos las interrupciones generadas.

Un tiempo de 100-200ms es correcto para un pulsador, pero en otros casos se deberá ajustar el tiempo de forma que eliminemos el rebote, sin tener que ignorar dos posibles eventos cercanos "verdaderos".

En un montaje real, lo mejor es emplear una combinación de **ambos sistemas**, a la vez que ajustamos correctamente el valor del condensador y los tiempos del filtro por software para adaptarlos a nuestro sistema. (www.luisllamas.es, 2019)

2.3.2.4. El microswitch del botón

La duración del botón depende principalmente del microswitch que tenga incorporado.

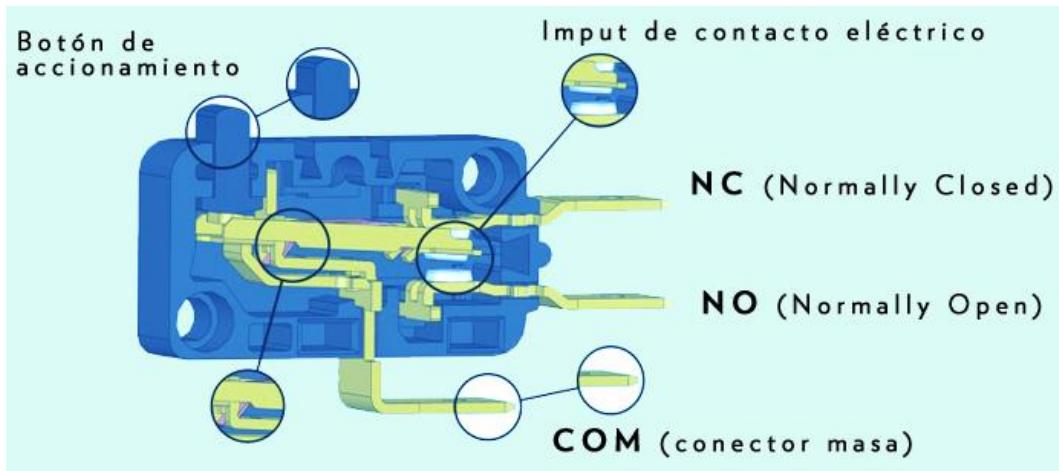


FIGURA 11: MICROSWITCH INCORPORADO EN LOS BOTONES DEL PROTOTIPO
FUENTE: (MIARCADE.COM, 2019)

Esos tres conectores están distribuidos en el microswitch de la siguiente manera: uno solo situado en uno de los lados más largos del rectángulo y otros dos en pareja, en uno de los lados más cortos. El primero se denomina masa, tierra o ground. Y los otros son los de fase, es decir, los que llevan corriente.

El de arriba es el NC (Normally Closed) y el de abajo el NO (Normally Open), que es el que cierra el circuito cuando el botón está pulsado. El otro, el NC, deja el circuito cerrado cuando el botón está en reposo.

Todas estas indicaciones, NC, NO y COM (para el conector masa), suelen venir especificadas con letras grabadas en el microswitch. (miarcade.com, 2019)

2.3.3. Monedero Electrónico

Uno de los dispositivos de sensado que se utilizan en el presente proyecto, es el monedero electrónico. Si bien existen los monederos mecánicos, éstos no son tan fiables en cuanto a su uso, ya que un monedero electrónico, puede rechazar las monedas falsas, monedas dobladas y arandelas. Su precisión es mayor cuando es necesario detectar dos monedas de tamaño similar, pero de diferente corte.

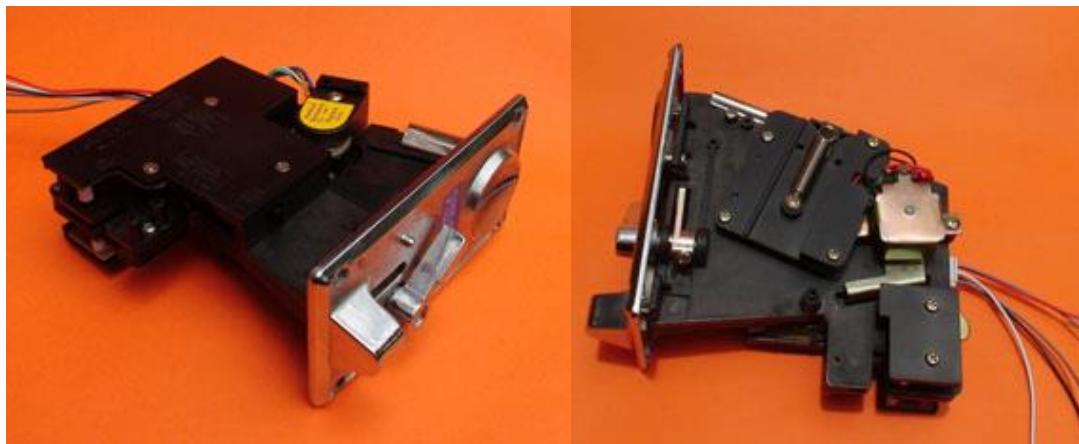


FIGURA 12: MONEDERO ELECTRÓNICO SIMPLE, DE UNA MONEDA

Fuente: (videorockola.com, 2019)

Existe una amplia gama de monederos electrónicos: algunos con comunicación RS232, con un contador interno y leds de 7 segmentos e indicadores led, multimoneda, monomoneda (simple).

En todos los casos, el funcionamiento de forma general, se basa en la comparación. Este dispositivo gracias a un solenoide, se encarga de identificar las monedas que se le introduzca con base en la comparación del tamaño y peso de una moneda patrón, que se inserta previamente. En el caso de que la moneda coincida con el patrón, ésta pasa por un conducto que conecta a la cabina donde reposan las monedas, caso contrario, ésta es devuelta a la boquilla del monedero.



FIGURA 13: ESPACIO DONDE RESIDE LA MONEDA PATRÓN DE COMPARACIÓN

Fuente: (videorockola.com, 2019)

Los monederos en general, manejan 12 Voltios de Corriente Continua (12VDC), y cuando se detecta una moneda, emiten un pulso de 2 a 3 VDC que dura entre 20 y 100 ms (parámetro configurable)

Según el portal videorockola.com: “Hay casos en que el pulso es muy rápido y el relevo no alcanza a detectarlo. Es en esos casos que debemos usar una compuerta lógica llamada **CD4066** que es un relevo de estado sólido cuádruple.”

El pulso de detección se configura de dos formas:

Normalmente Abierto (NO): El monedero cuando está en espera se mantiene todo el tiempo entregando un voltaje en su salida. Al momento que recibe una moneda, el monedero corta el voltaje momentáneamente y vuelve a su estado de normalmente abierto en el que entrega voltaje.

Normalmente Cerrado (NC): El monedero cuando está en espera, se encuentra en estado cerrado y no entrega ningún voltaje o pulso. Al momento de recibir una moneda, entrega un pulso y luego vuelve a su estado de normalmente cerrado, en el que no entrega ningún voltaje

Tolerancia de aceptación: Posee un potenciómetro que regula la sensibilidad del monedero para la detección de monedas. Mientras más sensible sea, más monedas puede rechazar.

Velocidad de detección: Dentro del monedero, existe un switch de 3 posiciones, que permite seleccionar la velocidad correcta de aceptación de la moneda. Se tiene el pulso largo (100 milisegundos) un pulso medio (50 milisegundos) y un pulso rápido (30 milisegundos), que se calibra según la velocidad del sistema que se esté diseñando.

Colores de Cables: De forma general, existen 4 cables, que se detallan a continuación:

- Cable gris – contador (\pm)
- Cable rojo – Entrada de poder o alimentación DC +12 V
- Cable blanco – Señal de salida de la moneda.
- Negro – GND (tierra)
- Cable gris-Contador (\pm)

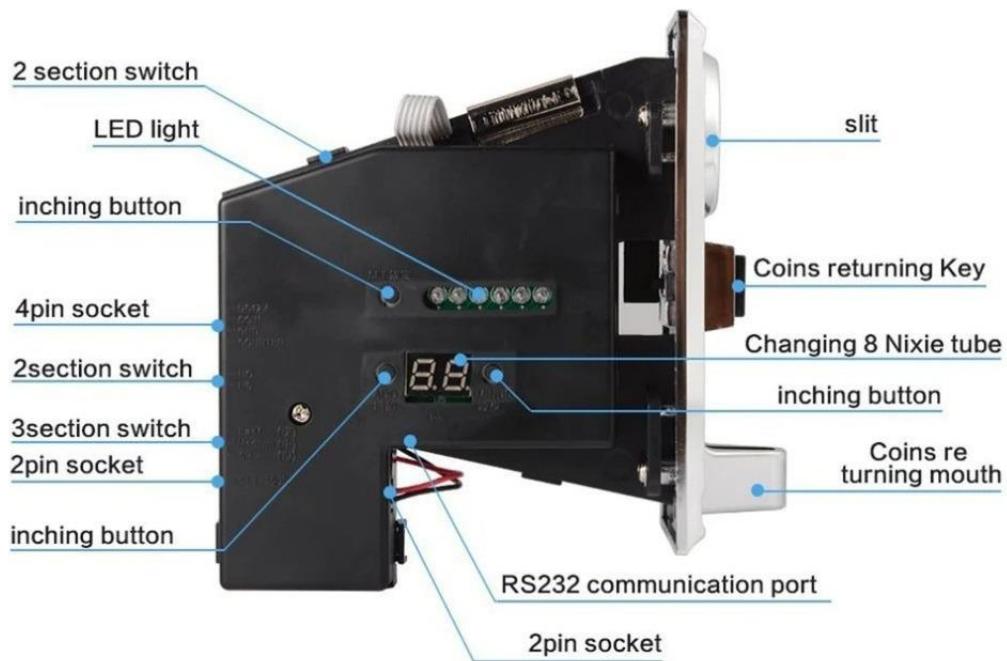


FIGURA 14: MONEDERO ELECTRÓNICO MULTIMONEDA
Fuente: (mercadolibre.com, 2019)

2.3.4. Sensor ultrasónico HC-SR04



FIGURA 15: SENSOR HC-SR04
Fuente: (naylampmechatronics.com, 2019)

El sensor HC-SR04 es un sensor de distancia de bajo costo que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. El sensor HC-SR04 es el más utilizado dentro de los sensores de tipo ultrasonido, principalmente por la cantidad de información y proyectos disponibles en la web. Es el más empleado en proyectos de robótica

como robots laberinto o sumo, y en proyectos de automatización como sistemas de medición de nivel o distancia.

El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación. El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido (40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y así se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar). (naylampmechatronics.com, 2019)

La distancia se puede calcular utilizando la siguiente formula:

$$Distancia (m) = \frac{Tiempo\ del\ pulso\ ECO\ (s) * Velocidad\ del\ Sonido\ (m/s)}{2}$$

Especificaciones Técnicas:

- Voltaje de Operación: 5V DC
- Corriente de reposo: < 2mA
- Corriente de trabajo: 15mA
- Rango de medición: 2cm a 450cm
- Precisión: +- 3mm
- Ángulo de apertura: 15°
- Frecuencia de ultrasonido: 40KHz
- Duración mínima del pulso de disparo TRIG (nivel TTL): 10 µS
- Duración del pulso ECO de salida (nivel TTL): 100-25000 µS
- Dimensiones: 45mm x 20mm x 15mm
- Tiempo mínimo de espera entre una medida y el inicio de otra 20ms (recomendable 50ms)

Pines de Conexión:

- VCC (+5V DC)
- TRIG (Disparo del ultrasonido)
- ECHO (Recepción del ultrasonido)
- GND (0V)

2.3.5. Bomba de Agua RS385

La bomba RS385 que utiliza un motor fabricado por Mabuchi Motor, trabaja como bomba de vacío o de succión, ya que no necesita una posición especial para desplazar el líquido de un lugar a otro.



FIGURA 16: BOMBA RS385
Fuente: (330ohms.com, 2019)

Se requieren entre 6V y 12V DC para accionar esta bomba y una corriente entre 500 mA y 700 mA. Esta bomba puede desplazar líquidos calientes hasta de una temperatura de 80°C y cuando está energizada correctamente puede bombear agua a una distancia de 2m y 3m hacia arriba. (330ohms.com, 2019)

Especificaciones Técnicas:

- Máximo rango de bombeo de 2m.
- Máxima cabeza de 3m.
- Maneja líquidos con temperaturas de hasta 80°C
- Flujo máximo entre 1 - 3 L/min.
- Voltaje de trabajo 6V - 12V DC

- Corriente: 0.5A - 0.7A
 - Horas de servicio: 2500h

2.3.6. Pantalla LCD 16x2

El LCD (Liquid Crystal Display) o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo. Está gobernado por un microcontrolador el cual dirige todo su funcionamiento. (todoelectrodo.blogspot.com, 2019)

El denominativo de 16x2, nos indica que se dispone de 2 filas de 16 caracteres cada una. Los píxeles de cada símbolo o carácter, varían en función de cada modelo.



FIGURA 17: PANTALLA LCD 16x2
Fuente: Elaboración propia

Podemos distinguir los siguientes pinos:

Pines de alimentación:

- Vss: Gnd
 - Vdd: +5 voltios
 - Vee: corresponde al pin de contraste, lo regularemos con un potenciómetro de 10K conectado a Vdd.

Pines de control:

- RS: Corresponde al pin de selección de registro de control de datos (0) o registro de datos (1). Es decir, el pin RS funciona paralelamente a los pines del bus de datos. Cuando RS es 0 el dato presente en el bus pertenece a un registro de

control/instrucción. y cuando RS es 1 el dato presente en el bus de datos pertenece a un registro de datos o un carácter.

- RW: Corresponde al pin de Escritura (0) o de Lectura (1). Nos permite escribir un dato en la pantalla o leer un dato desde la pantalla.
- E: Corresponde al pin *Enable* o de habilitación. Si E(0) esto quiere decir que el LCD no está activado para recibir datos, pero si E(1) se encuentra activo y podemos escribir o leer desde el LCD.

Pines de Bus de datos:

- El Bus de datos bidireccional comprende desde los pines D0 a D7. Para realizar la comunicación con el LCD podemos hacerlo utilizando los 8 bits del bus de datos(D0 a D7) o empleando los 4 bits más significativos del bus de datos(D4 a D7).

2.3.6.1.Comunicación I2C

I2C significa Circuito Inter integrado (Por sus siglas en Inglés Inter-Integrated Circuit) es un protocolo de comunicación serial desarrollado por Phillips Semiconductors allá por la década de los 80s. Básicamente se creó para poder comunicar varios chips al mismo tiempo dentro de los televisores.

El protocolo I2C toma e integra lo mejor de los protocolos SPI y UART. Con el protocolo I2C podemos tener a varios maestros controlando uno o múltiples esclavos. Esto puede ser de gran ayuda cuando se van a utilizar varios microcontroladores para almacenar un registro de datos hacia una sola memoria o cuando se va a mostrar información en una sola pantalla.

El protocolo I2C utiliza sólo dos vías o cables de comunicación, así como también lo hace el protocolo UART.

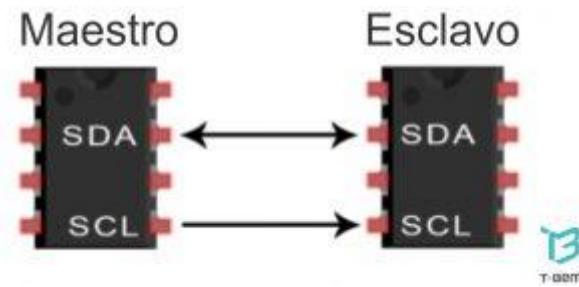


FIGURA 18: FORMA DE COMUNICACIÓN I2C

Fuente: (teslabem.com, 2019)

SDA – Serial Data. Es la vía de comunicación entre el maestro y el esclavo para enviarse información.

SCL – Serial Clock. Es la vía por donde viaja la señal de reloj.

I2C es un protocolo de comunicación serial. El protocolo I2C envía información a través de una sola vía de comunicación. La información es enviada bit por bit de forma coordinada.

I2C es un protocolo síncrono. Esto quiere decir que el envío de bits por la vía de comunicación SDA está sincronizado por una señal de reloj que comparten tanto el maestro como el esclavo a través de la vía SCL. (teslabem.com, 2019)

2.3.6.2. Módulo conversor I2C LCD

Este conversor, permite manejar una pantalla LCD con tan solo 2 pines de conexión. Está basado en el integrado PCF8574P que es un expansor remoto de 8 bits de entradas/salidas digitales por comunicación I2C. Existen muchas variantes de este módulo y todas tienen sus cambios, tanto a nivel hardware, como también en la configuración de la librería.

Generalmente, estos circuitos llevan: un led indicador de alimentación, potenciómetro para el control de contraste, transistor y jumper para la conmutación del blacklight. (electrohobby.org, 2019)

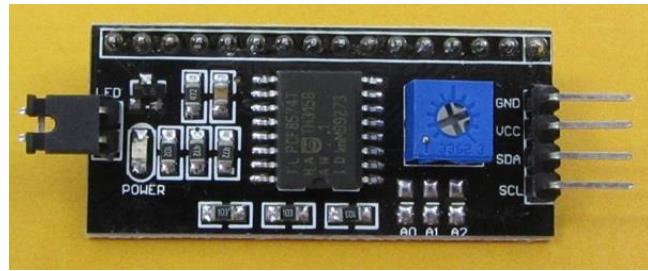


FIGURA 19: CONVERSOR I2C PARA LCD

Fuente: (electrohobby.org, 2019)

La alimentación del módulo es de 3 a 5V. Es muy importante identificar correctamente la dirección I2C de nuestro modulo, para que el programa que desarrollemos funcione correctamente. Existen algunas herramientas muy útiles para identificar la dirección específica del módulo. Por ejemplo, el Sketch de Arduino denominado “I2C Scanner”, nos permite identificar la dirección I2C del dispositivo conectado al Arduino. Existe la posibilidad de trabajar con más de un LCD, hasta 8 al mismo tiempo. Se debe modificar la dirección I2C del módulo adaptador. Para esto es necesario soldar los puentes A0, A1 y A2 presentes en el módulo, estos tres puentes son los bits menos significativos de la dirección I2C del módulo. La dirección 0x3F en binario sería: 0|0|1|1|1|A2|A1|A0 y la dirección 0x27: 0|0|1|0|0|A2|A1|A0. Por defecto A0, A2, A1 valen 1 pero si soldamos los puentes, estos se conectan a tierra teniendo un valor 0. Por ejemplo, si soldamos los tres puentes la nueva dirección sería 0|0|1|0|0|0|0|0 (0x20), para un chip que anteriormente era 0x27. (naylampmechatronics.com, 2019)



FIGURA 20: CONVERSOR I2C CONECTADO A LA PANTALLA LCD

Fuente: (naylampmechatronics.com, 2019)

2.4. CONCEPTOS DE INGENIERÍA DE SOFTWARE

La Ingeniería del Software se caracteriza por establecer marcos de trabajo en los cuales los procesos de desarrollo de software adquieren relevancia. Dichos procesos definen artefactos de desarrollo, documentación a producir, herramientas y notaciones a ser utilizadas, y actividades a realizar y su orden de ejecución, entre otras definiciones. Como resultado, los procesos generan gran cantidad de documentación con el objetivo de facilitar compartir el conocimiento entre los integrantes del equipo de trabajo. Si bien existen varios procesos de desarrollo – Proceso Unificado, Proceso V, etc., la mayoría de estos procesos se derivan de un Modelo de Cascada.

Estos procesos, denominados tradicionales, han demostrado ser efectivos, particularmente en lo que respecta a la administración de recursos a utilizar y a la planificación de los tiempos de desarrollo, en proyectos de gran tamaño con requerimientos estables. Sin embargo, debido a entornos comerciales más competitivos, conducidos por la rapidez para producir y entregar servicios, el enfoque propuesto por estos métodos no resulta el más adecuado. Usualmente, estos nuevos entornos se caracterizan por el desarrollo de proyectos donde los requerimientos del sistema son desconocidos, inestables o muy cambiantes, los tiempos de desarrollo se deben reducir drásticamente, y al mismo tiempo, se espera la producción de un producto de alta calidad, que a su vez garantice mínimo riesgo ante la necesidad de introducir cambios a los requerimientos.

Como alternativa a los métodos tradicionales de desarrollo, surgen las Metodologías Ágiles. Estas metodologías están especialmente indicadas para productos cuya definición detallada es difícil de obtener desde el comienzo, o que, si se definiera, tendría menor valor que si el producto se construye con una retro-alimentación continua durante el proceso de desarrollo. Tal el caso de **sistemas donde los usuarios no reconocen sus necesidades reales sin antes ver un componente funcionando**. La característica de estos productos y usuarios, sumado a un entorno de cambio acelerado, hacen de los métodos ágiles una alternativa viable para el mejor entendimiento de los requerimientos de un sistema embebido.

2.4.1. Postulados del manifiesto Ágil

- **Valorar al individuo y a las interacciones del equipo de desarrollo por encima del proceso y las herramientas.** Este postulado enuncia que las personas son componentes primordiales en cualquier desarrollo. Tres premisas sustentan este postulado: a) los integrantes del equipo son el factor principal de éxito; b) es más importante construir el equipo de trabajo que construir el entorno; y c) es mejor crear el equipo y que éste configure el entorno en base a sus necesidades.
- **Valorar el desarrollo de software que funcione por sobre una documentación exhaustiva.** El postulado se basa en la premisa que los documentos no pueden sustituir ni ofrecer el valor agregado que se logra con la comunicación directa entre las personas a través de la interacción con los prototipos.
- **Valorar la colaboración con el cliente por sobre la negociación contractual.** En el desarrollo ágil el cliente se integra y colabora con el equipo de trabajo. Se asume que el contrato en sí, no aporta valor al producto, sino que es sólo un formalismo que establece líneas de responsabilidad entre las partes.
- **Valorar la respuesta al cambio por sobre el seguimiento de un plan.** La evolución rápida y continua deben ser factores inherentes al proceso de desarrollo. por sobre la capacidad de seguimiento y aseguramiento de planes preestablecidos.

2.4.2. Doce Principios del manifiesto Ágil

Los postulados descritos anteriormente, inspiraron los doce principios del Manifiesto Ágil. Son las características que diferencian un proceso ágil de uno tradicional. Si bien los dos primeros son principios generales, los demás tienen que ver con los procesos a seguir, el equipo de desarrollo y su organización:

- La prioridad es satisfacer al cliente mediante entregas de software tempranas y continuas.
- Los cambios en los requerimientos son aceptados.
- Software que funcione se entrega frecuentemente, con el menor intervalo posible entre entregas.

- El cliente y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- El proyecto se construye en base a individuos motivados.
- El dialogo cara a cara es el método más eficiente y efectivo para comunicar información dentro del equipo.
- El software que funcione es la medida principal del progreso
- Los procesos ágiles promueven el desarrollo sostenido.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requerimientos y diseños surgen de equipos autoorganizados.
- El equipo reflexiona en cómo ser más efectivos, y ajusta su comportamiento en consecuencia.

2.4.3. Metodologías Ágiles de desarrollo

El ciclo de desarrollo que aplican las Metodologías Ágiles es iterativo e incremental. Este modelo permite entregar el software en partes pequeñas y utilizables, conocidas como incrementos. Estas metodologías aportan nuevos métodos de trabajo que requieren una cantidad de procesos, que no se desgastan con cuestiones administrativas (tales como planificación, control, documentación) ni tampoco defienden la postura extremista de la total falta de proceso. Debido a que se tiene conciencia que los cambios indefectiblemente se producirán, el objetivo es reducir el costo de rehacer parte del producto por los cambios introducidos. Algunas de las más conocidas son:

- **Programación Extrema (XP):** Fue la que le dio impulso al movimiento actual de metodologías ágiles. Entre los principios más importantes de esta metodología, se pueden mencionar: en cada iteración, se determina el alcance de la próxima iteración combinando prioridades del negocio y estimaciones técnicas, definiéndose la estrategia de planeamiento durante el proceso de desarrollo. Las entregas son frecuentes y continuas. Cada versión liberada debe ponerse en producción rápidamente. El desarrollo es guiado a través de historias de usuario simples. El cliente o un representante del cliente son integrados al equipo de desarrollo, quien está disponible

todo el tiempo para responder preguntas del equipo relacionadas con las reglas de negocio. El cliente es responsable por escribir los casos de prueba, demostrando que la funcionalidad está finalizada. Recomienda que el desarrollo de las funciones del producto sea realizado por dos personas en el mismo puesto – programación por pares. Cuando se finaliza la implementación de una funcionalidad, se deben corregir todos los defectos encontrados.

- **Metodología Cristal:** Se trata de un conjunto de metodologías caracterizadas por la valoración de las personas que componen el equipo de trabajo y la reducción al máximo del número de artefactos producidos. Su nombre se debe a las facetas de una gema: cada faceta es otra versión del proceso, y todas se sitúan en torno a un núcleo idéntico. Enfatiza en esfuerzos para mejorar las habilidades de los integrantes del equipo y para definir políticas de trabajo. Se estableció una clasificación por colores en función del número de integrantes del equipo, por ejemplo *Crystal Clear* corresponde a equipos de 3 a 8 integrantes, *Cristal Yellow* en equipos de 10 a 20 integrantes, *Crystal Orange* en equipos de 25 a 50 integrantes, y así sucesivamente hasta azul. Los principios destacados de la metodología incluyen: Entregas Frecuentes - entregar software a los clientes con una frecuencia diaria, semanal o mensual dependiendo del proyecto; Comunicación - promueve el uso de espacios destinados a que todos puedan ver el avance del trabajo.
- **Método Dinámico de Desarrollo de Sistemas (DSDM):** Cumple con las características generales de definir un proceso iterativo e incremental. Propone cinco etapas de desarrollo: (1) Estudio de Viabilidad, (2) Estudio del Negocio, (3) Modelado Funcional, (4) Diseño y Construcción y (5) Implementación. La iteración se produce en las tres últimas etapas, sin embargo, prevé retro-alimentación en todas.
- **Desarrollo de Software Adaptativo (ASD):** Es un proceso iterativo, tolerante a cambios y orientado a los componentes de software. Define tres etapas para el ciclo de vida: (1) Especulación – se inicia el proyecto y se planifican las características del software; (2) Colaboración – se desarrolla el producto; y (3) Aprendizaje – se revisa la calidad del producto y se entrega al cliente. La revisión tiene como objetivo aprender de los errores cometidos y volver a iniciar el ciclo de desarrollo.

- **Desarrollo Basado en Funcionalidades (FDD):** Define un proceso iterativo, con iteraciones cortas de dos semanas como máximo. El ciclo de vida consta de cinco pasos: (1) Desarrollo de un modelo global, (2) Construcción de una lista de funcionalidades, (3) Planeación por funcionalidad, (4) Diseño por funcionalidad y (5) Construcción por funcionalidad.
- **Metodología Scrum:** Indicada para proyectos con alto ratio de cambio de requerimientos, su principal característica es la definición de *sprints*, cada una de las iteraciones del proceso con una duración máxima de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. Otra característica importante son las reuniones diarias que se llevan a cabo a lo largo del proyecto. Dichas reuniones no requieren más de 15 minutos del equipo de desarrollo y su objetivo son la coordinación e integración del producto a entregar. Al principio del proyecto se define el *Product Backlog*, que contiene todos los requerimientos funcionales y no funcionales que deberá satisfacer el sistema a construir. Los mismos estarán especificados de acuerdo a los procesos definidos en la organización, ya sea mediante: funcionalidades, casos de uso, diagramas de flujo de datos, incidentes, tareas, etc. El *Product Backlog* será definido durante reuniones de planeamiento con los *stakeholders*. A partir de ahí se definirán las iteraciones, conocidas como *sprint*, en las que se irá desarrollando la aplicación evolutivamente. Cada *Sprint* tendrá su propio *Sprint Backlog* que será un subconjunto del *Product Backlog* con los requerimientos a ser construidos en dicho *sprint*.

Es necesario aclarar que no existe una metodología específica para desarrollo de sistemas embebidos. No obstante, las metodologías ágiles son las que más se asemejan a la *sistemática práctica* que se sigue para desarrollar este tipo de software. En el presente trabajo se utilizarán las características principales de la metodología SCRUM, adaptándolas al desarrollo del prototipo y aplicaciones precisadas. (Mendes Calo, Estevez, & Fillottrani, 2010)

2.4.4. Metodología SCRUM

Scrum es un framework que permite trabajar en una serie de interacciones en equipo. Las fases que definen y en las que se divide un proceso de SCRUM son las siguientes:

- **El quién y el qué:** identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto.
- **El dónde y el cuándo:** que representan el Sprint.
- **El por qué y el cómo:** representan las herramientas que utilizan los miembros de Scrum.

2.4.4.1. Roles en Scrum: quién y qué

El equipo de Scrum consiste en tres diferentes roles:

El **Product Owner/Dueño del producto**, es la “voz del cliente” y el responsable de desarrollar, mantener y priorizar las tareas en el backlog.

El **Scrum Master** es responsable de asegurarse que el trabajo del equipo vaya bien siguiendo las bases de Scrum. Además, se encarga de remover cualquier obstáculo que pueda encontrar el equipo de desarrollo.

