

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

IMPLEMENTACIÓN DE UN PROTOTIPO PARA EXTRACCIÓN DE AGUA DE UN POZO MEDIANTE LA MEDICIÓN DE CAUDAL Y NIVEL DEL AGUA UTILIZANDO LA PLATAFORMA ARDUINO Y APLICACIÓN MÓVIL ANDROID

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES**

EDISON XAVIER QUINGATUÑA TASIGCHANA

DIRECTOR: ING. RICARDO LLUGSI, MSc.

Quito, enero 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Edison Xavier Quingatuña Tasigchana, bajo mi supervisión.

Ing. Ricardo Llugsí, MSc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Edison Xavier Quingatuña Tasigchana, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Edison Xavier Quingatuña Tasigchana

DEDICATORIA

A mis hijos Lexandre y Sayda Quingatuña quienes me apoyaron con su paciencia y locuras para la culminación de mi carrera universitaria.

A mi esposa Verónica Patricia por darme; una familia, un hogar y lo más valioso de su vida, su tiempo y apoyo incondicional en todo momento.

AGRADECIMIENTO

A Dios, por darme la sabiduría de escoger el camino correcto y adecuado; y por darme las fuerzas para afrontar situaciones difíciles de la vida.

A mis padres María Tasigchana y Segundo Quingatuña, por darme la vida y amor incondicional en todo momento.

A mi hermano Henry, por el apoyo e incentivo en todos los momentos complicados durante este tiempo.

A mis padrinos Isabel y Cesar, por guiarme y darme consejos sabios de vida.

Al Ingeniero Ricardo Llugsi, por su tiempo en la revisión de este proyecto.

A mis amig@s que he tenido la dicha de conocerlos en la trayectoria universitaria y formar un vínculo de una fuerte de amistad.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	IX
ABSTRACT	X
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS	1
1.2 ALCANCE	2
1.3 MARCO TEÓRICO.....	6
1.3.1 ESTACIÓN DE BOMBEO DE AGUA.....	6
1.3.1.1 Tipos de estaciones de bombeo de agua	7
1.3.2 BOMBAS DE AGUA	8
1.3.2.1 Bombas de desplazamiento positivo (BDP).....	9
1.3.2.2 Bombas dinámicas	9
1.3.3 FUENTES DE ABASTECIMIENTO DE AGUA.....	9
1.3.3.1 Fuentes subterráneas.....	10
1.3.3.2 Fuentes superficiales.....	10
1.3.3.3 Fuentes pluviales.....	11
1.3.4 MEDICIÓN DE NIVEL DE AGUA.....	11
1.3.4.1 Ondas ultrasónicas.....	11
1.3.4.2 Tipos de emisión sónica	12
1.3.4.3 Técnica impulso-eco para emisión y recepción de ultrasonidos.....	13
1.3.4.4 Medidor ultrasónico	14
1.3.5 MEDICIÓN DE CAUDAL VOLUMÉTRICO	15
1.3.5.1 Medidor de turbina.....	15
1.3.5.2 Convertidores de señal.....	16
1.3.5.3 Principio de funcionamiento del medidor de turbina.....	17
1.3.6 TECNOLOGÍAS INALÁMBRICAS	19
1.3.6.1 Tecnología bluetooth	19
1.3.6.2 Tecnología celular	23

2.	METODOLOGÍA	26
2.1	DISEÑO DEL PROTOTIPO	26
2.1.1	CARACTERÍSTICAS FÍSICAS DEL SISTEMA DE BOMBEO.....	27
2.1.1.1	Características físicas del pozo de agua.....	27
2.1.1.2	Características físicas del tanque de almacenamiento	30
2.1.2	REQUERIMIENTOS TÉCNICOS DEL SISTEMA DE BOMBEO	31
2.1.2.1	Requerimientos técnicos en la parte de succión	31
2.1.2.2	Requerimientos técnicos en la parte de descarga	33
2.1.3	DIAGRAMA HIDRÁULICO DEL SISTEMA DE BOMBEO	36
2.1.4	DIMENSIONAMIENTO DE LA BOMBA DE AGUA	37
2.1.4.1	Propiedades de los fluidos.....	38
2.1.4.2	Regímenes del flujo del fluido	40
2.1.4.3	Ecuación general de energía	41
2.1.4.4	Pérdidas de energía en la tubería.....	43
2.1.4.5	Principio de la continuidad.....	45
2.1.4.6	Altura total de presión de retención (NPSH)	47
2.1.4.7	Cálculos del dimensionamiento de la bomba.....	47
2.2	SELECCIÓN DEL HARDWARE.....	55
2.2.1	PLACAS DE ARDUINO	55
2.2.1.1	Descripción de la placa Arduino Mega 2560.....	55
2.2.1.2	Características de la placa Arduino Mega 2560.....	56
2.2.2	MÓDULO DE CONEXIÓN BLUETOOTH	56
2.2.2.1	Descripción del módulo HC-05	56
2.2.2.2	Especificaciones técnicas del módulo HC-05	57
2.2.2.3	Conexión del módulo HC-05 y Arduino Mega 2560	57
2.2.3	MÓDULO DE CONEXIÓN GSM/GPRS	58
2.2.3.1	Descripción del módulo GSM/GPRS SIM 900	58
2.2.3.2	Especificaciones técnicas del módulo GSM/GPRS SIM 900.....	58
2.2.3.3	Conexión del módulo SIM 900 y Arduino Mega 2560	59
2.2.4	MÓDULO ADAPTADOR DE TARJETA MICRO_SD.....	59
2.2.4.1	Descripción del adaptador de tarjeta Micro_SD	59
2.2.4.2	Especificaciones técnicas del adaptador de tarjeta Micro_SD	60
2.2.4.3	Conexión del adaptador Micro_SD y Arduino Mega 2560.....	60
2.2.5	CONTACTOR.....	61
2.2.5.1	Descripción del relé	61
2.2.5.2	Especificaciones técnicas del relé de un canal	61

2.2.5.3	Conexión del relé y Arduino Mega 2560	62
2.2.6	SERVO MOTOR.....	62
2.2.6.1	Descripción del servo motor MG996R	62
2.2.6.2	Especificaciones técnicas del Servo MG996R	63
2.2.6.3	Conexión servo MG996R y Arduino Mega2560	63
2.2.7	SENSOR ULTRASÓNICO.....	63
2.2.7.1	Descripción del módulo HC-SR04	64
2.2.7.2	Especificaciones técnicas del módulo HC-SR04.....	64
2.2.7.3	Conexión del módulo HC-SR04 y Arduino Mega 2560	65
2.2.8	SENSOR DE CAUDAL	65
2.2.8.1	Descripción del módulo YF-S201.....	65
2.2.8.2	Especificaciones técnicas del módulo YF-S201	66
2.2.8.3	Conexión del módulo YF-S201 y Arduino Mega 2560.....	66
2.2.9	LED RGB.....	66
2.2.9.1	Descripción del Led RGB cátodo común	67
2.2.9.2	Especificaciones técnicas del Led RGB cátodo común.....	67
2.2.9.3	Conexión del Led RGB y Arduino Mega 2560	67
2.2.10	FUENTE DE ALIMENTACIÓN.....	68
2.2.11	DISPOSITIVO DE VISUALIZACIÓN DE DATOS.....	68
2.2.11.1	Especificaciones técnicas del smartphone SM-G532M.....	69
2.2.12	VÁLVULAS.....	69
2.2.12.1	Válvula de pie	70
2.2.12.2	Válvula esférica	70
2.2.13	ACCESORIOS.....	70
2.2.13.1	Codos.....	70
2.2.13.2	Unión.....	71
2.2.13.3	Reducción	71
2.2.13.4	Tubería.....	71
2.2.14	BOMBA DE AGUA.....	71
2.3	CALIBRACIÓN DE SENSORES	73
2.4	IMPLEMENTACIÓN	74
2.4.1	SISTEMA DE TRANSMISIÓN	74
2.4.2	SISTEMA DE RECEPCIÓN.....	74
2.4.3	MONTAJE DEL PROTOTIPO.....	75
2.5	PROGRAMACIÓN DEL SISTEMA.....	76
2.5.1	HERRAMIENTAS DE PROGRAMACIÓN.....	76

2.5.1.1	Plataforma Arduino	76
2.5.1.2	App Inventor	77
2.5.2	FLUJOGRAMA DE AUTOMATIZACIÓN DE BOMBEO DE AGUA	78
2.5.3	PROGRAMACIÓN DEL MICROCONTROLADOR ATMEL	82
2.5.3.1	Declaración de librerías	82
2.5.3.2	Declaración de variables globales	82
2.5.3.3	Configuración del puerto serial del módulo GSM/GPRS SIM900	83
2.5.3.4	Configuración de la velocidad de los puertos seriales.....	84
2.5.3.5	Configuración del módulo GSM/GPRS SIM90	84
2.5.3.6	Configuración del adaptador de la tarjeta Micro_SD	84
2.5.3.7	Funciones de programación	85
2.5.3.8	Grabación de datos en tarjeta Micro_SD	86
2.5.3.9	Transmisión de datos	87
2.5.4	FLUJOGRAMA DE PROGRAMACIÓN DE LA APP MÓVIL.....	87
2.5.5	PROGRAMACIÓN EN APP INVENTOR.....	88
2.5.5.1	Componentes de App Inventor	89
2.5.5.2	Código de programación de la App Móvil	89
3.	RESULTADOS Y DISCUSIÓN	94
3.1	PRUEBAS DE FUNCIONAMIENTO.....	94
3.1.1	INICIALIZACIÓN DEL SISTEMA	94
3.1.2	CONECTIVIDAD DEL SISTEMA	96
3.1.2.1	Conectividad con la tecnología bluetooth.....	96
3.1.2.2	Conectividad con la tecnología celular.....	98
3.1.3	MODO DE OPERACIÓN DEL SISTEMA.....	99
3.1.3.1	Modo de operación standby.....	99
3.1.3.2	Modo de operación automático.....	100
3.1.4	PRUEBAS DEL SISTEMA DE ALARMAS	103
3.2	ANÁLISIS DE LOS RESULTADOS.....	105
3.3	COSTOS REFERENCIALES	107
4.	CONCLUSIONES Y RECOMENDACIONES.....	109
4.1	CONCLUSIONES.....	109
4.2	RECOMENDACIONES	111
5.	REFERENCIAS BIBLIOGRÁFICAS	112
	ANEXOS	117

RESUMEN

El presente proyecto tiene como propósito principal extraer agua de un pozo y almacenarla en un tanque de reserva de manera automática, manteniendo un nivel de agua suficiente y un caudal volumétrico considerable para las necesidades del hogar.

El prototipo se podrá monitorear y controlar mediante un teléfono celular por medio de una aplicación móvil desarrollada en App Inventor y/o por medio de mensajes de texto.

Para el desarrollo del prototipo, el proyecto está dividido en cuatro capítulos.

En el primer capítulo se presenta los fundamentos teóricos, de un sistema de extracción de agua, fuentes de abastecimiento del líquido vital, tipos de sistemas de extracción de agua, tipos de bombas de agua, método para medir el nivel de agua y caudal volumétrico en una tubería; y por último se describen las tecnologías inalámbricas utilizadas en el desarrollo del proyecto.

En el segundo capítulo se describe la metodología para el diseño del prototipo, selección del hardware, calibración de los sensores, implementación del prototipo, programación del microcontrolador y programación de la aplicación móvil.

En el tercer capítulo se efectúa las pruebas de funcionamiento del prototipo, donde se evaluará los modos de operación del sistema, además se realizará un análisis de resultados y costos referenciales para la implementación del sistema.

Finalmente, en el cuarto capítulo se presenta conclusiones y recomendaciones de acuerdo con el desarrollo y operación del prototipo.

PALABRAS CLAVE: Nivel de agua, flujo volumétrico, comunicación inalámbrica, sensores.

ABSTRACT

The main purpose of this project is to extract water from a well and store it in a reserve tank automatically, maintaining a sufficient water level and a considerable volumetric flow rate for household needs.

The prototype can be monitored and controlled by a cell phone by means of a mobile application developed in App Inventor and/or by means of text messages.

For the development of the prototype, the project is divided into four chapters.

The first chapter presents the theoretical foundations of a water extraction system, vital liquid supply sources, types of water extraction systems, types of water pumps, method for measuring water level and volumetric flow in a pipe; and finally describe the wireless technologies used in the development of the project.

The second chapter describes the methodology for prototype design, hardware selection, sensor calibration, prototype implementation, microcontroller programming, and mobile application programming.

The third chapter performs the prototype operation tests, where the operating modes of the system will be evaluated, and a result and reference cost analysis will be performed for the system implementation.

Finally, the fourth chapter presents conclusions and recommendations in accordance with the development and operation of the prototype.

KEY WORDS: Water level, volumetric flow, wireless communication, sensors.

1. INTRODUCCIÓN

El trabajo de titulación que se presenta a continuación intenta solventar o solucionar una problemática que ha existido por años y que se mantiene hasta la presente, como es el acceso y uso del líquido vital por medio de la red pública en las partes rurales del país.

Existen algunas formas de obtener el líquido vital (agua) en los sectores rurales del país, entre las cuales se da por medio de las fuentes de agua subterráneas que generalmente son los pozos, además se tiene otras maneras de acceder al líquido por medio de carros repartidores, vertientes, entre otros.

En caso de disponer de un pozo de agua se propone diseñar e implementar un prototipo de extracción o bombeo de agua de manera automática, controlado y monitoreado por medio de un celular.

Aprovechando los servicios que brinda el teléfono celular se propone diseñar una aplicación móvil para un Smartphone que disponga del Sistema Operativo Android para controlar y monitorear el prototipo mediante una interfaz gráfica y/o por medio de mensajes de texto por medio de la red celular.

Con el propósito de evitar el trabajo manual y optimizar el tiempo de las personas, se propone realizar un sistema automático de extracción de agua para luego almacenarla en el tanque de almacenamiento de concreto.

Mediante la medición del nivel del agua por medio de ondas sonoras se pretende mantener un registro del volumen o nivel de agua que existe en el pozo y en el tanque de almacenamiento, también se medirá el caudal volumétrico que atraviesa por la tubería por medio de un dispositivo caudalímetro que ayudará a tener un registro de la cantidad de agua extraída del pozo.

Para la construcción del presente prototipo se utilizará elementos electrónicos, dispositivos, accesorios y herramientas tecnológicas que existen en el mercado para obtener un sistema eficiente, económico y fácil de interactuar y operar.

1.1 OBJETIVOS

El objetivo general de este estudio técnico es: Implementar un prototipo para extracción de agua de un pozo mediante la medición de caudal y nivel del agua utilizando la plataforma Arduino y aplicación móvil Android.

Los objetivos específicos de este estudio técnico son:

- Describir las características de la bomba de agua eléctrica centrífuga, plataforma Arduino, módulos, sensores, led RGB, diodo led, shields y del IDE (*Integrated Development Environment*, Entorno de Desarrollo Integrado), así como también de la plataforma MIT APP INVENTOR y el sistema operativo Android.
- Realizar la programación en el IDE de Arduino y cargar el programa en el microcontrolador.
- Realizar el diseño y programación de la aplicación móvil para un Smartphone con sistema operativo Android.
- Montar los módulos en los puertos GPIO (*General Purpose Input/Output*, Entrada/Salida de Propósito General) del microcontrolador ATMEL e implementar la aplicación móvil en el Smartphone para el control y monitoreo del sistema.
- Comprobar el respectivo funcionamiento del sistema.

1.2 ALCANCE

La implementación del prototipo tiene como propósito automatizar la succión de agua cruda de un pozo, mediante el monitoreo de los niveles de agua del pozo como del tanque de almacenamiento, la cual será utilizada para cuestiones domésticas, limpieza y de regadío de ser necesario.

Para la implementación del prototipo se utilizará la plataforma Arduino, junto con dos sensores ultrasónicos que servirán para medir los niveles de agua del pozo y del tanque de almacenamiento; y así poder calcular la capacidad o volumen de abastecimiento de agua que tendrá el pozo hacia el tanque de almacenamiento, ya que no se puede medir el caudal interno del pozo porque es una fuente de agua natural que brota del suelo hacia la superficie. Dependiendo de la relación de estos niveles o volúmenes se activará una bomba de agua eléctrica mediante un contactor o relay para la extracción del líquido del pozo para luego ser transferida al tanque de almacenamiento. Previamente esta bomba será dimensionada de acuerdo con las características y requerimientos del sistema.

Además, se implementará un plan de alarmas para evitar la succión en vacío de la bomba de agua eléctrica la cual estará en función del volumen o niveles de agua mínimo y máximo que se encuentre en el pozo y/o en el tanque de almacenamiento. Este sistema de alarmas contará de dos leds RGB, uno representará la capacidad o nivel de agua en el pozo y el otro en el tanque de almacenamiento, dependiendo de los volúmenes o niveles se

encenderá un color del led RGB para indicar la capacidad de agua en el pozo y/o en el tanque de almacenamiento.

También se realizará un acoplamiento mecánico de un servomotor a una válvula esférica para la salida de agua que llegará al tanque de almacenamiento, y de igual manera esta válvula esférica se abrirá/cerrará dependiendo de la señal de on/off de la bomba de agua eléctrica, también se instalará un caudalímetro para obtener información que va al Arduino para monitorear el flujo mínimo y/o máximo de agua que atraviesa por la tubería el cual ayudará a tener una alerta de la operación de la bomba de agua y que estará representado por un diodo led, si la bomba está operando de manera incorrecta este led se apagará caso contrario se mantendrá encendido, además con estas medidas de flujo se pretende determinar la cantidad de agua extraída del pozo la cual ayudará al usuario a planificar sus actividades diarias de limpieza, domésticas y de regadío.

Todos estos datos serán almacenados en una tarjeta micro_SD mediante un módulo SD card para tener registros de los eventos del sistema.

Para la comunicación inalámbrica, transmisión y visualización de datos se utilizará un módulo Bluetooth para sincronizar el prototipo con la aplicación móvil desarrollada del Smartphone, además mediante un módulo GSM/GPRS y junto con una tarjeta SIM se utilizará la red celular para monitorear y controlar el sistema por medio de mensajes de texto ya que el sistema que se menciona será de gran utilidad para los usuarios ubicados en la parte rural del país.

La aplicación móvil será desarrollada en la plataforma MIT APP INVENTOR para un Smartphone con sistema operativo Android en el cual se podrá visualizar el estado del sistema como por ejemplo estado de alarmas, encendido de la bomba de agua eléctrica y activación de servomotor junto con la válvula esférica. También se podrá observar las medidas tomadas por los sensores como: el nivel de líquido que existe tanto en el pozo como en el tanque de almacenamiento, la cantidad de flujo de agua que atraviesa por la tubería y también la cantidad de agua extraída.

Este prototipo será alimentado por medio de la red eléctrica y se implementará en el vecindario ubicado al Sur de Quito en la parroquia La Argelia barrio Hierba Buena 1, para fines de funcionamiento de prueba se utilizará un pozo de agua de forma cilíndrica de medidas de 1.4 [m] de diámetro y 7[m] de profundidad, también se utilizará un tanque de almacenamiento de concreto de forma paralelepípedo rectangular de medidas 1.9 [m] de largo, 0.73 [m] de alto y 0.86 [m] de profundidad.

El alcance final que busca el presente trabajo es obtener un producto final demostrable, que consiste en un prototipo de extracción de agua cruda de un pozo de manera automática, y que será controlado y monitoreado por la red celular y/o una aplicación móvil.

En la figura 1.1 se presenta como será implementado el prototipo y el funcionamiento de este.

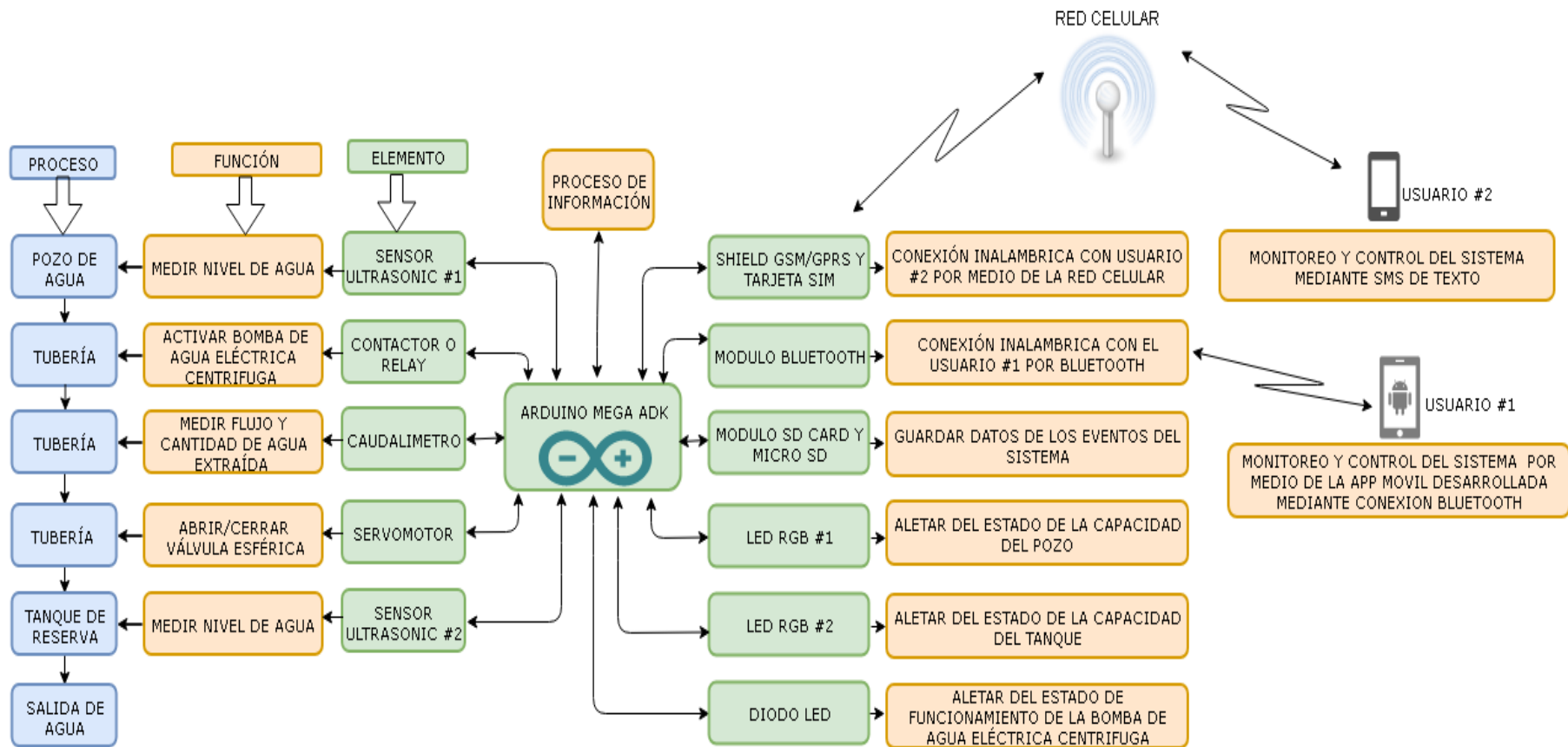


Figura 1.1 Diagrama del prototipo a implementar

1.3 MARCO TEÓRICO

Cerca de 2 millones de personas de la población en el Ecuador lo que representa el 15% de la población total no accede al consumo y uso del agua potable según los registros del Instituto Nacional de Estadística y Censos (INEC) y del Ministerio de Desarrollo Urbano y Viviendas del Ecuador (MIDUVI); y que básicamente sus fuentes de abastecimiento de agua se da por medio de pozos, ríos, vertientes, carros repartidores y otros [1].

Además, existe una mala práctica de conservación de agua en el Ecuador [2]. Los estudios realizados por la Organización Mundial de la Salud (OMS) dice que el gasto mínimo de agua debe ser de 100 litros/día para higiene y consumo para garantizar una buena salud de un habitante, no obstante La Ley de Recursos Hídricos del país establece un consumo de 200 litros/día [3] por habitante, pero sin embargo la realidad en el Ecuador es que una persona consume o gasta 237 litros/día, lo cual ocasiona una mala repartición de agua a ciertos sectores de la ciudad produciendo un flujo deficiente para las necesidades del hogar.

Existen varios tipos de sistemas de extracción o bombeo de agua que van desde sistemas automático, semiautomático o totalmente manual, siendo el automático el sistema más costoso en la actualidad e incluso difícil de operar para las personas que no tengan conocimiento sobre estos tipos de sistemas, ya que son enfocados a dar soluciones industriales y no domésticas.

En el apartado actual se revisará conceptos teóricos y además se realizará una síntesis de los siguientes temas: estaciones de bombeo de agua, bombas de agua, fuentes de abastecimiento de agua, medición de nivel de agua, medición de caudal volumétrico y también de las tecnologías inalámbricas de comunicaciones que se utilizará para el control y monitoreo del sistema.

1.3.1 ESTACIÓN DE BOMBEO DE AGUA

El funcionamiento de una estación de bombeo de agua prácticamente es tomar el agua directa o indirectamente de alguna fuente de abastecimiento e impulsar a un tanque de almacenamiento o directamente a la red de distribución de agua. Una estación de bombeo de agua consta de varios elementos o accesorios entre los cuales se pueden mencionar las tuberías, los equipos, las estructuras civiles, entre otros [4], además una estación de bombeo también se la conoce como sistemas de bombeo.

1.3.1.1 Tipos de estaciones de bombeo de agua

Básicamente existen dos tipos de estaciones de bombeo de agua y como tal se dividen en estaciones primarias y secundarias, misma que dependiendo de la infraestructura o sitio de la instalación se deberá escoger el tipo de estación más adecuado.

A. Estaciones primarias.

En este tipo de estaciones de bombeo de agua, su proceso de aspiración se puede observar en la figura 1.2 donde se realiza de la siguiente manera, primero se toma el agua de alguna fuente de abastecimiento o de alguna cámara, y luego se la eleva a otro tanque de almacenamiento para su respectivo tratamiento bacteriológico, o simplemente se envía directo al servicio de repartición de agua [5].

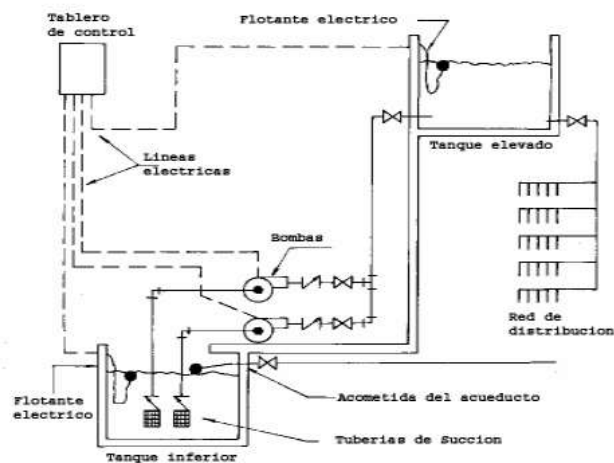


Figura 1.2 Estación de bombeo primaria de una cámara [6]

Además, se puede construir dos tipos de estaciones primarias que dependiendo de la necesidad o requerimientos de la instalación se puede construir estaciones primarias de dos cámaras o de una cámara.

- **Estaciones de dos cámaras.**

Estas estaciones de bombeo básicamente constan de dos cámaras, en la primera cámara se tendrá la entrada del líquido y un depósito que sirva para conectar la succión; y en la segunda cámara que se conoce como cámara seca sirve para colocar los equipos de bombeo. Además la primera cámara como tal no puede existir, si se dispone de una fuente de abastecimiento natural [5].

- **Estaciones de una cámara.**

Las estaciones de bombeo de una cámara generalmente se usan para sistemas donde se introduce una bomba de eje vertical o una bomba sumergible. Por lo tanto, los equipos de bombeo, la entrada del agua y el almacenamiento se tendrá dentro de la misma cámara que se denomina cámara única [5].

B. Estaciones secundarias.

En este tipo de estaciones de bombeo, como se exhibe en la figura 1.3 el sistema de succión constan de varios equipos como por ejemplo un compresor, tanque de presión, manómetro, entre otros, todos estos equipos tienen como finalidad ayudar a mejorar las condiciones de succión de una estación primaria aumentando la presión, pero con la alimentación de una estación de bombeo primaria [5].

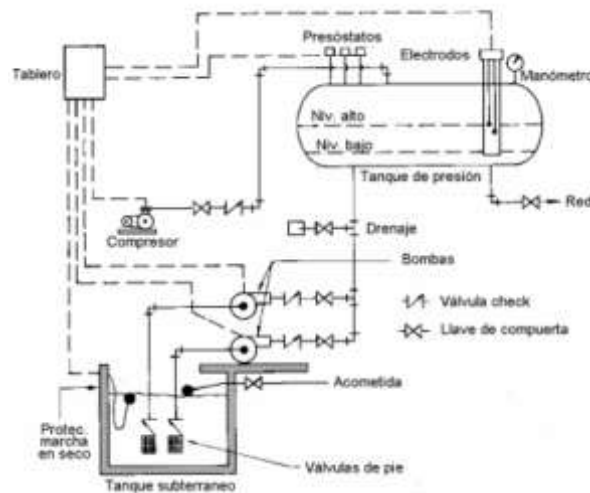


Figura 1.3 Estación de bombeo secundaria [6]

El desarrollo de este proyecto de titulación está implementado sobre una estación primaria de bombeo tal como se menciona en la figura 1.2.

1.3.2 BOMBAS DE AGUA

Como tal la bomba es el equipo primordial dentro de un sistema de bombeo, que realiza la función de succionar, extraer e impulsar un fluido en una determinada dirección o trayectoria. Para esto la bomba ejecuta el siguiente proceso, realiza la conversión de la energía proporcionada por un elemento motriz en energía de fluido, ya sea energía potencia, cinética o de presión [7].

Dentro del mercado comercial existe una infinidad de bombas ya sea por su capacidad, modelo, diseño, etc. Pero sin embargo para una mejor clasificación de las bombas se

dispone de dos grandes grupos como son las bombas de desplazamiento positivo y las bombas dinámicas.

1.3.2.1 Bombas de desplazamiento positivo (BDP)

La estructura interna de las BDP básicamente consta de un contacto móvil que, por variación de volumen, exige al fluido a recorrer a través de la máquina, donde se abre una cavidad en la que el fluido ingresa a través de la toma. Después se cierra la cavidad y se arroja el fluido por la abertura de salida. Todas las BDP proporcionan un flujo volumétrico pulsante o periódico como resultado de que la cavidad se abre, atrapa y expulsa al fluido [8].

La principal utilidad de este tipo de bombas es que son auto-cebantes en la mayor parte de aplicaciones y puede bombear cualquier fluido, independientemente de la viscosidad.

1.3.2.2 Bombas dinámicas

En este tipo de bombas la estructura interna dispone de alabes giratorios o paletas que ayudan a generar cantidad de movimiento al fluido. No existe volúmenes cerrados, el fluido incrementa su cantidad de movimiento mientras se desplaza a través de pasajes abiertos, para transformar después su alta velocidad en aumento de presión al salir de un difusor [8].

Estas bombas son poco efectivas para succionar fluidos muy viscosos, y generalmente necesitan ser cebadas y además proporcionan una descarga más estacionaria [8].

En este grupo están las bombas periféricas que son económicas, eficientes, ligeras y silenciosas aptas para sistemas pequeños de bombeo ofreciendo una alta presión, pero con un bajo caudal.

1.3.3 FUENTES DE ABASTECIMIENTO DE AGUA

Las fuentes de suministro de líquido deben ser permanentes y suficientes, ya que deben de abastecer el líquido a un grupo considerado de usuarios, para esto se debe realizar algunos estudios previos del sitio entre los cuales: estudio demográfico, estudio hidrográfico, estudios geológicos, estudios topográficos, entre otros.

Existen fuentes de agua que debido a su presentación por la naturaleza pueden ser subterráneas, superficiales y pluviales; a continuación, se describen cada una de ellas.

1.3.3.1 Fuentes subterráneas

Las fuentes protegidas de aprovisionamiento subterráneas están libres de microorganismos patógenos lo cual podría ser apto para el consumo humano. Pero sin embargo siempre será necesario saber las propiedades del agua, para lo cual se requiere realizar estudios fisicoquímicos y bacteriológicos [9].

Normalmente la captación de agua subterránea se puede efectuar a través de: manantiales, pozos (ver figura 1.4), nacientes, acuíferos, entre otros.



Figura 1.4 Fuente de agua subterránea (pozo) [10]

1.3.3.2 Fuentes superficiales

Los ríos, mares, lagos, arroyos entres otros son los principales lugares donde se capta el líquido de manera superficial, este tipo de fuente de abastecimiento de agua no es aconsejable para el consumo de manera directa para el humano, en vista que presenta muchos contaminantes derivados de la descarga de desagües domésticos, residuos de actividades mineras o industriales, uso de defensivos agrícolas, presencia de animales, residuos sólidos, y otros [9].

En la figura 1.5 se contempla la captación del líquido a través de un rio que dispone de un excelente caudal.



Figura 1.5 Fuente de abastecimiento superficial [11]

1.3.3.3 Fuentes pluviales

En la figura 1.6 se observa como es el proceso de captación de las aguas lluvias, donde normalmente se necesita de algún tipo de infraestructura para la recolecta de las aguas lluvias.



Figura 1.6 Captación de aguas lluvias [12]

La fuente de abastecimiento que dispondrá el sistema de bombeo para el desarrollo del trabajo de titulación es una fuente natural subterránea (pozo) tal como se observa en la figura 1.4.

1.3.4 MEDICIÓN DE NIVEL DE AGUA

Para el proceso de medición y control de nivel puede basarse en diferentes tecnologías: mecánica, capacitiva, conductiva, por presión, ultrasonidos, microondas, óptica y radar. Pero sin embargo el criterio fundamental e importante para elegir el principio de medición de nivel dependerá del estado de la sustancia, si es gaseosa, líquida o sólida. Para el desarrollo del prototipo se utilizará una tecnología no invasiva como es la ultrasónica.

1.3.4.1 Ondas ultrasónicas

Las ondas ultrasónicas no se propagan en el vacío y son ondas de igual tipo que los sonidos audibles, oscilan a frecuencias superiores a 20 KHz dentro del espectro acústico como se percibe en la figura 1.7.

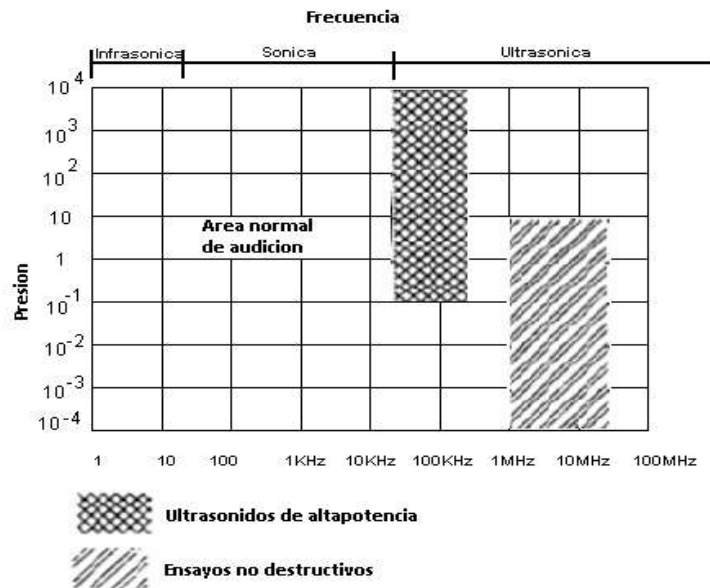


Figura 1.7 Espectro acústico [13]

1.3.4.2 Tipos de emisión sónica

Dentro de la tecnología ultrasónica se definen dos tipos de emisión de ondas sonoras o ultrasónicas que son la emisión sónica continua y la emisión sónica por impulsos.

A. Emisión sónica continúa.

En este tipo de emisión sónica se emite la onda de forma continua, donde la partícula en cada instante tiene propiedades de movimiento que cambian de forma periódica con el tiempo. Este tipo de onda se emplea con las técnicas de resonancia y transparencia [13].

En la figura 1.8 se exhibe la emisión de una onda sonora continua, además este tipo de emisión solo ayuda a localizar la existencia de algún tipo de discontinuidad dentro del material.



Figura 1.8 Onda sonora continua [13]

B. Emisión sónica por impulsos.

Como se exhibe en la figura 1.9 existe varios tipos de emisión de ondas sonoras por impulso entre las cuales se tiene el impulso largo, corto, de alta frecuencia y de baja frecuencia, mismas que se emiten a intervalos variables de tiempo.

Estos impulsos poseen una duración del orden de microsegundos, y el tiempo que transcurre entre impulsos consecutivos es del orden de milisegundos.

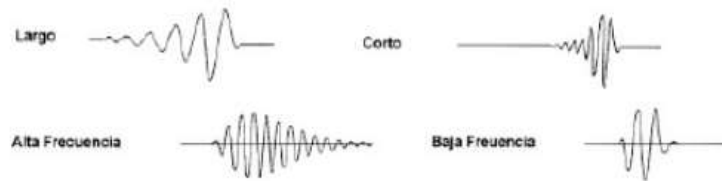


Figura 1.9 Tipos de ondas sonoras por impulsos [13]

La onda por impulso se emplea en las inspecciones por la técnica de impulso-eco. Esta técnica ayuda a definir el tiempo que tarda en ir y volver la onda, esto ayuda a revelar la existencia de cualquier discontinuidad dentro del material y también calcular la distancia a la que se encuentra desde la superficie [13].

1.3.4.3 Técnica impulso-eco para emisión y recepción de ultrasonidos

Esta técnica se fundamenta en el efecto de eco que produce el haz ultrasónico al chocar un objeto. En la figura 1.10 se contempla el funcionamiento de la técnica Impulso-Eco, donde se acopla el transmisor y receptor en el mismo dispositivo. Esta técnica tiene dos métodos que se mencionan a continuación.

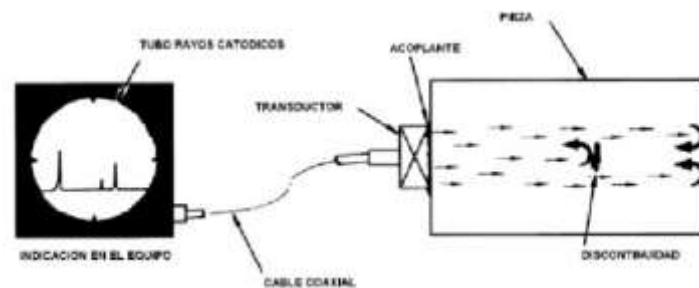


Figura 1.10 Emisión y recepción de la onda sonora [13]

A. Método de la medida del tiempo de recorrido (T).

Existen instrumentos que están diseñados para calcular el tiempo que demora la onda en recorrer (ida y vuelta) el espesor de un material dado después de la reflexión en la superficie límite, este tiempo transcurrido se puede traducir como el espesor del material.

B. Método de la medida de la intensidad acústica y tiempo de recorrido (IT).

Estos equipos envían durante intervalos de tiempo fijos, impulsos acústicos cortos, que al propagarse en la muestra y reflejarse en una discontinuidad interna, o en la

superficie límite de cada muestra, son capturados por el oscilador que los convierte en una señal eléctrica, siempre y cuando la dirección sea la correcta. Esta señal radica en una determinada indicación cuya altura es proporcional a la presión acústica de la onda reflejada [13].

1.3.4.4 Medidor ultrasónico

El funcionamiento se justifica en la emisión de pulsos ultrasónicos a la superficie del líquido y la acogida del eco en un receptor. El tiempo de demora en la captación del eco depende de la distancia a la cual está la superficie del líquido.

La continuidad de los muestreos depende del intervalo de tiempo que transcurre entre la emisión de paquetes de pulsos ultrasónicos. Este intervalo debe ser superior al tiempo que tarda en ir y volver la onda ultrasónica [14].

Este dispositivo es adecuado para la medición del nivel del fluido, puesto que es un sistema no invasivo por lo tanto no ocasiona turbulencias o movimiento del agua y además ayuda a evitar las lecturas erróneas de nivel.

Para medir la profundidad “d” de una cisterna, en la figura 1.11 se exhibe que se debe considerar que “Do” es un valor conocido al igual que la velocidad del sonido en el aire (v=334 m/s) y el único valor a variar es el tiempo de retardo que demora entre la emisión y la aceptación de la onda ultrasónica.

