

**UNIVERSIDAD NACIONAL AGRARIA LA
MOLINA**

FACULTAD DE INGENIERIA AGRÍCOLA



**“IMPLEMENTACIÓN, CONTROL Y MONITOREO DE UN
SISTEMA DE RIEGO POR GOTEO SUBTERRÁNEO CON
MICROCONTROLADORES”**

Presentado por:

GEORGE ANTHONY SIMÓN MORI

TESIS PARA OPTAR POR EL TÍTULO DE:

INGENIERO AGRÍCOLA

Lima – Perú

2018

DEDICATORIA

A Jehová Dios por haberme dado la vida y por permitirme cumplir mis metas.

Dedico esta Tesis a mis padres Alicia Mori Morales y Apolinario Simón Aquino que siempre me apoyaron incondicionalmente en la parte moral y económica para poder llegar a ser un profesional de la Universidad Nacional Agraria La Molina.

A mi hermano Ray Joseph Simón Mori por el apoyo que siempre me brindó día a día en el transcurso de cada año de mi carrera Universitaria y por ser un paradigma de mi persona.

A mis familiares en general por ser las personas que más han influenciado en mi vida, dándome los mejores consejos, guiándome y haciéndome una persona de bien.

AGRADECIMIENTOS

- A Dios por permitirme tener tan buena experiencia dentro de mi universidad, gracias a mi universidad por permitirme convertirme en ser un profesional en lo que tanto me apasiona, y como recuerdo y prueba viviente en la historia; esta tesis, que perdurará dentro de los conocimientos y desarrollo de las demás generaciones que están por llegar.
- A la Universidad Nacional Agraria La Molina, a la Facultad de Ingeniería Agrícola y a los profesores por la formación ética y profesional impartida a lo largo de la carrera.
- Agradezco a mi asesor y co-asesor Mg. Sc. David Ascencios y el Ing. Jeisson Lluen respectivamente, por su paciencia, dedicación, consejos y velar que el presente documento sea un aporte científico y técnico en servicio del agro. A los doctores Eduardo Chávarri, Lía Fernández, Néstor Montalvo y la Ing. Karem Meza que mediante sus aportes técnicos enriquecieron la investigación. Al técnico Alan Santibáñez, por su disponibilidad y estar presto a apoyar en la investigación.

ÍNDICE GENERAL

RESUMEN

ABSTRACT

I.	INTRODUCCIÓN	1
II.	REVISIÓN DE LITERATURA	3
2.1.	SISTEMA DE RIEGO POR GOTEO SUBTERRÁNEO	3
2.2.	DOMÓTICA.....	3
2.2.1.	Automatización.....	4
2.2.2.	Sistema electrónico.....	4
2.2.3.	Microcontrolador	5
2.2.4.	Algoritmo	5
2.3.	ARDUINO.....	5
2.3.1.	Software Arduino	6
2.3.2.	Lenguaje del Arduino	7
2.3.3.	Librería Arduino	8
2.3.4.	Señales digitales	8
2.3.5.	Señales analógicas	8
2.3.6.	Características de una placa Arduino	9
2.4.	ARDUINO MEGA 2560.....	11
2.5.	SENSORES	13
2.5.1.	Sensor de humedad GS1	13
2.5.2.	Sensor de flujo de agua.....	14
2.5.3.	Transductor de presión	15
2.5.4.	Sensor ultrasonido HC-SR04	16
2.6.	ACTUADORES	17
2.6.1.	Módulo relé	17
2.6.2.	Válvula eléctrica.....	18
2.7.	MÓDULO BLUETOOTH	18
2.8.	APP INVENTOR	19
2.8.1.	Historia	19
2.8.2.	Características.....	19
2.8.3.	Forma de uso	20
III.	MATERIALES Y MÉTODOS	26

3.1.	UBICACIÓN DEL ÁREA EXPERIMENTAL.....	26
3.2.	ANTECEDENTES.....	27
3.3.	MATERIALES Y DISPOSITIVOS UTILIZADOS EN EL PROYECTO	29
3.4.	MÉTODOS.....	30
3.4.1.	Selección del microcontrolador y entorno de programación	30
3.4.2.	Selección del tipo de comunicación	31
3.4.3.	Variables a monitorear y dispositivos a controlar	31
3.4.4.	Selección de los sensores y actuadores.....	32
3.4.5.	Proceso de calibración de los sensores	33
3.4.6.	Proceso de programación del microcontrolador y aplicativo móvil.....	35
IV.	RESULTADOS Y DISCUSIÓN	42
4.1.	CARACTERÍSTICAS GENERALES DEL SISTEMA DE RIEGO SUBTERRÁNEO AUTOMATIZADO.....	42
4.2.	DISEÑO ELECTRÓNICO.....	44
4.2.1.	Sensor de humedad	45
4.2.2.	Transductor de presión	46
4.2.3.	Sensor de flujo de agua.....	47
4.2.4.	Sensor de ultrasonido.....	48
4.2.5.	Módulo relés asociados a la válvula solenoide y electrobomba.....	49
4.3.	CALIBRACIÓN.....	51
4.3.1.	Calibración de sensores de presión.....	51
4.3.2.	Calibración del sensor de humedad	54
4.3.3.	Calibración de sensores de flujo de agua.....	55
4.3.4.	Calibración del sensor de ultrasonido.....	57
4.4.	PROGRAMACIÓN.....	58
4.4.1.	Declaración de variables.....	59
4.4.2.	Configuración del código de programación (Setup).....	59
4.4.3.	Diagrama de flujo	60
4.5.	INTERFAZ DE USUARIO.....	62
4.5.1.	Interfaz de conexión	62
4.5.2.	Interfaz de presión	62
4.5.3.	Interfaz de caudal.....	63
4.5.4.	Interfaz sensor de humedad	64
4.5.5.	Interfaz sensor de ultrasonido.....	65
4.5.6.	Interfaz control de válvula solenoide y bomba.....	66

4.6.	MONITOREO DE LA HUMEDAD	67
4.7.	COMPARACIÓN DEL SISTEMA AUTOMATIZADO Y MANUAL.....	68
V.	CONCLUSIONES	69
VI.	RECOMENDACIONES.....	70
VII.	REFERENCIAS BIBLIOGRAFICAS	71
VIII.	ANEXOS	75

INDICE DE CUADROS

Cuadro 1: Características de las principales placas Arduino.....	12
Cuadro 2: Características del sensor de flujo de 1” y ½”	14
Cuadro 3: Resumen de entradas y salidas de la placa Arduino	51
Cuadro 4: Ecuaciones particulares de cada sensor instalado en campo	52
Cuadro 5: Cálculo del porcentaje de error de los transductores de presión	53
Cuadro 6: Factor de corrección para la conversión de frecuencia a caudal	55
Cuadro 7: Cálculo de porcentaje de error para la subunidad 4.....	56
Cuadro 8: Cálculo de porcentaje de error para la subunidad 1.....	56
Cuadro 9: Cálculo de porcentaje de error para la subunidad 2.....	57
Cuadro 10: Cálculo de porcentaje de error para la subunidad 3.....	57
Cuadro 11: Calculo de porcentaje de error para el cabezal.	57
Cuadro 12: Registro del volumen de agua para las subunidades automatizadas.....	68
Cuadro 13: Registro del volumen de agua para las subunidades de operación manual .	68
Cuadro 14: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 1.....	78
Cuadro 15: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 2.....	78
Cuadro 16: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 3.....	79
Cuadro 17: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 4.....	79
Cuadro 18: Resultado del factor de conversión (k) para el sensor de flujo del cabezal de riego	80
Cuadro 19: Porcentaje de humedad de suelo y señal analógica del sensor de humedad GS1.....	82

INDICE DE FIGURAS

Figura 1: Entrada USB PC-Arduino	10
Figura 2: Características del Arduino Mega 2560.....	11
Figura 3: Sensor GS1.....	13
Figura 4: Sensor de presión HK3022.	15
Figura 5: Sensor de ultrasonido HC-SR04.	16
Figura 6: Modulo Relé de 4 canales.	17
Figura 7: Válvula Eléctrica de ½”	18
Figura 8: Pantalla principal de App Inventor.	20
Figura 9: Elementos a utilizar y vista previa de app inventor	21
Figura 10: Propiedades de elementos.	22
Figura 11: Editor de bloques.	22
Figura 12: Códigos de control.	23
Figura 13: Códigos de lógica.....	23
Figura 14: Códigos de matemáticas.	24
Figura 15: Códigos de texto.	24
Figura 16: Códigos de lista.....	24
Figura 17: Códigos de variable.....	24
Figura 18: Códigos de procedimientos.....	25
Figura 19: Ubicación geográfica del área destinada a la unidad experimental.	26
Figura 20: Reservorio de agua.....	28
Figura 21: Cabezal de riego y sus principales componentes.	28
Figura 22: Arco de riego.....	29
Figura 23: Vaso comunicante.....	32
Figura 24: Diagrama eléctrico de las botoneras del tablero de control de la bomba.....	33
Figura 25: Microcontrolador Arduino, sensores y actuador.....	35
Figura 26: Comando Serial.print().	36
Figura 27: Transmisión de datos del microcontrolador al aplicativo.	37
Figura 28: Datos impresos de los sensores instalados.....	37
Figura 29: Interfaz de presiones del sistema de riego.	38
Figura 30: Esquema de instalación de los sensores en campo.	41
Figura 31: Vaso comunicante para el sensor de ultrasonido.	42

Figura 32: Tablero y relés de control de encendido y apagado de la bomba.	43
Figura 33: Sensor de presión antes y después del filtro.	43
Figura 34: Arco de riego automatizado.	44
Figura 35: Diseño electrónico sensor de humedad.	45
Figura 36: Diseño electrónico transductor de presión.	46
Figura 37: Diseño electrónico sensor de flujo de agua.	47
Figura 38: Diseño electrónico sensor de ultrasonido.	48
Figura 39: Diseño electrónico válvulas solenoides.	49
Figura 40: Diseño electrónico de la electrobomba.	50
Figura 41: Curvas de calibración de los transductores de presión.	53
Figura 42: Ecuación de ajuste para la transformación de señal, sensor GS1.	54
Figura 43: Calibración del sensor de humedad GS1.	54
Figura 44: Declaración de variables.	58
Figura 45: Configuración de variables.	60
Figura 46: Diagrama de flujo del código de programación para el control y monitoreo del sistema de riego.	61
Figura 47: Interfaz de inicio y conexión Bluetooth.	62
Figura 48: Interfaz de monitoreo de presiones.	63
Figura 49: Interfaz de monitoreo de caudales.	63
Figura 50: Interfaz de monitoreo de humedad y cálculo del tiempo de riego.	64
Figura 51: Interfaz de monitoreo de volumen de agua y distancia de vacío.	65
Figura 52: Interfaz de control del sistema de riego.	66
Figura 53: Monitoreo de la humedad de suelo.	67
Figura 54: Valores de la señal analógica para el sensor de la subunidad 4.	75
Figura 55: Ecuación de ajuste de los transductores de presión instalados.	76
Figura 56: Monitoreo de la frecuencia de pulsos para la calibración del sensor de flujo de agua.	77
Figura 57: Valores de la señal analógica para el sensor de humedad GS1.	81
Figura 58: Análisis de regresión simple para la ecuación de ajuste del sensor de humedad GS1.	82
Figura 59: Monitor serial del sensor de ultrasonido.	83
Figura 60: Ensayo para el sensor de flujo de agua de 1”	91
Figura 61: Materiales para la instalación del sensor de ultrasonido.	91

Figura 62: Calibración del sensor de ultrasonido	91
Figura 63: Delimitación de la zanja para los conductores eléctricos.	91
Figura 64: Excavación de zanjas para los conductores eléctricos.	92
Figura 65: Manguera de polietileno de 32 mm utilizada para conducir los cables eléctricos.....	92
Figura 66: Instalación de la manguera de polietileno para conducir los cables eléctricos.	92
Figura 67: Instalación de la red eléctrica.....	92
Figura 68: Instalación del tablero donde se alberga a los transformadores de 12V para las válvulas solenoides.	92
Figura 69: Instalación del tablero donde se alberga a los componentes electrónicos. ...	92
Figura 70: Instalación del arco de riego automatizado.....	92
Figura 71: Arco de riego manual y automatizado.	92
Figura 72: Instalación de los sensores de presión en las subunidades	92
Figura 73: Instalación del sensor de humedad GS1 a 10 cm de la superficie.	92
Figura 74: Instalación del cableado eléctrico para la automatización del tablero de la bomba.	92
Figura 75: Toma de muestra de suelo para la calibración del sensor de humedad.	92
Figura 76: Arco de riego de la subunidad 2 en funcionamiento.....	92
Figura 77: Arco de riego de la subunidad 2 en funcionamiento.....	92
Figura 78: Sensor de flujo de agua de 1” instalado en la tubería principal	92
Figura 79: Rejilla de PVC que protege al sensor de flujo de agua de 1”	92
Figura 80: Caja pase de 100x100mm que alberga al sensor de ultrasonido.....	92
Figura 81: Muestras de suelo para la calibración del sensor de humedad GS1	92
Figura 82: Alerta en el interfaz de nivel de agua.....	92
Figura 83: Registro de volumen en la válvula volumétrica.....	92
Figura 84: Sensor de caudal FS400 de 1” utilizado en el proyecto.....	92
Figura 85: Sensor de caudal YF-S 201 de ½” utilizado en el proyecto.....	92
Figura 86: Válvula solenoide de ½” utilizado en el proyecto	92
Figura 87: Arduino Mega 2560 utilizado en el proyecto	92
Figura 88: Sensor de ultrasonido HC-SR04 utilizado en el proyecto.....	92
Figura 89: Transductor de presión HK3022 utilizado en el proyecto	92
Figura 90: Subunidades automatizadas	92

INDICE DE ANEXOS

ANEXO 1: MANUAL DE CALIIBRACIÓN DE LOS SENSORES DE PRESIÓN....	75
ANEXO 2: MANUAL DE CALIIBRACION DE LOS SENSORES FLUJO DE AGUA (CAUDAL).....	76
ANEXO 3: MANUAL DE CALIIBRACION DEL SENSOR DE HUMEDAD GS1...	81
ANEXO 4: MANUAL DE CALIIBRACION DEL SENSOR DE ULTRASONIDO...	83
ANEXO 5: CÓDIGO DE PROGRAMACIÓN PARA EL PRENDIDO Y APAGADO DE UN RELÉ.....	84
ANEXO 6: CÓDIGO FUENTE ARDUINO IMPLEMENTADO EN EL MONITOREO Y CONTROL DE RIEGO.....	84
ANEXO 7: PRESUPUESTO PARA LA AUTOMATIZACIÓN.....	90
ANEXO 8: PANEL FOTOGRÁFICO	91

GLOSARIO

Automatización: Este verbo, por su parte, alude a hacer que determinadas acciones se vuelvan automáticas es decir, que se desarrollen por sí solas y sin la participación directa de una persona.

Bits: La placa Arduino contiene un convertidor analógico a digital. Esto significa que mapeará voltajes de entrada entre 0 y 5 voltios en valores enteros entre 0 y 1023 bits.

Hardware: Conjunto de componentes que conforman la parte física de una computadora suele distinguirse entre básico (los dispositivos necesarios para iniciar el funcionamiento de un ordenador) y complementario (realizan ciertas funciones específicas).

IDE: Del inglés Integrated Development Environment, o Entorno de Desarrollo Integrado, es un entorno de programación que ha sido desarrollado como un programa de aplicación, el cual consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

PWM: La modulación por ancho o de pulso del inglés pulse width modulation, es un tipo de señal de voltaje utilizada para enviar información o para modificar la cantidad de energía que se envía, emulando una señal analógica.

Software: Es el equipamiento lógico e intangible de un ordenador que permite al usuario tener el control sobre el hardware (componentes físicos) y dar soporte a otros programas informáticos.

Tensión: La tensión o tensión eléctrica también se conoce como voltaje, cuya unidad de medida es el voltio.

APP: Tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos para dispositivos móviles o tabletas

Interrupción: Señal recibida por el procesador de una computadora, para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.

Microcontrolador: Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres

principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

Opensource: Código abierto es la expresión con la que se conoce al software o hardware distribuido y desarrollado libremente. Se focaliza más en los beneficios prácticos (acceso al código fuente) que en cuestiones éticas o de libertad que tanto se destacan en el software libre.

Puerto Serie: Interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

Smartphone: El teléfono inteligente o “inteléfono” (en inglés: smartphone) es un tipo teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades, semejante a la de una minicomputadora, y con una mayor conectividad que un teléfono móvil convencional.

Piezoeléctrico: Es un fenómeno que ocurre en determinados cristales que, al ser sometidos a tensiones mecánicas, en su masa adquiere una polarización eléctrica y aparece una diferencia de potencial y cargas eléctricas en su superficie.

Optoacoplador: Es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo, que garantiza el aislamiento eléctrico entre dos circuitos mientras se mantiene un nexo de unión que se usa como un interruptor.

Hertz (Hz): Unidad de frecuencia, es conocida como Hertz o Hercio en castellano, está relacionada con la propagación de las ondas electromagnéticas.

RESUMEN

Los desarrollos tecnológicos actuales hacen que la combinación de la electrónica y la agricultura se necesiten mutuamente. Es así que la automatización del sistema de riego tiene como principales ventajas aumentar la eficiencia de los recursos (inversión, tiempo, mano de obra), gestión del recurso hídrico, mejorar la producción del cultivo en cantidad y calidad. En este proyecto de investigación se automatizó un sistema de riego por goteo subterráneo, implementando el microcontrolador de software libre Arduino (basado en el lenguaje de programación C), el cual es controlado y monitoreado por un teléfono inteligente. Se implementaron sensores analógicos y digitales para medir la presión antes y después de los filtros en el cabezal de riego, así también en los arcos de riego, humedad de suelo, caudal en el cabezal y arco de riego, nivel de agua en el reservorio. Se usó un microcontrolador Arduino Mega 2560, para recibir la información de los sensores y enviar información a los actuadores, mediante un código de programación, permitiendo el control del sistema de riego a través de relés, para el encendido y apagado de las electroválvulas y bomba. Los sensores y actuadores están gestionados por el módulo Arduino que se comunica con el usuario vía Bluetooth a través de un aplicativo desarrollado denominado “Ardunalm” en el software App Inventor. En el aplicativo se monitorea en tiempo real los sensores y se controla las electroválvulas y bomba. Además puede calcular el tiempo de riego en base a la información del sensor de humedad, iniciando el riego desde un temporizador. Los sensores utilizados poseen ecuaciones de ajuste y factor de conversión que pueden ser usados para códigos de programación, debido a que se ajustan a calibraciones con porcentajes de error dentro de las especificaciones técnicas. Finalmente, el aplicativo “Ardunalm” y el sistema de riego automatizado, permiten eliminar el control de riego manual por la manipulación del sistema (apertura y cierre de válvulas), aumentando la eficiencia de aplicación. Para el trabajo futuro en este proyecto, se recomienda implementar la conectividad WiFi, así como una datalogger para el registro de datos.

Palabras clave: Microcontrolador, control, monitoreo, aplicativo, sensores y actuadores.

ABSTRACT

The current technological developments mean that the combination of electronics and agriculture needs each other. Thus, the automation of the irrigation system has as main advantages to increase the efficiency of resources (investment, time, labor), water resource management, improve crop production in quantity and quality. In this research project, an underground drip irrigation system was automated, implementing the Arduino free software microcontroller (based on the C programming language), which is controlled and monitored by a smart phone. Analog and digital sensors were implemented to measure the pressure before and after the filters in the irrigation head, as well as in the irrigation arches, soil moisture, flow in the head and irrigation arc, water level in the reservoir. An Arduino Mega 2560 microcontroller was used to receive the information from the sensors and send information to the actuators, by means of a programming code, allowing control of the irrigation system through relays, for the switching on and off of the solenoid valves and pump. The sensors and actuators are managed by the Arduino module that communicates with the user via Bluetooth through a developed application called "Ardunalm" in the App Inventor software. In the application, the sensors are monitored in real time and the solenoid valves and pump are controlled. You can also calculate the watering time based on the humidity sensor information, starting the watering from a timer. The sensors used have adjustment equations and conversion factor that can be used for programming codes, because they adjust to calibrations with percentages of error within the technical specifications. Finally, the "Ardunalm" application and the automated irrigation system allow eliminating the manual irrigation control by manipulating the system (opening and closing of valves), increasing the application efficiency. For future work on this project, it is recommended to implement WiFi connectivity, as well as a datalogger for data logging.

Keywords: Microcontroller, control, monitoring, application, sensors and actuators.

I. INTRODUCCIÓN

En el año 2012 en el Perú, la superficie total equipada para el riego ascendió a 2 579 900 ha, de las cuales 2 362 144 ha (92 por ciento) es riego por superficie, 86 873 ha (3 por ciento) es riego por aspersión y 130 883 ha (5 por ciento) es riego localizado. La superficie efectivamente regada es de 1 808 302 ha (INEI 2012). Por estas estadísticas el Perú utiliza el 80% del agua en agricultura y se estima que para el 2030 el consumo de esta en la agricultura crecerá en 14%. Para enfrentar el desperdicio innecesario de agua en la agricultura, debido a la falta de conocimiento de los agricultores sobre el suministro adecuado de agua, se hace latente la necesidad del desarrollo de prácticas de gestión eficiente del recurso hídrico (Ingale y Kasat 2012; López y Choez 2016). La tecnificación y automatización de los sistemas de riego es una de las prácticas que se están desarrollando alrededor del mundo, para optimizar el uso de este recurso, aumentar la productividad del cultivo, reducir tiempo y costos operativos, permitir el manejo de información más rápida y almacenable (Mahzabin et al. 2016; Lucia et al. 2017).

En la actualidad, la electrónica se encuentra apoyando todos los campos del conocimiento, entre los que se encuentra el agro, sin embargo muchos estudiantes que se inician en carreras de ingenierías, no se sienten motivados en aprender a programar, por el hecho que la enseñanza tradicional de los lenguajes de programación de alto nivel no es el adecuado (Murillo et al. 2015), sin embargo la programación y los microcontroladores se están desarrollando en el plan de estudios de la Universidad Estatal de Portland de E.E.U.U. que se utiliza en los cursos de primer año tomadas por muchas disciplinas, incluyendo la ingeniería biomédica, ingeniería química, ingeniería civil e ingeniería industrial, mejorando la retención y atrayendo a mayor población de estudiantes (Recktenwald et al. 2011).