Los **Development Team Members/Miembros del Equipo de desarrollo** son los encargados de escribir y probar el código.

2.4.4.2. El Sprint: dónde y cuándo

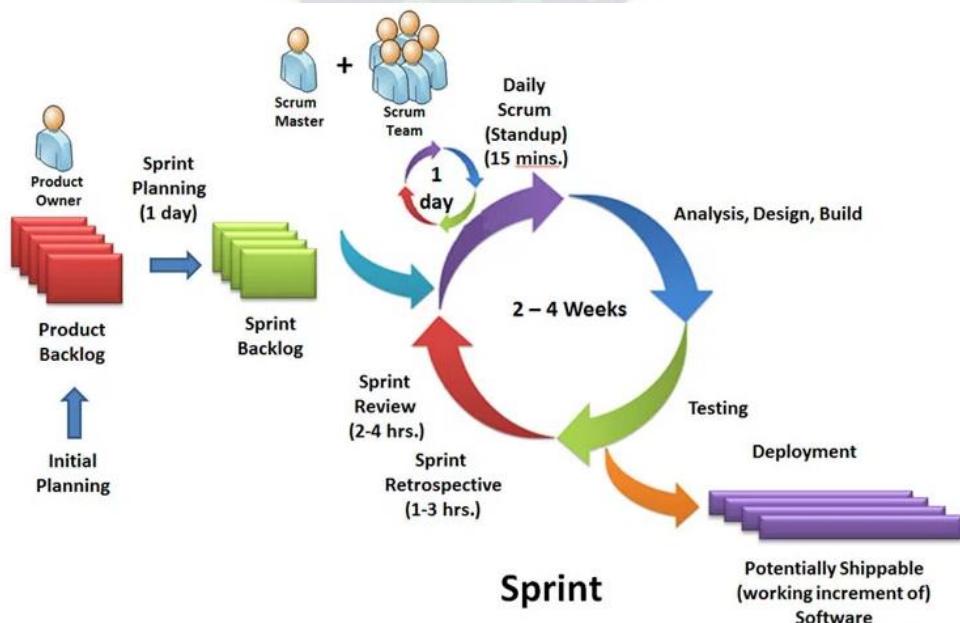


FIGURA 21: ETAPAS DE LA METODOLOGÍA SCRUM

Fuente: (Lara)

El Sprint es la unidad básica de trabajo para un equipo Scrum. Esta es la característica principal que marca la diferencia entre Scrum y otros modelos para el desarrollo ágil. Es una simple iteración llevada a cabo por los miembros del equipo. **Un equipo puede completar varios sprints durante el desarrollo del proyecto.** Un Sprint inicia con un equipo que se compromete a realizar el trabajo y finaliza con la demostración de un entregable. El tiempo mínimo para un Sprint es de una semana y el máximo es de 4 semanas. Dentro del desarrollo de un Sprint se llevan a cabo ciertos eventos, estos reciben el nombre de **Scrum Events o Eventos Scrum.** Estos son:

- Planeación del Sprint/Sprint Planning:

Todos los involucrados en el equipo se reúnen para **planificar el Sprint.** Durante este evento se decide qué requerimientos o tareas se le asignará a cada uno de los elementos del equipo. Cada integrante deberá asignar el tiempo que crea prudente para llevar a cabo sus requerimientos. **De esta manera se define el tiempo de duración del Sprint.**

- Reunión de Equipo de Scrum/Scrum team meeting:

A estas reuniones se les deberían dedicar máximo 15 minutos diarios, y deberían ser **siempre en el mismo horario y lugar.** En ellas, cada miembro del equipo deberá responder tres simples preguntas:

- ✓ ¿Qué hiciste ayer?
- ✓ ¿Qué tienes planeado hacer hoy?
- ✓ ¿Qué obstáculos encontraste en el camino?

Estas reuniones sirven para que todos los miembros del equipo se apoyen entre ellos. Si alguno de ellos tiene algún **inconveniente** que obligue a extender el encuentro, este debe tratarse más a fondo en una reunión enfocada en buscar la mejor solución para ello.

- Refinamiento del Backlog/Backlog Refinement:

El Product Owner revisa cada uno de los elementos dentro del Product Backlog con el fin de esclarecer cualquier duda que pueda surgir por parte del equipo de desarrolladores. También sirve para volver a estimar el tiempo y esfuerzo dedicado a cada uno de los requerimientos.

- Revisión del Sprint/Sprint Review

Los miembros del equipo y los clientes se reúnen para mostrar el trabajo de desarrollo de software que se ha completado. Se hace una demostración de todos los requerimientos finalizados dentro del Sprint. En este punto no es necesario que todos los miembros del equipo hablen, pueden simplemente estar presentes, pero la presentación está a cargo del **Scrum Master y el Product Owner**.

- Retrospectiva del Sprint/Retrospective

En este evento **el Product Owner se reúne con todo su equipo de trabajo y su Scrum Master** para hablar sobre lo ocurrido durante el Sprint. Los puntos principales a tratar en esta reunión son:

- ✓ Qué se hizo mal durante el Sprint para poder mejorar el próximo.
- ✓ Qué se hizo bien para seguir en la misma senda del éxito.
- ✓ Qué inconvenientes se encontraron y no permitieron poder avanzar como se tenía planificado.

2.4.4.3.Herramientas Scrum: por qué y cómo

Para poder definir las respuestas a estas preguntas nos valemos de ciertas herramientas que Scrum nos provee. Estas son:

- Backlog de Producto/Product Backlog

Esto puede referirse a **todo elemento que sea parte del proyecto**: puede ser un bug, una referencia o parte de un requerimiento. Brindan información muy general del proyecto y muchas veces no son tomados como requerimientos oficiales.

- Historias de Usuario/User Stories

Es un elemento especial del product Backlog. Se llaman **historias** porque en ellas se proporciona información sobre cómo debe ser el comportamiento del requerimiento que se está trabajando. Su función es proporcionar información directa del cliente en caso de existir algún cambio. Generalmente estos sí son tomados como **requerimientos oficiales**.

- Backlog del Sprint/Sprint Backlog

Es el conjunto de elementos tomados del Product Backlog que fueron **priorizados, medidos y aceptados en las reuniones de Sprint Planning**. Estos, en conjunto con sus respectivos User Stories, forman oficialmente los requerimientos a elaborar en cada uno de los Sprints que tendrá el proyecto.

- El panel de Tareas/The Taskboard

Este panel **muestra las tareas** que tienen asignadas los miembros del equipo. Este panel se divide en tres columnas que representan el estado de la actividad:

- ✓ Por hacer.
- ✓ Haciendo.
- ✓ Terminado.

Al inicio del Sprint todas están en la primera columna. Cuando una tarea pasa a la segunda columna, el Scrum Master y el Product Owner son notificados respecto a **qué está haciendo cada miembro del equipo y cuánto tiempo lleva trabajando en dicha tarea**. Al finalizarla, esta debe cambiarse a la última columna. Esto quiere decir que está listo para que QA haga las pruebas necesarias.

- Definición de “Listo” / Definition of Done

Todo equipo eficaz y ágil tiene ciertos acuerdos que deben cumplirse antes de dar por finalizado un proyecto. Estos son:

- ✓ Todas las tareas están completas.
- ✓ Revisión de Código / Code Reviewed.
- ✓ Pruebas realizadas a cada elemento desarrollado.
- ✓ Revisión por parte de los clientes (que cumpla sus necesidades).
- ✓ La revisión de las condiciones de Aceptación por parte del Product Owner.

Estas herramientas son útiles, no sólo durante un Sprint, sino también a lo largo del proyecto, pues ayudan al equipo a entender el porqué de cada actividad. Además, son visibles para el equipo y para los externos. (Lara)

2.5. INTERNET DE LAS COSAS (IOT)

La Internet de las cosas es un concepto bastante simple que consiste en varias tecnologías bien probadas. En la base, es la idea de conectar servidores y actuadores a Internet para recopilar datos de un área grande (a menudo una gran área geográfica) y para controlar una gran cantidad de dispositivos (de nuevo a menudo en un área geográfica grande). (Alan Grier, 2018)

La IoT involucra sensores, circuitos, sistemas embebidos, comunicaciones, interfaces inteligentes, gestión de energía, gestión de datos, fusión de datos, gestión de conocimiento, sistemas en tiempo real, procesamiento distribuido, diseño de sistemas y técnicas sofisticadas de software que se relacionan con los grandes datos. La IoT también incluye desde la Internet industrial, la cual podría cambiar profundamente como los productos son producidos, hasta la salud electrónica, que ofrece nuevas formas para el cuidado de humanos y el tratamiento de dolencias. (Piuri & Minerva, 2015)

El crecimiento explosivo de los smartphones y tablets elevó el número de dispositivos conectados a Internet a 12 500 millones en 2010, mientras que la población mundial llegó a los 6800 millones, lo que significa que el número de dispositivos conectados por persona era de más de uno (1,84, para ser exactos) por primera vez en la historia (Figura 2.5). (Dave, 2011)

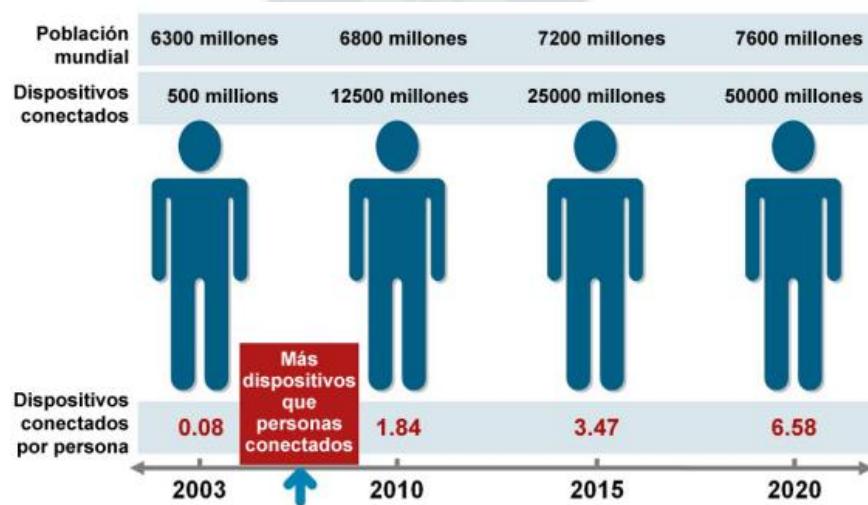


Figura 22: Relación entre dispositivos y personas conectadas a internet. IoT nació entre 2008 y 2009, por el aumento en el uso de los Smartphones.

Fuente: (Dave, 2011)

Además, el número de dispositivos conectados por persona puede parecer bajo. Esto se debe a que el cálculo se basa en toda la población mundial, gran parte de la cual todavía no está conectada a Internet. Si reducimos la muestra de población a las personas realmente conectadas a Internet, el número de dispositivos conectados por persona aumenta significativamente. Por ejemplo, se sabe que aproximadamente 2000 millones de personas utilizan Internet hoy en día. Si utilizamos este dato, el número de dispositivos conectados por persona pasa a 6,25 en 2010, en lugar del dato inicial de 1,84. (Dave, 2011)

El IoT básicamente se compone de las siguientes partes:

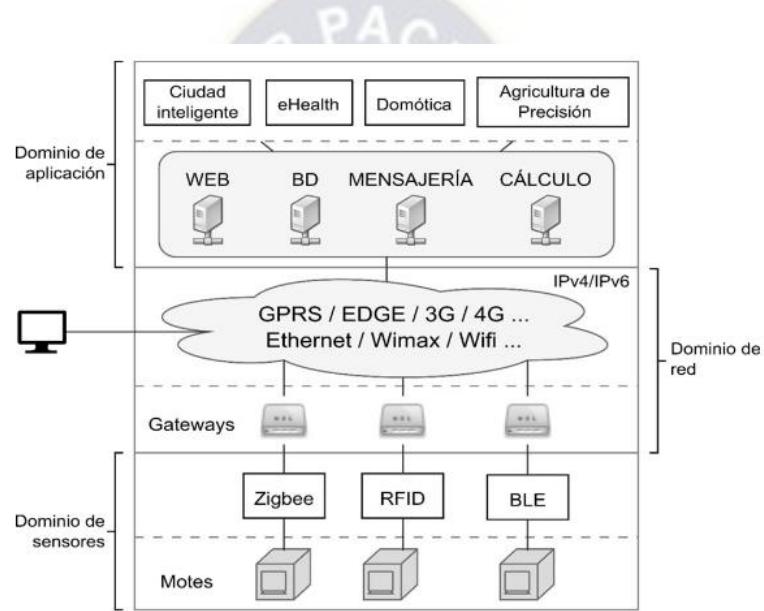


Figura 23: Dominios del concepto del IoT
Fuente: (Campoverde, Hernandez, & Mazón, 2015)

Un circuito transmisor (sensores), diseñado para el propósito, recoge la información necesaria, y mediante un medio de comunicación (Red), la envía a un servidor que, para el caso, no puede ser uno convencional, debido a las exigencias de velocidad y rápida respuesta que posee un sistema basado en IoT. Comenzaremos describiendo estos dominios desde la base.

2.5.1. Dominio de Sensores

Es el hardware necesario para recoger y enviar la información que requerimos recopilar. Es importante que este hardware sea lo más eficientemente posible en cuanto a seguridad, energía

y principalmente a la compatibilidad con el medio de comunicación que se utilice para enviar la información.

Por otra parte, la configuración del hardware es otra de las características importantes. Algunos dispositivos proveen programas de configuración, mientras que otros requieren de una programación interna para modificar su comportamiento.

La existencia de placas prefabricadas, junto con la posibilidad de adquirir sensores y actuadores a muy bajo costo, hace que se pueda crear propios proyectos de una manera relativamente sencilla. Gracias también a las capacidades de interconexión y de proceso de estos dispositivos permiten realizar proyectos de gran complejidad. (Fernandez Cejas, 2017)

Es importante que se evalúe el uso de la fuente de energía (baterías, paneles solares, aire acondicionado, etc.), ya que tienen un fuerte impacto en el mantenimiento del sistema.

2.5.2. Dominio de Red

Otro de los factores importantes que ha contribuido a poder desarrollar proyectos IoT ha sido el abaratamiento de las comunicaciones. Hoy en día tenemos acceso a Internet a bajo coste, con gran cobertura. En función del tipo de proyecto y los requerimientos del mismo, se están aplicando alguna de las siguientes estrategias:

- Uso de dispositivos multimedia tipo Smartphone con conexión directa a Internet. Esta opción supone un alto consumo de batería.
- Uso de dispositivos menos inteligentes, más baratos, colocados en lugares poco accesibles y con una batería de no mucha duración.

Asimismo, existen una variedad de protocolos de comunicación que se pueden utilizar, pero, debemos ver algunos factores para ser elegidos (RS Online: Internet de las Cosas):

- Velocidad de transferencia de datos – ¿Cuánta información se enviará?
- Consumo de energía - Como en el caso de los wearables ¿tienen una batería con una vida útil pequeña?
- Rango – ¿Necesita transmitir a unos pocos metros o a unos pocos kilómetros?
- Frecuencia – ¿Cuáles son las frecuencias disponibles en la región?

2.5.3. Dominio de Aplicación

En esta capa se encuentra la infraestructura de comunicación, almacenamiento y procesamiento de datos, así como las herramientas de análisis y presentación de la información al usuario. La infraestructura puede estar formada desde un servidor físico o virtualizado a un Centro de Procesamiento de Datos (CPD) complejo que involucre un conjunto de recursos físicos, lógicos y humanos para el control de los procesos y datos en el contexto de IoT. La virtualización de los recursos físicos y disponibilidad en internet se conoce como computación en la nube o Cloud Computing. Se describen algunas de las principales funciones de esta capa:

- **Recolección de datos crudos.** El CPD se comunica con la capa Dominio de Red mediante el internet y usa un protocolo de comunicación para recolectar los datos crudos. Existen varios protocolos de comunicación, por ejemplo: MQTT, CoAP, REST, XMPP, etc. MQTT es muy popular por su bajo consumo de ancho de banda y bajo consumo de recursos.
- **Pre-procesamiento y almacenamiento de datos.** Consiste en la limpieza y transformación de datos para luego ser almacenados en sistemas gestores de bases de datos, y/o pasar a un sistema de cálculo o simplemente ser monitoreados y controlados en tiempo real
- **Monitoreo y control.** Los datos de sensores se presentan en un tablero de control (*dashboard IoT*) visual para que el usuario comprenda el estado actual de la zona o área que está vigilando. Un *dashboard IoT* además de monitorear sensores también puede controlar actuadores, por ejemplo, encender o apagar una bomba, abrir o cerrar una electroválvula.
- **Aplicaciones o dominios IoT.** Software con interfaz web o móvil que interactúa con el usuario y a su vez con los componentes IoT. Las aplicaciones IoT también se las conoce como Smart: ciudades inteligentes (Smart Cities), hogar y edificio inteligente (Smart Home and Building), cuidado y salud inteligente (Smart Healthcare), agricultura inteligente o Agricultura de Precisión
- **Ánalisis de datos.** Procesa los datos crudos y los combina con datos extraídos de los sistemas transaccionales para obtener información útil que ayude la toma de

decisiones. La Estadística, Inteligencia de Negocios, Minería de Datos, Inteligencia Artificial, Machine Learning son algunas de las disciplinas aplicables para análisis de datos en el contexto de IoT. El sistema que coordina y gestiona todos los componentes del dominio de aplicación de internet de las cosas se le conoce como Plataforma IoT. (Hernández Rojas, Mazon-Olivo, & Escudero, 2018)

2.5.4. Empresas dedicadas al IoT

Si bien se pueden montar todos estos dominios de forma autónoma, existen empresas dedicadas a proporcionar todo el material para cada dominio. Tal es así, que existen sensores ya equipados con acceso a internet inalámbrico y baterías.

Empresas como ZigBee, LoRa o Sigfox ofrecen redes propias que permiten una conexión independiente de Internet para transmitir sus datos.

De la misma manera, existen plataformas que ofrecen (incluso de forma gratuita, pero con ciertas restricciones) todas las necesidades del dominio de aplicación. Utilizando un API de comunicación para que los dispositivos antes mencionados puedan comunicarse directamente.

Existen un primer grupo de plataformas que están orientadas a startups y empresas pequeñas. Permiten su uso de una manera gratuita pero con limitaciones en cuanto al número de mensajes enviados y de dispositivos conectados. Podemos mencionar a: Samsung Artik Cloud, aREST Framework, Thinger.io, Arduino Cloud, ThingSpeak, Cayenne.

El segundo grupo engloba a plataformas que también ofrecen servicios gratuitos o versiones de prueba. Están más centradas en ofrecer servicios globales a sistemas basados en el IoT. Ya no es solo recibir datos, en estas plataformas nos permiten almacenar webs, API para móviles, bases de datos, etc. Podemos mencionar a: TheThingsIO, IFTTT Maker, Particle y Firebase Google.

El último grupo serían las plataformas que ofrecen las grandes empresas y corporaciones. Están orientadas sobre todo al sector industrial y a grandes proyectos del IoT, donde se ven involucrados cientos o miles de dispositivos. Mencionamos a: Ayla Networks, Autodesk Fusion Connect, AWS (Amazon Web Services) IoT, Google Cloud Platform IoT, IBM Watson IoT, Azure IoT Suite, Thingworx, Xively, Zatar. (del Valle Hernández)

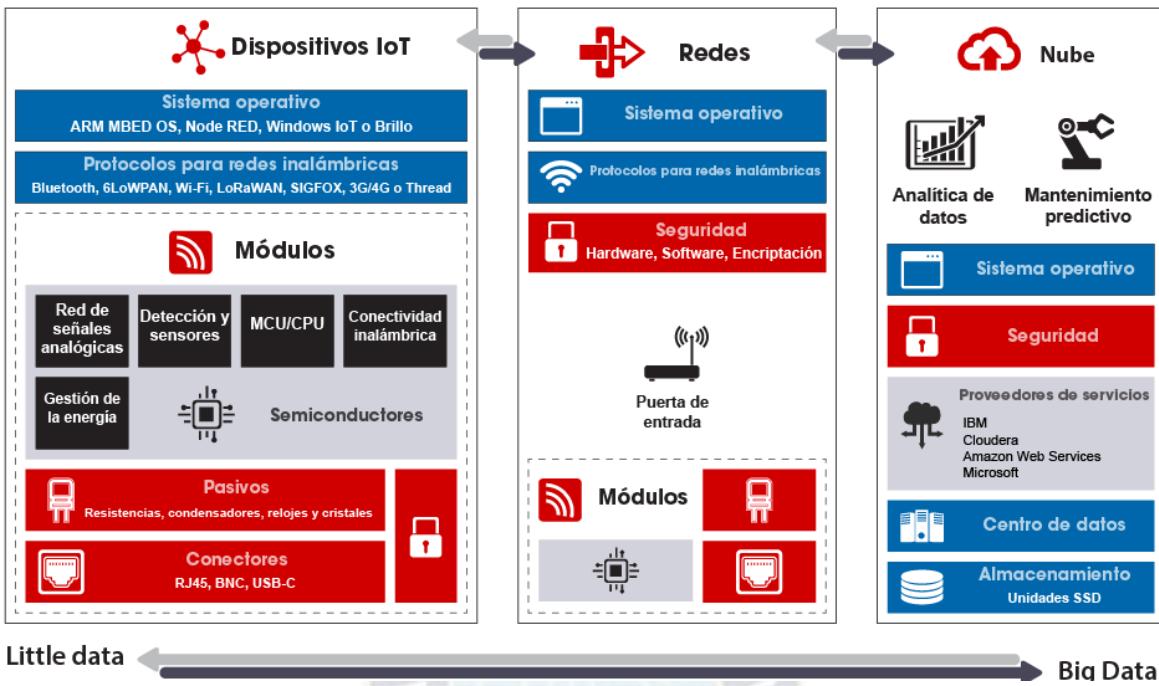


Figura 24: Dominios desagregados de la arquitectura IoT

Fuente: (RS Online: Internet de las Cosas)

2.6. TECNOLOGÍAS DE DESARROLLO WEB

Una de las mejores prácticas dentro del desarrollo de aplicaciones Web, se encuentra la utilización de Frameworks de desarrollo. Un Framework es un conjunto de herramientas prefabricadas para un lenguaje de programación específico. Herramientas que se encuentran desarrolladas y plenamente optimizadas.

Los frameworks en tiempo real ofrecen mejores alternativas para IoT que supone una gran cantidad de dispositivos conectados al mismo tiempo. Para tal objetivo, se manejan operaciones multihilo, es decir, cuando se manejan grandes cantidades de peticiones concurrentes, en lugar de dejar un proceso bloqueado mientras espera su finalización, la aplicación puede comenzar el proceso y darle una llamada de retorno (callback) que notificará a la aplicación cuando aquel proceso se complete; deja el bucle de entrada-salida disponible para servir a otros clientes.

2.6.1. Stack WAMP

WAMP es un acrónimo que describe a una infraestructura de desarrollo que utiliza a las siguientes herramientas:

- Windows – Como sistema operativo base
- Apache – Como servidor web
- MySQL – como gestor de base de datos
- PHP – como lenguaje de desarrollo.

Este paquete de herramientas es más conocido como LAMP, donde la “L” representa a Linux, que es el sistema operativo más seguro y con el que muchos servidores web en el mundo funcionan como sistema operativo base, debido a que es *open source* (de código abierto), por lo que no se paga una licencia por utilizarlo.

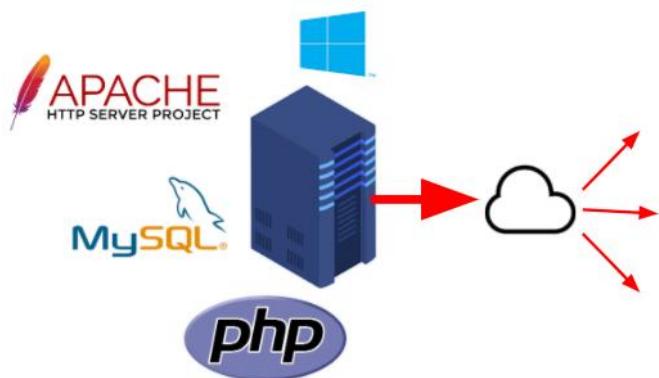


FIGURA 25: COMPONENTES DE UN SERVIDOR WAMP
Fuente: Elaboración Propia

2.6.1.1. Apache

Un servidor web es un sistema de computación que procesa peticiones HTTP. El protocolo básico de red utilizado para distribuir información en la World Wide Web. La función primaria de un servidor web es guardar, procesar y mostrar páginas web a los clientes. La comunicación entre el cliente y el servidor toma lugar usando el Protocolo de Transferencia de Hipertexto (HTTP por sus siglas en inglés), las páginas mostradas, son frecuentemente documentos HTML que pueden incluir imágenes, hojas de estilo, y código JavaScript en conjunción con el contenido de los textos.

Apache es un servidor web HTTP de código abierto, desarrollado por la Apache Web Foundation. Gracias a esta comunidad de desarrolladores que tiene miembros por todo el mundo, goza de un soporte inmenso.

Apache es uno de los servidores web más estable del mercado. Junto a su capacidad de modularidad, su facilidad de adaptación multiplataforma y su popularidad lo hacen el más utilizado en todo el mundo.

Su estructura se basa en un núcleo (core), y varios módulos (mod_ssl para seguridad, mod_rewrite para reescribir direcciones, mod_jk para enlazar tomcat, etc) que lo complementan.

2.6.1.2. MySQL

MySQL es un sistema de gestión de bases de datos relacional, considerada como la base de datos *open source* (de código abierto) más conocida en el mundo. Para interactuar con MySQL es necesario utilizar el lenguaje SQL (Structured Query Language), el mismo lenguaje que utilizan las bases de datos de Oracle y Microsoft SQL Server. Entre estas tres, dominan el mercado de las bases de datos relacionales, principalmente para el desarrollo web.

Las principales ventajas de MySQL son la velocidad y robustez. Esta soporta una gran cantidad de tipos de datos para formatear las columnas. Otra de las grandes fortalezas es que su portabilidad, le permite trabajar en distintas plataformas y sistemas operativos.

Cada base de datos, cuenta con 3 archivos: Uno de estructura, uno de datos y otro de índices. Soporta hasta 32 índices por tabla. Implementa el procesamiento por hilos, por lo que se le considera un sistema multiprocesos. Es bastante flexible en cuanto a las contraseñas y la gestión de usuarios, mantiene un buen nivel de seguridad en los datos. Soporta varios lenguajes para mostrar los mensajes de error.

En resumen sus mayores ventajas son:

- Alta velocidad al realizar operaciones.
- Bajo costo en requerimiento para la creación de bases de datos
- Sencilla configuración e instalación

- Soporte multiplataforma y en varios sistemas operativos.
- Baja probabilidad de corrupción de los datos, a pesar de que la plataforma pueda corromperse, la información no.
- Ampliamente usado por su conectividad y la seguridad de la información.

2.6.1.3. PHP

El lenguaje de Pre Procesamiento de Hipertexto (PHP por sus siglas en inglés), es un lenguaje de programación de código abierto, originalmente del lado del servidor, diseñado para el desarrollo web de contenido dinámico. El servidor web procesa el código y posteriormente genera el archivo HTML (página web) que se muestra al cliente.

PHP ha evolucionado bastante, tanto que se puede crear aplicaciones gráficas independientes mediante línea de código y su interfaz de línea.

Debido a su categoría de uso libre (open source), es conocido y utilizado por millones de desarrolladores en el mundo, sin que esto signifique algún costo.

PHP a diferencia de JavaScript, se ejecuta del lado del servidor, motivo por el cual, se lo considera bastante seguro.

Su compatibilidad con todos los sistemas operativos y los servidores web al igual que su facilidad para crear imágenes, PDF, flash, manipular documentos XML y XHTML, lo hace el lenguaje de programación ideal para desarrollar cualquier entorno web.

2.6.2. Stack MEAN

El lenguaje JavaScript se está haciendo un hueco cada vez mayor en el mercado del desarrollo de software. Aquel humilde lenguaje que empezó en los años '90 como una vía sencilla de validar formularios, se ha convertido en parte fundamental del desarrollo de todo tipo de aplicaciones: web, móviles, bases de datos, administración de sistemas...

Esta proliferación ha llevado a JavaScript a todas las capas de desarrollo, empezando por el lado cliente en sus inicios (el navegador), pero yendo también al servidor y a la capa de

almacenamiento. En cualquiera de esos puntos podemos encontrar JavaScript listo para ser utilizado.

Gracias a eso, hoy en día es posible crear aplicaciones distribuidas utilizando el mismo lenguaje JavaScript en todas sus fases y capas. A este concepto y las tecnologías que lo posibilitan se les ha bautizado con el nombre de MEAN, acrónimo formado por las iniciales de las cuatro tecnologías principales que entran en juego: MongoDB, Express, Angular y Node.js (Alarcón, 2019)

- MongoDB: Para la Base de datos
- Express Framework: para el desarrollo del backend del sistema de información
- Angular: para el desarrollo del front-end del sistema de información
- Node.js: para el desarrollo del servidor de la aplicación.