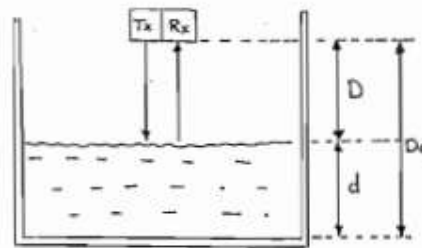


Figura 1.11 Cálculo de la profundidad “d” en cisterna [14]

$$d = D_0 - D \quad (1)$$

$$d = D_0 - \frac{v * t}{2} \quad (2)$$

1.3.5 MEDICIÓN DE CAUDAL VOLUMÉTRICO

El caudal o flujo pertenece a la porción de fluido que pasa por un lugar (canal, tubería, etc.) en un determinado tiempo, en otras palabras, concierne a un volumen de agua (litros, metros cúbicos, etc.), por unidad de tiempo (segundos, minutos, horas, etc.) [15].

Existen varias maneras de medir el caudal entre las cuales se tienen técnicas directas como indirectas para medir parámetros cinéticos en fluidos, los que miden exclusivamente velocidad (sondas de velocidad), caudal volumétrico (m^3/s) o caudal másico (Kg/s). A continuación, se mencionan los métodos para calcular el caudal volumétrico.

A. Método directo.

En este método se utiliza un recipiente y un cronómetro, donde con el cronómetro se registra el tiempo transcurrido que demora en llenar un determinado volumen de agua en el recipiente, y con la relación del volumen sobre el tiempo se logra determinar el flujo o caudal volumétrico.

B. Método indirecto.

Para el cálculo del flujo volumétrico se tienen varias tácticas apoyadas en conceptos físicos de electrónica, electromagnetismo, óptica, ultrasonido, entre otros; donde se determina la altura, la diferencia de presiones o de velocidades en varios puntos de una sección [7].

Existen varios criterios para elegir el método y el dispositivo adecuado para medir el caudal, que dependerá del parámetro de medición, del tipo y geometría de la tubería, de la naturaleza del fluido a medir (gas, líquido o mezcla de los dos, limpio o sucio, con o sin partículas disueltas, conductividad, etc.), de la precisión que se desea alcanzar, y costo.

1.3.5.1 Medidor de turbina

Este medidor dispone de aspas en el interior de la turbina acopladas al eje del rotor que al girar ocasiona revoluciones las cuales se relacionan con el caudal del fluido.

La estructura interna del medidor de turbina dispone en un rodete donde el núcleo de este contiene un imán, que, al girar el rotor, crea un campo magnético variable. El campo se detecta en una bobina eléctrica que va acoplada en la armadura externa de la unidad. La frecuencia de los impulsos magnéticos indica el caudal en un instrumento de lectura [7]. Estos medidores son de gran exactitud y son recomendables para operaciones de automatización.



Figura 1.12 Medidor de turbina

La figura 1.12 se indica las partes internas que conforman un medidor de turbina. Para procesar las señales que emiten las aspas se debe optar por convertidores de señal.

1.3.5.2 Convertidores de señal

Existen dos tipos de convertidores de señal para captar la velocidad de las turbinas, los de tipo reluctancia y los de tipo inductancia.

A. Reluctancia.

En la figura 1.13 se indica la conversión de la señal por reluctancia, para el proceso de captación de la señal de la velocidad de las turbinas, existe una bobina exterior para captar la interacción de las aspas sobre un imán exterior permanente. El paso de las aspas cambia el circuito magnético, provocando un campo eléctrico en la bobina proporcional al giro de las aspas.

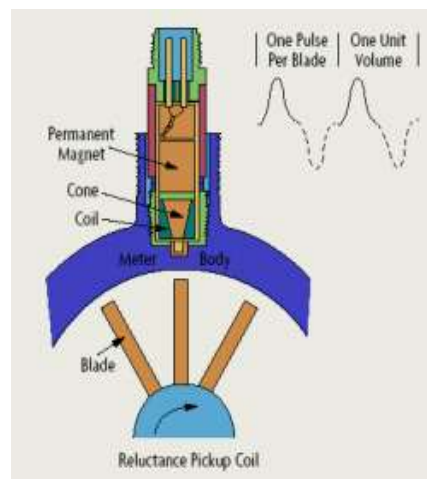


Figura 1.13 Convertidor de señal por reluctancia

B. Inductancia.

En cambio, el convertidor de señal tipo inductancia (ver figura 1.14) el rotor lleva incorporado un imán permanente y el campo magnético giratorio que se origina, induce una corriente alterna en la bobina captadora exterior.

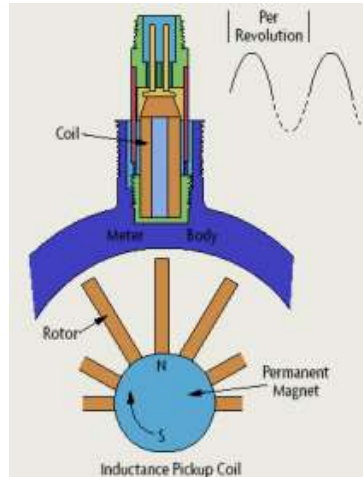


Figura 1.14 Convertidor de señal por inductancia

Con la ayuda de estos convertidores se puede calcular el caudal volumétrico del fluido que será proporcional a la frecuencia generada que es del orden de los 250 a 1200 ciclos por segundo para velocidades máximas [16].

1.3.5.3 Principio de funcionamiento del medidor de turbina

Las aspas generalmente están inclinadas a un ángulo estático con respecto al flujo incidente experimentando un torque que produce el giro del rotor [16].

En la figura 1.15 exhibe las componentes axial y transversal del vector velocidad (v), que será el foco de análisis para deducir la ecuación número 8.

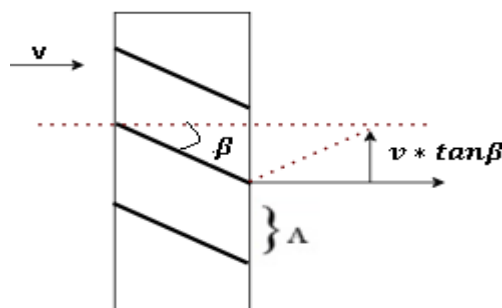


Figura 1.15 Descomposición del vector velocidad

Por tanto, la velocidad tangencial del rotor es:

$$r * w = v * \tan\beta \quad (3)$$

Donde:

r = Es el radio medio de las aspas.

ω = Es la velocidad angular del rotor.

v = Es la velocidad del fluido.

Y el caudal volumétrico Q es:

$$Q = V * A \quad (4)$$

Donde:

V = Es la velocidad del fluido.

A = Es el área transversal de la tubería.

Relacionando las ecuaciones (3) y (4) se obtiene:

$$\frac{r * w}{Q} = \frac{v * \tan\beta}{V * A} \quad (5)$$

Simplificando la ecuación (5) se logra la correspondencia entre la velocidad angular y el caudal.

$$\frac{w}{Q} = \frac{\tan\beta}{r * A} \quad (6)$$

La expresión $\frac{\tan\beta}{r * A}$ son parametros proporcionado por el fabricante y representan las características de la geometría del rotor.

El factor K ya puede darse expresado en Pulsos/Volumen, donde entra en juego el número de aspas que forman el rotor.

$$w = 2 * \pi * f = n * f \quad (7)$$

Donde:

n : Es la cantidad de aspas del rotor y es igual con el número de pulso que emite el sistema por vuelta.

f : Es la frecuencia con la que gira las paletas.

Entonces ω queda expresada en pulsos/tiempo [16].

Por tanto, el factor K es:

$$K = \frac{f \left[\frac{\text{Pulsos/seg}}{\text{Litros/seg}} \right]}{Q} = \frac{60 * f \left[\frac{\text{Pulsos/seg}}{\text{Galones/min}} \right]}{Q} \quad (8)$$

1.3.6 TECNOLOGÍAS INALÁMBRICAS

La transferencia de datos en un sistema de comunicación inalámbrica es por medio de ondas electromagnéticas que envían las señales a largas distancias a través del aire hasta arribar a su receptor.

Las comunicaciones inalámbricas han tenido un rápido crecimiento y evolución, y que van de la mano con la proliferación de laptops, smartphome, tablets, etc. Pero sin embargo existen desafíos por resolver con respecto al soporte de nuevas aplicaciones.

Las comunicaciones celulares han presentado un crecimiento exponencial, cada vez implementando nuevas funciones y aplicaciones que son útiles para el desarrollo tecnológico de la sociedad.

Dentro de las comunicaciones inalámbricas se encuentran las redes WPAN (*Personal Area Network*, Red de Área Personal), y dentro de este grupo se hallan las tecnologías UWB (*Ultra Wide-Band*, Ultra Banda Ancha), Bluetooth, Zigbee e infrarrojo que permiten el intercambio de información mediante la interconexión con otros dispositivos a corto alcance. A continuación, se describe la tecnología Bluetooth y la tecnología celular que son utilizadas para la transmisión y recepción de información del prototipo.

1.3.6.1 Tecnología bluetooth

Esta tecnología esta estandarizado en la IEEE 802.15.1 (*Institute of Electrical and Electronics Engineers*, Instituto de Ingenieros Eléctricos y Electrónicos), trabaja en una banda de frecuencia libre llamada banda ISM (*Industrial, Scientific and Medical*; Industrial, Científica y Medica) a 2.4 GHz, la cual no necesita de licencia para poder operar con dispositivos de corto alcance.

A. Características.

Las características más representativas de la tecnología inalámbrica Bluetooth se exhiben en la tabla 1.1 donde se describe el parámetro junto con su valor característico; el alcance dependerá de la potencia de transmisión como por ejemplo para 1 [mW] se tiene un rango de hasta 10 [m] y 100 [mW] para un rango de hasta 100 [m].

Tabla 1.1 Características generales de la tecnología Bluetooth [17]

Parámetro	Características
Bluetooth	IEEE 802.15.1
Banda de frecuencia	2.4 [GHz]
Velocidad de la señal máxima	1 [Mbps]
Alcance	10 [m]
Potencia de tx nominal	0-10 [dBm]
Ancho de banda del canal	1 [MHz]
Tipo de modulación	GFSK
Spreading	FHSS
Celda básica	Piconet
Extensión de la celda básica	Scatternet
Numero de nodos celulares	8
Protección de datos	CRC de 16 bits

B. Topologías de conectividad.

Básicamente existen dos tipos de topologías de conectividad de la tecnología bluetooth, mismas que se mencionan a continuación.

- **Piconets:** Es una WPAN conformada por varios dispositivos Bluetooth donde uno funciona como maestro en la piconet y el resto como esclavos. Todos los dispositivos que participan en las comunicaciones en una piconet determinada se sincronizan utilizando el reloj del maestro. Los esclavos se comunican solo con su maestro de manera punto a punto bajo el control del maestro. Las transmisiones del maestro pueden ser punto a punto o punto-multipunto. Un dispositivo esclavo además de estar en un modo activo también puede estar en los modos parado o espera para reducir los consumos de energía [18].

Un dispositivo maestro puede comunicarse con 256 dispositivos, pero solo 7 se comunican simultáneamente. En la figura 1.16 se exhibe la configuración de una piconet con un maestro y tres esclavos junto con otros dispositivos dentro del área de cobertura.

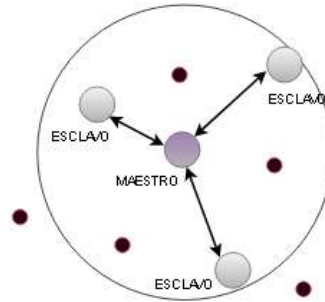


Figura 1.16 Formación de una Piconet [19]

- **Scatternet:** Es un conjunto de piconets sobrelapadas en tiempo y espacio. Se pueden conectar dos piconets para formar un scatternet como se observa en la figura 1.17. Un dispositivo Bluetooth puede participar en varias piconets al mismo tiempo, permitiendo así la posibilidad de que la información pueda fluir más allá del área de cobertura de la piconet individual. Un dispositivo en un scatternet podría ser un esclavo en varias piconets, pero maestro en solo uno de ellos [18].

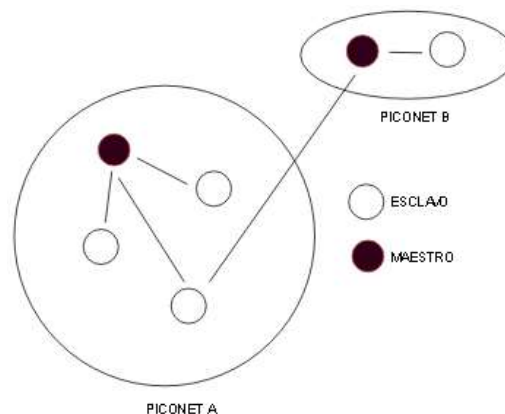


Figura 1.17 Formación de una scatternet [19]

C. Estados de un dispositivo Bluetooth.

Esta tecnología dispone de varios estados como se observa en la figura 1.18, estos estados ayudan a optimizar el gasto de energía, logrando así la durabilidad de la batería de dispositivos Bluetooth.

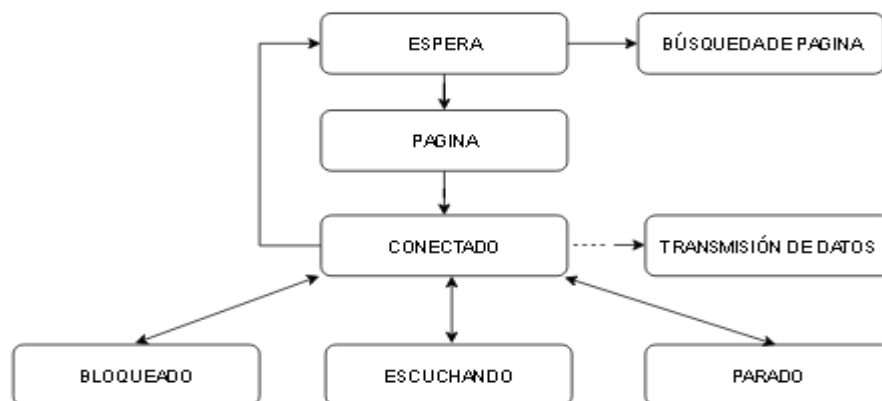


Figura 1.18 Estados de un dispositivo Bluetooth [19]

- **Espera:** Este estado es cuando los dispositivos de la piconet no están conectados.
- **Página/Búsqueda de página:** Para realizar una conexión entre dos dispositivos, en caso de conocer detalles del dispositivo remoto solo será necesario la página, caso contrario se debe realizar una búsqueda de página.
- **Conectado:** Es el estado donde se realiza la transmisión de datos.
- **Bloqueado:** Estado donde el dispositivo se mantiene inactivo durante un lapso, sin importar que llegue o no información.
- **Escuchando:** Estado en el cual el esclavo y el maestro se ponen de acuerdo para transmitir y recibir en slots de tiempo determinados.
- **Parado:** Estado en el cual el dispositivo no abandona por completo la piconet, manteniéndose en un estado de actividad mínimo receptando eventos.

D. Evolución de Bluetooth.

El avance de nuevas tecnologías en el mercado ha obligado implementar mejoras en Bluetooth, creando mejoras en las versiones desde el año 1998 en la que se publicó la primera versión y ajustándose con la demanda de velocidad de transmisión, seguridad y optimización de potencia.

La evolución de la tecnología Bluetooth se clasifica en dos tipos y varias versiones que al pasar del tiempo se ha ido cada vez mejorando. La primer tipo es conocida

como BR/EDR (*Basic Rate/ Enhanced Data Rate*, Velocidad Básica/ Velocidad de Datos Mejorada) adoptada en la versión 2.0 / 2.1 y el segundo tipo que se denomina LE (*Low Energy*, Energía Baja) acogida en la versión 4.0 / 4.1 / 4.2 / 5.0 [20].

Para la comunicación inalámbrica de corto alcance entre el prototipo y la aplicación móvil desarrollada del smartphone se utilizará un módulo Bluetooth que viene embebido la versión v2.0+EDR, por lo consiguiente se procede a la descripción de esta versión, y un resumen del resto.

Bluetooth v2.0 + EDR es una extensión de la versión 1.2 por lo tanto son compatibles, fue lanzada en el año 2004.

En esta versión, BR es obligatorio y corresponde a una velocidad de trasmisión de 1 Mbps, mientras que EDR es opcional, que utiliza modulación PSK (*Phase Shift Keying*, Modulación por Desplazamiento de Fase) y tiene dos variantes: $\pi/4$ DQPSK (*Differential Quaternary Phase Shift Keying*, Modulación por Desplazamiento de Fase Diferencial en Cuadratura) y 8 DPSK (*Differential Phase Shift Keying*, Modulación por Desplazamiento Diferencial de Fase), mismas que permiten conseguir, respectivamente tasas de 2.1 Mbps y 3 Mbps; y ayuda a optimizar el gasto de potencia ya que acorta un ciclo de trabajo [21] [22].

En la tabla 1.2 se exhibe un resumen de algunas propiedades de las versiones de Bluetooth.

Tabla 1.2 Evolución de la tecnología Bluetooth [21] [22] [23] [24] [25]

Versión	Adoptada	Velocidad de transmisión	Alcance (m)
v1.0 y v1.0B	1994 - 1995	721 Kbps	10
v1.1	2002	721 Kbps	10
v1.2	2003	721 Kbps	10
v2.0 + EDR	2004	2.1 Mbps	10
v2.1 + EDR	2007	3 Mbps	10
v3.0 + HS	2009	24 Mbps	10
v4.0 <i>Low Energy</i>	2010	1 Mbps	50
v5	2016 - 2017	2 Mbps	40
v5.1	2019	En desarrollo	En desarrollo

1.3.6.2 Tecnología celular

La telefonía celular es un sistema de comunicación totalmente inalámbrica [26] y ha tenido un avance tecnológico de manera exponencial brindando servicios tales como datos, audio y video [27] demandado cada vez más ancho de banda para las aplicaciones que se van

presentando, aunque a principios fue desarrollado solo para llamadas de voz a larga distancia debido a las limitaciones tecnológicas de la época.

Desde la aparición de la tecnología celular (ver figura 1.19) se han ido mejorando cada vez más la velocidad de transmisión e implementando más servicios.

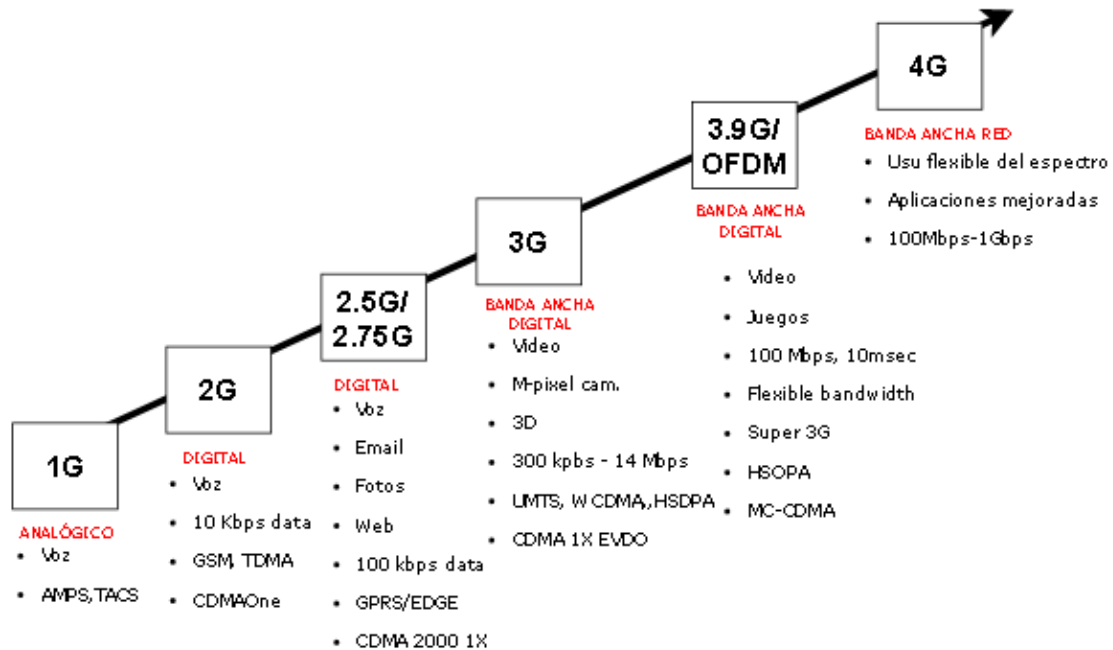


Figura 1.19 Evolución de la tecnología celular

En la actualidad se está desarrollando las redes 5G que proponen sumar distintas tecnologías inalámbricas como Wi-Fi (*Wireless Fidelity*, Fidelidad Inalámbrica), redes celulares, sistemas de corto alcance con WSN (*Wireless Sensor Network*, Redes de Sensores Inalámbricos), comunicaciones M2M (*Machine-to-Machine*, Máquina a Máquina) y generando el aclamado IoT (*Internet of Things*, Internet de Las Cosas) [28].

5G está evolucionado sobre la base de 4G LTE (*Long Term Evolution*, Evolucion a Largo Plazo), su estandarización empezó en el año 2017 y se espera que esté lista para el año 2020, 5G ofrecerá escalabilidad (capacidad para soportar más tráfico y millones de dispositivos interconectados), versatilidad (disponibilidad, movilidad, calidad de servicio) y eficiencia (menos consumo de energía, recursos compartidos y reducción de costos) [29].

Para la comunicación con el prototipo por medio de mensajes de texto se utilizará un módulo GSM/GPRS (Global System for Mobile/ General Packet Radio Service, Sistema Global para Comunicaciones Móviles/ Servicio General de Paquetes vía Radio) para la conexión inalámbrica, para lo cual se presenta un resumen de las tecnologías GSM Y GPRS que son las que viene embebido en el módulo a utilizar.

A. Estándar GSM.

El estándar GSM pertenece a la generación 2G, se inicia en el año de 1990 y marca el cambio en la utilización de una red analógica a digital, proporcionando protocolos de codificación más sofisticados con diferentes niveles de encriptación y que toleran velocidades de información más altas para voz, pero limitados en comunicación de datos, además ofrece servicios auxiliares como datos, fax y SMS (*Short Message Service*, Servicio de Mensaje Corto).

En 1991 se instala la primera red en Europa y posteriormente expandiéndose a los continentes de Asia, África y América abarcando una población del 71% de usuarios de telefonía móvil digital del mundo, además GSM también se deriva de TDMA (*Time Division Multiple Access*, Acceso Múltiple por División de Tiempo).

Entre las características que dispone el estándar GSM se presenta algunas de ella en la tabla 1.3.

Tabla 1.3 Propiedades del estándar GSM [30] [31]

Estándar	GSM
Tecnología	Digital, libre y abierta.
Bandas de frecuencia de operación	850, 900, 1800 y 1900 MHz
Cobertura	A nivel mundial
Flexibilidad	Utiliza SIM (<i>Subscriber Identity Module</i> , Módulo de Identificador de Abonado)
Calidad de voz	Mejorada
Soporte de roaming internacional	Si
Compartición de canal con otros usuarios	Si
Espaciado de portador	200 MHz
Método de acceso	TDMA/FDMA
Modulación	GMSK (<i>Gaussian Minimum Shift Keying</i> , Modulación Desplazamiento Mínimo Gaussiano)

B. Estándar GPRS.

El estándar GPRS pertenece a la generación 2.5G siendo más rápida y económica; y es conocida como una tecnología intermedia antes de migrar hacia 3G [32].

GPRS es una ramificación mejorada de GSM, eleva la velocidad de transmisión desde 56 Kbps hasta 115 Kbps o incluso más alto, esta tecnología es mediante conmutación de paquetes, donde no se utiliza los recursos de forma permanente sino más bien solo cuando se transmite, lo que hace perfecto para aplicaciones que no sean en tiempo real (correo electrónico o navegación por Internet) y además

introduce QoS (*Quality of Service*, Calidad De Servicio). La red GPRS es más costoso ya que necesita nuevos componentes y no solo de un software como en HSCSD (*High-Speed Circuit-Switched Data*, Datos por Conmutación de Circuitos de Alta Velocidad) [33].

2. METODOLOGÍA

El presente proyecto es un diseño e implementación de un prototipo de extracción de agua de forma automatizada que estará controlado y monitoreado por una aplicación móvil desarrollada y/o mensajes de texto enviados desde un teléfono celular.

En este capítulo se especifican 4 procesos fundamentales para la construcción del prototipo entre los cuales se mencionan el diseño del sistema, selección de hardware, programación del sistema y la implementación del prototipo.

En la fase de diseño del sistema corresponde al estudio de las características físicas del sistema y de los requerimientos técnicos del mismo para el dimensionamiento de la bomba.

En la fase de selección de hardware concierne la elección de los dispositivos electrónicos, accesorios y de la bomba para el sistema hidráulico.

En el proceso de programación del sistema pertenece al estudio de las herramientas de software, programación del microcontrolador ATMEL y al diseño de la aplicación móvil.

En la fase de implementación del sistema corresponde a la puesta en marcha del prototipo en campo o ambiente real.

2.1 DISEÑO DEL PROTOTIPO

En este aparato se define las propiedades físicas del sistema de bombeo de agua tales como infraestructura del sitio, características del pozo y del tanque de almacenamiento, también se analizará los requerimientos técnicos necesarios para garantizar un caudal y nivel de agua apropiado para brindar un servicio eficiente de disponibilidad de agua para un promedio de 10 personas.

Además, se realizará un diagrama hidráulico del sistema para los correspondientes cálculos del dimensionamiento de la bomba de agua, que estará en función de los

parámetros: HB (cabeza dinámica de la bomba) y NPSH (altura total de presión de retención).

2.1.1 CARACTERÍSTICAS FÍSICAS DEL SISTEMA DE BOMBEO

A continuación, se describen las propiedades físicas del pozo y del tanque de almacenamiento, así como las capacidades máximas de volumen o niveles de ambos.

2.1.1.1 Características físicas del pozo de agua

En la figura 2.1 se exhibe la estructura física y las dimensiones del pozo, el cual dispone de tres tipos de materiales de construcción y de dos diámetros (\varnothing) tanto en la parte superior como en la parte inferior del pozo.

Además, en la parte inferior del pozo se dispone de dos escalones construidos de cangahua, los cuales sirven para el ingreso al pozo para cuestiones de mantenimiento y limpieza.

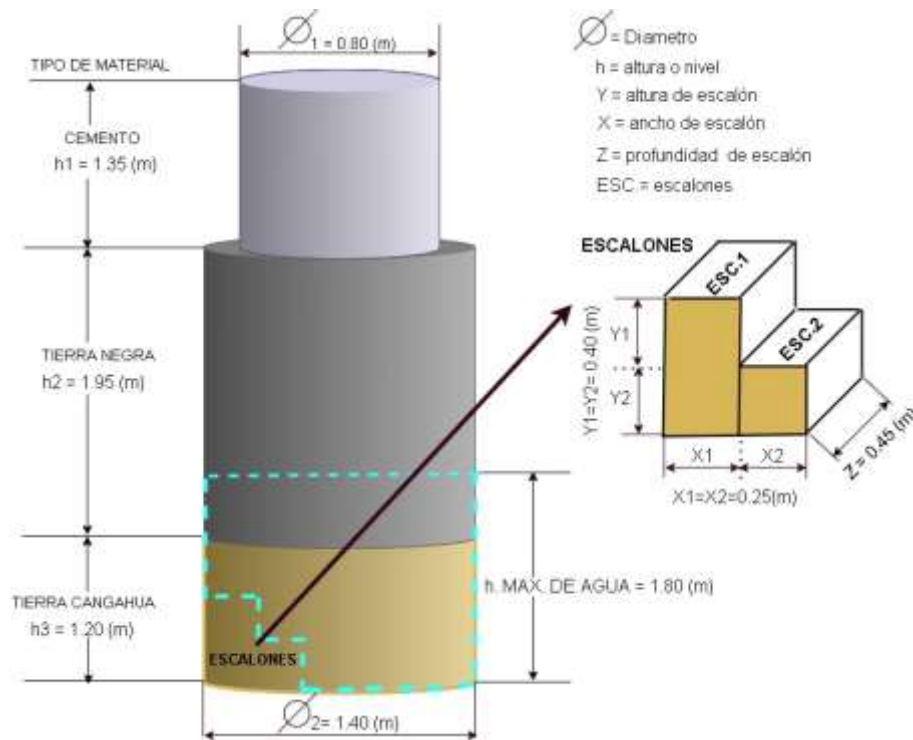


Figura 2.1 Características físicas del pozo

En la figura 2.1 se muestra una altura de 1.80 (m), que es la altura máxima de llenado de agua del pozo para la estación de invierno. En la tabla 2.1 se exhiben valores tomados con un flexómetro de manera manual, tanto para la estación verano como para la estación invierno.

Tabla 2.1 Medidas de nivel o altura de agua en el pozo.

Numero de muestra	Altura de agua estación de verano (hv) [m]	Altura de agua estación de invierno(hi) [m]
Día 1	hv1= 1.12	hi1= 1.78
Día 2	hv2= 1.30	hi2= 1.68
Día 3	hv3= 1.24	hi3= 1.40
Día 4	hv4= 1.30	hi4= 1.79
Día 5	hv5= 1.25	hi5= 1.80
Día 6	hv6= 1.29	hi6= 1.50
Día 7	hv7= 1.15	hi7= 1.80
Día 8	hv8= 1.27	hi8= 1.55
Día 9	hv9= 1.12	hi9= 1.80
Día 10	hv10= 1.01	hi10= 1.67

A continuación, se realizar el cálculo del volumen máximo de agua dentro del pozo el cual está descrita por la ecuación 9. Este volumen está en función de una altura de 1.80 [m] como se menciona en la tabla 2.1, y es la altura máxima para la estación de invierno donde existe más afluencia de líquido a diferencia de lo que sucede en la estación de verano donde la altura máxima es de 1.30 [m].

$$V. \text{ max. agua1[m}^3] = V. \text{ cilindro[m}^3] - V. \text{ escalones[m}^3] \quad (9)$$

Donde:

V. max. agua1[m³]: Es el volumen máximo de agua en la estación de invierno.

V. cilindro[m³]: Es el volumen de agua real del pozo.

V. escalones[m³]: Es el volumen de los escalones dentro del pozo.

Entonces para el cálculo del volumen del cilindro esta descrita por la siguiente ecuación número 10.

$$V. \text{ cilindro[m}^3] = \pi * \left(\frac{\emptyset[\text{m}]}{2}\right)^2 * h[\text{m}] \quad (10)$$

Donde:

\emptyset [m]: Es el diámetro de la parte inferior del pozo.

h [m]: Es la altura o nivel de agua.

$$V. \text{ cilindro} = \pi * \left(\frac{1.4[\text{m}]}{2}\right)^2 * 1.8[\text{m}]$$

$$V. \text{ cilindro} = 2.77\text{m}^3$$

Y el volumen de los escalones dentro del pozo se calcula por separado tal como está indicado en la figura 2.2.

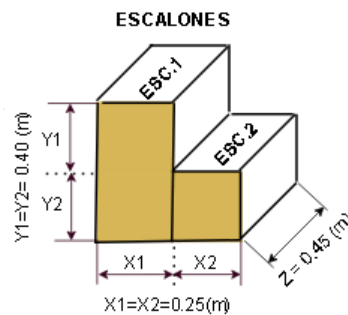


Figura 2.2 Escalones dentro del pozo

Como se observa en la figura 2.2, se presentan dos escalones que se calcula a través de la ecuación número 11.

$$V. \text{escalones} [\text{m}^3] = V. \text{esc. 1} [\text{m}^3] + V. \text{esc. 2} [\text{m}^3] \quad (11)$$

Donde:

V. escalones $[\text{m}^3]$: Es el volumen total de los escalones.

V. esc. 1 $[\text{m}^3]$: Es el volumen del escalón 1.

V. esc. 2 $[\text{m}^3]$: Es el volumen del escalón 2.

Para el cálculo del volumen del primer escalón dentro del pozo se calcula por medio de la ecuación 12, misma que describe el volumen de un paralelepípedo rectangular.

$$V. \text{paralelepípedo} [\text{m}^3]: \text{alto} [\text{m}] * \text{ancho} [\text{m}] * \text{profundidad} [\text{m}] \quad (12)$$

Entonces de acuerdo a la figura 2.2 el volumen del escalón 1 sería igual a:

$$V. \text{esc. 1} [\text{m}^3] = (Y1 + Y1) [\text{m}] * X1 [\text{m}] * Z [\text{m}]$$

$$V_{\text{esc. 1}} = (0.40 + 0.40) [\text{m}] * 0.25 [\text{m}] * 0.45 [\text{m}] = 0.09 [\text{m}^3]$$

Y de igual manera se procede a calcular el volumen del escalón 2 por medio de la ecuación número 12.

$$V. \text{esc. 2} [\text{m}^3] = (Y2) [\text{m}] * X2 [\text{m}] * Z [\text{m}]$$

$$V_{\text{esc. 2}} = 0.40 [\text{m}] * 0.25 [\text{m}] * 0.45 [\text{m}] = 0.045 [\text{m}^3]$$

Y el volumen total de los escalones es:

$$V_{\text{escalones}} = 0.09 \text{ [m}^3\text{]} + 0.045 \text{ [m}^3\text{]} = 0.135 \text{ [m}^3\text{]}$$

Por lo tanto, el volumen máximo de agua que existe dentro del pozo para la estación de invierno donde en nivel o altura máximo de agua es 1.8 (m) es:

$$V_{\text{max. agua1}} \text{ [m}^3\text{]} = 2.77 \text{ [m}^3\text{]} - 0.135 \text{ [m}^3\text{]} = 2.64 \text{ [m}^3\text{]}$$

Para el cálculo del volumen máximo en la estación climática de verano ($V_{\text{max. agua2}}$) se realiza cálculos idénticos al cálculo del $V_{\text{max. agua1}}$, donde se considera una altura de 1.3 (m) como nivel máximo de agua en dicha estación climática.

$$V_{\text{max. agua2}} \text{ [m}^3\text{]} = 2.00 \text{ [m}^3\text{]} - 0.135 \text{ [m}^3\text{]} = 1.87 \text{ [m}^3\text{]}$$

Y finalmente se tiene el cálculo del volumen mínimo de agua dentro del pozo considerando la altura mínima de $h_{v10}=1.01$ [m] según la tala 2.1, este volumen servirá para tener una referencia de cantidad de agua mínima que se puede extraer del pozo para el consumo y uso diario del mismo.

$$V_{\text{minimo agua}} \text{ [m}^3\text{]} = 1.55 \text{ [m}^3\text{]} - 0.135 \text{ [m}^3\text{]} = 1.42 \text{ [m}^3\text{]}$$

En la tabla 2.2 se exhibe las propiedades más importantes del pozo de agua.

Tabla 2.2 Características del pozo de agua.

Característica	Medida
Ubicación del pozo agua	0°16'23.3"S 78°31'08.4"W
Profundidad del pozo	$h_{\text{pozo}} = 7$ [m]
Diámetro del pozo	$\varnothing 1 = 0.8$ [m] $\varnothing 2 = 1.4$ [m]
Volumen máximo de agua en invierno	$V_{\text{max. Agua1}} = 2.64$ [m ³]
Volumen máximo de agua en verano	$V_{\text{max. Agua2}} = 1.87$ [m ³]
Volumen mínimo de agua	$V_{\text{minimo Agua}} = 1.42$ [m ³]

2.1.1.2 Características físicas del tanque de almacenamiento

En la figura 2.3 se especifica las medidas del tanque de almacenamiento, el cual está construido de concreto con fines de evitar la absorción del agua en las paredes.

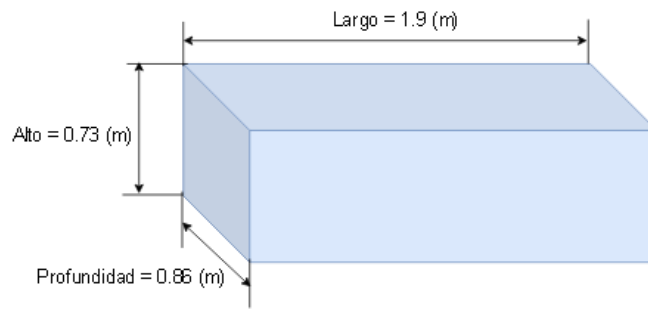


Figura 2.3 Características físicas del tanque de almacenamiento

El volumen máximo de agua del tanque de almacenamiento ($V_{\text{max.tanq}} [m^3]$) esta especificada por la ecuación número 12 que describe el volumen de un paralelepípedo rectangular.

$$V_{\text{max.tanq}} [m^3]: \text{alto}[m] * \text{largo}[m] * \text{profundidad} [m]$$

$$V_{\text{max.tanq}} [m^3] = 0.73[m] * 1.9[m] * 0.86[m]$$

$$V_{\text{max.tanq}} = 1.19 m^3$$

En la tabla 2.3 se exhibe de las características físicas del tanque de almacenamiento.

Tabla 2.3 Características del tanque de almacenamiento.

Característica	Medida
Material de construcción del tanque de almacenamiento	Concreto
Dimensiones del tanque de almacenamiento	Largo = 1.9 [m] Alto = 0.73 [m] Profundidad = 0.86 [m]
Volumen máximo de agua del tanque de almacenamiento	$V_{\text{max.tanq}} = 1.19 [m^3]$

2.1.2 REQUERIMIENTOS TÉCNICOS DEL SISTEMA DE BOMBEO

Para obtener un sistema de extracción de agua eficiente y económico, se analizó varios requerimientos técnicos que se describen a continuación, entre ellos el volumen o nivel mínimo de agua en el pozo, volumen o nivel máximo de agua de llenado en el tanque de almacenamiento y el caudal mínimo en la línea de conducción (tubería) que servirá para proteger la bomba de agua de posibles lecturas erróneas por parte de los sensores.

2.1.2.1 Requerimientos técnicos en la parte de succión

El nivel o volumen mínimo de agua ($V_{\text{min. agua. pozo}} [m^3]$) que requiere tener el pozo para evitar la succión de la bomba en vacío esta descrita por la ecuación número 9, donde el

volumen de cilindro ($V_{\text{cilindro}}[\text{m}^3]$) está considerado una altura (h [m]) de 0.15 [m] y el volumen del escalón ($V_{\text{escalones}}[\text{m}^3]$) está en función de las siguientes medidas alto 0.15 [m], ancho 0.45 [m] y largo 0.25 [m].

$$V_{\text{min. agua. pozo}}[\text{m}^3] = V_{\text{cilindro}}[\text{m}^3] - V_{\text{escalones}}[\text{m}^3]$$

$$V_{\text{min. agua. pozo}}[\text{m}^3] = \left(\pi * \left(\frac{\phi[\text{m}]}{2} \right)^2 * h[\text{m}] \right) - (\text{alto}[\text{m}] * \text{ancho}[\text{m}] * \text{profundidad} [\text{m}])$$

$$V_{\text{min. agua. pozo}} = \left(\pi * \left(\frac{1.4[\text{m}]}{2} \right)^2 * 0.15[\text{m}] \right) - (0.15[\text{m}] * 0.45[\text{m}] * 0.25[\text{m}])$$

$$V_{\text{min. agua. pozo}} = 0.231 \text{ m}^3 - 0.0168 \text{ m}^3$$

$$V_{\text{min. agua. pozo}} = 0.214 \text{ [m}^3\text{]}$$

El nivel de 0.15 [m] es la altura mínima que se requiere de agua para evitar la succión de la bomba en vacío, esta altura se indica en la figura 2.4 junto con las medidas de la tubería PVC.