En el presente proyecto se abordó desde un punto de vista teórico y práctico el proceso de automatización de un sistema de riego por goteo subterráneo con microcontroladores, que estará ubicado en una parcela demostrativa del campus de la Universidad Nacional Agraria La Molina (UNALM).

En el sistema de riego se implementó los sensores y actuadores conjuntamente con los microcontroladores precede de un proyecto de investigación existente el cual se controlaba de forma manual, en este se identificó los elementos principales para automatizarlos. El microcontrolador que se utilizó es un Arduino Mega 2560, el cual tiene como ventaja su conectividad alámbrica e inalámbrica, para comunicarse con el programador o usuario y así sea monitoreada a través de un módulo Bluetooth enviando información a un aplicativo móvil o envíe información que sea interpretado por el microcontrolador y tome las acciones respectivas, otra ventaja es la cantidad de pines digitales y analógicos disponibles donde serán conectados los sensores y actuadores.

En la parte académica, busca mejorar la calidad de enseñanza frente a la era tecnológica presente, el sistema que se empleó tiene grandes ventajas, entre ellas, los bajos costos de implementación, tener un conocimiento básico de programación, software libre del Arduino y App Inventor, además la información requerida para el entendimiento de estos es accesible al ser plataformas libres, de esta manera puedan realizar modificaciones futuras para la mejora y el entendimiento de esta rama de la Ingeniería.

Finalmente el objetivo fue desarrollar el proceso de automatización, donde se implementó los dispositivos electrónicos al sistema de riego, previamente calibrados elaborando los códigos de programación y presentar los valores de los sensores en el interfaz del aplicativo desarrollado para el control y monitoreo.

OBJETIVO PRINCIPAL

- Implementar, controlar y monitorear un sistema de riego por goteo, utilizando microcontroladores.

OBJETIVOS ESPECÍFICOS

- Seleccionar los sensores, actuadores y el microcontrolador pertinente para la automatización de un sistema de riego, para el posterior desarrollo de la configuración y códigos de programación.
- Calibrar los sensores, integrar con los actuadores para la implementación e instalación del prototipo del sistema del riego automatizado, elaborando manuales que contribuyan al desarrollo académico del estudiante.
- Desarrollar un interfaz de vinculación entre el microcontrolador y el usuario, para el control y monitoreo de los componentes y parámetros del sistema de riego.

II. REVISIÓN DE LITERATURA

2.1. SISTEMA DE RIEGO POR GOTEO SUBTERRÁNEO

El riego por goteo subterráneo (SDI) es una variación del riego por goteo de superficie convencional. Los laterales están enterrados a una profundidad debajo de la superficie del suelo dependiendo principalmente de las prácticas de labranza y el cultivo a ser regado.

Las ventajas del SDI son amplias, a continuación se describe las tres principales:

- La eficiencia del uso del agua es alta desde la evaporación del suelo, la escorrentía superficial y la profundidad la percolación se reduce o se elimina en gran medida.
- Se mejora la eficiencia en la aplicación de agua porque los fertilizantes y pesticidas pueden ser aplicado con precisión. En cultivos ampliamente espaciados, una pequeña fracción del volumen del suelo puede humedecerse, reduciendo aún más las pérdidas innecesarias de agua de riego.
- El tiempo de cosecha puede ser mejorado debido a que el sistema no necesita ser eliminado en la cosecha ni ser reinstalado antes de plantar la segunda cosecha. Por otro lado, los laterales y porta laterales pueden experimentar menos daño, además el potencial de vandalismo también se reduce.

El SDI es un sistema que posee grandes ventajas en aplicación y eficiencia del recurso hídrico, sin embargo posee desventajas:

- La inspección de un SDI es muy difícil, debido a que está enterrado.
- La obstrucción de los emisores por las raíces y los sólidos puede causar un bajo rendimiento del sistema
- Un SDI es difícil de reparar y realizar mantenimiento a los laterales enterrados (Sinobas y Rodríguez 2005; Kalfountzos y Vyrilas 2002).

2.2. DOMÓTICA

La domótica se refiere a la automatización y control encendido, apagado, apertura, cierre y regulación, de aparatos y sistemas de instalaciones eléctricas y electrotécnicas como iluminación, climatización, persianas, puertas y ventanas motorizadas, el riego, entre otros. De forma centralizada y/o remota. Está determinado por una medición avanzada de

la infraestructura de red eléctrica con un medidor inteligente, el cual cuenta con aparatos inteligentes interconectados (Cervantes 2016).

2.2.1. Automatización

La automatización es el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas, al disminuir las necesidades de mano de obra, asegurar cosechas correctas, conociendo la información del cultivo por los sensores instalados, interpretar la información y resolver los problemas que se van presentando, facilitando en este caso el riego de una manera fiable y eficaz, esta tecnología presenta una serie de ventajas e inconvenientes. Las principales ventajas son:

- Facilitar el buen manejo del riego, pues permite conocer la situación real en cada momento, detectando anomalías en el funcionamiento de la instalación.
- Reduce la mano de obra necesaria proporcionalmente al grado de automatización y evita el error humano.
- Cuando utiliza los parámetros de manejo: condiciones climáticas, humedad en suelo o planta, permite una mayor racionalización del riego, aportando el agua cuando la planta lo requiera.
- La optimización del uso del agua permite, generalmente, una disminución de costes del riego.
- Los inconvenientes también existen:
- El principal es un mayor costo de los equipos, que para ciertas zonas o para ciertos cultivos puede hacer inviable este sistema.
- Para su correcto funcionamiento, exige una especialización del regante o de la mano de obra.
- Se debe tener en cuenta la necesidad de disponer de energía eléctrica de forma continua, para obtener información en todo momento y no interrumpir las operaciones del riego (Lion 2004).

2.2.2. Sistema electrónico

Un sistema electrónico es un conjunto de sensores, circuitería de procesamiento y control, actuadores y fuente de alimentación.

Los sensores obtienen información del mundo físico externo y la transforman en una señal eléctrica que puede ser manipulada. Existen sensores de todo tipo: de temperatura, de humedad, de movimiento, de sonido (micrófonos), etc.

Los circuitos internos de un sistema electrónico procesan la señal eléctrica convenientemente, para que con un conjunto lógico de instrucciones que dicho hardware tenga pregrabado sea capaz de ejecutar de forma autónoma.

Los actuadores transforman la señal eléctrica acabada de procesar por la circuitería interna en energía que actúa directamente sobre el mundo físico externo. Ejemplos de actuadores son: un motor (energía mecánica), una bombilla (energía lumínica), un altavoz (energía acústica), etc.

La fuente de alimentación proporciona la energía necesaria para que se pueda realizar todo el proceso descrito. Ejemplos de fuentes son las pilas, baterías, adaptadores corriente alterna (AC) o corriente continua (DC), etc. (Torrente 2013).

2.2.3. Microcontrolador

Un microcontrolador es un circuito integrado o “chip” que tiene la característica de ser programable. Es decir, que es capaz de ejecutar de forma autónoma una serie de acciones previamente definidas. Este tiene tres elementos básicos:

- Unidad Central de Proceso (CPU), es la parte encargada de ejecutar cada instrucción y de controlar que dicha ejecución se realice correctamente.
- Diferentes tipos de memorias, son en general las encargadas de alojar tanto las instrucciones como los diferentes datos que estas necesitan.
- Diferentes puertos de entrada/salida (E/S), son las encargadas de comunicar el microcontrolador con el exterior (Torrente 2013).

2.2.4. Algoritmo

Un algoritmo es una secuencia de instrucciones que representan un modelo de solución para determinado tipo de problemas, o bien como un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema.

2.3. ARDUINO

Es una placa de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra (los cuales están unidos internamente a los puertos de E/S del

microcontrolador) que permiten conectar allí de forma muy sencilla y cómoda diferentes sensores, actuadores y módulos (Bluetooth). Esta placa posee un software gratis, libre y multiplataforma (funciona en Linux, MacOS y Windows) que se instala en el ordenador y permite escribir, verificar y guardar (“cargar”) en la memoria del microcontrolador de la placa Arduino el conjunto de instrucciones que deseamos que este empiece a ejecutar (Torrente 2013).

Los proyectos Arduino pueden ser autónomos o no. En el primer caso, una vez programado su microcontrolador, la placa no necesita estar conectada a ningún computador y puede funcionar de manera autónoma si dispone de alguna fuente de alimentación. En el segundo caso, la placa debe estar conectada de alguna forma permanente (por cable USB, por cable de red Ethernet, etc.) a un computador (Torrente 2013).

Por último este microcontrolador tiene un lenguaje de programación libre. Por “lenguaje de programación” se entiende cualquier idioma artificial diseñado para expresar instrucciones, siguiendo unas determinadas reglas sintácticas que pueden ser llevadas a cabo por máquinas (Torrente 2013).

A diferencia de otras plataformas menos comunes, Arduino viene con una gran cantidad de material complementario, como tutoriales, ejemplos y otros recursos. Esto anima a los estudiantes a aprender de forma autónoma, posiblemente más allá de lo que espera el docente. Una ventaja notable de esta plataforma es la libertad de elegir los diferentes sensores para obtener los datos necesarios. Por lo tanto, uno puede usar sensores para experimentos en todas las áreas de la educación de Ciencias Físicas. Además, el entorno de desarrollo integrado de Arduino está especialmente diseñado para presentar a los recién llegados al desarrollo de software (Zieris et al. 2015).

2.3.1. Software Arduino

El código de Arduino se desarrolló en el Entorno de Desarrollo Integrado. El programa que ejecuta este código consiste en leer los puertos de los sensores, calcular el valor de los sensores debido a que estos son analógicos, y enviarlos al puerto serial (David et al. 2015).

- Entorno de Desarrollo Integrado (IDE)

Las siglas IDE vienen de “Integrated Development Environment”, lo que traducido a nuestro idioma significa “Entorno de Desarrollo Integrado”, es un conjunto de herramientas software que permite a los programadores poder desarrollar sus propios programas con comodidad. Un programa es un conjunto de instrucciones, ordenadas y agrupadas de forma adecuada que pretende obtener un resultado determinado. En el caso de Arduino, necesitamos un IDE que nos permita escribir y editar nuestro programa (también llamado “sketch” en el mundo de Arduino), que nos permita comprobar que no se haya cometido ningún error y que además permita, cuando se está seguro de que el sketch es correcto, grabarlo en la memoria del microcontrolador de la placa Arduino para que este se convierta a partir de entonces en el ejecutor autónomo de dicho programa (Herrero et al. 2015).

2.3.2. Lenguaje del Arduino

Un programa diseñado para ejecutarse sobre un Arduino (un “sketch”) siempre se compone de tres secciones:

- La sección de declaraciones de variables globales: ubicada directamente al principio del sketch.
- La sección llamada “void setup ()”: delimitada por llaves de apertura y cierre.
- La sección llamada “void loop ()”: se ejecuta justo después de la sección “void setup” delimitada por llaves de apertura y cierre

La primera sección del sketch se escribe las diferentes declaraciones de variables que necesitamos. Una variable es un nombre elegido por nosotros que guarda un determinado contenido, puede haber sido asignado literalmente, ser el dato obtenido por un sensor, o el resultado de un cálculo. El valor podrá ser siempre cambiado en cualquier instante posterior de la ejecución del sketch, si así lo deseamos, de ahí el nombre de variable.

En la segunda sección “void setup()” es la parte encargada de recoger la configuración, que debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar o inicializar pinMode (modo de trabajo de las E/S), configuración de la comunicación en puerto serie (los datos de salida se muestren en una pantalla) y otras.

En la tercera sección “void loop” contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc), esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

En resumen “void setup()” normalmente sirve para realizar ciertas preconfiguraciones iniciales y las instrucciones del interior de “void loop()” son, de hecho, el programa en sí que está funcionando continuamente (Torrente 2013).

2.3.3. Librería Arduino

Una librería es un conjunto de instrucciones de un lenguaje de programación agrupadas de una forma coherente. Las librerías sirven para facilitar el desarrollo de nuestros proyectos, debido a que a la hora de escribir nuestro programa no sea necesario conocer los detalles técnicos sobre el manejo de un determinado hardware.

2.3.4. Señales digitales

Señal digital es aquella que solo tiene un número finito de valores posibles. Por ejemplo, si consideramos como señal el color emitido por un semáforo, es fácil ver que esta es de tipo digital, porque solo puede tener tres valores concretos, diferenciados y sin posibilidad de transición progresiva entre ellos: rojo, ámbar y verde (Torrente 2013).

– Entradas y salidas digitales

En las entradas y salidas digitales es donde conectaremos nuestros sensores para que la placa pueda recibir datos del entorno, y también donde conectaremos los actuadores para que la placa pueda enviarles las órdenes pertinentes, además de poder conectar cualquier otro componente que necesite comunicarse con la placa de alguna manera.

2.3.5. Señales analógicas

Señal analógica es aquella que tiene infinitos valores posibles dentro de un rango determinado. La mayoría de magnitudes físicas (temperatura, sonido, luz, etc.) son analógicas, así como también las magnitudes eléctricas (voltaje, intensidad de corriente, potencia, etc.) porque todas ellas, de forma natural, pueden sufrir variaciones continuas sin saltos (Torrente 2013).

– Entradas analógicas

Las entradas analógicas pueden recibir voltajes dentro de un rango de valores continuos de entre 0 y 5 V. No obstante, la electrónica de la placa tan solo puede trabajar con valores digitales, por lo que es necesaria una conversión previa del valor analógico recibido a un

valor digital lo más aproximado posible, para ello el software interpreta el valor de voltaje en un rango entero de 0 y 1023. Esto implica que la resolución de la lectura es de $5V/1024$, es decir, de 0.049V (Torrente 2013).

– **Salidas analógicas (PWM)**

Las siglas PWM vienen de “Pulse Width Modulation” (Modulación de Ancho de Pulso), estas salidas se usan a menudo para enviar al entorno señales analógicas, por ejemplo, para variar progresivamente la velocidad de un motor, la frecuencia de un sonido emitido por un zumbador o la intensidad con la que luce un LED (Torrente 2013).

2.3.6. Características de una placa Arduino

a. Voltaje de funcionamiento y entrada

Es el voltaje entre dos puntos de un circuito eléctrico a la que opera el Arduino, para la mayoría de placas el voltaje de salida es de 5V y el voltaje de alimentación recomendado oscila entre 7 a 12V.

b. Velocidad de la unidad central de procesos (CPU)

El CPU es la parte del microcontrolador en la que se encuentran los elementos que sirven para procesar datos, la velocidad es en ciclos por segundo (medidas en hercios) con la que realiza las operaciones más básicas. Generalmente las placas Arduino tienen una velocidad de 16 MHz.

c. Memoria EEPROM

Es la memoria que permite mantener grabados los datos aunque la placa deje de recibir alimentación eléctrica (o se resetee).

d. Memoria SRAM

Memoria volátil donde se alojan los datos que en ese instante el programa necesita crear o manipular para su correcto funcionamiento.

e. Memoria Flash

Memoria persistente donde se almacena permanentemente el programa que ejecuta el microcontrolador (hasta una nueva reescritura si se da el caso).

f. Puerto de serie UART

Es el puerto serie hardware que todos los microcontroladores tienen al menos uno y la comunicación serie es la base de casi cualquiera de las comunicaciones de los microcontroladores (Crespo 2014).

El puerto de comunicación para la computadora es llamada tipo A, esta es la convencional que se encuentra en todo tipo de computador, mientras que el puerto de comunicación del Arduino Mega es llamado tipo B, este puerto se apreciará en la Figura 1, se precisa que el puerto es igual al conector de un cable de impresora.



Figura 1: Entrada USB PC-Arduino

FUENTE: Elaboración propia

g. Comunicación Serial

El microcontrolador dispone de un receptor/transmisor serie que permite comunicar la placa Arduino con otros dispositivos (normalmente un computador), para así poder transferir datos entre ambos. El canal físico de comunicación en estos casos suele ser el cable USB, pero también pueden ser los pines digitales 0 (RX) y 1 (TX) de la placa. Si se usan estos dos pines para comunicar la placa con un dispositivo externo, tendremos que conectar concretamente el pin TX de la placa con el pin RX del dispositivo, el RX de la placa con el TX del dispositivo y compartir la tierra de la placa con la tierra del

dispositivo, de esta forma puede interactuar a través del lenguaje Arduino llamado “Serial”, el cual utiliza al monitor serial que es una pantalla dentro del software IDE para realizar la comunicación previa activación de las funciones en la programación (Patra 2017)

2.4. ARDUINO MEGA 2560

Es un tablero de microcontroladores basado en el ATmega2560 . Tiene 54 pines digitales de entrada / salida (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente se conecta para la alimentación a una computadora con un cable USB al conector tipo B del microcontrolador, con un adaptador de corriente alterna (AC) a corriente continua (DC) a la fuente externa de 7 a 12 V, o batería para comenzar (Arduino 2017). En la Figura 2 se observa las partes del Arduino Mega 2560 y en el Cuadro 1 las principales características de las placas comerciales.

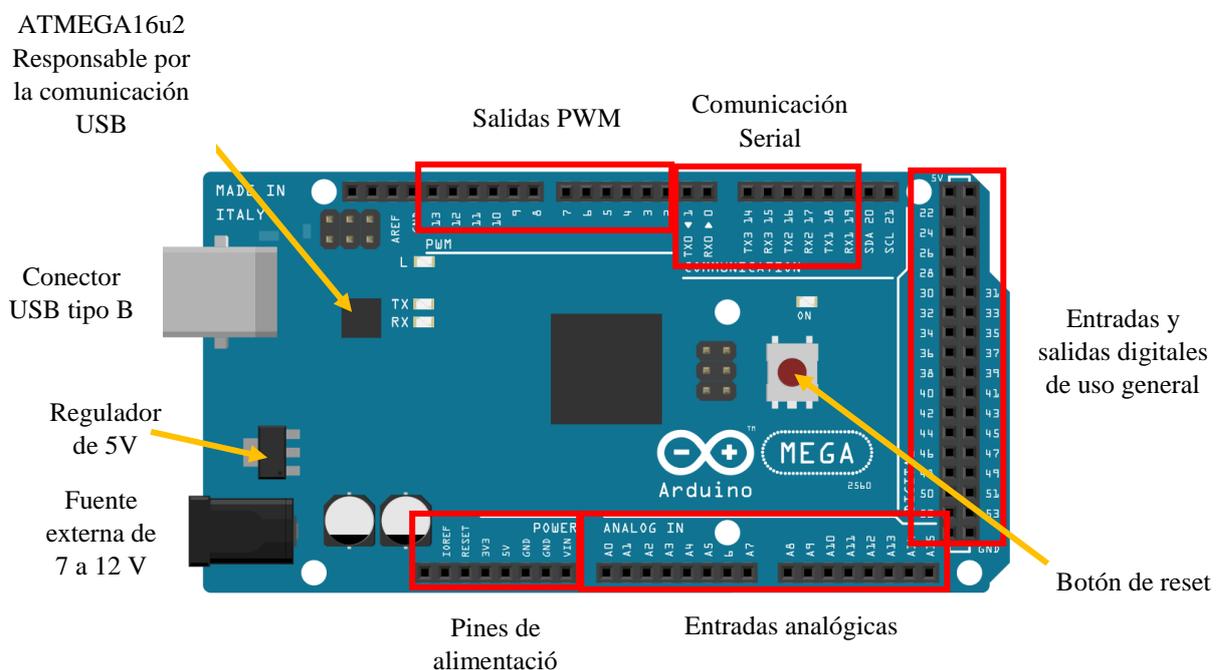


Figura 2: Características del Arduino Mega 2560
 FUENTE: Elaborado con información de Windows et al. (2014)

Cuadro 1: Características de las principales placas Arduino

Nombre	Procesador	Voltaje de funcionamiento/ entrada	Velocidad de la CPU	Entrada/salida analógica	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
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/ 5 V	48 MHz	7/1	8/4	-	32	256	Micro	1
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
Nano	ATmega168 Atmega 328P	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 400 MHz	12/0	20/7	1	2.5 16MB	32 64MB	Micro	1

FUENTE: Tomado de Arduino (2017)

2.5. SENSORES

2.5.1. Sensor de humedad GS1

La sonda de humedad del suelo GS1 está construida para durar más tiempo en el campo bajo condiciones más cálidas y húmedas. El cuerpo epoxi resiste la intrusión de agua y el sensor está diseñado para instalaciones a largo plazo, sus agujas de acero inoxidable clavadas en el suelo para mejorar el contacto suelo-sensor y facilitar la instalación. Las agujas son también más duraderas que los sensores estándar, es útil para toda la gama de humedad de contenido volumétrico (VWC), se puede instalar tanto en suelos secos del desierto como en turbas muy húmedas. Es sensible a pequeños cambios en el VWC en cualquier medio (Edaphic 2004). Este sensor se observa en la Figura 3, tiene una señal variable por ello va conectado a un pin analógico.

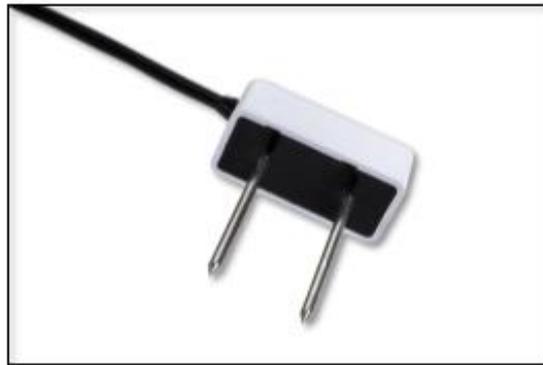


Figura 3: Sensor GS1.

FUENTE: Tomado de Edaphic (2004)

Especificaciones técnicas:

- Tiempo de medición: 10 ms (milisegundos)
- Precisión: ± 0.03 m³ / m³ en suelos típicos
- Con calibración específica del suelo: ± 0.01 a 0.02 m³ / m³ en cualquier material.
- Requisitos de alimentación: 3.0 VDC a 15 VDC (máximo absoluto) @ 15 mA
- Salida: 1,000 a 2,500 mV
- Entorno de funcionamiento: -40 a 50 °C
- Rango de medición: 0 a 57% VWC
- Dimensiones del sensor: 8.9 cm x 1.8 cm x 0.7 cm
- Tipos de conectores: conector de 3,5 mm (estéreo) o cable desforrado y estañado.