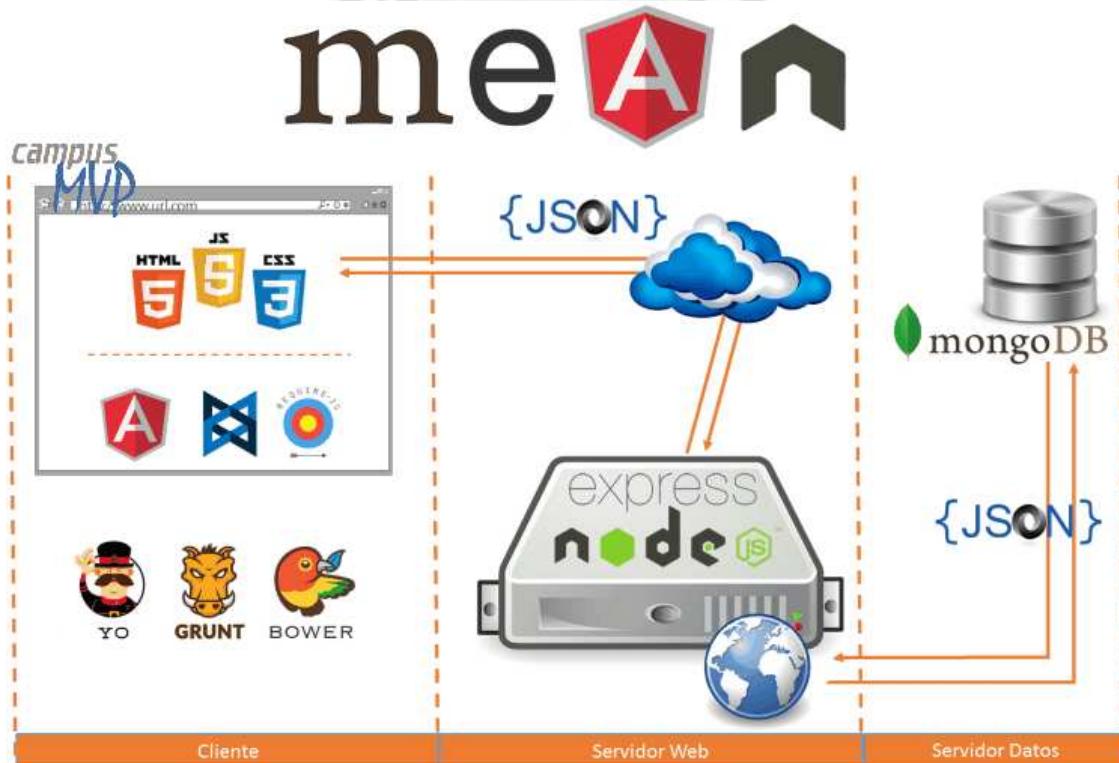


FIGURA 26: FUNCIONAMIENTO DEL STACK MEAN

FUENTE: (ALARCÓN, 2019)

2.6.2.1. MongoDB

MongoDB es un SGBD NoSQL de tipo documento, el cual usa documentos JSON. JSON es un formato para el intercambio de datos similar a XML pero su estructura es más simple. Un ejemplo de un documento JSON es:

```
{  
  "id": 1,  
  "nombre": "Juan",  
  "edad": 23,  
  "grado": 8,  
  "dep": 10  
}
```

FIGURA 27: EJEMPLO DE UN DOCUMENTO JSON

Fuente: (Moreno Arboleda, Quintero Rendón, & Rueda Vásquez, 2016)

Un documento JSON está encerrado entre llaves {}. En su interior hay parejas “clave”: valor, separadas por comas, donde valor puede ser un número, una cadena de caracteres, un documento JSON o un arreglo de valores (que incluso pueden ser documentos JSON). Si el valor es un arreglo, este se encierra entre corchetes [] y en su interior se colocan sus valores separados por comas.

En MongoDB los documentos JSON se agrupan en una colección. Una colección es el equivalente a una relación en un SGBD relacional. Los documentos JSON de una misma colección no se apegan necesariamente al mismo esquema. Un documento equivale a una fila de una relación y una clave de un documento equivale a un atributo de una relación. En la Figura siguiente se establece un paralelo entre los términos de una BD relacional y una BD en MongoDB.

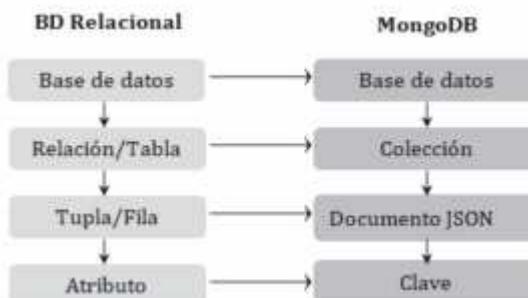


Figura 28: Comparación de términos entre una BD relacional y una BD orientada a documentos: MongoDB

Fuente: (Moreno Arboleda, Quintero Rendón, & Rueda Vásquez, 2016)

De la misma manera, en la siguiente tabla se muestran las principales diferencias en la forma general de las sentencias básicas de definición del esquema de datos.

Operación	SQL	MongoDB
Creación de una BD	CREATE DATABASE nombreBD;	USE nombreBD
Creación de tabla (SQL)/Colección (MongoDB)	CREATE TABLE nombreTabla (atributo1 tipo_de_dato restricciones);	db.createCollection("nombreColección")
Creación de Indice	CREATE INDEX nombreIndice ON nombreTabla (atributo1);	db.nombreColeccion.ensureIndex({"atributo": 1})
Destrucción de tabla (SQL)/Colección (MongoDB)	DROP TABLE nombreTabla;	db.nombreColeccion.drop()
Inserción	INSERT INTO nombreTabla VALUES (valorAtributo1, <, valorAtributo2...>);	db.nombreColeccion.insert({ atributo1: valorAtributo1 <, atributo2: valorAtributo2, ...})
Actualización	UPDATE nombreTabla SET atributo = nuevoVa	db.nombreColeccion.update({<condición>}, {\$set: {atributo: nuevoValor} }<, [true false],[true false]>)
Borrado	DELETE FROM nombreTabla <WHERE condición>	db.nombreColeccion.remove({<condición>})
Búsqueda	SELECT <DISTINCT> [*atributo1 <, atributo2...>] FROM relación1 <, relación2...> WHERE condición GROUP BY atributos HAVING condición de grupo;	db.collection.find(<condición> <, proyección>)

Tabla 3: Formas generales de las sentencias básicas de definición del esquema en SQL y en MongoDB

Fuente: (Moreno Arboleda, Quintero Rendón, & Rueda Vásquez, 2016)

En particular, la función `ensureIndex()` de MongoDB recibe como parámetro una pareja “clave”: valor. La clave indica el atributo que se indexará, y el valor puede ser 1 o -1. El 1 indica que el orden es ascendente, y el -1 que es descendente.

En MongoDB la sentencia `insert` recibe como parámetro un documento JSON. La clave `id` equivale a la clave primaria en el modelo relacional; si no se indica MongoDB, la asigna automáticamente.

En las búsquedas, La función `find()` recibe dos parámetros: una condición y una proyección. La condición es equivalente a la cláusula WHERE de SQL pero aplicada a los documentos. La proyección es equivalente a la sentencia SELECT de SQL, allí se indican los atributos (claves en MongoDB) que se desean seleccionar. En la siguiente tabla se presentan ejemplos de consultas en SQL y en MongoDB. (Moreno Arboleda, Quintero Rendón, & Rueda Vásquez, 2016)

SQL	MongoDB	Resultado en MongoDB
<code>SELECT * FROM empleado WHERE edad < 30 AND grado = 8;</code>	<code>db.empleado.find({edad: {\$lt: 30}, grado: 8})</code>	<code>[{"_id": 1, "nombre": "Juan", "edad": 23, "grado": 8, "dep": 10}]</code>
<code>SELECT * FROM empleado WHERE nombre IN ('Pablo', 'Ana');</code>	<code>db.empleado.find({nombre: {\$in: ["Pablo", "Ana"]}})</code>	<code>[{"_id": 1, "nombre": "Juan", "edad": 23, "grado": 8, "dep": 10}]</code>
<code>SELECT * FROM empleado WHERE nombre LIKE '%blo';</code>	<code>db.empleado.find({nombre: /blo/})</code>	<code>[{"_id": 2, "nombre": "Pablo", "edad": 25, "grado": 10, "dep": 10}]</code>
<code>SELECT SUM(edad) AS total FROM empleado;</code>	<code>db.empleado.aggregate([{\$group: {_id: null, total: {\$sum: "\$edad"}}}])</code>	<code>[{"_id": null, "total": 69}]</code>
<code>SELECT dep, MAX(grado) AS mg FROM empleado GROUP BY dep;</code>	<code>db.empleado.aggregate({\$group: {_id: "\$dep", mg: {\$max: "\$grado"}}})</code>	<code>[{"_id": 10, "max": 10}, {"_id": 20, "mg": 6}]</code>

Tabla 4: Ejemplos de las consultas en SQL y en MongoDB.
Fuente: (Moreno Arboleda, Quintero Rendón, & Rueda Vásquez, 2016)

2.6.2.2. Express Framework

Express es el framework web más popular de Node, y es la librería subyacente para un gran número de otros frameworks web de Node populares. Proporciona mecanismos para:

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

Express fue lanzado inicialmente en Noviembre de 2010 y está ahora en la versión 4.16 de la API.

Express es un framework popular, de acuerdo con el número de compañías de perfil alto que lo usan, el número de gente que contribuye al código base, y el número de gente que proporciona soporte tanto libre como pagado.

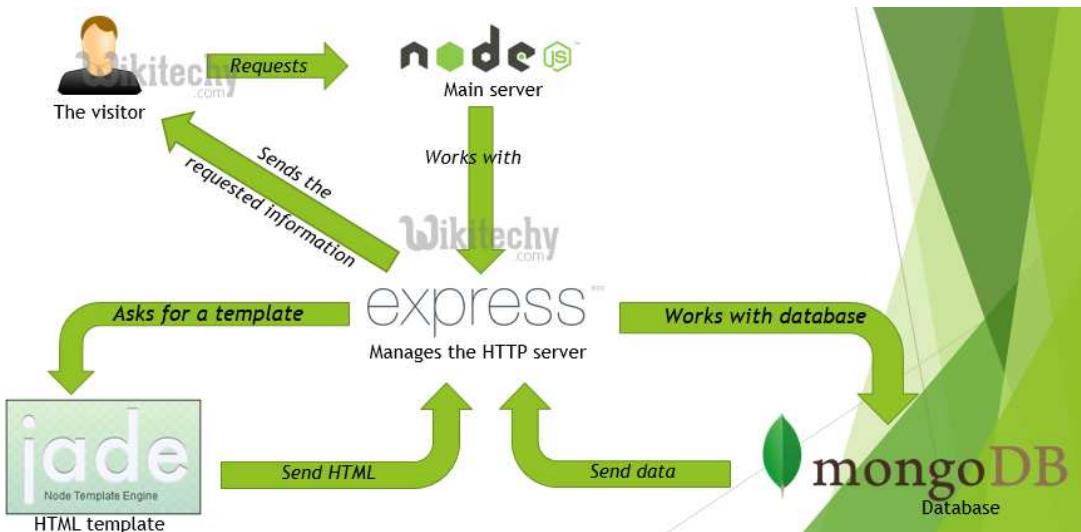


FIGURA 29: FUNCIONAMIENTO DE EXPRESS FRAMEWORK.

FUENTE: (WIKITECHY.COM, 2019)

A pesar de que Express es en sí mismo bastante minimalista, los desarrolladores han creado paquetes de middleware compatibles para abordar casi cualquier problema de desarrollo web. Hay librerías para trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, datos POST, cabeceras de seguridad y muchos más. Puedes encontrar una lista de paquetes middleware mantenida por el equipo de Express en Express Middleware (junto con una lista de algunos de los paquetes más populares de terceros).

Express es no dogmático, transigente. Puedes insertar casi cualquier middleware compatible que te guste dentro de la cadena de manejo de la petición, en casi cualquier orden que te apetezca. Puedes estructurar la app en un fichero o múltiples ficheros y usar cualquier estructura de directorios. (mozilla, 2019)

2.6.2.3. Angular Framework

AngularJS es un framework MVC de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA Single-Page Applications. Entra dentro de la familia de frameworks como BackboneJS o EmberJS.

Anteriormente en la parte Front-End de las aplicaciones web sólo teníamos a jQuery (además de otras librerías parecidas como Mootools, Prototype,...) para ayudarnos con el código JavaScript del cliente. Podíamos manipular el DOM de una forma más sencilla, añadir efectos, llamadas AJAX, etc, pero no teníamos un patrón a seguir. Todo el código JS iba en funciones que íbamos creando según necesitáramos, lo que provocaba que con el tiempo el código fuera difícilmente manejable y se convirtiese en el temido Spaghetti Code.

Por suerte surgieron frameworks que implementaban el patrón MVC (Modelo, Vista, Controlador) y nos ayudaban a separar conceptos. El más conocido es BackboneJS, que surgió en 2010 creada por Jeremy Ashkenas (Creador también de CoffeeScript) y depende de otras 2 librerías: jQuery y Underscore.js Es usado por múltiples Start-ups como Pinterest, Foursquare, AirBnB, Trello, etc.

BackboneJS te permite crear tu app rápidamente aunque en ocasiones es complicado de utilizar. La mayoría de los desarrolladores eligen BackboneJS porque parece la opción más segura, lleva más tiempo entre nosotros, hay mucha documentación sobre él y está mantenido por una gran comunidad.

Sin embargo AngularJS está pisando fuerte. Aunque su primera versión es de 2009, se ha hecho muy popular a finales de 2012 y ahora en 2013 está en pleno auge. Tanto que ya se habla de una nueva *technology stack* como antes era LAMP (Linux + Apache + MySQL + PHP) ahora la tendencia es MEAN (MongoDB/Mongoose + ExpressJS + AngularJS + NodeJS), lo que también se traduce a aplicaciones JavaScript End-to-End. AngularJS está mantenido por Google y bastante comunidad. También como punto a su favor está lo sencillo que crear Tests unitarios y End-to-End con Jasmine y Karma, algo que suele ser un poco costoso al principio.

AngularJS permite extender el vocabulario HTML con directivas y atributos, manteniendo la semántica y sin necesidad de emplear librerías externas como jQuery o Underscore.js para que funcione. (carlosazaustre.es, 2019)

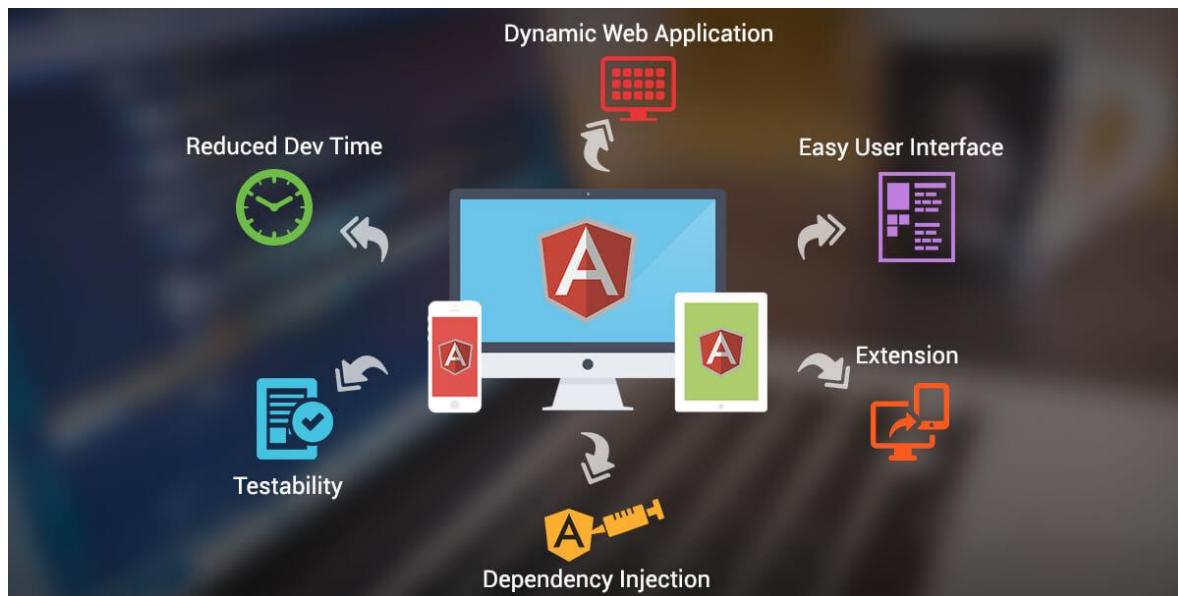


FIGURA 30: FUNCIONAMIENTO DE ANGULAR FRAMEWORK.

FUENTE: (MINDINVENTORY.COM, 2019)

2.6.2.4. Node

Node.js es un framework que utiliza un modelo no bloqueante, orientado a eventos, que lo hace ligero y eficiente. Una de sus cualidades es su capacidad nativa de trabajar con web sockets. Node.js es la plataforma web más popular para aplicaciones en tiempo real y por ende es una excelente alternativa para IoT, además cuenta con una gran comunidad que desarrolla cada vez más librerías que se incorporan con facilidad en cualquier proyecto. Actualmente ya se disponen incluso de librerías IoT para open hardware como: Arduino, Raspberry, Beaglebone, entre otros. Node funciona con JavaScript y ha evolucionado hasta ser utilizado en dispositivos embebidos, pudiendo llegar a convertirse en un lenguaje de programación para aplicaciones de servidor, cliente (navegador web) y dispositivos embebidos. (Campoverde, Hernandez, & Mazón, 2015)

Node.js está escrito en JavaScript. JavaScript es un lenguaje de programación ampliamente utilizado para la programación de las interfaces de los sitios web, por otro lado Node.js permite la aplicación de JavaScript en más contextos y particularmente en servidores web.

Al igual que el código fuente de JavaScript del lado del cliente es interpretado y ejecutado por un motor JavaScript, Node.js utiliza un intérprete rápido V8 JavaScript, escrito en C++ creado por Google, subyacente en su browser Google Chrome.

El intérprete V8 JavaScript, podría ser descargado e incorporado en las aplicaciones, sin tener ningún tipo de restricción en la ejecución del mismo, en otros navegadores. (Cangás Cangás, 2015)

Por ejemplo, PayPal redujo significativamente el tiempo de respuesta en un 35%, manejando el doble de la cantidad de solicitudes por segundo utilizando Node.js.

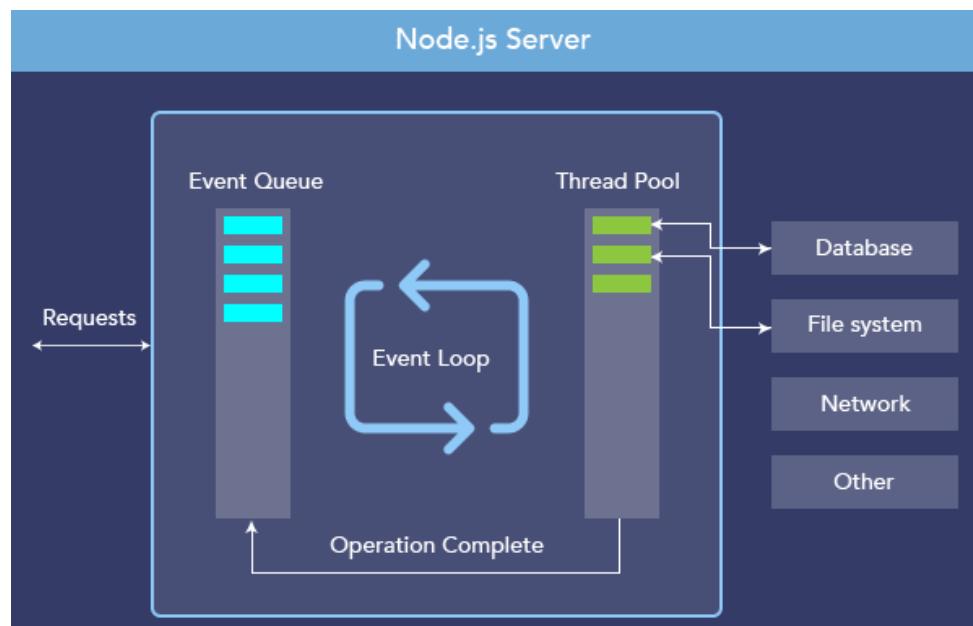


FIGURA 31: FUNCIONAMIENTO DE UN SERVIDOR NODE
Fuente: (peerbits, 2019)

CAPÍTULO 3. INGENIERÍA DEL PROYECTO

3.1. ESTABLECIMIENTO DE FASES DEL PROYECTO

Para el presente trabajo, se definieron algunas etapas, conforme a la administración de procesos de desarrollo de software basado en algunos de los principios del Scrum (Mendes Calo, Estevez, & Fillottrani, 2010) :

- **Fase 1: Recolección de necesidades de usuario.** Que defina las entradas, salidas y procesos intermedios del sistema embebido a desarrollar, y que además permita definir el hardware a utilizarse. Se define un diseño conceptual del sistema para identificar los principales actores y la relación entre los mismos.
- **Fase 2. Elección de tecnologías a utilizar.** En base a diferentes criterios que se definirán en este punto, se seleccionará el hardware y software que encaje de mejor manera a los requerimientos del sistema.
- **Fase 3: Realización de Sprints:** Se define el alcance y la duración del Sprint. En cada Sprint se hace una entrega parcial de los avances del proyecto.
- **Fase 4: Entrega de Código y retroalimentación.** Se hace la entrega del código y se proporciona una retroalimentación sobre la manera en que se llevó a cabo el trabajo y si hay alguna oportunidad de mejora.

3.1.1. Recolección de necesidades de usuario

Se realizó una reunión con el Sr. Walter Quiroga, miembro del Grupo Gundlach y Daniel Quiroga, Jefe de la División de Purificación de Aguas. Tanto Walter como Daniel (denominados en adelante Clientes), presentaron las siguientes ideas en torno al proyecto:

- Requieren encontrar los medios tecnológicos para que la División de purificación de Aguas, aplique políticas en favor del cambio climático.
- Se desea adquirir algún prototipo que reduzca la utilización de material plástico en la comercialización de agua.

- Se planteó la idea de un dispensador de líquidos, que podrá distribuir una cantidad de agua en función a la cantidad de monedas que acepte.
- Este dispensador no proporcionará envases donde recibir el producto, sino que pedirá al usuario que tenga su envase reciclado.
- El dispensador proporcionará agua de mesa de alta calidad a precios económicos, debido a que se evitará el costo de la botella plástica, denominada “de un solo uso”.
- El dispensador utilizará 4 tanques del líquido.
- Se dispone de Bs1.400 para el sistema de control de un dispensador, sin incluir el armazón exterior.

A partir de esta reunión, se analizaron los posibles métodos para cumplir con las ideas anteriormente señaladas.

En una segunda reunión, junto con los Clientes, y el Ing. Ricardo Gotrett (Product Owner en adelante), se definió realizar un prototipo de dispensador de líquidos basado en Internet de las Cosas (IoT) con las siguientes características:



FIGURA 32: RESUMEN DEL SISTEMA IoT EN FUNCIONAMIENTO.
FUENTE: ELABORACIÓN PROPIA

- El dispensador proporcionará un volumen de líquido, definido por el cliente, a cambio de una moneda de un corte específico, mediante un motor que tendrá acceso al tanque de líquido.
- Para controlar el volumen y la periodicidad de cambio de los tanques, el contenedor tendrá conexión a internet, a un servidor, que recogerá la información de consumo de agua.
- Se comenzará con un prototipo y otros 4 dispensadores más, para iniciar con la red de dispensadores de agua.
- Se desarrollará una aplicación web cuyo uso objetivo será visualizar un tablero informativo con datos generados por el consumo de los dispensadores, para que el cliente pueda tomar decisiones en base a la información mostrada.

En base a esta información, se elaboró un diagrama de flujo del proceso en general:

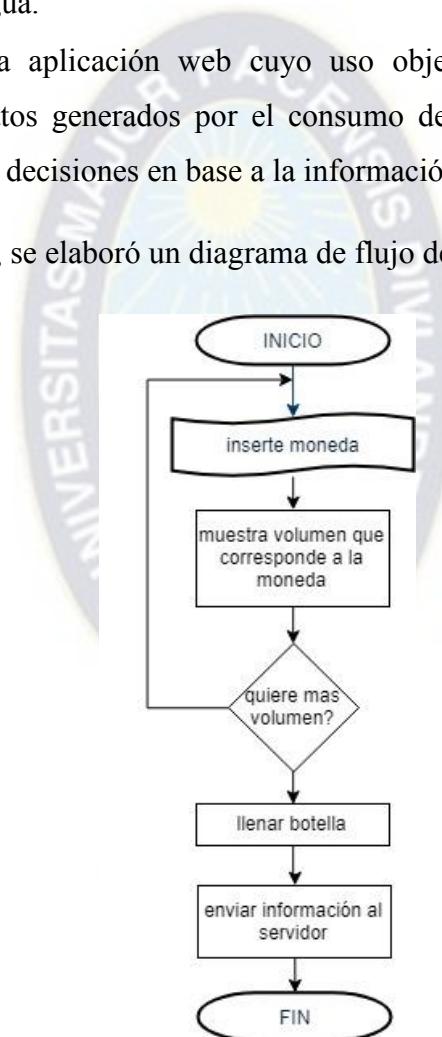


Figura 33: Diagrama de Flujo general del proceso del Dispensador.

Fuente: Elaboración propia

En la siguiente tabla, se describen los diferentes procesos del diagrama de flujo:

Etapa del flujo	Descripción
Inserte moneda	Es el mensaje de inicio que el dispensador estará mostrando mientras no realiza ninguna tarea, se espera la introducción de una moneda, de un corte previamente acordado con el Cliente, que debe ser leído por el sistema embebido que controle el dispensador.
Muestra el volumen que le corresponde a la moneda	El sistema embebido, realiza la operación para convertir la moneda ingresada en la cantidad de volumen que será mostrado en pantalla (dicho volumen debe ser previamente definido por el Cliente), para que el Usuario, pueda decidir posteriormente si desea más liquido o es suficiente.
¿Quiere más volumen?	Si el Usuario desea más volumen del líquido deberá ingresar más monedas, de lo contrario, deberá presionar un botón denominado “llenar botella” el cual será leído por el sistema embebido únicamente cuando se hayan insertado monedas previamente.
Llenar Botella	<p>Se deberá calcular previamente, el tiempo que le toma expedir el volumen requerido al sistema embebido, con base a las características del motor que succione el líquido del tanque, en la ubicación que se encuentre. Asimismo, el sistema deberá prever que el tanque pueda abastecer la capacidad que el Usuario pide. De no ser así, el sistema deberá enviar al Usuario una advertencia de no ingreso de más monedas, debido a la falta de líquido.</p> <p>Por otro lado, si el usuario retira de alguna manera la botella en medio del proceso de llenado, el sistema deberá ser capaz de considerarlo, y parar el proceso de llenado. Además de emitir una alerta, sobre este cambio al Usuario.</p>
Enviar información al servidor	Una vez terminado el proceso de llenado, el sistema deberá enviar la información de la cantidad consumida y nivel del tanque al finalizar el servicio a un servidor vía internet, con la finalidad de poder generar estadísticas y alertas de baja capacidad al Cliente.

TABLA 5: TABLA DESCRIPTIVA DE TODO EL PROCESO DEL DISPENSADOR.

FUENTE: ELABORACIÓN PROPIA

Con esta información, se realiza la investigación de las plataformas de desarrollo de sistemas embebidos, y posteriormente la elección de la más óptima para el proyecto.

3.1.2. Elección de tecnologías a utilizar

3.1.2.1. Plataforma de Hardware

De acuerdo a las especificaciones anteriores, se realizó el diagrama del tablero de control y los elementos a controlar por el sistema embebido, para determinar el número de puertos a utilizarse.