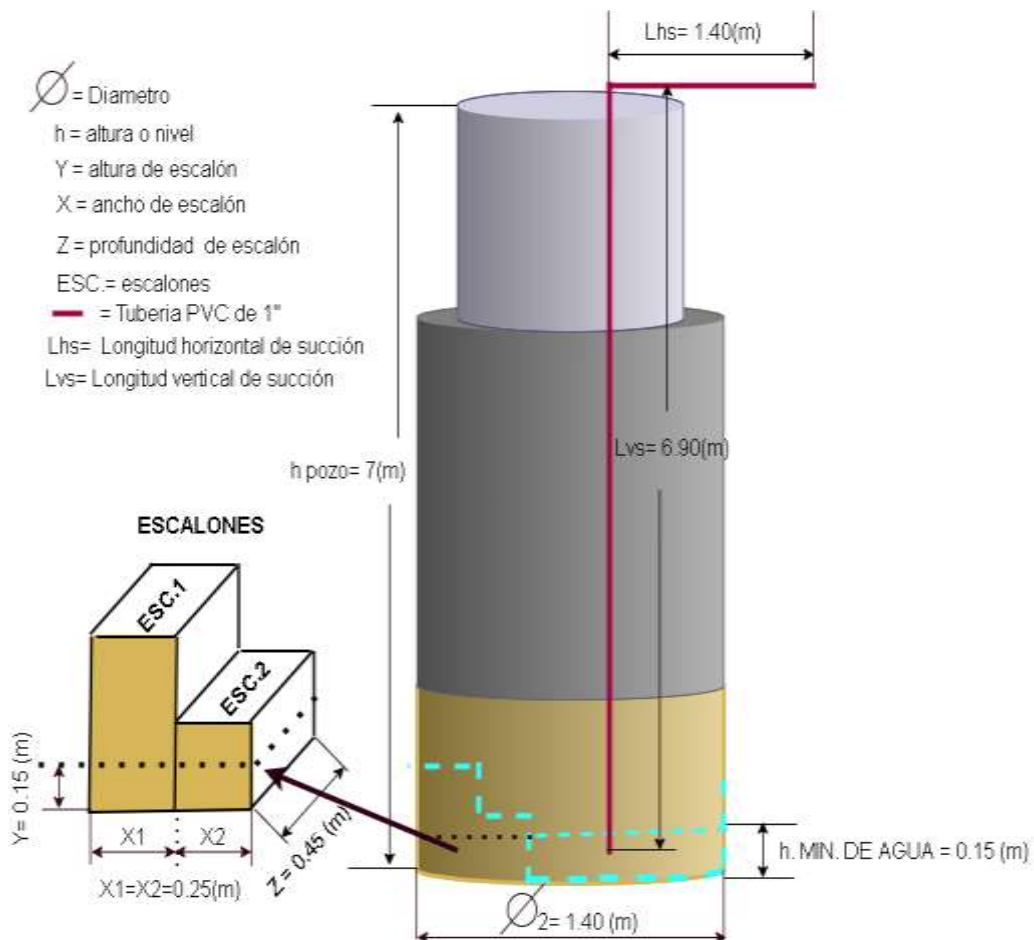


Figura 2.4 Nivel mínimo de agua dentro del pozo

Este $V_{\text{min. agua. pozo}} [\text{m}^3]$ corresponde al 8,12 % de la capacidad máxima que dispone el pozo de agua. Por lo tanto, se necesita una tubería de longitud 6.90 [m] en la parte de la succión dejando introducida una porción de tubería de 0.05 (m) por debajo del nivel de agua mínimo para evitar la succión de la bomba en vacío como se indica en la figura 2.4.

De acuerdo con el entorno donde se implementará el prototipo será necesario la utilidad de tubería de 1" y varios accesorios como: válvula de pie, codo de 90°, unión universal, unión de rosca, entre otros.

Además, se considera que el promedio de usuarios que harán uso del líquido vital (agua) son de 10 personas, y teniendo en cuenta que según la OMS estipula que para garantizar una buena salud se debe tener un consumo mínimo de agua de 100 litros/día por habitante se llega a obtener el consumo mínimo de agua por día para las 10 personas arrojando un resultado de 1000 litros/día como consumo mínimo de agua total.

En la tabla 2.4 se sintetiza los requerimientos técnicos en la parte de la succión del sistema hidráulico.

Tabla 2.4 Requerimientos técnicos en la parte de succión.

Parámetro	Medida
Volumen mínimo de agua en el pozo para evitar succión de la bomba en vacío	$V_{\text{min. agua. pozo}} = 0.2142 [\text{m}^3]$ 8.12%
Número de usuarios consumidores	10
Consumo de agua por día total	1000 litros = $1 [\text{m}^3]$
Diámetro interno de tubería de succión	$\varnothing_{\text{int. suc.}} = 1''$
Longitud vertical de succión	$L_{\text{vs}} = 6.90 [\text{m}]$
Longitud horizontal de succión	$L_{\text{hs}} = 1.40 [\text{m}]$
Longitud de corriente en succión	$L_{\text{s}} = L_{\text{vs}} + L_{\text{hs}} = 8.30 [\text{m}]$
1 Válvula de pie	1''
1 Regilla	1''
1 Codo de 90°	1''
1 Unión de rosca	1''
1 Unión universal	1''

2.1.2.2 Requerimientos técnicos en la parte de descarga

A continuación, se realizará el cálculo del volumen máximo de llenado del tanque de almacenamiento ($V_{\text{max. llenado. tanq}} [\text{m}^3]$), este volumen está descrito por la ecuación número 12, donde el nivel o altura máxima de agua que debe disponer el tanque de almacenamiento debe ser 0.65 [m] para evitar que el sensor alojado dentro del tanque de almacenamiento sea mojado tal como se enseña en la figura 2.5.

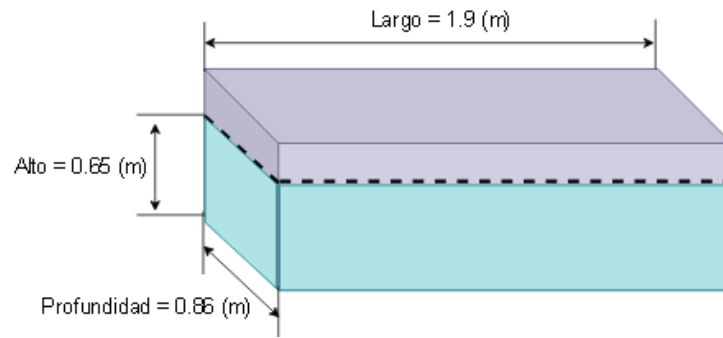


Figura 2.5 Llenado máximo en el tanque de almacenamiento

$$V_{\text{max. llenado. tanq}} [\text{m}^3]: \text{alto}[\text{m}] * \text{largo}[\text{m}] * \text{profundidad} [\text{m}]$$

$$V_{\text{max. llenado. tanq}} [\text{m}^3] = 0.65[\text{m}] * 1.9[\text{m}] * 0.86[\text{m}]$$

$$V_{\text{max. llenado. tanq}} = 1.0621 [\text{m}^3]$$

Este valor de volumen máximo de llenado en el tanque de almacenamiento corresponde al 97.26% de la capacidad total, que evitará el desbordamiento de agua y servirá para satisfacer las necesidades de limpieza e higiene de los consumidores.

Además, será necesario calcular un volumen promedio en el tanque de almacenamiento ($V_{\text{pro. tanq}} [\text{m}^3]$) que servirá para mantener una cantidad de agua suficiente antes de que sea activada la bomba eléctrica para realizar el proceso de succión de agua del pozo. Para el cálculo de este volumen se requiere un nivel o altura de 0.35[m] que este alrededor de la mitad de la capacidad máxima del tanque de almacenamiento como se exhibe en la figura 2.6.

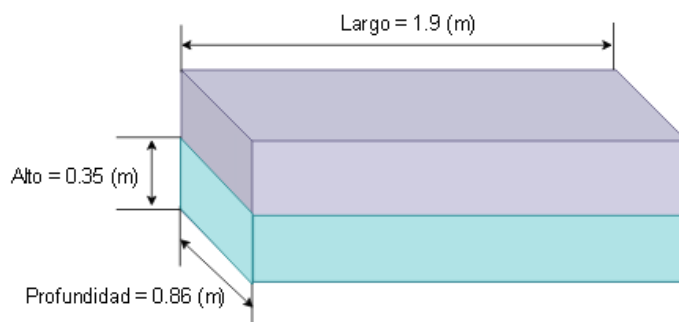


Figura 2.6 Volumen promedio en el tanque de almacenamiento

$$V_{\text{pro. tanq}} [\text{m}^3]: \text{alto}[\text{m}] * \text{largo}[\text{m}] * \text{profundidad} [\text{m}]$$

$$V_{\text{pro. tanq}} [\text{m}^3]: = 0.35 * 1.9 * 0.86$$

$$V_{\text{pro. tanq}} = 0.58[\text{m}^3] \rightarrow 48.73\%$$

Y para garantizar un caudal mínimo en la tubería del sistema, existen rangos de velocidad (V) aceptables para líquidos pocos viscosos (agua) que están definidos en la tabla 2.5.

Tabla 2.5 Velocidad de fluidos [34].

Fluido	Tipo de flujo	Velocidad (V)	
		pies/s	m/s
Líquidos poco viscosos	Flujo por gravedad	0,5-1	0,15-0,30
	Entrada de bomba	1-3	0,3-0,9
	Salida de bomba	4-10	1,2-3
	Línea de conducción	4-8	1,2-2,4
Líquidos viscosos	Entrada de bomba	0,2-0,5	0,06-0,15
	Salida de bomba	0,5-2	0,15-0,6
Vapor de agua		30-50	9-15
Aire o Gas		30-100	90-30

Por medio de la ecuación 4 del capítulo 1, se procede a efectuar el cálculo del flujo volumétrico mínimo ($Q_{\min} \left[\frac{m^3}{s} \right]$), que está en función de la velocidad mínima del fluido poco viscoso en la entrada de la bomba que es igual a $0.3 \left[\frac{m}{s} \right]$ y considerando un diámetro interno de la tubería de descarga ($\emptyset_{\text{int. des. [m]}}$) igual de $0.0127[m]$.

$$Q_{\min} \left[\frac{m^3}{s} \right] = V \left[\frac{m}{s} \right] * A [m^2]$$

$$Q_{\min} \left[\frac{m^3}{s} \right] = V \left[\frac{m}{s} \right] * \left(\pi * \left(\frac{\emptyset_{\text{int. des. [m]}}}{2} \right)^2 \right)$$

$$Q_{\min} \left[\frac{m^3}{s} \right] = 0.3 \left[\frac{m}{s} \right] * \left(\pi * \left(\frac{0.0127[m]}{2} \right)^2 \right)$$

$$Q_{\min} = 0.000038 \left[\frac{m^3}{s} \right]$$

$$Q_{\min} = 2.28 \left[\frac{L}{\text{min}} \right]$$

Este valor de caudal ($Q_{\min} = 2.28 \left[\frac{L}{\text{min}} \right]$) será considerado como el valor mínimo utilizado dentro del sistema de extracción de agua para proteger la bomba eléctrica de una posible succión en vacío y de lectura falsas por parte de los sensores ultrasónicos.

De igual manera para la sección de descarga será necesario el uso de tubería de ½" juntos con varios accesorios como: codos de 90°, unión universal, unión de rosca, válvula esférica, neplo corrido, reducción brusca, Te, entre otros.

A continuación, se sintetiza en la tabla 2.6 los requerimientos técnicos necesarios del sistema hidráulico en la sección de la descarga.

Tabla 2.6 Requerimientos técnicos en la parte descarga.

Parámetro	Medida
Volumen máximo de llenado	$V_{\max.\text{llenado.tanq}}=1.0621 \text{ [m}^3\text{]}$
Volumen promedio del tanque	$V_{\text{pro.tanq}}=0.58 \text{ [m}^3\text{]}$
Velocidad en la entrada de la bomba	$0.3 \left[\frac{\text{m}}{\text{s}} \right]$
Diámetro interno de tubería de descarga	$\phi_{\text{int.des.}} = 1/2"$
Caudal mínimo de descarga	$Q_{\min} = 2.28 \left[\frac{\text{L}}{\text{min}} \right]$
Longitud vertical de descarga	$L_{\text{vd}}= 0.73 \text{ [m]}$
Longitud horizontal de descarga	$L_{\text{hd}}= 11.22 \text{ [m]}$
Longitud de corriente en descarga	$L_{\text{d}}= L_{\text{vd}}+L_{\text{hd}}=11.95 \text{ [m]}$
1 Reducción brusca	1" a ½"
9 codos de 90°	½"
3 Unión de rosca	½"
1 Unión universal	½"
1 Válvula esférica	½"
1 neplo corrido	½"
Te	½"
Medidor de agua (flujometro)	½"

De acuerdo a los requerimientos descritos se estipula que, considerando un nivel de agua mínimo en el pozo de 1.01 [m] y manteniendo un consumo diario de 1[m³] para un total de 10 usuarios, se deduce que el sistema de extracción de agua si satisface las exigencias y necesidades de los usuarios, en vista que en el taquen se puede almacenar una cantidad de agua de 1,0621 [m³], mientras que en el pozo como mínimo brindara una cantidad de agua de 1.2 [m³].

2.1.3 DIAGRAMA HIDRÁULICO DEL SISTEMA DE BOMBEO

El sistema hidráulico que se encuentra actualmente instalado en los medios del predio consta de dos secciones o partes. En la parte de la succión consta básicamente de una tubería de una pulgada (1") de material PVC junto con varios accesorios como: válvula de pie, codo de 90° y uniones, todos estos accesorios tienen la misma medida de diámetro que la tubería PVC de 1"; y en la sección de descarga del sistema hidráulico consta de la misma manera de varios accesorios como: reducción brusca de 1" a 1/2", neplo corrido, te,

válvula esférica, caudalímetro, codos de 90°, uniones y tubería de 1/2", en esta sección las medidas de diámetros de los accesorios es el mismo que la tubería de 1/2".

En la figura 2.7 se exhibe el diseño del sistema hidráulico, junto con todos los accesorios que conforma el mismo.

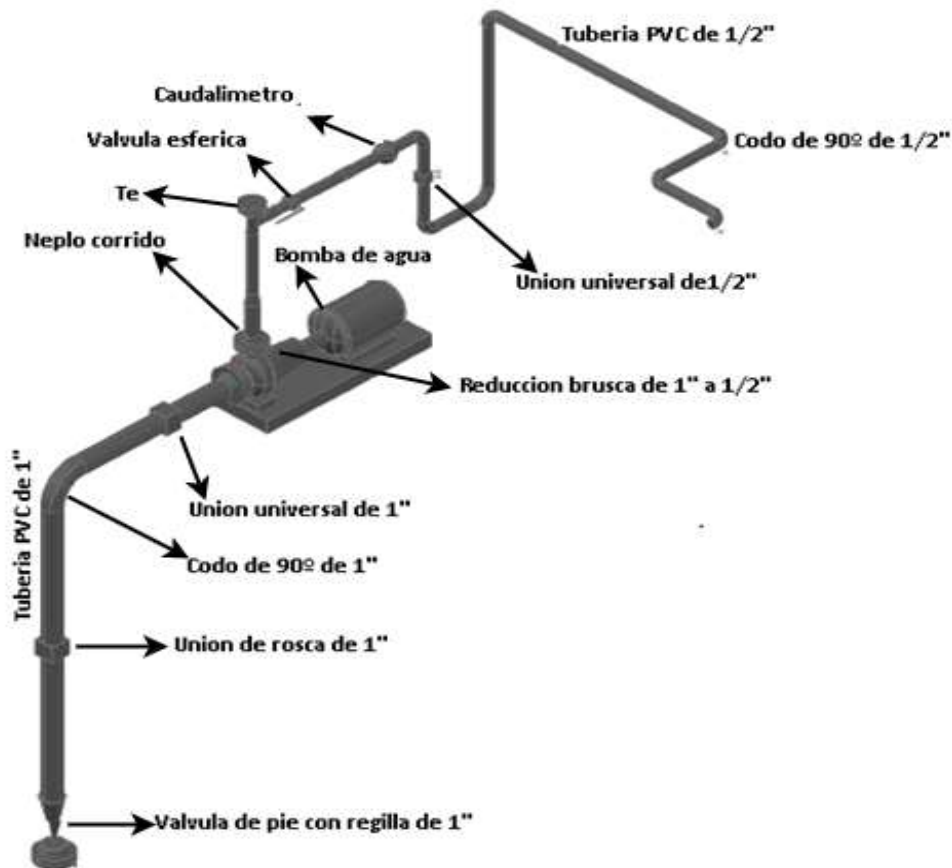


Figura 2.7 Diagrama del sistema hidráulico.

Como parte del trabajo de ingeniería en el proyecto de titulación, es ajustar los requerimientos descritos en el apartado 2.1.2 al sistema instalado actualmente, ya que para el dimensionamiento de la bomba será necesario tener en cuenta las características y requerimientos del sistema.

2.1.4 DIMENSIONAMIENTO DE LA BOMBA DE AGUA

Para el desarrollado de los cálculos del dimensionamiento de la bomba, están en función de las características físicas y requerimientos técnicos anteriormente analizados, así como de otros parámetros: propiedades del fluido, regímenes de flujo del fluido, ecuación general de energía, pérdidas de energía en la tubería, ecuación de la continuidad, cabeza dinámica de la bomba y de la altura total de presión de retención.

2.1.4.1 Propiedades de los fluidos

Los fluidos tienen propiedades físicas que permiten caracterizar y cuantificar su comportamiento. Algunas de las propiedades son exclusivas de los fluidos y otras son típicas de toda sustancia. Entre las cuales tenemos las siguientes:

A. Densidad (ρ): es masa por unidad de volumen, la densidad en los líquidos casi permanece constante [8] y se describe con la siguiente ecuación número 13.

$$\rho = \frac{m}{v} \left[\frac{\text{Kg}}{\text{m}^3} \right] \quad (13)$$

Donde:

$m[\text{Kg}]$: Es la masa.

$v[\text{m}^3]$: Es el volumen de la sustancia.

B. Viscosidad: es una medida cuantitativa de la resistencia de un fluido a fluir. Más concretamente, la viscosidad determina la velocidad de deformación del fluido cuando se le aplica un esfuerzo cortante dado [8].

La ley de viscosidad de Newton establece que, para una tasa dada de deformación angular del fluido, el esfuerzo cortante es directamente proporcional a la viscosidad y se la conoce como viscosidad absoluta o dinámica (μ) [35].

Otra manera de representar la viscosidad es calculando la viscosidad cinemática que es la relación de la viscosidad absoluta con la densidad, como se describe en la siguiente ecuación número 14.

$$\vartheta = \frac{\mu}{\rho} \left[\frac{\text{m}^2}{\text{s}} \right] \quad (14)$$

Donde:

$\vartheta \left[\frac{\text{m}^2}{\text{s}} \right]$: Es la viscosidad cinemática.

$\mu \left[\frac{\text{Kg}}{\text{m.s}} \right]$: Es la viscosidad absoluta.

$\rho \left[\frac{\text{Kg}}{\text{m}^3} \right]$: Es la densidad de la sustancia.

En la tabla 2.7 se declara los valores de densidad y viscosidad de algunos fluidos, entre ellos el agua a condiciones estándar de presión y temperatura H_2 .

Tabla 2.7 Propiedades de los fluidos a 1 atm y 20°C [8].

Fluido	$\mu \left[\frac{\text{Kg}}{\text{m} \cdot \text{s}} \right]$	Relación $\mu/\mu (\text{H}_2)$	$\rho \left[\frac{\text{Kg}}{\text{m}^3} \right]$	$\theta \left[\frac{\text{m}^2}{\text{s}} \right]$	Relación $\theta/\theta (\text{H}_2)$
Hidrógeno	8.8×10^{-6}	1.0	0.084	1.05×10^{-4}	920
Aire	1.8×10^{-5}	2.1	1.20	1.51×10^{-5}	130
Gasolina	2.9×10^{-4}	33	680	4.22×10^{-7}	3.7
Agua	1.0×10^{-3}	114	998	1.01×10^{-6}	8.7
Alcohol etílico	1.2×10^{-3}	135	789	1.52×10^{-6}	13
Mercurio	1.5×10^{-3}	170	13.580	1.16×10^{-7}	1.0
Aceite SAE 30	0.29	33.000	891	3.25×10^{-4}	2.850
Glicerina	1.5	170	1.264	1.18×10^{-3}	10.300

C. Peso específico (γ): el peso específico de un fluido es su peso por unidad de volumen, la densidad y el peso específico están relacionados por la gravedad [8] como se exhibe en la siguiente ecuación número 15.

$$\gamma = \rho * g \left[\frac{\text{N}}{\text{m}^3} \right] \quad (15)$$

Donde:

$\rho \left[\frac{\text{Kg}}{\text{m}^3} \right]$: Es la densidad de la sustancia.

$g \left[\frac{\text{m}}{\text{s}^2} \right]$: Es la aceleración de la gravedad terrestre y su valor estándar es 9.807.

Por lo tanto, si la densidad del agua es 998 $\left[\frac{\text{Kg}}{\text{m}^3} \right]$ y la gravedad igual a 9.807 $\left[\frac{\text{m}}{\text{s}^2} \right]$ entonces el peso específico (γ) del agua es igual a 9787.34 $\left[\frac{\text{N}}{\text{m}^3} \right]$ aplicando la ecuación número 15.

D. Presión (P): es la fuerza superficial que ejerce un fluido sobre las paredes del recipiente que lo contiene [34]. Existen varios tipos de presión que se explican a continuación.

- Presión atmosférica [atm]: Esta presión es debido al peso del aire en un sitio particular de la superficie terrestre, por lo que su valor depende de la ubicación geográfica. También conocida como la presión barométrica debido a que se utiliza un instrumento llamado barómetro para su medición.
- Presión absoluta: Es la presión que se mide con relación al cero absoluto de presión, donde no existe presencia del aire, o también se puede calcular como la suma algebraica de la presión atmosférica más la relativa.

- Presión relativa: Esta presión toma como referencia la presión atmosférica y se mide con un instrumento que se llama manómetro, por lo que a esta presión también se le conoce como presión manométrica.
- Vacío: Es la depresión por debajo del nivel atmosférico y se puede calcular como la diferencia entre presiones atmosférica y absoluta.

En la figura 2.8 se exhibe la relación que existe entre las diferentes presiones: la atmosférica, la absoluta, la relativa y la de vacío.

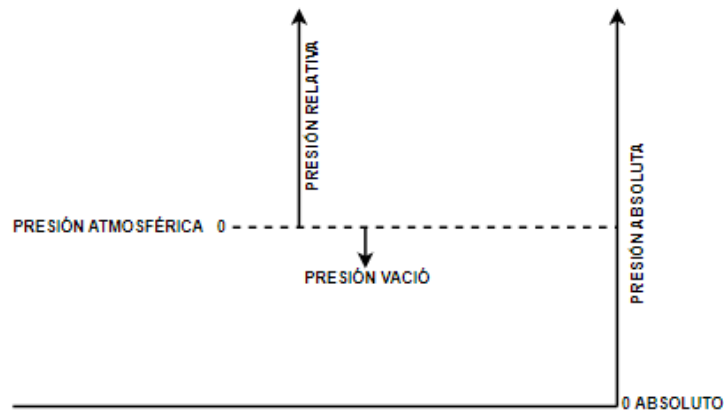


Figura 2.8 Tipos de presiones.

En la tabla 2.8 se indica las propiedades del agua para los cálculos del dimensionamiento de la bomba.

Tabla 2.8 Propiedades del agua.

Propiedad	Símbolo	Valor
Densidad	ρ	998 $\left[\frac{\text{Kg}}{\text{m}^3} \right]$
Viscosidad absoluta o dinámica	μ	$1.0 \times 10^{-3} \left[\frac{\text{Kg}}{\text{m} \cdot \text{s}} \right]$
Viscosidad cinética	θ	$1.01 \times 10^{-6} \left[\frac{\text{m}^2}{\text{s}} \right]$
Peso específico	γ	9787.34 $\left[\frac{\text{N}}{\text{m}^3} \right]$

2.1.4.2 Regímenes del flujo del fluido

El tipo de flujo está directamente relacionado con la velocidad del fluido, así se tiene que a velocidades más bajas que la crítica, este flujo se le conoce como flujo laminar, pero si la velocidad es mayor que la crítica, el régimen es turbulento.

Numero de Reynolds (NRe): Osborne Reynolds demostró que el régimen del flujo en tuberías depende del diámetro de la tubería, de la densidad, de la viscosidad y de la velocidad del flujo. El número de Reynolds es un valor adimensional donde relaciona las fuerzas dinámicas de la masa del fluido respecto a los esfuerzos de deformación ocasionados por la viscosidad [36] como se explica a continuación en la ecuación número 16.

$$NRe = \frac{\text{Fuerzas Inerciales}}{\text{Fuerzas Vizcosas}} = \frac{\rho * V * \emptyset}{\mu} \quad (16)$$

Donde:

$\rho \left[\frac{\text{Kg}}{\text{m}^3} \right]$: Es la densidad del fluido.

$V \left[\frac{\text{m}}{\text{s}} \right]$: Es la velocidad del fluido.

$\emptyset[\text{m}]$: Es el diámetro interno de la tubería.

$\mu \left[\frac{\text{Kg}}{\text{m*s}} \right]$: Es la viscosidad dinámica del fluido.

Para estudios técnicos se considera las siguientes condiciones: Si $NRe < 2000$ se considera como flujo laminar, si $2000 < NRe < 4000$ es un flujo impredecible o de transición y si $NRe > 4000$ es un flujo turbulento.

2.1.4.3 Ecuación general de energía

La ecuación general de energía aparece a partir del teorema de Bernoulli, en otras palabras, es una extensión de la ecuación de Bernoulli, lo que posibilita resolver problemas en los que hay pérdidas y ganancias de energía. En la figura 2.9 se interpreta la manera lógica la ecuación general de energía considerando pérdidas de carga.

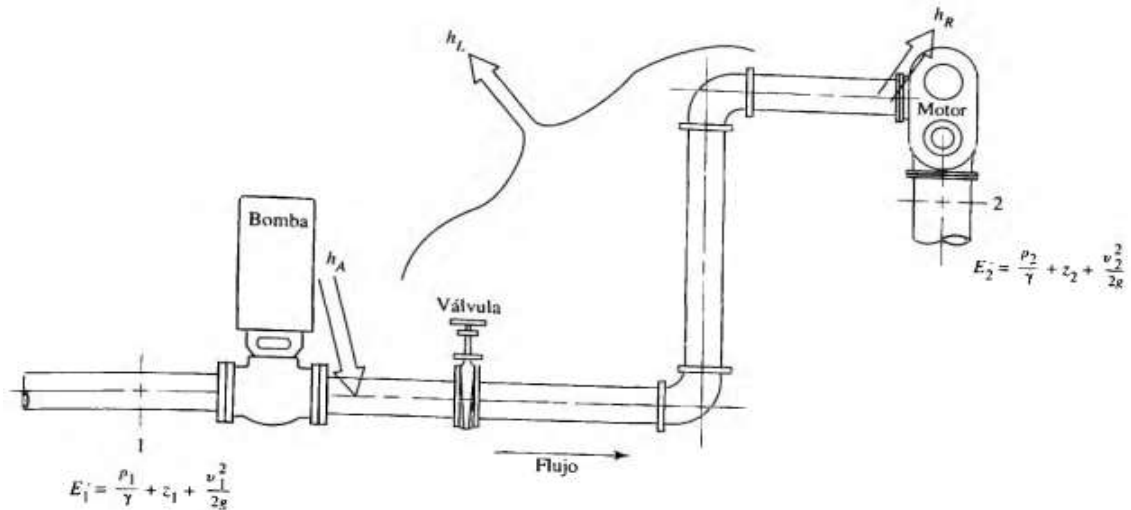


Figura 2.9 Sistema de flujo de un fluido [37]

Y a continuación se describe de manera matemática la ecuación general de energía descrita por la expresión número 17.

$$E_1 + h_A - h_R - h_L = E_2 \quad (17)$$

Donde:

E1: Energía en la etapa de succión.

E2: Energía en la etapa de descarga.

hA: Es la energía que se añade al fluido con un instrumento mecánico, como una bomba.

hR: Es la energía que se remueve del fluido por medio de un instrumento mecánico como un motor.

hL: Es las pérdidas de energía del sistema por la fricción en tuberías, o pérdidas menores por válvulas u otros accesorios.

Por lo tanto, reemplazando valores en la ecuación número 17 se obtiene la ecuación número 18, misma que describe la ecuación general de energía más detallada.

$$Z_1 + \frac{P_1}{\gamma} + \frac{V_1^2}{2g} + h_A - h_R - h_L = Z_2 + \frac{P_2}{\gamma} + \frac{V_2^2}{2g} \quad (18)$$

Donde:

Z[m]: Es la altura.

P[KPa]: Es la presión del fluido.

$V \left[\frac{m}{s} \right]$: Es la velocidad del fluido.

$\gamma \left[\frac{KN}{m^3} \right]$: Es el peso específico del fluido.

$g \left[\frac{m}{s^2} \right]$: Es la aceleración de la gravedad.

2.1.4.4 Pérdidas de energía en la tubería

Las pérdidas de carga que tienen lugar en el transporte de fluidos por tuberías, representan la pérdida de energía de un flujo a lo largo de la misma, estas pérdidas en tuberías se dividen en dos clases.

A. Pérdidas primarias.

Estas pérdidas son producidas por el contacto entre la superficie y el fluido en la tubería (capa límite), rozamiento de unas capas de fluido con otras (régimen laminar) o de partículas de fluido entre sí (régimen turbulento) [35].

Para el cálculo de estas pérdidas se hace uso de la ecuación general de la pérdida de presión, también conocido como la fórmula de Darcy (ecuación 19), la cual es válida tanto para flujo laminar como turbulento de cualquier líquido en una tubería [37].

$$h_f = f * \frac{L}{D} * \frac{V^2}{2g} \quad (19)$$

Donde:

$h_f[m]$: Es la pérdida de energía debido a la fricción.

f : Es el factor de fricción.

$L[m]$: Es la longitud de la corriente del fluido.

$D[m]$: Es el diámetro de la tubería.

$V \left[\frac{m}{s} \right]$: Es la velocidad del fluido.

$g \left[\frac{m}{s^2} \right]$: Es la aceleración de la gravedad.

El factor de fricción (f) es un valor adimensional y para un flujo laminar se calcula mediante la ecuación número 20.

$$f = \frac{64}{NRe} \quad (20)$$

Mientras que para un flujo turbulento donde el movimiento de las partículas es caótico, se dispone de varios métodos para calcular el factor de fricción (f) entre ellos los más importantes: Diagrama de Moody y ecuación de P. k. Swamee y S. K. Jainy.

El Diagrama de Moody (ver ANEXO A) depende del Número de Reynolds y de la rugosidad relativa. Esta rugosidad relativa está relacionada con la rugosidad absoluta del material y del diámetro de la tubería como se indica en la expresión número 21.

$$\text{Rugosidad relativa} = \frac{\varepsilon}{D} \quad (21)$$

Donde:

ε [mm]: Es la rugosidad absoluta.

D[mm]: Es el diámetro de la tubería.

En la tabla 2.9 se exhibe los valores de rugosidad de algunos materiales en diferentes condiciones.

Tabla 2.9 Valores de rugosidad para conductos comerciales [8].

Rugosidad absoluta ε		
Material	Condición	mm
Acero	Lámina metálica, nueva	0.05
	Inoxidable	0.02
	Comercial, nuevo	0.046
	Estriado	3.0
	Oxidado	2.0
Hierro	Fundido, nuevo	0.26
	Forjado, nuevo	0.046
	Galvanizado, nuevo	0.15
	Fundido asfáltico	0.12
Latón	Laminado	0.002
Plástico	Tuvo laminado	0.0015
Vidrio	-----	Liso
Hormigón	Liso	0.04
	Rugoso	2.0
Caucho	Liso	0.01
Madera	En duelas	0.5

Por otro lado, se tiene la ecuación de P. k. Swamee y S. K. Jain que sirve también para el computo del factor de fricción (f) y está en función de las siguientes variables:

rugosidad absoluta (ε [mm]), diámetro de la tubería (D [mm]) y del Número de Reynolds (NRe) tal como se describe en la ecuación número 22.

$$f = \frac{0.25}{\left[\log_{10} \left(\frac{1}{3.7 * \left(\frac{D}{\varepsilon} \right)} + \frac{5.74}{NRe^{0.9}} \right) \right]^2} \quad (22)$$

B. Pérdidas secundarias.

Para instalaciones reales se utiliza accesorios como codos, válvulas, reducciones, entre otros, según sea el requerimiento del entorno a realizar la instalación hidráulica. Estas pérdidas por accesorios son también conocidas como pérdidas secundarias y se calculan mediante la expresión número 23.

$$h_l = K * \frac{V^2}{2g} \quad (23)$$

Donde:

h_l [m]: Es la pérdida por accesorios.

K: Es el coeficiente o factor de pérdidas por accesorio.

$V \left[\frac{m}{s} \right]$: Es la velocidad del fluido.

$g \left[\frac{m}{s^2} \right]$: Es la aceleración de la gravedad.

Y el coeficiente o factor de pérdidas por accesorios (K) de los elementos que se utiliza para la implementación del sistema hidráulico se registra en la tabla 2.10.

Tabla 2.10 Coeficiente de pérdidas por accesorios [35] [34] [38] [39].

Elemento	Factor K
Válvula de pie	3.00
Codo de 90°	0.90
Válvula de bola o esférica	10
Reducción brusca	0.35
Te	1.8
Medidor de agua	7

2.1.4.5 Principio de la continuidad

El principio de la continuidad es un caso particular de la conservación de la masa, donde el flujo másico (M) de un fluido se conserva tanto en la sección 1 como en la sección 2, tal como se exhibe en la figura 2.10.

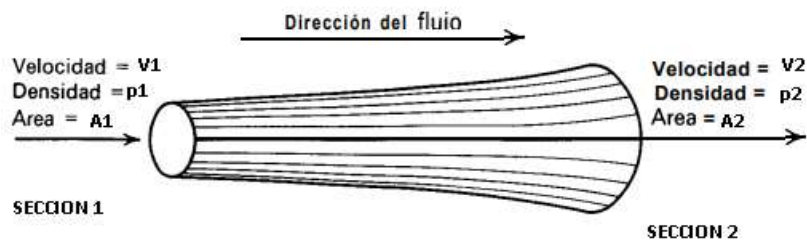


Figura 2.10 Continuidad de un fluido [37]

Esto sucede debido a que no se agrega, almacena o retira fluido, en la sección 1 como en la sección 2. A continuación, se describe el principio de la continuidad para cualquier fluido mediante la ecuación número 24.

Donde:

$$M1 = M2 \quad (24)$$

Pero el flujo másico $M = \rho * A * V$

Entonces:

$$\rho_1 * A_1 * V_1 = \rho_2 * A_2 * V_2 \quad (25)$$

Donde:

$\rho \left[\frac{\text{Kg}}{\text{m}^3} \right]$: Es la densidad del líquido.

$A[\text{m}^2]$: Es el área transversal de la tubería.

$V \left[\frac{\text{m}}{\text{s}} \right]$: Es la velocidad del líquido.

Pero si se considera un líquido incomprensible, entonces ρ_1 y ρ_2 son iguales. Así se obtiene la ecuación de la continuidad de líquidos, donde el caudal (Q) permanece constante tanto en la sección 1 como en la sección 2.

$$A_1 * V_1 = A_2 * V_2 \quad (26)$$

Pero por definición el caudal $Q = A * V$ (ver ecuación 4), entonces se obtienen la ecuación de la continuidad descrita por la expresión número 27.

$$Q_1 = Q_2 \quad (27)$$

2.1.4.6 Altura total de presión de retención (NPSH)

El NPSH está relacionado directamente con el fenómeno de la cavitación y para evitar este fenómeno debe cumplir con la siguiente condición descrita por la ecuación número 28.

$$\text{NPSHdis}[m] > \text{NPSHr}[m] + 0.5[m] \quad (28)$$

Donde el NPSH requerido (NPSHr) está dado por el fabricante y depende estrictamente de las características de la bomba y no de la instalación como es el caso del NPSH disponible (NPSHdis) [37] el cual se puede calcular por medio de la expresión número 29.

$$\text{NPSHdis} = \text{hsp} - \text{hvp} \pm \text{hs} - \text{hf} \quad (29)$$

Donde:

$\text{hsp}[m]$: Es la carga de presión estática (absoluta) sobre el fluido en el almacenamiento y es igual a la expresión: $\frac{P_{sp}}{\gamma}[m]$ donde P_{sp} es la presión estática (absoluta) sobre el fluido en el depósito.

$\text{hvp}[m]$: Es la carga de presión de vapor del líquido a la temperatura de bombeo y es igual a: $\frac{P_{vp}}{\gamma}[m]$ donde P_{vp} es la presión de vapor (absoluta) del líquido a la temperatura de bombeo.

$\text{hs}[m]$: Es la diferencia de elevación desde el nivel del fluido en el depósito a la línea central de la entrada de succión de la bomba, si la bomba está arriba del depósito, hs es negativa caso contrario es positiva.

$\text{hf}[m]$: Es la pérdida de carga en la tubería de succión debido a la fricción y pérdidas menores.

Pero sin embargo existen en el mercado bombas que no disponen en la información técnica el parámetro HPSHr. En este caso se podrán utilizar curvas de referencia que contengan características similares de la bomba, es decir: diámetro y tipo de impulsor, velocidad, tamaño, etc [35].

2.1.4.7 Cálculos del dimensionamiento de la bomba

Para efectuar los respectivos cálculos del dimensionamiento de la bomba se realiza un diagrama simplificado del sistema de extracción de agua en el cual está detallado la ubicación del nivel de referencia y de los puntos 1 y 2 que serán analizados para la aplicación de la ecuación general de energía (ecuación 18).

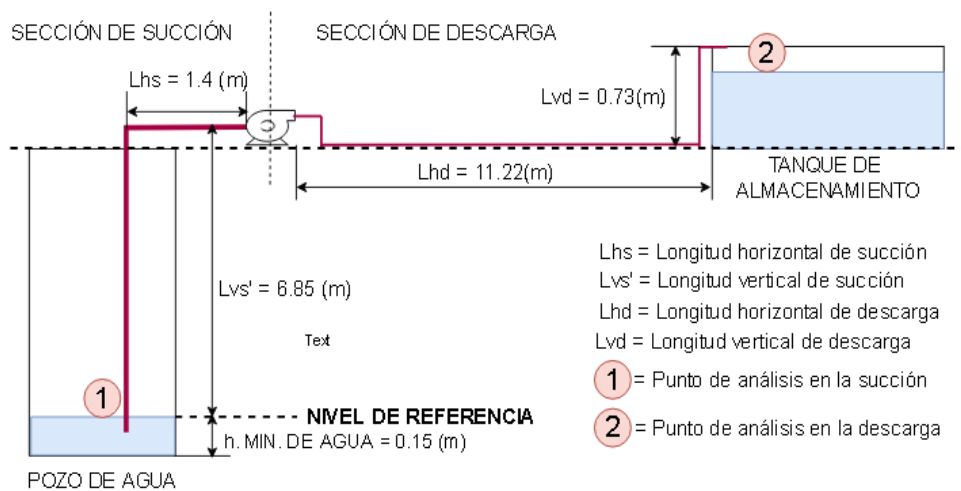


Figura 2.11 Diagrama simplificado del sistema de bombeo

El punto 1 de análisis está ubicado en la sección de la succión del sistema a una altura o nivel de 0.15 [m], considerando que es el caso más crítico del sistema donde el nivel es el mínimo considerado dentro de los requerimientos estudiado del sistema, de igual manera el punto 2 de análisis está ubicado en la sección de la descarga tomando en cuenta que la longitud vertical de descarga es constante de 0.73 [m].

A continuación, para el desarrollo de los cálculos se procede a realizar el análisis de la sección de descarga, luego la parte de succión y finalmente del sistema en general, ya que el sistema está constituido de dos tipos de diámetros de tubería.

A. Cálculos en la sección de descarga.

Para el cálculo en esta sección se considera un diámetro de la tubería de 0.0127[m] y se toma una velocidad promedio de 1.6 $\left[\frac{m}{s}\right]$ en líneas de conducción de un fluido poco viscoso (agua) como se indica en la tabla 2.5. Considerando que es un sistema pequeño y de uso doméstico la velocidad considerada es aceptable dentro del rango de velocidad.

Caudal volumétrico promedio:

$$Qd \left[\frac{m^3}{s} \right] = Vd \left[\frac{m}{s} \right] ** Ad[m^2]$$

$$Qd = 1.6 \left[\frac{m^3}{s} \right] * \left(\pi * \left(\frac{\phi_{int.d}[m]}{2} \right)^2 \right)$$

$$Qd = 1.6 \left[\frac{m^3}{s} \right] * \left(\pi * \left(\frac{0.0127[m]}{2} \right)^2 \right)$$

$$Qd = 0.00020 \left[\frac{m^3}{s} \right]$$

Número de Reynolds (NRe):

$$NRe = \frac{\rho \left[\frac{Kg}{m^3} \right] * Vd \left[\frac{m}{s} \right] * \varnothing_{int. d}[m]}{\mu \left[\frac{Kg}{m * s} \right]}$$

$$NRe. d = \frac{998 \left[\frac{Kg}{m^3} \right] * 1.6 \left[\frac{m}{s} \right] * 0.0127[m]}{1.0 \times 10^{-3} \left[\frac{Kg}{m * s} \right]}$$

NRe. d = 20279.36 ∴ Flujo en regimen turbulento.