2.5.2. Sensor de flujo de agua

El sensor se acopla a la tubería de agua domiciliaria o de riego y contiene un sensor de molino para medir la cantidad de líquido que ha pasado a través de él, tiene un sensor magnético de efecto Hall (fuerza magnética que genera un voltaje) que emite un impulso eléctrico con cada revolución. El sensor de efecto Hall está sellado para permanecer seguro y seco, viene con tres cables, rojo (energía 5-24VDC), negro (tierra) y amarillo (salida de pulsos del sensor) (Mecatronics 2015).

Al contar los pulsos se puede medir fácilmente el volumen o caudal de salida del agua (el volumen promedio por pulso es de 2.25mL). La salida de pulsos es una onda cuadrada cuya frecuencia es proporcional al caudal, los pulsos son contados por el microcontrolador mediante un código para habilitar interrupciones. El factor de conversión de frecuencia (Hz) a caudal (L/min) varía entre modelos y depende de la presión, densidad, tiempo de conteo de las interrupciones e incluso del mismo caudal (Mecatronics 2015).

La orientación, presión del agua, el tiempo asignado en la programación para las interrupciones y otras condiciones pueden afectar la medición, por ello calibrar el sensor es recomendado, este sensor utiliza un pin digital. En el Cuadro 2 se muestra las características de los dos sensores de flujo de agua utilizados.

Cuadro 2: Características del sensor de flujo de 1" y ½"

Sensor de flujo	Modelo	Conexión	Caudal
	FS400A	1 "	1-60 L/min
	YF-S201	½ "	1-30 L/min

FUENTE: Tomado de Mecatronics (2015)

Especificaciones técnicas:

- Voltaje de operación: 5V - 18V DC
- Corriente de operación: 15mA (5V)
- Capacidad de carga: 10mA (5 VDC)
- Temperatura de funcionamiento: -25°C a 80°C
- Presión de funcionamiento máximo: 1.2MPa (12 bar)
- Volumen promedio por pulso: 2.25mL
- Error de medición: +_ 10 % (MaxElectronica 2018)

2.5.3. Transductor de presión

El transductor, mide la presión en tuberías, funciona transformando una magnitud física en una eléctrica, en este caso transforman una fuerza por unidad de superficie en un voltaje equivalente a esa presión ejercida, este sensor trabaja en el rango de 0 a 5 Bar, un rango común para aplicaciones experimentales en sistemas de presión constante (agua, aceite, combustible). La salida del sensor es un voltaje analógico que varía linealmente desde 0.5V para 0 Bar hasta 4.5V para 8 Bar. Esta salida analógica es compatible con las entradas analógicas (ADC) de Arduino (Mechatronics 2017). El sensor se observa en la Figura 4, el pin al que se conecta es analógico.



Figura 4: Sensor de presión HK3022.

FUENTE: Tomado de Mechatronics (2017)

Especificaciones Técnicas:

- Voltaje de operación: 5.0 VDC
- Voltaje de salida: 0.5 - 4.5 VDC
- Corriente de funcionamiento: ≤ 10 mA
- Rango de presión de trabajo: 0 - 5 Bar
- Presión máx: 15 Bar
- Error de medición: +_ 1 %

2.5.4. Sensor ultrasonido HC-SR04

El sensor HC-SR04 es un sensor de distancia de bajo costo que utiliza ultrasonido para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y bajo precio.

El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación. El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido(40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y así se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar) (Naylamp 2018). Este sensor se observa en la Figura 5, utiliza pines digitales.

La distancia se puede calcular utilizando la siguiente formula:

$$\text{Distancia (m)} = \{(\text{Tiempo del pulso ECO}) * (\text{Velocidad del sonido}=340\text{m/s})\}/2$$



Figura 5: Sensor de ultrasonido HC-SR04.

FUENTE: Tomado de Naylamp (2018)

Especificaciones Técnicas:

- Voltaje de Operación: 5V DC
- Corriente de reposo: < 2mA
- Corriente de trabajo: 15mA
- Rango de medición: 2cm a 450cm
- Precisión: +- 3mm

- Ángulo de apertura: 15°
- Dimensiones: 45mm x 20mm x 15mm
- Tiempo mínimo de espera entre una medida y el inicio de otra 20ms (recomendable 50ms)

2.6. ACTUADORES

2.6.1. Módulo relé

El relé permite controlar componentes de alto amperaje o alto voltaje, los cuales no se puede controlar directamente desde el Arduino, con este módulo se controla motores AC (220V), motores DC, solenoides de electroválvulas, electroválvulas y una gran variedad de actuadores.

El modulo es capaz de manejar hasta 250V. Cada canal posee un optoacoplador para un uso seguro y un Led indicador de estado (Mecatronics 2017). Este actuador se observa en la Figura 6, usa puertos digitales.



Figura 6: Modulo Relé de 4 canales.

FUENTE: Tomado de Mecatronics (2017)

Especificaciones Técnicas:

- Voltaje de Operación: 5V DC
- Corriente de activación por relé 15mA~20mA
- N° de Relés: 4, 8, etc.
- Capacidad máx: 10A/250VAC, 10A/30VDC
- Corriente máx: 10A (NO, normalmente abierto), 5A (NC, normalmente cerrado);
- Tiempo de acción: 10 ms / 5 ms.

2.6.2. Válvula eléctrica

La Válvula solenoide 1/2" 12V DC permite controlar el flujo de agua en una tubería. Las válvulas solenoides son un tipo de electroválvula todo, nada o abierto, cerrado. Tienen dos partes: el solenoide y el cuerpo de plástico.

El solenoide es un electroimán que al ser energizado se desplaza junto con el diafragma de la válvula y permite el paso del fluido. La válvula se mantiene abierta mientras el solenoide está energizado. Cuando no está alimentado un resorte se encarga de regresar la válvula a su posición de reposo, que en este caso es del tipo normalmente cerrada (NC) (Mechatronics 2017). Esta válvula está conectada con el relé, quien controla su funcionamiento, se observa en la Figura 7.



Figura 7: Válvula Eléctrica de ½"

FUENTE: Tomado de Mechatronics (2017)

Especificaciones Técnicas:

- Voltaje de operación: 12V DC
- Corriente de operación: 0.6A
- Potencia consumo: 8W
- Presión de funcionamiento mínima: 0.2 Bar
- Presión de funcionamiento máximo: 8 Bar
- Tiempo de respuesta (apertura): ≤ 0.15 s

2.7. MÓDULO BLUETOOTH

El módulo serie Bluetooth se utiliza para comunicarse con la placa Arduino y después de la comunicación entre el módulo Bluetooth y la placa Arduino, el LED parpadeará en la placa Arduino mostrando la completa conectividad. El Módulo Bluetooth HC05 es altamente compatible y altamente exigente en el mercado en comparación con otros Módulos Bluetooth, estas operaciones de alto rendimiento implican más fiabilidad y flexibilidad en el funcionamiento en comparación con los otros módulos. En general, el módulo Bluetooth

funciona en corta distancia, pero en comparación a Bluetooth Zigbee opera a una velocidad más alta, siendo el HC-05 hasta 3000 kbit/s mientras que el modelo Zigbee hasta 250 kbit/s (Patra 2017).

2.8. APP INVENTOR

App Inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web (Almaraz et al. 2015).

2.8.1. Historia

En la creación de App Inventor para Android, Google se basó en la investigación previa importante en la informática educativa, y el trabajo realizado en entornos de desarrollo de Google en línea.

El editor de bloques utiliza la biblioteca de bloques abiertos de Java para la creación de lenguajes de programación de bloques visuales.

El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza el marco Kawa idioma y el dialecto de Kawa del lenguaje de programación Scheme, desarrollada por Per Bothner y distribuido como parte del sistema operativo GNU por la Free Software Foundation.

En agosto de 2011, Google anunció que App Inventor se suspendería como un producto de Google, y que el código se convertirá en abierto originario. Ahora es parte del Centro Massachusetts Institute of Technology (MIT) para el aprendizaje móvil en Media Lab, dirigido por App Inventor creador Hal Abelson, junto con su compañero de profesores Eric Klopfer y Mitchel Resnick (López 2015).

2.8.2. Características

El editor de bloques de la aplicación utiliza la librería Open Blocks de Java para crear un lenguaje visual a partir de bloques. Estas librerías están distribuidas por Massachusetts Institute of Technology bajo su licencia libre (MIT License). El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza Kawa como lenguaje de programación, distribuido como parte del sistema operativo GNU de la Free Software Foundation.

App Inventor pueden tener su primera aplicación en funcionamiento en una hora o menos, y se pueden programar aplicaciones más complejas en mucho menos tiempo que con los lenguajes más tradicionales, basados en texto. Una herramienta de código abierto que pretende realizar la programación y la creación de aplicaciones accesibles a una amplia gama de audiencias (López 2015).

2.8.3. Forma de uso

Es una herramienta muy útil de programación, el cual permite el desarrollo de aplicaciones para dispositivos móviles que tengan como Sistema Operativo Android, este desarrollo es mediante Internet, el cual se ejecuta en el dispositivo y los cambios se guardan en línea.

Además que para crear el código, es necesario unir las piezas, cada parte del código que se necesitan está en forma de rompecabezas, lo cual lo hace más interesante y fácil para las personas que empiezan a familiarizarse con la programación o creación de aplicaciones para el Sistema Operativo Android.

Esta herramienta cuenta con dos secciones principales: Modulo Web y el Editor de Bloques (López 2015).

Modulo Web: es la parte en donde se tiene acceso a los proyectos, en la cual se puede añadir los componentes, así como la configuración de los mismos. Allí mismo se puede ver la vista previa de cómo se observara en el dispositivo a instalarse. Para que el desarrollador se familiarice con el desarrollo de aplicaciones Android (López 2015).

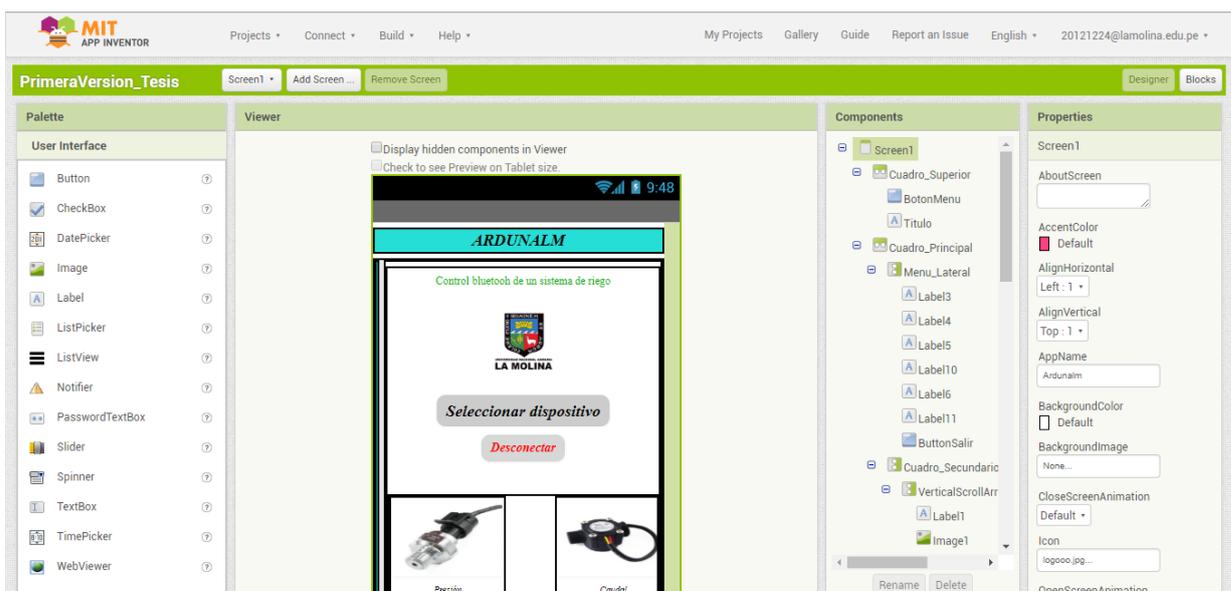


Figura 8: Pantalla principal de App Inventor.

FUENTE: Tomado del software App Inventor

En la pantalla principal del entorno que se observa en la Figura 8, en el centro se visualiza una pantalla de un dispositivo móvil en la cual se añade todos los objetos que se crean convenientes para el desarrollo del aplicativo, tales como botones, imágenes, listas, títulos, etc.

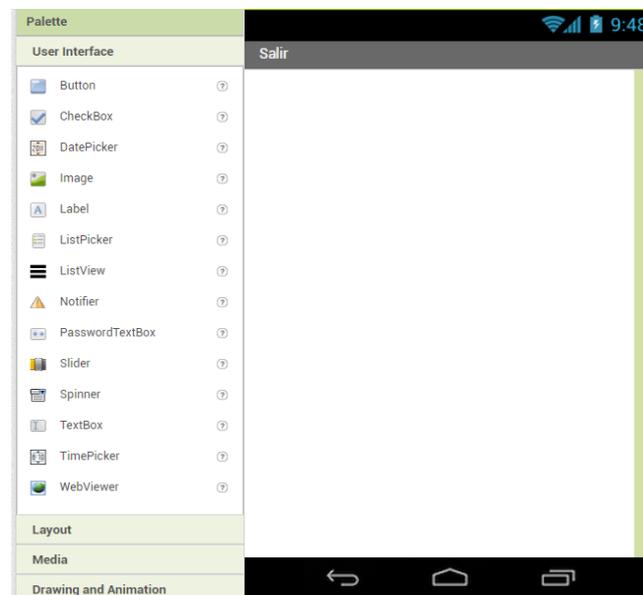


Figura 9: Elementos a utilizar y vista previa de app inventor

FUENTE: Tomado del software App Inventor

En la parte izquierda se visualiza los elementos a utilizar para la creación del aplicativo y a la derecha las propiedades de dichos elementos.

En la Figura 9, se observa los principales elementos como botones, casilleros para introducir textos, imágenes, títulos, listas, notificaciones, gráficos, etc. estos se arrastran hacia la pantalla para que sean añadidos y de esa manera creando el aplicativo.

Los elementos que se crean, poseen propiedades como posición de texto, color, tipo de letra y otras propiedades según el elemento a utilizar, estas propiedades se observan en la Figura 10.

Editor de Bloques: Es el encargado de realizar la conexión entre el teléfono para ser probada la ejecución de la aplicación. En caso de no contar un teléfono con Android, es posible que la prueba se pueda realizar mediante un emulador que se ejecutara en el mismo editor de bloques. Sin embargo se tiene que tomar en cuenta que al probarse en el emulador, no todas las funciones funcionarían de la manera correcta (López 2015). El editor de bloques se observa en la Figura 11.

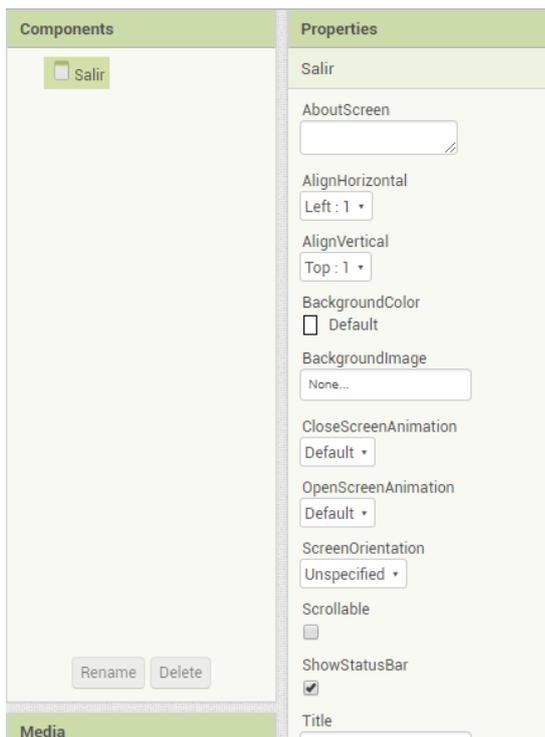


Figura 10: Propiedades de elementos.

FUENTE: Tomado del software App Inventor

Existen varios bloques de códigos con los cuales se crean los códigos necesarios para la aplicación que se está realizando. Como se mencionaran a continuación:

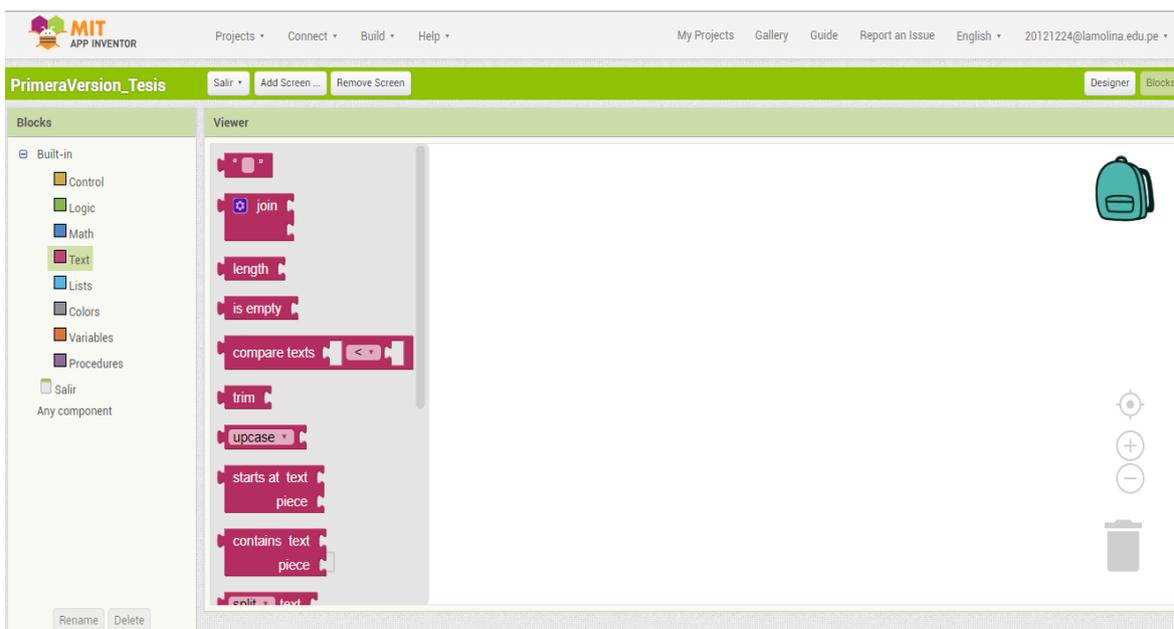


Figura 11: Editor de bloques.

FUENTE: Tomado del software App Inventor

Códigos de control: Son los códigos donde se encuentran a los comandos como el if, else, while-do, if-then-else, entre otros.

Códigos de lógica: Son los códigos donde se encuentran el true, false, not, los signos de comparación, etc.

Códigos de matemáticas: Son los códigos donde se encuentran los comandos para realizar una suma, resta, multiplicación, división, los aleatorios, los cuadrados, etc.

Códigos de texto: Son los códigos donde se encuentran los comandos para realizar comparación entre textos, líneas de palabras, una longitud, entre otros.

Códigos de variables: Son los códigos donde se encuentran los comandos para inicializar variables o números locales, para obtener datos e inicializar variables globales.

Códigos de lista: Son los códigos donde se encuentran comandos para crear listas, la longitud de una lista.

Códigos de procedimientos: Son los códigos donde se encuentran los comandos para realizar procedimientos u obtener los resultados de un procedimiento (López 2015).

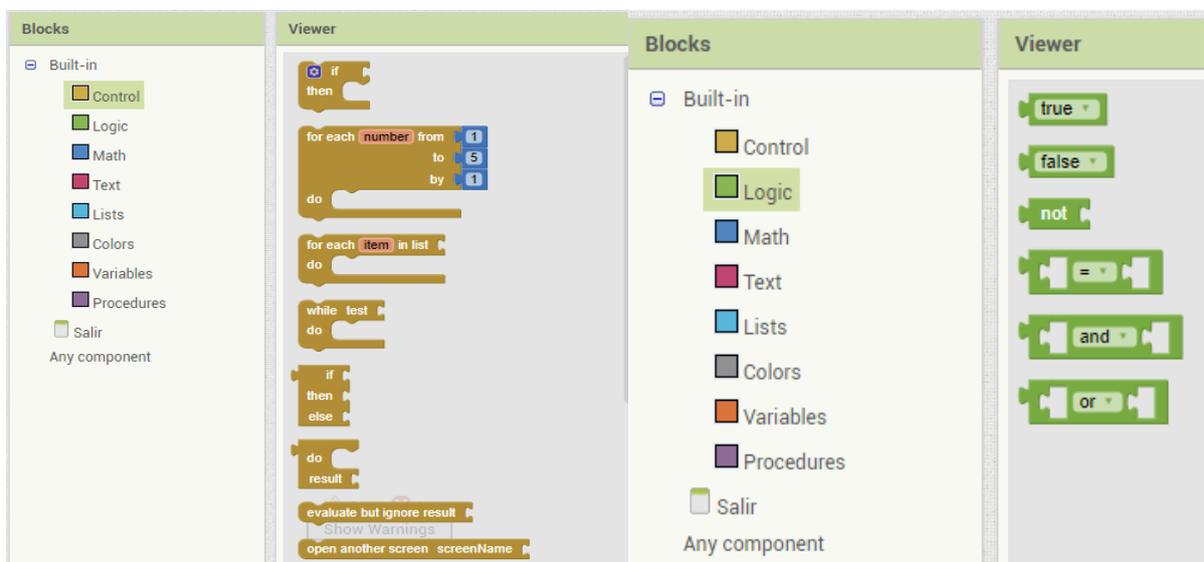


Figura 13: Códigos de control.

Figura 12: Códigos de lógica.