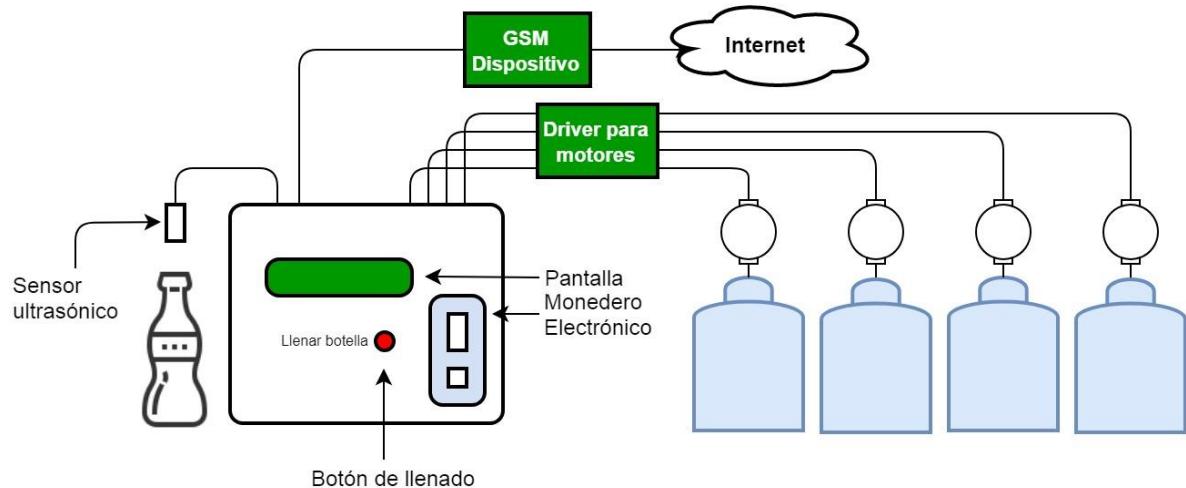


FIGURA 34: DIAGRAMA DE CONTROL DE DISPENSADOR
FUENTE: ELABORACIÓN PROPIA

A continuación, se describen los elementos que se muestran en el Diagrama de control del Dispensador y los puertos que se deben utilizar:

Elemento	Descripción	Tipo de Conexión	Puertos a Utilizar	Observaciones
Sensor ultrasónico	Para la detección de la presencia de la botella a la hora del llenado		2	un pin de envío de señal ultrasónica un pin de recepción de señal ultrasónica
Pantalla LCD 16*2	Para mostrar los mensajes al Usuario	I2C	2	2 pines de comunicación I2C
Monedero electrónico	Para clasificar y detectar la moneda ingresada		1	1 pin de lectura de monedero
Dispositivo GSM	para realizar la conexión con el servidor	Serial	3	un pin de envío un pin de recepción un pin de reinicio del Modem GSM
Driver para 4 motores	para controlar el líquido que sale de los tanques		4	4 pines de activación de motores
Botón de llenado	Para que el Usuario inicie el llenado de Botella		1	1 pin de lectura de botón
Botón de reinicio	Para que el sistema pueda ser reiniciado		1	1 pin de lectura de botón
LED señalizador de operación	Para identificar el presionado de los botones		1	1 pin de escritura
Pin de Inhabilitación de sistema	Para cuando el líquido sea insuficiente para distribuir al Usuario		1	1 pin de escritura
TOTAL ENTRADAS/SALIDAS			16	

TABLA 6. PUERTOS A UTILIZAR DEL MICROCONTROLADOR

FUENTE: ELABORACIÓN PROPIA

Se ha visto por conveniente utilizar un microcontrolador, ya que puede satisfacer ampliamente los requerimientos, en vez de una placa de desarrollo basada en un microprocesador, como una *Raspberry* por ejemplo.

Se ha realizado una tabla comparativa entre los diferentes microcontroladores de las marcas Atmel y Microchip que existen en el mercado boliviano, para identificar al que mejor se adapta a nuestras necesidades.

microcontrolador	Fabricante	Memoria Flash (Kb)	RAM (Bytes)	Reloj (Mhz)	interfaz I2C	RS232	Pines de E/S	Precio (Bs)
PIC16F84A	Microchip	2	68	hasta 20 MHz	NO	SI	13	25
PIC16F628	Microchip	4	224	hasta 20 MHz	NO	SI	16	25
PIC16F877	Microchip	14	368	hasta 20 MHz	SI	SI	33	35
PIC18F2550	Microchip	32	2048	hasta 48 MHz	SI	SI	24	40
PIC18F4550	Microchip	32	2048	hasta 48 MHz	SI	SI	35	40
AT89C51	Atmel	4	128	hasta 24 MHz	NO	SI	32	20
AT98C52	Atmel	8	256	hasta 24 MHz	NO	SI	32	22
ATMEGA32A	Atmel	32	2048	hasta 16 MHz	NO	SI	32	48
ATMEGA328P	Atmel	32	2048	hasta 16 MHz	SI	SI	23	25
ATMEGA8	Atmel	8	1024	hasta 16 MHz	NO	SI	23	20

TABLA 7. TABLA COMPARATIVA DE MICROCONTROLADORES EN RELACIÓN A LOS REQUERIMIENTOS.
FUENTE: ELABORACIÓN PROPIA

Finalmente, se han enmarcado los microcontroladores que cumplen con los requerimientos anteriores. Si bien los 4 pueden cumplir las funciones de forma eficiente, resulta más sencillo

trabajar con un microcontrolador Atmel, por lo que se utilizará el Atmega328P para desarrollar el sistema. No obstante, si por algún motivo los requerimientos aumentaran, se optará por recodificar para el PIC16F877.

3.1.2.2. Plataforma de Software

Los dispensadores anteriores deben comunicarse con una plataforma que permita gestionar los datos que se generen del consumo de líquidos con el Dispensador.

- **Plataforma de recolección de información IoT:**

ThingSpeak es una plataforma IoT que permite a sus usuarios recoger y almacenar datos y desarrollar aplicaciones IoT. Pertenece a MathWorks y ofrece aplicaciones que permiten analizar y visualizar los datos en MATLAB. Puede almacenar datos enviados desde dispositivos Arduino, Raspberry y Beaglebone, entre otros más. Crear una cuenta en ThingSpeak es totalmente gratuito y es posible mejorar las cualidades de la versión inicial mediante la adquisición de una versión de pago (Castillo Izquierdo, 2017)

Se utilizará esta plataforma para el presente proyecto, dadas las cualidades que ofrece para proyectos basados en IOT, ya que se encuentra desarrollada bajo el stack MEAN, lo que le proporciona la rapidez necesaria para actualizar los datos de forma constante y veloz.

- **Tablero de Control:**

Se realizará un Tablero de Control o *Dashboard*, mediante el cual se visualizará la información de los dispensadores. Dicho *Dashboard* se realizará en base a PHP y MySQL, dada la sencillez de estas herramientas y la aplicación que se le dará a las mismas.

Dicho sistema tendrá una apariencia similar a la figura siguiente:



FIGURA 35: MODELO DE DASHBOARD PARA EL SISTEMA IOT DE DISPENSADORES.

FUENTE: (MUÑOZ, 2014)

3.1.3. Realización de Sprints

Se definieron nuevas reuniones o Sprints cada 3 semanas, con el propósito de realizar el seguimiento y cumplimiento de los requerimientos.

3.1.3.1. Sprint 1

En este primer Sprint se presentó el funcionamiento básico del Dispensador al cliente (sin contar con el envío de información al servidor). Se aplicó el diagrama de Flujo inicial y se habló sobre la forma de identificar el tanque que se encuentre con un nivel bajo de líquidos.

- Identificación del nivel de líquidos:

Se trató el tema luego de que se pensara en la utilización de un sensor ultrasónico para determinar el nivel de líquido en cada tanque. Esto significaría utilizar 4 sensores ultrasónicos (uno para cada tanque) y, en consecuencia, 8 pines del microcontrolador.

Finalmente, se decidió contabilizar la cantidad de líquido que se emana por cada servicio, para luego restarla de la capacidad total del tanque.

- Alertas a cierto nivel de líquidos:

Se definió además un nivel de alerta (nivel que será definido por el cliente). En el momento en el que el tanque llegue a un nivel alerta, se cambiará al siguiente tanque.

- Reposición de un tanque vacío:

Si bien se analizó la lógica de cómo cambiar de tanque en un nivel de alerta, no se analizó la posibilidad de cómo indicar al microcontrolador, la reposición de un tanque por uno con más cantidad de líquido.

3.1.3.2. Sprint 2

En este segundo Sprint se analizaron y encontraron soluciones a los requerimientos del Cliente, tanto los planteados al principio del proyecto, como los que surgieron en el anterior Sprint.

- Identificación de la reposición de tanque:

Se incluyó al funcionamiento básico del dispensador, dos botones (reinicio y selección de tanque), que permitirán que el microcontrolador identifique la reposición de un tanque. Asimismo, para evitar cualquier contingencia durante el funcionamiento, se agregó un botón de reinicio de sistema.

- Identificación del nivel de líquidos por tanque:

El microcontrolador, ahora utiliza su memoria interna EEPROM, para almacenar la cantidad de líquido de cada tanque, cuyo valor va reduciendo conforme el tanque sea utilizado.

- Comunicación con la plataforma ThingSpeak:

Se utilizó una placa de entrenamiento basada en la SIM900, un circuito integrado que posee las capacidades de ser utilizado como un celular.



FIGURA 36: PLACA DE DESARROLLO SIM900
FUENTE: (BANGGOOD)

Cabe destacar que, para comunicar esta placa con nuestro microcontrolador, es necesario utilizar comandos AT.

A partir de esta placa de comunicación, se pudo ingresar datos a ThingSpeak los datos de prueba que se muestran a continuación.



Figura 37: Información que muestra ThingSpeak, en relación al proyecto
FUENTE: (THINGSPEAK)

3.1.3.3. Sprint 3

En este Sprint se presentó la interfaz de usuario del Dashboard de visualización de datos.

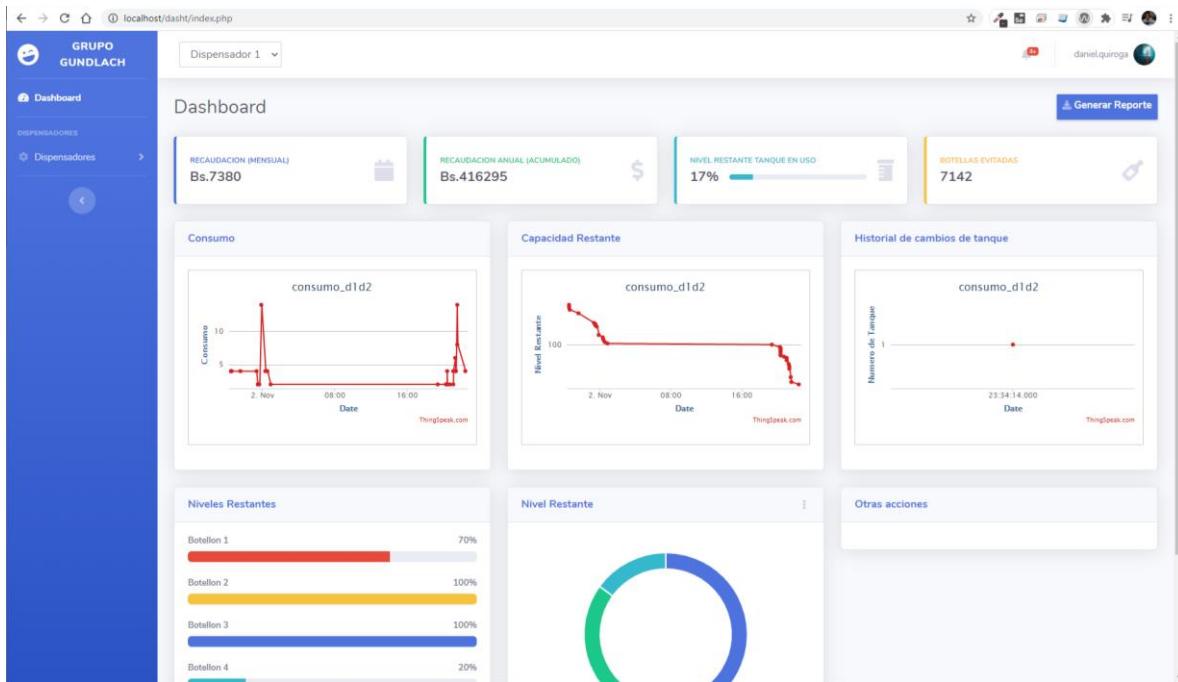


Figura 38: Tablero de control del dispensador
FUENTE: ELABORACIÓN PROPIA.

Básicamente este tablero de control tiene una primera interfaz de acceso, mediante la cual solo los usuarios autorizados pueden ingresar a visualizar los datos.

Acceder

Correo

Contraseña

Ingresar

Recuerdame

Olvidaste la contraseña?

Crear una nueva cuenta

Figura 39: Interfaz de acceso para los usuarios.
FUENTE: ELABORACIÓN PROPIA

Asimismo, se ha desarrollado un módulo de control de usuarios, en donde se diferencian dos roles: “admin” y “superadmin”. Los “superadmin” pueden gestionar la lectura, creación,

actualización o eliminación (CRUD por sus siglas en inglés) de otros usuarios con cualquiera de los roles anteriormente mencionados.

ID	Name	Admin type	Actions
10	admin	admin	 
9	daniel.quiroga	super	 
7	walter.quiroga	super	 
4	superadmin	super	 

Figura 40: Interfaz de gestión de usuarios.

FUENTE: ELABORACIÓN PROPIA

De esta manera, se permite al superadmin, actualizar o en algún caso, revocar los accesos de un usuario en específico.

localhost/dash/edit_admin.php?admin_user_id=10&operation=edit

Update User

agregar nuevo usuario admin

Nombre usuario

Contraseña

Tipo usuario

Super admin

Admin

Save

Figura 41: Interfaz de creación / actualización de usuarios.

FUENTE: ELABORACIÓN PROPIA

Asimismo, se ha utilizado Bootstrap para el diagramado de las interfaces, con el objetivo de que este *Dashboard* sea accesible y gestionable mediante el dispositivo móvil.

El detalle de los casos y diagramas de uso, se encuentran en el anexo.

3.1.3.4. Sprint 4

Si bien el diseño final de la estructura del dispensador no es parte de este proyecto, se plantearon algunas consideraciones para el mismo, y se hizo un borrador del posible dispensador de líquidos.

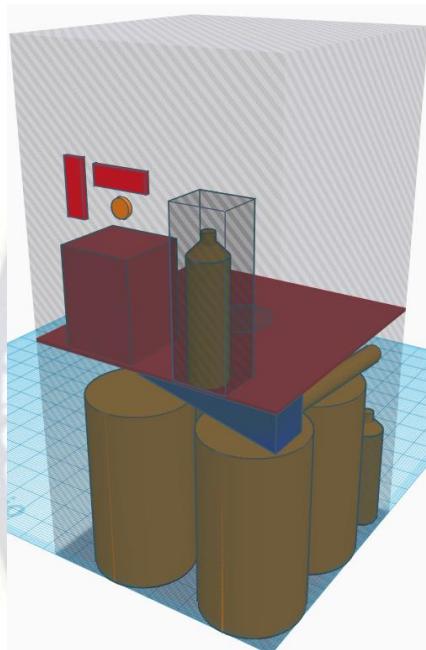


Figura 42: Estructura planteada al cliente para el dispensador de líquidos.

FUENTE: ELABORACIÓN PROPIA.

Se ha considerado el centro de gravedad del dispensador, la posición de la pila de dispensación y la posición del módulo GSM para el envío de información al servidor.

Esta propuesta fue bien recibida por el Cliente, motivo por el cual, se utilizó para elaborar el armazón que albergará al sistema de control. Se ha tomado en cuenta el rebalse que podría ocurrir si la manguera gotea mientras no haya una botella. Se ha incluido un recipiente que se sitúa debajo de la botella.

3.1.4. Entrega del prototipo

Finalmente se hizo la entrega del prototipo en funcionamiento, el cual se muestra a continuación.



Figura 43: Prototipo del Dispensador en funcionamiento
FUENTE: ELABORACIÓN PROPIA.

3.2. DESCRIPCIÓN TÉCNICA DEL PROTOTIPO

3.2.1. Circuito Base

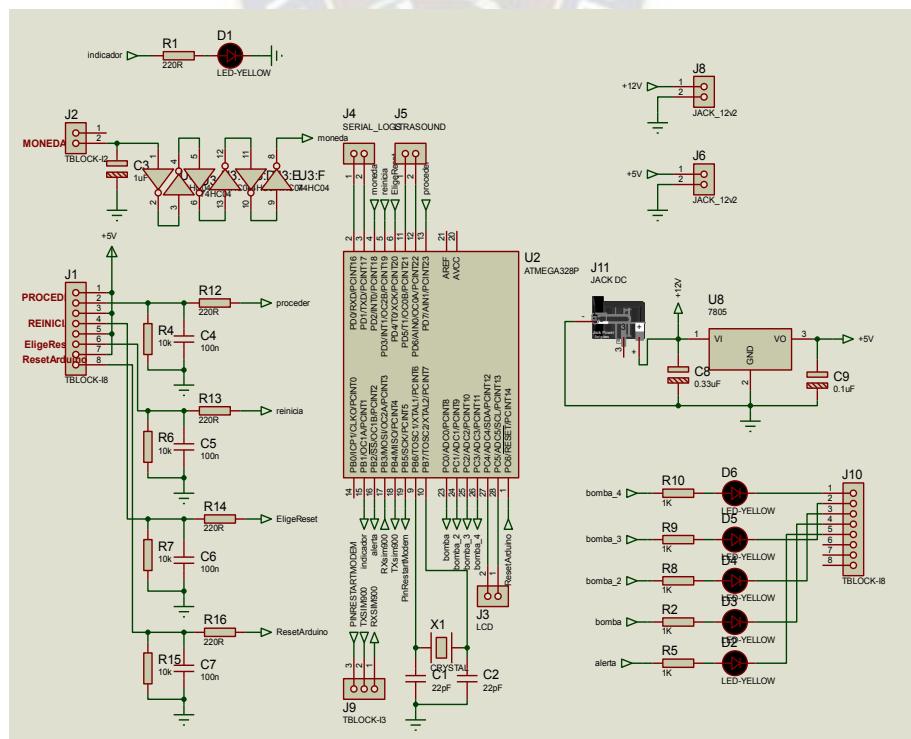


Figura 44: Circuito previo al final, utilizando el microcontrolador ATMEGA328P
FUENTE: ELABORACIÓN PROPIA.

En un principio se utilizó la placa de desarrollo Arduino uno, debido a su facilidad en el desarrollo del proyecto, posteriormente, se desarrolló la placa que utiliza el microcontrolador ATMEGA328P, para el desarrollo del prototipo.

Se utilizan los siguientes sensores y actuadores:

Dispositivo	Tipo	Utilidad
Botones	sensor	Iniciar el llenado Reiniciar el contenido de un tanque Reiniciar el circuito
Monedero	sensor	Detectar el ingreso de monedas
Sensor ultrasónico	Sensor	Detectar la presencia de la botella en el lugar indicado
Bomba de agua RS385	actuador	Dispensar el líquido
Placa sim900	actuador	Enviar información al servidor IoT
Chip móvil	actuador	medio físico para enviar información al servidor IoT
Pantalla LCD	actuador	Muestra información al usuario final

TABLA 8: SENSORES Y ACTUADORES DEL SISTEMA DE CONTROL DEL DISPENSADOR.
FUENTE: ELABORACIÓN PROPIA

3.2.1.1. Botones

Los botones utilizados son los siguientes:



Figura 45: Botones utilizados para la construcción del prototipo
FUENTE: WWW.AMAZON.ES.

Los mismos son botones de simple contacto. Se utilizaron este tipo de botones, debido a la sencillez de los mismos, además de su larga duración (4 años con uso diario de 16 horas, según la hoja de datos del fabricante).

Existen 5 funciones principales de los botones en el dispensador:

- **Proceder:** Sirve para iniciar el despachado de líquido del dispensador
- **ReiniciaModem:** Que sirve para reiniciar el funcionamiento de la tarjeta encargada de enviar la información de consumo del dispensador al servidor en la nube, si por algún motivo presentara alguna falla.
- **Reinicia:** Sirve para reiniciar el contador del tanque que se encontraba vacío y ha sido reemplazado, para realizar dicha tarea, trabaja en conjunción con el siguiente botón
- **EligeReset:** Con este botón se elige el tanque al que se le va a reiniciar el contador.
- **ReiniciaArduino:** Aunque ya no se utiliza Arduino, este botón se quedó con este nombre por su finalidad. Sirve para reiniciar el sistema de control, basado en el microcontrolador ATmega328P.

3.2.1.2. Monedero electrónico

El monedero utilizado es el de una sola moneda, dada su simplicidad y principalmente costo, ya que un monedero de estas características, cuesta casi la tercera parte de uno multimoneda.

Se utiliza un negador a la salida del pin de pulso de detección de moneda, para convertir el nivel (que se registra en 2,2V en un nivel lógico válido (0V – estado bajo y 5V – estado alto).

3.2.1.3. Sensor Ultrasónico

El sensor ultrasónico sirve para detectar la presencia de la botella. Al principio se pensó en utilizar hasta un contactor o un detector de presencia infrarrojo, debido a que ambos también pueden detectar la presencia de la botella, sin embargo, ninguna de éstas dos soluciones puede detectar a qué distancia se encuentra la boquilla de la botella y el pico del dispensador, con el objetivo de evitar el derrame de líquidos en cualquier lugar.

El dispensador solo provee del producto cuando la botella se encuentra a menos de 10 cm de distancia de la boquilla, caso contrario, se emite un mensaje en pantalla, que indica que se debe acercar más la botella.



Figura 46: Mensaje de pantalla cuando la botella se encuentra lejos del dispensador

FUENTE: ELABORACIÓN PROPIA

3.2.1.4. Bombas de Agua

Éste es uno de los elementos más importantes del sistema, debido a que son las bombas encargadas de generar el flujo de líquido para dispensar al usuario. De acuerdo al requerimiento del cliente, se utilizan 4 bombas, una para cada tanque.

Cada bomba es operada individualmente por el sistema de control. Cuando un tanque se encuentra dentro de los niveles críticos, se pasa a utilizar el siguiente tanque, y en consecuencia, la siguiente bomba.

El dispositivo utilizado se encuentra detallado en el punto 2.3.5.

Para mitigar el ruido eléctrico producido por las revoluciones efectuadas por este dispositivo, se hizo uso de un diodo 1n4148, un capacitor de 1000uF y un capacitor cerámico de 100nF

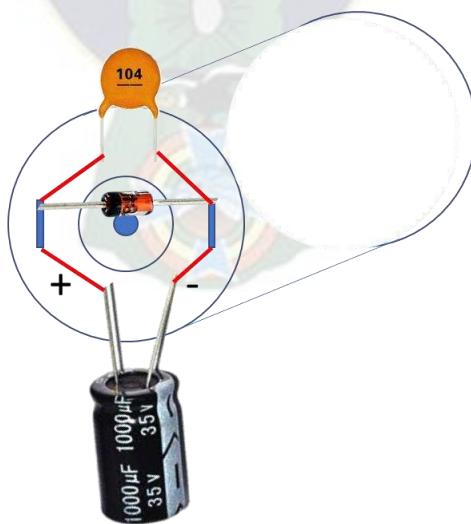


Figura 47: mitigación del ruido eléctrico del motor

FUENTE: ELABORACIÓN PROPIA

3.2.1.5.Pantalla LCD

Este elemento sirve principalmente para mostrar los mensajes al usuario final del dispensador. Se han programado 6 mensajes:

Listo para dispensar: Que indica la carga completa del sistema para poder dispensar líquido y además enviar información a la nube.



Figura 48: Mensaje de pantalla cuando el dispensador se encuentra listo para operar

FUENTE: ELABORACIÓN PROPIA

Líquido a dispensar: Una vez que se introduce una moneda, el sistema automáticamente muestra en pantalla el volumen correspondiente a esa moneda, en este caso, 300ml. Si se introduce otra moneda, el volumen incrementa de acuerdo a la cantidad especificada por el corte de moneda.

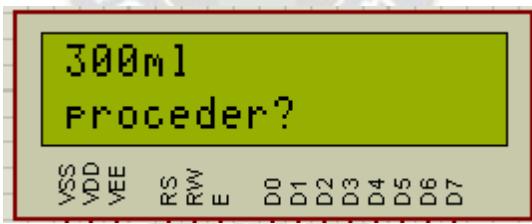


Figura 49: Mensaje de pantalla cuando el dispensador recibe la primera moneda

FUENTE: ELABORACIÓN PROPIA

Mensaje previo a la dispensación: Una vez elegido el volumen a ser adquirido del dispensador, se presiona el botón de proceder y a continuación el sistema muestra una serie de mensajes que preparan al usuario para colocar el recipiente en la boquilla del dispensador.

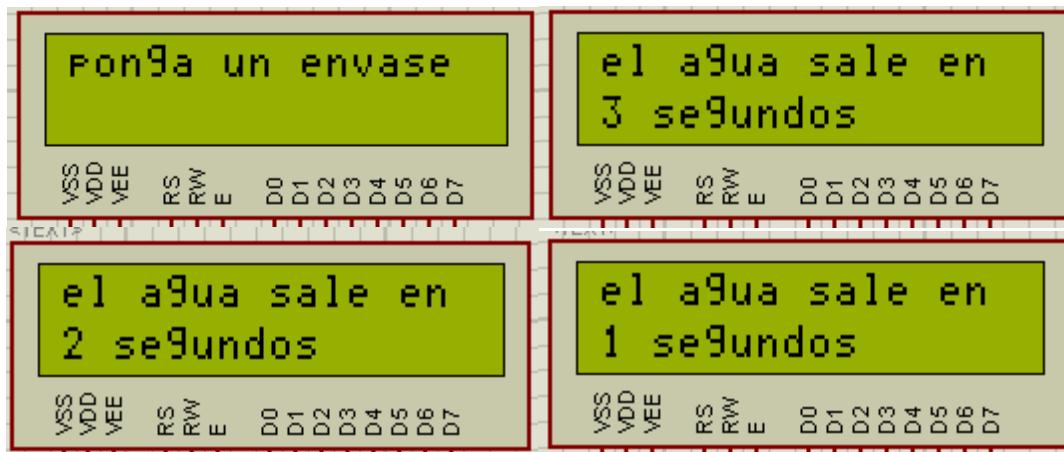


Figura 50: Mensajes de pantalla del dispensador para que el usuario se prepare
FUENTE: ELABORACIÓN PROPIA

Mensaje durante la dispensación: Se muestra un mensaje durante el proceso de llenado, para que el usuario sepa que debe mantener el recipiente bajo la boquilla del dispensador.

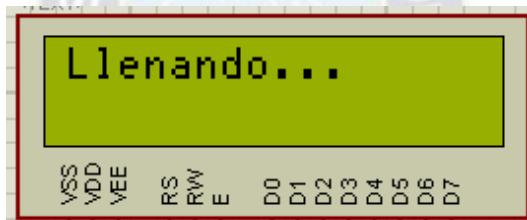


Figura 51: Mensaje de pantalla cuando el dispensador está en pleno proceso
FUENTE: ELABORACIÓN PROPIA

Mensaje de indisponibilidad: Se muestra en el caso de que los tanques se encuentren en un estado crítico, es decir, no tengan el suficiente volumen para dispensar un pedido.



Figura 52: Mensaje de pantalla cuando el dispensador se encuentra sin el líquido suficiente
FUENTE: ELABORACIÓN PROPIA

Cabe destacar que, usualmente, estas pantallas LCD necesitan de al menos 7 pines de conexión con el microcontrolador que se use (4 pines de datos y los pines RS, E, RW), sin embargo, para optimizar el uso de pines de este proyecto, se optó por trabajar con un adaptador basado en el PCF8574, que permite utilizar únicamente dos líneas de comunicación, vía protocolo I2C. Asimismo, se utiliza una librería, para utilizar dicho protocolo.

3.2.2. Canales de recepción de información en ThingSpeak

La base de recolección de información del presente proyecto es el servidor Thingspeak. Se ha creado tres canales denominados:

- **Consumo:** Que almacena el volumen consumido, el nivel del tanque utilizado y el identificador del dispensador.
- **Cambio Tanque:** Que almacena la información de la evolución del volumen disponible por tanque y el identificador del dispensador.
- **Posición del Dispensador:** Se ha creado este canal para identificar la posición geográfica de cada dispensador, con la información de latitud, longitud y elevación, sin embargo, no se ha utilizado aún.



Figura 53: Canales generados en Thingspeak
FUENTE: ELABORACIÓN PROPIA

Para acceder a estos canales, se utilizaron las siguientes URL's:

Escritura del canal:

```
GET https://api.thingspeak.com/update?api\_key=YRLUQXIE09T0W3TG&field1=0
```

Lectura del canal:

```
GET https://api.thingspeak.com/channels/724296/feeds.json?api_key=08LVWJ2KVXGNEW02&results=2
```

Como se puede observar, existen dos api_key (Llaves API), una que permite la escritura y otra que permite la lectura.

Esta información es enviada por la SIM900 para realizar la actualización de los datos de consumo por cada tanque y cada dispensador.

3.2.3. Comunicación con el servidor en la nube

Para realizar la comunicación entre el circuito de control y el servidor ThingSpeak, se ha utilizado una placa de comunicación denominada Sim900, cuyo control es vía protocolo RS232. Por este medio, se envían comandos AT a dicho circuito y de esa manera se remite la información al servidor IoT ThingSpeak.

La Sim900 funciona como un teléfono celular, ya que utiliza una SIM, mediante el cual, puede registrarse en la red de telefonía celular y así poder enviar la información.