Rugosidad relativa del material (plástico tubo laminado):

$$\frac{\varepsilon [mm]}{D[mm]}$$

$$\frac{\varepsilon [mm]}{\varnothing_{int. d}[mm]}$$

$$\frac{0.0015[mm]}{12.7[mm]} = 0.00012$$

Factor de fricción (f):

$$f = \frac{0.25}{\left[\log_{10} \left(\frac{1}{3.7 * \left(\frac{D[mm]}{\varepsilon[mm]} \right)} + \frac{5.74}{NRe^{0.9}} \right) \right]^2}$$

$$f. d = \frac{0.25}{\left[\log_{10} \left(\frac{1}{3.7 * \left(\frac{\varnothing_{int. d}[mm]}{\varepsilon[mm]} \right)} + \frac{5.74}{NRe. d^{0.9}} \right) \right]^2}$$

$$f. d = \frac{0.25}{\left[\log_{10} \left(\frac{1}{3.7 * \left(\frac{12.7[mm]}{0.0015[mm]} \right)} + \frac{5.74}{20279.36^{0.9}} \right) \right]^2}$$

$$f. d = 0.026$$

Para el cálculo de las pérdidas primarias en descarga la longitud de la corriente del fluido (L) es $L = L_d = L_{hd} + L_{vd} = 11.22 + 0.73 = 11.95 \text{ [m]}$ como se indica en la figura 2.12.

$$h_{f.d} = f.d * \frac{L_d[m]}{\varnothing_{int.d}[m]} * \frac{V_d^2}{2g \left[\frac{m}{s^2} \right]}$$

$$h_{f.d} = 0.026 * \frac{11.95[m]}{0.0127[m]} * \frac{\left(1.6 \left[\frac{m}{s} \right] \right)^2}{2 * \left(9.807 \left[\frac{m}{s^2} \right] \right)}$$

$$h_{f.d} = 3.19 \text{ [m]}$$

Pérdidas secundarias.

$$h_l = \sum_{i=0}^n K_i * \frac{V \left[\frac{m}{s} \right]^2}{2g \left[\frac{m}{s^2} \right]}$$

$$h_{l.d} = \left((K_r) + (9 * K_c) + (K_v) + (2 * K_t) + (K_f) \right) * \frac{V_d^2}{2g}$$

$$h_{l.d} = \left((0.35) + (9 * 0.90) + 10 + (2 * 1.8) + (7) \right) * \frac{\left(1.6 \left[\frac{m}{s} \right] \right)^2}{2 * \left(9.807 \left[\frac{m}{s^2} \right] \right)}$$

$$h_{l.d} = 3.79 \text{ [m]}$$

Entonces las pérdidas de energía en la tubería en lado de la descarga son:

$$h_{L.d} = h_{f.d} + h_{l.d} = 6.98 \text{ [m]}$$

B. Cálculos en la parte de la succión.

Por la definición de la ecuación de la continuidad se concluye que:

$$Q_d = Q_s$$

$$Q_d = V_s * A_s$$

$$V_s = \frac{Q_d \left[\frac{m^3}{s} \right]}{A_s [m]}$$

$$V_s = \frac{Q_d \left[\frac{m^3}{s} \right]}{\pi * \left(\frac{\varnothing_{int.s}}{2} \right)^2}$$

$$V_s = \frac{0.00020 \left[\frac{\text{m}^3}{\text{s}} \right]}{\left(\pi * \left(\frac{0.0254}{2} \right)^2 \right)}$$

$$V_s = 0.4 \left[\frac{\text{m}}{\text{s}} \right]$$

Número de Reynolds (NRe):

$$\text{NRe} = \frac{\rho \left[\frac{\text{Kg}}{\text{m}^3} \right] * V \left[\frac{\text{m}}{\text{s}} \right] * \phi [\text{m}]}{\mu \left[\frac{\text{Kg}}{\text{m} * \text{s}} \right]}$$

$$\text{NRe.s} = \frac{\rho \left[\frac{\text{Kg}}{\text{m}^3} \right] * V_s \left[\frac{\text{m}}{\text{s}} \right] * \phi_{\text{int.s}} [\text{m}]}{\mu \left[\frac{\text{Kg}}{\text{m} * \text{s}} \right]}$$

$$\text{NRe.s} = \frac{998 \left[\frac{\text{Kg}}{\text{m}^3} \right] * 0.4 \left[\frac{\text{m}}{\text{s}} \right] * 0.0254 [\text{m}]}{1.0 \times 10^{-3} \left[\frac{\text{Kg}}{\text{m} * \text{s}} \right]}$$

NRe.s = 10139.68 ∴ Flujo en regimen turbulento.

Rugosidad relativa del material:

$$\frac{\varepsilon [\text{mm}]}{D[\text{mm}]}$$

$$\frac{\varepsilon [\text{mm}]}{\phi_{\text{int.s}} [\text{mm}]}$$

$$\frac{0.0015 [\text{mm}]}{25.4 [\text{mm}]} = 0.000059$$

Factor de fricción (f):

$$f = \frac{0.25}{\left[\log_{10} \left(\frac{1}{3.7 * \left(\frac{D[\text{mm}]}{\varepsilon[\text{mm}]} \right)} + \frac{5.74}{\text{NRe}^{0.9}} \right) \right]^2}$$

$$f.s = \frac{0.25}{\left[\log_{10} \left(\frac{1}{3.7 * \left(\frac{\phi_{\text{int.s}} [\text{mm}]}{\varepsilon[\text{mm}]} \right)} + \frac{5.74}{\text{NRe.s}^{0.9}} \right) \right]^2}$$

$$f.s = \frac{0.25}{\left[\log_{10} \left(\frac{1}{3.7 * \left(\frac{2.54[\text{mm}}{0.0015[\text{mm}} \right)} \right) + \frac{5.74}{10139.68^{0.9}} \right) \right]^2}$$

$$f.s = 0.0319$$

Para el cálculo de las pérdidas primarias en succión la longitud de la corriente del fluido (L) es $L = Ls' = Lhs + Lvs' = 1.40 + 6.85 = 8.25$ [m] como se indica en la figura 2.12.

$$hf = f * \frac{L[\text{m}]}{D[\text{m}]} * \frac{V \left[\frac{\text{m}}{\text{s}} \right]^2}{2 * g \left[\frac{\text{m}}{\text{s}^2} \right]}$$

$$hf.s = f.s * \frac{Ls'[\text{m}]}{\emptyset_{\text{int. s}}[\text{m}]} * \frac{Vs^2}{2g \left[\frac{\text{m}}{\text{s}^2} \right]}$$

$$hf.s = 0.0319 * \frac{8.25[\text{m}]}{0.0254[\text{m}]} * \frac{\left(0.4 \left[\frac{\text{m}}{\text{s}} \right] \right)^2}{2 * \left(9.807 \left[\frac{\text{m}}{\text{s}^2} \right] \right)}$$

$$hf.s = 0.085 \text{ [m]}$$

Pérdidas secundarias:

$$hl = \sum_{i=0}^n Ki * \frac{V \left[\frac{\text{m}}{\text{s}} \right]^2}{2g \left[\frac{\text{m}}{\text{s}^2} \right]}$$

$$hl.s = (Kv + Kc) * \frac{Vs^2}{2g \left[\frac{\text{m}}{\text{s}^2} \right]}$$

$$hl.s = (3 + 0.9) * \frac{\left(0.4 \left[\frac{\text{m}}{\text{s}} \right] \right)^2}{2 * \left(9.807 \left[\frac{\text{m}}{\text{s}^2} \right] \right)}$$

$$hl.s = 0.032 \text{ [m]}$$

Entonces las pérdidas de energía en la tubería en lado de la succión son:

$$hL.s = hf.s + hl.s = 0.117[\text{m}]$$

A continuación, se realiza el cálculo del NPSH disponible, donde se considera la Psp en Quito de 71.9941[KPa] y la Pvp del agua de 2.338 [KPa] a 20°C [37].

$$NPSH_{dis} = h_{sp} - h_{vp} \pm h_s - h_f$$

$$NPSH_{dis} = h_{sp} - h_{vp} - L_{vs} - h_{L.s}$$

$$NPSH_{dis} = \frac{P_{sp} - P_{vp}}{\gamma} - L_{vs} - h_{L.s}$$

$$NPSH_{dis} = \frac{71994.1[\text{Pa}] - 2338[\text{Pa}]}{9787.34 \left[\frac{\text{N}}{\text{m}^3} \right]} - 6.85[\text{m}] - 0.117[\text{m}]$$

$$NPSH_{dis} = 0.15[\text{m}]$$

C. Cálculos en el sistema hidráulico.

A continuación, se procede a realizar el análisis del sistema en general con lo cual se puede concluir que las pérdidas de energía del sistema por la fricción en tuberías, o pérdidas menores por válvulas u otros accesorios son:

$$h_L = h_{L.d} + h_{L.s} = 6.98[\text{m}] + 0.117[\text{m}] = 7.097 [\text{m}]$$

Los puntos de análisis y el nivel de referencia como se exhiben en la figura 2.12, se ubicaron estratégicamente con fines de poder simplificar varios parámetros de la ecuación 18.

Como los puntos del sistema a examinar están sometidos a la presión atmosférica, se puede concluir que $P_1 = P_2$, además el agua del pozo está en constante llenado por tanto la $V_1 = 0$ y como no existe ningún dispositivo mecánico que remueva energía del sistema entonces $h_R=0$, con lo cual se obtiene la siguiente ecuación simplificada.

$$h_A[\text{m}] - h_L[\text{m}] = Z_2[\text{m}] + \frac{V_2 \left[\frac{\text{m}}{\text{s}} \right]^2}{2 * g \left[\frac{\text{m}}{\text{s}^2} \right]} \quad (30)$$

Donde:

$h_A[\text{m}]$: Es cabeza dinámica de la bomba (HB) y es igual a:

$$HB = (Z_2[\text{m}]) + \frac{V_2 \left[\frac{\text{m}}{\text{s}} \right]^2}{2 * g \left[\frac{\text{m}}{\text{s}^2} \right]} + h_L[\text{m}] \quad (28)$$

$$HB = (L_{vd}[\text{m}] + l_{vs}[\text{m}]) + \frac{V_d^2}{2 * g \left[\frac{\text{m}}{\text{s}^2} \right]} + h_L[\text{m}]$$

$$HB = (0.73[m] + 6.85[m]) + \frac{(1.6 \left[\frac{m}{s} \right])^2}{2 * (9.807 \left[\frac{m}{s^2} \right])} + 7.097$$

$$HB = 14.81 [m]$$

Y por tanto la potencia transmitida al fluido (P) es:

$$P = HB[m] * \gamma \left[\frac{N}{m^3} \right] * Q \left[\frac{m^3}{s} \right] * Fs \quad (29)$$

$$P = HB[m] * \gamma \left[\frac{N}{m^3} \right] * Q.d \left[\frac{m^3}{s} \right] * Fs$$

$$P = 14.81 [m] * 9787.34 \left[\frac{N}{m^3} \right] * 0.00020 \left[\frac{m^3}{s} \right] * 1.5$$

$$P = 43.48[W]$$

$$P = 0.06[hp]$$

Según el resultado del cálculo de potencia (P) bastaría con la adquisición de una bomba de 0.25 [hp], pero en el mercado lo más típico es encontrar bombas de 0.5 [hp], con lo cual se obtiene una eficiencia de: $n\% = \frac{0.06}{0.5} * 100 = 12\%$.

A continuación, en la tabla 2.11 se exhibe un resumen de los parámetros calculados del sistema.

Tabla 2.11 Resumen de los parámetros del sistema.

Parámetro	Valor	
	Descarga	Succión
Velocidad	$V_d = 1.6 \left[\frac{m}{s} \right]$	$V_s = 0.4 \left[\frac{m}{s} \right]$
Caudal	$Q_d = 0.0002 \left[\frac{m^3}{s} \right]$	$Q_s = 0.0002 \left[\frac{m^3}{s} \right]$
Número de Reynolds	$NRe.d = 20279.36$	$NRe.s = 10139.68$
Rugosidad relativa	0.00012	0.000059
Factor de fricción	$f.d = 0.026$	$f.s = 0.0319$
Pérdidas primarias	$hf.d = 3.19 [m]$	$hf.s = 0.085 [m]$
Pérdidas secundarias	$hl.d = 3.79 [m]$	$hl.s = 0.032 [m]$
Pérdidas de energía en la tubería	$hL.d = 6.98 [m]$	$hL.s = 0.117 [m]$
NPSHd (disponible)	-----	$NPSHdis = 0.15 [m]$
Pérdidas de energía en la tubería del sistema	$hL = 7.097 [m]$	
Cabeza dinámica de la bomba	$HB = 14.81 [m]$	
Factor de seguridad	$F_s = 1.5$	
Potencia teórica	$P = 0.06 [hp]$	
Potencia real	$P = 0.5 [hp]$	
Eficiencia	$n = 12\%$	

2.2 SELECCIÓN DEL HARDWARE

Para la construcción del prototipo será necesario utilizar una placa Arduino con un número determinado de pines, ya que el sistema está conformado de varios dispositivos, módulos y elementos electrónicos, a continuación, se describe más a detalles los dispositivos y accesorios de acuerdo con las características y requerimientos del sistema.

2.2.1 PLACAS DE ARDUINO

Dentro de la plataforma Arduino se encuentra varios modelos de PCB (*Printed Circuit Board*, Placa de Circuito Impreso) que dependiendo de la complejidad y servicios de los proyectos a implementar varía sus números de pines y funcionalidades de la placa, en el ANEXO B [40] se aprecia con más detalle todos los modelos de placas existentes en el mercado junto a sus características técnicas.

2.2.1.1 Descripción de la placa Arduino Mega 2560.

Para el desarrollo del prototipo se manejará una PCB de un determinado número de pines, ya que sobre ella se montará varios sensores, módulos y dispositivos electrónicos, que se describen más adelante.

La placa más adecuada para el diseño e implementación del prototipo de acuerdo a las características y requerimientos del prototipo es la placa Arduino Mega 2560 como se exhibe en la figura 2.12, donde detalla la asignación de pines y puertos.

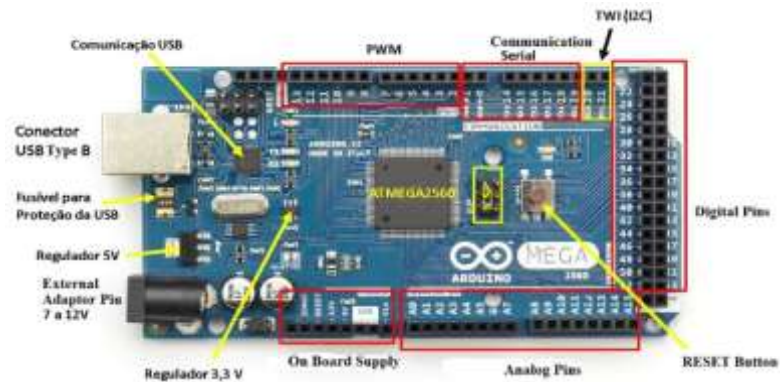


Figura 2.12 Placa Arduino Mega 2560

2.2.1.2 Características de la placa Arduino Mega 2560

Las características más importantes de este modelo de placa se exhiben a continuación en la tabla 2.12.

Tabla 2.12 Características de placa Arduino Mega 2560 [41]

Arduino Mega 2560	
Microcontrolador	ATmega 2560
Voltaje de operación	5 [V] DC
Voltaje de entrada (recomendado)	7-12 [V] DC
Voltaje de entrada (limite)	6-20 [V] DC
Corriente por pin de E/S	20 [mA] DC
Corriente para 3.3 [V] Pin	50 [mA] DC
Pines digitales de E/S	54 (de los cuales 15 proporcionan salidas PWM)
Memoria flash	256 KB de los cuales 8 KB utilizados por el gestor de arranque
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz
Peso	37 gr

2.2.2 MÓDULO DE CONEXIÓN BLUETOOTH

Es el dispositivo que sirve para realizar una comunicación inalámbrica de corto alcance, con el smartphone mediante una aplicación móvil desarrollada en App inventor para el control y monitoreo del sistema.

2.2.2.1 Descripción del módulo HC-05

Los módulos más comunes son HC-05 y el HC-06, y se diferencian en el modo de operación y número de pines, HC-05 opera como maestro-esclavo y tiene 6 pines

distribuidos como se exhiben en la figura 2.13, mientras que el HC-06 trabajo solo como esclavo y consta de 4 pines.



Figura 2.13 Módulo HC-05

Para el desarrollo del proyecto se ha escogido el módulo HC-05, que viene por defecto configurado con los siguientes parámetros:

- Modo o role: Esclavo.
- Nombre por defecto: HC-05.
- Código de emparejamiento por defecto: 1234.
- La velocidad por defecto: 9600 baudios.

2.2.2.2 Especificaciones técnicas del módulo HC-05

Las propiedades técnicas más importantes del módulo HC-05 son las que se exhiben en la tabla 2.13.

Tabla 2.13 Características del módulo HC-05

Módulo Bluetooth HC-05	
Versión Bluetooth	v2.0 + EDR
Frecuencia de operación	2.4 GHz
Modulación	GFSK
Potencia de emisión	≤ 4 dBm, clase 2
Alcance	5 - 10 m
Sensibilidad	≤ -84 dBm a 0.1% VER
Corriente modo sleep	< 1 mA
Consumo de corriente	50 mA (DC)
Voltaje de alimentación	3.6 - 6 V (DC)
Temperatura de operación	-20 °C a +75 °C
Configuración	Por comandos AT
Seguridad	Autenticación y encriptación

El lector puede dirigirse al apartado [42] donde podrá encontrar más a detalle las características del módulo HC-05.

2.2.2.3 Conexión del módulo HC-05 y Arduino Mega 2560

En la figura 2.14 se exhibe el modo de conexión entre la PCB escogida y el módulo HC-05.



Figura 2.14 Conexión Arduino Mega 2560 y módulo HC-05

2.2.3 MÓDULO DE CONEXIÓN GSM/GPRS

Un Shield GSM/GPRS es un dispositivo de expansión que sirve para ampliar las capacidades de las placas Arduino, brindando servicios de conectividad inalámbrica con la red celular el cual permite enviar, recibir mensajes de texto, realizar llamadas e incluso conectarnos al internet.

2.2.3.1 Descripción del módulo GSM/GPRS SIM 900

Existen varios modelos de tarjetas o Shield entre los más comunes son GSM/GPRS SIM900 y GSM/GPRS SIM800, entre ellos compatibles.

Para el control y monitoreo del prototipo a través de mensajes de texto, se ha establecido utilizar el Shield de Arduino GSM/GPRS SIM900 (figura 2.15) el cual está fundamentado en el SIM900 de SIMCOM y es compatible con la placa Mega 2560.

Para la configuración y control del módulo SIM900 se ejecuta a través de su UART (*Universal Asynchronous Receiver-Transmitter / Transmisor-Receptor Asíncrono Universal*) utilizando comando AT (*Attention / Atención*).

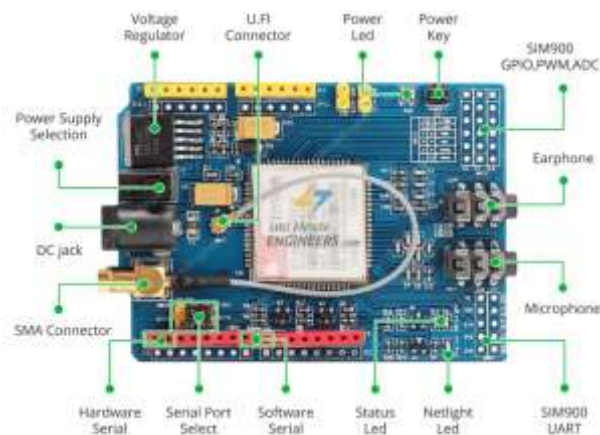


Figura 2.15 Shield GSM/GPRS SIM900

2.2.3.2 Especificaciones técnicas del módulo GSM/GPRS SIM 900

Las características más importantes del shields SIM900 se mencionan a continuación en la tabla 2.14, dando la opción al lector ver más a detalle en [43].

Tabla 2.14 Características de Shield GSM/GPRS SIM900

Módulo GSM/GPRS SIM900	
Voltaje de alimentación	3.4 - 4.5 V (DC)
Temperatura de operación	-30°C a +80°C
Banda cuádruple	850, 900, 1800 y 1900 MHz
Consumo promedio de potencia en modo de transmisión	1.25 [W]
Estación móvil GPRS	Clase B

2.2.3.3 Conexión del módulo SIM 900 y Arduino Mega 2560

En la figura 2.16 se exhibe la conexión entre la PCB escogida y el módulo Shield GSM/GPRS SIM900.

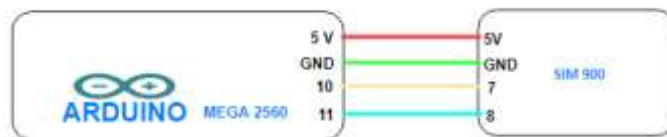


Figura 2.16 Conexión Arduino Mega 2560 y módulo SIM900

2.2.4 MÓDULO ADAPTADOR DE TARJETA MICRO_SD

Los módulos de tarjeta SD son especialmente útil para proyectos que requieren el registro de datos.

Existen dos modelos de módulos dentro de la plataforma Arduino:

- 1) Módulo micro SD: posibilita introducir una tarjeta micro SD que son las más típicas en el mercado.
- 2) Módulo SD card: permite insertar tarjetas SD de tamaño estándar, pero usando un adaptador se puede usar también memorias micro SD.

Para el diseño del proyecto se utilizará una tarjeta micro SD junto con un módulo micro SD para obtener un diseño compacto y económico.

2.2.4.1 Descripción del adaptador de tarjeta Micro_SD

En la figura 2.17 se observa los pines y las partes del módulo lector de tarjetas micro SD.



Figura 2.17 Módulo Adaptador De Tarjeta Micro_SD

Arduino soporta formatos SD (*Secure Digital / Seguro Digital*) o SDSC (*Secure Digital Standard Capacity / Capacidad Estándar Digital Segura*) y SDHC (*Secure Digital High Capacity / Capacidad Alta Digital Segura*).

SD es un formato de tarjeta de memoria no volátil. Para que la librería de Arduino pueda reconocer sin ningún problema la tarjeta SD se recomienda formatear en formato FAT16 o FAT32, por otro lado, la tasa garantizada de transmisión de datos viene definida por la “clase” de la tarjeta.

2.2.4.2 Especificaciones técnicas del adaptador de tarjeta Micro_SD

Las especificaciones técnicas principales del módulo micro SD [44] son las que se mencionan en la tabla 2.15.

Tabla 2.15 Características de modulo micro SD

Módulo Adaptador De Tarjeta Micro_SD	
Comunicación	Interfaz SPI
Rango de voltaje	3.5 a 5 V (DC)
Consumo de corriente	100 [mA], máximo 200 [mA]
Dimensiones	68.7x53.5x19.00 [mm],
Peso neto	14.8 gr

2.2.4.3 Conexión del adaptador Micro_SD y Arduino Mega 2560

En la figura 2.18 se exhiben la conexión entre el módulo de almacenamiento de datos y la PCB Arduino Mega 2560. Hay que tomar en cuenta si se está utilizando un módulo SD Card estándar o un módulo micro SD como indica en el apartado [45].

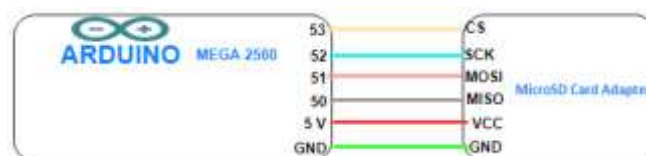


Figura 2.18 Conexión Arduino Mega 2560 y Adaptador De Tarjeta Micro_SD

2.2.5 CONTACTOR

También es conocido como módulo relé de Arduino, existe desde 1 canal hasta 16 canales en el mercado, para la implementación del proyecto solo será necesario utilizar un módulo relé de un canal para el encendido de la bomba eléctrica a 110 V AC.

2.2.5.1 Descripción del relé

Un módulo relé es un interruptor accionado electrónicamente que posibilita a un procesador como Arduino controlar cargas a un nivel de tensión o intensidad superiores a las que su electrónica puede soportar. En la figura 2.19 se tiene las partes del módulo relé de un canal.



Figura 2.19 Módulo relé de un canal.

Los relés disponen de dos circuitos los cuales están acoplados por medio de un optoacoplador, el circuito primario que se conecta con la electrónica de baja tensión y toma la señal de encendido y apagado; y el circuito secundario es el interruptor encargado de encender o apagar la carga, consta de tres contactos que son NC (Normalmente Cerrado), C (Común), NO (Normalmente Abierto) [46].

2.2.5.2 Especificaciones técnicas del relé de un canal

Las características [47] principales de módulo relé se presentan a continuación en la tabla 2.16.

Tabla 2.16 Características de módulo relé de 1 canal

Módulo relé de 1 canal	
Rango de voltaje	3.75 a 6 V (DC)
Suministro de corriente con relé desenergizado	2 mA (DC)
Suministro de corriente con relé energizado	70 a 72 mA (DC)
Señal de control de entrada	activa baja
Corriente de señal de control de entrada	1.5 a 1.9 mA
Voltaje de contacto máximo del relé	250VAC o 30VDC
Relé máximo contacto corriente	10 A
Tiempo de conmutación	10 ms
Soporta señales PWM	No

2.2.5.3 Conexión del relé y Arduino Mega 2560

En la figura 2.20 se presenta el esquema de conexión entre la PCB Mega 2560 y el módulo relé.

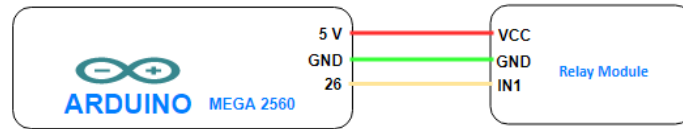


Figura 2.20 Conexión Arduino Mega 2560 y módulo relé

2.2.6 SERVO MOTOR

Existe una gran variedad de servos, que varían en tamaño, fuerza, velocidad y precisión.

Para la implementación del prototipo, se utilizará un servo motor MG996R como se observa en la figura 2.21 para controlar la abertura y obstrucción de la válvula esférica la cual estará instalada a la sección de la descarga del sistema.



Figura 2.21 Servo MG996R.

2.2.6.1 Descripción del servo motor MG996R

Un servomotor es llamado comúnmente servo y es un tipo de accionador, en el que se indica directamente el ángulo deseado entre 0° y 180° , proporcionan alto grado de precisión y disponen de velocidades pequeñas de giro frente a los motores DC.

El control de giro del motor es por medio de la señal PWM, donde el tiempo en alto es equivalente al ángulo o posición del servo tal como se exhibe en la figura 2.22

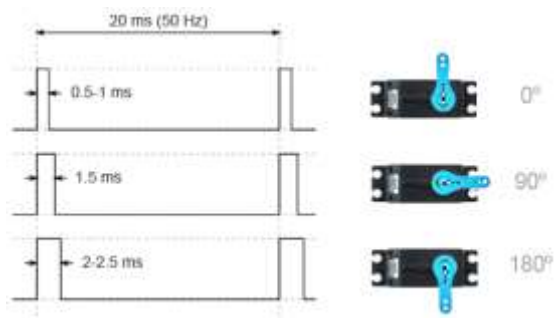


Figura 2.22 Posición de ángulo en un periodo de 20 ms [48]

2.2.6.2 Especificaciones técnicas del Servo MG996R

En la tabla 2.17 se exhibe las propiedades técnicas [49] más importantes del servo MG996R.

Tabla 2.17 Características de servo MG996R.

Servo MG996R	
Voltaje de operación	4.8 - 6.6 V (DC)
Voltaje recomendable	5 - 6 V (DC)
Velocidad de operación	0.19 s/ 60° (4.8 V), 0.15 s/ 60° (6 V)
Torque	9,4 kg/cm (4.8 V), 11 kg/cm. (6 V)
Peso	55 g
Rango de temperatura	0 °C – 55 °C
Giro máximo	180 °

2.2.6.3 Conexión servo MG996R y Arduino Mega2560

En la figura 2.23 se exhibe el diagrama de conexión entre el Mega 2560 y el accionador MG 996R.



Figura 2.23 Conexión Arduino Mega 2560 y Servo MG 996R

2.2.7 SENSOR ULTRASÓNICO

Es un dispositivo que mide la distancia entre el emisor y el objeto mediante ondas ultrasónicas, tal como se aprecia en la figura 2.24.

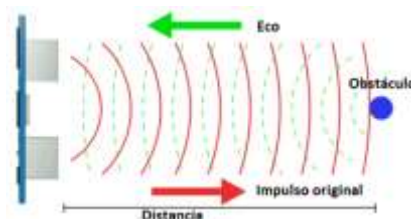


Figura 2.24 Emisión y recepción de una onda sonora [50]

Entre la variedad de sensores ultrasónicos que existen en el mercado, se utilizará el modelo HC-SR04 (figura 2.25) ya que cumple con los requerimientos (rango de distancia) para el diseño del prototipo, y además es de fácil acceso en el mercado y económico.



Figura 2.25 Sensor ultrasónico HC-SR04

2.2.7.1 Descripción del módulo HC-SR04

El módulo HC-SR04 dispone de un emisor y receptor por separados, este diseño permite recibir el eco de mejor manera permitiendo mantener una mejor precisión de 3mm con un ángulo de apertura de 15°.

Según el datasheet [51] del sensor HC-SR04, para lanzar un impulso sonoro se deberá generar un pulso de al menos 10us en la entrada del trig, tal como se exhibe en la figura 2.26.

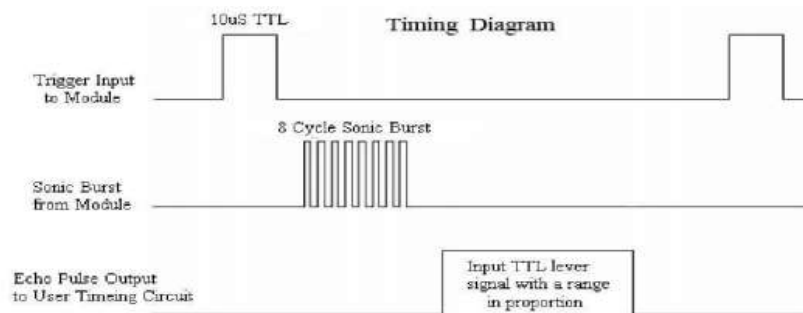


Figura 2.26 Diagrama de tiempo para el impulso sonoro

2.2.7.2 Especificaciones técnicas del módulo HC-SR04

Las propiedades técnicas del módulo HC-SR04 se presentan en la tabla 2.18

Tabla 2.18 Características del módulo HC-SR04 [51].

Sensor ultrasónico HC-SR04	
Voltaje de operación	5 V (DC)
Corriente de operación	15 mA
Corriente en reposo	< 2 mA
Frecuencia de operación	40 KHz
Rango distancia	2 cm – 4 m
Pulso de disparo TRIG	10 us TTL
Pulso ECO de salida	100 – 25000 us
Precisión	+/- 3 mm

2.2.7.3 Conexión del módulo HC-SR04 y Arduino Mega 2560

En la figura 2.27 se exhibe la conexión entre la PCB Arduino Mega 2560 y el módulo HC-SR04.



Figura 2.27 Conexión Arduino Mega 2560 y módulo HC-SR04

2.2.8 SENSOR DE CAUDAL

Existen caudalímetros que dependen del diámetro de la tubería; en Arduino se dispone de los siguientes modelos: YF-S401 para tuberías de $\frac{1}{4}$ ", YF-S201 para tuberías de diámetro de $\frac{1}{2}$ ", FS300A para tuberías $\frac{3}{4}$ " y el modelo FS400A que es para tuberías con diámetro de 1".

Para satisfacer los requerimientos que se necesita para el diseño del sistema de automatización de bombeo de agua se dispondrá del módulo YF-S201 debido a que en la sección de descarga se utilizará tubería de $\frac{1}{2}$ " de diámetro.

2.2.8.1 Descripción del módulo YF-S201

En la figura 2.28 se puede apreciar el caudalímetro o flujómetro YF-S201 junto con su distribución de pines, YF-S201 es un flujómetro electrónico a turbina que permite medir el caudal volumétrico, normalmente se expresa en m^3/s aunque también se puede expresar en l/s o l/min.



Figura 2.28 Flujómetro YF-S201

Este módulo dispone de un imán ubicado en el rotor que genera un pulso positivo cada vez que existe flujo de agua y es detectado por el sensor de efecto hall, de esta manera se puede conocer las revoluciones por minuto (RPM) que genera el rotor.

La salida del sensor es una onda cuadrada cuya frecuencia es correspondiente al caudal atravesado como se describe en la siguiente formula: $Q(l/min) = f(Hz)/K$, donde $K = 7.5$ [52]. Para mejorar la precisión se recomienda realizar ensayos para calibrar el caudalímetro y tener un valor de K más real de acuerdo con los requerimientos del sistema.

Además de poder estimar el flujo que atraviesa por la tubería también permite calcular la cantidad de agua extraída sabiendo que el caudal es la variación del volumen respecto al tiempo como se indica en la siguiente fórmula: $Q = \Delta V / \Delta t \rightarrow V = V_0 + Q \cdot \Delta t$ [53].

2.2.8.2 Especificaciones técnicas del módulo YF-S201

El módulo YF-S201 dispone de las siguientes características técnicas descritas en la tabla 2.19.

Tabla 2.19 Características técnicas de sensor YF-S201 [54].

Sensor YF-S201	
Voltaje de trabajo	4.5V - 24V (DC)
Voltaje normal DC	5V a 18V (DC)
Consumo de corriente	15 mA (DC 5V)
Sensor primario	Molinete de 8 alabes
Sensor secundario	Efecto Hall
Rango de flujo	1 - 30 L/min
Volumen promedio por pulso	2.25 mL
Pulsos por litro	450
Factor de conversión	$K=7.5$
Rango de temperatura	-25°C a + 80°C
Presión máxima de trabajo	1.75MPa (17 bar)
Temperatura del liquido	< 120 °C

2.2.8.3 Conexión del módulo YF-S201 y Arduino Mega 2560

Para la conexión del sensor con la PBC Arduino Mega2560 se realiza mediante el pin 2 en vista que se utilizará interrupciones externas como aprecia en la figura 2.29.

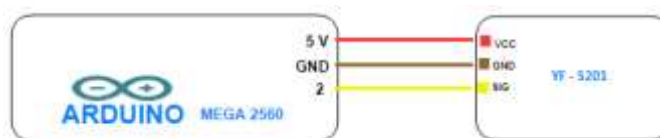


Figura 2.29 Conexión Arduino Mega 2560 y sensor YF-S201

2.2.9 LED RGB

Un led RGB es un instrumento electrónico que tiene la facilidad de reproducir casi cualquier color, existen en dos variantes que se exhibe en la figura 2.30, los de cátodo y ánodo común respectivamente, sin embargo, el más utilizado es el primero por la facilidad de

modificar los valores del ánodo que las del cátodo y el cual se utilizará en el proyecto para mostrar los niveles de agua tanto en el pozo como en el tanque de reserva.

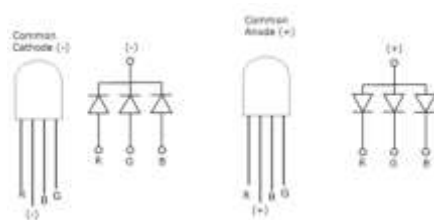


Figura 2.30 Tipos de led RGB [55]

2.2.9.1 Descripción del Led RGB cátodo común

El led RGB de cátodo común consta de un encapsulado de 3 diodos led con los tres colores primarios, rojo, verde y azul, donde la intensidad de cada color va desde 0 hasta 255 y el producto de la mezcla es un color monocromático.

El led RGB dispone de 4 terminales, siendo uno de ellos el cátodo común y el que va conectado a GND de la placa Arduino y para protección de los otros tres terminales se recomienda conectarlos a una resistencia de 220 ohmios a cada terminal.

2.2.9.2 Especificaciones técnicas del Led RGB cátodo común

Las especificaciones técnicas del led RGB cátodo común se describen en la siguiente tabla 2.20.

Tabla 2.20 Características de Led RGB [56]

LED RGB		
Pin común	Cátodo	
Voltaje de operación	Rojo	1.9 -2.5 V (DC)
	Verde	3.3 - 4.1 V (DC)
	Azul	3.3 - 4 V (DC)
Corriente de operación	20 mA	
Temperatura de operación	- +40°C a 85°C	
Ángulo de visión	50°	

2.2.9.3 Conexión del Led RGB y Arduino Mega 2560

En la figura 2.31 se exhibe el boceto de conexión entre el Mega 2560 y el Led RGB cátodo común.

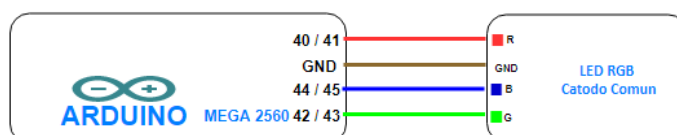


Figura 2.31 Conexión Arduino Mega 2560 y Led RGB cátodo común

2.2.10 FUENTE DE ALIMENTACIÓN

Una fuente de alimentación es conocido como adaptador AC/DC o también como convertidor AC/DC, para el desarrollo del proyecto se utilizará la fuente XBS-1220 como se exhibe en la figura 2.32.



Figura 2.32 Adaptador AC/DC XBS-1220

La implementación del prototipo consta de varios módulos y sensores, la fuente a utilizar debe mínimo disponer de 2000 [mA]. En la tabla 2.21 se describe las propiedades del adaptador AC/DC.

Tabla 2.21 Características del Adaptador AC/DC

Adaptador AC / DC	
Modelo	XBS-1220
Voltaje de entrada y frecuencia	100 – 240 [VAC] 50/60 [Hz].
Voltaje de salida	12 [VDC]
Corriente de salida	2000 [mA].

2.2.11 DISPOSITIVO DE VISUALIZACIÓN DE DATOS

Para interactuar (control y monitoreo) con el prototipo y visualizar la información obtenida del mismo se utilizará un Smartphone de la marca Samsung Galaxy J2 Prime modelo SM-G532M (figura 2.33), para la descargar e instalación de la aplicación desarrollada en el software App Inventor.



Figura 2.33 Smartphone SM-G532M

2.2.11.1 Especificaciones técnicas del smartphone SM-G532M

Las principales características de este modelo de Smartphone se mencionan a continuación en la tabla 2.22.

Tabla 2.22 Características de Smartphone SM-G532M [57].

Samsung Galaxy J2 Prime modelo SM-G532M	
Sistema operativo	Android 6.0.1
CPU Procesador	1.4 GHz Quad-core ARM Cortex-A53
Memoria RAM	1,5GB LPDDR3
Memoria interna	8GB (3GB accesible al usuario)
Frecuencia GSM MHz	Quad-Band 850/900/1800/1900
Red 2G	GSM 850/900/1800/1900
Red 3G	UMTS 850/900/1700/1900/2100
Red 4G	LTE 1(2100), 2(1900), 3(1800), 4(1700/2100), 5(850), 7(2600), 8(900), 17(700), 28(700)
Bluetooth	Versión 4.2 con A2DP

2.2.12 VÁLVULAS

La clasificación de las válvulas es compleja debido a la existencia de una gran variedad de diseños de válvulas, pero sin embargo existe una clasificación según su resistencia que ofrece al flujo [36].

Las válvulas que ofrecen una baja resistencia son las válvulas de compuerta, bola, macho y de mariposa, debido a muestran un paso directo del flujo. En cambio, las válvulas que presentan un cambio en la dirección del flujo son las válvulas que tiene una alta resistencia como las válvulas de globo y angulares.

Las válvulas que se ajustan a los requerimientos técnicos para el diseño del sistema son las válvulas de baja resistencia como son las válvulas de pie y esférica.

2.2.12.1 Válvula de pie

Esta válvula sirve para retención del líquido y se instala en la parte inferior de la succión de un sistema de bombeo; la válvula de pie dispone de una canastilla tipo rejilla (figura 2.34) la cual sirve para evitar la succión de residuos o basuras al momento de la succión del agua por parte de la bomba.

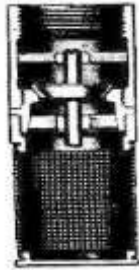


Figura 2.34 Válvula de pie [36]

2.2.12.2 Válvula esférica

También es conocida como válvula de bola, es de apertura rápida debido a que basta con girar un ángulo de 90° para habilitar o interrumpir totalmente el flujo del líquido.

En la figura 2.35 se observa la válvula de bola la cual servirá para acoplarla al servomotor MG996R.

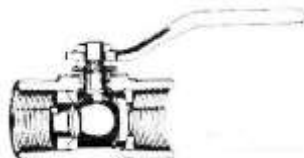


Figura 2.35 Válvula de bola [36]

2.2.13 ACCESORIOS

Un sistema de bombeo necesita de varios accesorios o también llamados acoplamientos de conexión, estos accesorios se separan en: derivación, reducción, ampliación y desviación.

A continuación, se mencionan los accesorios que serán utilizados en la construcción del sistema.

2.2.13.1 Codos

Son considerados accesorios de desvío ya que cambian la dirección del flujo, normalmente existen de 90° y 45°, de acuerdo con el diseño del proyecto será necesario utilizar codos de 90° como se exhibe en la figura 2.36.



Figura 2.36 Codo de 90°

2.2.13.2 Unión

Es un accesorio que sirve para unir las tuberías y dar continuidad a la circulación del agua. En la figura 2.37 se presenta las uniones a utilizar en la implementación el proyecto.