FUENTE: Tomado del software App Inventor

FUENTE: Tomado del software App Inventor

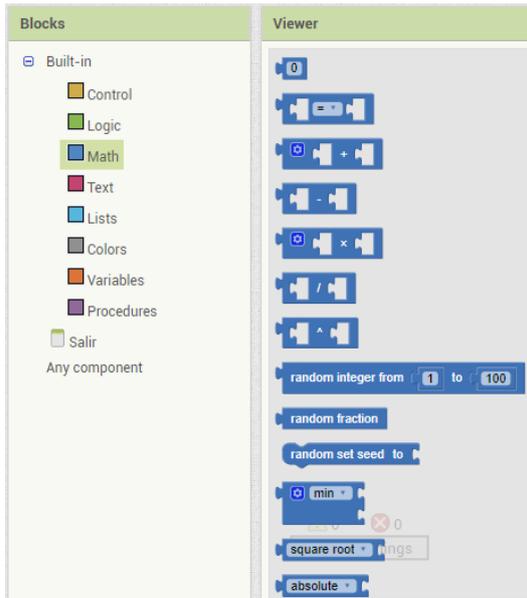


Figura 14: Códigos de matemáticas.

FUENTE: Tomado del software App Inventor

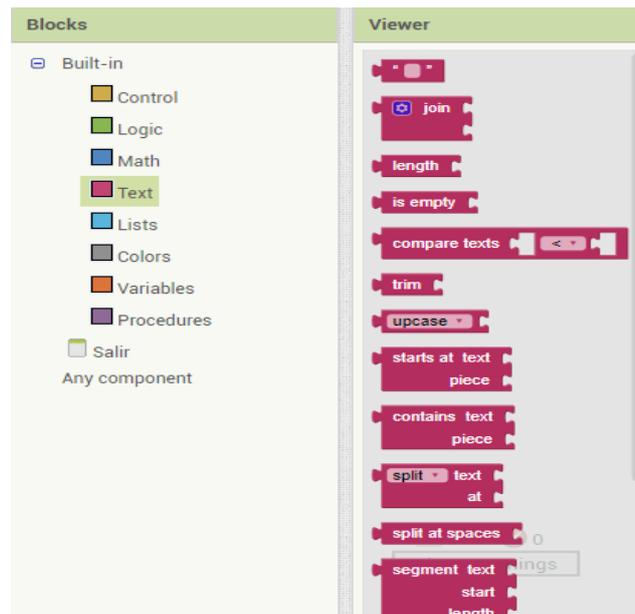


Figura 15: Códigos de texto.

FUENTE: Tomado del software App Inventor

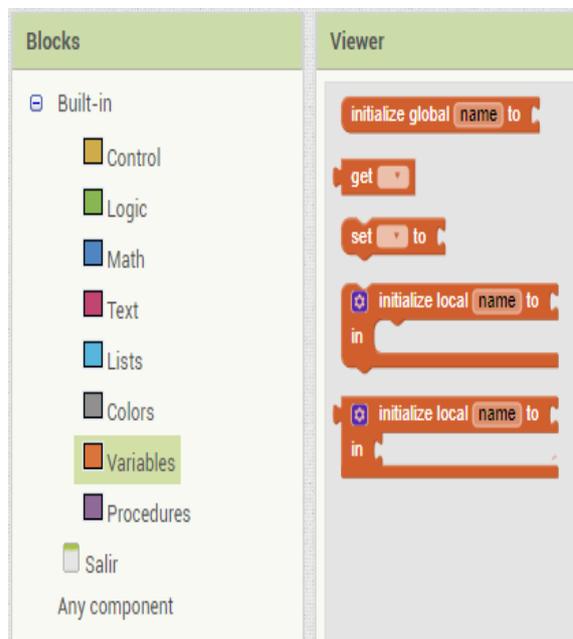


Figura 16: Códigos de variable.

FUENTE: Tomado del software App Inventor



Figura 17: Códigos de lista.

FUENTE: Tomado del software App Inventor

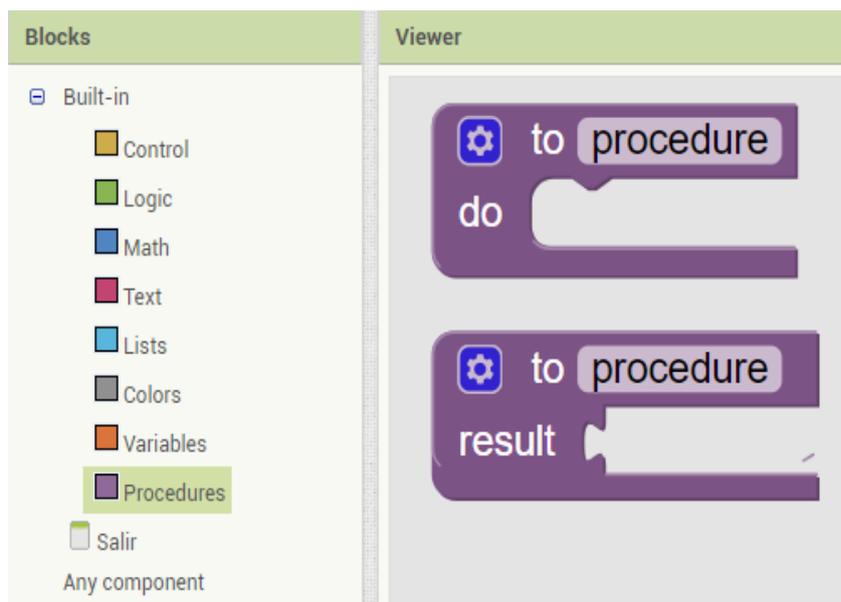


Figura 18: Códigos de procedimientos.

FUENTE: Tomado del software App Inventor

III. MATERIALES Y MÉTODOS

3.1. UBICACIÓN DEL ÁREA EXPERIMENTAL

El proyecto se llevó a cabo en la parcela demostrativa, perteneciente a la Facultad de Ingeniería Agrícola de la Universidad Nacional Agraria La Molina.

Departamento:	Lima
Provincia:	Lima
Distrito:	La Molina
Longitud:	12°4'49''
Latitud:	72°56'58''
Altitud:	245 msnm

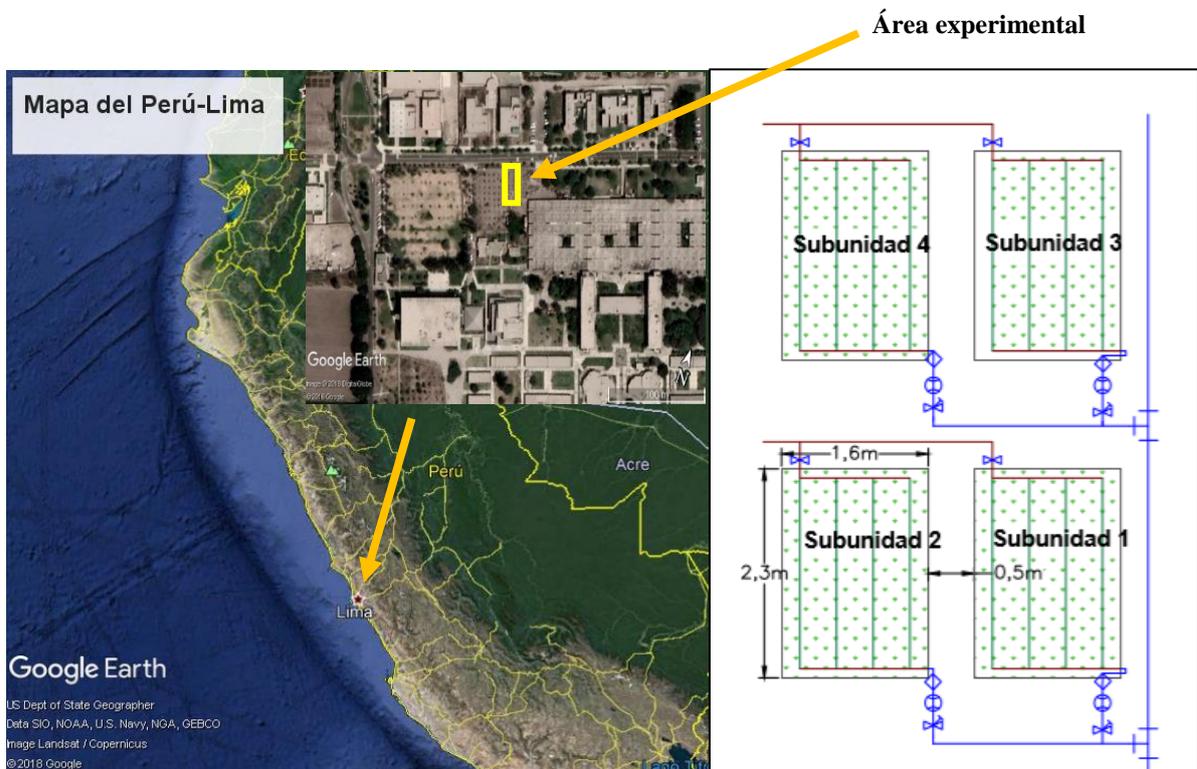


Figura 19: Ubicación geográfica del área destinada a la unidad experimental.

FUENTE: Elaboración propia

3.2. ANTECEDENTES.

El presente trabajo se realizó en un sistema de riego presurizado precedente de una investigación titulado, “Requerimiento de agua para el césped americano (*Stenotaphrum secundatum*) empleando riego por goteo subterráneo, durante el otoño en la UNALM”, en el cual se aprovechó los componentes que estaban in situ para automatizarlos.

El sistema de riego presenta componentes similares al de un sistema de riego por goteo superficial, sin embargo, en el riego por goteo subterráneo (RGS) el uso de un sistema de filtrado, de válvulas de aire y de líneas de purga son requisitos indispensables para que el sistema tenga un buen rendimiento y una duración a lo largo del tiempo. Las características del suelo es de textura franca, humedad volumétrica a capacidad de campo de 0.28 m³/m³, mientras que el punto de marchitez permanente es de 0.14 m³/m³ y una densidad aparente de 1.60 gr/cc. Los componentes principales en el cabezal de riego y subunidad son:

- Gotero auto compensado de caudal 2.3 L/h Rain bird modelo XFS0612500
- Electrobomba Pedrollo modelo CPm 610X de 0.85 HP
- Dos filtros en paralelo
- Válvulas de aire
- Tomas de presión antes y después de los filtros.
- Válvula volumétrica
- Válvula plasson para cada subunidad
- Válvula de aire
- Líneas de purga
- Fuente de agua, tanque Eternit de 1100L (Cifuentes, 2016) .

Estos componentes fueron analizados para la implementación de diversos sensores y actuadores que hacen el control y monitoreo del sistema de riego más sencillo y factible para el usuario, siendo los componentes automatizados los siguientes:

- Bomba y válvulas encendidas de manera automática
- Presiones y caudales controladas por sensores de presión y flujo de agua respectivamente, que son monitoreadas en tiempo real a través de un aplicativo móvil.
- Humedad de suelo monitoreada a través de un sensor comercial GS1

Actualmente se encuentran automatizadas cuatro válvulas del área experimental, donde el control y monitoreo del sistema de riego se da mediante el aplicativo móvil.

a. Fuente de agua

En este sistema de riego se utiliza como fuente de agua un tanque de agua de 1100 L, su llenado es de forma manual desde una válvula de acople rápido. En la Figura 20, se aprecia el tanque antes de ser automatizado para su control de nivel de agua. La altura del tanque (H) es de 1.40 metros y su diámetro (D) es de 1.10 metros.

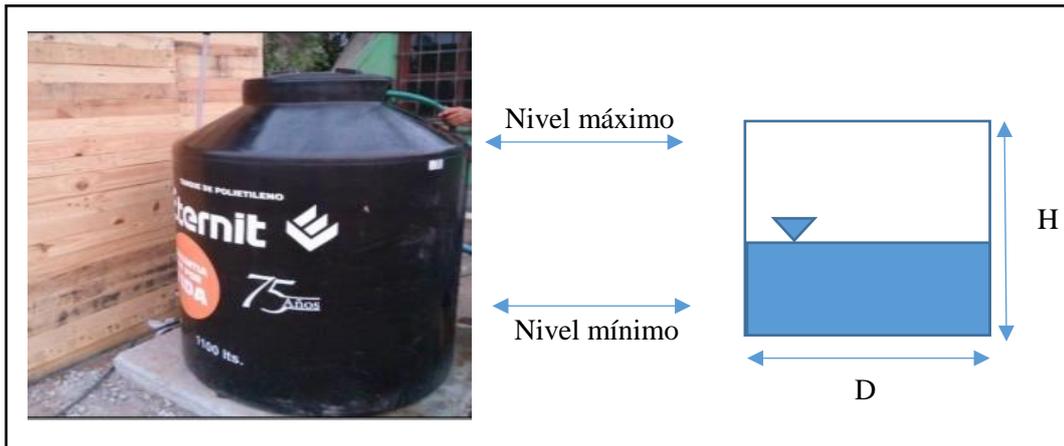


Figura 20: Reservorio de agua.

FUENTE: Elaboración propia

b. Cabezal de riego

El cabezal está ubicado en una caseta en el área experimental, donde se operaba de forma manual, se apreciaba los siguientes componentes del sistema de riego:



Figura 21: Cabezal de riego y sus principales componentes.

FUENTE: Elaboración propia

- Una bomba hidráulica
- Filtros de anillo
- Puntos de toma de presión
- válvula volumétrica
- Dos válvulas check
- Una válvula de globo para permitir el paso del agua a las parcelas experimentales

c. Subunidad de riego

Este sistema de riego comprendía de 12 sub unidades de 1.6 x 2.3m, cuya operación era manual, cada arco de riego estaba conformado por una válvula tipo plason de 1/2", una válvula de aire de doble efecto de 1", un rociador UNISPRAY 1/2''x 4, y un punto de toma de presión, la velocidad de aplicación es de 19.17 mm/hr, siendo un caudal calculado por cada subunidad de 0.02 l/s, para este proyecto que se automatizaron 4 subunidades.

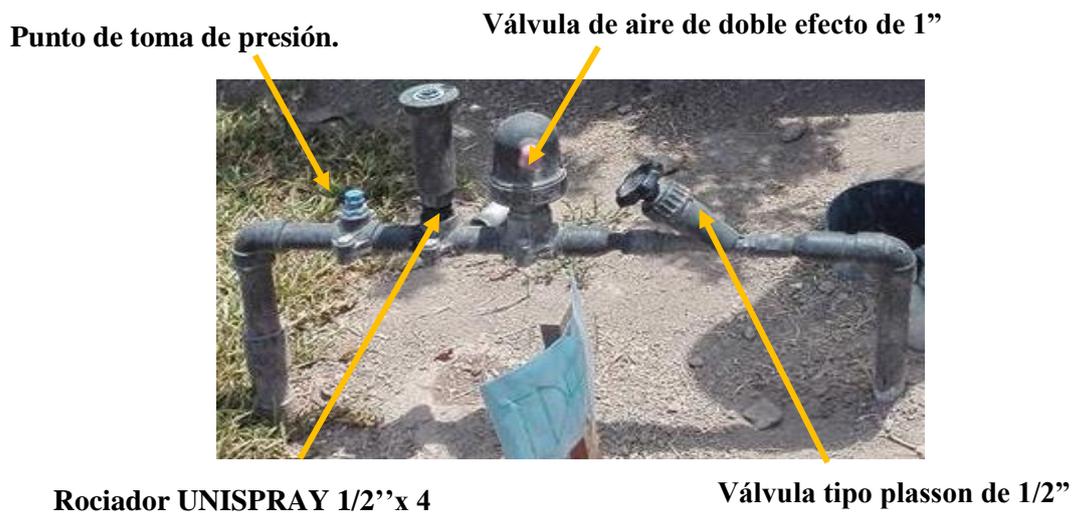


Figura 22: Arco de riego.

FUENTE: Elaboración propia

3.3. MATERIALES Y DISPOSITIVOS UTILIZADOS EN EL PROYECTO

Materiales de campo

- Pala
- Pico
- Rastrillo
- Libreta de campo
- Cable eléctrico N°18 AWG

- Tubería polietileno y corrugada con sus respectivos accesorios
- Cajas pase

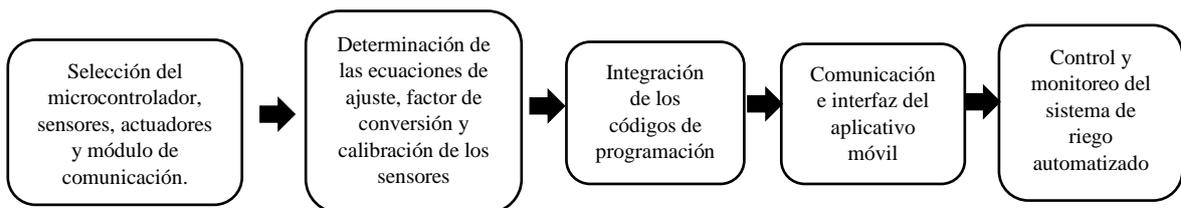
Dispositivos

- 01 Arduino Mega 2560
- 01 Sensor GS1
- 06 Sensor de presión HK3022
- 01 Sensor ultrasonido HC-SR04
- 05 Sensor de flujo de agua
- 06 Módulo relé
- 04 Válvula Eléctrica 12 voltios
- 05 Adaptador de voltaje de 12 voltios y 1.5 Amperios
- 01 Módulo Bluetooth HC-05

El costo para automatizar el cabezal, cuatro subunidades y el reservorio asciende a 3,188 nuevos soles, tomando en cuenta que existe un sistema de riego presurizado instalado, este monto contempla dispositivos electrónicos de reemplazo. En el Anexo 7 se encuentra el presupuesto desagregado.

3.4. MÉTODOS

Se desarrolló las siguientes etapas durante el proceso de automatización



3.4.1. Selección del microcontrolador y entorno de programación

El desarrollo de códigos propios para el funcionamiento de sensores y actuadores, para transmitir la información a través de señales inalámbricas, se trabajó con el hardware de nombre comercial Arduino Mega 2560 y el software del propio microcontrolador IDE, que tiene como ventaja su descargar libre, dominio público, así como la información necesaria en la internet para aprender a usarla, el modelo de Arduino que se utilizó posee la cantidad necesaria de pines para los 19 dispositivos que se implementaron al sistema de riego. Conjuntamente a este entorno de programación y microcontrolador, se trabajó con el

software app inventor, para la creación de un aplicativo móvil para controlar y monitorear los actuadores y sensores respectivamente que son procesados por el microcontrolador.

3.4.2. Selección del tipo de comunicación

La comunicación fue mediante Bluetooth HC-05, compatible con el microcontrolador, este permitió comunicar la aplicación creada con el microcontrolador, para poder recibir y enviar información en tiempo real.

3.4.3. Variables a monitorear y dispositivos a controlar

Las variables a controlar y monitorear, fueron establecidos de acuerdo a las necesidades para evaluar distintos parámetros de riego, a continuación se mencionaran cada uno de ellos:

- El monitoreo de la presión antes y después de los filtros, es una necesidad para tomar la decisión de la limpieza de estos, así como el registro de las presiones a nivel de subunidad para determinar la pérdida de presión del cabezal hasta la subunidad y por último a nivel de subunidad para la correcta presión de los emisores según lo requerido.
- El monitoreo del caudal en el cabezal y a nivel de subunidad, es necesario para llevar un monitoreo en tiempo real y detectar posibles fugas de agua u obstrucción de goteros.
- El monitoreo de la humedad, esencial para determinar el tiempo de riego y monitoreada diariamente, conllevó a una aplicación más eficiente del recurso hídrico.
- El monitoreo del nivel de agua del tanque, debido a que la fuente de agua es un tanque de polietileno negro que no permite visualizar el nivel del agua, se controló esta variable para evitar la succión de aire al estar el tanque vacío o la alerta cuando esté lleno.
- El control de los relés asociados a la bomba y válvulas, permitió la fácil operación en el sistema de riego, controlando el funcionamiento o no de los dispositivos.

El sistema automatizado consistió en lo siguiente:

- Medición de humedad en el suelo.
- Encendido y apagado de circuitos 220v (Bomba).
- Encendido y apagado de circuitos 12v (Electroválvula).
- Medición de la presión en la red de tuberías.
- Medición del caudal repartido a nivel de cabezal y válvula.
- Alerta de nivel de agua en el tanque y monitoreo del volumen de agua disponible.

3.4.4. Selección de los sensores y actuadores

- El sensor de presión HK3022 se escogió por su compatibilidad con el microcontrolador, pues tiene un voltaje de funcionamiento de 5V, un rango de presión hasta 50 metros, es un sensor analógico y para su implementación en el sistema se calibró en el banco de pruebas del laboratorio de riego y drenaje de la UNALM.
- El sensor de flujo se utilizó en dos diámetros diferentes. El de 1" que se instaló en el cabezal de riego debido a que la tubería matriz era del mismo diámetro, y en las cuatro válvulas automatizadas se instaló sensores de ½" pulgada.
- Para el registro de humedad se escogió el sensor GS1 Decagon, el cual es un sensor comercial y compatible con el microcontrolador, además de tener una estructura robusta, y de mayor durabilidad.
- El nivel del agua se automatizó mediante un sensor de ultrasonido, que mide la distancia del sensor hacia el primero cuerpo que se presente, esta medida es mediante ondas sonoras, como el llenado del tanque es manual, para evitar posibles accidentes con el sensor energizado y no generar error por la turbulencia del agua dentro del tanque, se instaló un vaso comunicante, con un tubo de alcantarillado de 160 mm, su diámetro debido a que el sensor tiene un ángulo de apertura de 15° y así minimizar el error de mediciones. En la Figura 23 se representa el vaso comunicante.

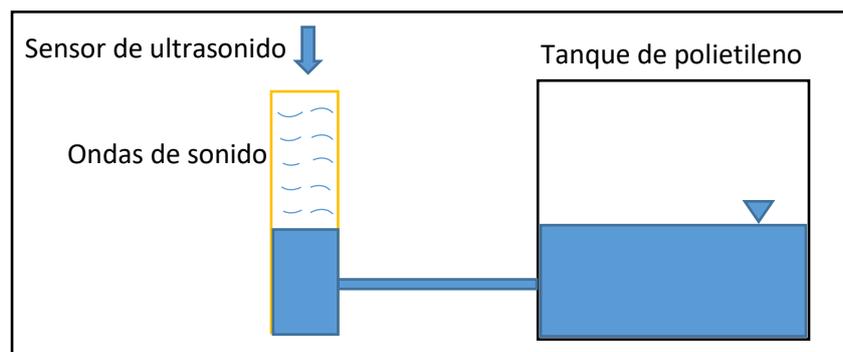


Figura 23: Vaso comunicante.