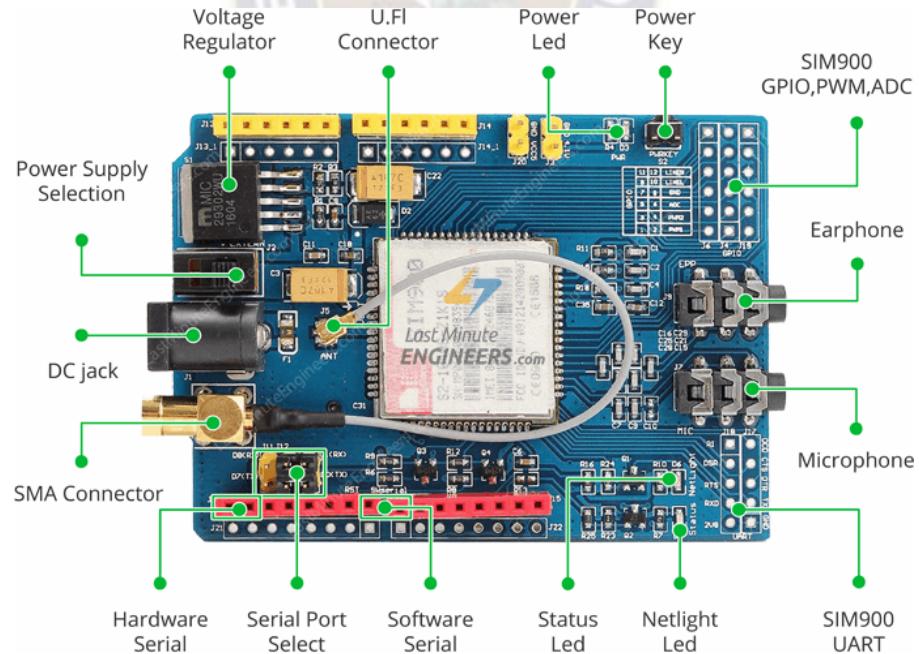


Figura 54: Descripción de la placa de comunicaciones SIM900.

Fuente: (<https://lastminuteengineers.com>, 2019)

Cuando el LED (Power Led) al lado del botón de encendido (Power Key) se encuentra encendido, la Sim900 se encuentra alimentada.

El LED de estado (Status Led) indica el estado de funcionamiento del circuito, si se encuentra encendido, el circuito se encuentra en modo activo.

El led de Red (Netlight Led) indica el estado del registro en la red telefónica:

- Estado 1: Si se encuentra apagado, la Sim900 no tiene comunicación alguna con la red.
- Estado 2: 64ms encendido y 800ms apagado: la Sim900 está funcionando, pero no se encuentra registrado en la red.
- Estado 3: 64ms encendido y 3s apagado: la Sim900 está funcionando, y se encuentra plenamente registrado en la red, podemos llamar, mensajear o conectar a internet.
- Estado 4: 64ms encendido y 300ms apagado: La conexión de datos GPRS que solicitó está activa.

3.2.3.1. Registro en la red

Para iniciar el registrado en la red telefónica, se debe presionar el botón de encendido (Power Key) durante 3 segundos. El NetLight parpadeará intermitentemente (Estado 2) mientras busca la red para registrarse, y a continuación, parpadeará a un ritmo más lento que el anterior (estado 3).

Para realizar el encendido (o reinicio) por software, es decir, por un pulso que provenga del microcontrolador, se debe unir los dos pines del circuito, que se muestran en la siguiente figura:



Figura 55: Pines a unir para habilitar el encendido por software de la SIM900.

Fuente: (prometec.mx, 2019)

3.2.3.2. Conexión con el Circuito de control

Se utilizaron los pines del Software Serial (7 y 8) que posee la Sim900. Para ello previamente se debe unir mediante un jumper los pines de comunicación, descritos a continuación:

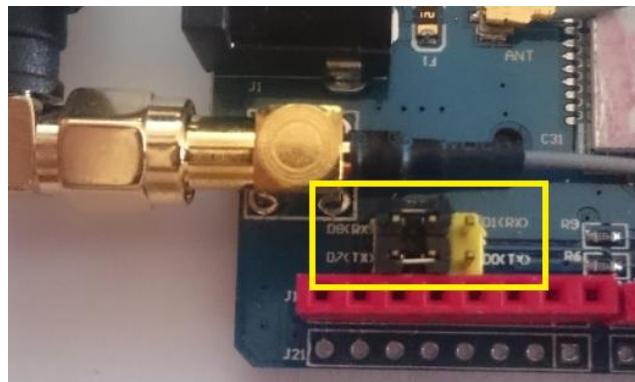


Figura 56: Pines a unir para habilitar el serial por software de la SIM900.

Fuente: (prometec.mx, 2019)

La velocidad de comunicación entre el circuito de control y la Sim900 es de 19200 baudios, debido a que se considera la velocidad óptima de funcionamiento, de acuerdo a la bibliografía consultada sobre el tema.

3.2.4. Transmisión de información al servidor en la nube

Para transmitir información vía serial, a la Sim900, son necesarios los comandos AT. A continuación, se describirán solo los comandos que fueron utilizados en la comunicación con el servidor en la nube y las respuestas afirmativas que son relevantes:

- *Comando AT*: Es el comando por esencia de prueba, debido a que se necesita iniciar la comunicación con este prefijo para casi todos los demás comandos. Enviándolo solo, recibe la respuesta de *OK* si existe comunicación con la Sim900.
- *Comando AT+CREG?*: Sirve para preguntar al circuito si se encuentra registrado en la red. La respuesta importante es *+CREG: 0,1* que nos indica que el circuito se encuentra registrado a la red que la SIM estaba configurada.
- *Comando AT+CGATT?*: Este comando verifica si el circuito se encuentra conectado a la red GPRS. Si lo está, responde un *+CGATT:1*, caso contrario, se debe enviar el

comando *AT+CGATT=1*, para establecer la conexión o *AT+CGATT=0* si se desea eliminar la misma.

- *Comando AT+CGDCONT*: Mediante este comando, definimos el contexto PDP, que significa activar una sesión de usuario con la red. Si la red acepta, la respuesta será *OK*.
- *Comando AT+CSTT?*: Sirve para verificar la información de conexión a la red de datos GPRS: el Nombre del Punto de Acceso (APN), usuario y contraseña. La respuesta a este comando tiene el formato: *+CSTT: "APN", "USER", "PWD"*.

En nuestro medio, las APN utilizadas por las operadoras VIVA NUEVATEL y TIGO son las siguientes:

VIVA NUEVATEL: Internet.nuevatel.com

TIGO: Internet.tigo.bo

Se hizo la prueba con los chips de ambas compañías, en ambos casos los resultados fueron similares, sin problemas para realizar la conexión.

Por lo que si el APN no se encuentra establecido, se envía: *AT+CSTT="internet.nuevatel.com","",""*. A lo que la Sim900 responderá simplemente *OK*.

- *Comando AT+CIICR*: Se utiliza para establecer la conexión inalámbrica con GPRS, cuando ésta no está establecida. Si se establece por primera vez, la respuesta será *OK*.
- *Comando AT+CIFSR*: Obtiene la dirección IP local, que la red le asigna. La respuesta de este comando es justamente, la IP asignada, por ejemplo: *10.211.204.224*
- *Comando AT+CIPSTART*: Pone en marcha la conexión TCP o UDP que se tenga definida. Si no existe una conexión definida, ésta puede ser definida de la siguiente manera: *AT+CIPSTART="modo (TCP o UDP)", "dirección IP", "puerto de conexión"*.

En nuestro caso, para conectar con el servidor ThingSpeak, se utilizó: *AT+CIPSTART="TCP", "api.thingspeak.com", "80"*.

Cuando la conexión es satisfactoria, la respuesta es: *CONNECT OK*. Si existe una conexión anterior, la respuesta es: *ALREADY CONNECT*.

- *Comando AT+CIPSEND*: Sirve para enviar datos a través de una conexión TCP o UDP ya establecida. En este comando se define el tamaño de la cadena de texto a enviar y a continuación, la cadena a enviar. En el caso de nuestra conexión a ThingSpeak, si queremos enviar que el dispensador N°1 ha dispensado 1.500ml y el contenido actual del tanque es de 15 litros, se envía:

AT+CIPSEND=92

>GET/update?api_key=ABCDEFGHIJKLMNP&field1=1&field2=1500&field3=15000 Connection: close.

La respuesta a esta cadena debe ser *SEND OK*.

De esta manera, el servidor ThingSpeak recibe la información del dispensador, la misma puede verse de la siguiente manera.



Figura 57: Graficas que genera ThingSpeak.

Fuente: (ThingSpeak, s.f.)

3.2.5. Cálculo del flujo de agua por segundo

De acuerdo a las hojas de datos de la bomba utilizada, el flujo de agua varía entre 1 a 3 litros por minuto si se le aplica de 6V a 12V. La fuente utilizada para las bombas, es de 12V nominal, y 10,4V real.

Se hicieron las pruebas para llenar un vaso de 100ml de agua, y en promedio, se obtuvo un tiempo de 2,5 segundos para tal objetivo.

Por lo que se puede aseverar que el flujo de agua es de 2.4 l/min.

3.2.6. Diagrama de Flujo de la Solución final

Debido a la longitud del diagrama de flujo de la solución final, esta se encuentra descrita en los anexos. Sin embargo, se realiza el detalle de las constantes y los casos de uso.

Se establecieron las siguientes constantes

Constante	Unidad de medida	Cantidad
Volumen de líquido por moneda	ml	200
Corte de moneda a introducir	Bs	1
Nivel de Alerta	Porcentaje	10
Nivel Critico	Porcentaje	5

TABLA 9: CONSTANTES DEFINIDAS POR EL CLIENTE.

FUENTE: ELABORACIÓN PROPIA

En base a las cuales, se describe el procedimiento de funcionamiento del Dispensador:

- Caso general

El caso común inicia ingresando 1Bs en el monedero, el cual, es detectado e internamente un contador se incrementa. El Dispensador muestra el mensaje de “200ml, ¿proceder?”, para el cual el usuario decide si ingresa una nueva moneda (en donde seguirá incrementando el

volumen mostrado en pantalla) o procede a recibir el líquido que ha comprado, presionando el botón de “proceder”.

- Descuento de líquido y cambio de tanque

Antes de iniciar con la dispensación del líquido, internamente, se descuenta el volumen despachado del registro que almacena la capacidad del tanque en uso. Una vez llegado al *nivel de alerta*, el dispensador automáticamente y de forma invisible al usuario, calcula la cantidad de líquido que puede dispensar desde el actual tanque en uso, y la cantidad a utilizar del siguiente tanque, si correspondiese un cambio en medio del despacho de líquido.

Si el siguiente tanque a utilizarse no se encuentra lleno, el dispensador no permite el ingreso de más monedas y dispensa únicamente la capacidad que le resta. Finalmente se indisponer para seguir suministrando el servicio.

- Interrupción del líquido en el proceso de dispensado

El usuario tiene 3 segundos para acomodar la botella en el pico del dispensador. Pero si por algún motivo no alcanza a colocarlo, o decide moverlo antes de que el líquido haya sido totalmente dispensado, el sistema interrumpe el llenado y guarda la cantidad de líquido que no ha dispensado, para posteriormente despachar en cuanto el usuario coloque nuevamente la botella bajo el pico del dispensador.

- Envío de información al servidor

En todos los casos en los que el dispensador haya terminado de dar el servicio, remite al servidor el reporte del despacho del servicio

Cada dispensador tiene una placa de comunicación GSM, denominada SIM900. Esta se encarga de enviar la información anterior. Se analizó la posibilidad de utilizar un módulo Wifi como el esp8266, pero se vio por conveniente la SIM900 debido a su independencia, ya que no es necesario tener una red WiFi para que el dispensador envíe la información al servidor, sino un chip de alguna empresa telefónica.

- Control del servicio

La ventaja de tener un dispensador con una placa SIM900 incorporada, es la facilidad de comunicarse con el mismo. Desde el Tablero de control, es posible apagar y reiniciar el dispensador, si se encontrase alguna falla en el mismo, vía SMS.

El código fuente de todo el prototipo se encuentra detallado en los anexos.

3.2.7. Desarrollo del Tablero de Control

El tablero de control ha sido desarrollado sobre PHP y MySQL. El mismo cuenta con un módulo de acceso y gestión de usuarios, donde solo los Clientes y el desarrollador pueden ingresar.

Una vez ingresado, el *Dashboard* muestra las estadísticas del uso de los dispensadores, donde existe la posibilidad de analizar cada dispensador por separado o finalmente todos en conjunto, por un lapso de tiempo o un histórico.

Asimismo, se ha añadido una funcionalidad donde se puede realizar la descarga de un reporte en Excel del periodo elegido.

A la vez, el tablero muestra, mediante barras coloreadas, la capacidad de los tanques de cada dispensador, para que se pueda decidir el momento de hacer el cambio de tanque.

3.2.8. Acceso vía dispositivo Móvil

El desarrollo de las interfaces del *Dashboard*, fueron desarrolladas con el framework Bootstrap, lo que le permite ser visualizado tanto en un ordenador de escritorio, portátil y por un dispositivo móvil.

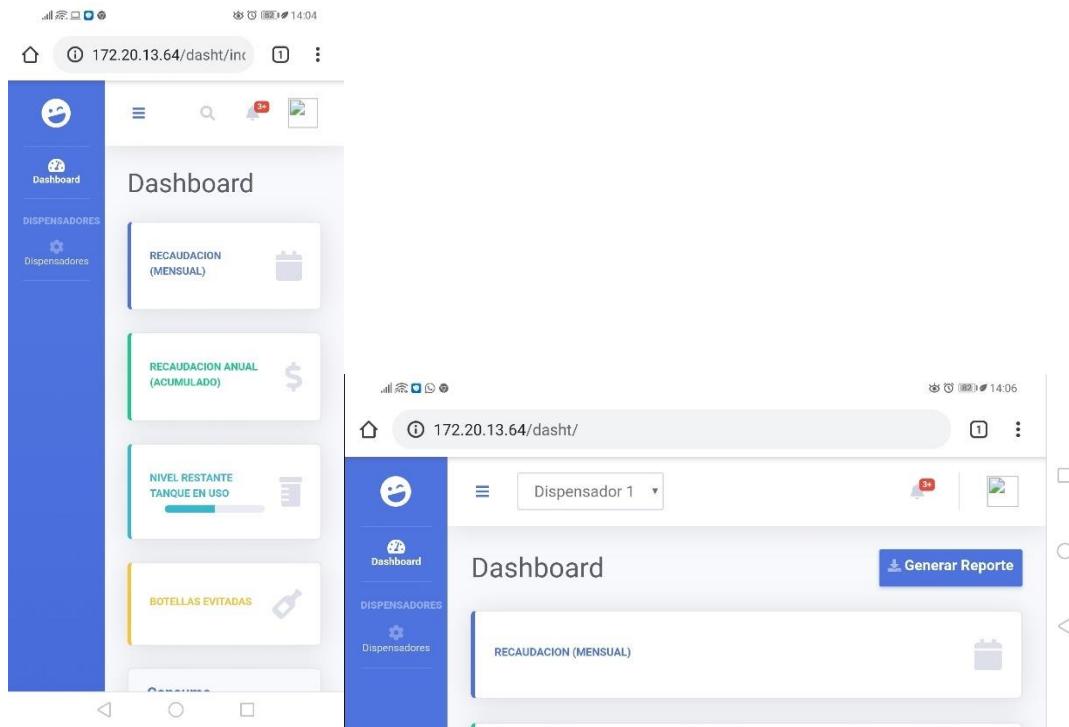


Figura 58: Vista de las interfaces en versión móvil
FUENTE: ELABORACIÓN PROPIA

3.2.9. Indicadores técnicos importantes

A continuación, se muestran los indicadores que caracterizan al dispensador desarrollado, al sistema de información:

Indicador	Descripción	Unidad	Valor
Consumo eléctrico mínimo	Mientras el dispensador no despacha agua, en stand by	mA	10
Consumo eléctrico máximo	Mientras el dispensador se encuentra en funcionamiento	mA	350
Memoria utilizada	memoria utilizada para almacenar el código	Kbytes	21
Puertos utilizados	Puertos utilizados para realizar todas las funcionalidades del dispensador	Puertos	25
Información promedio enviada al servidor	Cantidad de bytes enviados al servidor por cada consumo	Bytes	100
Tiempo medio de envío de información	Tiempo que tarda la SIM900 en enviar la información al servidor	Segundos	5

Tiempo de actualización del sistema de información	Tiempo que tarda en actualizar la información del consumo	Segundos	60
Tiempo promedio de consumo		Minutos	34
Tiempo mínimo	el menor tiempo entre dos consumos	Minutos	14
Tiempo máximo	el mayor tiempo entre dos consumos	Minutos	53

TABLA 10: INDICADORES DEL SISTEMA COMPLETO DEL DISPENSADOR.

FUENTE: ELABORACIÓN PROPIA

3.3. IMPLEMENTACIÓN PRÁCTICA

La implementación práctica, es decir, las primeras pruebas de la red de dispensadores y el prototipo funcional, fueron en la Universidad San Francisco de Asís, por instrucción de los Clientes.

El dispensador tuvo bastante aceptación en el lugar, tanto así que, en menos de una semana de haber sido implementado, el primer grupo de tanques se terminó, y se procedió a realizar el primer llenado. Así se puede ver en el reporte en Excel del *Dashboard*.

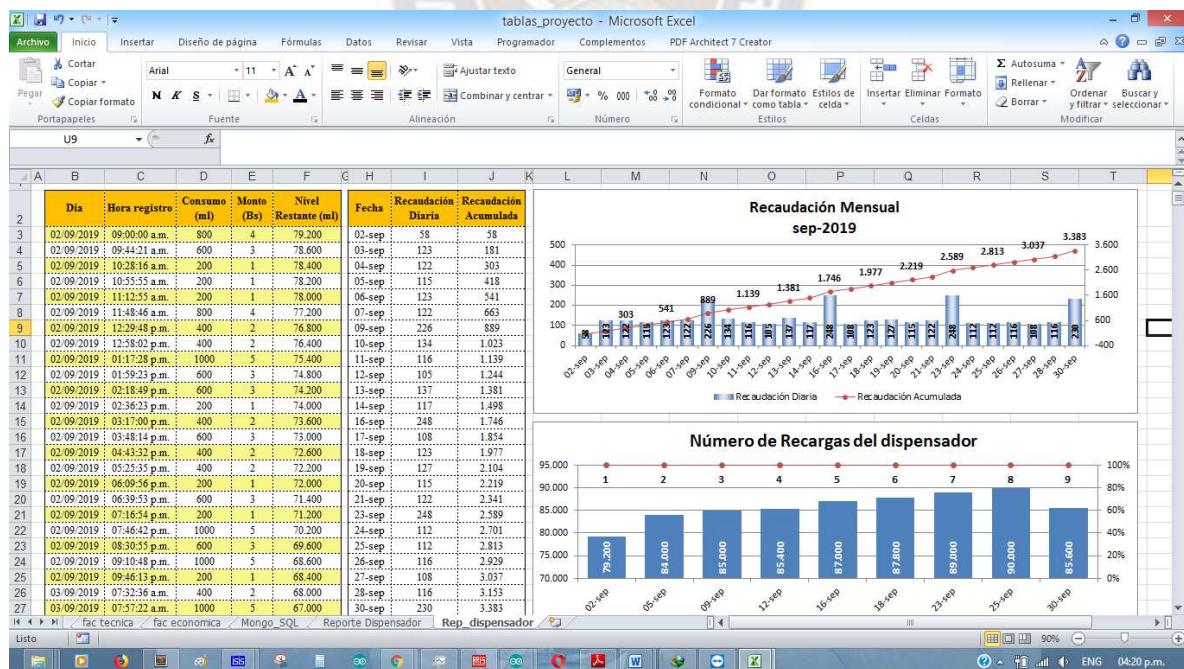


Figura 59: Reporte de consumo del Dispensador de prueba.

FUENTE: ELABORACIÓN PROPIA

3.4. PRUEBAS PREVIAS A LA IMPLEMENTACIÓN EN SERIE

Basados en el primer éxito, se ha iniciado con la fabricación de otros 4 dispensadores, los cuales, se ubicaron en lugares estratégicos, mismos que fueron definidos por el Cliente. La lista de lugares se detalla a continuación:

Id de Dispensador	Lugar
2	Colegio Montessori - Auquisamaña
3	Tienda Casa Ideas - 21 de Calacoto
4	Colegio La Salle - La Florida
5	Colegio Saint Andrews - La Florida

TABLA 11: LUGARES ESTRATÉGICOS DONDE SE INSTALARON LOS DISPENSADORES 2, 3, 4 Y 5.

FUENTE: ELABORACIÓN PROPIA

Estos otros 4 dispensadores han reportado diferentes resultados, todos con una aceptación similar al primer dispensador.



CAPÍTULO 4. ANALISIS DE FACTIBILIDAD

4.1.FACTIBILIDAD TECNICA

El detalle de equipamiento necesario para el proyecto se detalla a continuación:

Tipo de recurso	Nombre del recurso	Descripción	Cantidad
Recurso Humano	Desarrollador	de sistemas embebidos de sistemas de información	1
Hardware	Atmega328P	Sistema central del Dispensador	1
	Sensor ultrasónico	Para la detección de la presencia de la botella a la hora del llenado	1
	Pantalla LCD	Para mostrar los mensajes al Usuario	1
	Monedero electrónico	Para clasificar y detectar la moneda ingresada	1
	Dispositivo de comunicación GSM	para realizar la conexión con el servidor	1
	Driver para 4 motores	para controlar el líquido que sale de los tanques	1
	Botones de control	Para la interacción del usuario y el sistema	2
	Leds señalizadores	Para identificar el presionado de los botones	2
	Fuente de Poder	Para alimentar al circuito final	1
	Motores de agua	Para dispensar líquido	4
Software	Editor de Código compatible con el microcontrolador	Para desarrollar el sistema embebido	1
	Editor de desarrollo web	Para desarrollar el Panel de Control	1
	MySQL	Base de Datos	1
	PHP	Lenguaje de desarrollo del <i>Dashboard</i>	
	Apache	Servidor del <i>Dashboard</i>	
	Bootstrap	Framework de maquetación basado en CSS y Jquery	

TABLA 12: DETALLE DE RECURSOS TÉCNICOS PARA LA ELABORACIÓN DEL SISTEMA COMPLETO.

FUENTE: ELABORACIÓN PROPIA

Para evitar cualquier imprevisto por algún tipo de mala manipulación del Dispensador, se ha optado por elaborar un manual de usuario del uso del Dispensador, y otro detalle técnico para la administración del equipo, mismos que se encuentran en los anexos.

Además, para facilitar la obtención de cualquier componente que pueda ser dañado, se ha optado por trabajar con proveedores regionales, que garantizan la disponibilidad de cualquier repuesto para el Dispensador.

4.1.1 Redes IoT

Uno de los principales problemas con los que se ha encontrado en este proyecto, es la falta de una red específica para dispositivos IoT. Empresas como ZigBee, LoRa o Sigfox ofrecen redes propias que permiten una conexión independiente de Internet para transmitir sus datos en otros países, a diferencia de Bolivia que no posee una red propia para IoT.

No obstante, basándose en el principio del IoT (“enlazar a internet a cualquier dispositivo”), se ha optado por utilizar la red telefónica móvil, debido a la facilidad de enlace a internet mediante las tecnologías GSM o LTE.

4.2.FACTIBILIDAD ECONOMICA

Se muestran los gastos tanto en hardware, como en software para el sistema embebido y el servidor:

Componentes	Cantidad	Precio Unitario (Bs)	Precio Total (Bs)	Especificación técnica	Depreciación (años)	por año (Bs)	por mes (Bs)
Sistema de control completo	1	1200	1200	incluye los motores, tuberías, fuentes y transmisores GSM	10	120	10
Caja del Dispensador	1	1800	1800	incluye el armazón completo	10	180	15
Servidor en la nube (ThingSpeak)	1	0	0	-	-	-	-
Herramientas de desarrollo	5	0	0	-	-	-	-
Total Gastos Equipos			3000			300	25

TABLA 13: DETALLE DE INVERSIÓN EN RECURSOS TÉCNICOS PARA LA ELABORACIÓN DEL SISTEMA COMPLETO.

FUENTE: ELABORACIÓN PROPIA

Asimismo, se presenta el detalle de la inversión en recurso humano, para la elaboración del sistema embebido, el sistema de control y el armazón del dispensador.

Personal	Cantidad	Horas de Trabajo	Precio Unitario (Bs/hora)	Total Honorarios
Diseño del sistema embebido	1	800	22	17.600
Diseño del tablero de control	1	240	22	5.280
Diseño del armazón	1	40	18	720
Total Gastos personal				23.600

TABLA 14: DETALLE DE INVERSIÓN EN RECURSO HUMANO PARA LA ELABORACIÓN DEL SISTEMA COMPLETO.

FUENTE: ELABORACIÓN PROPIA

Finalmente existen gastos que se prevén sean de forma semanal, tal es el recambio de tanques de líquido en el dispensador.

Componentes recurrentes	Cantidad	Tamaño	Costo unitario (Bs)	Nro. de recambios previsto	Total (Bs)
Agua embotellada	4	20 litros	20	4	320

TABLA 15: DETALLE DE INVERSIÓN EN RECURSOS DE TIPO RECURRENTE PARA EL FUNCIONAMIENTO DEL DISPENSADOR.

FUENTE: ELABORACIÓN PROPIA

Cabe destacar que al hacer 4 recambios (se incluye el primer lote de botellones) se está dotando de 320 litros, de los cuales, el 95% se dispensa al usuario final:

ID	Ítem	Operación	Resultado
A	Nro. de Botellones por mes	-	16
B	Volumen por botellón (l)	-	20
C	Cantidad de líquido Total Mensual (l)	A * B	320
D	Cantidad de agua no dispensable por nivel alerta (l)	4 * B * 5%	4
E	Volumen efectivo de líquido a dispensar (l)	C - D	316
F	Volumen de líquido por moneda (l)	-	0,2
G	Corte de moneda a introducir (Bs)	-	1
	Total ingresos por dispensación de líquido (Bs)	E / F * G	1.580

TABLA 16: DETALLE DE RECAUDACIÓN DE UN DISPENSADOR.

FUENTE: ELABORACIÓN PROPIA

Finalmente se tiene la tabla del flujo de caja mensual y anual

Flujo de Caja mensual	Egreso	Ingreso
Recambio semanal de botellones de líquido	320	
Recaudación prevista por un dispensador		1.580
Total	320	1.580
Total Flujo Mensual		1.260
Total Flujo Anual		15.120

TABLA 17: DETALLE DE FLUJO DE CAJA ANUAL DE UN DISPENSADOR.

FUENTE: ELABORACIÓN PROPIA

La proyección del flujo anual hasta el 2030 es el siguiente:

	Año										
	0	1	2	3	4	5	6	7	8	9	10
Ingresos por ventas		15.120,0	15.120,0	15.120,0	15.120,0	15.120,0	15.120,0	15.120,0	15.120,0	15.120,0	15.120,0
Depreciación		(300,0)	(300,0)	(300,0)	(300,0)	(300,0)	(300,0)	(300,0)	(300,0)	(300,0)	(300,0)
UAII		14.820,0	14.820,0	14.820,0	14.820,0	14.820,0	14.820,0	14.820,0	14.820,0	14.820,0	14.820,0
impuestos (14%)		(2.074,8)	(2.074,8)	(2.074,8)	(2.074,8)	(2.074,8)	(2.074,8)	(2.074,8)	(2.074,8)	(2.074,8)	(2.074,8)
UDII		12.745,2	12.745,2	12.745,2	12.745,2	12.745,2	12.745,2	12.745,2	12.745,2	12.745,2	12.745,2
Depreciación		300,0	300,0	300,0	300,0	300,0	300,0	300,0	300,0	300,0	300,0
Flujo de Caja Libre	(26.600,0)	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2

TABLA 18: DETALLE DEL FLUJO DE CAJA PROYECTADO DE UN DISPENSADOR.

FUENTE: ELABORACIÓN PROPIA

Considerando la tasa de rendimiento conservadora de 12%, los indicadores VAN y TIR son:

Parámetro	Año										
	0	1	2	3	4	5	6	7	8	9	10
Flujo de Caja Libre	(26.600,0)	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2	13.045,2
Tasa de Rendimiento	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%
VAN		(14.952,5)	(4.552,9)	4.732,4	13.022,8	20.425,0	27.034,1	32.935,1	38.203,9	42.908,1	47.108,3
TIR		-50,96%	-1,28%	22,10%	33,69%	39,89%	43,40%	45,49%	46,76%	47,56%	48,07%

TABLA 19: VALORES DEL VAN Y TIR A DIFERENTES AÑOS.

FUENTE: ELABORACIÓN PROPIA

Por lo que el proyecto, con un solo dispensador, es rentable desde el 3er año. Cabe mencionar que, si tomamos en cuenta a los otros 4 dispensadores, tendríamos una rentabilidad aun mayor, ya que el monto inicial de mano de obra reduciría sustancialmente, debido a que se tienen todas las herramientas ya desarrolladas, y el tiempo de trabajo sería mucho menor.

	Inversión	Total
Elaboración de 5 dispensadores	15.000,0	38.600,0
Costo mano de obra	23.600,0	
Otros Datos		
Depreciación 5 dispensadores	(1.500,0)	(1.500,0)
Recaudación 5 dispensadores	75.600,0	75.600,0

TABLA 20: EVALUACIÓN DE COSTOS PARA 5 DISPENSADORES.