Figura 2.37 Unión roscable y unión universal

2.2.13.3 Reducción

Es un accesorio que sirve para acoplar accesorios de diferentes diámetros, en la figura 2.38 se presenta la reducción de 1" a 1/2".



Figura 2.38 Reducción de 1" a 1/2"

2.2.13.4 Tubería

Es un accesorio que sirve para el transporte del líquido en determinada dirección.



Figura 2.39 Tubería PVC

Todos los accesorios antes mencionados son de material PVC debido a su alta durabilidad, y bajo costo.

2.2.14 BOMBA DE AGUA

De acuerdo con los cálculos realizados y resultados adquiridos en la fase de diseño del prototipo, se utilizará la siguiente bomba PKM60-1 (figura 2.40) que satisface los requerimientos del sistema tales como P[hp], HB [m] y NPSH [m].



Figura 2.40 Bomba de agua PKM60-1

La bomba PKM60-1 es una bomba periférica eléctrica que se utiliza para tanques de reserva, jardines de riego, insuficiencia de presión y aplicaciones sanitarias. Entre sus ventajas que dispone esta bomba es: ligera, silenciosa y eficiente, ayuda a disponer de presiones altas, pero con bajo caudal para altura considerables. En la tabla 2.23 se presenta las características principales de la bomba.

Tabla 2.23 Características de la bomba PKM60-1.

Bomba PKM60-1	
Capacidad máxima	40 (L/min)
Cabeza dinámica de la bomba máxima	36 (m)
Longitud máxima de succión	9 (m)
Potencia	370 (W)
Diámetro de la bomba (succión/descarga)	1"
Voltaje (AC)	115/230 (V)
Corriente (AC)	2.5 (A)
Frecuencia	60 (Hz)

La curva de rendimiento de la bomba PKM60-1 se presenta en la figura 2.41, la cual está en función del caudal y cabeza dinámica.

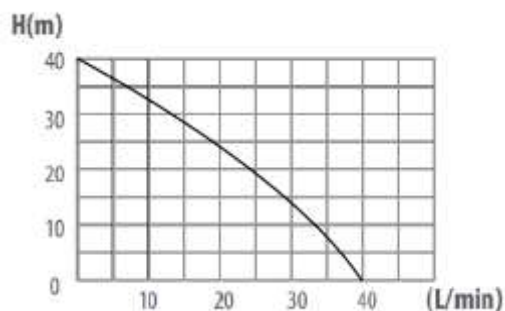


Figura 2.41 Curva de rendimiento de la bomba PKM60-1

Los resultados logrados en la fase de diseño del prototipo son: Cabeza Dinámica de la bomba de 14.81 [m] y un caudal de $12 \left[\frac{L}{min} \right]$ a una velocidad promedio de 1.6 (m/s), con lo que al comparar estos resultados con la curva de rendimiento de la bomba de la figura 2.41 satisface los requerimientos del sistema ya que el punto de trabajo (W) está dentro de la zona de operación de la bomba como se nota en la figura 2.42.

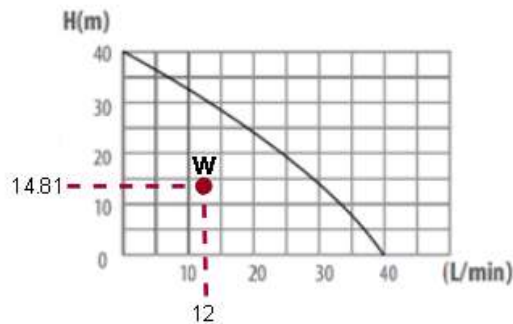


Figura 2.42 Punto de trabajo del sistema

Como se indicó el NPSHr es un dato facilitado por el fabricante. Pero sin embargo este valor no existe en catálogo de la bomba PKM60-1 por ser una bomba periférica de baja potencia (0.5 hp), por lo tanto, no se puede realizar un análisis del fenómeno de cavitación, ya que por lo general este fenómeno se produce en bombas centrífugas de alta potencia es decir mayores a 1.5 (hp).

2.3 CALIBRACIÓN DE SENSORES

Uno de los parámetros a considerar en la marcha del sistema es el valor de lectura de los sensores HC-SR04 y YF-S201, que son los que toman las medidas de nivel y caudal respectivamente.

Para obtener medidas más exactas de nivel se procede a la calibración del sensor HC-SR04 donde se varía la velocidad del sonido, como se mencionó en el capítulo 1 es de 334 (m/s) con lo que se obtiene valores totalmente dispares a las medidas reales, como solución a este error se considera una velocidad de $331,3 + 0,606 * T$ (m/s) [58], donde T es la temperatura en la ciudad de Quito (15 °C) y se reemplaza en la ecuación descrita en el código 2.7.

De igual manera para obtener una medida de caudal correcta se procede a ejecutar ensayos utilizando el método directo volumétrico para obtener un valor de caudal promedio el cual servirá para realizar el cálculo del factor K descrita por la ecuación 8 del capítulo 1

en caso de ser necesario, a continuación, se presenta en la tabla 2.24 los valores tomados por el método descrito y los valores captados por el sensor YF-S02.

Tabla 2.24 Lectura de valores de caudal volumétrico.

Método directo volumétrico			Sensor YF-S02 con K=7.5
Volumen(L)	Tiempo(s)	Caudal (L/min)	Caudal (L/min)
1	6,1	9,84	10
1	6,11	9,82	9,87
1	6,31	9,51	9,73
1	6,45	9,30	9,73
1	6,03	9,95	9,87
1	6,21	9,66	9,87
1	6,01	9,98	9,73
1	6,23	9,63	9,87
1	6,15	9,76	9,73
1	6,24	9,62	9,87
CAUDAL PROMEDIO		9,71	9,83

Como se observa los valores de caudal en la tabla 2.24 están dentro del rango de medición y sobre el valor mínimo calculado por lo tanto se asume que el sensor YF-S201 esta calibrado desde fabrica tomando un valor de $K=7.5$.

2.4 IMPLEMENTACIÓN

Para la implementación de la parte electrónica del prototipo se considera las siguientes partes que se puntualizan a continuación.

2.4.1 SISTEMA DE TRANSMISIÓN

El sistema de trasmisión de datos consta básicamente de un módulo HC-05 para la comunicación inalámbrica vía Bluetooth y de un módulo GSM/GPRS SIM900 para la comunicación por medio de la red celular, los datos a ser transmitidos son los tomados por los sensores HC-SR04 y YF-S201; que son procesados por el microcontrolador ATMEL 2560 para ser enviados por el módulo HC-05.

El módulo GSM/GPRS SIM900 sirve para enviar mensajes de texto en respuesta al funcionamiento y estado del prototipo.

2.4.2 SISTEMA DE RECEPCIÓN

Este sistema costa de un Smartphone con sistema operativo Android para la ejecución de la aplicación móvil desarrollada en la plataforma App Inventor y que sirve para receptor los

datos enviados por el sensor HC-05 para monitorear y/o controlar el sistema de bombeo por medio de una interfaz gráfica.

Además, se dispone de otro dispositivo celular sin importar el sistema operativo para la recepción de mensajes de texto que son enviados por el módulo GSM/GPRS SIM900 indicando el estado del prototipo.

2.4.3 MONTAJE DEL PROTOTIPO

Una vez estudiado el sistema de transmisión, recepción y verificado el funcionamiento de todos los dispositivos electrónicos, se procede a ubicar de manera estratégica dentro de una caja plástica dichos dispositivos electrónicos con su respectiva circuitería tal como se aprecia en la figura 2.43.

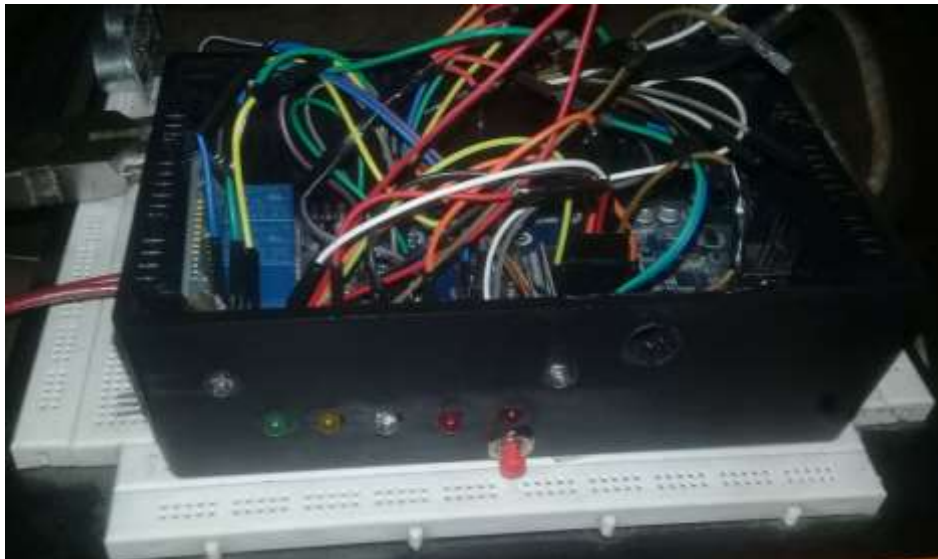


Figura 2.43 Montaje del prototipo

Adicional se exhibe en la figura 2.44 la instalación del sensor HC-SR04 en el taque de almacenamiento y para una mejor precisión el sensor se utilizó un tubo de 3" para evitar movimientos bruscos del agua al momento de tomar el agua del tanque.



Figura 2.44 Instalación de sensores HC-SR04

2.5 PROGRAMACIÓN DEL SISTEMA

En la siguiente sección se describe las herramientas de programación, además se explica de manera breve el código de programación del microcontrolador ATMEL 2560 y el código respectivo de la Aplicación Móvil.

2.5.1 HERRAMIENTAS DE PROGRAMACIÓN

Existen varias herramientas de programación tales como las librerías, lenguajes de programación y otros mediante un ambiente gráfico, que sirven para realizar aplicativos, programas, rutinas, entre otros. En el mercado existen herramientas de programación tanto para expertos como para novatos.

Dentro de todas las herramientas de programación que existen en el mercado, para el presente proyecto se ha optado por utilizar APP INVENTOR para el desarrollo de la aplicación móvil, y para la programación del microcontrolador ATMEL se utilizara la plataforma Arduino.

2.5.1.1 Plataforma Arduino

Arduino nació en el año del 2005 y es un proyecto de hardware y software libre de ámbito mundial, su lenguaje de programación está basado C++, y apareció por la necesidad de contar con dispositivos que fueran de precio accesible, que funcionase bajo cualquier

sistema operativo y que contase con documentación adaptada a gente que quisiera iniciar de cero.

Arduino es una plataforma de desarrollo basada en una placa electrónica que incorpora un microcontrolador ATMELE AVR reprogramable y una serie de pines hembra, los que permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla [59].

Arduino está pensado e inspirado en artistas, diseñadores, estudiantes de computación o robótica y para cualquier persona interesado en crear objetos o entornos interactivos. En la figura 2.45 se exhibe el Entorno de Desarrollo Integrado (IDE) de Arduino, el cual sirve para escribir el código de programación del microcontrolador.

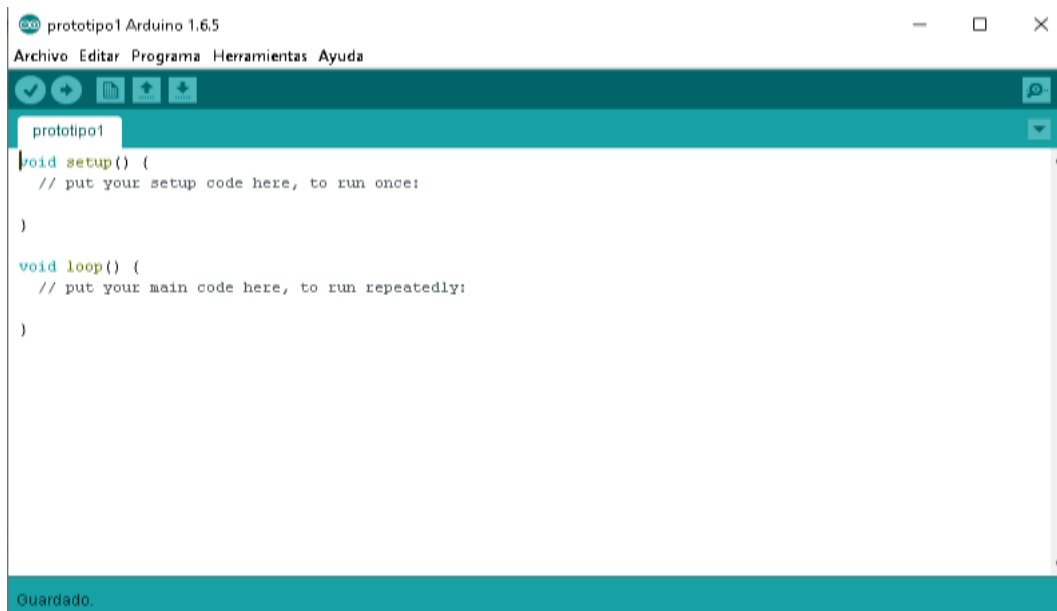


Figura 2.45 IDE de Arduino

2.5.1.2 App Inventor

En la actualidad las aplicaciones móviles han tenido un impacto sorprendente en la sociedad, brindando facilidades al usuario para realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otras posibilidades.

MIT App Inventor brinda la facilidad de diseñar de manera gratuita nuestra propia aplicación móvil para dispositivos con sistema operativo Android, mediante un entorno de programación visual, basado en bloques.

El usuario debe estar sujeto a las políticas de privacidad y condiciones de uso de la plataforma MIT App Inventor que se mencionan en la página oficial [60].

Para el control y monitoreo del sistema automático de extracción de agua se utilizará la herramienta de programación App Inventor, en el cual se desarrollará una aplicación móvil de acuerdo a las características y requerimientos del sistema con una interfaz amigable y fácil de interactuar.

En la figura 2.46 se exhibe la interfaz de programación de la aplicación en la plataforma MIT App Inventor.

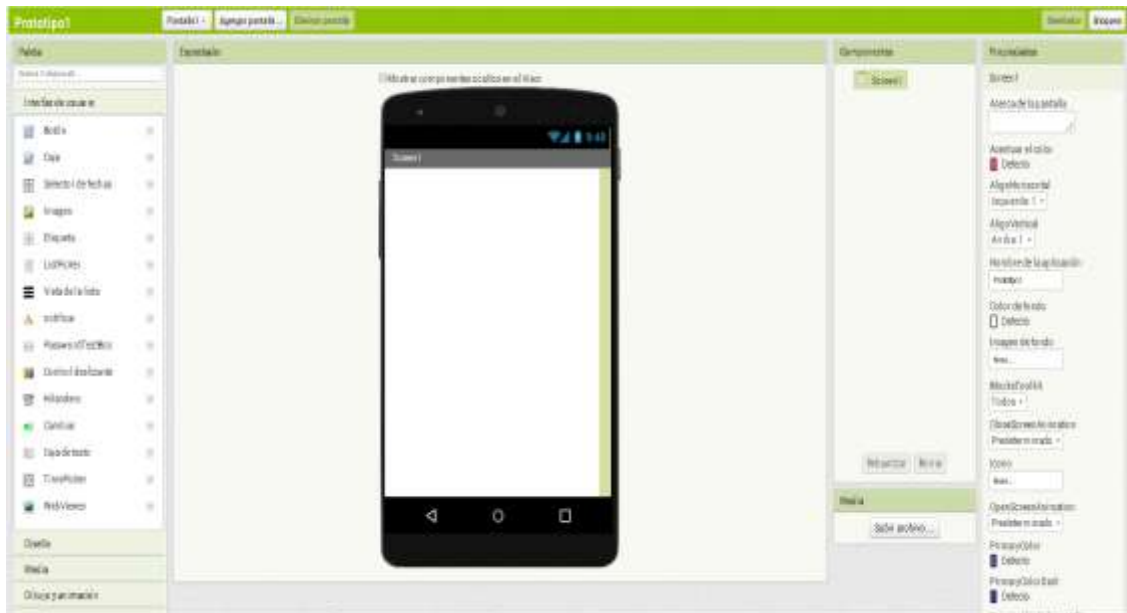
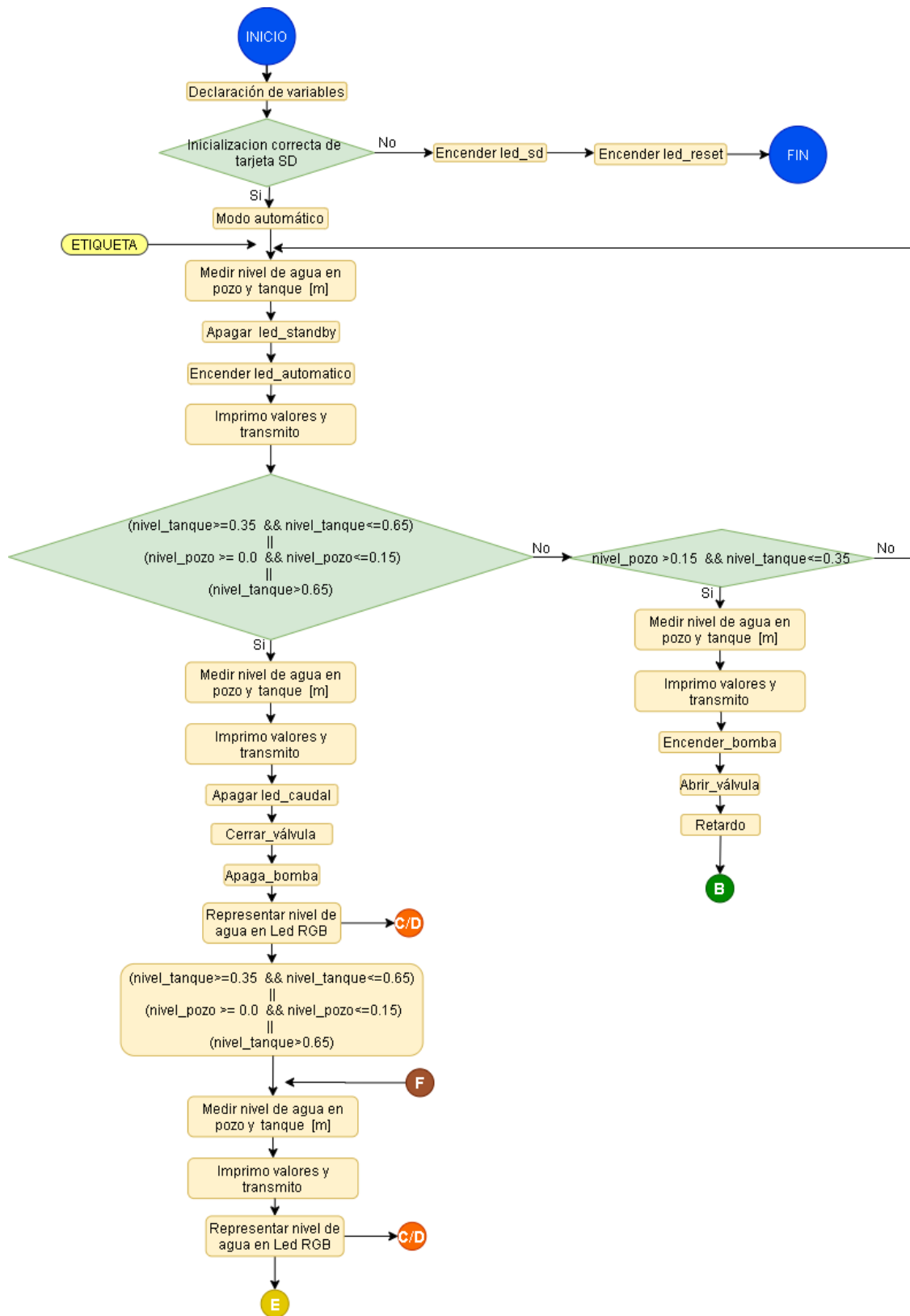
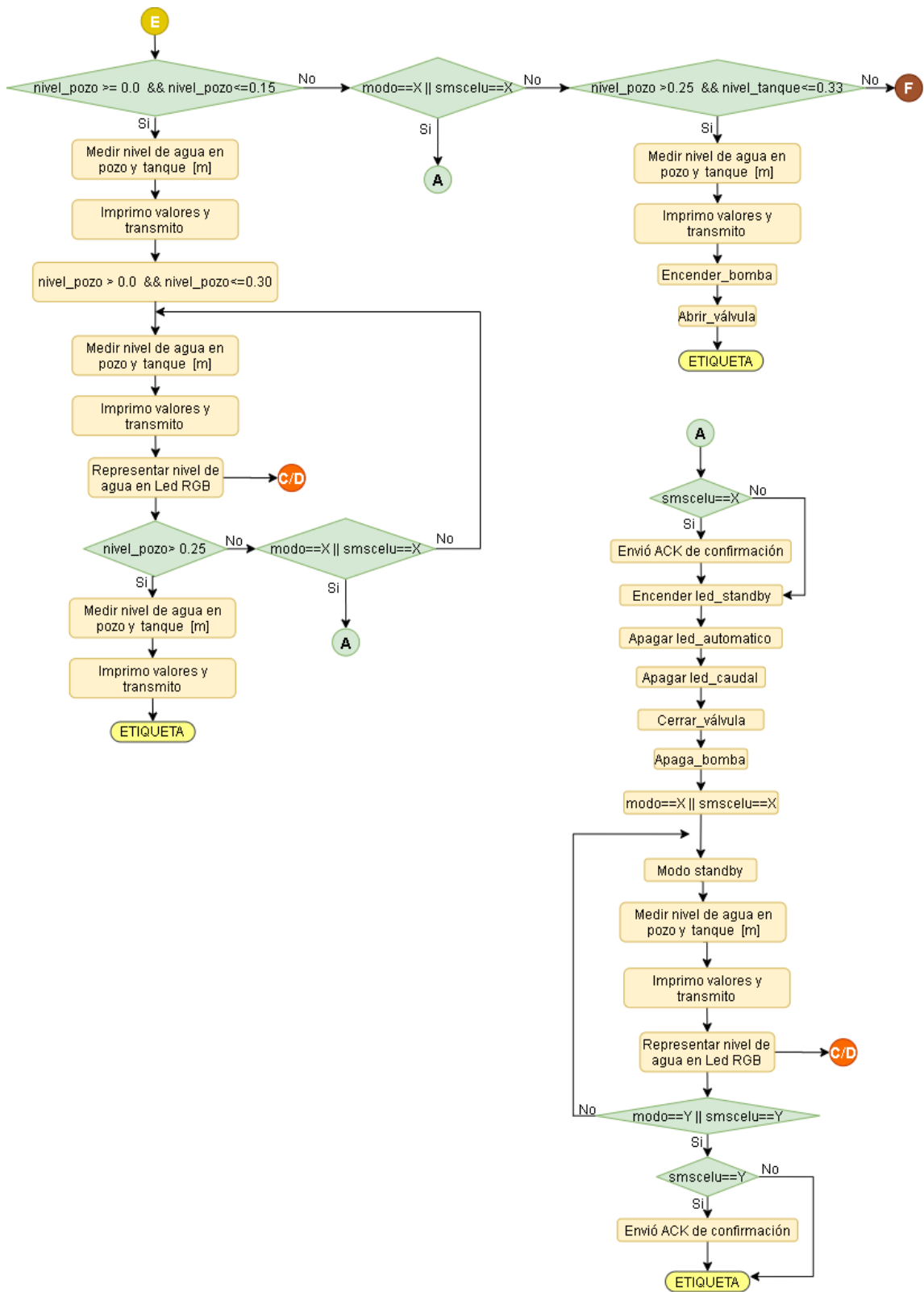


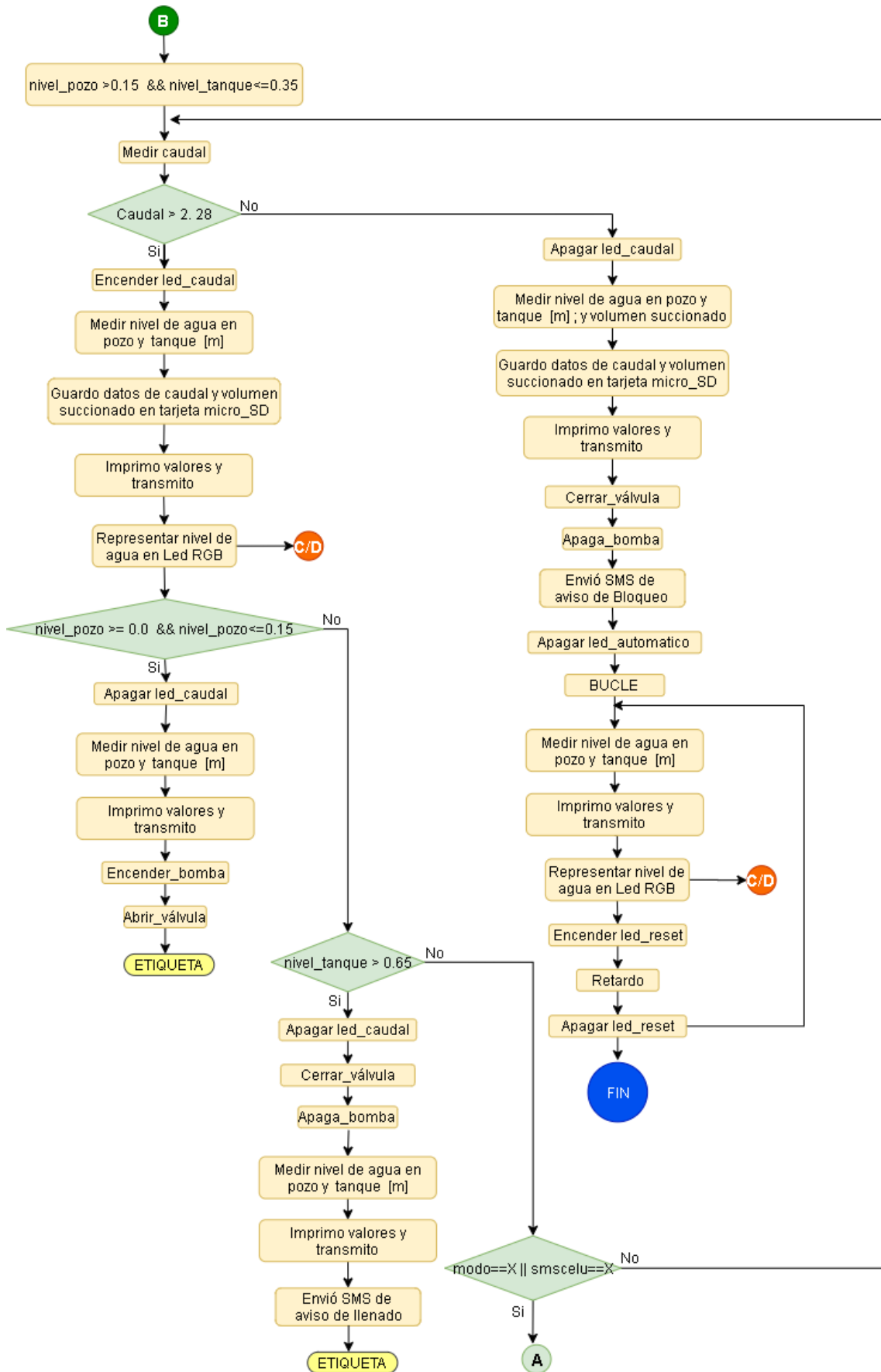
Figura 2.46 Interfaz de programación de MIT App Inventor

2.5.2 FLUJOGRAMA DE AUTOMATIZACIÓN DE BOMBEO DE AGUA

Para la programación del microcontrolador ATmega 2560 de Arduino y de los sensores escogidos en el aparato anterior (selección de hardware) se realizó un diagrama de flujo donde se explica sobre el procedimiento, algoritmos y la secuencia de las operaciones del sistema de bombeo de agua. En la figura 2.47 se exhibe el flujograma del sistema automático de bombeo de agua, el cual ayudará a conseguir un funcionamiento eficiente y óptimo.







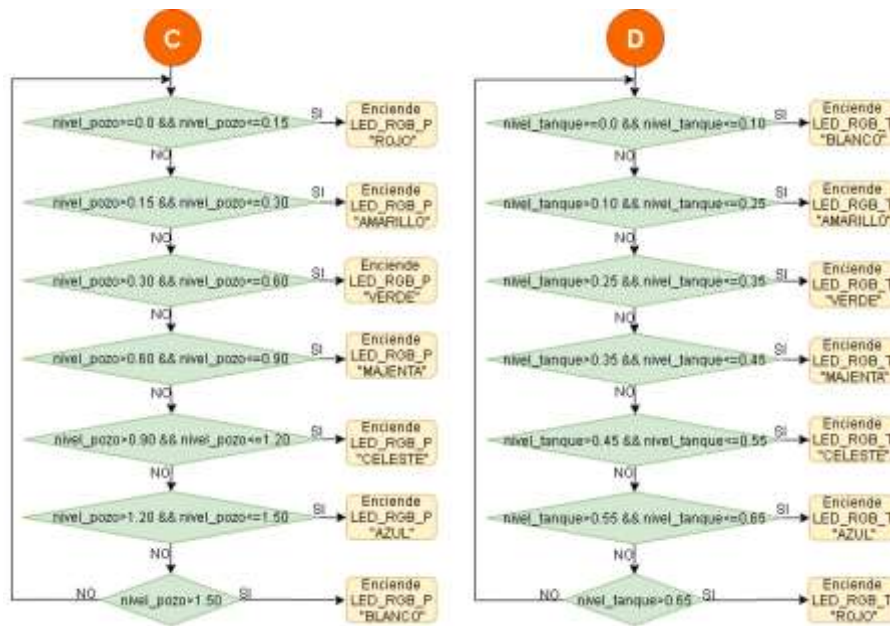


Figura 2.47 Flujograma de automatización para bombeo de agua

2.5.3 PROGRAMACIÓN DEL MICROCONTROLADOR ATMEL

A continuación, se realizará el código de programación en la plataforma de Arduino, que está estructurado de la siguiente forma: librerías, variables globales y funciones de manera ordenada y entendible. El lector podrá dirigirse al ANEXO C para poder apreciar el código de programación de manera completa.

2.5.3.1 Declaración de librerías

Una librería es una suma de funciones que están orientadas para un propósito específico y facilitar la programación del código. En el código 2.1 se exhibe las librerías utilizadas para la correspondiente programación de los siguientes módulos: adaptador de la tarjeta micro SD, shield GSM/GPRS SIM900 y servo.

```
#include <SPI.h>           // Incluye libreria interfaz SPI
#include <SD.h>            // Incluye libreria para tarjetas SD
#include <SoftwareSerial.h> // Incluye libreria para la SIM900
#include <Servo.h>        // Incluye libreria para controlar el servo
```

Código 2.1 Declaración de librerías en Arduino

2.5.3.2 Declaración de variables globales

A continuación, se enseña el código para la declaración de las variables globales dentro de la programación.

En el código 2.2 se exhibe la exposición de las variables para la utilidad del módulo adaptador de la tarjeta micro SD, así como del servo que servirá para abrir y/o cerrar la

válvula esférica, del módulo YF-S201 que mide el caudal y volumen extraído, de los sensores HC-SR04 que miden los niveles de agua, de los led RGB catado común que muestra por colores los niveles del agua, el módulo relé que acciona o desactiva la bomba PKM60-1, además se dispone de otros led que indican los estados de funcionamiento del sistema y finalmente se muestra la variables de configuración que almacena los caracteres que serán transmitidos ya sea por la App o por mensajes de texto.

```
#define CSpin 53      //Define al pin 53 como Chip Select
File archivo;       //Declaro el objeto archivo del tipo File
Servo servoMotor;   //Declaro la variable para controlar el servo
int pinservo=8;     //Variable del tipo entero asignada a la señal PWM del servo
int pincaudal = 2;  //Variable del tipo entero asignada al pin 2
volatile int Numpulsos = 0; //Variable que almacena el número de pulsos
unsigned long pulsosAcumulados = 0; //Variable que almacena el número de pulsos
//acumulados
int triggerp=22;    //Variable asignada al pin TRIG del sensor HC-SR04
int echop=23;       //Variable asignada al pin ECHO del sensor HC-SR04
int trigert=24;     //Variable asignada al pin TRIG del sensor HC-SR04
int echot=25;       //Variable asignada al pin ECHO del sensor HC-SR04
int pinredp = 40;   //Variable asignada al pin RED del LED RGB
int pingreenp = 42; //Variable asignada al pin GREEN del LED RGB
int pinbluep = 44;  //Variable asignada al pin BLUE del LED RGB
int pinredt = 41;   //Variable asignada al pin RED del LED RGB
int pingreent= 43;  //Variable asignada al pin GREEN del LED RGB
int pinbluet = 45;  //Variable asignada al pin BLUE del LED RGB
int rele=26;        //Variable asignada al pin IN del MODULO RELE
int led_caudal=3;   //Variable que indica la cantidad de flujo
int led_sd = 4;     //Variable que indica el estado de la tarjeta micro_SD
int led_standby=5;  //Variable que indica el modo de operacion STANDBY
int led_automatico=6; //Variable que indica el modo de operacion AUTOMATICO
int led_reset=7;    //Variable que indica cuando el sistema esta BLOQUEADO
char modo;          //Variable que almacena el caracter enviado desde la APP
char smscelu;       //Variable que almacena el caracter enviado por SMS de text
```

Código 2.2 Variables globales de configuración.

2.5.3.3 Configuración del puerto serial del módulo GSM/GPRS SIM900

En el código 2.3 se describe la configuración del puerto serial del módulo GSM/GPRS SIM900, que está configurado por software y sirve para la comunicación entre el Mega 2560 y la red celular por medio de mensajes de texto.

```
SoftwareSerial SIM900(10, 11); // Declaracion del puerto Serial para SIM900 (RX,TX)
```

Código 2.3 Declaración del puerto serial del módulo SIM900

2.5.3.4 Configuración de la velocidad de los puertos seriales

Para la comunicación entre el prototipo y la aplicación diseñada en App Inventor se dispone de un puerto serial específico, así como para la comunicación con la red celular, mismos que disponen de una velocidad expresado en baudios. En el código 2.4 se describe la configuración de los puertos y su velocidad de transmisión.

```
SIM900.begin(19200); //Comunicacion puerto serial del modulo SIM900
Serial1.begin(9600); //Comunicacion puerto serial del modulo HC-05
Serial.begin(9600); //Comunicacion puerto serial para el monitor serial
```

Código 2.4 Configuración de velocidad de los puertos serial

2.5.3.5 Configuración del módulo GSM/GPRS SIM90

En el código 2.5 se exhibe el programa de inicialización del módulo GSM/GPRS SIM900 para la transmisión y recepción de los caracteres en forma de mensajes de texto, la configuración de este módulo se ha realizado mediante comandos AT.

```
delay(15000); //Retardo que demora para buscar la red celular
SIM900.print("AT+CMGF=1\r"); //Configuracion del puerto serie SIM900 en modo
//mensajes de texto
delay(200); //Retardo
SIM900.print("AT+CNMI=2,2,0,0,0\r"); //Configuracion para que todos los datos
//recibidos en el puerto serie SIM900 sean leidos
delay(200); //retardo
```

Código 2.5 Configuración de inicio del módulo GSM/GPRS SIM900

2.5.3.6 Configuración del adaptador de la tarjeta Micro_SD

En el código 2.6 se obtiene el programa de verificación del adaptador de la tarjeta micro_SD, donde si la tarjeta micro_SD está insertada de manera incorrectamente o simplemente no existe la tarjeta micro_SD en el puerto del adaptador el programa no compilará y entrará en un bucle infinito del tipo while.

```
if(!SD.begin(CSpin)){ //Inicializacion de tarjeta SD
  digitalWrite(led_reset,HIGH); //Pin led_reset estado encendido
  while(1){ //Bucle infinito
    digitalWrite(led_sd,HIGH); //Pin led_sd estado encendido
    delay(1000); //Retardo de 1 segundo
    digitalWrite(led_sd,LOW); //Pin led_sd estado apagado
    delay(1000); //Retardo de 1 segundo
  }
}
```

Código 2.6 Configuración inicial del módulo Micro_SD

2.5.3.7 Funciones de programación

Las funciones son porciones de código que son escritas de tal manera que al realizar el programa general tenga una mejor estructura de programación. Las funciones detalladas a continuación esta escritas de tal forma que al momento de realizar el código principal de la función *void loop* sea mucho más compacto y fácil de interpretar para el usuario.

A. Función de programación del sensor HC-SR04

A continuación, se presenta la función que sirve para medir el nivel de agua, donde se puede escribir la altura a la cual va a estar colocado el sensor HC-SR04 desde un nivel de referencia tal como se describe en el código 2.7.

```
float sensor_pozo(float posicion_sensor_pozo){           //Funcion para medir nivel de agua del pozo
digitalWrite(trigerp,LOW);                             //Asegurar que este apagado
delayMicroseconds(2);                                  //Espero 2 microsegundo
digitalWrite(trigerp,HIGH);                             //Disparar un impulso sonoro
delayMicroseconds(10);                                  //Duracion del impulso sonoro de almenos 10 us
digitalWrite(trigerp,LOW);                              //Evitar interferencia al momento de leer el ECHO
long tiempo=pulseIn(echop,HIGH);                       //Tiempo en que demora la onda sonora en ir y volver
float nivelp=posicion_sensor_pozo-(tiempop*0.00017015); //Nivel=3.50-((tiempop/2)*340.3*(1/1000000))[m]
return nivelp;                                         //Retorna el valor del nivel de agua
```

Código 2.7 Función para medir el nivel de agua

B. Función de programación led RGB

La siguiente función (código 2.8) describe la combinación de los colores RGB que sirven para encender un color específico dependiendo del nivel de agua.

```
void colorp (int rojop,int verdep,int azulp){           //Funcion para combinar colores RGB del RGB del pozo
analogWrite(pinredp,rojop);                            //Guarda un valor entre 0-255 y muestra en pinredp
analogWrite(pingreenp,verdep);                        //Guarda un valor entre 0-255 y muestra en pingreenp
analogWrite(pinbluep,azulp);                          //Guarda un valor entre 0-255 y muestra en pinbluep
}
```

Código 2.8 Función para combinación de colores RGB

C. Función de programación de relé

El contactor o relé es el dispositivo electrónico que mediante un pulso en bajo realiza el encendido de la bomba eléctrica y con un pulso en alto se realiza el proceso de apagado de la misma, en el código 2.9 se exhibe la programación del proceso descrito en donde el relé está configurado en el modo normalmente abierto.

```
void enciende_bomba(){                                 //Funcion para encender la bomba
digitalWrite(rele,LOW);                                //Envia un pulso en bajo
}
void apaga_bomba(){                                   //Funcion para apagar la bomba
digitalWrite(rele,HIGH);                               //Envia un pulso en alto
}
```


Código 2.9 Función para encendido/apagado de la bomba

D. Función de programación servomotor

Para la apertura y cierre de la válvula esférica se procede a realizar un giro del servomotor entre 0° y 90° respectivamente, tal como se exhibe en el código 2.10.

```
void abrir_valvula(){ //Funcion para abrir la valvula esferica
  servoMotor.write(0); //Realiza un giro de 0°
}
void cerrar_valvula(){ //Funcion para cerrar la valvula esferica
  servoMotor.write(90); //Realiza un giro de 90°
}
```

Código 2.10 Función para abrir y cerrar válvula esférica

E. Función de programación del sensor YF-S201

Dentro de las funciones utilizadas para el código del programa se tiene el código 2.11, que sirve para obtener la frecuencia con la que gira las aspas de la turbina del sensor de flujo YF-S201.

```
int frecuencia(){ //Funcion para calcular la frecuencia
  Numpulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(1000); //muestra de 1 segundo
  noInterrupts(); //Desabilitamos las interrupciones
  pulsosAcumulados +=Numpulsos; //Número de pulsos acumulados
  return pulsosAcumulados; //Retorna el numero de pulsos acumulados
}
```

Código 2.11 Función para obtener la frecuencia del módulo YF-S201

F. Función de programa para realizar envío de SMS de texto

A continuación, en el código 2.12 se explica la función para el envío de SMS de texto como respuesta al bloqueo del sistema.

```
void mensaje(){ //Funcion para enviar un SMS cuando sistema se bloquea
  SIM900.print("AT+CMGF=1\r"); //Configuracion en modo SMS de texto
  delay (4000);
  SIM900.println("AT+CMGS=\0980364494\"); //Numero telefonico que receptara del SMS
  delay(4000);
  SIM900.println("SISTEMA BLOQUEADO, REVISE EL SISTEMA Y PRESIONE BOTON RESET");//Contenido que se envia en el SMS
  delay(1000);
  SIM900.println((char)26); //El simbolo 26 es un caracter ASCII de control que en arduino
  //corresponde a CTRL+Z, con lo que el modulo sabe que el sms termino
  delay(8000); //Tiempo necesario para transmitir el SMS de texto
}
```

Código 2.12 Función que realiza envío de SMS de texto

2.5.3.8 Grabación de datos en tarjeta Micro_SD

Para tener un registro continuo de datos o lecturas de caudal y cantidad de agua succionada del sistema de bombeo por cada vez que se active la bomba PKM60-1, en las

siguientes líneas de programación (código 2.13) se exhibe el programa para tal proceso mencionado.

```
archivo = SD.open("DATOS.txt", FILE_WRITE);           //Crea el archivo en .txt
                                                    //y da privilegios de solo de lectura
archivo.print(caudal);                               //Guardo valores de caudal en archivo DATOS
archivo.print(" ");                                  //Simbolo de separacion
archivo.println(volumen);                            //Guardo valores de volumen en archivo DATOS
archivo.close();                                     //Cierro el archivo
```

Código 2.13 Programa que realiza la grabación de datos

2.5.3.9 Transmisión de datos

Para el proceso de transmisión de los datos adquiridos por los sensores del sistema de bombeo se realiza por medio del módulo HC-05. El programa que realiza este proceso esta descrito en el código 2.14, mismo que está escrito de manera que al ser receptados en la App Móvil estos datos sean separados de manera ordenada y sea fácil de interpretar por el código de la App móvil y mostrar en la interfaz gráfica diseñada.

```
Serial1.print(nivel_pozo);                          //Imprimo el valor de nivel del pozo
Serial1.print(",");                                  //Simbolo de separacion
Serial1.print(nivel_tanque);                         //Imprimo el valor de nivel del tanque
Serial1.print(",");                                  //Simbolo de separacion
Serial1.print(caudal);                               //Imprimo el valor de nivel del caudal
Serial1.print(",");                                  //Simbolo de separacion
Serial1.print(volumen);                              //Imprimo el valor de volumen succionado
Serial1.print("\n");                                 //Salto de linea
```

Código 2.14 Programa para transmisión de datos por medio del módulo HC-05

2.5.4 FLUJOGRAMA DE PROGRAMACIÓN DE LA APP MÓVIL

Para la elaboración del código de programación de la App Móvil está realizada en bloques que es característico la plataforma APP Inventor, y para lo cual se dispone de un flujograma (Figura 2.48) en el que se describe el algoritmo y el funcionamiento del mismo; y que está en función del alcance (10 m) del módulo HC-05, además se anexa la opción de control del sistema por medio de SMS de texto en caso de superar la distancia mencionada.