FUENTE: Elaboración propia

- El control de la bomba se dio mediante dos relés, este dispositivo es similar a un interruptor que permite el paso de la corriente, con la ventaja que se puede accionar mediante el comando del microcontrolador. Para encender la bomba se utilizaba una botonera mecánica con un circuito normalmente abierto (NO) que estaba conectado al

contactor, este al ser presionada, la bobina del contactor se energiza permitiendo el paso del fluido eléctrico y encendiendo la bomba, esta botonera fue reemplaza por un relé con las misma característica que es de un circuito normalmente abierto; para el apagado de la bomba se utilizaba otra botonera mecánica, en este caso circuito normalmente cerrado (NC) que permitía el paso de energía, hasta que al presionarlo, activaba el relé térmico, el cual cortaba el fluido eléctrico, apagando la bomba, de igual forma esta botonera fue cambiada por un relé con un circuito normalmente cerrado, el diagrama eléctrico de las botonera se aprecia en la Figura 24. El encendido y apagado de las válvulas solenoides realizaron por medio de relés con un circuito normalmente abierto y adaptadores de voltaje, debido a que las válvulas solenoide funcionaban a 12V.

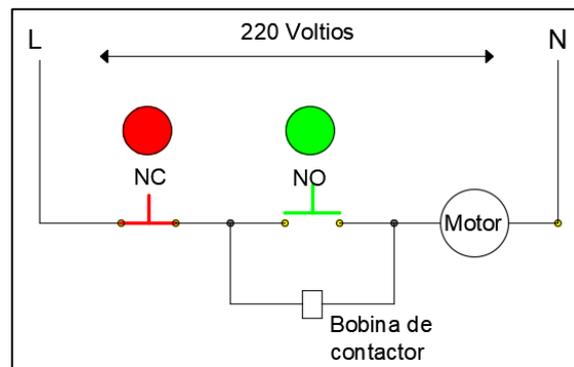


Figura 24: Diagrama eléctrico de las botoneras del tablero de control de la bomba.

FUENTE: Elaboración propia

3.4.5. Proceso de calibración de los sensores

La programación de los dispositivos es uno de los procesos más importantes, debido a que en este se desarrolló el código de programación de cada sensor y actuador individualmente para corroborar su funcionamiento, se realizó ensayos por separado de cada módulo en el banco de pruebas, comprobando la buena interacción con la placa Arduino Mega.

a. Calibración del transductor de presión

Los transductores de presión se hallaron sus ecuaciones de ajuste haciendo uso de un collarín de doble entrada instalados en una tubería de PVC, en la primera entrada se instaló un punto de prueba, mientras que la segunda el transductor de presión. La salida de medición del transductor es en bits, tomando un registro con el manómetro desde los 5 metros de presión hasta los 20 metros, siendo el intervalo cada metro, con los datos se halló la ecuación para la relación de bits y la presión en metros, posteriormente a la instalación se calibró.

b. Calibración del sensor de caudal

Para hallar el factor de conversión para la transformación de frecuencia o número de pulsos a caudal, el sensor se instaló en una red de tuberías del banco de pruebas, para poder medir el volumen de agua en un recipiente de 20 litros. Las mediciones se realizaron cada 5 litros y tres repeticiones, para obtener un promedio y tener mayor precisión en el factor de conversión, estos pulsos son contados por el software IDE previa programación, de esta manera comparando la cantidad de pulsos y el volumen de agua del recipiente en un mismo tiempo. Para calcular el factor de conversión con estas dos variables podemos usar la siguiente formula:

$$K = \frac{n^{\circ}Pulsos}{Volumen * 60}$$

Donde:

K: es el factor de corrección

N° Pulsos: Número de pulsos leídos en el monitor serial

Volumen: Litros de agua en el recipiente en el mismo instante de la visualización del número de pulsos.

El factor hallado se utilizara en la siguiente fórmula para determinar el caudal en litros por segundo del sensor, el número 60 es una constante que se utiliza para que las unidades del tiempo se mantengan en segundos.

$$Q = \frac{N^{\circ}pulsos}{K * 60}$$

Los sensores instalados en el sistema de riego, se calibraron mediante la comparación con la lectura de las válvulas volumétricas.

c. Calibración del sensor de humedad

En la parcela experimental, el sensor fue enterrado a 10 cm de la superficie, el área fue saturada para extraer muestras de suelo y tomar la medida en bits que arroja el sensor, obteniendo una ecuación que relacione la humedad del suelo con los bits tomados en cada muestra. Durante los días de riego se extrajeron muestras de suelo con la finalidad que sea calibrado con el valor de humedad que mostraba el sensor.

d. Calibración del sensor de ultrasonido

Este sensor mide la distancia a través de la velocidad del sonido y el tiempo de retorno que contabiliza el propio microcontrolador, mediante la fórmula $\text{Distancia (m)} = \{(\text{Tiempo del pulso ECO}) * (\text{Velocidad del sonido}=340\text{m/s})\}/2$, se realizó el ensayo con la tubería de alcantarillado con un diámetro de 160mm que se utilizó para el vaso comunicante con el tanque de polietileno donde fue instalado el sensor, fue instalado para monitorear el nivel de agua debido a que el llenado es de forma manual, de esta manera evitar que rebalse el agua del tanque, así también para evitar la succión de aire cuando el agua se agote.

e. Integración

El proceso de integración consistió en unir los códigos de programación de cada sensor y actuador en un solo sketch para que pueda interactuar con el dispositivo móvil. Además en esta etapa se incluyó las condicionales para enviar alertas en caso de un nivel de agua mínimo o máximo en el tanque de polietileno, para que el usuario tome acciones.

3.4.6. Proceso de programación del microcontrolador y aplicativo móvil

El proceso de programación se da inicio con el Hardware de control, el cual es el microcontrolador Arduino Mega 2560, este es el circuito impreso que recibe y envía la información, de los sensores y actuadores respectivamente. En la Figura 25, se observa los sensores acoplados al microcontrolador y el actuador usado para prender y apagar las válvulas solenoides y bomba.

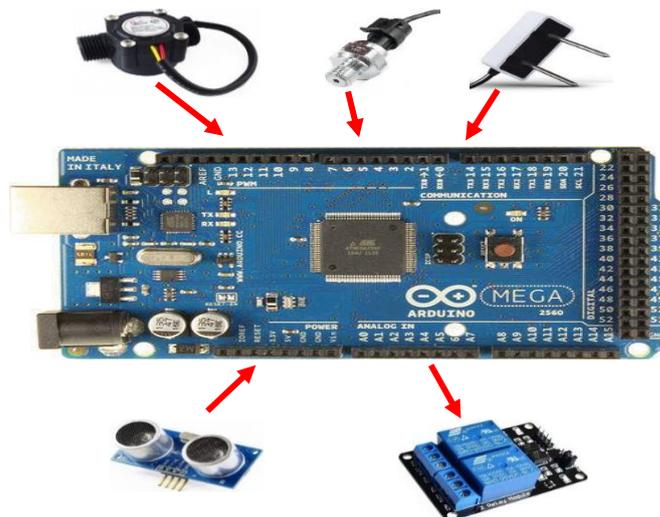


Figura 25: Microcontrolador Arduino, sensores y actuador.

FUENTE: Elaboración propia

El software usado para la programación de las acciones es propia de la tarjeta Arduino (IDE), en esta se define las acciones de cada sensor y actuador. Como producto final de la programación se imprime los datos recepcionados de cada sensor, mediante el comando “Serial.print()”, este se visualiza en la Figura 26.

```
Sistema__de_riego_autom_tico_de_control_y_monitoreo
255
256 Serial.print(presion1_m); //se imprime los
257 Serial.print(","); // se añade la separacio
258 Serial.print(presion2_m);
259 Serial.print(",");
260 Serial.print(presion3_m);
261 Serial.print(",");
262 Serial.print(presion4_m);
263 Serial.print(",");
264 Serial.print(presion5_m);
265 Serial.print(",");
266 Serial.print(presion6_m);
267 Serial.print(",");
268 Serial.print (caudal_L_s2,3);
269 Serial.print(",");
270 Serial.print (caudal_L_s3,3);
271 Serial.print(",");
272 Serial.print (caudal_L_s4,3);
273 Serial.print(",");
274 Serial.print (caudal_L_s5,3);
275 Serial.print(",");
276 Serial.print (caudal_L_s6,3);
277 Serial.print(",");
278 Serial.print(humedad);
279 Serial.print(",");
280 Serial.print(distancia);
281 Serial.print(",");
282 Serial.println(volumenlitros);
283
```

Figura 26: Comando Serial.print().

FUENTE: Tomando del software IDE

Los datos impresos se visualizan en una pantalla propia del mismo software IDE, el cual se le denomina “Monitor Serial”, estos datos son impresos en forma horizontal hasta culminar el orden en que fueron establecidos para que se impriman, posterior se vuelve a imprimir en la siguiente fila, en el mismo orden, este proceso se visualiza en la Figura 27.

Los datos impresos separados por comas en el “Monitor Serial” del software IDE, indican el orden en el cual se están imprimiendo, de esta manera vayan a la casilla donde se les ubico en el aplicativo, esta parte es la recepción de información la cual es vía bluetooH.

```

COM4 (Arduino/Genuino Mega or Mega 2560)
55.60,53.95,40.21,33.96,29.35,22.32,0.231,0.133,0.167,0.153,0.000,10.45,126.96,123.94
93.53,117.65,103.12,118.42,137.99,142.83,0.255,0.133,0.152,0.168,0.000,30.58,126.58,127.53
57.28,66.82,57.02,59.51,64.40,63.31,0.255,0.133,0.167,0.153,0.000,18.25,125.65,136.35
89.60,111.38,98.30,113.39,132.60,138.43,0.255,0.133,0.152,0.168,0.000,30.58,126.60,127.36
82.71,98.67,84.95,94.13,106.27,107.56,0.255,0.133,0.152,0.153,0.000,25.33,125.79,135.04
98.59,120.16,104.48,118.98,137.20,140.88,0.255,0.133,0.152,0.168,0.000,30.67,126.63,127.04
71.33,83.29,71.23,76.68,85.02,84.86,0.255,0.133,0.152,0.153,0.000,22.00,125.45,138.30
93.53,114.83,100.65,115.63,134.82,140.39,0.255,0.133,0.152,0.168,0.000,31.15,125.98,133.24
68.38,79.21,67.40,72.07,79.15,78.66,0.255,0.133,0.152,0.153,0.000,21.10,125.91,133.90
95.08,116.55,102.13,117.30,136.72,142.02,0.255,0.133,0.152,0.168,0.000,31.27,114.30,244.28
64.31,73.88,62.83,67.47,74.55,73.92,0.255,0.000,0.000,0.153,0.000,20.53,112.10,265.18
84.54,103.22,90.76,105.44,123.71,128.14,0.255,0.120,0.152,0.168,0.000,29.47,123.61,155.78
82.15,96.94,83.35,93.15,105.63,106.09,0.255,0.133,0.152,0.153,0.000,25.27,124.05,151.53
97.32,117.81,102.26,116.46,133.71,136.14,0.255,0.133,0.152,0.168,0.000,29.80,124.45,147.78
60.65,69.48,58.87,61.74,66.94,65.76,0.255,0.133,0.167,0.153,0.000,19.00,125.40,138.79
94.09,115.77,101.51,116.88,136.24,141.53,0.255,0.133,0.152,0.168,0.000,31.12,125.81,134.88
61.78,70.90,59.98,62.72,67.89,66.90,0.255,0.133,0.152,0.153,0.000,19.21,113.40,252.77

```

Figura 28: Datos impresos de los sensores instalados.

FUENTE: Tomando del software IDE

En la Figura 28, se muestra parte de la programación del aplicativo, donde el comando “select list item list” hace referencia de que lista se tomara los datos y el comando “index” la posición de la cual tomara el dato.

Figura 27: Transmisión de datos del microcontrolador al aplicativo.

FUENTE: Tomando del software App Inventor

Finalmente se muestra en la pantalla del aplicativo los datos recepcionados de los sensores, este se aprecia en la Figura 29

Presión antes del filtro	21.73	metros
Presión después del filtro	20.38	metros
Presión Válvula-1	2.88	metros
Presión Válvula-2	18.88	metros
Presión Válvula-3	20.47	metros
Presión Válvula-4	0	metros

Figura 29: Interfaz de presiones del sistema de riego.

FUENTE: Tomado del aplicativo móvil

3.4.7. Instalación del sistema de riego automatizado

a. Delimitación e instalación de la red eléctrica

Se efectuó al trazado la zanja de 0.20 metros de ancho por 0.20 metros de profundidad para las cuatro válvulas que se automatizaron, donde se instaló el tubo de polietileno de 32 mm utilizado para la conducción de los cables eléctricos, conjuntamente con sus accesorios (codos y tee), se utilizó esta tubería porque es más resistente al aplastamiento que la tubería de luz convencional; con la ayuda de una wincha pasacable se procedió a instalar los cables para las cuatro parcelas y los terminales en la caseta de riego, donde se encontró la caja pase que alberga al microcontrolador.

La red eléctrica para el tablero de control de la bomba de agua y el sensor de ultrasonido fueron conducidos por una tubería corrugada.

La protección eléctrica de las electroválvulas con cajas de pase y los cables eléctricos con tubo de polietileno fueron necesarios, porque este proyecto estará a disposición de los estudiantes de la UNALM.

b. Instalación de sensores y válvulas.

Para la instalación del sensor de presión, flujo de agua y humedad, así como para la electroválvula en la subunidad, se implementó un arco de riego paralelo al arco manual que estaba en un principio. Por cada arco de riego se instaló un collarín de ½” de doble entrada, donde está el sensor de presión y un punto de prueba; se instaló el sensor de caudal de ½”, la electroválvula de ½” y un sensor de humedad para la subunidad 2.

En el cabezal de riego se instaló un sensor de caudal de 1", antes y después de los filtros se implementó collarines de doble entrada para los sensores de presión.

Finalmente el sensor de ultrasonido se instaló en la parte superior de la tubería de alcantarillado de 160 mm de diámetro y 1.60 m de alto, el cual fue el sustituyente al tubo que controlaba el nivel de agua del tanque.

c. Instalación del microcontrolador

El microcontrolador se instaló en una caja pase de 0.3m de ancho por 0.3m de largo con un grado de protección contra el polvo y humedad (IP65), ubicado en el cabezal de riego, en este espacio también se encuentran los relés que prenden y apagan a las electroválvulas y bomba; por encima de esta caja pase se encuentra otra con dimensiones de 0.2m de ancho por 0.2m de largo, el cual alberga a los transformadores de 12V que energizan a las electroválvulas y al microcontrolador.

d. Control y monitoreo del sistema de riego automatizado

Para el control y monitoreo se dio a través del aplicativo móvil, el cual tiene una interfaz de conexión bluetooth, donde se conectará con el microcontrolador y se estará transmitiendo las variables monitoreadas, y se controlará las electroválvulas y bomba, además se creó una interfaz para calcular el tiempo de riego en base a la humedad de suelo, en esta interfaz que se mostrara más adelante, se visualizan cuadros donde se debe colocar valores por defecto como la humedad a capacidad de campo, profundidad de raíz, densidad aparente del suelo, para que el aplicativo mediante un botón, pueda calcular la lámina de riego real, finalmente con los datos de eficiencia de aplicación y velocidad de aplicación del emisor, se calcula el tiempo de riego, el cual se transmite a un temporizador para el encendido de las válvulas y bomba por el tiempo determinado.

Las fórmulas utilizadas para hallar la lámina real y tiempo de riego se detallan a continuación.

$$Lr = \frac{(CC - H) * Pr}{100}$$

Donde:

Lr: lamina de riego real.

CC: Capacidad de campo (%).

H: Contenido de humedad volumétrico en tiempo real (%).

Pr: Profundidad de raíz (cm).

Para hallar el tiempo de riego se utilizó la siguiente formula

$$Tr = \frac{Lr * 60}{Ea * Va}$$

Donde:

Tr: tiempo de riego (min).

Ea: Eficiencia de aplicación del emisor (decimal).

Va: Velocidad de aplicación del emisor (mm/hr) y el número 60 como una constante para que el tiempo resulte en minutos.

Estas fórmulas son introducidas en la programación del aplicativo para que procese la información que se añade en cada casillero, de esta manera calculando el tiempo de riego en un menor tiempo. El esquema de instalación de proyecto se observa en la Figura 30.

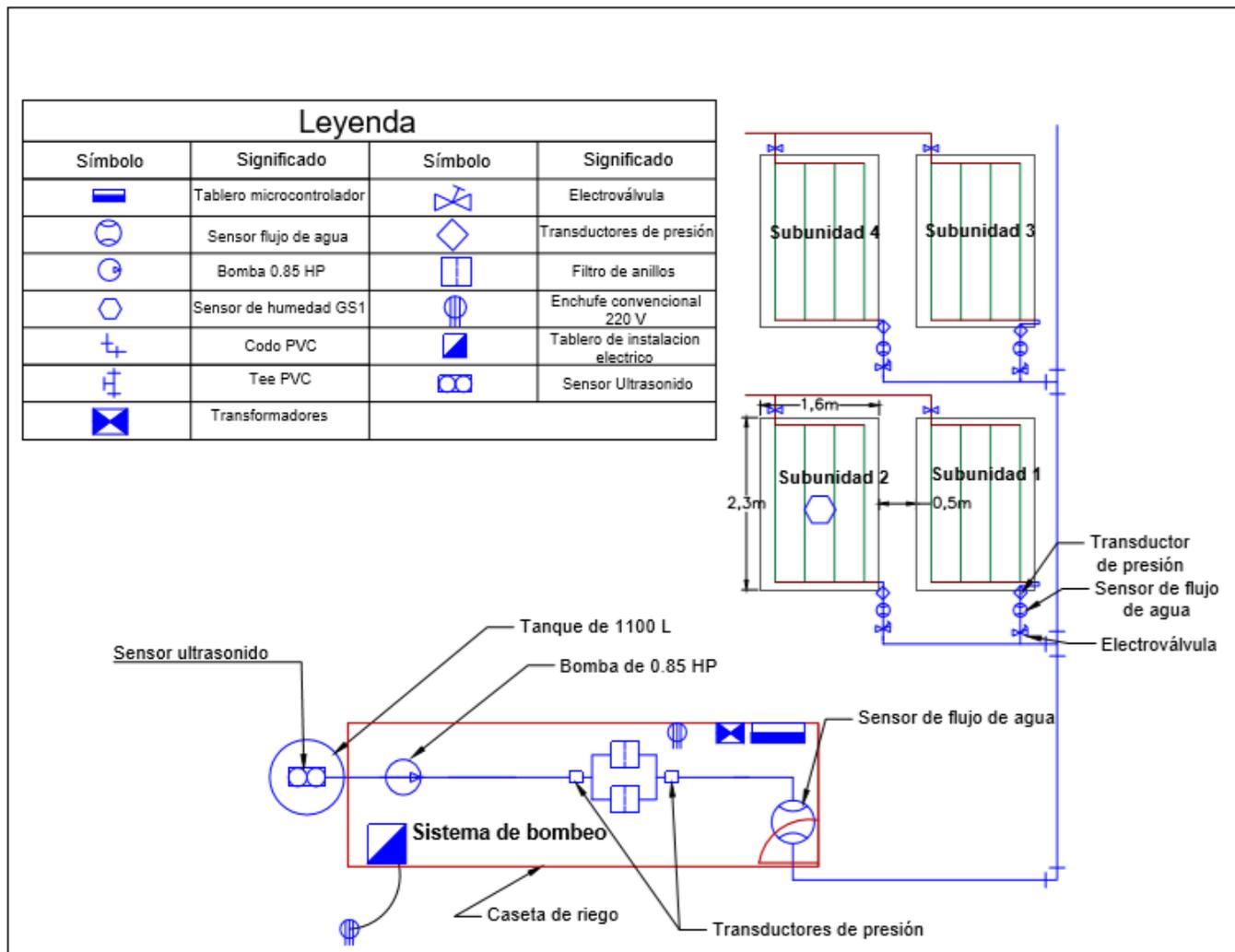


Figura 30: Esquema de instalación de los sensores en campo.

FUENTE: Elaboración propia

IV. RESULTADOS Y DISCUSIÓN

4.1. CARACTERÍSTICAS GENERALES DEL SISTEMA DE RIEGO SUBTERRÁNEO AUTOMATIZADO.

a. Control del nivel de agua en el tanque de polietileno

Se aprecia en la Figura 31, una tubería de 160 mm que está conectada mediante una red de tubería al tanque de polietileno. El sensor de ultrasonido está ubicado en la parte superior y está protegida por una caja pase de 100mmx100mm.

Este sensor envía información del volumen de agua en litros que esta almacenada en el tanque, así también envía la distancia del vacío que existe entre el sensor y el espejo de agua. Según Moreno et al. (2013), este dispositivo ofrece una garantía de protección adecuada que impide que la electrobomba trabaje en seco.



Figura 31: Vaso comunicante para el sensor de ultrasonido.

FUENTE: Elaboración propia

b. Cabezal de riego

La automatización del cabezal de riego comprende el prendido y apagado de la electrobomba, monitoreo de los sensores de presión antes y después de los filtros, sensor de flujo de agua a nivel de tubería principal. Además en este lugar se alberga el microcontrolador con los terminales de los cables de los sensores y actuadores.

En la Figura 32, se puede observar el tablero de control, el cual posee dos botoneras (verde y rojo) los relés asociados al control de esas botoneras se encuentran en la parte inferior del tablero del microcontrolador.

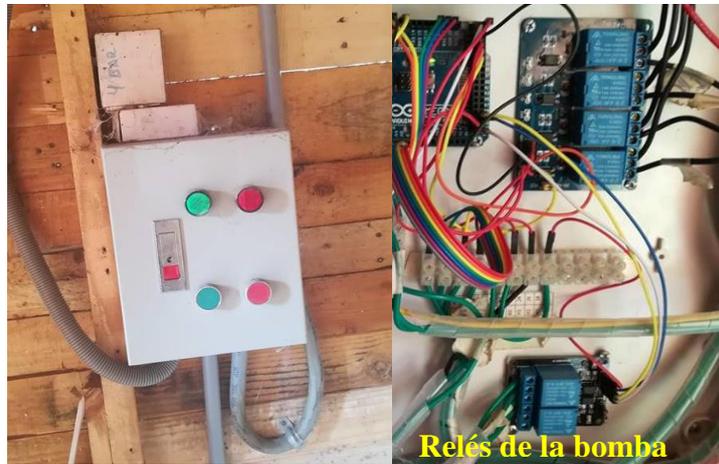


Figura 32: Tablero y relés de control de encendido y apagado de la bomba.