FUENTE: ELABORACIÓN PROPIA

Al fabricar 5 dispensadores, el costo de elaboración (al igual que la depreciación) se quintuplica, no obstante, el costo de mano de obra por el diseño, se mantiene constante. Con lo que nuestro flujo de caja hasta el 2030 sería el siguiente:

	Año										
	0	1	2	3	4	5	6	7	8	9	10
Ingresos por ventas		75.600,0	75.600,0	75.600,0	75.600,0	75.600,0	75.600,0	75.600,0	75.600,0	75.600,0	75.600,0
Depreciación		(1.500,0)	(1.500,0)	(1.500,0)	(1.500,0)	(1.500,0)	(1.500,0)	(1.500,0)	(1.500,0)	(1.500,0)	(1.500,0)
UAII		74.100,0	74.100,0	74.100,0	74.100,0	74.100,0	74.100,0	74.100,0	74.100,0	74.100,0	74.100,0
impuestos (14%)		(10.374,0)	(10.374,0)	(10.374,0)	(10.374,0)	(10.374,0)	(10.374,0)	(10.374,0)	(10.374,0)	(10.374,0)	(10.374,0)
UDII		63.726,0	63.726,0	63.726,0	63.726,0	63.726,0	63.726,0	63.726,0	63.726,0	63.726,0	63.726,0
Depreciación		1.500,0	1.500,0	1.500,0	1.500,0	1.500,0	1.500,0	1.500,0	1.500,0	1.500,0	1.500,0
Flujo de Caja Libre	(38.600,0)	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0

TABLA 21: DETALLE DEL FLUJO DE CAJA PROYECTADO DE 5 DISPENSADORES.

FUENTE: ELABORACIÓN PROPIA

Finalmente, los indicadores VAN y TIR indican que el rendimiento es óptimo desde el primer año:

Parámetro	Año										
	0	1	2	3	4	5	6	7	8	9	10
Flujo de Caja Libre	(38.600,0)	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0	65.226,0
Tasa de Rendimiento	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%	12%
VAN		19.637,5	71.635,3	118.061,8	159.514,1	196.525,1	229.570,7	259.075,6	285.419,3	308.940,4	56.682,7
TIR		68,98%	139,53%	159,29%	165,58%	167,75%	168,53%	168,81%	168,92%	168,96%	48,08%

TABLA 22: VALORES DEL VAN Y TIR A DIFERENTES AÑOS.

FUENTE: ELABORACIÓN PROPIA

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Se ha logrado diseñar el sistema de control del dispensador de agua, bajo los requerimientos del cliente, y posteriormente la red de dispensadores sobre el concepto del Internet de las Cosas (IoT).

Dicho sistema de control ha sido en un principio diseñado sobre Arduino, para posteriormente tener un circuito propio y personalizado, con componentes obtenidos de importadores locales, para evitar la escasez de repuestos, si correspondiese realizar reposiciones.

- Se ha logrado diseñar un tablero de control, para el análisis y toma de decisiones por parte de la división de purificación de agua del grupo Gundlach bajo el Stack WAMP, es decir, con el lenguaje PHP, base de datos MySQL (para la gestión de usuarios) y Apache como servidor web.

El mismo trabaja en conjunción con un servidor en la nube (ThingSpeak) que tiene una respuesta mucho más rápida que un servidor web normal, el cual se encarga de gestionar la información que los dispensadores le envían, debido a que fue justamente diseñada para administrar grandes cantidades de información de una forma eficiente y veloz dentro del ámbito del IoT.

- Se ha logrado diseñar dicho tablero de control, con compatibilidad de visualización para dispositivos móviles, gracias al framework Bootstrap, que de forma similar a la versión de escritorio, permite recibir alertas sobre el consumo de los botellones, los reportes de consumo, y la recaudación.
- Se ha logrado incentivar al reciclaje de botellas plásticas, mediante la dosificación de agua de mesa de forma automática y a precios competitivos, comparados con una botella de agua de 500ml.

Se ha visto que los estudiantes del lugar de prueba, utilizaban sus botellas de agua de 1 litro, para recibir el líquido que el dispensador dosifica, en vez de comprar botellas de agua de un solo uso.

Esto sin lugar a dudas es un aporte a la conservación de nuestro planeta, reduciendo la contaminación por botellas plásticas de un solo uso.

5.2. Recomendaciones

Si bien se ha tomado el concepto de IoT para este proyecto, no se ha podido encontrar redes destinadas específicamente al desarrollo sobre IoT, como en otros países (España por ejemplo), donde el IoT se ha convertido en una parte importante de la sociedad.

Por lo que la recomendación está dirigida a la creación de redes propias para IoT por las empresas telefónicas o emprendedores, así como el uso de dispositivos que sean controlados sobre esta red y de esta forma, explorar y explotar el sinnúmero de aplicaciones que permite la utilización de IoT.

Durante el desarrollo del presente proyecto, no se ha encontrado normativa alguna que regule la infraestructura, seguridad, tecnologías o servicios que se pueden ofrecer sobre IoT, por lo que, para estandarizar el uso, se debería elaborar la normativa correspondiente.

Otro punto importante, en este sistema de control, no se ha tomado en cuenta la independencia energética, por lo que, siguiendo la filosofía verde del presente proyecto, la utilización de tecnologías energéticas amigables con el medio ambiente pueden ser un objetivo a analizarse para futuros trabajos.

La elección de los lugares de expendio del dispensador desarrollado, fueron instruidos por el cliente. Por lo que para futuros trabajos, puede realizarse un estudio de mercado previo, mediante el cual se identifique los lugares con mayor venta del producto, y alcanzar resultados mayores al actual.

La realización del PCB ha sido otra dificultad con la que se ha cruzado este proyecto. Para tener una placa final, con la calidad de las que se encuentran en otros dispositivos electrónicos, se debe enviar el diseño a algún fabricante en china, lo que supone un retardo de casi 3 meses en solo obtener una placa PCB de alta calidad.

En relación al desarrollo del sistema embebido del microcontrolador: Durante la investigación sobre la metodología de desarrollo del sistema embebido para el microcontrolador, no se ha encontrado una que sea puramente para este fin. Por lo que se ha adaptado una metodología ya conocida. Para este trabajo, se ha considerado SCRUM como la más sugerente, debido a la familiaridad con el projectista. Sin embargo, es aconsejable establecer una metodología de desarrollo para sistemas embebidos como tal.



Bibliografía

- 330ohms.com. (15 de Septiembre de 2019). *330ohms.com*. Obtenido de <https://www.330ohms.com/products/bomba-de-agua-12v-rs-385>
- Alan Grier, D. (2018). *IEEE Computer Society, Invisible Internet Of Things*. Recuperado el 09 de Junio de 2019, de <https://www.computer.org/publications/tech-news/closer-than-you-might-think/invisible-internet-of-things>
- Alarcón, J. M. (7 de Septiembre de 2019). *www.campusmvp.es*. Obtenido de <https://www.campusmvp.es/recursos/post/Que-es-el-stack-MEAN-y-como-escoger-el-mejor-para-ti.aspx>
- Andalucia, J. d. (2009). *Curso de Administrador de Servidores Internet*. Desconocido.
- Banggood. (s.f.). Recuperado el 23 de junio de 2019, de https://es.banggood.com/SIM900-Quad-band-GSM-GPRS-Shield-Development-Board-For-Arduino-p-964229.html?cur_warehouse=CN
- Cahuaya, W. (10 de febrero de 2019). *Plástico inunda las calles y EMSA recoge 35 T de botellas cada día*. (Los Tiempos) Recuperado el 19 de marzo de 2019, de <https://www.lostiempos.com/actualidad/cochabamba/20190210/plastico-inunda-calles-emsa-recoge-35-t-botellas-cada-dia>
- Campoverde, A., Hernandez, D., & Mazón, B. (2015). Cloud computing con herramientas open-source para Internet de las cosas. *MASKANA, VI*(Especial), 173-182.
- Cangás Cangás, F. X. (2015). *sistema web transur con node.js para la gestión de transporte de la cooperativa de transporte de pasajeros inter cantonal urcuquí*. Ibarra.
- carlosazaustre.es. (03 de Septiembre de 2019). *carlosazaustre.es*. Obtenido de <https://carlosazaustre.es/empezando-con-angular-js/>
- Carrod.mx. (22 de 08 de 2019). *Módulo GPRS/GSM Shield con Antena Integrada SIM900*. Obtenido de <https://www.carrod.mx/products/modulo-gsm-gprs-sim900>
- Castillo Izquierdo, J. (2017). *Diseño e implementación de un dispositivo IoT de bajo coste para entornos agrícolas*. Granada: Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación.
- Dave, E. (Abril de 2011). *Internet of Things, La próxima evolución de Internet lo está cambiando todo*. Recuperado el 10 de 06 de 2019, de https://www.cisco.com/c/dam/global/es_es/assets/executives/pdf/Internet_of_Things_IoT_IBSG_0411FINAL.pdf
- del Valle Hernández, L. (s.f.). *Programar Facil: Proyectos IoT con Arduino, las plataformas más importantes*. Recuperado el 11 de Junio de 2019, de <https://programarfácil.com/podcast/proyectos-iot-con-arduino/>
- electrohobby.org. (18 de septiembre de 2019). *electrohobby.org*. Obtenido de <http://www.electrohobby.org/conversor-lcd-i2c-arduino/>

- Fernandez Cejas, M. (15 de noviembre de 2017). *Tecnología y Negocio, IoT: ¿Cuáles son sus componentes principales?* Recuperado el 10 de junio de 2019, de <https://www.itop.es/blog/item/iot-cuales-son-sus-componentes-principales-y-aplicaciones.html>
- Hernández Rojas, D., Mazon-Olivo, B., & Escudero, C. (2018). Internet de las cosas (IoT). En *Ánalisis de Datos Agropecuarios* (págs. 73-97). Machala, Ecuador: UTMACH.
- <https://lastminuteengineers.com>. (22 de Julio de 2019). Obtenido de <https://lastminuteengineers.com/sim900-gsm-shield-arduino-tutorial/>
- Lara, W. (s.f.). *Platzi: ¿Cómo funciona la metodología Scrum?* Recuperado el 15 de Junio de 2019, de <https://platzi.com/blog/metodologia-scrum-fases/>
- Mandado Pérez, E., & Mandado Rodríguez, Y. (2008). Sistemas Electrónicos Digitales 9na Edición. Barcelona: MARCOMBO EDICIONES TECNICAS.
- Mendes Calo, K., Estevez, C., & Fillottrani, P. (2010). Evaluación de Metodologías Ágiles para Desarrollo de Software. *XII Workshop de Investigadores en Ciencias de la Computación*, 455-459.
- [mercadolibre.com](https://mercadolibre.com.mx). (14 de septiembre de 2019). mercadolibre.com. Obtenido de https://articulo.mercadolibre.com.mx/MLM-699029227-monedero-electronico-multimonedas-_JM
- miarcade.com. (11 de Septiembre de 2019). miarcade.com. Obtenido de <https://miarcade.com/el-microswitch-arcade-funcionamiento/>
- Microcontroladores y sus aplicaciones.* (s.f.). Recuperado el 2 de Junio de 2019, de <https://microcontroladoressv.wordpress.com/empresas-fabricantes-de-microcontroladores/>
- mindinventory.com. (04 de septiembre de 2019). mindinventory.com. Obtenido de <https://www.mindinventory.com/blog/angularjs-development-skills-benefits-for-developers/>
- Moreno Arboleda, F. J., Quintero Rendón, J. E., & Rueda Vásquez, R. (2016). Una comparación de rendimiento entre Oracle y MongoDB. *Ciencia e Ingeniería Neogranadina*, 109-129.
- [mozilla](https://developer.mozilla.org). (03 de septiembre de 2019). developer.mozilla.org. Obtenido de https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction
- Muñoz, G. (25 de febrero de 2014). *¿Dónde está Avinash cuando se le necesita?* Recuperado el 21 de junio de 2019, de <http://dondeestaavinashcuandosele necesita.com/2014/02/the-greatest-value-of-picture-is-when.html>
- naylampmechatronics.com. (15 de Septiembre de 2019). naylampmechatronics.com. Obtenido de <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- [peerbits](https://www.peerbits.com). (2 de septiembre de 2019). www.peerbits.com. Obtenido de <https://www.peerbits.com/blog/choose-nodejs-for-web-application-development-project.html>
- Pérez, W. (13 de marzo de 2019). *Cada boliviano utiliza al año 372 bolsas de plástico.* (La Razón, Sección Sociedad) Recuperado el 19 de marzo de 2019, de http://www.la-razon.com/sociedad/bolsas-plastico-uso-bolivia-por-ano_0_3110088990.html

- Piuri, V., & Minerva, R. (Julio de 2015). *Computing Now, Building the Internet of Things*. Recuperado el 09 de Mayo de 2019, de <http://www.computer.org/publications/tech-news/computing-now/building-the-internet-of-things>
- prometec.mx. (23 de 07 de 2019). Obtenido de <https://www.prometec.net/gprs-llamar-enviar-sms/>
- RS Online: *Internet de las Cosas*. (s.f.). Recuperado el 11 de Junio de 2019, de <https://es.rs-online.com/web/generalDisplay.html?id=i/iot-internet-of-things>
- Segura, D., Noguez, R., & Espín, G. (s.f.). Contaminación ambiental y bacterias productoras de plásticos biodegradables. *Biotecnología V14 CS3. indd*, 361-372.
- Soria, C. (18 de septiembre de 2019). *Alternativas a la botella de plástico que también puedes llevar a todas partes*. (Hola.com) Recuperado el 1 de marzo de 2019, de <https://www.hola.com/estar-bien/20180918129700/alternativas-a-la-botella-de-plastico-que-tambien-puedes-llevar-a-todas-partes-cs/>
- teslabem.com. (18 de septiembre de 2019). *teslabem.com*. Obtenido de <https://teslabem.com/nivel-intermedio/fundamentos-del-protocolo-i2c-aprende/>
- ThingSpeak. (s.f.). Recuperado el 23 de junio de 2019, de https://thingspeak.com/channels/694263/private_show
- todoelectrodo.blogspot.com. (18 de septiembre de 2019). *todoelectrodo.blogspot.com*. Obtenido de <http://todoelectrodo.blogspot.com/2013/02/lcd-16x2.html?m=1>
- Valdez Pérez, F., & Pallás Areny, R. (2007). Microcontroladores: Fundamentos y aplicaciones con PIC. Barcelona: MARCOMBO, Ediciones Técnicas.
- videorockola.com. (15 de septiembre de 2019). [www.videorockola.com](http://www.videorockola.com/tutoriales/conexion-de-un-monedero-electronico/). Obtenido de <http://www.videorockola.com/tutoriales/conexion-de-un-monedero-electronico/>
- Viloria Núñez, C., Cardona Peña, J., & Lozano Garzón, C. (2009). Análisis comparativo de tecnologías inalámbricas para una solución de servicios de telemedicina. *Ingeniería y Desarrollo*, 200-217.
- wikitechy.com. (03 de septiembre de 2019). www.wikitechy.com. Obtenido de <https://www.wikitechy.com/tutorials/node-js/node-js-express-framework-material>
- www.diarioelectronicohoy.com. (27 de Agosto de 2019). Obtenido de <https://www.diarioelectronicohoy.com/blog/pulsadores-sin-rebotes>
- www.luisllamas.es. (28 de agosto de 2019). www.luisllamas.es. Obtenido de <https://www.luisllamas.es/debounce-interrupciones-arduino/>
- www.retrowiki.es. (27 de agosto de 2019). Obtenido de <http://www.retrowiki.es/viewtopic.php?t=200030598>

Material Anexo

Anexo a. código fuente a nivel de hardware

```

#include <SoftwareSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
LiquidCrystal_I2C lcd(0x3F,2,1,0,4,5,6,7); // 0x27 is
the I2C bus address for an unmodified backpack
//config(lcd_Adr, EN, RW, RS, D4, D5, D6, D7);
//LiquidCrystal_I2C lcd(0x27,16,2);
//LiquidCrystal_I2C lcd(0x38,16,2); //para el proteus

#define id_disp 1
//const int id_disp = 1;
byte mostMensaje =1;
byte mostSerial =1;
int capacidad = 0;
volatile byte EstBtnMoneda = 0; // current state of
the button
byte UltEstBtnMoneda = 0;
byte EstBtnProceder = 0; // current state of the
button
byte UltEstBtnProceder = 0;
//const int pinMoneda = 2;
#define pinMoneda 2
#define pinProceder 7
//const int pinProceder = 7;

#define pinIndicador 9
//const int pinIndicador = 9;
//tiempo de llenado de la botella
unsigned long tiempoAnterior = 0;
unsigned long tiempoActual = 0;
unsigned long tiempoOtroTanque=0;
byte proceder = 0;
int tiempoLlenado = 0;
int actMedida=0;
//para el sensor de posicion del vaso
float tiempo;
#define pinecho 6
#define pintrigger 5
//const int pinecho = 6;
//const int pintrigger = 5;
int tamMinVaso= 15; //cambiar para el real a maximo
30 cm
byte capturarPos;
unsigned long tiempoInterrupLlenado;

byte capTanque = 20; /*100 valor en ml
byte actCapTanque = capTanque; /*100 valor normal
200
byte futCapTanque;
byte actTanqueUso = 1;
byte sigTanque = 1;
byte cambioTanque = 0;
byte capSigTanque = capTanque; //valor normal 200
//int profTanque = 50; //cambiar para el real a maximo
50 cm
float tiempoCapLlenado= 3; //tiempo en que se llena el
capLlenado.
int capLlenado = 3; /*100cantidad a mostrar por cada
moneda

//int pinReinicia = 3;
#define pinReinicia 3
byte EstBtnReinicia = 0; // current state of the
button
byte UltEstBtnReinicia = 0;
byte reiniciaTanque = 0;
byte opReini=1;
byte EstBtnEligeTanque=0;
byte UltEstBtnEligeTanque=0;
//int pinEligeTanque = 4;
#define pinEligeTanque 4
int porcAlerta = 10; //porcentaje de capacidad a la cual
se debe alertar.
int porcCritico = 5; //porcentaje que inhabilita el
funcionamiento
int nivelCritico=
float(capTanque)*(float(porcCritico)/100.0); //si el
tanque es de 20000ml, el nivel critico es de 1000ml
//float nivelAlerta = float(profTanque)*(1-
float(porcAlerta)/100.0);
int nivelAlerta =
float(capTanque)*(float(porcAlerta)/100.0); //si el
tanque es de 20000ml, el nivel critico es de 2000ml
byte alertarNoDisp =0;
byte alertarNivel =0;
//const int pinAlerta=10;
#define pinAlerta 10

//Comunicacion con el SIM900
#define CONFIG_GSM_RXPIN 11
#define CONFIG_GSM_TXPIN 12
#define CONFIG_GSM_BAUDRATE 19200

SoftwareSerial SIM900 =
SoftwareSerial(CONFIG_GSM_RXPIN,
CONFIG_GSM_TXPIN); //pines RX 11 y TX 12

//const char
conexion[]="AT+CSTT=\"internet.tigo.bo\",\\\"\\\"";
const char
conexion[]="AT+CSTT=\"internet.nuevotel.bo\",\\\"\\\"";
const String destino="AT+CMGS=\"+59179512858\"";
//String destino =
"AT+CMGS=\"+591\"+phone_user+\"";
#define pinModem 13

const char instr0[] PROGMEM = "AT"; //respuesta0
const char instr1[] PROGMEM = "AT+CREG?";
const char instr2[] PROGMEM = "AT+CGATT=1\r";
const char instr3[] PROGMEM = "AT+CIICR";
const char instr4[] PROGMEM = "AT+CIFSR";
const char resp0[] PROGMEM = "OK";
const char resp1[] PROGMEM = "+CREG: 0,1";

const char * const tablaInst[] PROGMEM =
{ instr0, instr1, instr2, instr3, instr4, //conexion,
};

const char * const tablaResp[] PROGMEM =

```

```

{ resp0, resp1
};

char bufferInst[15]; // make sure this is large enough
for the largest string it must hold
char bufferResp[15]; // make sure this is large enough
for the largest string it must hold

//interrupcion del monedero
#define NOT_AN_INTERRUPT -1
long timeCounter = 0;

//para thingspeak
const char wrKey[]="";
char aux_str[100];

//para info
String info="";

void setup() {
  //start serial connection
  //Serial.begin(9600);
  lcd.begin(16,2);
  lcd.setBacklightPin(3,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("iniciando... ");
  //configure pin 2 as an input and enable the internal
  pull-up resistor
  pinMode(pinMoneda, INPUT_PULLUP); //2
  pinMode(pinReinicia, INPUT_PULLUP); //3
  pinMode(pinEligeTanque, INPUT_PULLUP); //4
  pinMode(pintrigger, OUTPUT); //5
  pinMode(pinecho, INPUT); //6
  pinMode(pinProceder, INPUT_PULLUP); //7
  pinMode(pinIndicador, OUTPUT); //9
  pinMode(pinAlerta, OUTPUT); //10
  pinMode(pinModem, OUTPUT); //13
  pinMode(actTanqueUso+13, OUTPUT); //14
  pinMode(actTanqueUso+14, OUTPUT); //15
  pinMode(actTanqueUso+15, OUTPUT); //16
  pinMode(actTanqueUso+16, OUTPUT); //17
  //pinMode(pinecho2, INPUT);
  //pinMode(pintrigger2, OUTPUT);
  actTanqueUso = EEPROM.read(0);
  delay(50); //el nro de tanque que se encuentra en uso en
  el momento de iniciar el programa
  if(actTanqueUso < 1 || actTanqueUso > 4)
  actTanqueUso = 1; //si el nro guardado no corresponde
  a uno entre 1 y 4, se asigna 1 automaticamente

  actCapTanque = EEPROM.read(actTanqueUso);
  delay(50); //capacidad deltanque en uso actual
  if(actCapTanque < 0 || actCapTanque > capTanque)
  actCapTanque = capTanque; //si no se tiene un nro
  razonable entre 0 y 200, se anota 200 como consumo
  basico del tanque actual,
  sigTanque= actTanqueUso + 1; //utilizamos el
  siguiente tanque
  if(sigTanque == 5) sigTanque=1;
  capSigTanque = EEPROM.read(sigTanque);

  delay(50); //capacidad del tanque siguiente a usarse
  if(capSigTanque < 0 || capSigTanque > capTanque)
  capSigTanque = capTanque; //si no se tiene un nro
  razonable entre 0 y 200, se anota 200 como consumo
  basico del tanque actual,
  lcd.setCursor(0,0);
  lcd.print(F("t. act: "));
  lcd.print(actTanqueUso);
  lcd.print(actCapTanque*100);
  lcd.setCursor(0,1);
  lcd.print(F("s. tan: "));
  lcd.print(sigTanque);
  lcd.print(capSigTanque*100);
  delay(500);
  futCapTanque = actCapTanque;
  attachInterrupt(digitalPinToInterrupt(pinMoneda),
  CambioEstadoMoneda, RISING);
  //attachInterrupt(digitalPinToInterrupt(pinMoneda),
  CambioEstadoMoneda, CHANGE);
  //attachInterrupt(digitalPinToInterrupt(pinMoneda),
  CambioEstadoMoneda, FALLING);
  //sim900
  SIM900.begin(CONFIG_GSM_BAUDRATE);
  Serial.begin(CONFIG_GSM_BAUDRATE);
  delay(500);
  //iniciar();
  info="T1:"+String(EEPROM.read(1))+";
  T2:"+EEPROM.read(2)+"; T3:"+EEPROM.read(3)+";
  T4:"+EEPROM.read(4);
  //delay(50);
  Serial.println(F("iniciando: \n niv de tanques:"));
  Serial.println(info);
  Serial.print(F("T en uso:"));
  Serial.println(actTanqueUso);
  Serial.print(F("niv:"));
  Serial.println(actCapTanque);
  Serial.print(F("nivel alerta: "));
  Serial.println(nivelAlerta);
  Serial.print(F("nivel critico:"));
  Serial.println(nivelCritico);
  reiniciar();
  iniciar();
  Serial.println(F("fin inicio"));
  EstBtnMoneda=0;
}

void loop() {
  //read the pushbutton value into a variable
  //int EstBtnMoneda = digitalRead(pinMoneda); //pin 6
  como boton simula la moneda
  int EstBtnProceder = digitalRead(pinProceder); //pin 7
  boton proceder
  //actProfTanque = medida(pinecho2, pintrigger2);
  //mide la profundidad del tanque
  int EstBtnReinicia = digitalRead(pinReinicia); //pin 3
  boton reiniciar el contador de liquido
  int EstBtnEligeTanque =
  digitalRead(pinEligeTanque); //pin 4 boton elige tanque
  a reiniciar
  //delay(1000);
  if(capSigTanque<=nivelCritico &&
  actCapTanque<(nivelCritico + capLlenado))
}

```

```

    {
        lcd.setCursor(0,1);
        lcd.print(String(actcapTanque));
        //actProfTanque = medida(pinecho2, pintrigger2);
        //mide la profundidad del tanque
        if(alertarNoDisp==0)
        {
            lcd.setCursor(0,0);
            lcd.print("no disponible ");
            lcd.setCursor(0,1);
            lcd.print("disculpe... ");
            Serial.println(F("nivel de agua critico, no
disponible"));
            Serial.print(F("cap tanque actual: "));
            Serial.println(actCapTanque*100);
            Serial.print(F("cap tanque sig. "));
            Serial.println(capSigTanque*100);
            digitalWrite(pinAlerta, HIGH);
            digitalWrite(14, LOW);
            digitalWrite(15, LOW);
            digitalWrite(16, LOW);
            digitalWrite(17, LOW);
            //alertar la no disponibilidad
            //mensajeAlerta(String(porcCritico));//envia el SMS
de alerta
            //digitalWrite(pinAlerta, LOW);
            alertarNoDisp=1;
        }
        else
        {
            if(actCapTanque<=nivelAlerta &&
capSigTanque<=nivelAlerta && alertarNivel==0)
            {
                //enviar alerta del nivel
                //mensajeAlerta(String(porcAlerta));
                Serial.println(F("nivel de agua 20%, solicitar
llenado"));
                alertarNivel=1;
            }
            else
            {
                //if(capTanque-actCapTanque>nivelAlerta &&
capTanque-capSigTanque>nivelAlerta )
                if(capSigTanque>=capTanque-nivelAlerta)
                {
                    alertarNivel=0;
                    alertarNoDisp=0;
                } //se levanta el flag para enviar las alertas de bajo
nivel y ND
            }
        }
        if(mostMensaje == 1)
        {
            //lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(F("preparado    "));
            lcd.setCursor(0,1);
            lcd.print(F("inserte moneda "));
            if(mostSerial==1)
            {
                Serial.println(F("preparado, inserte moneda"));
                mostSerial=0;
            }
        }
    }
}

if( (EstBtnMoneda != UltEstBtnMoneda) &&
( (EstBtnMoneda == 1) )
{
    digitalWrite(pinIndicador, HIGH);
    capacidad = capacidad + capLlenado;
    //Serial.println("capacidad: "+String(capacidad));
    futCapTanque = futCapTanque - capLlenado;
    if(capSigTanque<=nivelCritico &&
futCapTanque<(capLlenado+nivelCritico))
    {
        digitalWrite(pinAlerta, HIGH);
        lcd.setCursor(0,0);
        lcd.print(capacidad*100);
        lcd.print(F("ml MAXIMA"));
    }
    else
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(capacidad*100);
        lcd.print(F("ml"));
        digitalWrite(pinAlerta, LOW);
    }
    lcd.setCursor(0,1);
    lcd.print(F("proceder?"));
    Serial.print(F("moneda recibida: "));
    Serial.println(capacidad);

    //lcd.setCursor(0,1); lcd.print(medida(pinecho,
pintrigger));
    //Serial.print(F("posicion vaso: "));
    //Serial.println(medida(pinecho, pintrigger));
    mostMensaje = 0;
    EstBtnMoneda = 0;
}
else
{
    digitalWrite(pinIndicador, LOW);
}
UltEstBtnMoneda = EstBtnMoneda;

if (EstBtnProceder != UltEstBtnProceder) {
    if (EstBtnProceder == HIGH && proceder == 0)
    {
        digitalWrite(pinIndicador, HIGH);
        if(capacidad != 0)
        {
            lcd.clear();
            //mandar un mensaje que diga: "ponga el vaso
bajo la pila, el agua sale en 5, 4, 3, 2, 1."
            //SIM900.println("AT+CMGF=1"); //envio de
mensajes en modo texto
            Serial.println(F("btn proceder presionado"));
            lcd.setCursor(0,0);
            lcd.print(F("ponga un envase"));
            delay(1000);
        }
        //SIM900.println("AT+CMGS=\\"+59179512858\\\"");
        lcd.setCursor(0,0);
    }
}