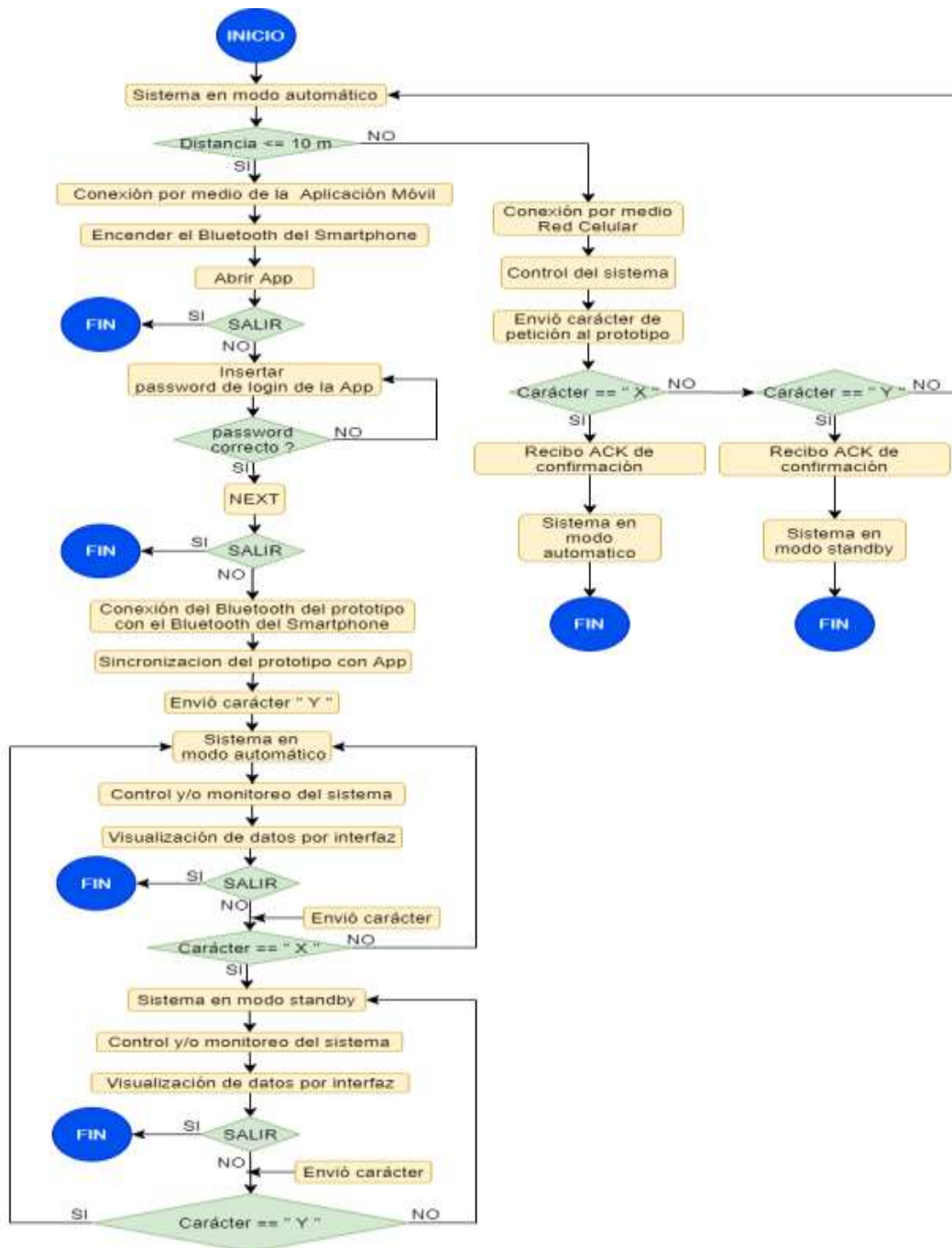


Figura 2.48 Flujograma de programación para la App Móvil

2.5.5 PROGRAMACIÓN EN APP INVENTOR

En el siguiente aparato se procede a explicar el código para el diseño de la App Móvil, el cual sirve para controlar y monitorear el sistema.

La siguiente App Móvil desarrollada consta de un *screen* principal, que contiene siete *sub-screen* las cuales están diseñadas a partir de los *Horizontal Arrangement*, *Vertical Arrangement* y componentes no visibles que se mencionan a continuación.

2.5.5.1 Componentes de App Inventor

Dentro de la plataforma App Inventor hay varios componentes entre los cuales se mencionan algunos de ellos que se utilizaron para el desarrollo de la App Móvil.

- ✓ *Notifier* permite mostrar avisos y/o alertas que al ejecutar alguna acción dentro de la App Móvil sean visualizadas por el usuario.
- ✓ *Clock*, dispara un temporizador a intervalos establecidos, para el desarrollo de la App Móvil se utilizó tres *Clock*, uno para la animación inicial de la aplicación y los dos restantes para la transmisión y recepción de datos del sistema de bombeo.
- ✓ *Activity Starter* permite ejecutar desde la App Móvil desarrollada la activación del dispositivo Bluetooth del smartphone.
- ✓ *Bluetooth Client* admite ejecutar la conexión entre el smartphone y el módulo HC-05 para la comunicación inalámbrica y la correspondiente Tx y Rx de datos.

En la figura 2.49 se exhibe los componentes utilizados en el desarrollo de la App Móvil.



Figura 2.49 Componentes utilizados para el desarrollo de la App Móvil

2.5.5.2 Código de programación de la App Móvil

La programación de la App Móvil se ha realizado de la siguiente manera: consta de un interfaz de inicio, interfaz de validación de contraseña, interfaz de conexión con dispositivo

Bluetooth e interfaz de lectura de datos. El lector podrá dirigirse al ANEXO D para revisar a detalle y de manera completa el código implementado.

A. Interfaz de inicio.

En el código 2.15 se describe la programación en bloques que ayuda a dar una animación de carga a la interfaz principal de inicio tal como se exhibe en la figura 2.50.



Código 2.15 Programa de animación de interfaz de inicio.



Figura 2.50 Interfaz de inicio de App Móvil.

B. Interfaz de validación de contraseña.

En esta interfaz se desplazará un campo donde se deberá inserta la contraseña “Tesis2020” tal como se exhibe en la figura 2.51, y para lograr esta acción dentro de la App Móvil se ha realizado el código 2.16 que describe el proceso de validación de contraseña.



Figura 2.51 Interfaz de validación de contraseña

```

cuando BTN_NEXT_DE_PANTALLA_CONTRASEÑA - .Clic
ejecutar
  si
  entonces
    poner CampoDeContraseña1 - .Texto - como " Tesis2020 "
    poner PANTALLA_TITULO - .Visible - como falso -
    poner PANTALLA_CONTRASENA - .Visible - como falso -
    poner PANTALLA_CONEXION - .Visible - como cierto -
    poner BTN_DESCONEXION_BT - .Visible - como falso -
    poner PANTALLA_AUTO - .Visible - como cierto -
    poner BOMBA_LLAVES_OFF_AUTO - .Visible - como falso -
    poner BOMBA_LLAVES_ON_AUTO - .Visible - como falso -
    poner ETI_SUCCIONANDO - .Visible - como falso -
    poner ETI_TANQUE_LLENO - .Visible - como falso -
    poner ETI_BLOQUEO - .Visible - como falso -
  sino
    llamar Notificador1 - .MostrarAlerta
      aviso " CONTRASENA INCORRECTA "
    poner CampoDeContraseña1 - .Texto - como " "
    poner PANTALLA_TITULO - .Visible - como cierto -
    poner PANTALLA_CONTRASENA - .Visible - como cierto -
  
```

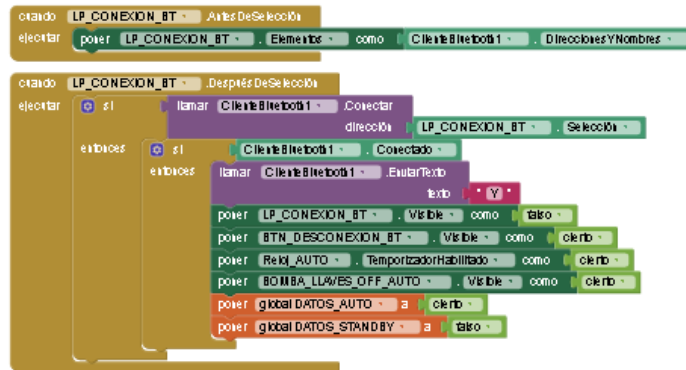
Código 2.16 Programa de validación de contraseña.

C. Interfaz para la conexión Bluetooth.

En la figura 2.52 se exhibe la interfaz gráfica que consta con un botón “CONECTAR DISPOSITIVO” que al presionar sirve para realizar una búsqueda de los dispositivos bluetooth que están visibles, entre los cuales tenemos el módulo HC-05, para dicho proceso se realizó el programa descrito en el código 2.17 donde primero se asegura la conexión Bluetooth para luego transmitir el carácter “Y” para que el sistema siempre inicie funcionando en modo automático por cada vez que se inicie la App Móvil.



Figura 2.52 Interfaz para conexión Bluetooth.



Código 2.17 Programa para la conexión Bluetooth desde la App Móvil.

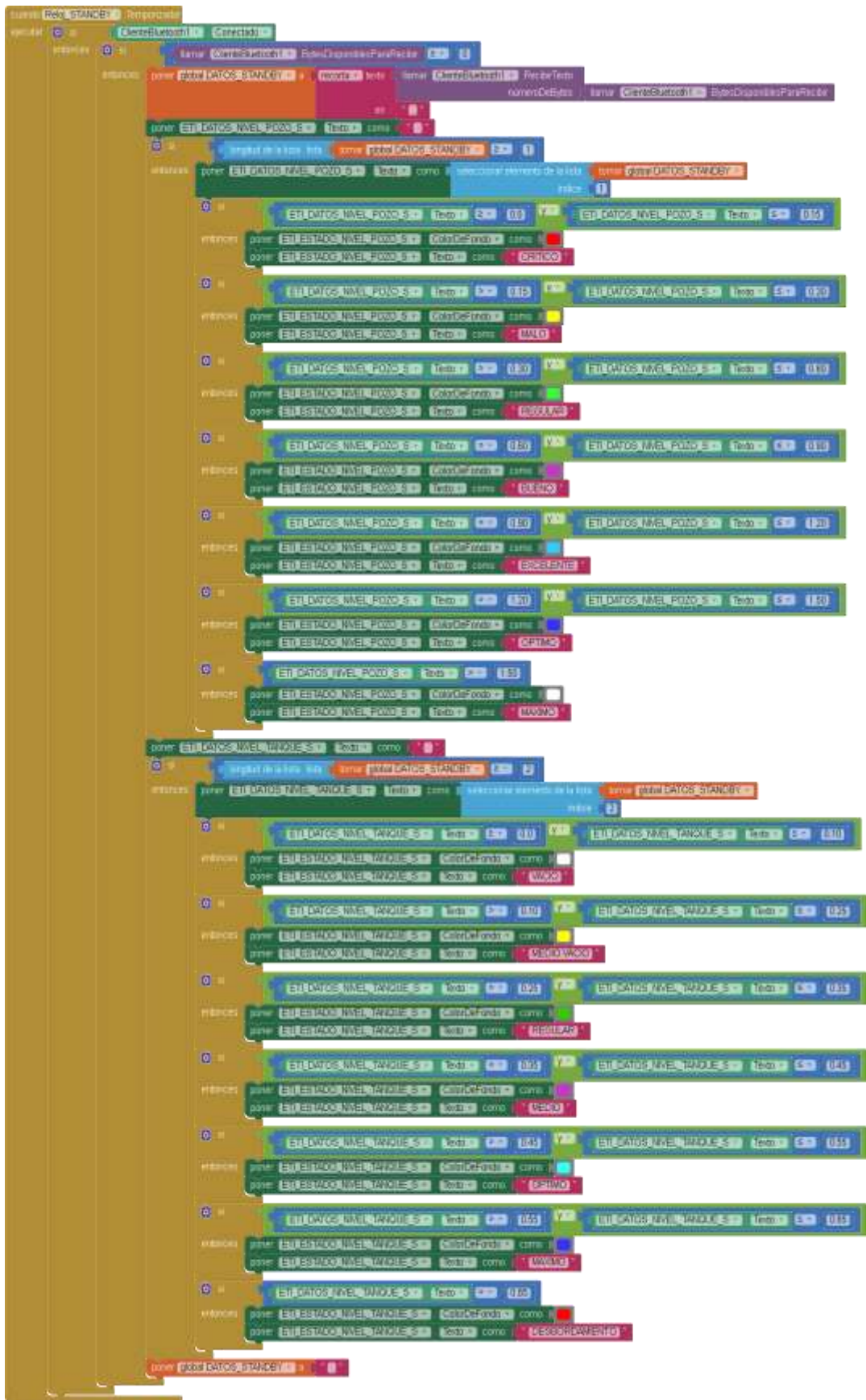
D. Interfaz de lectura de datos.

Para la visualización de los datos del sistema de succión se dispone de la interfaz gráfica que se exhibe en la figura 2.53, misma que se presenta cuando el sistema está operando en modo standby, adicional existe un botón llamado “AUTOMATICO” el cual sirve para cambiar de modo de operación.



Figura 2.53 Interfaz de visualización de datos en modo standby.

Adicional se exhibe el programa en el código 2.18 que ejecuta dichas funciones para el modo de operación standby.



Código 2.18 Programa para la lectura de los datos en modo standby.

3. RESULTADOS Y DISCUSIÓN

En esta sección se procede a realizar las respectivas pruebas de funcionamiento del prototipo tales como inicialización, conectividad, modos de operación y alarmas.

Además, se evaluará los resultados obtenidos del proceso de pruebas y finalmente se realizará un presupuesto aproximado de costos.

3.1 PRUEBAS DE FUNCIONAMIENTO

Una vez ensamblado el prototipo se procede a la instalación correspondiente del mismo como se exhibe en la figura 3.1 para las realizar las respectivas pruebas de funcionamiento y verificación del mismo.



Figura 3.1 Instalación del prototipo

Como se observa, este sistema de extracción o bombeo de agua consta de una bomba de agua, accesorios, tubería, módulos, sensores, y entre otros. El sistema se encuentra instalado dentro de una cámara seca que sirve para proteger de la lluvia a todos los elementos electrónicos y eléctricos.

3.1.1 INICIALIZACIÓN DEL SISTEMA

Para una inicialización correcta del prototipo, se considera que se debe tener insertada de manera adecuada una tarjeta micro_SD, caso contrario el sistema iniciara tomando datos incorrectos de nivel de agua tanto del pozo como del tanque de almacenamiento tal como se puede observar en la figura 3.2, además en esta tarjeta se pretende guardar los datos

de caudal y volumen de agua extraído por cada vez que el sistema realice el proceso de succión de agua del pozo.



Figura 3.2 Inicialización del sistema sin tarjeta micro_SD

Al mismo tiempo se puede observar que al iniciar de manera incorrecta el funcionamiento del prototipo debido a la ausencia de la tarjeta micro_SD se encenderá dos diodos leds que alertan la ausencia de la tarjeta micro_SD y la inicialización incorrecta del sistema.

Asimismo, se recalca que una vez que el sistema inicie el funcionamiento, este iniciará operando en modo automático por defecto, este proceso de inicialización del sistema tiene un retardo de alrededor de 00:27:18 segundos como se observa en la figura 3.3, este tiempo se da justamente porque dentro del código del microcontrolador se le asigna un tiempo para que el Shield GSM/GPRS SIM900 pueda buscar la red celular y conectarse a la misma, además dependiendo de los niveles de agua que dispongan el pozo y el tanque de almacenamiento el microcontrolador ATmega 2560 enviará la señal para encender la bomba PKM60-1 para realizar el proceso de succión del agua, caso contrario solo se mantendrá en modo de lectura de los niveles.



Figura 3.3 Retardo de inicialización del sistema de bombeo de agua.

3.1.2 CONECTIVIDAD DEL SISTEMA

A continuación, se presenta las respectivas pruebas de control y/o monitoreo de acuerdo con la tecnología utilizada.

3.1.2.1 Conectividad con la tecnología bluetooth

Por medio de la Aplicación Móvil se logró realizar una comunicación inalámbrica Bluetooth de corto alcance entre el Smartphone y el prototipo para el control y/o monitoreo del sistema, el tiempo de respuesta en la ejecución de la aplicación móvil tiene un pequeño retardo en el orden de los 00:04.97 segundos aproximadamente tal como se muestra en la figura 3.4 lo cual es aceptable para el emparejamiento con el módulo HC-05 del prototipo.

Además, en la figura 3.4 se exhibe como el smartphone ejecuta una búsqueda de todos los dispositivos Bluetooth encendidos que están dentro de su alcance, siendo el de interés el dispositivo nombrado como “CONTROL_MONITOREO” que ayudará a la comunicación entre el prototipo y la App móvil desarrollada para la transmisión y recepción de datos.



Figura 3.4 Retardo de emparejamiento de la comunicación Bluetooth

Y de la misma manera se procede a medir el tiempo o retardo que tiene el sistema cuando se procede a realizar los cambios de modo de operación, teniendo así los siguientes datos: un retardo de 00:00:73 segundos para el cambio de operación de automático a standby y de manera viceversa un retardo de 00:01:58 segundos tal como se observa en la figura 3.5.

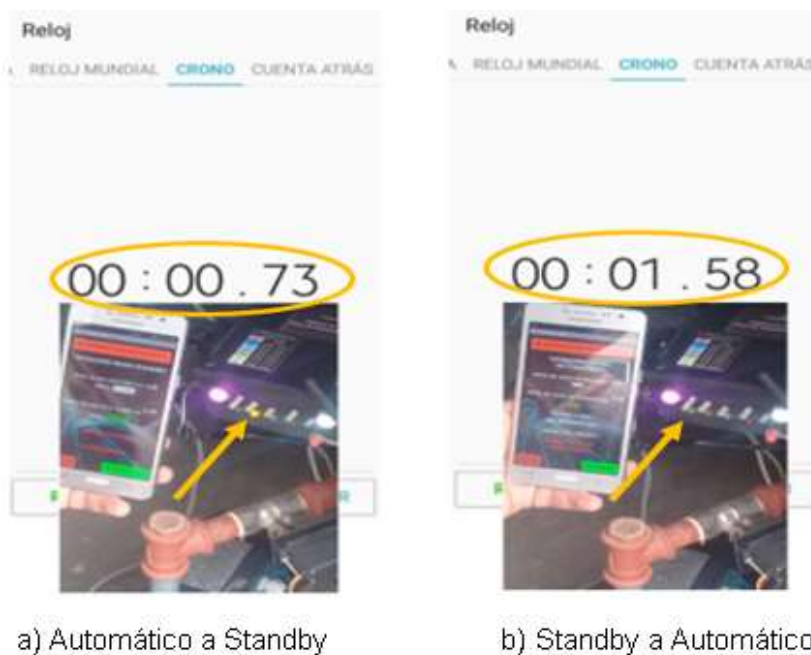


Figura 3.5 Retardo de cambios de modo de operación por la App Móvil

3.1.2.2 Conectividad con la tecnología celular

Por medio de mensaje de texto se logra una comunicación inalámbrica celular de corta y/o larga distancia para el control del sistema como se indica en la figura 3.6, donde se muestra la transmisión o envió del carácter en forma de mensaje de texto y además se observa otro mensaje de texto como respuesta a la confirmación de haber realizado el cambio correcto dentro del funcionamiento del prototipo.

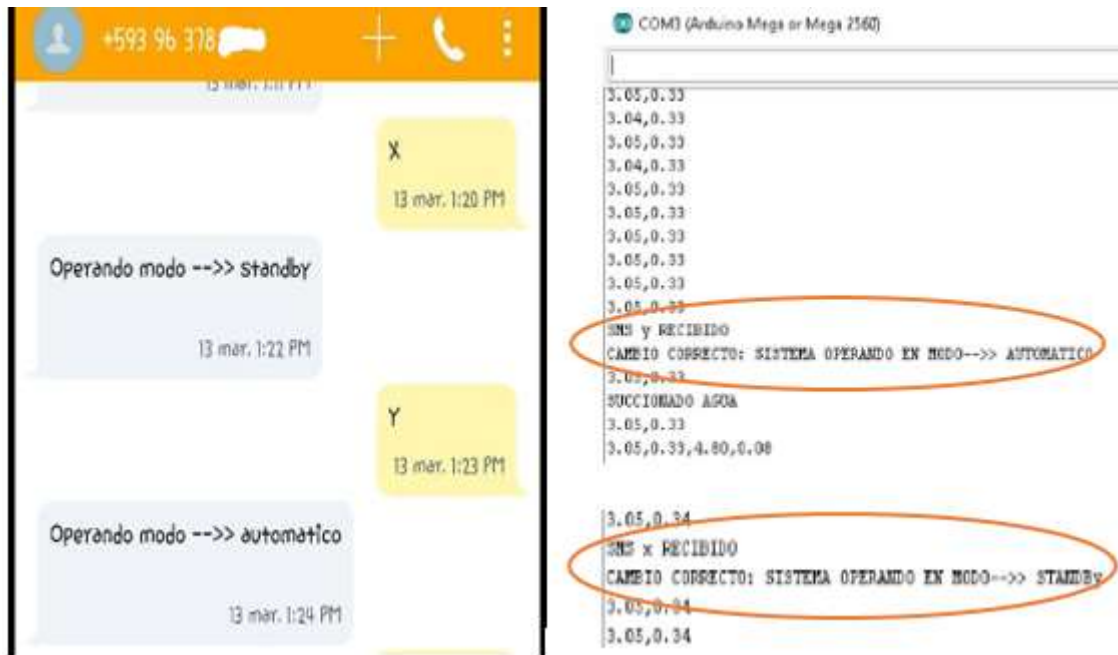


Figura 3.6 Control del sistema mediante red celular

Además, se muestra en la figura 3.7 que existe un retardo de 01:11:49 minutos para un cambio de modo de operación de automático a standby, y de igual forma se tiene un retardo de 01:04:50 minutos para el cambio de standby a automático, estos datos de retardos se tienen porque el sistema está instalado en un sitio donde por la infraestructura de la vivienda no existe una buena cobertura de la red celular y también por la arquitectura misma de la red celular.

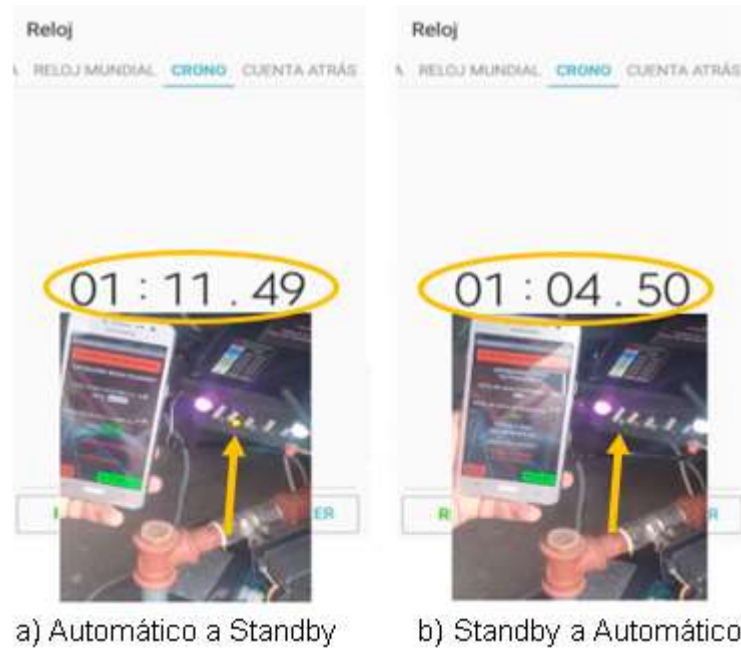


Figura 3.7 Retardo de cambios de modo de operación por SMS de texto

3.1.3 MODO DE OPERACIÓN DEL SISTEMA

El funcionamiento del prototipo dispone de dos modos de operación: el modo de standby y modo automático. A continuación, se muestra las pruebas respectivas.

3.1.3.1 Modo de operación standby

Este modo de operación del sistema es exclusivamente para lectura de niveles de agua, donde se puede realizar el proceso de mantenimiento y limpieza del pozo y del tanque de almacenamiento. En la figura 3.8 se exhibe la interfaz de este modo de operación.



Figura 3.8 Operación en modo standby

Para poder salir o cambiar este modo de operación se tiene varias maneras entre las cuales se mencionan las siguientes: cuando se presiona el botón denominado AUTOMATICO que sirve para cambiar el modo de operación o cuando se presiona el botón SALIR que sirve para cerrar la aplicación móvil o cuando se presiona el botón RESET que está instalado en el prototipo tal como se indica en la figura 3.9 o simplemente cuando se desconecta de la fuente de alimentación.



Figura 3.9 Reinicio del sistema al presionar el botón Reset.

3.1.3.2 Modo de operación automático

En este modo de operación del prototipo, el sistema funciona de manera automática dando como beneficio un ahorro del tiempo del usuario, un llenado automático y un nivel adecuado de agua en el tanque de almacenamiento, además en este modo de operación se puede observar los niveles y caudal que atraviesa por la tubería en la sección de descarga una vez que el sistema comienza el proceso de succión. En la figura 3.10 se aprecia este modo de operación del sistema donde se observa que está atravesando un caudal de 11.33 (L/min) en la tubería de descarga y también el volumen de agua extraído de 1429.50 (L).



Figura 3.10 Operación en modo automático

Además, en este modo de operación se tienen varios estados críticos del sistema, por ejemplo cuando el nivel del pozo está en su nivel mínimo de agua de 0.15 (m) y el tanque de almacenamiento está por debajo del nivel de promedio de 0,35 (m) como se puede observar en la figura 3.11, en este caso el sistema entra en el modo de espera hasta cuando el pozo logre por lo menos llegar a obtener un nivel de 0.25 (m) con lo cual se obtiene una cantidad de 154 litros de agua que se pueden extraer.

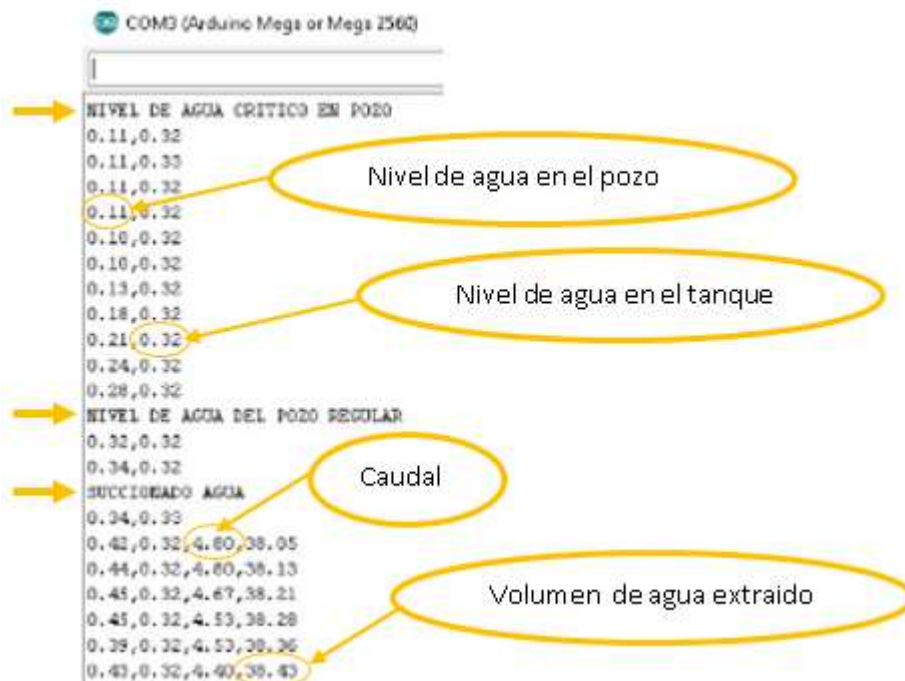


Figura 3.11 Nivel de agua crítico en el pozo

Además, se tiene otro punto crítico de análisis cuando el tanque de almacenamiento llega al nivel de 0,35 (m), que es cuando procede a realizar el proceso de succión. Para realizar este proceso de manera exitosa es importante setear dentro del código del microcontrolador un nivel de 0,33 (m) para asegurarse que la lectura de los datos nivel sea la adecuada para realizar la succión del agua del pozo como se observa en la figura 3.12.



Figura 3.12 Nivel límite de agua en el tanque para proceso de succión.

La estrategia de setear dos valores de nivel de agua en el tanque de almacenamiento es para evitar falsas lecturas debido a los retardos de 1 segundo de los sensores ultrasónico, el sistema tiende a procesar los datos de nivel con pequeños errores, ocasionando que el sistema se active para la succión y de manera seguida se apague el mismo, por tal situación dentro del programa se procede a setear un valor de nivel de 0,33 (m) como nivel mínimo en el tanque de almacenamiento para realizar la succión.

Y finalmente se tiene la variable caudal que es otro valor importante a analizar, como tal, se tiene un valor de 2.28 (L/min) que es el valor mínimo para realizar el proceso de succión sin problema caso contrario el sistema entrara en estado de bloqueo para salvaguardar al sistema y especial la bomba de agua para que no realice la succión en vacío tal como se muestra en la figura 3.13.

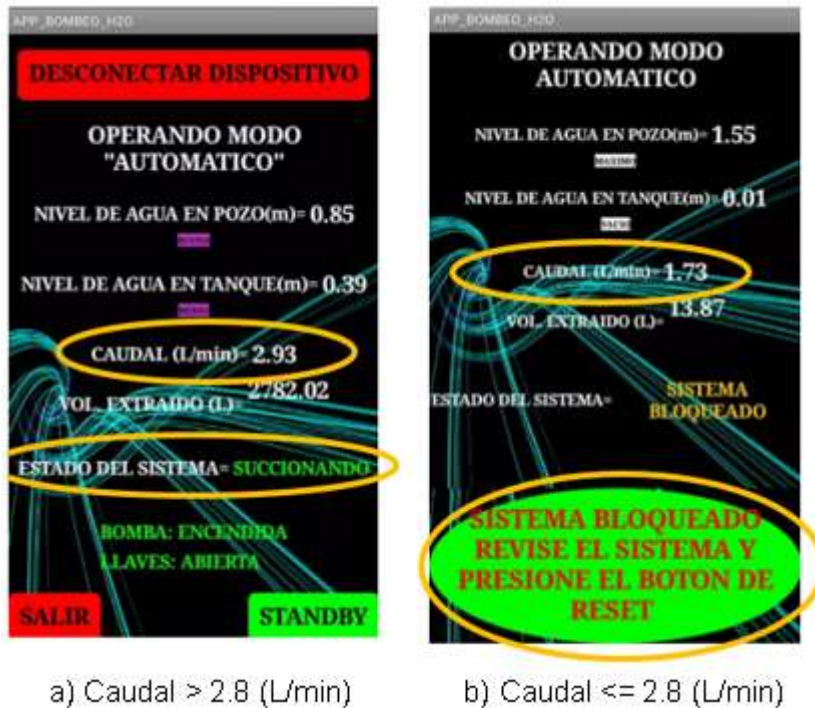
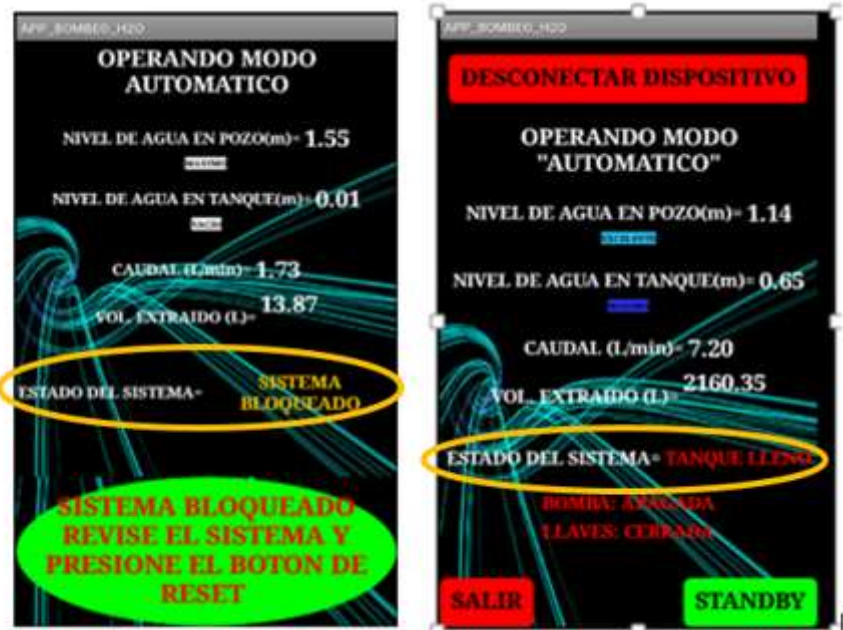


Figura 3.13 Valor de caudal mínimo

3.1.4 PRUEBAS DEL SISTEMA DE ALARMAS

Las pruebas de funcionamiento de las alarmas del prototipo corresponden a las alertas o señalización que adopta el sistema como medio de protección frente a un acontecimiento de un evento dentro del control y monitoreo de este.

En la figura 3.14 se indica las correspondientes pruebas y verificación de las alarmas, cuando el sistema entra en estado de bloqueo al tener un caudal inferior al caudal mínimo calculado (2.28 L/min), en la aplicación móvil se mostrara un botón de color verde con una notificación "SISTEMA BLOQUEADO, REVISE EL SISTEMA Y PRESIONE EL BOTÓN RESET" y de igual forma sucederá cuando el tanque de almacenamiento este en su nivel máximo, se enviara una notificación por medio de un mensaje de texto diciendo "TANQUE LLENO A UN NIVEL DE 0.65 (m)".



a) Advertencia de sistema en bloqueo b) Advertencia de tanque lleno

Figura 3.14 Alertas del sistema sometido a pruebas de funcionamiento

De igual manera se puede observar en la figura 3.15 dos diodos leds RGB que indica el nivel de agua del pozo y del tanque de almacenamiento; además existe otros diodos leds que son indicadores del estado del sistema tales como el modo de operación, el caudal, el estado de bloqueo y la inserción de la tarjeta micro_SD.



Figura 3.15 Leds indicadores de alertas y alarmas.

3.2 ANÁLISIS DE LOS RESULTADOS

Una vez culminado con las pruebas correspondientes del sistema, se expone que:

Existe una diferencia en la toma de datos de la variable física (nivel) por parte de los sensores ultrasónicos, esta diferencia es porque en el sketch realizado en la plataforma Arduino está desarrollado de tal forma que las medidas de nivel que imprime el programa es en centímetros y no en milímetros como se indica en el apartado de pruebas, con lo cual al comparar las medidas tomadas por los sensores ultrasónicos y un flexómetro tiene una pequeña discrepancia, ya que con el flexómetro se puede leer los niveles de agua ya sea en milímetros, centímetros o metros, cosa que no sucede con los sensores ultrasónicos como se indica en la figura 3.16.



Figura 3.16 Toma de valores de nivel de agua

En la tabla 3.1 se muestra los valores tomados con el flexómetro, así como con los sensores ultrasónicos, se puede observar que existe unas pequeñas diferencias que no afecta al funcionamiento del prototipo en gran magnitud en vista que es un sistema constante es decir, que la misma cantidad de agua que succiona del pozo es la misma cantidad de agua que descarga en el tanque de almacenamiento por el principio de la continuidad.

Tabla 3.1 Medidas de niveles de agua

# de muestra	Tanque de almacenamiento		Pozo	
	Flexómetro (cm)	Sensor HC-SR04 (cm)	Flexómetro (cm)	Sensor HC-SR04 (cm)
1	11	10	20,1	18
2	13,4	14	25,5	23
3	15,5	15	34,2	35
4	18,3	16	46,5	48
5	24,3	23	52,9	54
6	32,5	34	61,2	63
7	33,2	35	78,7	76
8	46,4	45	123,3	125
9	57,7	58	158,5	155
10	64,5	65	180,6	178

Respecto a la conectividad mediante la tecnología bluetooth para la transmisión y recepción de datos se tiene un pequeño retardo tal como se muestra en la tabla 3.2, esto sucede porque el programa está compilando y para ejecutar la siguiente instrucción enviada desde la aplicación móvil desarrollada, primero debe terminar compilando la subrutina que se está ejecutando.

Tabla 3.2 Retardos de cambio de operación por medio de la App móvil

# de muestra	Cambio de modo de operación de:	
	Automático a Standby (s)	Standby a Automático (s)
1	00:00:73	00:01:58
2	00:00:70	00:01:45
3	00:00:75	00:01:62
4	00:00:80	00:01:58
5	00:00:69	00:01:49
6	00:00:71	00:01:61
7	00:00:71	00:01:51
8	00:00:73	00:01:55
9	00:00:74	00:01:61
10	00:00:70	00:01:58

Como se puede observar se tiene dos rangos de valores de retardo, el primer rango es menor que el segundo, esto sucede por la diferencia de la carga computacional o líneas de código que debe procesar el microcontrolador para lograr salir del bucle y ejecutar la instrucción que se la realiza.

De igual forma se realiza un análisis de los retardos que se obtiene al controlar y/o monitorear al sistema por medio de la red celular, como se observa en la tabla 3.3 se nota que se tiene retardos muy altos en comparación a la tecnología bluetooth, como se mencionó esto sucede por el sitio donde está ubicado el sistema y también por la

arquitectura mismo de la red celular, pero, como el sistema es constante no afecta en el funcionamiento el retardo que se menciona.

Tabla 3.3 Retardos de cambio de operación por medio de SMS de texto

# de muestra	Cambio de modo de operación de:	
	Automático a Standby (s)	Standby a Automático (s)
1	01:11:49	01:04:50
2	01:40:09	00:58:51
3	02:05:06	01:04:50
4	01:55:51	02:01:04
5	02:06:58	01:00:45

Respecto al funcionamiento de los cambios de los modos de operación del sistema son correctos y adecuados dependiendo del carácter "X" o "Y" que se le envié ya sea desde la App Móvil o por medio de SMS de texto, pero como se mencionó anteriormente en cada caso tiene sus retardos ya sea por medio de la tecnología bluetooth o por medio de la red celular.

Y por último se analiza el sistema de alarmas o alertas, estas alarmas responden de manera eficiente y correcta dependiendo de los eventos o cambio que tenga el prototipo, como por ejemplo los cambios de modo de operación, el caudal, el estado de bloqueo y la inserción de la tarjeta micro_SD.