FUENTE: Elaboración propia

En la Figura 33 se visualiza el sistema de filtrado de anillos, aguas arriba se encuentra un collarín de doble entrada, en la parte superior se ubica un punto de prueba de presión,

Sensores de presión HK3022



Figura 33: Sensor de presión antes y después del filtro.

FUENTE: Elaboración propia

mientras que en la parte de abajo se encuentra el sensor de presión HK3022, de igual manera se encuentra aguas abajo del sistema de filtrado.

El sensor de flujo de agua de 1" se encuentra ubicada a nivel de suelo, por ello para su protección se optó por un caja de protección de PVC que tiene como tapa unas rejillas para la visualización de este sensor. Esta protección se puede visualizar en el Anexo 5 Figura 82.

c. Arco de riego

En cada arco de riego de cada una de las 4 subunidades, se encuentra automatizado el control de apertura y cierre de electroválvulas, monitoreo de presiones y caudales, se aprecia en la Figura 34.

Las electroválvulas o válvulas solenoide están protegidas por una caja de pase, tal como se muestra en el Anexo 8 de la Figura 77, debido a que el empalme de los cables que energizan al solenoide estaban desprotegidas, para evitar que este en contacto con el agua y se produzca un cortocircuito se colocó estas cajas de protección.

El sensor de presión por arco se colocó en un collarín de doble entrada para verificar la presión del sensor con un manómetro, los sensores de caudal que se instalaron son de ½". Se conservó el arco de riego que estaba en un principio, como alternativa de riego.

El sensor de humedad está ubicado en la subunidad 2, a 10 cm de la superficie, este sensor se aprecia en el Anexo 8 Figura 73.

Sensores de presión HK3022

Válvula solenoide



Sensor de flujo de agua

Figura 34: Arco de riego automatizado.

FUENTE: Elaboración propia

4.2. DISEÑO ELECTRÓNICO

En este apartado se observará cómo se instaló y conectó todos los dispositivos del sistema, teniendo en cuenta las características que deben tenerse para conexión y el cableado.

Para la instalación eléctrica se usó el cable eléctrico N°18 AWG, teniendo como caída de tensión para la válvula más lejana 1.1V. Los diagramas eléctricos del sistema automatizado de control y monitoreo se diseñaron en el software Autocad, que se presenta a continuación.

4.2.1. Sensor de humedad

Su función es la medición de la humedad en el suelo, este sensor posee tres cables, dos de ellos son para la alimentación 5V y GND, mientras que el tercero es el que envía la información al microcontrolador. El sensor de humedad le corresponde el pin A7, se muestra en la Figura 35.

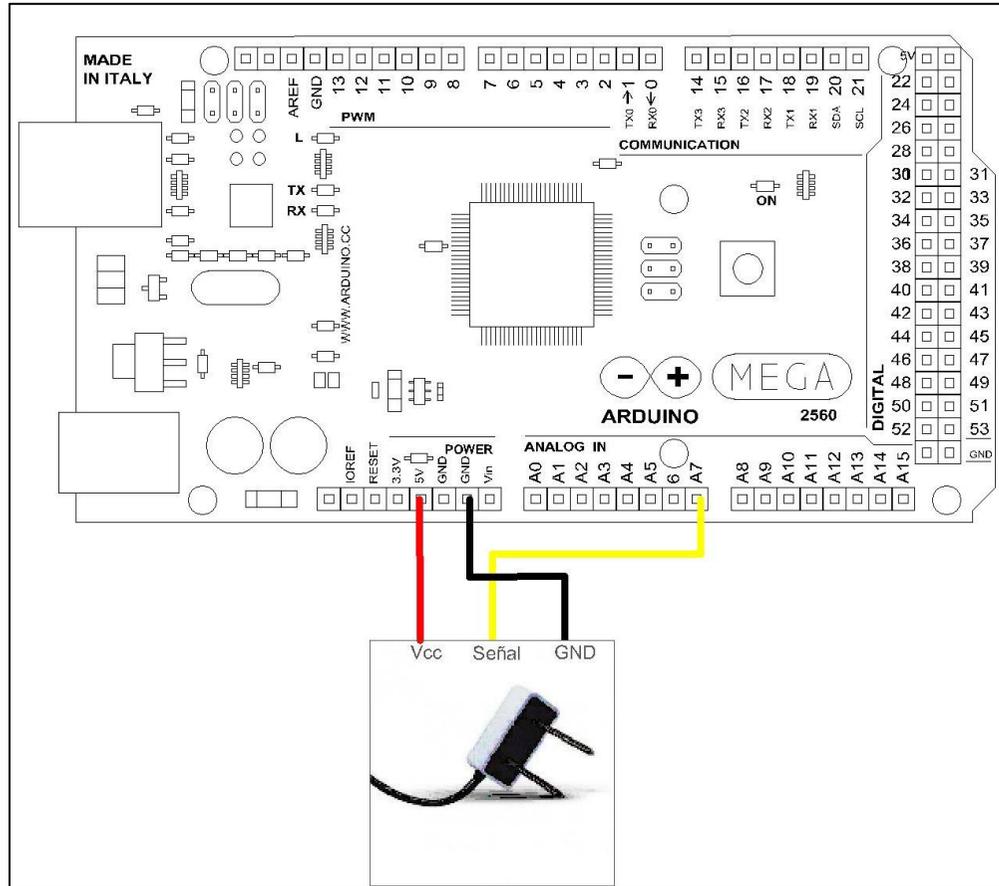


Figura 35: Diseño electrónico sensor de humedad.

FUENTE: Elaboración propia

4.2.2. Transductor de presión

Se instalaron seis sensores de presión, cuatro en cada una de las subunidades y dos a nivel de cabezal, este sensor posee tres cables, dos de ellos para la alimentación y el tercero para enviar la información al microcontrolador. Los pines para el sensor de presión antes y después de los filtros son A1 y A2 respectivamente, los pines A3, A4, A5 y A6 se relacionan con los sensores de presión que están en cada una de las cuatro subunidades, se muestra en la Figura 36.

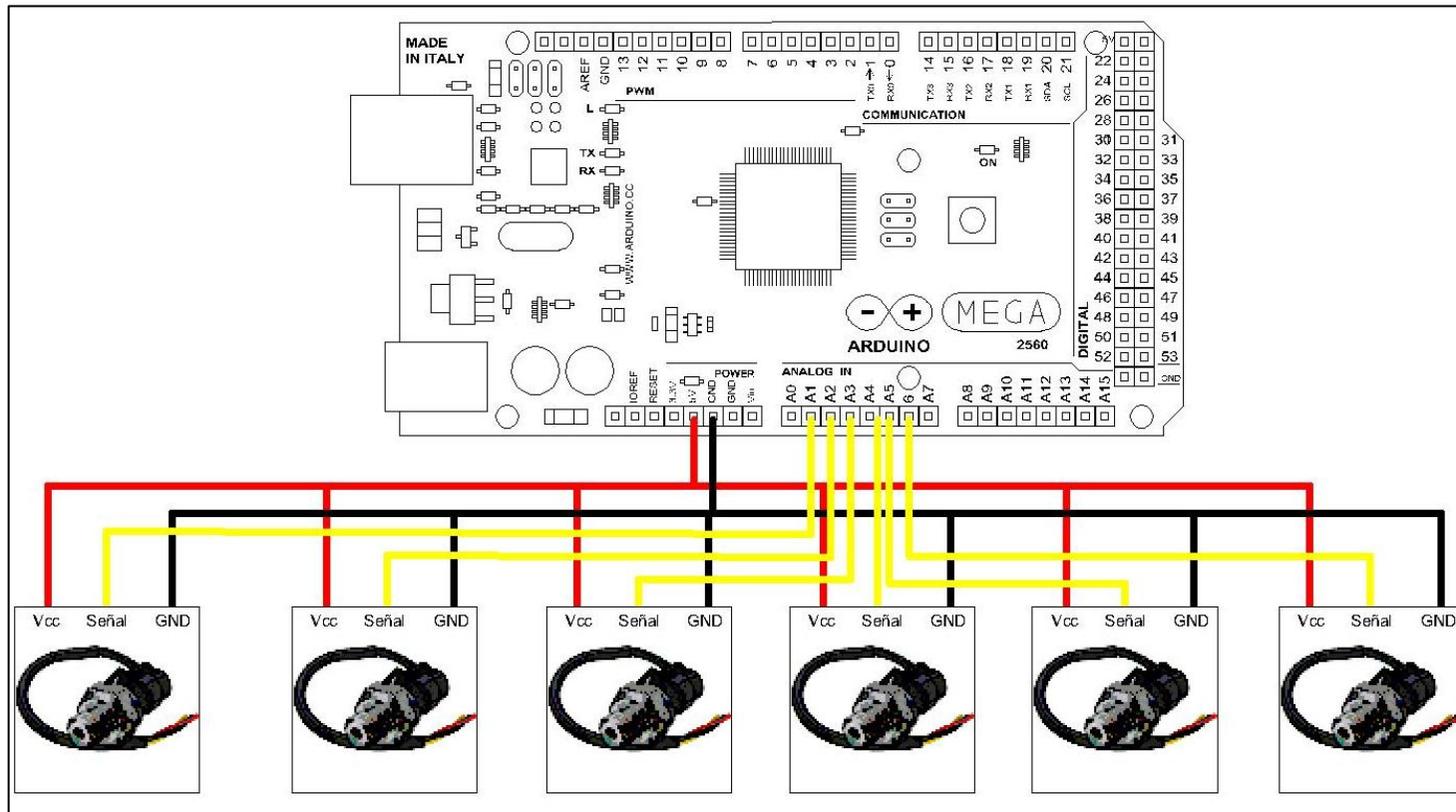


Figura 36: Diseño electrónico transductor de presión.

FUENTE: Elaboración propia

4.2.3. Sensor de flujo de agua

Su función es medir el caudal que pasa por las subunidades y a nivel de cabezal, cabe señalar que este sensor tiene un código de programación el cual tiene como fundamento el conteo de pulsos mediante interrupciones que se generan al moverse las hélices que están dentro de este sensor.

Los pines digitales del Arduino Mega 2560 con la función para habilitar interrupciones son 2, 3, 18, 19, 20 y 21, en la Figura 37 muestra a los pines que fueron conectados.

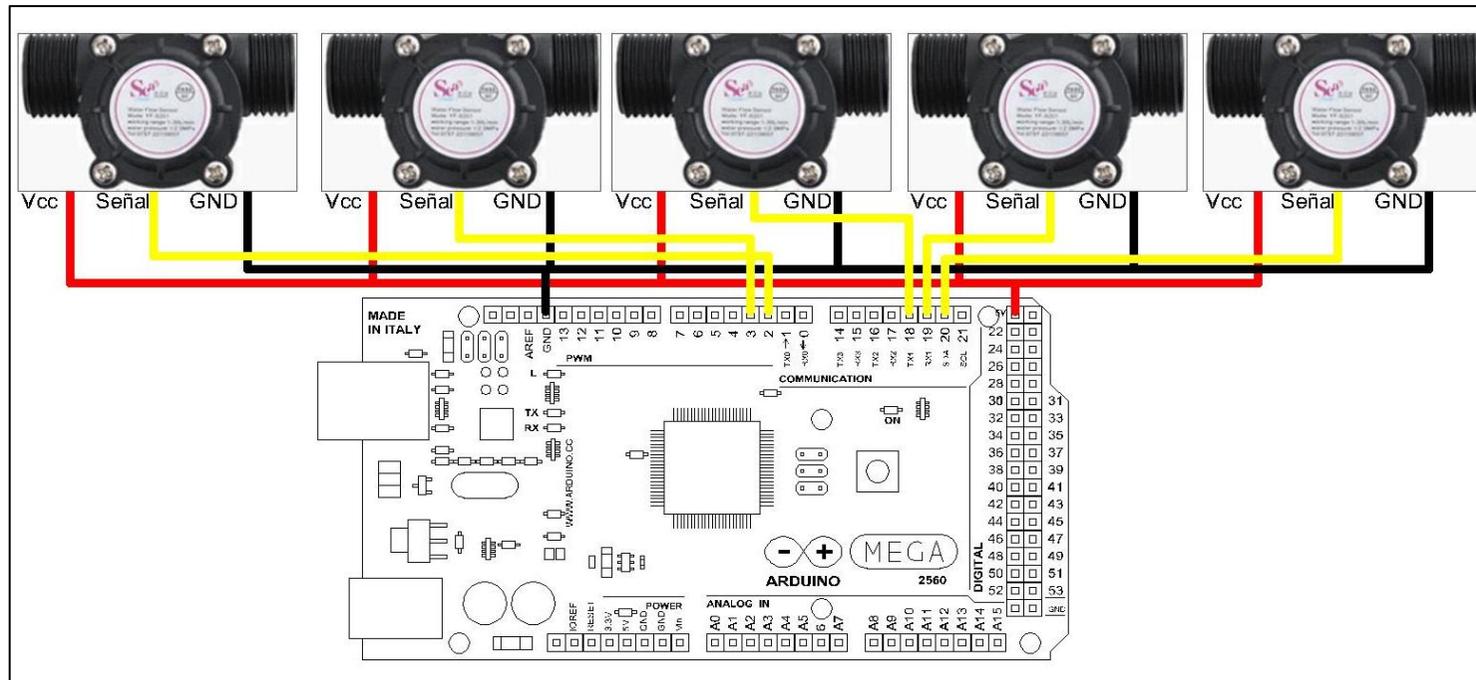


Figura 37: Diseño electrónico sensor de flujo de agua.

FUENTE: Elaboración propia

4.2.4. Sensor de ultrasonido

Mide la distancia desde el sensor al nivel de agua del tanque, este utiliza dos pines, el primero el cual es ECHO para enviar la onda sonora y el segundo que es TRIGGER para recibir la señal sonora que rebota del cuerpo hacia donde fue emitido. El pin Trigger está asociado al pin PWM 13 mientras que el Echo al pin PWM 12, se muestra en la Figura 38.

Este sensor utiliza la velocidad del sonido para determinar la distancia, por ello se utilizó la siguiente formula en la programación $distancia = Velocidad\ del\ sonido * tiempo$.

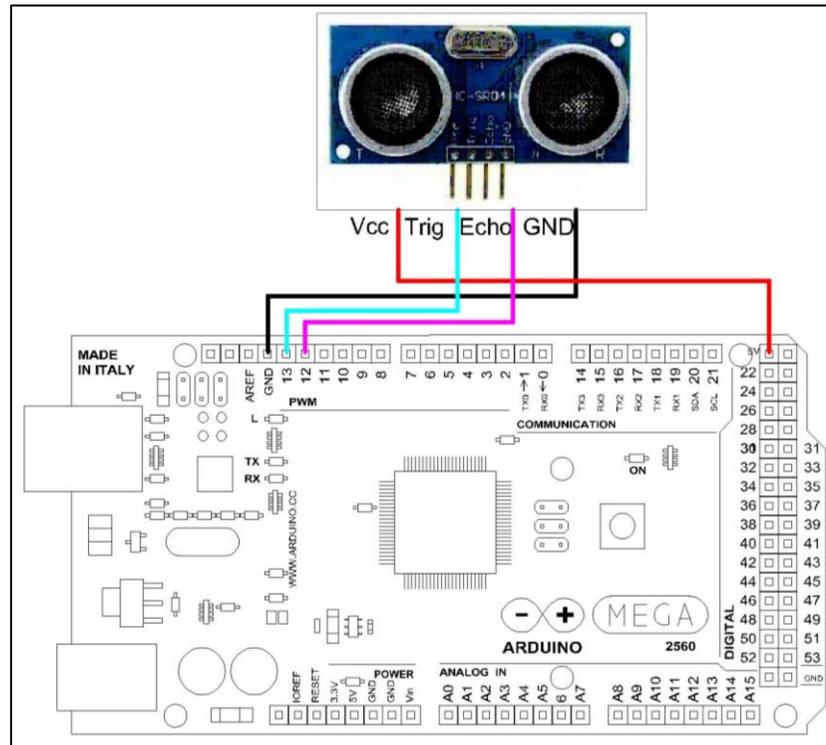


Figura 38: Diseño electrónico sensor de ultrasonido.

FUENTE: Elaboración propia

4.2.5. Módulo relés asociados a la válvula solenoide y electrobomba.

Su función es el accionamiento de los circuitos hasta 230V. El modulo será accionado por salidas digitales.

La placa cuenta con opto-acopladores que aseguran la separación eléctrica entre la parte del control que funciona a 5V y la parte de potencia a 230V, ideal para no dañar el microcontrolador. En las siguientes figuras se observa el diagrama de conexión de las válvulas solenoide de cada una de las cuatro válvulas con los pines 24, 26, 28 y 30, la electrobomba asociada a los pines 50 y 51, para el encendido y apagado de la bomba respectivamente, se muestran en las Figuras 39 y 40.

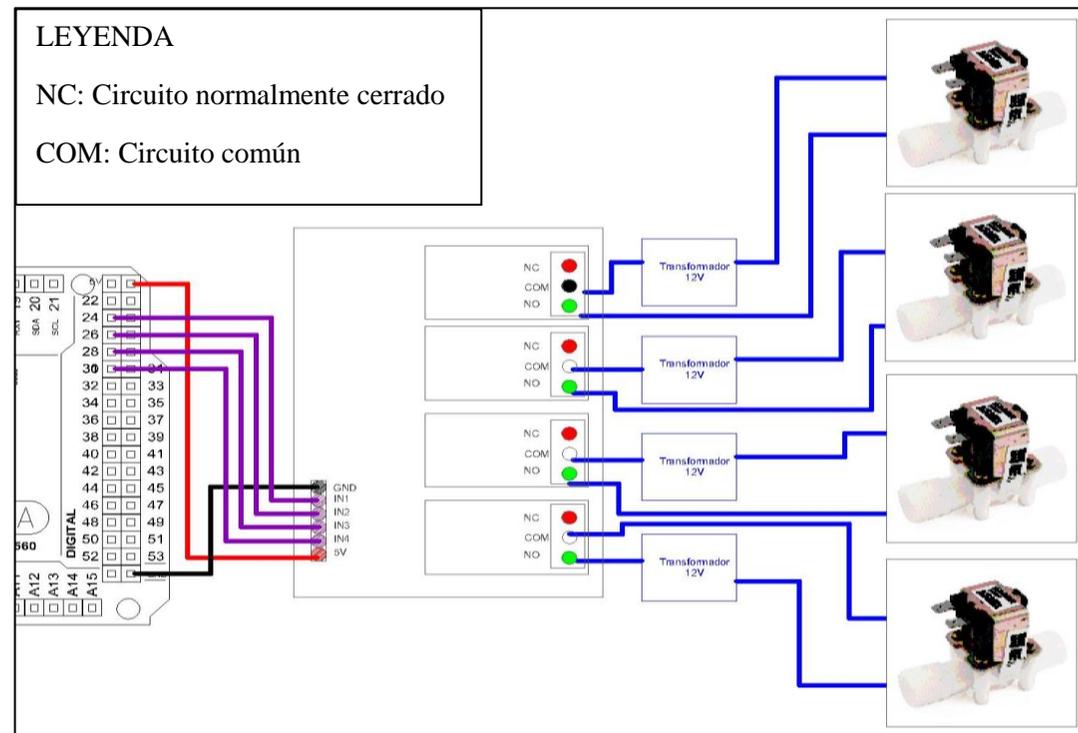


Figura 39: Diseño electrónico válvulas solenoides.

FUENTE: Elaboración propia

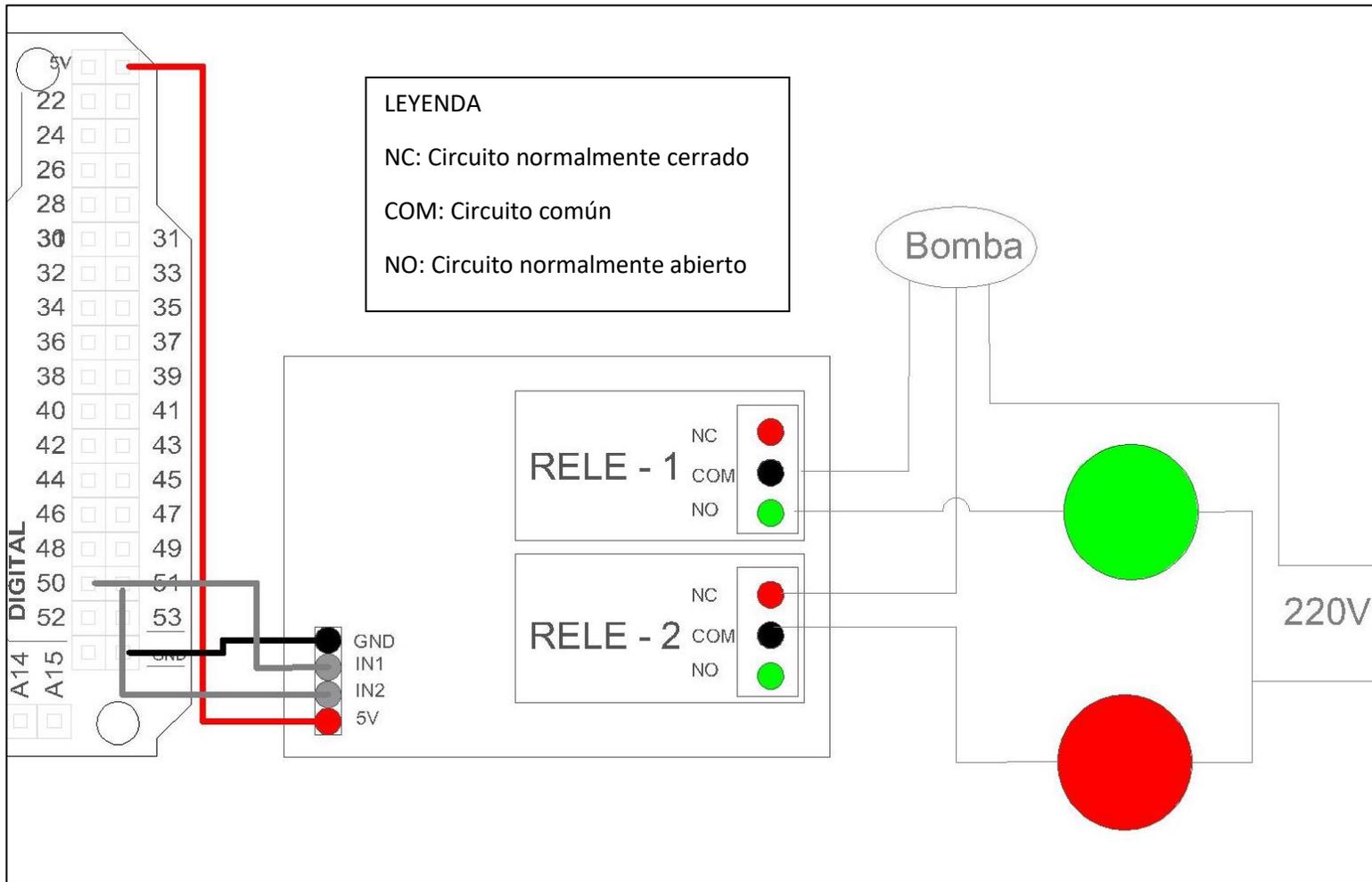


Figura 40: Diseño electrónico de la electrobomba.