```

```

lcd.print(F("el agua sale en"));
lcd.setCursor(0,1);
lcd.print(F("3 segundos    "));
delay(1000);
//SIM900.print("disp_1;"+String(capacidad)+"ml;
prof_tanq:"+String(actCapTanque));
lcd.setCursor(0,1);
lcd.print(F("2 segundos"));
delay(1000);
//SIM900.print(char(0x1A));
lcd.setCursor(0,1);
lcd.print(F("1 segundo"));
delay(1000);
lcd.clear();
//llenar vaso, encender el led del pin 13 por 5
segundos
proceder = 1; //flag de llenado del vaso
//lcd.clear(); lcd.setCursor(0,0);
lcd.print(proceder); delay(1000);
tiempoAnterior=millis(); //captura el tiempo
actual de inicio del llenado
//lcd.clear(); lcd.setCursor(0,0);
lcd.print(tiempoAnterior); delay(1000);
if(actCapTanque-capacidad<=nivelCritico)
{
    //capOtroTanque = nivelCritico-(actCapTanque-
capacidad); //capacidad que el nuevo tanque debe llenar
    tiempoOtroTanque = int((actCapTanque-
nivelCritico)* tiempoCapLlenado/capLlenado)*1000;
//tiempo en el cual debe cambiarse a otro tanque
    cambioTanque = 1;
}
tiempoLlenado = int(capacidad *
tiempoCapLlenado/capLlenado)*1000; //calcula 3
segundos para cada 300ml de liquido.
//lcd.clear(); lcd.setCursor(0,0);
lcd.print(tiempoLlenado);
Serial.print(F("llenando: "));
Serial.println(capacidad*100);
}
else
{
    digitalWrite(pinIndicador, LOW);
}
UltEstBtnProceder = EstBtnProceder;

if(proceder == 1) //debe llenar el vasito
{ //lcd.clear(); lcd.setCursor(0,0); lcd.print(millis());
actMedida= medida(pinecho, pintrigger);
if(actMedida<tamMinVaso)
{
    capturarPos=1;
    tiempoActual = millis();
    //if(millis()-tiempoAnterior<=tiempoLlenado)
    if(tiempoActual-tiempoAnterior<=tiempoLlenado)
    {
        if(tiempoActual-
tiempoAnterior>=tiempoOtroTanque &&
cambioTanque == 1)
{
            //modificar esta parte
            capacidad = nivelCritico-(actCapTanque-
capacidad);
            EEPROM.write(actTanqueUso, nivelCritico);
            digitalWrite(actTanqueUso+13,LOW);
            actTanqueUso = sigTanque; //utilizamos el
siguiente tanque
            EEPROM.write(0, actTanqueUso);
            delay(50); //grabar en la memoria el nuevo
tanque en uso
            //EEPROM.write(actTanqueUso, capTanque);
            delay(50); //grabar en la memoria la capacidad del
tanque recien usado
            //actCapTanque=capTanque;
            actCapTanque =
EEPROM.read(actTanqueUso);
            //actCapTanque = capTanque;
            sigTanque= actTanqueUso + 1; //utilizamos el
siguiente tanque
            if(sigTanque == 5) sigTanque=1;
            capSigTanque = EEPROM.read(sigTanque);
            delay(50); //capacidad del tanque siguiente a
usarse
            cambioTanque = 0;
            Serial.print(F("cambio de tanque: "));
            Serial.println(actTanqueUso);
            Serial.print(F("capacidad llenara sig Tanque: "));
            Serial.println(capacidad);
            //pinBomba=actTanqueUso+13;
        }
        digitalWrite(actTanqueUso+13,HIGH);
        //digitalWrite(pinBomba,HIGH);
        //lcd.clear(); lcd.setCursor(0,0);
        lcd.print(tiempoLlenado);
        lcd.setCursor(0,0);
        lcd.print(F("Llenando...    "));
        lcd.setCursor(0,1);
        lcd.print(F("    "));
        //lcd.setCursor(0,1); lcd.print(String(actMedida));
    }
    else
    {
        digitalWrite(actTanqueUso+13,LOW);
        //digitalWrite(pinBomba,LOW);
        proceder = 0;
        //lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(F("llenado listo! "));
        lcd.setCursor(0,1);
        lcd.print(F("gracias    "));
        delay(1000);
        mostMensaje = 1;
        mostSerial = 1;
        actCapTanque = actCapTanque-capacidad;
        futCapTanque = actCapTanque;
        EEPROM.write(actTanqueUso, actCapTanque);
        Serial.println(F("subiendo a internet"));
        //iniciar();
        cargaInternet(capacidad, actCapTanque);
        Serial.print(F("cap act tanque:"));
        Serial.println(actCapTanque);
    }
}
}

```

```

        capacidad =0;
    }
}
else
{
    if(capturarPos==1)
    {
        tiempoInterrupLlenado=millis();
        tiempoLlenado=tiempoLlenado -
        tiempoInterrupLlenado + tiempoAnterior;
        Serial.println("interrumpido:"+String(tiempoLlenado));
        //como el tiempo de llenado se ha interrumpido,
        el nuevo tiempo serÃ¡, la resta del tiempo de llenado,
        menos el tiempo en el que se ha interrumpido
        capturarPos=0;
    }
    tiempoAnterior =millis();
    //lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F("debe poner      "));
    lcd.setCursor(0,1);
    lcd.print(F("un recipiente   "));
    digitalWrite(actTanqueUso+13,LOW);
    //digitalWrite(pinBomba,LOW);
}
}

if( (EstBtnReinicia != UltEstBtnReinicia) {
    if (EstBtnReinicia == HIGH)
    {
        digitalWrite(pinIndicador, HIGH);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(F("reiniciar?"));
        //opReini = 1;
        lcd.setCursor(0,1);
        lcd.print(F("tanque:"));
        lcd.print(opReini); //mostrar opcion del tanque a
llenarse
        Serial.print(F("reiniciar? tanque:"));
        Serial.println(opReini);
        reiniciaTanque++;
        mostMensaje=0;
        alertarNoDisp =1;
        if(reiniciaTanque>1) //boton que definitivamente
reinicia
        {
            digitalWrite(pinIndicador, HIGH);
            Serial.println(F("reiniciando contador tanque"));
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(F("reiniciando   "));
            lcd.setCursor(0,1);
            lcd.print(F("cont. tanque "));
            lcd.print(opReini);
            EEPROM.write(opReini, capTanque);
            delay(1000);
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print(F("tanque "));
        }
        lcd.print(opReini);
        lcd.setCursor(0,1);
        lcd.print(F("tanque*100));
        lcd.print(F("ml "));
        actCapTanque=capTanque;
        Serial.print(F("tanque "));
        Serial.print(opReini);
        Serial.println(F(" lleno"));
        //Serial.println(capTanque);
        delay(1000);
        lcd.setCursor(0,0);
        lcd.print("tanque 1: ");
        lcd.print(EEPROM.read(1));
        lcd.setCursor(0,1);
        lcd.print("tanque 2: ");
        lcd.print(EEPROM.read(2));
        delay(1000);
        //lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(F("tanque 3: "));
        lcd.print(EEPROM.read(3));
        lcd.setCursor(0,1);
        lcd.print(F("tanque 4: "));
        lcd.print(EEPROM.read(4));
        delay(1000);
        reiniciaTanque=0;
        mostMensaje=1;
        opReini=1;
        setup();
        //aqui se debe mandar al thing speak el cambio de
tanque
    }
}
else
{
    digitalWrite(pinIndicador, LOW);
}
}
UltEstBtnReinicia = EstBtnReinicia;

if( (EstBtnEligeTanque != UltEstBtnEligeTanque) {
    if (EstBtnEligeTanque == HIGH &&
reiniciaTanque>0)
    {
        digitalWrite(pinIndicador, HIGH);
        //lcd.setCursor(0,0); lcd.print("reiniciar?");
        if(opReini==4){
            opReini=1;
        }
        else {
            opReini++;
        }
        //opReini=(opReini==4)?1:opReini+1;
        //lcd.clear();
        lcd.setCursor(0,1);
        lcd.print(F("tanque:"));
        lcd.print(opReini); //mostrar opcion
        Serial.print(F("reiniciar? tanque:"));
        Serial.println(opReini);
    }
}
else
{
}
}

```

```

        digitalWrite(pinIndicador, LOW);
    }
}
UltEstBtnEligeTanque = EstBtnEligeTanque;
}

int medida(int echo, int trigger){
    digitalWrite(trigger, LOW);
    delayMicroseconds(2);
    digitalWrite(trigger, HIGH);
    delayMicroseconds(5); //10us en estado alto
    digitalWrite(trigger, LOW);
    tiempo = pulseIn(echo, HIGH);
    return tiempo/58;
}

void mensajeAlerta(String porc)
{
    SIM900.println(F("AT+CMGF=1"));

    //String
    destino="AT+CMGS="+591+phone_user+"";
    SIM900.println(F("AT+CMGF=1")); //envio de
    mensajes en modo texto
    delay(100);
    SIM900.println(destino);
    delay(100);
    SIM900.print("Help! Disp_"+String(id_disp)+" con
    menos de "+porc+"% de liquido!");
    delay(100);
    SIM900.print(char(0x1A));
    delay(1000);
}

void CambioEstadoMoneda()
{
    if(millis()> timeCounter + 90)
    { EstBtnMoneda = 1; timeCounter = millis(); }
    // EstBtnMoneda = 1;
}

void cargaInternet(int capacidad, int actCapTanque)
{
    strcpy_P(bufferInst,
    (char*)pgm_read_word(&(tablaInst[1])));
    //AT+CREG?
    strcpy_P(bufferResp,
    (char*)pgm_read_word(&(tablaResp[1]))); //"+CREG:
    0,1"
    char direccion[100]="";
    Serial.println(F("funcion cargaInternet"));
    //char direccion[] = "GET /update/ HTTP/1.1\r\nHost:
    api.thingspeak.com\r\nConnection: close\r\n\r\n";
    //char direccion[] = "GET
    /update?api_key=QYZWPJRGFGVILR98&field1=1&
    field2=1500&field3=15000 Connection: close\r\n\r\n";
    snprintf(direccion, sizeof(direccion), "GET
    /update?api_key=%s&field1=%d&field2=%d&field3=
    %d Connection: close\r\n\r\n", wrKey, id_disp,
    capacidad, actCapTanque);
    Serial.println(direccion);
}

if (enviarAT(bufferInst, bufferResp, 1000) == 1)
//Comprueba la conexion a la red
//if (enviarAT("AT+CREG?", "+CREG: 0,1", 1000)
== 1) //Comprueba la conexion a la red
{
    enviarAT("AT+CIPSTART=\"TCP\",\"api.thingspeak.c
    om\",\"80\"", "CONNECT OK", 5000); //Inicia una
    conexion TCP con el host
    // Enviamos datos a travs del TCP
    sprintf(aux_str, "AT+CIPSEND=%d",
    strlen(direccion));
    if (enviarAT(aux_str, ">", 10000) == 1)
    {
        enviarAT(direccion, "1", 10000);
    }
    //enviarAT("AT+CIPCLOSE", "CLOSE OK",
    10000); //Cerramos la conexion
    //enviarAT("AT+CIPSHUT", "OK", 10000); //Cierra
    el contexto PDP del GPRS
    }
    else
    {
        reiniciar();
        iniciar();
    }
}

int enviarAT(String ATcommand, char* resp_correcta,
unsigned int tiempo)
{
    int x = 0;
    bool correcto = 0;
    char respuesta[100];
    unsigned long anterior;

    memset(respuesta, '\0', 100); // Inicializa el string
    delay(100);
    while ( SIM900.available() > 0) SIM900.read(); // Limpia el buffer de entrada
    SIM900.println(ATcommand); // Envia el comando
    AT
    x = 0;
    anterior = millis();
    // Espera una respuesta
    do {
        // si hay datos el buffer de entrada del UART lee y
        //comprueba la respuesta
        if (SIM900.available() != 0)
        {
            respuesta[x] = SIM900.read();
            x++;
            // Comprueba si la respuesta es correcta
            if (strstr(respuesta, resp_correcta) != NULL)
            {
                correcto = 1;
            }
        }
    }
    // Espera hasta tener una respuesta
    while ((correcto == 0) && ((millis() - anterior) <
    tiempo));
}

```

```

Serial.println(respuesta);
return correcto;
}

void iniciar()
{
    strcpy_P(bufferInst,
    (char*)pgm_read_word(&(tablaInst[1])));
    //AT+CREG?
    strcpy_P(bufferResp,
    (char*)pgm_read_word(&(tablaResp[1]))); //+CREG: 0,1

    //enviarAT("AT+CPIN=\\"1867\\\"", "OK", 1000);
    //Introducimos el PIN de la SIM
    Serial.println("Connecting net..."); 
    //delay (5000);
    //Espera hasta estar conectado a la red movil
    //enviarAT("AT+CREG?", "+CREG: 0,1", 1000)
    while ( enviarAT(bufferInst, bufferResp, 1000) == 0 )
    {
    }

    Serial.println("Connected");
    strcpy_P(bufferInst,
    (char*)pgm_read_word(&(tablaInst[2])));
    //AT+CGATT=1\r
    strcpy_P(bufferResp,
    (char*)pgm_read_word(&(tablaResp[0]))); //OK
    enviarAT(bufferInst, bufferResp, 1000); //Iniciamos la conexión GPRS
    //enviarAT("AT+CGATT=1\r", "OK", 1000);
    //Iniciamos la conexión GPRS
    //enviarAT(conexion, "OK", 3000); //Definimos el APN, usuario y clave a utilizar
    enviarAT(conexion, bufferResp, 3500); //Definimos el APN, usuario y clave a utilizar
    strcpy_P(bufferInst,
    (char*)pgm_read_word(&(tablaInst[3])));
    //AT+CSTT
    enviarAT(bufferInst, bufferResp, 3000); //Activamos el perfil de datos inalámbrico
    //enviarAT("AT+CIICR", "OK", 3000); //Activamos el perfil de datos inalámbrico
    strcpy_P(bufferInst,
    (char*)pgm_read_word(&(tablaInst[4])));
    //AT+CIICR
    enviarAT(bufferInst, "", 3000); //Activamos el perfil de datos inalámbrico
    //enviarAT("AT+CIFSR", "", 3000); //Activamos el perfil de datos inalámbrico
}

/*
void setApn()
{
    char aux_str[100];
    //AT+CGDCONT=1,"IP","internet.tigo.bo"
    snprintf(aux_str, sizeof(aux_str),
    "AT+CGDCONT=1,\"IP\",\\\"%s\\\"", apn);
    enviarAT(aux_str, "OK", 2000);
}

//AT+CSTT="internet.tigo.bo","",""
snprintf(aux_str, sizeof(aux_str),
"AT+CSTT=\\"%s\\\",\\\"%s\\\",\\\"%s\\\" ", apn, user_name,
password);
enviarAT(aux_str, "OK", 2000);

//enviarAT("AT+CSTT=\\"internet.tigo.bo\\\",\\\"\\\",\\\"\\\"", "OK", 3000); //Definimos el APN, usuario y clave a utilizar
enviarAT("AT+CIICR", "OK", 3000); //Activamos el perfil de datos inalámbrico
enviarAT("AT+CIFSR", "", 3000); //Activamos el perfil de datos inalámbrico
//Serial.println("apn listo");
}

*/
void power_off()
{
    digitalWrite(pinModem, HIGH);
    delay(1000);
    digitalWrite(pinModem, LOW);
    delay(500);
}

void reiniciar()
{
    Serial.println("restarting..."); 
    power_off();
    delay (5000);
    power_on();
}

void power_on()
{
    int respuesta = 0;
    strcpy_P(bufferInst,
    (char*)pgm_read_word(&(tablaInst[0]))); //AT
    strcpy_P(bufferResp,
    (char*)pgm_read_word(&(tablaResp[0]))); //OK

    // Comprueba que el modulo SIM900 esta arrancado
    if (enviarAT(bufferInst, bufferResp, 2000) == 0)
    //if (enviarAT("AT", "OK", 2000) == 0)
    {
        Serial.println("GPRS on...");

        digitalWrite(pinModem, HIGH);
        delay(1000);
        digitalWrite(pinModem, LOW);
        delay(1000);

        // Espera la respuesta del modulo SIM900
        while (respuesta == 0) {
            // Envia un comando AT cada 2 segundos y espera la respuesta
            //respuesta = enviarAT("AT", "OK", 2000);
            respuesta = enviarAT(bufferInst, bufferInst, 2000);
            SIM900.println(respuesta);
        }
    }
}

```

Anexo b. código fuente a nivel de software (DashBoard)

login.php

```

<?php
session_start();
require_once 'config/config.php';
$token = bin2hex(openssl_random_pseudo_bytes(16));

// If User has already logged in, redirect to dashboard
page.
if (isset($_SESSION['user_logged_in']) &&
$_SESSION['user_logged_in'] === TRUE)
{
    header('Location:index.php');
}

// If user has previously selected "remember me
option":
if (isset($_COOKIE['series_id']) &&
isset($_COOKIE['remember_token']))
{
    // Get user credentials from cookies.
    $series_id =
filter_var($_COOKIE['series_id']);
    $remember_token =
filter_var($_COOKIE['remember_token']);
    $db = getDbInstance();
    // Get user By series ID:
    $db->where('series_id', $series_id);
    $row = $db->getOne('admin_accounts');

    if ($db->count >= 1)
    {
        // User found. verify
remember token
        if
(password_verify($remember_token,
$row['remember_token']))
        {
            // Verify if expiry time is
modified.
            $expires =
strtotime($row['expires']);
            if (strtotime(date()) >
$expires)
            {
                // Remember
Cookie has expired.

                clearAuthCookie();

                header('Location:login.php');
                exit;
            }
        }
        $_SESSION['user_logged_in'] = TRUE;
        $_SESSION['admin_type'] =
$row['admin_type'];

        header('Location:index.php');
        exit;
    }
    else
    {
        clearAuthCookie();
        header('Location:login.php');
        exit;
    }
}
else
{
    clearAuthCookie();
    header('Location:login.php');
    exit;
}
}

include BASE_PATH.'/include/header.php';
?>

<div class="login-form">
<form action="authenticate.php" method="post">
<h2 class="text-center">Acceder</h2>
<div class="form-group">
<input type="text" class="form-control"
name="username" placeholder="Correo"
required="required">
<!--<input type="email" class="form-control"
name="username" placeholder="Correo"
required="required">-->
</div>
<div class="form-group">
<input type="password" class="form-control"
name="passwd" placeholder="Contraseña"
required="required">
</div>
<div class="form-group">
<button type="submit" class="btn btn-primary
btn-block">Ingresar</button>
</div>
<div class="clearfix">
<label class="pull-left checkbox-inline"><input
name="remember" type="checkbox" value="1">
Recuerdame</label><br>
<a href="#" class="pull-right">Olvidaste la
contraseña?</a>
</div>
<?php if (isset($_SESSION['login_failure'])): ?>
<div class="alert alert-
danger alert-dismissible fade in">
<a href="#" class="close"
data-dismiss="alert" aria-label="close">&times;</a>
<?php
echo
$_SESSION['login_failure'];

unset($_SESSION['login_failure']);
?>
</div>
<?php endif; ?>
</form>

```

```

<p class="text-center"><a
href="../users/registro">Crear una nueva
cuenta</a></p>

```

Index.php

```

<?php
session_start();
require_once 'config/config.php';
require_once 'include/auth_validate.php';

//Get DB instance. function is defined in config.php
$db = getInstance();

//Get Dashboard information
// $numCustomers = $db->getValue ("customers",
"count(*)");

include BASE_PATH."/include/header.php";
?>

<!-- Page Wrapper -->
<div id="wrapper">

    <!-- Sidebar -->
    <ul class="navbar-nav bg-gradient-primary sidebar
    sidebar-dark accordion" id="accordionSidebar">

        <!-- Sidebar - Brand -->
        <a class="sidebar-brand d-flex align-items-center
        justify-content-center" href="index.html">
            <div class="sidebar-brand-icon rotate-n-15">
                <i class="fas fa-laugh-wink"></i>
            </div>
            <!-- <div class="sidebar-brand-text mx-3">Grupo
            Gundlach<sup>2</sup></div>-->
            <div class="sidebar-brand-text mx-3">Grupo
            Gundlach</div>
        </a>

        <!-- Divider -->
        <hr class="sidebar-divider my-0">

        <!-- Nav Item - Dashboard -->
        <li class="nav-item active">
            <a class="nav-link" href="index.html">
                <i class="fas fa-fw fa-tachometer-alt"></i>
                <span>Dashboard</span></a>
            </li>

        <!-- Divider -->
        <hr class="sidebar-divider">

        <!-- Heading -->
        <div class="sidebar-heading">
            Dispensadores
        </div>

        <!-- Nav Item - Pages Collapse Menu -->
        <li class="nav-item">

```

```

            </div>
            <?php include BASE_PATH."/include/footer.php'; ?>

```

```

            <a class="nav-link collapsed" href="#" data-
            toggle="collapse" data-target="#collapseTwo" aria-
            expanded="true" aria-controls="collapseTwo">
                <i class="fas fa-fw fa-cog"></i>
                <span>Dispensadores</span>
            </a>
            <div id="collapseTwo" class="collapse" aria-
            labelledby="headingTwo" data-
            parent="#accordionSidebar">
                <div class="bg-white py-2 collapse-inner
                rounded">
                    <h6 class="collapse-header">Estadisticas</h6>
                    <a class="collapse-item"
                    href="buttons.html">Dispensador 1</a>
                    <a class="collapse-item"
                    href="cards.html">Dispensador 2</a>
                    <a class="collapse-item"
                    href="buttons.html">Dispensador 3</a>
                    <a class="collapse-item"
                    href="buttons.html">Dispensador 4</a>
                </div>
            </div>
        </li>

        <!-- Divider -->
        <hr class="sidebar-divider">

        <!-- Sidebar Toggler (Sidebar) -->
        <div class="text-center d-none d-md-inline">
            <button class="rounded-circle border-0"
            id="sidebarToggle"></button>
        </div>

        </ul>
        <!-- End of Sidebar -->

        <!-- Content Wrapper -->
        <div id="content-wrapper" class="d-flex flex-
        column">

            <!-- Main Content -->
            <div id="content">

                <!-- Topbar -->
                <nav class="navbar navbar-expand navbar-light
                bg-white topbar mb-4 static-top shadow">

                    <!-- Sidebar Toggle (Topbar) -->
                    <button id="sidebarToggleTop" class="btn btn-
                    link d-md-none rounded-circle mr-3">
                        <i class="fa fa-bars"></i>
                    </button>

```

```

<!-- Topbar Search -->
<form class="d-none d-sm-inline-block form-
inline mr-auto ml-md-3 my-2 my-md-0 mw-100
navbar-search">
    <div>
        <select class="form-control" id="selDisp1">
            <option value="1">Dispensador 1</option>
            <option value="2">Dispensador 2</option>
            <option value="3">Dispensador 3</option>
            <option value="4">Dispensador 4</option>
        </select>
    </div>
</form>

<!-- Topbar Navbar -->
<ul class="navbar-nav ml-auto">

    <!-- Nav Item - Search Dropdown (Visible Only
XS) -->
    <li class="nav-item dropdown no-arrow d-sm-
none">
        <a class="nav-link dropdown-toggle" href="#" id="searchDropdown" role="button" data-
        toggle="dropdown" aria-haspopup="true" aria-
        expanded="false">
            <i class="fas fa-search fa-fw"></i>
        </a>
    <!-- Dropdown - Messages -->

        <div class="dropdown-menu dropdown-menu-
        right p-3 shadow animated--grow-in" aria-
        labelledby="searchDropdown">
            <form class="form-inline mr-auto w-100
            navbar-search">
                <div class="input-group">
                    <input type="text" class="form-control bg-
                    light border-0 small" placeholder="Search for..." aria-
                    label="Search" aria-describedby="basic-addon2">
                    <div class="input-group-append">
                        <button class="btn btn-primary"
                        type="button">
                            <i class="fas fa-search fa-sm"></i>
                        </button>
                    </div>
                </div>
            </form>
        </div>
    </li>
    <!-- Nav Item - Alerts -->
    <li class="nav-item dropdown no-arrow mx-1">
        <a class="nav-link dropdown-toggle" href="#" id="alertsDropdown" role="button" data-
        toggle="dropdown" aria-haspopup="true" aria-
        expanded="false">
            <i class="fas fa-bell fa-fw"></i>
        <!-- Counter - Alerts -->
        <span class="badge badge-danger badge-
        counter">3+</span>
    </a>
    <!-- Dropdown - Alerts -->
</ul>
<div class="dropdown-list dropdown-menu
dropdown-menu-right shadow animated--grow-in" aria-
labelledby="alertsDropdown">
    <h6 class="dropdown-header">
        Notificaciones
    </h6>
    <a class="dropdown-item d-flex align-items-
    center" href="#">
        <div class="mr-3">
            <div class="icon-circle bg-primary">
                <i class="fas fa-file-alt text-white"></i>
            </div>
        </div>
        <div>
            <div class="small text-gray-500"><?php
            echo date("F j, Y, g:i a"); ?></div>
            <span class="font-weight-bold">No
            existen notificaciones</span>
        </div>
    </a>
    <a class="dropdown-item text-center small
    text-gray-500" href="#">Show All Alerts</a>
</div>
</li>
<div class="topbar-divider d-none d-sm-
block"></div>

<!-- Nav Item - User Information -->
<li class="nav-item dropdown no-arrow">
    <a class="nav-link dropdown-toggle" href="#" id="userDropdown" role="button" data-
    toggle="dropdown" aria-haspopup="true" aria-
    expanded="false">
        <span class="mr-2 d-none d-lg-inline text-
        gray-600 small"><?php echo
        filter_var($_SESSION['username']); ?></span>
        
    </a>
    <!-- Dropdown - User Information -->
    <div class="dropdown-menu dropdown-menu-
    right shadow animated--grow-in" aria-
    labelledby="userDropdown">
        <?php if($_SESSION['admin_type'] ==
        'super'): ?>
        <a class="dropdown-item"
        href="admin_users.php">
            <i class="fas fa-user fa-sm fa-fw mr-2 text-
            gray-400"></i>
            Administrar usuarios
        </a>
        <?php endif; ?>
        <a class="dropdown-item" href="#">
            <i class="fas fa-list fa-sm fa-fw mr-2 text-
            gray-400"></i>
            Cambiar Contraseña
        </a>
    </div>
</li>