Se puede decir que el sistema funciona de manera correcta siempre y cuando se tenga en consideración que es un sistema constante donde el retardo no es crítico para el funcionamiento del mismo.

3.3 COSTOS REFERENCIALES

Para obtener un precio total estimado del proyecto se ha dividido en varias secciones como se explica a continuación.

Se presenta en la tabla 3.1 un resumen de los dispositivos electrónicos seleccionados para la implementación del sistema automático de bombeo de agua, especificando el modelo, cantidad y precio incluido IVA.

Tabla 3.4 Lista de elementos electrónicos y precio.

Elemento	Modelo	Cantidad	Precio unitario (\$)	Precio (\$)
Arduino	Mega 2560	1	27	27
Módulo Bluetooth	HC-05	1	10	10
Shield GSM/GPRS	SIM900	1	66	66
Sim card chip	Tuenti	1	5	5
Módulo micro SD		1	7,28	7,28
Tarjeta Micro SD.	Kingston clase 10	1	16,80	16,80
Relé de un canal	SRD – 5VDC-SL-C	1	4,68	4,68
Servomotor	MG996R	1	15	15
Sensor Ultrasónico	HC-SR04	2	4,5	9
Caudalímetro	YF-S201	1	12	12
Led RGB	Cátodo común	2	0,50	1
Diodo Led	Rojo	5	0,20	1,00
Adaptador AC/DC	WY-003	1	5,00	5,00
Elementos electrónicos adicionales				5,00
TOTAL				184,76

Además, en la tabla 3.2 se exhibe los accesorios útiles para la construcción del sistema hidráulico, tanto en el lado de la succión como en el lado de la descarga junto con su respectivo precio incluido IVA.

Tabla 3.5 Lista de accesorios del sistema hidráulico y precio.

Accesorio	Medida	Cantidad	Precio unitario (\$)	Precio (\$)
Succión				
Válvula de pie con rejilla	1"	1	13,79	13,79
Codos de 90°	1"	1	1,60	1,60
Unión de rosca	1"	1	1,55	1,55
Unión universal	1"	1	4,40	4,40
Tubería	1"	8,58 (m)	20	40
Descarga				
Tubería	½"	11,95 [m]	15	15
Codos de 90°	½"	9	0,40	3,6
Neplo	½"	10 cm	0,55	0,55
Reducción brusco	De 1" a ½"	1	2,50	2,50
Unión con rosca	½"	3	0,55	1,65
Unión universal	½"	1	1,05	1,05
Te	½"	1	0,70	0,70
Válvula esférica	½"	1	11,56	11,56
Bomba de agua PKM60-1		1	70	70
Varios				1,00
TOTAL				168,95

Adicional se presenta en la tabla 3.3 costos correspondientes por mano de obra para el diseño y construcción del sistema de automatización de extracción de agua, así como otros accesorios necesarios para la implementación.

Tabla 3.6 Costos adicionales para la construcción del prototipo.

Descripción	Cantidad	Precio unitario (\$)	Precio (\$)
Mano de obra y construcción	1	400	400
Diseño y asesoramiento	1	500	500
Cable UTP	20 (m)	0,35	7
Tubo PVC de 3"	3 (m)	15	15
Caja platica	1	4,5	4,5
Acoplamiento mecánico servo - válvula esférica	1	10	10
Plan de saldo mensual	1	5	5
Total			941,5

Una vez analizado los costos de los elementos electrónicos así como de los accesorios, mano de obra y entre otros, que conlleva a la implementación del prototipo se determinó su costo total de \$1295,21.

4. CONCLUSIONES Y RECOMENDACIONES.

Finalmente se presenta conclusiones y recomendación que se obtuvieron una vez culminado el prototipo entre las cuales se mencionan las siguientes:

4.1 CONCLUSIONES

- Existe un retardo de alrededor de 1 o 2 minutos en recibir el carácter para el cambio de modo de operación del sistema desde el momento que fue enviado, esto se debe a que la ubicación del prototipo está en una sitio donde por cuestiones de construcción está en una área donde la cobertura de la red celular es inestable.
- El sistema de alertas que está configurado e instalado en el prototipo tiene como finalidad facilitar el manejo y operatividad del mismo, dando la disposición al usuario de saber cuándo sucedió un evento dentro del funcionamiento del sistema.
- El sistema de alarmas o avisos que envía el prototipo por medio de la red celular está configurado de manera que el usuario obtenga un ahorro en su plan de saldo,

como se mencionó el sistema emitirá un SMS de respuesta cuando sea controlado por medio de SMS de texto, exista un bloqueo o cuando el tanque de almacenamiento llegue a su nivel máximo de 0.65 (m).

- El programa realizado en la plataforma Arduino está diseñado de manera estratégica para que sea fácil de entender, modificar o escalar en caso que se desee añadir o quitar alguna función dentro de la actividad del prototipo.
- La plataforma App Inventor se presta de manera eficiente para realizar una diversidad de aplicaciones en el sistema operativo Android, sin embargo es indispensable mantener un valor de temporizador sincronizado con el módulo HC-05 para la transición y recepción de datos en la aplicación móvil, ya que si no está configurado de manera correcta puede ocasionar que el buffer de la App se llene y los datos a mostrar no sean los que se transmiten en ese momento.
- El prototipo diseñado es óptimo y eficiente para trabajo en cualquier sistema de bombeo que se desee automatizar presto que está diseñado con sensores no invasivos (HC-SR04) y lo único que se requiere para su acoplamiento a otros sistema de bombeo es el tamaño y tipo de bomba que depende de la profundidad del pozo.
- La operación del prototipo está planteado de manera que al momento que comience el funcionamiento siempre inicie operando en el modo automático, con el fin de evitar la necesidad de conectarnos por Bluetooth o por la red celular para la inicialización del sistema.
- La implementación de este proyecto sirve para optimizar el tiempo del usuario, ya que es un sistema automático donde cada vez que el tanque de reserva está a la mitad de su capacidad (0.35 m), este automáticamente succiona el agua del pozo, manteniendo un nivel de agua adecuado en el tanque de almacenamiento para que el usuario haga uso del mismo sin tener contratiempo en tener que sacar el agua de manera manual.
- Si bien el nivel de agua que existe en el pozo y en el tanque son datos sumamente importantes, no se tendría ningún problema si existe una pequeña variación al momento de tomar los datos con el sensor HC-SR04, puesto que el vaciado del pozo así como el llenado del tanque es de manera lineal y al no existir perturbación en el proceso, el sistema funcionaría de manera eficiente y correcta.

4.2 RECOMENDACIONES

- Se recomienda realizar la programación del microcontrolador por medio de funciones para una mejor ejecución, lectura, compresión y modificación del mismo para aplicaciones futuras, pudiendo tener un programa escalable.
- Para el control y/o monitoreo del sistema se recomienda utilizar la app móvil desarrollada si se encuentra a distancia considerable (10 m) del sistema ya que no demanda recursos (plan de saldo) para la utilización de la misma.
- Una vez que el sistema se encuentre bloqueado es necesario que el usuario revise el nivel de agua que existe en el pozo así como en el tanque de almacenamiento, puesto que puede existir una posible avería en los sensores HC-SR04.
- Para tener un prototipo con sensores totalmente no invasivo se recomienda utilizar un caudalímetro ultrasónico el cual ayudaría a evitar cortes en la tubería.
- Para una mejor transmisión y recepción de información a través de la red celular se propone que la ubicación del sistema sea en un área donde exista buena cobertura.
- Si bien está normado la polaridad de los circuitos eléctricos, pues se recomienda tener presente dicha situación, ya que si la bomba no se conecta de manera correcta puede ocasionar graves problemas al circuito electrónico de control así como a la misma bomba.
- Cuando el sistema se encuentra bloqueado, el usuario deberá realizar solo el respectivo reset o reinicio del sistema, mas no desconectar el sistema de la red eléctrica ya que ocasionaría el apagado del módulo GSM/GPRS SIM900, esta recomendación se menciona con el fin de mantener disponible la comunicación inalámbrica de la red celular.
- Se recomienda verificar la inserción de la tarjeta micro_SD dentro del adaptador ya que si no está colocada de manera correcta o no existe la tarjeta micro_SD, el programa no compilara y por ende no realizara las respectivas lecturas de nivel de agua y el sistema entrara en bloqueo.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Á. Lema Carrera, Diseño e implementación de un sistema automático de purificación de Agua por medio de energía y reflexión solar y luz ultravioleta, Quito: Escuela Politécnica Nacional, 2015.
- [2] INEN, «Siete de cada diez hogares en Ecuador no realizan ninguna práctica de ahorro de agua,» 25 junio 2013. [En línea]. Available: <http://www.ecuadorencifras.gob.ec/siete-de-cada-diez-hogares-en-ecuador-no-realizan-ninguna-practica-de-ahorro-de-agua/>. [Último acceso: 27 enero 2019].
- [3] J. Ortiz Merchán, «EL UNIVERSO,» 20 junio 2018. [En línea]. Available: <https://www.eluniverso.com/opinion/2018/06/20/nota/6819557/acceso-consumo-agua-potable-todos>. [Último acceso: 30 enero 2019].
- [4] O. P. d. I. Salud, «GUÍAS PARA EL DISEÑO DE ESTACIONES DE BOMBEO DE AGUA POTABLE,» 2005. [En línea]. Available: <http://www.bvsde.paho.org/tecapro/documentos/agua/161esp-diseno-estbombeo.pdf>. [Último acceso: 11 Abril 2019].
- [5] SIAPA, «ACTUALIZACIÓN DE LOS CRITERIOS Y LINEAMIENTOS TÉCNICOS PARA FACTIBILIDADES EN LA Z.M.G.,» Febrero 2014. [En línea]. Available: http://www.siapa.gob.mx/sites/default/files/capitulo_13._estaciones_de_bombeo.pdf. [Último acceso: 2019 Abril 2019].
- [6] J. Wekker V, «SISTEMAS DE BOMBEO,» Junio 2004. [En línea]. Available: https://aducarte.weebly.com/uploads/5/1/2/7/5127290/sistema_hidroneumatico.pdf. [Último acceso: 4 Abril 2019].
- [7] A. Reyes Salinas, ESCURRIMIENTO DE FLUIDOS, Santiago de Chile: USACH, 2010.
- [8] F. M. White, Mecánica de Fluidos, España: McGraw-Hill, 2004.
- [9] C. Barrios Napurí, R. Torres Ruiz , T. C. Lampoglia y R. Agüero Pittman, «La Guía de Orientación en Saneamiento Básico para Alcaldías de Municipios Rurales y Pequeñas Comunidades,» 2009. [En línea]. Available: <http://www.bvsde.paho.org/bvsacg/guialcalde/2sas/2-2sas.htm#arriba>. [Último acceso: 4 Abril 2019].
- [10] M. P. Collazo Caraballo y J. Montaña Xavier, «Manual de Agua Subterránea,» Montevideo, 2012.
- [11] E. COMERCIO, «El río Chalpi Grande, una fuente para el norte de Quito,» 22 Marzo 2017. [En línea]. Available: <https://www.elcomercio.com/actualidad/riochalpigrande-aguapotable-quito-abastecimiento-servicio.html>. [Último acceso: 15 Abril 2019].
- [12] P. Duran Escamilla, «Captación de agua de lluvia, alternativa sustentable,» [En línea]. Available:

<http://www.conama10.conama.org/conama10/download/files/CT%202010/41008.pdf>.
[Último acceso: 15 Abril 2019].

- [13] E. Gómez de León, ENSAYOS NO DESTRUCTIVOS ULTRASONICOS, Madrid: FUNDACIÓN CONFEMETAL, 2006.
- [14] J. M. Flores Perez, Medidor ultrasónico de nivel de líquidos en movimiento en un canal, Quito: Escuela Politécnica Nacional, 1990.
- [15] M. A. Bello U y M. T. Pino Q, «MEDICIÓN DE PRESIÓN Y CAUDAL,» 2000. [En línea]. Available: <http://biblioteca.inia.cl/medios/biblioteca/boletines/NR25635.pdf>. [Último acceso: 23 Abril 2019].
- [16] G. Ghelfi y J. Tello, Caudal, Puerto Cabello: Universidad Nacional Experimental Politecnica de la Fuerza Armada, 2013.
- [17] D. R. Ortega Segura, Implementación de un prototipo para el Monitoreo de dióxido de carbono (CO₂), monóxido de carbono (CO) y metano (CH₄) en lugares residenciales utilizando el sistema operativo Android y Arduino, Quito: Escuela Politécnica Nacional, 2017.
- [18] J.-S. Lee, Y.-W. Su y C.-C. Shen, «A Comparative Study of Wireless Protocols:Bluetooth, UWB, ZigBee, and Wi-Fi,» Noviembre 2007. [En línea]. Available: http://eee.guc.edu.eg/Announcements/Comparaitive_Wireless_Standards.pdf. [Último acceso: 30 Abril 2019].
- [19] E. Villa Téllez y L. A. Gama Moreno, «Aplicabilidad del estandar IEEE 802.15.1,» [En línea]. Available: <https://www.yumpu.com/es/document/view/41090784/bluetooth-aplicabilidad-del-estandar-ieee-802151>. [Último acceso: 30 Abril 2019].
- [20] M. C. Paredes, Escritor, *Bluetooth*. [Performance]. 2017.
- [21] J. R. Luque Giráldez, Modelado del retardo de transmisión en Bluetooth 2.0 +EDR, Málaga: Universidad de Málaga, Escuela Técnica Superior de Ingenieros de Telecomunicación, 2010.
- [22] J. Santamaría Velázquez, Control remoto usando Bluetooth, Sevilla: Universidad de Sevilla, Escuela Técnica Superior de Ingeniería, 2015.
- [23] L. Avila y C. Reyes, «Revisión estado del Arte de la tecnología Bluetooth,» [En línea]. Available: <http://publicaciones.unisimonbolivar.edu.co/rdigital/inovacioning/index.php/identific/article/viewFile/66/100>. [Último acceso: 3 Mayo 2019].
- [24] Muhyi Bin Yaakop, Izwan Arief Abd Malik, Zubir bin Suboh, Aizat Faiz Ramli y Mohd Azlan Abu, «Bluetooth 5.0 throughput comparison for internet of thing usability a survey,» IEEE, Malaysia, 2017.
- [25] Bluetooth SIG, «How AoA & AoD Changed the Direction of Bluetooth Location Services,» 27 Marzo 2019. [En línea]. Available:

<https://www.bluetooth.com/blog/new-aoa-aod-bluetooth-capabilities/>. [Último acceso: 3 Mayo 2019].

- [26] M. Inzaurrealde, J. Isi y J. Garderes , «TELEFONIA CELULAR,» 6 Junio 2014. [En línea]. Available: http://www.sistemamid.com/panel/uploads/biblioteca/2014-06-06_11-27-11104608.pdf. [Último acceso: 5 Mayo 2019].
- [27] E. Martínez, «La evolucion de la telefonía móvil,» 1 Mayo 2001. [En línea]. Available: <http://www.eveliux.com/mx/La-evolucion-de-la-telefonía-móvil.html>. [Último acceso: 5 Mayo 2019].
- [28] R. G. Perea Tamayo, *5G: UNA NUEVA GENERACIÓN PARA UNA NUEVA SOCIEDAD*, Peru: Dirección de Instrumentación Científica de la Agencia Espacial del Perú, 2014.
- [29] J. González y O. Salamanca, «EL CAMINO HACIA LA TECNOLOGÍA 5G,» Enero-Junio 2016. [En línea]. Available: <http://redalyc.org/articulo.oa?id=78445977002>. [Último acceso: 16 Mayo 2019].
- [30] T. GSM, «Ingeniería de sistemas,» 16 Abril 2012. [En línea]. Available: <http://eveningsistemas-u.blogspot.com/2012/04/el-sistema-global-para.html>. [Último acceso: 9 Mayo 2019].
- [31] L. Flores Barreto, «SlideShare,» 21 Junio 2012. [En línea]. Available: <https://es.slideshare.net/leobarflo/telefonía-móvil-2g-final>. [Último acceso: 9 Mayo 2019].
- [32] J. J. Jiménez, *Evolución e historia de la telefonía celular*, El Cid Editor, 2009.
- [33] M. Figueroa de la Cruz, *Introducción a los sistemas de telefonía celular*, Buenos Aires: Editorial Hispano Americana HASA, 2008.
- [34] W. L. McCabe, J. C. Smith y P. Harriott, *Operación unitarias en ingeniería química*, España: McGraw-Hill.
- [35] E. A. Ramírez Pozo, *ESTUDIO DE BOMBAS CENTRÍFUGAS Y SU FACTIBILIDAD DE APLICACIÓN EN LA FACULTAD DE INGENIERÍA MECÁNICA CON LA FINALIDAD DE OBTENER PARÁMETROS TÉCNICOS CON VARIACIÓN DE CAUDAL*, AMBATO: UNIVERSIDAD TÉCNICA DE AMBATO, 2012.
- [36] CRANE, *Flujo de Fluidos en válvulas Accesorios y Tuberías*, México: McGraw-Hill Interamericana, 1987.
- [37] R. L. Mott, *Mecánica de fluidos*, México: Pearson Educación, 2006.
- [38] S. I. Jiménez Jiménez, «Pérdidas de carga localizadas o en accesorios,» 7 Julio 2017. [En línea]. Available: https://www.hidraulicafacil.com/2017/07/perdida-de-carga-localizada-o-en.html?fbclid=IwAR1dOuFDAjsIEN_IsUO1BQV7YSDZnGW27zOd-4DjdHRPW56w3BuqYxzvUU8. [Último acceso: 15 Noviembre 2019].

- [39] I. Martín , R. Salcedo y R. Font, «MECÁNICA DE FLUIDOS Tema1. Flujo interno de fluidos incompresibles y compresibles,» 2011. [En línea]. Available: https://rua.ua.es/dspace/bitstream/10045/20299/1/tema1_Flujo%20interno.pdf. [Último acceso: 15 Noviembre 2019].
- [40] ARDUINO, «Arduino Products,» [En línea]. Available: <https://www.arduino.cc/en/Products/Compare>. [Último acceso: 2 Junio 2019].
- [41] ARDUINO, «ARDUINO MEGA 2560 REV3,» [En línea]. Available: <https://store.arduino.cc/usa/mega-2560-r3>. [Último acceso: 3 Junio 2019].
- [42] ITead Studio, «HC-05,» [En línea]. Available: <http://www.electronicaestudio.com/docs/istd016A.pdf>. [Último acceso: 5 Junio 2019].
- [43] SIM900, «The GSM/GPRS Module for M2M applications,» [En línea]. Available: <http://www.propox.com/download/docs/SIM900.pdf>. [Último acceso: 9 Junio 2019].
- [44] datalogger, «Micro SD Card Micro SDHC Mini TF Card Adapter Reader Module for Arduino,» [En línea]. Available: <http://datalogger.pbworks.com/w/file/attach/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf>. [Último acceso: 12 Junio 2019].
- [45] Naylamp, «Tutorial Arduino y memoria SD y micro SD.,» [En línea]. Available: https://naylampmechatronics.com/blog/38_Tutorial-Arduino-y-memoria-SD-y-micro-SD-.html. [Último acceso: 13 Junio 2019].
- [46] L. LLAMAS, «MANEJAR CARGAS DE MÁS DE 220V CON ARDUINO Y SALIDA POR RELÉ,» 23 Julio 2016. [En línea]. Available: <https://www.luisllamas.es/arduino-salida-rele/>. [Último acceso: 15 Junio 2019].
- [47] Circuit Basics, «SONGLE RELAY,» [En línea]. Available: <http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C-Datasheet.pdf>. [Último acceso: 17 Junio 2019].
- [48] L. LLAMAS, «CONTROLAR UN SERVO CON ARDUINO,» 6 Junio 2016. [En línea]. Available: <https://www.luisllamas.es/controlar-un-servo-con-arduino/>. [Último acceso: 18 Junio 2019].
- [49] towerpro, «MG996R Robot servo 180° Rotation,» [En línea]. Available: <http://www.towerpro.com.tw/product/mg995-robot-servo-180-rotation/>. [Último acceso: 18 Junio 2019].
- [50] Arduproject, «Medir distancia con Arduino y sensor de ultrasonidos HC-SR04,» 23 Noviembre 2018. [En línea]. Available: <http://arduproject.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>. [Último acceso: 19 Junio 2019].
- [51] ELEC Freaks, «Ultrasonic Ranging Module HC - SR04,» [En línea]. Available: <https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf>. [Último acceso: 20 Junio 2019].
- [52] L. LIAMAS, «MEDIR CAUDAL Y CONSUMO DE AGUA CON ARDUINO Y CAUDALÍMETRO,» 13 Diciembre 2016. [En línea]. Available:

- <https://www.luisllamas.es/caudal-consumo-de-agua-con-arduino-y-caudalimetro/>. [Último acceso: 21 Junio 2019].
- [53] NAYLAM, «Tutorial sensor de flujo de agua,» [En línea]. Available: https://naylampmechatronics.com/blog/47_tutorial-sensor-de-flujo-de-agua.html. [Último acceso: 21 Junio 2019].
- [54] MANTECH, «YF-S201,» [En línea]. Available: http://www.mantech.co.za/Datasheets/Products/YF-S201_SEA.pdf. [Último acceso: 21 Junio 2019].
- [55] M. Zaragoza Mulas, «LED RGB, JUGANDO CON LOS COLORES PRIMARIOS DE LA LUZ,» [En línea]. Available: <https://educarparaelcambio.com/arduino/reto-10-led-rgb-jugando-con-los-colores-primarios-de-la-luz/>. [Último acceso: 23 Junio 2019].
- [56] Kingbright, «Round Tri-Color LED,» 4 Abril 2019. [En línea]. Available: <https://www.kingbrightusa.com/images/catalog/SPEC/WP154A4SUREQBFZGC.pdf>. [Último acceso: 23 Junio 2019].
- [57] Samsung, «Samsung Galaxy J2 Prime SM-G532,» [En línea]. Available: <https://www.samsungservice.es/phones/galaxy-j2-prime.html>. [Último acceso: 24 Junio 2019].
- [58] S. Shadadpuri Goplani, Analisis, caracterizacion y calibracion de sensores de bajo coste para Arduino, San Cristóbal de La Laguna: Universidad de La Laguna, 2017.
- [59] Ó. Torrente Artero, Arduino. Curso práctico de formación, México: Alfaomega, 2013.
- [60] MIT, «APP INVENTOR,» 20 Abril 2015. [En línea]. Available: <http://ai2.appinventor.mit.edu/about/termservice.html>. [Último acceso: 2019 Mayo 28].
- [61] E. Martínez, «Métodos de medida a nivel,» 17 Junio 2010. [En línea]. Available: <http://www.interempresas.net/Medicion/Articulos/41365-Metodos-de-medida-a-nivel.html>. [Último acceso: 18 Abril 2019].
- [62] Honeywell, «What is Bluetooth Adaptive Frequency Hopping (AFH)?,» 10 Julio 2018. [En línea]. Available: <https://support.honeywellaidc.com/s/article/What-is-Bluetooth-Adaptive-Frequency-Hopping-AFH?>. [Último acceso: 1 Mayo 2019].
- [63] J. Wannstrom, «3GPP,» [En línea]. Available: <https://www.3gpp.org/technologies/keywords-acronyms/99-hspa>. [Último acceso: 14 Mayo 2019].
- [64] V. A. Guachán Morales y P. J. Cajias Castillo, Estudio de factibilidad y diseño de una red LTE (Long Term Evolution) en la zona norte de Quito -Ecuador, Quito: Universidad Politécnica Salesiana, 2014.

ANEXOS

ANEXO A: Diagrama de Moody

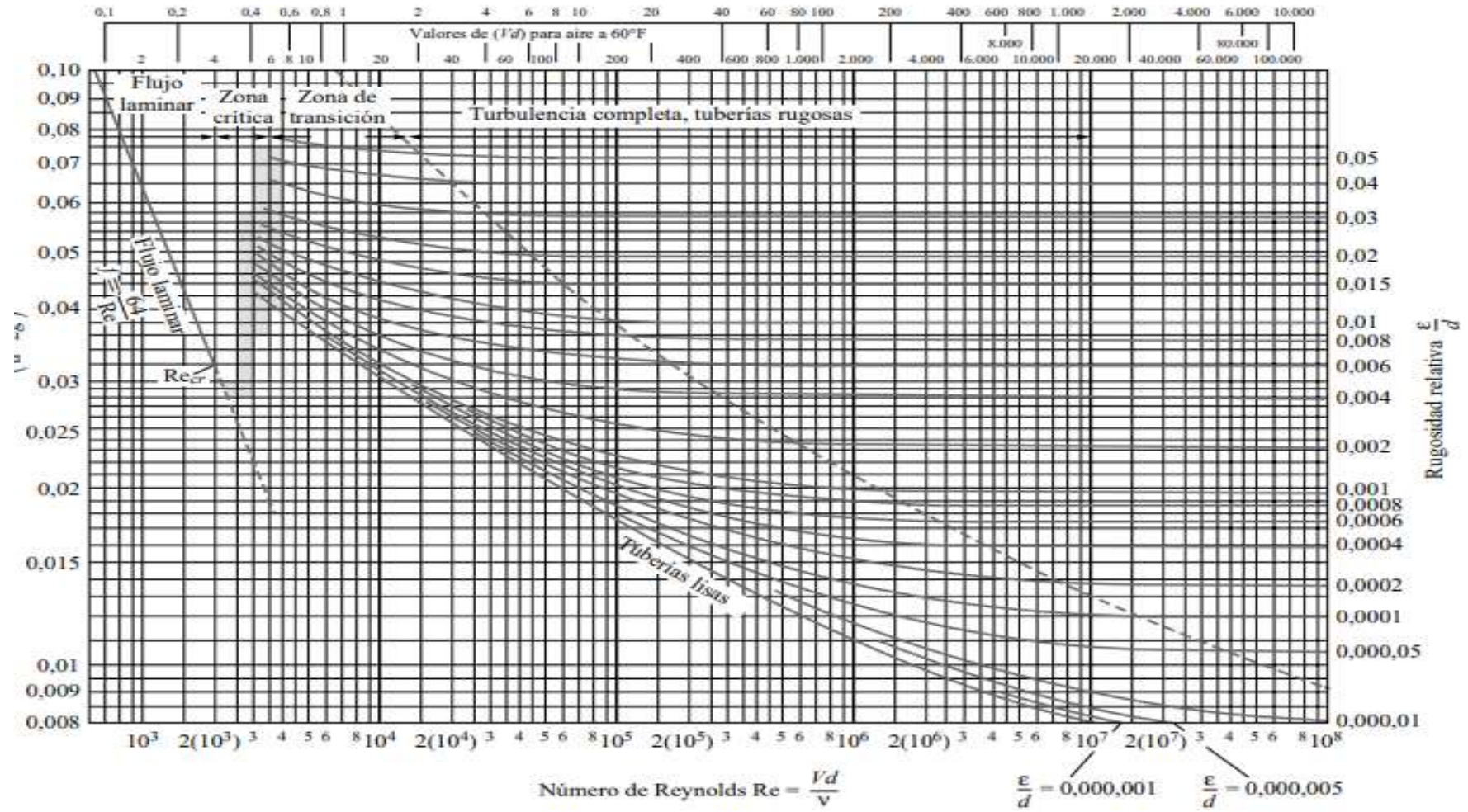
ANEXO B: Características de las placas de Arduino

ANEXO C: Código fuente del microcontrolador ATMEL 2560 de Arduino.

ANEXO D: Código fuente de la aplicación móvil implementada en App Inventor.

ANEXO A

Diagrama de Moody [8]



ANEXO B

Características de placas Arduino

Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
IO1	Intel® Curie	3.3 V / 7-12V	32MHz	6/0	14/4	-	24	196	Regular	-
Gemma	ATtiny85	3.3 V / 4-16 V	8 MHz	1/0	3/2	0.5	0.5	8	Micro	0
LilyPad	ATmega168V ATmega328P	2.7-5.5 V / 2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
LilyPad SimpleSnap	ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	4/0	9/4	1	2	32	-	-
LilyPad USB	ATmega32U4	3.3 V / 3.8-5 V	8 MHz	4/0	9/4	1	2.5	32	Micro	-
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
MKR1000	SAMD21 Cortex-M0+	3.3 V/ 5V	48MHz	7/1	8/4	-	32	256	Micro	1
Pro	ATmega168 ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	0.512 1	1 2	16 32	-	1
Pro Mini	ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	1	2	32	-	1
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
Zero	ATSAMD21G18	3.3 V / 7-12 V	48 MHz	6/1	14/10	-	32	256	2 Micro	2
Due	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Esplora	ATmega32U4	5 V / 7-12 V	16 MHz	-	-	1	2.5	32	Micro	-
Ethernet	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/4	1	2	32	Regular	-
Leonardo	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1
Yún	ATmega32U4 AR9331 Linux	5 V	16 MHz 400MHz	12/0	20/7	1	2.5 16MB	32 64MB	Micro	1
Arduino Robot	ATmega32u4	5 V	16 MHz	6/0	20/6	1 KB (ATmega32u4) 512 Kbit (I2C)	2.5 KB (ATmega32u4)	32 KB (ATmega32u4) of which 4 KB used by bootloader	1	1
MKRZero	SAMD21 Cortex-M0+ 32bit low power ARM MCU	3.3 V	48 MHz	7 (ADC 8/10/12 bit)/1 (DAC 10 bit)	22/12	No	32 KB	256 KB	1	1

ANEXO C

Código fuente del microcontrolador ATMEL 2560 de Arduino

```
/*
                                ESCUELA POLITÉCNICA NACIONAL
                                FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES
PROGRAMACION PARA LA IMPLEMENTACIÓN DE UN PROTOTIPO PARA EXTRACCIÓN DE AGUA DE UN POZO
MEDIANTE LA MEDICIÓN DE CAUDAL Y NIVEL DEL AGUA UTILIZANDO LA PLATAFORMA ARDUINO Y APLICACIÓN MÓVIL ANDROID

AUTOR: EDISON QUINGATUÑA
AÑO: 16/07/2020
CORREO: edison.quingatuna@epn.edu.ec
*/

//DECLARACION DE LIBRERIAS//
#include <SPI.h> //Incluye libreria interfaz SPI
#include <SD.h> //Incluye libreria para tarjetas SD
#include <SoftwareSerial.h> //Incluye libreria para la SIM900
#include <Servo.h> //Incluye libreria para controlar el servo
SoftwareSerial SIM900(10, 11); //Declaracion del puerto Serial para SIM900 (RX, TX)

//DECLARACION DE VARIABLES GLOBALES//
#define CSpin 53 //Define al pin 53 como Chip Select
File archivo; //Declaro el objeto archivo del tipo File
Servo servoMotor; //Declaro la variable para controlar el servo
int pinservo=8; //Variable del tipo entero asignada a la señal PWM del servo
int pincaudal = 2; //Variable del tipo entero asignada al pin 2
volatile int Numpulsos = 0; //Variable que almacena el número de pulsos
unsigned long pulsosAcumulados = 0; //Variable que almacena el número de pulsos acumulados
unsigned long tiempo; //Variable que almacena el tiempo
float triggerp=22; //Variable asignada al pin TRIG del sensor HC-SR04
float echop=23; //Variable asignada al pin ECHO del sensor HC-SR04
float trigert=24; //Variable asignada al pin TRIG del sensor HC-SR04
float echot=25; //Variable asignada al pin ECHO del sensor HC-SR04
int pinredp = 40; //Variable asignada al pin RED del LED RGB
int pingreenp = 42; //Variable asignada al pin GREEN del LED RGB
int pinbluep = 44; //Variable asignada al pin BLUE del LED RGB
int pinredt = 41; //Variable asignada al pin RED del LED RGB
int pingreent= 43; //Variable asignada al pin GREEN del LED RGB
int pinbluet = 45; //Variable asignada al pin BLUE del LED RGB
int rele=26; //Variable asignada al pin IN del MODULO RELE
int led_caudal=3; //Variable que indica la cantidad de flujo
int led_sd = 4; //Variable que indica el estado de la tarjeta micro_SD
int led_standby=5; //Variable que indica el modo de operacion STANDBY
int led_automatico=6; //Variable que indica el modo de operacion AUTOMATICO
int led_reset=7; //Variable que indica cuando el sistema esta BLOQUEADO
char modo; //Variable que almacena el caracter enviado desde la APP
char smscelu; //Variable que almacena el caracter enviado por SMS de texto

//CONFIGURACION INICIAL DEL SISTEMA//
void setup(){
  SIM900.begin(19200); //Comunicacion puerto serial del modulo SIM900
  Serial1.begin(9600); //Comunicacion puerto serial del modulo HC-05
  delay(15000); //Retardo que demora para buscar la red celular
  SIM900.print("AT+CMGF=1\r"); //Configuracion del puerto serie SIM900 en modo mensajes de texto
  delay(200); //Configuracion para que todos los datos recibidos
  SIM900.print("AT+CNMI=2,2,0,0,0\r"); //en el puerto serie SIM900 sean leidos
  delay(200);
  pinMode(led_sd, OUTPUT); //Pin configurado como salida
  digitalWrite(led_sd, LOW); //Pin estado apagado
  pinMode(led_caudal, OUTPUT); //Pin configurado como salida
  digitalWrite(led_caudal, LOW); //Pin estado apagado
  pinMode(led_standby, OUTPUT); //Pin configurado como salida
  digitalWrite(led_standby, LOW); //Pin estado apagado
}
```

```

pinMode(led_automatgico,OUTPUT); //Pin configurado como salida
digitalWrite(led_automatgico,LOW); //Pin estado apagado
pinMode(led_reset,OUTPUT); //Pin configurado como salida
digitalWrite(led_reset,LOW); //Pin estado apagado
pinMode(trigerp,OUTPUT); //Pin configurado como salida
pinMode(echop,INPUT); //Pin configurado como entrada
pinMode(trigert,OUTPUT); //Pin configurado como salida
pinMode(echot,INPUT); //Pin configurado como entrada
pinMode(pinredp,OUTPUT); //Pin configurado como salida
pinMode(pingreenp,OUTPUT); //Pin configurado como salida
pinMode(pinbluep,OUTPUT); //Pin configurado como salida
pinMode(pinredt,OUTPUT); //Pin configurado como salida
pinMode(pingreent,OUTPUT); //Pin configurado como salida
pinMode(pinbluet,OUTPUT); //Pin configurado como salida
pinMode(rele,OUTPUT); //Pin configurado como salida
digitalWrite(rele,HIGH); //Inicia apagado xq esta configurado N.0.
servoMotor.attach(pinservo); //Iniciamos el servo para que empiece a trabajar
pinMode(pinceudal,INPUT_PULLUP); //Pin configurado como entrada
attachInterrupt(digitalPinToInterrupt(pinceudal),flujo,RISING); //Interrupción INTO,
//llama a la ISR(Interrupt Service Routine) llamada "flujo"
//en cada flanco de subida en el pin digital 2

if(!SD.begin(CSpin)){ //Inicializacion de tarjeta SD
  digitalWrite(led_reset,HIGH); //Pin led_reset estado encendido
  while(1){ //Bucle infinito
    digitalWrite(led_sd,HIGH); //Pin led_sd estado encendido
    delay(1000); //Retardo de 1 segundo
    digitalWrite(led_sd,LOW); //Pin led_sd estado apagado
    delay(1000); //Retardo de 1 segundo
  }
}

//FUNCIONES//
float sensor_pozo(float posicion_sensor_pozo){ //Funcion para medir nivel de agua del pozo
  digitalWrite(trigerp,LOW); //Asegurar que este apagado
  delayMicroseconds(2); //Espero 2 microsegundo
  digitalWrite(trigerp,HIGH); //Disparar un impulso sonoro
  delayMicroseconds(10); //Duracion del impulso sonoro de almenos 10 us
  digitalWrite(trigerp,LOW); //Evitar interferencia al momento de leer el ECHO
  long tiempop=pulseIn(echop,HIGH); //Capturo el tiempo en que demora la onda sonora
  //en ir y volver asta que choca con el objeto
  //Nivel=Posicion del sensor-((tiempop/2)*340.3*(1/1000000))[m]
  float nivelp=posicion_sensor_pozo-(tiempop*0.00017015);
  return nivelp; //Retorna el valor del nivel de agua
}
float sensor_tanque(float posicion_sensor_tanque){ //Funcion para medir nivel de agua del tanque
  digitalWrite(trigert,LOW);
  delayMicroseconds(2);
  digitalWrite(trigert,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigert,LOW);
  long tiempot=pulseIn(echot,HIGH);
  float nivelt=posicion_sensor_tanque-(tiempot*0.00017015);
  return nivelt;
}

void colorp (int rojop,int verdep,int azulp){ //Funcion para combinar colores RGB del RGB del pozo
  analogWrite(pinredp,rojop); //Guarda un valor entre 0-255 y muestra en pinredp
  analogWrite(pingreenp,verdep); //Guarda un valor entre 0-255 y muestra en pingreenp
  analogWrite(pinbluep,azulp); //Guarda un valor entre 0-255 y muestra en pinbluep
}

```

```

void colort (int rojot,int verdet,int azult){ //Funcion para combinar colores RGB del RGB del Tanque
  analogWrite(pinredt,rojot);
  analogWrite(pingreent,verdet);
  analogWrite(pinbluet,azult);
}
void enciende_bomba(){ //Funcion para encender la bomba
  digitalWrite(rele,LOW); //Envia un pulso en bajo
}
void apaga_bomba(){ //Funcion para apagar la bomba
  digitalWrite(rele,HIGH); //Envia un pulso en alto
}
void abrir_valvula(){ //Funcion para abrir la valvula esferica
  servoMotor.write(0); //Realiza un giro de 0°
}
void cerrar_valvula(){ //Funcion para cerrar la valvula esferica
  servoMotor.write(90); //Realiza un giro de 90°
}
void flujo(){ //Funcion para capturar el numero de pulsos
  Numpulsos++; //Incrementa en una unidad el número de pulsos
}
int frecuencia(){ //Funcion para calcular la frecuencia
  Numpulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(1000); //muestra de 1 segundo
  noInterrupts(); //Desabilitamos las interrupciones
  pulsosAcumulados +=Numpulsos; //Número de pulsos acumulados
  return pulsosAcumulados; //Retorna el numero de pulsos acumulados
}
void mensaje_bloqueo(){ //Funcion para enviar un SMS cuando sistema se bloquea
  SIM900.print("AT+CMGF=1\r"); //Configuracion en modo SMS de texto
  delay (1000);
  SIM900.println("AT+CMGS=\"0980364494\""); //Numero telefonico que receptara del SMS
  delay(1000); //Contenido que se envia en el SMS
  SIM900.println("SISTEMA BLOQUEADO, REVISE EL SISTEMA Y PRESIONE BOTON RESET");
  delay(100);
  SIM900.println((char)26); //El simbolo 26 es un caracter ASCII de control que en arduino
  //corresponde a CTRL+Z, con lo que el modulo sabe que el sms term:
  delay(4000); //Tiempo necesario para transmitir el SMS de texto
}
void mensaje_llenado(){ //Funcion para enviar un SMS cuando el tanque esta lleno
  SIM900.print("AT+CMGF=1\r");
  delay (1000);
  SIM900.println("AT+CMGS=\"0980364494\"");
  delay(1000);
  SIM900.println("TANQUE LLENO A UN NIVEL DE 0.65(m)");
  delay(100);
  SIM900.println((char)26);
  delay(4000);
}
void mensaje_standby(){ //Funcion para enviar un SMS cuando el sistema cambia a modo standby
  SIM900.print("AT+CMGF=1\r");
  delay (1000);
  SIM900.println("AT+CMGS=\"0980364494\"");
  delay(1000);
  SIM900.println("CAMBIO CORRECTO: SISTEMA OPERANDO EN MODO-->> STANDBY");
  delay(100);
  SIM900.println((char)26);
  delay(4000);
}
}