FUENTE: Elaboración propia

En el Cuadro 3, se encuentra detallado el pin digital y analógico asociado a cada sensor y actuador, especificando al lugar donde pertenece cada uno de ellos.

Cuadro 3: Resumen de entradas y salidas de la placa Arduino

N°	Sensor/Relé	Tipo de entrada/salida	N° pin en Arduino	Función asociada
1	Relé 1	Entrada Digital	D24	Válvula Solenoide 1
2	Relé 2	Entrada Digital	D26	Válvula Solenoide 2
3	Relé 3	Entrada Digital	D28	Válvula Solenoide 3
4	Relé 4	Entrada Digital	D30	Válvula Solenoide 4
5	Relé 5	Entrada Digital	D50	Bomba 0.85 HP- PRENDER
6	Relé 6	Entrada Digital	D51	Bomba 0.85 HP- APAGAR
7	Sensor de presión HK3022-1	Entrada Analógica	A3	Sensor presión 1
8	Sensor de presión HK3022-2	Entrada Analógica	A4	Sensor presión 2
9	Sensor de presión HK3022-3	Entrada Analógica	A5	Sensor presión 3
10	Sensor de presión HK3022-4	Entrada Analógica	A6	Sensor presión 4
11	Sensor de presión HK3022-5	Entrada Analógica	A1	Sensor presión antes del filtro
12	Sensor de presión HK3022-6	Entrada Analógica	A2	Sensor presión después del filtro
13	Sensor GS1	Entrada Analógica	A7	Sensor de humedad
14	Sensor Caudal-1	Entrada Digital	D3	Sensor de flujo de agua 1
15	Sensor Caudal-2	Entrada Digital	D18	Sensor de flujo de agua 2
16	Sensor Caudal-3	Entrada Digital	D19	Sensor de flujo de agua 3
17	Sensor Caudal-4	Entrada Digital	D20	Sensor de flujo de agua 4
18	Sensor Caudal-5	Entrada Digital	D2	Sensor de flujo de agua cabezal
19	Sensor Ultrasonido	Entrada Digital TRIG	D13	Sensor de nivel de agua
		Entrada Digital ECO	D12	

FUENTE: Elaboración propia

4.3. CALIBRACIÓN

Se hizo ensayos a dos tipos de sensores, analógico (sensor de presión y humedad) y digital (sensor de flujo de agua y ultrasonido).

4.3.1. Calibración de sensores de presión

En el banco de pruebas se hallaron las ecuaciones de ajuste de los seis sensores de presión antes de ser instalados. El esquema del montaje experimental, la rutina de los ensayos y los

resultados obtenidos durante el ensayo de los sensores de presión son mostrados en el Anexo 1.

En el Cuadro 4, se muestra las ecuaciones de los transductores de presión, utilizando el Excel, situando los valores en bits en el eje de las abscisas y en el eje de las ordenadas la presión en metros registrado por un manómetro, de esta manera relacionando la presión en metros, en función de la señal de salida del microcontrolador ($P=f(\text{Bits})$). El rango de la presión varía de 5 a 40 metros y de los bits de 0 a 1023.

Las curvas generadas para hallar las ecuaciones tienen una tendencia lineal con valor $R^2=0.99$, en donde a mayor señal eléctrica, mayor es la presión registrada por el manómetro.

Cuadro 4: Ecuaciones particulares de cada sensor instalado en campo

Ubicación del transductor de presión en el sistema de riego	P= [metros]	B= [bits]
Referencia	Ecuación de ajuste	Coefficiente de determinación (R^2)
P1 (antes del filtro)	$P1=0.1405*B - 14.51$	0.99
P2(después del filtro)	$P2=0.1569*B - 15.56$	0.99
P3(válvula 1)	$P3=0.1236*B - 10.22$	0.99
P4(válvula 2)	$P4=0.1396*B - 16.16$	0.99
P5(válvula 3)	$P5=0.1586*B - 19.66$	0.99
P6(válvula 4)	$P6=0.1633*B - 20.79$	0.99

FUENTE: Elaboración propia

Según Wang et al. (2016), para un sistema de riego donde las presiones oscila de 1.5 a 2 bar se obtienen curvas de calibración de transductores de presión (modelo HK3022), con un coeficiente de correlación y determinación de 0.99, instalados al principio y final de la línea de goteos, con la finalidad de permitir saber si los goteos están obstruidos midiendo la diferencia entre ambos valores, si esa diferencia es igual a cero, significa que no hay descarga de agua y que los goteos están obstruidos, este sistema funciona con éxito, dando a conocer la aplicación de este tipo de tecnología en sus actividades ordinarias, este forma de utilizar los sensores es para longitud de laterales donde haya una pérdida de presión mayor a 1 metro.

Posterior a la instalación de los sensores se calibraron las lecturas de los sensores de presión, para ello individualmente se tomó la presión en cada punto en tres rangos distintos de 20m, 15m y 10m, siendo comparados con la lectura que el sensor estaba enviando al dispositivo móvil, obteniendo datos reales que son las presiones tomadas por el manómetro en cada punto de prueba instalado en el mismo collarín del sensor de presión y los datos teóricos que son los obtenidos por los sensores de presión, con estos dos datos fueron calculados el

porcentaje de error que presentan los sensores de presión. Estos datos se pueden visualizar en el Cuadro 5.

Cuadro 5: Cálculo del porcentaje de error de los transductores de presión

Transductores de presión	Presión Real (P) metros			%error P=20m	%error P=15m	%error P=10m
	20	15	10			
Antes del filtro	19.80	14.83	9.86	1.00	1.13	1.40
Después del filtro	19.43	14.78	9.74	2.85	1.47	2.60
Subunidad 1	20.20	14.89	10.18	1.00	0.73	1.80
Subunidad 2	19.58	15.25	9.73	2.10	1.67	2.70
Subunidad 3	19.59	15.26	9.87	2.05	1.73	1.30
Subunidad 4	19.72	15.14	9.83	1.40	0.93	1.70

FUENTE: Elaboración propia.

Según especificaciones técnicas del sensor de presión el error de medición es $\pm 1\%$, los porcentajes de error calculados en las pruebas oscila de 1% a 2.85%, presentando el mayor error en el sensor de presión después del filtro.

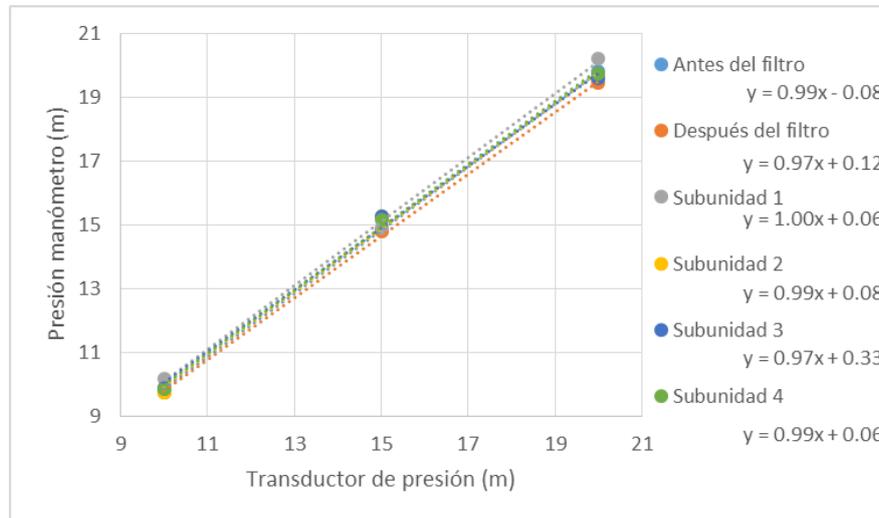


Figura 41: Curvas de calibración de los transductores de presión.

FUENTE: Elaboración propia

En la Figura 41 se aprecia las curvas de calibración de los transductores de presión donde el coeficiente de correlación y determinación es de 0.99, corroborando las ecuaciones de ajuste, además de tener una pendiente cercana a uno la que determina un ángulo de 45° .

En el transcurso del proyecto estos transductores sufrieron descalibraciones paulatinas, debido a que estaban expuestos a la intemperie y sufrían oxidación.

4.3.2. Calibración del sensor de humedad

Este sensor fue instalado en la subunidad 2, a 10 cm del suelo, se saturó la subunidad para tomar muestras de suelo y correlacionarlo con la resolución en bits que lee el microcontrolador.

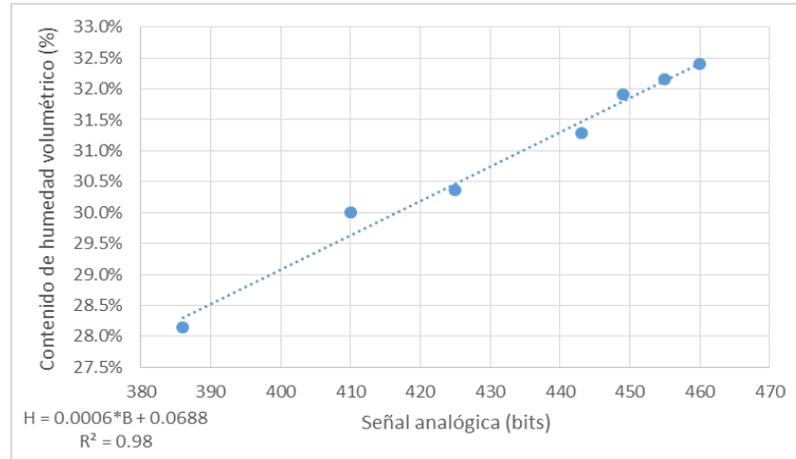


Figura 42: Ecuación de ajuste para la transformación de señal, sensor GS1.

FUENTE: Elaboración propia

Para obtener precisión en las mediciones, el sensor de humedad requiere calibración específica en el suelo (Peters 2016), por ello las pruebas de campo para la determinación de la ecuación de ajuste y calibración del sensor fueron con el sensor instalado en la subunidad 2. El coeficiente de determinación y correlación de la ecuación de ajuste es de 0.98, el cual indica la alta relación directa entre el la señal eléctrica enviada por el sensor y la humedad volumétrica de suelo.

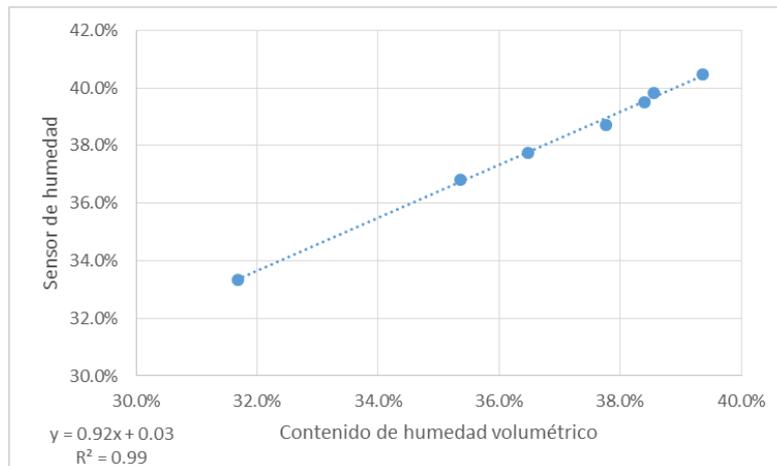


Figura 43: Calibración del sensor de humedad GS1.

FUENTE: Elaboración propia

En la Figura 43, se observa la curva de calibración del sensor humedad comparada con muestras de suelo extraídas alrededor del sensor instalado, obteniendo una pendiente de 0.92, indicando una relación cercana entre los datos de los ejes, así también obteniendo porcentaje de error de 2%.

4.3.3. Calibración de sensores de flujo de agua

Para tener una medición más exacta fue necesario calcular el valor del factor de conversión K, el cual mediante una formula descrita en la metodología, así como el proceso de calibración se trasformó la frecuencia de pulsos a caudal de agua.

La variable que podemos medir con exactitud es la cantidad de pulsos (usando el microcontrolador), y con ayuda de un recipiente de graduación se midió el volumen de agua. En el Cuadro 6, se muestra los factores de conversión hallados para cada sensor de flujo de agua.

Cuadro 6: Factor de corrección para la conversión de frecuencia a caudal

Ubicación del sensor de flujo de agua en el sistema de riego	Factor de corrección
Referencia	K
Q1 (cabezal)	0.72
Q2(válvula 1)	1.25
Q3(válvula 2)	1.10
Q4(válvula 3)	1.09
Q5(válvula 4)	1.15

FUENTE: Elaboración propia

El sensor de caudal utilizado en el cabezal es el modelo FS400 de 1” y los sensores que están en las subunidades son del modelo YF-201 de ½”, por este motivo el factor difiere, siendo para el diámetro de 1” de 0.72 y para el de ½” en promedio 1.10.

El sensor según ficha técnica puede llegar hasta un 10% de error, para ello se hizo la verificación con los sensores calibrados instalados en campo.

Para un tiempo de veinte minutos se encendió el riego para cada subunidad, tomando los datos de la válvula volumétrica inicial y final, de esta manera calculando el caudal real. Así mismo se tomaba el dato del sensor de flujo de agua que permanecía estable en el tiempo de riego, este dato siendo el caudal teórico se calculó el porcentaje de error que presentan estos sensores. El sensor de caudal registraba un valor de 0.02 l/s, siendo este el caudal teórico de la subunidad.

Para la calibración se realizó dos pruebas por cada sensor, la primera al encender el sistema considerando la existencia de aire en la red de tuberías y la segunda prueba cuando el flujo de agua es continua, para la primera prueba se obtuvo un error en la subunidad 4 de 4.27%, mientras que para la segunda un error mínimo de 0.39%. Los resultados se visualizan en el Cuadro 7.

Cuadro 7: Cálculo de porcentaje de error para la subunidad 4

SUBUNIDAD 4				Caudal sensor	0.0215	L/s
				tiempo=	20	min
N°	L.I.VAL (M3)	L.F.VAL (M3)	DIFERENCIA (M3)	Vol. Aplicado (Litros)	Vol. Sensor de flujo (Litros)	%error $\frac{Q_{real} - Q_{teorico}}{Q_{real}} * 100$
1	23.9583	23.9852	0.0269	26.95	25.80	4.27
2	23.9852	24.0109	0.0257	25.70	25.80	0.39

FUENTE: Elaboración propia

El porcentaje de error mínimo para el sensor de flujo YF-S201 de ½” de la subunidad 4 es de 0.39%, siendo este menor al porcentaje de error mencionado en la ficha técnica del producto en 9.61%. La diferencia entre la lectura de la válvula volumétrica (25.70 litros) y la lectura del sensor de caudal (25.80 litros) es de 0.10 litros para el tiempo de riego de 20 minutos, obteniendo un error por minuto de 5 ml.

El sensor de flujo de agua G ½” efecto Hall calibrado, similar al que se está usando en este proyecto posee errores máximo de 1.52 l/min, siendo un sensor confiable para medir el caudal de agua dentro de las tuberías del riego. La aplicación de este sensor Hall demuestra ser un buen sensor que puede detectar fugas en las tuberías y es más atractivo, porque proporciona una operación con gran precisión y el método más barato para medir el caudal de agua en la agricultura (Sood et al., 2013).

Cuadro 8: Cálculo de porcentaje de error para la subunidad 1.

SUBUNIDAD 1				Caudal sensor	0.02	L/s
				tiempo=	20	min
N°	L.I.VAL (M3)	L.F.VAL (M3)	DIFERENCIA (M3)	Vol. Aplicado (Litros)	Vol. Sensor de flujo (Litros)	%error
1	46.0774	46.10375	0.02635	26.35	24.00	8.92
2	46.10375	46.1288	0.025050	25.10	24.00	4.19

FUENTE: Elaboración propia

Cuadro 9: Cálculo de porcentaje de error para la subunidad 2.

SUBUNIDAD 2				Caudal sensor	0.0225	L/s
				tiempo=	20	min
N°	L.I.VAL (M3)	L.F.VAL (M3)	DIFERENCIA (M3)	Vol. Aplicado (Litros)	Vol. Sensor de flujo (Litros)	%error
1	46.1288	46.1578	0.029	29.00	27.00	6.90
2	46.1578	46.18395	0.026	26.20	27.00	3.25

FUENTE: Elaboración propia.

Cuadro 10: Cálculo de porcentaje de error para la subunidad 3.

SUBUNIDAD 3				Caudal sensor	0.023	L/s
				tiempo=	20	min
N°	L.I.VAL (M3)	L.F.VAL (M3)	DIFERENCIA (M3)	Vol. Aplicado (Litros)	Vol. Sensor de flujo (Litros)	%error
1	23.9035	23.9328	0.0293	29.30	27.60	5.80
2	23.9328	23.9598	0.0270	27.00	27.60	2.22

FUENTE: Elaboración propia.

Cuadro 11: Calculo de porcentaje de error para el cabezal.

CABEZAL				Caudal sensor	0.023	L/s
				tiempo=	20	min
N°	L.I.VAL (M3)	L.F.VAL (M3)	DIFERENCIA (M3)	Vol. Aplicado (Litros)	Vol. Sensor de flujo (Litros)	%error
1	169.8292	169.8578	0.02855	28.55	27.60	3.33
2	169.8578	169.88485	0.02705	27.10	27.60	2.03

FUENTE: Elaboración propia.

El sensor de flujo de agua instalada en el cabezal es de 1", siendo el error mínimo de la segunda prueba 2.03%, siendo este menor al porcentaje de error mencionado en la ficha técnica del producto en 7.97%. La diferencia entre la lectura de la válvula volumétrica (27.10 litros) y la lectura del sensor de caudal (27.60 litros) es de 0.50 litros para el tiempo de riego de 20 minutos, obteniendo un error por minuto de 25 ml.

4.3.4. Calibración del sensor de ultrasonido

El sensor de ultrasonido está ubicado en la parte superior de la tubería de PVC de 160 mm protegida por una caja pase de 100mmx100mm.

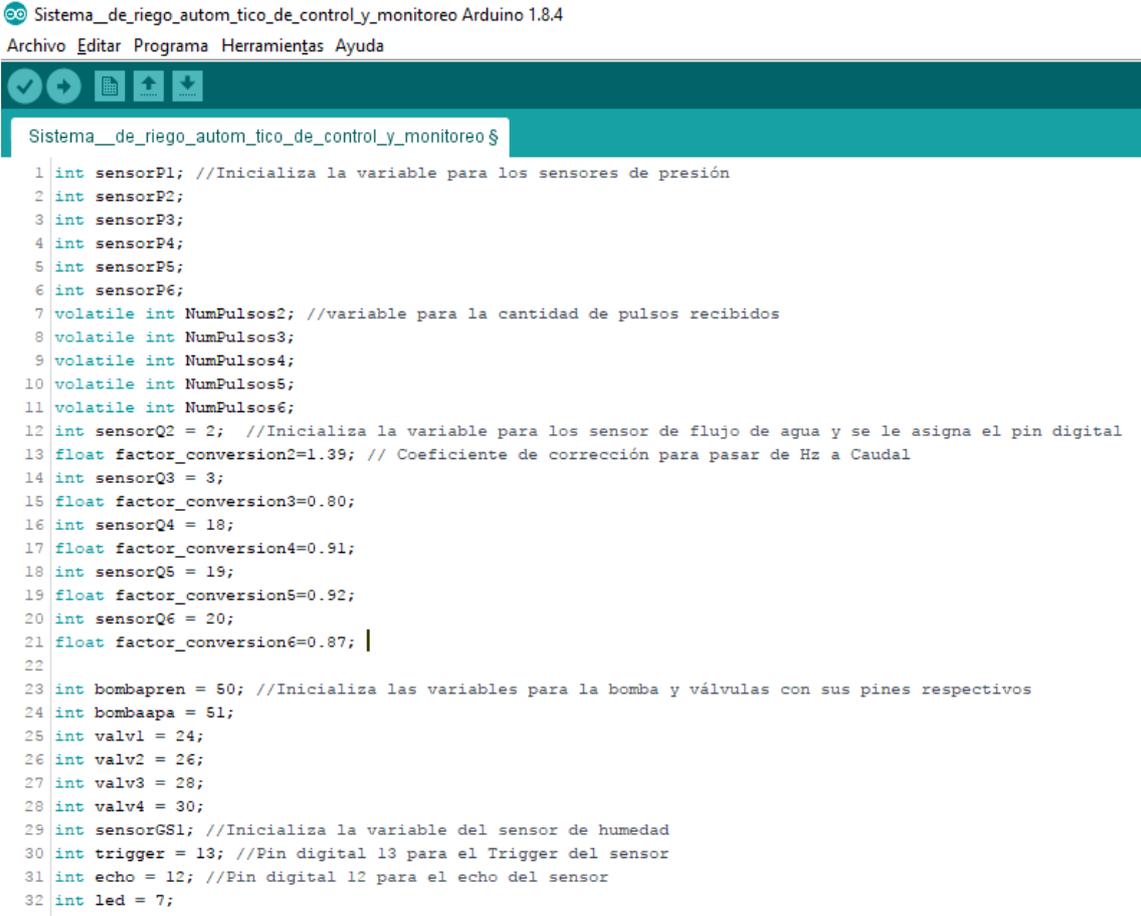
El sensor envía la medida en centímetros y es corroborada con una regla, encontrando un coeficiente de determinación igual a 1.