```

```

<a href="logout.php" class="dropdown-item"
href="#" data-toggle="modal" data-
target="#logoutModal">
    <i class="fas fa-sign-out-alt fa-sm fa-fw mr-
2 text-gray-400"></i>
        Cerrar Sesión
    </a>
</div>
</li>
</ul>
</nav>
<!-- End of Topbar -->

<!-- Begin Page Content -->
<div class="container-fluid">
    <!-- Page Heading -->
    <div class="d-sm-flex align-items-center justify-
content-between mb-4">
        <h1 class="h3 mb-0 text-gray-
800">Dashboard</h1>
        <a href="javascript:window.print()" class="d-
none d-sm-inline-block btn btn-sm btn-primary shadow-
sm"><i class="fas fa-download fa-sm text-white-
50"></i> Generar Reporte</a>
    </div>
    <!-- Content Row -->
    <div class="row">

        <!-- Earnings (Monthly) Card Example -->
        <div class="col-xl-3 col-md-6 mb-4">
            <div class="card border-left-primary shadow-
h-100 py-2">
                <div class="card-body">
                    <div class="row no-gutters align-items-
center">
                        <div class="col mr-2">
                            <div class="text-xs font-weight-bold text-
primary text-uppercase mb-1">Recaudacion
(Mensual)</div>
                            <div id='mensual' class="h5 mb-0 font-
weight-bold text-gray-800"></div>
                        </div>
                        <div class="col-auto">
                            <i class="fas fa-calendar fa-2x text-gray-
300"></i>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <!-- Earnings (Monthly) Card Example -->
        <div class="col-xl-3 col-md-6 mb-4">
            <div class="card border-left-success shadow-
h-100 py-2">
                <div class="card-body">
                    <div class="row no-gutters align-items-
center">
                        <div class="col mr-2">
                            <div class="text-xs font-weight-bold text-
success text-uppercase mb-1">Recaudacion anual
(Acumulado)</div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <!-- Earnings (Monthly) Card Example -->
        <div class="col-xl-3 col-md-6 mb-4">
            <div class="card border-left-info shadow-
h-100 py-2">
                <div id='anual' class="h5 mb-0 font-
weight-bold text-gray-800"></div>
            </div>
        </div>
        <!-- Pending Requests Card Example -->
        <div class="col-xl-3 col-md-6 mb-4">
            <div class="card border-left-warning shadow-
h-100 py-2">
                <div class="card-body">
                    <div class="row no-gutters align-items-
center">
                        <div class="col mr-2">
                            <div class="text-xs font-weight-bold text-
warning text-uppercase mb-1">Botellas Evitadas</div>
                            <div id='evitadas' class="h5 mb-0 font-
weight-bold text-gray-800"></div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <!-- Earnings (Monthly) Card Example -->
    <div class="col-xl-3 col-md-6 mb-4">
        <div class="card border-left-warning shadow-
h-100 py-2">
            <div id='restante1' class="h5 mb-0 mr-
3 font-weight-bold text-gray-800"></div>
        </div>
        <div class="col">
            <div class="progress progress-sm mr-
2">
                <div id="restante1bar" class="progress-bar bg-info" role="progressbar"
style="width: 50%" aria-valuenow="50" aria-
valuemin="0" aria-valuemax="100"></div>
            </div>
            <div class="col">
                <div class="progress progress-sm mr-
2">
                    <div id="restante1bar" class="progress-bar bg-info" role="progressbar"
style="width: 50%" aria-valuenow="50" aria-
valuemin="0" aria-valuemax="100"></div>
                </div>
            </div>
        </div>
        <div class="col-auto">
            <i class="fas fa-prescription-bottle fa-2x
text-gray-300"></i>
        </div>
    </div>
    <!-- Pending Requests Card Example -->
    <div class="col-xl-3 col-md-6 mb-4">
        <div class="card border-left-warning shadow-
h-100 py-2">
            <div class="card-body">
                <div class="row no-gutters align-items-
center">
                    <div class="col mr-2">
                        <div class="text-xs font-weight-bold text-
warning text-uppercase mb-1">Botellas Evitadas</div>
                        <div id='evitadas' class="h5 mb-0 font-
weight-bold text-gray-800"></div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        <i class="fas fa-wine-bottle fa-2x text-gray-300"></i>
        </div>
        </div>
        </div>
        </div>
        </div>
        </div>
        </div>
        </div>

        <!-- Content Row -->
        <div class="row">
            <!-- Area Chart -->
            <div class="col-xl-4 col-lg-5">
                <div class="card shadow mb-4">
                    <!-- Card Header - Dropdown -->
                    <div class="card-header py-3 d-flex flex-row align-items-center justify-content-between">
                        <h6 class="m-0 font-weight-bold text-primary">Consumo</h6>
                    </div>
                    <!-- Card Body -->
                    <div class="card-body">
                        <div class="chart-pie" id="consumo">
                            <iframe width="100%" style="border: 1px solid #cccccc; height: 16rem" src="https://thingspeak.com/channels/694263/charts/1?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&type=line&update=15"></iframe>
                        <!-- <canvas id="myAreaChart"></canvas>-->
                        </div>
                    </div>
                    <!-- Area Chart -->
                    <div class="col-xl-4 col-lg-5">
                        <div class="card shadow mb-4">
                            <!-- Card Header - Dropdown -->
                            <div class="card-header py-3 d-flex flex-row align-items-center justify-content-between">
                                <h6 class="m-0 font-weight-bold text-primary">Capacidad Restante</h6>
                            </div>
                            <!-- Card Body -->
                            <div class="card-body">
                                <div class="chart-pie" id="consumo2">
                                    <iframe width="100%" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/694263/charts/2?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&type=line&update=15"></iframe>
                                    <!-- <canvas id="myAreaChart"></canvas>-->
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            <!-- Card Header - Dropdown -->
            <div class="card-header py-3 d-flex flex-row align-items-center justify-content-between">
                <h6 class="m-0 font-weight-bold text-primary">Historial de cambios de tanque</h6>
            </div>
            <!-- Card Body -->
            <div class="card-body">
                <div class="card shadow mb-4">
                    <!-- Card Header - Dropdown -->
                    <div class="card-header py-3 d-flex flex-row align-items-center justify-content-between">
                        <h6 class="m-0 font-weight-bold text-primary">Niveles Restantes</h6>
                    </div>
                    <div class="card-body">
                        <h4 class="small font-weight-bold">Botellon 1 <span class="float-right">70%</span></h4>
                        <div class="progress mb-4">
                            <div class="progress-bar bg-danger" role="progressbar" style="width: 70%" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"></div>
                        </div>
                        <h4 class="small font-weight-bold">Botellon 2 <span class="float-right">100%</span></h4>
                        <div class="progress mb-4">
                            <div class="progress-bar bg-warning" role="progressbar" style="width: 100%" aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>
                        </div>
                        <h4 class="small font-weight-bold">Botellon 3 <span class="float-right">100%</span></h4>
                        <div class="progress mb-4">
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```


</div>
</div>
<h4 class="small font-weight-bold">Botellon 4 <span class="float-right">20%</span></h4>
<div class="progress mb-4">
  <div class="progress-bar bg-info" role="progressbar" style="width: 20%" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"></div>
</div>
</div>
</div>
<!-- Pie Chart -->
<div class="col-12 col-lg-5">
  <div class="card shadow mb-4">
    <!-- Card Header - Dropdown -->
    <div class="card-header py-3 d-flex flex-row align-items-center justify-content-between">
      <h6 class="m-0 font-weight-bold text-primary">Nivel Restante</h6>
      <div class="dropdown no-arrow">
        <a class="dropdown-toggle" href="#" role="button" id="dropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          <i class="fas fa-ellipsis-v fa-sm fa-fw text-gray-400"></i>
        </a>
        <div class="dropdown-menu dropdown-menu-right shadow animated--fade-in" aria-labelledby="dropdownMenuLink">
          <div class="dropdown-header">Dropdown Header:</div>
          <a class="dropdown-item" href="#">Action</a>
          <a class="dropdown-item" href="#">Another action</a>
          <div class="dropdown-divider"></div>
          <a class="dropdown-item" href="#">Something else here</a>
        </div>
      </div>
    </div>
    <!-- Card Body -->
    <div class="card-body">
      <div class="chart-pie pt-4 pb-2">
        <canvas id="myPieChart"></canvas>
      </div>
      <div class="mt-4 text-center small">
        <span class="mr-2">
          <i class="fas fa-circle text-primary"></i>
        </span>
        <span class="mr-2">
          <i class="fas fa-circle text-success"></i>
        </span>
      </div>
    </div>
  </div>
</div>
<!-- Referral -->
<span class="mr-2">
  <i class="fas fa-circle text-info"></i>
</span>
</div>
</div>
</div>
</div>
<div class="col-lg-4 mb-4">
  <!-- Approach -->
  <div class="card shadow mb-4">
    <div class="card-header py-3">
      <h6 class="m-0 font-weight-bold text-primary">Otras acciones</h6>
    </div>
    <div class="card-body">
      <!-- Container Fluid -->
      <div class="container-fluid">
        <!-- Main Content -->
        <div>
          <!-- End of Main Content -->
        </div>
        <!-- Footer -->
        <footer class="sticky-footer bg-white">
          <div class="container my-auto">
            <div class="copyright text-center my-auto">
              <span>Copyright &copy; MrSonels 2020</span>
            </div>
          </div>
        </footer>
        <!-- End of Footer -->
      </div>
      <!-- Content Wrapper -->
      <div>
        <!-- End of Content Wrapper -->
      </div>
      <!-- Page Wrapper -->
      <div>
        <!-- End of Page Wrapper -->
      </div>
      <!-- Scroll to Top Button-->
      <a class="scroll-to-top rounded" href="#page-top">
        <i class="fas fa-angle-up"></i>
      </a>
      <!-- Logout Modal-->
      <div class="modal fade" id="logoutModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog" role="document">
          <div class="modal-content">
            <div class="modal-header">
              <h5 class="modal-title" id="exampleModalLabel">Está seguro de salir?</h5>
              <button class="close" type="button" data-dismiss="modal" aria-label="Close">
                <span aria-hidden="true"></span>
              </button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>


```

```

        </button>
    </div>
    <div class="modal-body">Elija "Cerrar Sesión" si desea finalizar su sesión.</div>
    <div class="modal-footer">
        <button class="btn btn-secondary" type="button" data-dismiss="modal">Cancelar</button>
        <a class="btn btn-primary" href="logout.php">Cerrar Sesión</a>
    </div>
</div>
</div>
</div>
<!-- Bootstrap core JavaScript-->
<script src="assets/vendor/jquery/jquery.min.js"></script>
<script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<!-- Core plugin JavaScript-->
<script src="assets/vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->
<script src="assets/js/sb-admin-2.min.js"></script>

<!-- Page level plugins -->
<script src="assets/vendor/chart.js/Chart.min.js"></script>

<!-- Page level custom scripts -->
<script src="assets/js/demo/chart-area-demo.js"></script>
<script src="assets/js/demo/chart-pie-demo.js"></script>

<script type="text/javascript">
$(document).ready(function(){
    carga('1');
    $("#seldisp1").change(function(){
        var op=$("#select[id=seldisp1]").val();
        carga(op);
    });
});

function carga (op){
    var url, canal1=938336, canal1=694263, campo=1, iframe1, iframe2, key1='8PJ1D7MX5NRBKRW4', key2='AB134KGGVXN4BHVD', urlsuma, urlsuma2;
    // alert(op);

    switch(op) {
        case '1':
            iframe1 = '<iframe width="100%" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/+canal1+/charts/+campo+?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&type=line&update=15&yaxis=Consumo"></iframe>';
            iframe2 = '<iframe width="100%" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/+canal1+/charts/+campo+?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&type=line&update=15&yaxis=Consumo"></iframe>';
            url="https://api.thingspeak.com/channels/+canal1+/fields/6/last.json?api_key='+key1';
            urlsuma =
            'https://api.thingspeak.com/channels/+canal1+/fields/7.json?start=2020-02-01%2010:10:10&end=2020-02-29%2010:10:10&api_key=AB134KGGVXN4BHVD
            urlsuma =
            'https://api.thingspeak.com/channels/+canal1+/fields/3.json?start=2020-02-01%2010:10:10&end=2020-02-29%2010:10:10&api_key='+key1;
            urlsuma2='https://api.thingspeak.com/channels/+canal1+/fields/3.json?end=2020-02-29%2010:10:10&api_key='+key1;
            //url='https://api.thingspeak.com/channels/+canal1+/fields/last.json?api_key='+key1+'&results=1';
            break;
        case '2':
            iframe1 = '<iframe width="100%" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/+canal1+/charts/+campo+?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&type=line&update=15&yaxis=Nivel+Restante"></iframe>';
            iframe2 = '<iframe width="100%" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/+canal1+/charts/+campo+?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=60&type=line&update=15&yaxis=Nivel+Restante"></iframe>';
            url="https://api.thingspeak.com/channels/+canal1+/fields/6/last.json?api_key='+key1';
            urlsuma =
            'https://api.thingspeak.com/channels/+canal1+/fields/7.json?start=2020-02-01%2010:10:10&end=2020-02-29%2010:10:10&api_key=AB134KGGVXN4BHVD
            urlsuma =
            'https://api.thingspeak.com/channels/+canal1+/fields/3.json?start=2020-02-01%2010:10:10&end=2020-02-29%2010:10:10&api_key='+key1;
            urlsuma2='https://api.thingspeak.com/channels/+canal1+/fields/3.json?end=2020-02-29%2010:10:10&api_key='+key1;
            //url='https://api.thingspeak.com/channels/+canal1+/fields/last.json?api_key='+key1+'&results=1';
            break;
    }
    $("#consumo").html(iframe1);
    $("#consumo2").html(iframe2);
    $("#consumo3").html(iframe3);
}

$( "#consumo" ).val( $( this ).val() );
$( "#consumo" ).html(iframe1);
$( "#consumo2" ).html(iframe2);
$( "#consumo3" ).html(iframe3);

url="https://api.thingspeak.com/channels/+canal1+/fields/6/last.json?api_key='+key1';
urlsuma =
'https://api.thingspeak.com/channels/+canal1+/fields/7.json?start=2020-02-01%2010:10:10&end=2020-02-29%2010:10:10&api_key='+key1;

```

```

urlsuma2='https://api.thingspeak.com/channels/'+canal1
+'/fields/7.json?end=2020-02-
29%2010:10:10&api_key='+key1;

//url='https://api.thingspeak.com/channels/'+canal1+'/fe
eds/last.json?api_key='+key1+'&results=1';// code
block
    break;
    case '3':
        iframe1 = '<iframe width="100%"'
        height="260" style="border: 1px solid #cccccc;"'
src="https://thingspeak.com/channels/'+canal2+'/charts/
+(campo+3+4)+"?bgcolor=%23ffff&color=%23d6202
0&dynamic=true&results=60&type=line&update=15&yaxis=
Consumo"></iframe>';
        iframe2 = '<iframe width="100%"'
        height="260" style="border: 1px solid #cccccc;"'
src="https://thingspeak.com/channels/'+canal2+'/charts/
+(campo+1)+"?bgcolor=%23ffff&color=%23d62020
&dynamic=true&results=60&type=line&update=15&y
axis=Nivel+Restante"></iframe>';
        iframe3 = '<iframe width="100%"'
        height="260" style="border: 1px solid #cccccc;"'
src="https://thingspeak.com/channels/'+canal2+'/charts/
+(campo+3)+"?bgcolor=%23ffff&color=%23d62020
&dynamic=true&results=60&type=line&update=15&y
axis=Numero+de+Tanque"></iframe>';
        //$('#consumo').val($('#this).val());
        $('#consumo').html(iframe1);
        $('#consumo2').html(iframe2);
        $('#consumo3').html(iframe3);

//https://api.thingspeak.com/channels/938336/fields/6/la
st.json?api_key=AB134KGGVXN4BHVD

url='https://api.thingspeak.com/channels/'+canal2+'/fiel
ds/2/last.json?api_key='+key2;
urlsuma =
'https://api.thingspeak.com/channels/'+canal2+'/fields/3.
json?start=2020-02-01%2010:10:10&end=2020-02-
29%2010:10:10&api_key='+key2;

urlsuma2='https://api.thingspeak.com/channels/'+canal2
+'/fields/3.json?end=2020-02-
29%2010:10:10&api_key='+key2;

//url='https://api.thingspeak.com/channels/'+canal2+'/fe
eds/last.json?api_key='+key2+'&results=1';
    break;
    case '4':
        iframe1 = '<iframe width="100%"'
        height="260" style="border: 1px solid #cccccc;"'
src="https://thingspeak.com/channels/'+canal2+'/charts/
+(campo+4)+"?bgcolor=%23ffff&color=%23d62020
&dynamic=true&results=60&type=line&update=15&y
axis=Consumo"></iframe>';
        iframe2 = '<iframe width="100%"'
        height="260" style="border: 1px solid #cccccc;"'
src="https://thingspeak.com/channels/'+canal2+'/charts/
+(campo+1+4)+"?bgcolor=%23ffff&color=%23d6202
0&dynamic=true&results=60&type=line&update=15&y
axis=Nivel+Restante"></iframe>';
        iframe3 = '<iframe width="100%"'
        height="260" style="border: 1px solid #cccccc;"'
src="https://thingspeak.com/channels/'+canal2+'/charts/
+(campo+3+4)+"?bgcolor=%23ffff&color=%23d6202
0&dynamic=true&results=60&type=line&update=15&y
axis=Numero+de+Tanque"></iframe>';
        //$('#consumo').val($('#this).val());
        $('#consumo').html(iframe1);
        $('#consumo2').html(iframe2);
        $('#consumo3').html(iframe3);

url='https://api.thingspeak.com/channels/'+canal2+'/fiel
ds/6/last.json?api_key='+key2;
urlsuma =
'https://api.thingspeak.com/channels/'+canal2+'/fields/7.
json?start=2020-02-01%2010:10:10&end=2020-02-
29%2010:10:10&api_key='+key2;

urlsuma2='https://api.thingspeak.com/channels/'+canal2
+'/fields/7.json?end=2020-02-
29%2010:10:10&api_key='+key2;

//url='https://api.thingspeak.com/channels/'+canal2+'/fe
eds/last.json?api_key='+key2+'&results=1';// code
block
    break;
}

$.ajax({
    type: 'get',
    url: url,
    //data: data,
    dataType: 'json'
})
.done(function(data){
    if(op==1 || op==3)
    {
        $('#restante1').html(parseInt(100.0*(data.field2/200.0))
        +'%');
        $('#restante1 bar').attr("style","width:
        +(100.0*(data.field2/200.0))+\"");
        $('#evitadas').html(data.entry_id);
    }
    else
    {
        $('#restante1').html(parseInt(100.0*(data.field6/200.0))
        +'%');
        $('#restante1 bar').attr("style","width:
        +(100.0*(data.field6/200.0))+\"");
        $('#evitadas').html(data.entry_id);
    }
});

$.ajax({
    type: 'get',
    url: urlsuma,
    dataType: 'json'
})

```

```

        })
        .done(function(data){
            //alert('recibiendo!'+ urlsuma+' ---'
            +data['feeds'][4].field3);
            //console.log(data);
            if(op==1 || op==3)
            {
                var cont=0, total=0, valor=0;
                $.each(data['feeds'],function(){
                    if(data['feeds'][cont].field3!=null)
                        valor =
                });
                $('#mensual').html('Bs.'+total);
            }
            else
            {
                var cont=0, total=0, valor=0;
                $.each(data['feeds'],function(){
                    if(data['feeds'][cont].field7!=null)
                        valor =
                });
                $('#mensual').html('Bs.'+total);
            }
        });
        $.ajax({
            type: 'get',
            url: urlsuma2,
            dataType: 'json'
        })
        .done(function(data){
            //alert('recibiendo!'+ urlsuma+' ---'
            +data['feeds'][4].field3);
            //console.log(data);
            if(op==1 || op==3)
            {
                var cont=0, total=0, valor=0;
                $.each(data['feeds'],function(){
                    if(data['feeds'][cont].field3!=null)
                        valor =
                });
                $('#anual').html('Bs.'+total);
            }
            else
            {
                var cont=0, total=0, valor=0;
                $.each(data['feeds'],function(){
                    if(data['feeds'][cont].field7!=null)
                        valor =
                });
                $('#anual').html('Bs.'+total);
            }
        });
        //carga(1);
    });
    </script>
<?php include_once('include/footer.php'); ?>

```

Anexo c. Set de instrucciones de comandos AT

Las siguientes listas contienen los comandos AT más usados dividida en 4 grupos:

Comandos Básicos (recordar que se les debe anteponer AT)

$^3/4$ A : responder la llamada entrante

$^3/4$ A/ : repetir el último comando (no se le antepone AT)

$^3/4$ D : configura la forma de marcado: T (por tonos), P (por pulsos)

$^3/4$ E : deshabilita el eco para la terminal

$^3/4$ E1 : habilita el eco

$^3/4$ H : cuelga la llamada

$^3/4$ I : pedido de información

$^3/4$ L : regula el volumen del sonido de salida del módem

$^3/4$ O : retorna al modo En Línea desde el modo Comando.

$^3/4$ Q : configuración para mostrar los resultados

$^3/4$ Q1 : hace que el módem no arroje resultados de las operaciones

$^3/4$ Sn? : pregunta por el valor del registro n

$^3/4$ Sn=r : ingresa el valor r al registro n

$^3/4$ V : el módem devuelve resultados numéricos

$^3/4$ V1 : el módem devuelve resultados con palabras

$^3/4$ X : reporta los códigos básicos de conexión

$^3/4$ X1 : ídem al anterior y agrega la velocidad de la conexión

$^3/4$ X2 : ídem al anterior y además detecta tono de marcado

$^3/4$ X3 : ídem X1 y además es capaz de detectar tono de ocupado

$^3/4$ X4 : El módem reporta y detecta todos los acontecimientos anteriores

$^3/4$ Z : se resetea la configuración del módem con los datos del perfil 0

$^3/4$ Z1 : se resetea la configuración del módem con los datos del perfil 1

$^3/4$ W : envía códigos de progreso de la negociación

$^3/4$ +++ : carácter de escape para volver al modo comando estando en modo en línea sin colgar la comunicación (por defecto es el carácter + pero puede configurarse otro en el registro S2). Aquí tampoco se debe anteponer el prefijo AT.

Comandos Extendidos (recordar que se les debe anteponer AT)

$^3/4$ &C : mantiene activa la señal de “Carrier Detect”(contra el otro módem).

$^3/4$ &C1 : detecta e indica “Carrier Detect” (contra el otro módem).

$^3/4$ &D : ignora la señal de “Data Terminal Ready”.

$^3/4$ &D1 : si DTR se desactiva el módem para a modo comando.

$^3/4$ &D2 : el módem cuelga la comunicación si el DTR cae.

$^3/4$ &D3 : el módem cuelga, se resetea y vuelve a modo comando si cae el DTR.

$^3/4$ &F : carga el perfil de configuración de fábrica 0.

$^3/4$ &F1 : carga el perfil de configuración de fábrica 1 (IBM-PC compatible).

$^3/4$ &F2 : carga el perfil de configuración de fábrica 2 (MAC compatible).

$^3/4$ &F3 : carga el perfil de configuración de fábrica 3 (MAC compatible).

$^3/4$ &K : deshabilita el control local de flujo.

$^3/4$ &K1 : habilita el control local de flujo por hardware (RTS/CTS).

$^3/4$ &K2 : habilita el control local de flujo por software (XON/XOFF).

$^3/4$ &Q : deshabilita el control de errores.

$^3/4$ &Q5 : selecciona el control de errores V.42 (necesita control de flujo).

$\frac{3}{4}$ &Q8 : selecciona una corrección de errores alternativa: MNP

$\frac{3}{4}$ &Q9 : condiciona la compresión: si está activo la V.42 bis desactiva el MNP5.

$\frac{3}{4}$ &U : habilita la modulación Trellis según la norma V.32.

$\frac{3}{4}$ &U1 : deshabilita la modulación Trellis según la norma V.32.

$\frac{3}{4}$ &V : muestra la configuración activa.

$\frac{3}{4}$ &W : guarda la configuración actual en el perfil 0.

$\frac{3}{4}$ &W1 : guarda la configuración actual en el perfil 1.

$\frac{3}{4}$ &Y : hace que el perfil 0 sea el activo cuando se prende el equipo.

$\frac{3}{4}$ &Y1 : hace que el perfil 1 sea el activo cuando se prende el equipo.

Nota: estos comandos no son aceptados por la totalidad de los módems y es posible que algunos de los parámetros aquí mencionados no tengan los mismos efectos en todos los módems.

Comandos Propietarios (recordar que se les debe anteponer AT)

$\frac{3}{4}$ %C : deshabilita la compresión de los datos.

$\frac{3}{4}$ %C1 : habilita la compresión MNP5.

$\frac{3}{4}$ %C2 : habilita la V.42 bis.

$\frac{3}{4}$ %C3 : habilita la MNP5 y la V.42 bis.

Registros S.

$\frac{3}{4}$ S0 : número de “rings” al cabo del cual el módem debe contestar. Si el valor es 0 el módem no contesta nunca de forma automática.

$\frac{3}{4}$ S1 : lleva la cuenta de cuantos rings han ocurrido.

$\frac{3}{4}$ S2 : contiene el número en ASCII del carácter de escape. Por default: 43 (+)

$\frac{3}{4}$ S6 : tiempo de espera para el tono de discado.

$\frac{3}{4}$ S7 : tiempo de espera para la señal de “Carrier Detect” remoto.

$\frac{3}{4}$ S12 : tiempo de guarda para pasar al modo comando luego del “+++”

$\frac{3}{4}$ S37 : Negociación de la velocidad.

Nota: la cantidad de registros que posea un módem dependen del modelo del mismo.

Solo los primeros 12 son estándar.

Fuente: <http://www.etitudela.com/fpm/comind/downloads/apendicedecomandosat.pdf>

Anexo d. Resolución SENAPI



**DIRECCIÓN DE DERECHO DE AUTOR
Y DERECHOS CONEXOS**
RESOLUCIÓN ADMINISTRATIVA NRO. 1-426-D/2020
La Paz, 11 de Diciembre del 2020

VISTOS:

La solicitud de Inscripción de Derecho de Autor presentada en fecha 07 de Diciembre del 2020 vía online, por NELSON CRISTHIAN URUCHI QUISPE, con C.I. N° 6170067 LP, con número de trámite DA 178-DIG/2020, señala la pretensión inscripción del Proyecto de Grado titulado: "DISEÑO E IMPLEMENTACIÓN DE UNA RED DE DISPENSADORES AUTOMÁTICOS DE AGUA, BASADA EN INTERNET DE LAS COSAS, PARA EL ANÁLISIS DE VENTA DE AGUA POR PARTE DE LA DIVISIÓN DE PURIFICACIÓN DE AGUA DEL GRUPO GUNDLACH", cuyos datos y antecedentes se encuentran adjuntos y expresados en el Formulario de Declaración Jurada.

CONSIDERANDO

Que, en observación al Artículo 4º del Decreto Supremo N° 27938 modificado parcialmente por el Decreto Supremo N° 28152 el "Servicio Nacional de Propiedad Intelectual SENAPI, administra en forma descentralizada e integral el régimen de la Propiedad Intelectual en todos sus componentes, mediante una estricta observancia de los regímenes legales de la Propiedad Intelectual, de la vigilancia de su cumplimiento y de una efectiva protección de los derechos de exclusiva referidos a la propiedad industrial, al derecho de autor y derechos conexos; constituyéndose en la oficina nacional competente respecto de los tratados internacionales y acuerdos regionales suscritos y adheridos por el país, así como de las normas y regímenes comunes que en materia de Propiedad Intelectual se han adoptado en el marco del proceso andino de integración".

L

Que, el Artículo 16º del Decreto Supremo N° 27938 establece "Como núcleo técnico y operativo del SENAPI funcionan las Direcciones Técnicas que son las encargadas de la evaluación y procesamiento de las solicitudes de derechos de propiedad intelectual, de conformidad a los distintos regímenes legales aplicables a cada área de gestión". En ese marco, la Dirección de Derecho de Autor y Derechos Conexos otorga registros con carácter declarativo sobre las obras del ingenio cualquiera que sea el género o forma de expresión, sin importar el mérito literario o artístico a través de la inscripción y la difusión, en cumplimiento a la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina, Ley de Derecho de Autor N° 1322, Decreto Reglamentario N° 23907 y demás normativa vigente sobre la materia.

Que, la solicitud presentada cumple con: el Artículo 6º de la Ley N° 1322 de Derecho de Autor, el Artículo 26º inciso a) del Decreto Supremo N° 23907 Reglamento de la Ley de Derecho de Autor, y con el Artículo 4º de la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina.

Que, de conformidad al Artículo 18º de la Ley N° 1322 de Derecho de Autor en concordancia con el Artículo 18º de la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina, referentes a la duración de los Derechos Patrimoniales, los mismos establecen que: "la duración de la protección concedida por la presente ley será para toda la vida del autor y por 50 años después de su muerte, a favor de sus herederos, legatarios y cesionarios".

Que, se deja establecido en conformidad al Artículo 4º de la Ley N° 1322 de Derecho de Autor, y Artículo 7º de la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina que: "...No son objeto de protección las ideas contenidas en las obras literarias, artísticas, o el contenido ideológico o técnico de las obras científicas ni su aprovechamiento industrial o comercial".

Que, el Decreto Supremo N° 4218 de 14 de abril de 2020, regula el teletrabajo como una modalidad especial de prestación de servicios caracterizada por la utilización de Tecnologías de la Información y Comunicación – TIC, en los sectores públicos y

privados, estableciendo a través de su Artículo 12 que "con el objeto de implementar y promover el teletrabajo, las entidades públicas, deben desarrollar e implementar una estrategia de digitalización para la atención de trámites y servicios en línea en el marco del Plan de Implementación del Gobierno Electrónico ...".

Que, mediante Resolución Administrativa N° 14/2020 del 22 de abril de 2020, el Director General Ejecutivo del Senapi, Resuelve: "... Aprobar el Reglamento para trámites On Line de la Dirección de Derecho de Autor y Derechos Conexos del Servicio Nacional de Propiedad Intelectual...".

Que, el artículo 4, inciso e) de la ley 2341 de Procedimiento Administrativo, instituye que: "... en la relación de los particulares con la Administración Pública, se presume el principio de buena fe. La confianza, la cooperación y la lealtad en la actuación de los servidores públicos y de los ciudadanos ...", por lo que se presume la buena fe de los administrados respecto a las solicitudes de registro y la declaración jurada respecto a la originalidad de la obra.

POR TANTO

La Directora de Derecho de Autor y Derechos Conexos sin ingresar en mayores consideraciones de orden legal, en ejercicio de las atribuciones conferidas.



RESUELVE:

INSCRIBIR en el Registro de Tesis, Proyectos de Grado, Monografías y Otras Similares de la Dirección de Derecho de Autor y Derechos Conexos, el Proyecto de Grado titulado: "DISEÑO E IMPLEMENTACIÓN DE UNA RED DE DISPENSADORES AUTOMÁTICOS DE AGUA, BASADA EN INTERNET DE LAS COSAS, PARA EL ANÁLISIS DE VENTA DE AGUA POR PARTE DE LA DIVISIÓN DE PURIFICACIÓN DE AGUA DEL GRUPO GUNDLACH", a favor del autor y titular: NELSON CRISTHIAN URUCHI QUISPE, con C.I. N° 6170067 LP., bajo el seudónimo MRSNELS, quedando amparado su derecho conforme a Ley, salvando el mejor derecho que terceras personas pudieren demostrar.

Registrese, Comuníquese y Archívese.



Abg. Gabriela Arancibia Pérez
DIRECTORA DE DERECHO DE AUTOR
Y DERECHOS CONEXOS
SERVICIO NACIONAL DE PROPIEDAD INTELECTUAL

CELULAR: 79512858

CORREO: cristhian.uruchi@gmail.com