```

```

void mensaje_automatico(){ //Funcion que envia un SMS cuando el sistema cambia a modo automatico
  SIM900.print("AT+CMGF=1\r");
  delay (1000);
  SIM900.println("AT+CMGS=\`0980364494\`");
  delay(1000);
  SIM900.println("CAMBIO CORRECTO: SISTEMA OPERANDO EN MODO-->> AUTOMATICO");
  delay(100);
  SIM900.println((char)26);
  delay(4000);
}

void STANDBY(){ //MODO DE OPERACION STANDBY DEL SISTEMA
  int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
  float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del poz
  float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del tar
  Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
  Serial1.print(","); //serial del modulo BT cada 1 segundo
  Serial1.print(nivel_tanque);
  Serial1.print("\n");
  if (nivel_pozo>=0.0 && nivel_pozo<=0.15) {colorp(255,0,0);} //NIVEL 1 del pozo= CRITICO(ROJO)
  if (nivel_pozo>0.15 && nivel_pozo<=0.30) {colorp(255,255,0);} //NIVEL 2 del pozo= MALO(AMARILLO)
  if (nivel_pozo>0.30 && nivel_pozo<=0.60) {colorp(0,255,0);} //NIVEL 3 del pozo= REGULAR(VERDE)
  if (nivel_pozo>0.60 && nivel_pozo<=0.90) {colorp(255,0,255);} //NIVEL 4 del pozo= BUENO(MAJENTA)
  if (nivel_pozo>0.90 && nivel_pozo<=1.20) {colorp(0,255,255);} //NIVEL 5 del pozo= EXCELENTE(CELESTE)
  if (nivel_pozo>1.20 && nivel_pozo<=1.50) {colorp(0,0,255);} //NIVEL 6 del pozo= OPTIMO(AZUL)
  if (nivel_pozo>1.50) {colorp(255,255,255);} //NIVEL 7 del pozo= MAXIMO(BLANCO)
  if (nivel_tanque>=0.0 && nivel_tanque<=0.10) {colort(255,255,255);} //NIVEL 1 del tanque= VACIO(BLANCO)
  if (nivel_tanque>0.10 && nivel_tanque<=0.25) {colort(255,255,0);} //NIVEL 2 del tanque= MALO(AMARILLO)
  if (nivel_tanque>0.25 && nivel_tanque<=0.35) {colort(0,255,0);} //NIVEL 3 del tanque= REGULAR(VERDE)
  if (nivel_tanque>0.35 && nivel_tanque<=0.45) {colort(255,0,255);} //NIVEL 4 del tanque= MEDIO(MAJENTA)
  if (nivel_tanque>0.45 && nivel_tanque<=0.55) {colort(0,255,255);} //NIVEL 5 del tanque= OPTIMO(CELESTE)
  if (nivel_tanque>0.55 && nivel_tanque<=0.65) {colort(0,0,255);} //NIVEL 6 del tanque= MAXIMO(AZUL)
  if (nivel_tanque>0.65) {colort(255,0,0);} //NIVEL 7 del tanque= DESBORDAMIENTO(ROJO)
}

void AUTOMATICO(){ //MODO DE OPERACION STANDBY DEL SISTEMA
  inicio; //Etiqueta para salir de bucle while
  int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
  float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del poz
  float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del tanc
  digitalWrite(led_standby,LOW); //Apago led standby
  digitalWrite(led_automatico,HIGH); //Enciendo led automatico
  Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
  Serial1.print(","); //serial del modulo BT cada 1 segundo
  Serial1.print(nivel_tanque);
  Serial1.print("\n");
  //CONDICION DE CONSUMO DE AGUA EN EL TANQUE O BAJO NIVEL DE AGUA EN POZO O DESBORAMIENTO DE AGUA EN EL TANQUE
  if ((nivel_tanque>0.35 && nivel_tanque<=0.65)||((nivel_pozo>=0.0 && nivel_pozo<=0.15) || nivel_tanque>0.65)){
  int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
  float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del poz
  float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del tar
  Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
  Serial1.print(","); //serial del modulo BT cada 1 segundo
  Serial1.print(nivel_tanque);
  Serial1.print("\n");
  digitalWrite(led_caudal,LOW); //Apago led caudal
  cerrar_valvula(); //Cierro valvula esferica
  apaga_bomba(); //Apago bomba centrifuga

  if (nivel_tanque>=0.0 && nivel_tanque<=0.10) {colort(255,255,255);} //NIVEL 1 del tanque= VACIO(BLANCO)
  if (nivel_tanque>0.10 && nivel_tanque<=0.25) {colort(255,255,0);} //NIVEL 2 del tanque= MALO(AMARILLO)
  if (nivel_tanque>0.25 && nivel_tanque<=0.35) {colort(0,255,0);} //NIVEL 3 del tanque= REGULAR(VERDE)
  if (nivel_tanque>0.35 && nivel_tanque<=0.45) {colort(255,0,255);} //NIVEL 4 del tanque= MEDIO(MAJENTA)
  if (nivel_tanque>0.45 && nivel_tanque<=0.55) {colort(0,255,255);} //NIVEL 5 del tanque= OPTIMO(CELESTE)
  if (nivel_tanque>0.55 && nivel_tanque<=0.65) {colort(0,0,255);} //NIVEL 6 del tanque= MAXIMO(AZUL)
  if (nivel_tanque>0.65) {colort(255,0,0);} //NIVEL 7 del tanque= DESBORDAMIENTO(RO

```

```

if (nivel_pozo>=0.0 && nivel_pozo<=0.15) {colorp(255,0,0);} //NIVEL 1 del pozo= CRITICO(ROJO)
if (nivel_pozo>0.15 && nivel_pozo<=0.30) {colorp(255,255,0);} //NIVEL 2 del pozo= MALO(AMARILLO)
if (nivel_pozo>0.30 && nivel_pozo<=0.60) {colorp(0,255,0);} //NIVEL 3 del pozo= REGULAR(VERDE)
if (nivel_pozo>0.60 && nivel_pozo<=0.90) {colorp(255,0,255);} //NIVEL 4 del pozo= BUENO(MAJENTA)
if (nivel_pozo>0.90 && nivel_pozo<=1.20) {colorp(0,255,255);} //NIVEL 5 del pozo= EXCELENTE(CELESTE)
if (nivel_pozo>1.20 && nivel_pozo<=1.50) {colorp(0,0,255);} //NIVEL 6 del pozo= OPTIMO(AZUL)
if (nivel_pozo>1.50) {colorp(255,255,255);} //NIVEL 7 del pozo= MAXIMO(BLANCO)
while((nivel_tanque>0.35 && nivel_tanque<=0.65)||((nivel_pozo>=0.0 && nivel_pozo<=0.15) || nivel_tanque>0.65)).
int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del pozo
float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del tanque
Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
Serial1.print(","); //serial del modulo BT cada 1 segundo
Serial1.print(nivel_tanque);
Serial1.print("\n");
if (nivel_pozo>=0.0 && nivel_pozo<=0.15){//CONDICION DE POZO SIN AGUA
colorp(255,0,0); //NIVEL 1 del pozo= CRITICO(ROJO)
int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del pozo
float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del tanque
Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
Serial1.print(","); //serial del modulo BT cada 1 segundo
Serial1.print(nivel_tanque);
Serial1.print("\n");

while(nivel_pozo>=0.0 && nivel_pozo<=0.30){//Bucle de llenado minimo de agua del pozo
int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del pozo
float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del tanque
Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
Serial1.print(","); //serial del modulo BT cada 1 segundo
Serial1.print(nivel_tanque);
Serial1.print("\n");
if (nivel_pozo>=0.0 && nivel_pozo<=0.15) {colorp(255,0,0);} //NIVEL 1 del pozo= CRITICO(ROJO)
if (nivel_pozo>0.15 && nivel_pozo<=0.25) {colorp(255,255,0);} //NIVEL 2 del pozo= MALO(AMARILLO)
if (nivel_pozo>0.25){ //Condicion para tener un minimo de agua en el pozo
colorp(255,255,0); //NIVEL 2 del pozo= MALO(AMARILLO)
int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del pozo
float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del tanque
Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
Serial1.print(","); //serial del modulo BT cada 1 segundo
Serial1.print(nivel_tanque);
Serial1.print("\n");
goto inicio; //Salgo de bucle de llenado del pozo
}

if (nivel_tanque>=0.0 && nivel_tanque<=0.10){colort(255,255,255);} //NIVEL 1 del tanque= VACIO(BLANCO)
if (nivel_tanque>0.10 && nivel_tanque<=0.25){colort(255,255,0);} //NIVEL 2 del tanque= MALO(AMARILLO)
if (nivel_tanque>0.25 && nivel_tanque<=0.35){colort(0,255,0);} //NIVEL 3 del tanque= REGULAR(VERDE)
if (nivel_tanque>0.35 && nivel_tanque<=0.45){colort(255,0,255);} //NIVEL 4 del tanque= MEDIO(MAJENTA)
if (nivel_tanque>0.45 && nivel_tanque<=0.55){colort(0,255,255);} //NIVEL 5 del tanque= OPTIMO(CELESTE)
if (nivel_tanque>0.55 && nivel_tanque<=0.65){colort(0,0,255);} //NIVEL 6 del tanque= MAXIMO(AZUL)
if (nivel_tanque>0.65) {colort(255,0,0);} //NIVEL 7 del tanque= DESBORDAMIENTO(ROJO)
if ((Serial1.available()>0)|| (SIM900.available()>0)){//Evaluo si el sistema recibe carater para cambio de modo=Serial1.read(); //Guardo caracter enviado desde la App
smscelu=SIM900.read(); //Guardo caracter enviado desde SMS de texto
if((modo=='X')||(smscelu=='X')){ //Verifico de donde fue enviado el caracter X
if (smscelu=='X'){ //Carecter enviado por medio de SMS de texto
mensaje_stanby(); //Envio ACK de confirmacion
}
digitalWrite(led_standby,HIGH); //Enciendo led standby
digitalWrite(led_automatico,LOW); //Apago led automatico
digitalWrite(led_caudal,LOW); //Apago led caudal
cerrar_valvula(); //Cierro valvula esferica
apaga_bomba(); //Apago bomba centrifuga
}
}

```

```

while((modo=='X')||(smscelu=='X')){ //Bucle de modo operacion STANDBY
  STANDBY(); //Llamo la funcion standby
  if ((Serial1.read()=='Y')||(SIM900.read()=='Y')){//Evaluo si el sistema recibe carater para cambio de
    if(SIM900.read()=='Y'){ //Verifico si el caracter Y fue enviado por SMS
      mensaje_automatico(); //Envio ACK de confirmacion
    }
    goto inicio; //Salgo del bucle de operacion STANDBY
  }
}
}
}
}
if (nivel_pozo>0.15 && nivel_pozo<=0.30) {colorp(255,255,0);} //NIVEL 2 del pozo= MALO (AMARILLO)
if (nivel_pozo>0.30 && nivel_pozo<=0.60) {colorp(0,255,0);} //NIVEL 3 del pozo= REGULAR (VERDE)
if (nivel_pozo>0.60 && nivel_pozo<=0.90) {colorp(255,0,255);} //NIVEL 4 del pozo= BUENO (MAJENTA)
if (nivel_pozo>0.90 && nivel_pozo<=1.20) {colorp(0,255,255);} //NIVEL 5 del pozo= EXCELENTE (CELESTE)
if (nivel_pozo>1.20 && nivel_pozo<=1.50) {colorp(0,0,255);} //NIVEL 6 del pozo= OPTIMO (AZUL)
if (nivel_pozo>1.50) {colorp(255,255,255);} //NIVEL 7 del pozo= MAXIMO (BLANCO)
if (nivel_tanque>0.35 && nivel_tanque<=0.45){colort(255,0,255);}//NIVEL 4 del tanque= MEDIO (MAJENTA)
if (nivel_tanque>0.45 && nivel_tanque<=0.55){colort(0,255,255);}//NIVEL 5 del tanque= OPTIMO (CELESTE)
if (nivel_tanque>0.55 && nivel_tanque<=0.65){colort(0,0,255);} //NIVEL 6 del tanque= MAXIMO (AZUL)
if (nivel_tanque>0.65) {colort(255,0,0);} //NIVEL 7 del tanque= DESBORDAMIENTO (ROJO)
if ((Serial1.available()>0)||(SIM900.available()>0)){//Evaluo si el sistema recibe carater para cambio de
modo=Serial1.read(); //Guardo caracter enviado desde la App
smscelu=SIM900.read(); //Guardo caracter enviado desde SMS de texto
if((modo=='X')||(smscelu=='X')){ //Verifico de donde fue enviado el caracter X
  if (smscelu=='X'){ //Carecter enviado por medio de SMS de texto
    mensaje_stanby(); //Envio ACK de confirmacion
  } //cambio de operacion desde un SMS de texto
  digitalWrite(led_standby,HIGH); //Enciendo led standby
  digitalWrite(led_automatico,LOW); //Apago led automatico
  digitalWrite(led_caudal,LOW); //Apago led caudal
  cerrar_valvula(); //Cierro valvula esferica
  apaga_bomba(); //Apago bomba centrifuga
}
while((modo=='X')||(smscelu=='X')){ //Bucle de modo operacion STANDBY
  STANDBY(); //Llamo la funcion standby
  if ((Serial1.read()=='Y')||(SIM900.read()=='Y')){//Evaluo si el sistema recibe carater para cambio de
    if(SIM900.read()=='Y'){ //Verifico si el caracter Y fue enviado por SMS
      mensaje_automatico(); //Envio ACK de confirmacion
    }
    goto inicio; //Salgo del bucle de operacion STANDBY
  }
}
}
}
if (nivel_pozo>0.25 && nivel_tanque<=0.33){ //Condicion de succion de agua
  int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en
  float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor F
  float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor F
  Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
  Serial1.print(","); //serial del modulo BT cada 1 segundo
  Serial1.print(nivel_tanque);
  Serial1.print("\n");
  enciende_bomba(); //Enciende bomba centrifuga

  abrir_valvula(); //Abrir valvula esferica
  goto inicio; //Salgo de bucle de consumo de agua
}
}
}
//CONDICION DE SUCCION DE AGUA DEL POZO
if (nivel_pozo>0.15 && nivel_tanque<=0.35){
  int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
  float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del
  float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04 del

```



```

Serial1.print(nivel_pozo);           //Imprimo los valores que son tx por el
Serial1.print(",");                 //serial del modulo BT cada 1 segundo
Serial1.print(nivel_tanque);
Serial1.print("\n");
enciende_bomba();                   //Enciende bomba centrifuga
abrir_valvula();                     //Abrir valvula esferica
if (nivel_pozo>0.15 && nivel_pozo<=0.30) {colorp(255,255,0);} //NIVEL 2 del pozo= MALO(AMARILLO)
if (nivel_pozo>0.30 && nivel_pozo<=0.60) {colorp(0,255,0);} //NIVEL 3 del pozo= REGULAR(VERDE)
if (nivel_pozo>0.60 && nivel_pozo<=0.90) {colorp(255,0,255);} //NIVEL 4 del pozo= BUENO(MAJENTA)
if (nivel_pozo>0.90 && nivel_pozo<=1.20) {colorp(0,255,255);} //NIVEL 5 del pozo= EXCELENTE(CELESTE)
if (nivel_pozo>1.20 && nivel_pozo<=1.50) {colorp(0,0,255);} //NIVEL 6 del pozo= OPTIMO(AZUL)
if (nivel_pozo>1.50) {colorp(255,255,255);} //NIVEL 7 del pozo= MAXIMO(BLANCO)
if (nivel_tanque>=0.0 && nivel_tanque<=0.10){colort(255,255,255);} //NIVEL 1 del tanque= VACIO(BLANCO)
if (nivel_tanque>0.10 && nivel_tanque<=0.25){colort(255,255,0);} //NIVEL 2 del tanque= MALO(AMARILLO)
if (nivel_tanque>0.25 && nivel_tanque<=0.35){colort(0,255,0);} //NIVEL 3 del tanque= REGULAR(VERDE)
for (int i=0;i<7;i++){delay(10000);} //Retardo necesario para que comienze a medir caudal
while(nivel_pozo>0.15 && nivel_tanque<=0.35){ //BUCLE DE SUCCION DE AGUA
  int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en fi
  float caudal = (Numpulsos/7.5); //Calculo del caudal Q=frec/K
  if(caudal>2.28){ //Evaluo el valor de caudal (L/min)que este sobre el valc
    digitalWrite(led_caudal,HIGH); //Enciendo led caudal
    float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR
    float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR
    float volumen = pulsosAcumulados*1.0/450; //Calculo del volumen succionado
    archivo = SD.open("DATOS.txt", FILE_WRITE); //Crea el archivo en .txt y da privilegios de solo de lec
    archivo.print(caudal); //Imprimo valos de caudal
    archivo.print(" "); // y
    archivo.println(volumen); //volumen en la tarjeta micro-sd en el archivo datos
    archivo.close(); //Cierro el archivo DATOS
    Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
    Serial1.print(","); //serial del modulo BT cada 1 segundo
    Serial1.print(nivel_tanque);
    Serial1.print(",");
    Serial1.print(caudal);
    Serial1.print(",");
    Serial1.print(volumen);
    Serial1.print("\n");
    if (nivel_pozo>=0.0 && nivel_pozo<=0.15) { //CONDICION DE POZO SIN AGUA
      colorp(255,0,0); //NIVEL 1 del pozo= CRITICO(ROJO)
      digitalWrite(led_caudal,LOW); //Apago led caudal
      int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en fre
      float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR
      float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR
      Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
      Serial1.print(","); //serial del modulo BT cada 1 segundo
      Serial1.print(nivel_tanque);
      Serial1.print("\n");
      cerrar_valvula(); //Cierro valvula esferica
      apaga_bomba(); //Apago bomba centrifuga
      goto inicio; //Salgo del bucle succion de agua
    }
    if (nivel_pozo>0.15 && nivel_pozo<=0.30) {colorp(255,255,0);} //NIVEL 2 del pozo= MALO(AMARILLO)
    if (nivel_pozo>0.30 && nivel_pozo<=0.60) {colorp(0,255,0);} //NIVEL 3 del pozo= REGULAR(VERDE)
    if (nivel_pozo>0.60 && nivel_pozo<=0.90) {colorp(255,0,255);} //NIVEL 4 del pozo= BUENO(MAJENTA)
    if (nivel_pozo>0.90 && nivel_pozo<=1.20) {colorp(0,255,255);} //NIVEL 5 del pozo= EXCELENTE(CELE
    if (nivel_pozo>1.20 && nivel_pozo<=1.50) {colorp(0,0,255);} //NIVEL 6 del pozo= OPTIMO(AZUL)
    if (nivel_pozo>1.50) {colorp(255,255,255);} //NIVEL 7 del pozo= MAXIMO(BLANCO)
    if (nivel_tanque>=0.0 && nivel_tanque<=0.10){colort(255,255,255);} //NIVEL 1 del tanque= VACIO(BLANCC
    if (nivel_tanque>0.10 && nivel_tanque<=0.25){colort(255,255,0);} //NIVEL 2 del tanque= MALO(AMARILLI
    if (nivel_tanque>0.25 && nivel_tanque<=0.35){colort(0,255,0);} //NIVEL 3 del tanque= REGULAR(VERE
    if (nivel_tanque>0.35 && nivel_tanque<=0.45){colort(255,0,255);} //NIVEL 4 del tanque= MEDIO(MAJENTI
    if (nivel_tanque>0.45 && nivel_tanque<=0.55){colort(0,255,255);} //NIVEL 5 del tanque= OPTIMO(CELES
    if (nivel_tanque>0.55 && nivel_tanque<=0.65){colort(0,0,255);} //NIVEL 6 del tanque= MAXIMO(AZUL)

```

```

if (nivel_tanque>0.65) { //CONDICION DE LLENADO MAXIMO DEL TANQUE
  colort(255,0,0); //NIVEL 7 del tanque= DESBORDAMIENTO(ROJO)
  digitalWrite(led_caudal,LOW); //Apago led caudal
  cerrar_valvula(); //Cierro valvula esferica
  apaga_bomba(); //Apago bomba centrifuga
  int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
  float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SRO
  float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SRO
  Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
  Serial1.print(","); //serial del modulo BT cada 1 segundo
  Serial1.print(nivel_tanque);
  Serial1.print("\n");
  mensaje_llenado(); //Envio un SMS de aviso cuando el tanque esta lleno
  goto inicio; //Salgo de bucle se succion de agua
}
if ((Serial1.available()>0)|| (SIM900.available()>0)){//Evaluo si el sistema recibe carater para camb
modo=Serial1.read(); //Guardo caracter enviado desde la App
smscelu=SIM900.read(); //Guardo caracter enviado desde SMS de texto
if((modo=='X')||(smscelu=='X')){ //Verifico de donde fue enviado el caracter X
  if (smscelu=='X'){ //Carecater enviado por medio de SMS de texto
    mensaje_stanby(); //Envio ACK de confirmacion
  } //cambio de operacion desde un SMS de texto
  digitalWrite(led_standby,HIGH); //Enciendo led standby
  digitalWrite(led_automatico,LOW); //Apago led automatico
  digitalWrite(led_caudal,LOW); //Apago led caudal
  cerrar_valvula(); //Cierro valvula esferica
  apaga_bomba(); //Apago bomba centrifuga
}
while((modo=='X')||(smscelu=='X')){ //Bucle de modo operacion STANDBY
  STANDBY(); //Llamo la funcion standby
  if ((Serial1.read()=='Y')||(SIM900.read()=='Y')){//Evaluo si el sistema recibe carater para cambic
    if (SIM900.read()=='Y'){ //Verifico si el caracter Y fue enviado por SMS
      mensaje_automatico(); //Envio ACK de confirmacion
    }
    goto inicio; //Salgo del bucle de operacion STANDBY
  }
}
}
else { //Evaluo el valor de caudal que este por debajo del x
  digitalWrite(led_caudal,LOW); //Apago led_caudal
  float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor
  float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor
  float volumen = pulsosAcumulados*1.0/450; //Calculo del volumen succionado
  archivo = SD.open("DATOS.txt", FILE_WRITE); //Crea el archivo en .txt y da privilegios de solo de
  archivo.print(caudal); //Imprimo valos de caudal
  archivo.print(" "); //y
  archivo.println(volumen); //volumen en la tarjeta micro-sd en el archivo datos
  archivo.close(); //Cierro el archivo
  Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
  Serial1.print(","); //serial del modulo BT cada 1 segundo
  Serial1.print(nivel_tanque);
  Serial1.print(",");
  Serial1.print(caudal);

  Serial1.print(",");
  Serial1.print(volumen);
  Serial1.print("\n");
  cerrar_valvula(); //Cierro valvula esferica
  apaga_bomba(); //Apago bomba centrifuga
  mensaje_bloqueo(); //Envio SMS de aviso que el sistema esta bloqueado
  digitalWrite(led_automatico,LOW); //Apago led_automatico
}
}
}

```

```

while(1){
    //Ingreso al bucle cuando se bloquea el sistema
    int frec=frecuencia(); //Llamo a la funcion frecuencia y almaceno su valor en frec
    float nivel_pozo=sensor_pozo(3.46); //Seteo la altura a la cual sera instalado el sensor HC-SR04
    float nivel_tanque=sensor_tanque(0.72); //Seteo la altura a la cual sera instalado el sensor HC-SR04
    float caudal = (Mumpulsos/7.5); //Calculo del caudal Q=frec/K
    Serial1.print(nivel_pozo); //Imprimo los valores que son tx por el
    Serial1.print(","); //serial del modulo BT cada 1 segundo
    Serial1.print(nivel_tanque);
    Serial1.print(",");
    Serial1.print(caudal);
    Serial1.print(",");
    Serial1.print(volumen);
    Serial1.print("\n");
    if (nivel_pozo>=0.0 && nivel_pozo<=0.15) {colorp(255,0,0);} //NIVEL 1 del pozo= CRITICO (ROJO)
    if (nivel_pozo>0.15 && nivel_pozo<=0.30) {colorp(255,255,0);} //NIVEL 2 del pozo= MALO (AMARILLO)
    if (nivel_pozo>0.30 && nivel_pozo<=0.60) {colorp(0,255,0);} //NIVEL 3 del pozo= REGULAR (VERDE)
    if (nivel_pozo>0.60 && nivel_pozo<=0.90) {colorp(255,0,255);} //NIVEL 4 del pozo= BUENO (MAJENTA)
    if (nivel_pozo>0.90 && nivel_pozo<=1.20) {colorp(0,255,255);} //NIVEL 5 del pozo= EXCELENTE (CELESTE)
    if (nivel_pozo>1.20 && nivel_pozo<=1.50) {colorp(0,0,255);} //NIVEL 6 del pozo= OPTIMO (AZUL)
    if (nivel_pozo>1.50) {colorp(255,255,255);} //NIVEL 7 del pozo= MAXIMO (BLANCO)
    if (nivel_tanque>=0.0 && nivel_tanque<=0.10) {colort(255,255,255);} //NIVEL 1 del tanque= VACIO (BLANCO)
    if (nivel_tanque>0.10 && nivel_tanque<=0.25) {colort(255,255,0);} //NIVEL 2 del tanque= MALO (AMARILLO)
    if (nivel_tanque>0.25 && nivel_tanque<=0.35) {colort(0,255,0);} //NIVEL 3 del tanque= REGULAR (VERDE)
    if (nivel_tanque>0.35 && nivel_tanque<=0.45) {colort(255,0,255);} //NIVEL 4 del tanque= MEDIO (MAJENTA)
    if (nivel_tanque>0.45 && nivel_tanque<=0.55) {colort(0,255,255);} //NIVEL 5 del tanque= OPTIMO (CELESTE)
    if (nivel_tanque>0.55 && nivel_tanque<=0.65) {colort(0,0,255);} //NIVEL 6 del tanque= MAXIMO (AZUL)
    if (nivel_tanque>0.65) {colort(255,0,0);} //NIVEL 7 del tanque= DESBORDAMIENTO

    digitalWrite(led_reset,HIGH); //Enciende led_reset
    delay(4000); //Tiempo necesario para observar el encendido
    digitalWrite(led_reset,LOW); //Apago led_reset
}
}
}
}
}
//EJECUCION DEL CODIGO
void loop(){
    AUTOMATICO();
}

```

ANEXO D

Código fuente de la aplicación móvil implementada en App Inventor.

A. Interfaz y código de la pantalla de login.



```
cuando Screen1 . Inicializar
ejecutar
  llamar Activity Starter1 . IniciarActividad
  poner Reloj_INICIO . TemporizadorHabilitado como cierto
  poner PANTALLA_TITULO . Visible como cierto
  poner PANTALLA_INICO . Visible como cierto
  poner ETI_NUMERO . Texto como 0
  poner PANTALLA_CONTRASEÑA . Visible como falso
  poner PANTALLA_CONEXION . Visible como falso
  poner PANTALLA_AUTO . Visible como falso
  poner PANTALLA_STANDBY . Visible como falso
  poner PANTALLA_DE_ALERTA . Visible como falso
  poner Reloj_AUTO . TemporizadorHabilitado como falso
  poner Reloj_STANDBY . TemporizadorHabilitado como falso

cuando Reloj_INICIO . Temporizador
ejecutar
  si
    ETI_NUMERO . Texto <= 100
  entonces
    poner ETI_NUMERO . Texto como ETI_NUMERO . Texto + 1
    si
      ETI_NUMERO . Texto = 100
    entonces
      poner PANTALLA_INICO . Visible como falso
      poner PANTALLA_CONTRASEÑA . Visible como cierto
  sino
    poner Reloj_INICIO . TemporizadorHabilitado como falso
```

B. Interfaz y código de ingreso.



```
cuando BTN_NEXT_DE_PANTALLA_CONTRASEÑA .Clic
ejecutar
  si
    CampoDeContraseña1 . Texto = "Tesis2020"
  entonces
    poner CampoDeContraseña1 . Texto como ""
    poner PANTALLA_TITULO . Visible como falso
    poner PANTALLA_CONTRASEÑA . Visible como falso
    poner PANTALLA_CONEXION . Visible como cierto
    poner BTN_DESCONEXION_BT . Visible como falso
    poner PANTALLA_AUTO . Visible como cierto
    poner BOMBA_LLAVES_OFF_AUTO . Visible como falso
    poner BOMBA_LLAVES_ON_AUTO . Visible como falso
    poner ETI_SUCCIONANDO . Visible como falso
    poner ETI_TANQUE_LLENO . Visible como falso
    poner ETI_BLOQUEO . Visible como falso
  sino
    llamar Notificador1 .MostrarAlerta
      aviso "CONTRASEÑA INCORRECTA"
    poner CampoDeContraseña1 . Texto como ""
    poner PANTALLA_TITULO . Visible como cierto
    poner PANTALLA_CONTRASEÑA . Visible como cierto

cuando BTN_SALIR_PANTALLA_CONTRASEÑA .Clic
ejecutar
  llamar Notificador1 .MostrarAlerta
    aviso "CERRANDO..."
  cerrar la aplicación
```

C. Interfaz y código para la conexión Bluetooth.



```

estado LP_CONEXION_BT - Antes De Selección
ejecutar poner LP_CONEXION_BT - Elementos como ClienteBluetooth1 - DirecciónYNombres
    
```

```

estado LP_CONEXION_BT - Después De Selección
ejecutar si
    llamar ClienteBluetooth1 - Colectar
        dirección LP_CONEXION_BT - Selección
    entonces
        ClienteBluetooth1 - Colectado
            llamar ClienteBluetooth1 - EnviarTexto
                texto Y
            poner LP_CONEXION_BT - Visible como falso
            poner BTN_DESCONEXION_BT - Visible como verdadero
            poner Reloj_AUTO - TemporizadorHabilitado como verdadero
            poner BOMBA_LLAVES_OFF_AUTO - Visible como verdadero
            poner global DATOS_AUTO a verdadero
            poner global DATOS_STANDBY a falso
    sino
        poner global DATOS_AUTO a falso
        poner global DATOS_STANDBY a falso
    fin si
    
```

```

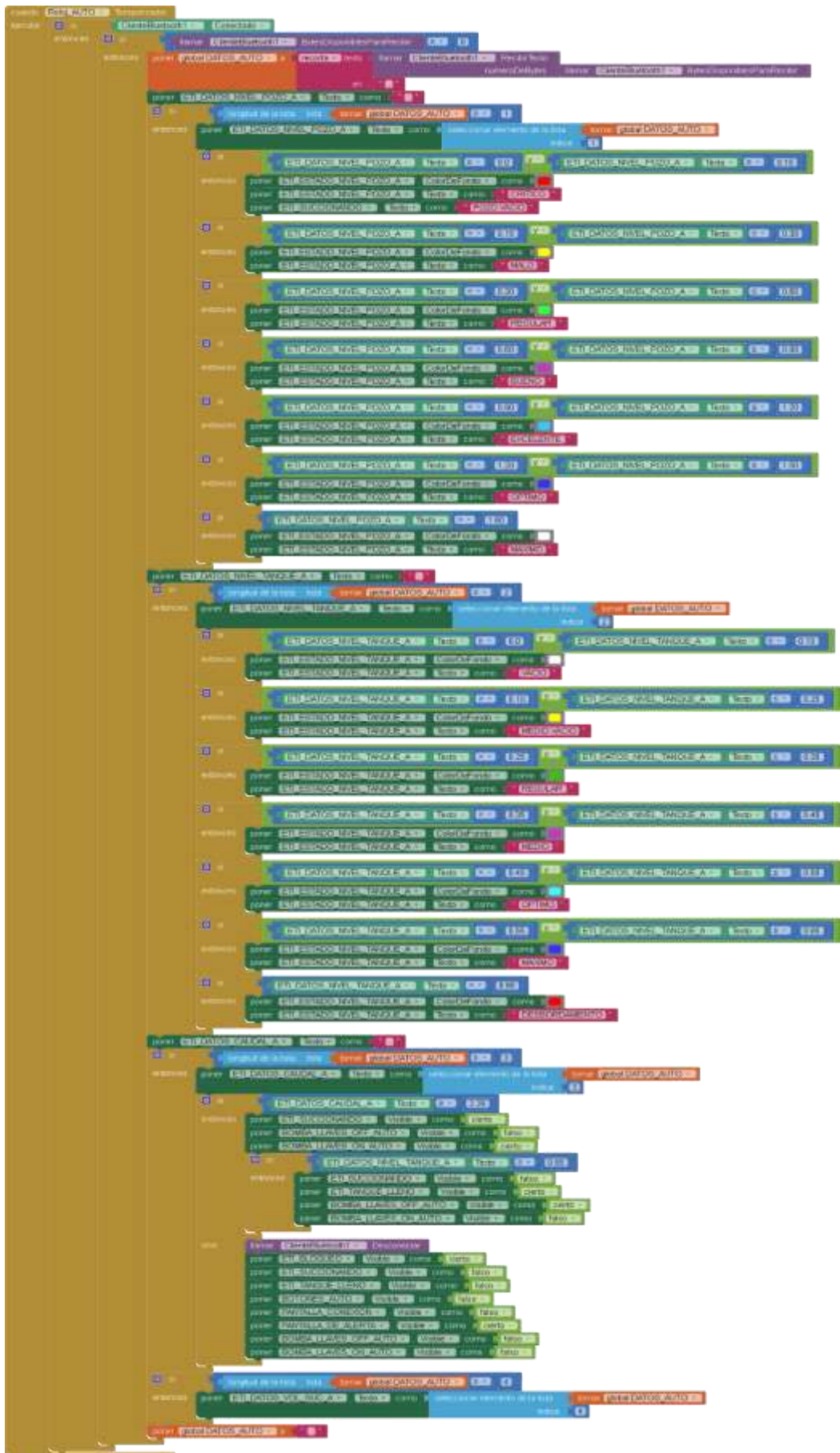
estado BTN_DESCONEXION_BT - Clic
ejecutar llamar Notificador1 - MostrarAlerta
    auto DESCONECTANDO...
    llamar ClienteBluetooth1 - Desconectar
    poner BTN_DESCONEXION_BT - Visible como falso
    poner LP_CONEXION_BT - Visible como verdadero
    poner PANTALLA_AUTO - Visible como verdadero
    poner PANTALLA_STANDBY - Visible como falso
    poner Reloj_STANDBY - TemporizadorHabilitado como falso
    poner Reloj_AUTO - TemporizadorHabilitado como falso
    poner global DATOS_AUTO a falso
    poner global DATOS_STANDBY a falso
inicializar global DATOS_AUTO como falso
inicializar global DATOS_STANDBY como falso
    
```

D. Interfaz y código para visualización de datos en modo automático del sistema.



```
cuando BTN_PASAR_PANTALLA_AUTO_A_STANDBY .Clic
ejecutar
  llamar ClienteBluetooth1 .EnviarTexto
    texto "X"
  poner PANTALLA_AUTO .Visible como falso
  poner PANTALLA_STANDBY .Visible como cierto
  poner Reloj_AUTO .TemporizadorHabilitado como falso
  poner Reloj_STANDBY .TemporizadorHabilitado como cierto
  poner global DATOS_AUTO a falso
  poner global DATOS_STANDBY a cierto

cuando BTN_SALIR_PANTALLA_AUTO .Clic
ejecutar
  llamar ClienteBluetooth1 .Desconectar
  llamar Notificador1 .MostrarAlerta
    aviso "CERRANDO.."
  cerrar la aplicación
```



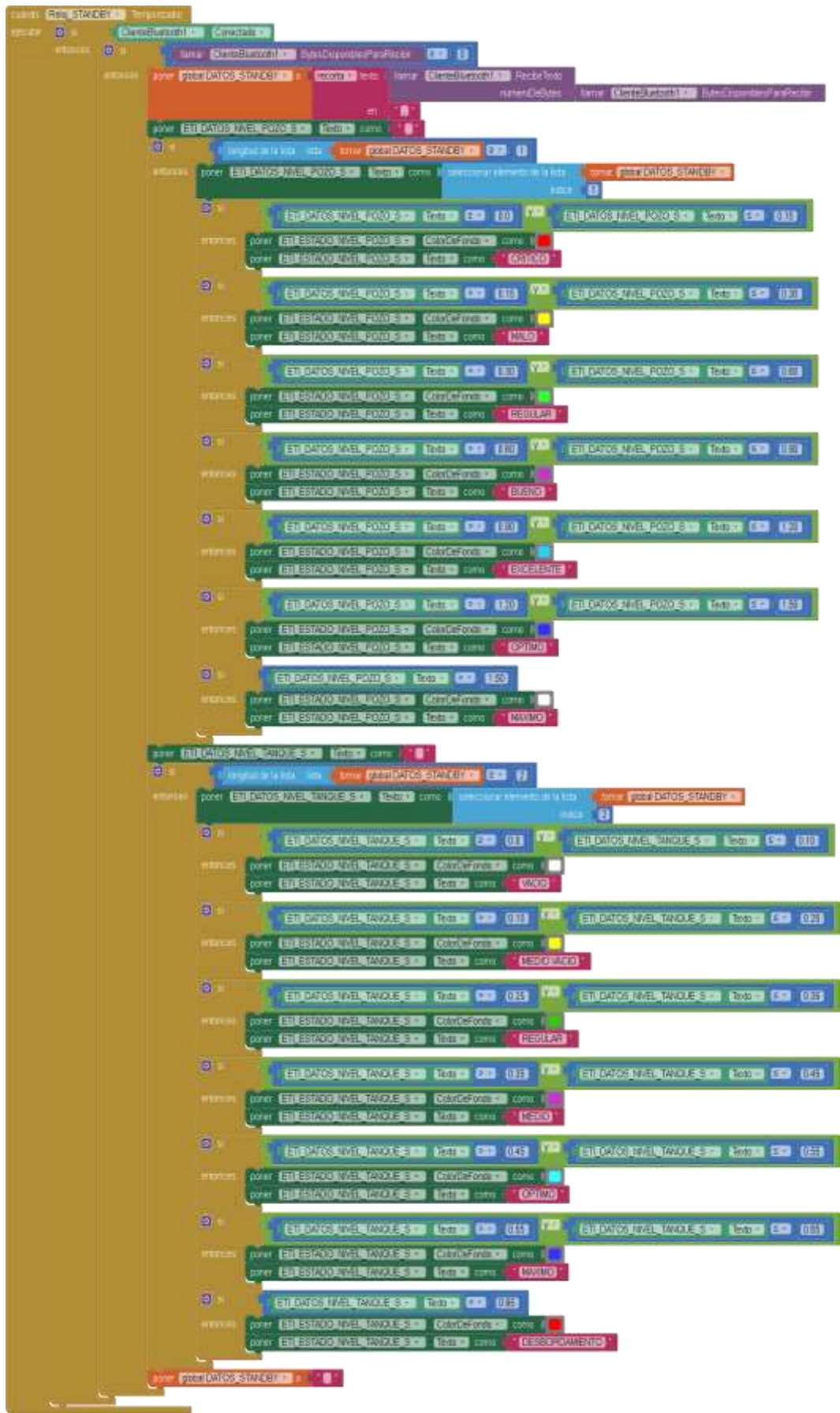
E. Interfaz y código para visualización de datos en modo standby del sistema



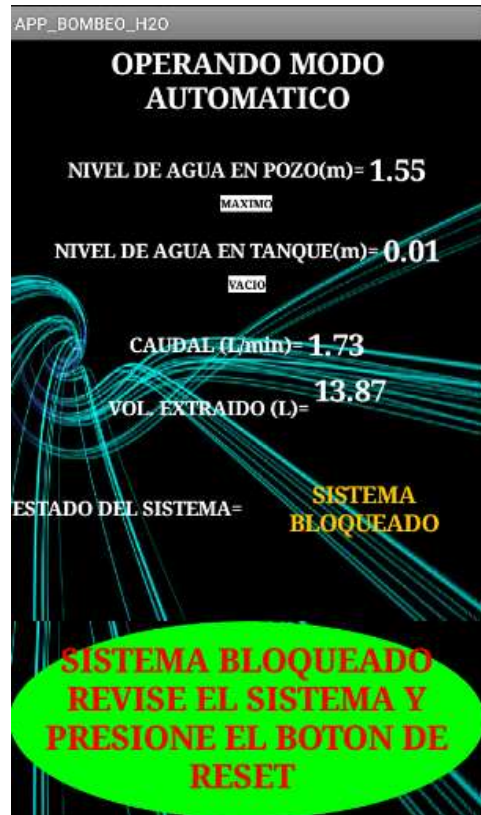
```

estado: BTN_PASAR_STANDBY_A_AUTO - Clic
ejecutar
  llamar ClienteBluetooth1 - .EnviarTexto
  texto: "Y"
  poner PANTALLA_AUTO - .Visible - como cliente
  poner PANTALLA_STANDBY - .Visible - como falso
  poner Reloj_AUTO - .TemporizadorHabilitado - como cliente
  poner Reloj_STANDBY - .TemporizadorHabilitado - como falso
  poner BOMBA_LLAVES_OFF_AUTO - .Visible - como cliente
  poner BOMBA_LLAVES_ON_AUTO - .Visible - como falso
  poner ETI_SUCCIONANDO - .Visible - como falso
  poner ETI_TANQUE_LLENO - .Visible - como falso
  poner ETI_BLOQUEO - .Visible - como falso
  poner global DATOS_AUTO - a cliente
  poner global DATOS_STANDBY - a falso

estado: BTN_SALIR_PANTALLA_STANDBY - Clic
ejecutar
  llamar ClienteBluetooth1 - .Desconectar
  llamar Notificador1 - .MostrarAlerta
  auto: "CERRANDO..."
  cerrar la aplicación
  
```



F. Interfaz y código cuando el sistema está bloqueado por escasez de caudal.



```
cuando BTN_ALERTA .Clic  
ejecutar llamar ClienteBluetooth1 .Desconectar  
cerrar la aplicación
```

ORDEN DE EMPASTADO