Según Moreno et al. (2013), este dispositivo ofrece una garantía de protección adecuada que impide que la electrobomba trabaje en seco.

4.4. PROGRAMACIÓN

En las Figuras 44 y 45, se muestra el principio del código elaborado en lenguaje C/C++ para el programa utilizado en el sistema de riego automático de control y monitoreo, el cual fue desarrollado en el software IDE del Arduino. El código completo estará en el Anexo 5, donde se explica a detalle el funcionamiento de cada lineamiento del código elaborado.

Los sensores digitales flujo de agua, válvulas, bomba, ultrasonido se le asignan los pines a los que estarán asociados, se visualiza en la Figura 44. Los sensores analógicos serán asociados más adelante.



```
Sistema_de_riego_autom_tico_de_control_y_monitoreo Arduino 1.8.4
Archivo Editar Programa Herramientas Ayuda

Sistema__de_riego_autom_tico_de_control_y_monitoreo $
1 int sensorP1; //Inicializa la variable para los sensores de presión
2 int sensorP2;
3 int sensorP3;
4 int sensorP4;
5 int sensorP5;
6 int sensorP6;
7 volatile int NumPulsos2; //variable para la cantidad de pulsos recibidos
8 volatile int NumPulsos3;
9 volatile int NumPulsos4;
10 volatile int NumPulsos5;
11 volatile int NumPulsos6;
12 int sensorQ2 = 2; //Inicializa la variable para los sensor de flujo de agua y se le asigna el pin digital
13 float factor_conversion2=1.39; // Coeficiente de corrección para pasar de Hz a Caudal
14 int sensorQ3 = 3;
15 float factor_conversion3=0.80;
16 int sensorQ4 = 18;
17 float factor_conversion4=0.91;
18 int sensorQ5 = 19;
19 float factor_conversion5=0.92;
20 int sensorQ6 = 20;
21 float factor_conversion6=0.87; |
22
23 int bombapren = 50; //Inicializa las variables para la bomba y válvulas con sus pines respectivos
24 int bombaapa = 51;
25 int valv1 = 24;
26 int valv2 = 26;
27 int valv3 = 28;
28 int valv4 = 30;
29 int sensorGS1; //Inicializa la variable del sensor de humedad
30 int trigger = 13; //Pin digital 13 para el Trigger del sensor
31 int echo = 12; //Pin digital 12 para el echo del sensor
32 int led = 7;
--
```

Figura 44: Declaración de variables.

FUENTE: Tomado del software IDE.

4.4.1. Declaración de variables

Se definió las variables que van a ser utilizadas durante la ejecución del programa y le asignamos un nombre representativo para una mejor comprensión. Se crea la variable para cuantificar la cantidad de pulsos recibidos del sensor de flujo de agua. Se visualiza en la Figura 44.

4.4.2. Configuración del código de programación (Setup)

La parte del setup se ejecuta una sola vez y fue donde se definió la función de los pines de la placa Arduino, también se incluyó la velocidad de transmisión para la comunicación serial, de esta manera dando inicio a las variables del programa.

Los componentes que se utilizaron para recibir datos son el sensor de humedad, sensor de flujo de agua y el pin Echo del sensor de ultrasonido, por otra parte los componentes que se usaron como salida son el pin Trigger del sensor de ultrasonido, los relé de la bomba y electroválvula.

Por seguridad cuando se encienda el microcontrolador, se añadió a la programación que se inicie con los relés de la bomba y electroválvula en estado apagado. La configuración se observa en la Figura 45.

En resumen el código creado procesa la información recibida de los sensores de presión, flujo de agua, humedad y ultrasonido, para ser impresos en el monitor serial del software y finalmente realizar la comunicación Bluetooth con el aplicativo móvil.

Se crearon tres alertas por las eventualidades que suceden en campo, estas son:

- La primera alerta con la siguiente condicional, si el volumen de agua es menor a los 250 Lts, entonces enviar una señal de vibración al celular por 2 segundos e impide el funcionamiento de la electrobomba en seco.
- Si la distancia vacío del sensor de ultrasonido al espejo de agua es menor a 30 cm, enviar una señal de vibración y una señal de alerta escrita en la pantalla del dispositivo móvil.
- Si la diferencia de presión antes y después del filtro supera los 5 metros, enviar una señal de vibración al celular por 10 segundos y plasmar un mensaje de alerta en la pantalla del dispositivo móvil.

Para el encendido de las válvulas y bomba se envió información hacia el monitor serial, para ello se creó condicionales de la siguiente manera:

Si el monitor serial recibe la letra “A” (ON) encender bomba, si el monitor serial recibe la letra “B” (OFF) apagar bomba y así sucesivamente para las demás válvulas con su respectiva letra.



```
Sistema_de_riego_autom_tico_de_control_y_monitoreo Arduino 1.8.4
Archivo Editar Programa Herramientas Ayuda

Sistema__de_riego_autom_tico_de_control_y_monitoreo $
111 void setup() {
112   pinMode(bombapren,OUTPUT); //Definimos las variables de bomba y válvulas como salida, enviamos información.
113   pinMode(bombaapa,OUTPUT);
114   pinMode(valv1,OUTPUT);
115   pinMode(valv2,OUTPUT);
116   pinMode(valv3,OUTPUT);
117   pinMode(valv4,OUTPUT);
118   pinMode(trigger, OUTPUT); // pin como salida
119   pinMode(echo, INPUT); // pin como entrada
120   Serial.begin(9600); //Activamos el monitor serial
121   digitalWrite(bombapren,HIGH); //Cuando el microcontrolador se encienda las variables de valvulas y bomba esten apagadas.
122   digitalWrite(bombaapa,HIGH);
123   digitalWrite(valv1,HIGH);
124   digitalWrite(valv2,HIGH);
125   digitalWrite(valv3,HIGH);
126   digitalWrite(valv4,HIGH);
127
128   pinMode(7, INPUT); //alarma de la distancia(encenderá el led)
129
130   pinMode(sensorQ2, INPUT); //Definimos las variables de sensor de flujo de agua como entrada, recibimos información.
131   attachInterrupt(digitalPinToInterrupt(2),ContarPulsos2,RISING); //activamos las interrupciones, se define el pin del se
132   pinMode(sensorQ3, INPUT);
133   attachInterrupt(digitalPinToInterrupt(3),ContarPulsos3,RISING);
134   pinMode(sensorQ4, INPUT);
135   attachInterrupt(digitalPinToInterrupt(18),ContarPulsos4,RISING);
136   pinMode(sensorQ5, INPUT);
137   attachInterrupt(digitalPinToInterrupt(19),ContarPulsos5,RISING);
138   pinMode(sensorQ6, INPUT);
139   attachInterrupt(digitalPinToInterrupt(20),ContarPulsos6,RISING);
140 }
141
```

Figura 45: Configuración de variables.

FUENTE: Tomado del software IDE.

4.4.3. Diagrama de flujo

En la Figura 46, se muestra el diagrama de flujo de la elaboración del código de programación, se definió las variables que van a ser utilizadas durante la ejecución del programa, también la función de entrada o salida según el sensor o actuador que se utilizó, establecer la velocidad del puerto serial, los comandos para el encendido o apagado de la bomba y electroválvulas, la lectura de los sensores para que finalmente sean mostrados en el monitor serial y se pueda interactuar con el aplicativo móvil.

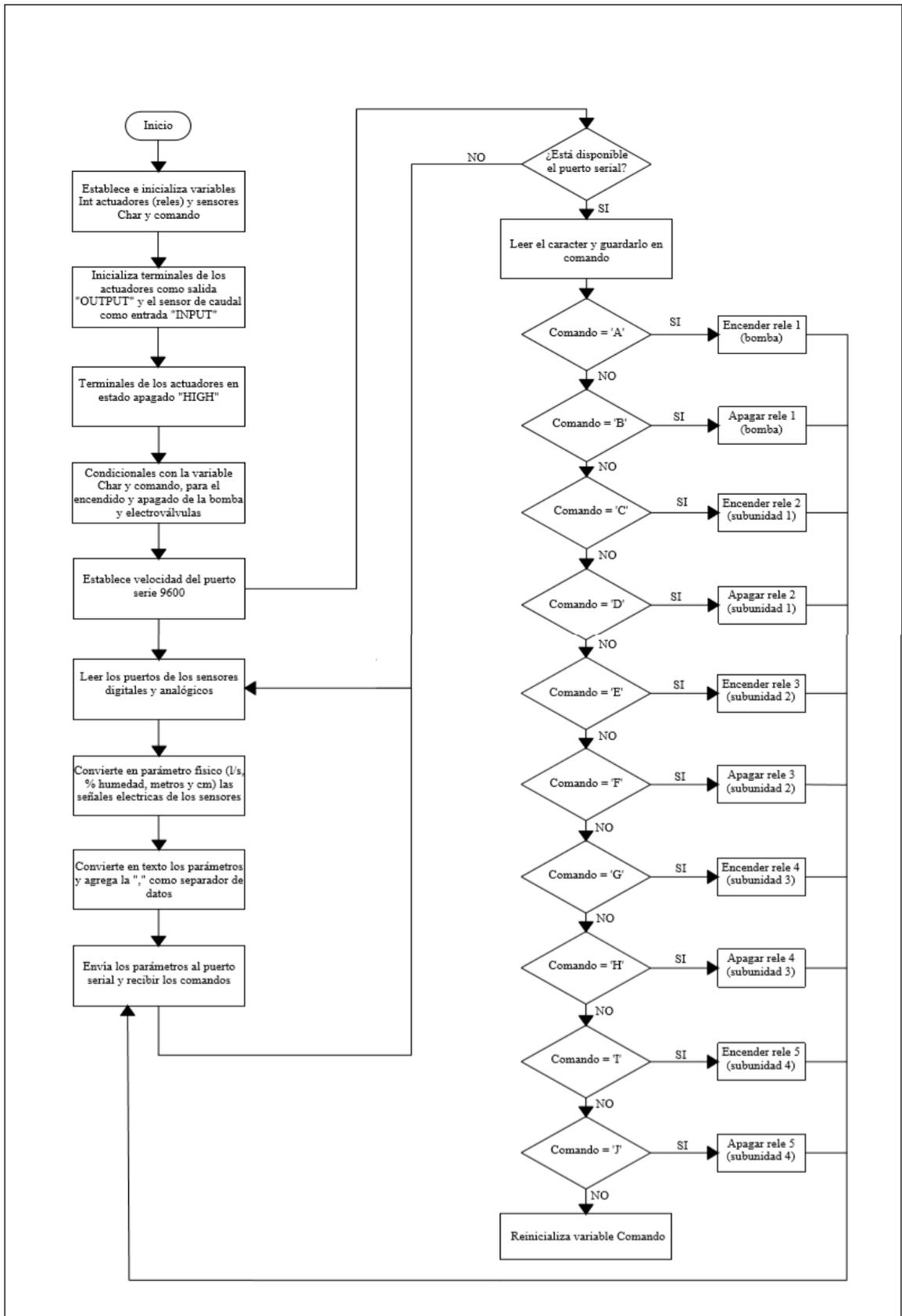


Figura 46: Diagrama de flujo del código de programación para el control y monitoreo del sistema de riego.

FUENTE: Elaboración propia.

4.5. INTERFAZ DE USUARIO

4.5.1. Interfaz de conexión

En la Figura 47, se visualiza la interfaz gráfica del aplicativo móvil denominado “Ardunalm”, en la cual el usuario debe conectarse al sistema Bluetooth, para escoger el tipo de sensor que desea monitorear o los actuadores que desea controlar.



Figura 47: Interfaz de inicio y conexión Bluetooth.

FUENTE: Tomado del aplicativo móvil.

En lo correspondiente a la aplicación móvil se ha realizado una programación para poder leer los datos obtenidos desde el puerto serie del microcontrolador, se realiza una secuencia de arranque muy intuitiva desde la conexión Bluetooth, para sincronizar con el Arduino y luego visualizaremos en la pantalla los datos que este nos envía.

Lo que se ha tenido en cuenta en esta aplicación a la hora de programar es tener una recepción de datos cada segundo, de esta manera actualice la información recibida en tiempo real.

4.5.2. Interfaz de presión

Se creó una pantalla donde está descrito los seis sensores de presión, que se visualiza en la Figura 48, con su respectivo casillero que recibe los datos que son mandados a través del monitor serial del software IDE, el monitoreo de la presión evalúa en un casillero que



Figura 48: Interfaz de monitoreo de presiones.

FUENTE: Tomado del aplicativo móvil.

está en la parte final de la pantalla si hay una variación de cinco metros entre la lectura del sensor antes y después del filtro, en caso la hubiese se crea una alerta la cual se muestra en ese casillero en letras rojas, además de vibrar el celular por diez segundos.

4.5.3. Interfaz de caudal

Se creó con el principal motivo de detectar fugas de agua u obstrucción de goteros, para ello los datos recibidos de los sensor de flujo de agua de las cuatro parcelas son sumadas para



Figura 49: Interfaz de monitoreo de caudales.

FUENTE: Tomado del aplicativo móvil.

que sean evaluadas con la lectura del sensor de flujo que está en el cabezal, al presionar el botón monitoreo, automáticamente el valor del caudal en ese momento se compara con la suma de los caudales de las cuatro válvulas. El monitoreo se realiza cada segundo, visualizándose el interfaz en la Figura 49.

4.5.4. Interfaz sensor de humedad

En la Figura 50 se muestra la pantalla donde se monitorea la humedad volumétrica del suelo, así también se crearon casilleros donde se establecen los parámetros agronómicos para el cálculo de la lámina de riego real, estableciendo la eficiencia de aplicación y los parámetros hidráulicos (velocidad de aplicación del emisor), se halla el tiempo de riego, que estará en función de la humedad de suelo tomada en tiempo real.

Al calcular el tiempo de riego en minutos, este se desplaza a un temporizador el cual posee tres botones iniciar, parar y reiniciar.

Si se presiona el botón iniciar automáticamente se encenderá la bomba y las cuatro válvulas automatizadas, iniciando el conteo regresivo del temporizador hasta llegar a cero, en donde



Figura 50: Interfaz de monitoreo de humedad y cálculo del tiempo de riego.

FUENTE: Tomado del aplicativo móvil.

se visualizan el color del texto en rojo y empieza a vibrar el teléfono móvil por 1 segundo, alertando que termino el riego.

Si por alguna circunstancia se tiene que detener el riego, se creó el botón de parar, el cual al presionarlo, manda la señal respectiva para apagar la bomba, las electroválvulas y deteniendo el conteo del temporizador, si se desea proseguir con el riego se presiona nuevamente el botón iniciar.

El botón reiniciar al presionarlo el temporizador vuelve a cero, en caso de presionarlo cuando el riego este encendido, automáticamente se apaga la bomba y electroválvulas.

4.5.5. Interfaz sensor de ultrasonido

En la Figura 51 se observa el interfaz, donde se visualiza dos parámetros asociados al sensor de ultrasonido, el primero es la cantidad de agua disponible en litros, este parámetro se puede saber conociendo las dimensiones de la fuente de agua en este caso el tanque de polietileno y la altura de agua que estará midiendo el sensor, se derivó una fórmula para hallar el volumen de agua en función a la distancia medida del sensor al espejo de agua.

El segundo parámetro que se visualiza en la interfaz es la distancia vacía del sensor al espejo de agua, para este proyecto es necesario este parámetro, debido a que el llenado del tanque

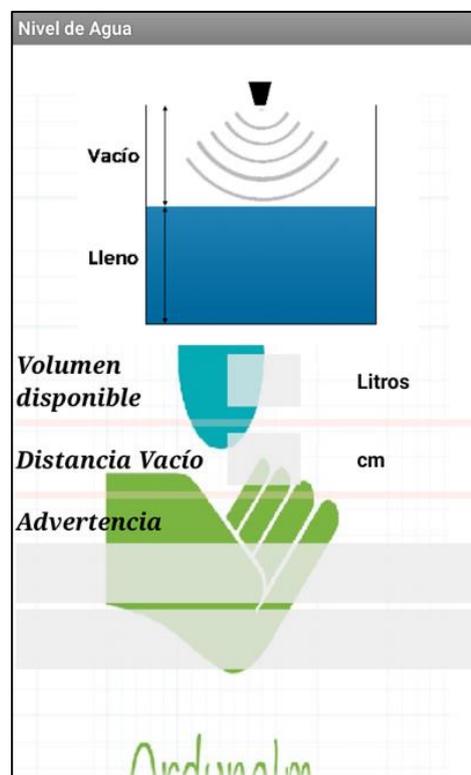


Figura 51: Interfaz de monitoreo de volumen de agua y distancia de vacío.

FUENTE: Tomado del aplicativo móvil.

se hace de forma manual, por lo tanto el vaso comunicante (tubo de diámetro 160mm) es 10 cm más alto que el rebose del tanque, de esta manera procurando el cuidado del sensor y enviando una alerta cuando el nivel de agua se acerque a este.

Se usó el tubo de 160 mm para evitar errores por el ángulo de apertura de 15° que tiene el sensor al enviar la señal de sonido hacia el espejo de agua.

Finalmente en la parte baja de la pantalla se visualiza dos cuadros de advertencias, en la primera se visualiza una advertencia si el volumen de la fuente de agua es menor a 250 litros, mostrando en letra de color rojo “Volumen de abastecimiento menor a 250 litros” y el celular vibra por diez segundos, la segunda advertencia cuando la distancia de vacío es menor a 30 cm, aparece en letra de color rojo “Volumen de abastecimiento por rebalsar, tomar medidas” y el celular vibra por diez segundos.

4.5.6. Interfaz control de válvula solenoide y bomba

En el interfaz del sensor de humedad se explica cómo se determina el tiempo de riego y se procede a un riego automático, pero en todo sistema de riego se requiere un riego independiente, ya sea por motivos ajenos a la dotación del agua para el cultivo, para ello se

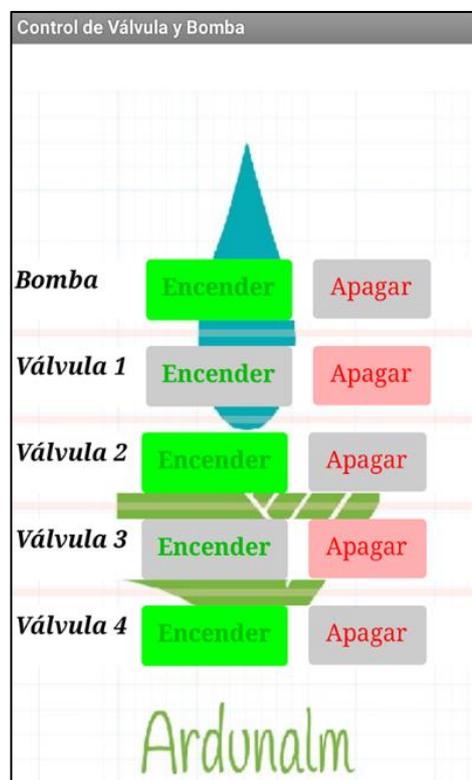


Figura 52: Interfaz de control del sistema de riego.

FUENTE: Tomado del aplicativo móvil.

creó esta interfaz, donde el encendido y apagado de la bomba y las válvulas lo maneja el usuario.

Al presionar el botón encender, este cambia de color a verde para verificar que la información fue enviada al microcontrolador y encienda el componente (bomba o electroválvula).

El botón apagar al presionarlo, este cambia a un color rosa, permitiendo verificar que la información fue enviada, además al presionarlo, el botón encender regrese a su color original. El interfaz se visualiza en la Figura 52.

4.6. MONITOREO DE LA HUMEDAD

En la Figura 53 se muestra la curva de humedad que se obtuvo a través del sensor GS1, diariamente leyendo el dato del sensor, de esta manera controlando que la humedad de suelo este cerca a la capacidad de campo, de esta manera evitando que esté a punto de marchitez, además se aprecia como el contenido de humedad va disminuyendo, hasta el momento de riego, que es calculado en base al sensor.

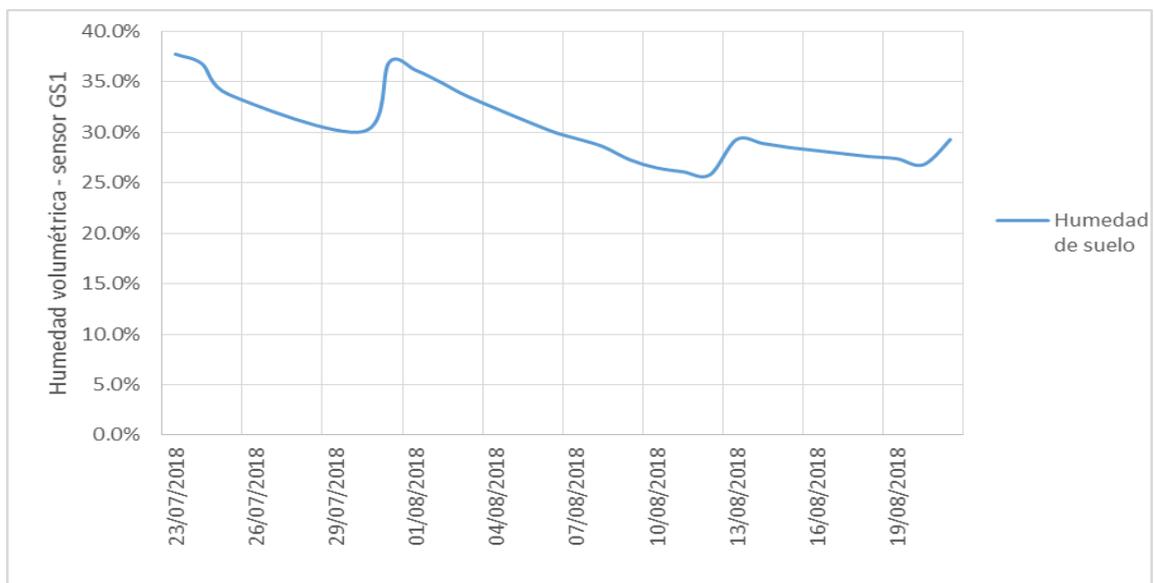


Figura 53: Monitoreo de la humedad de suelo.

FUENTE: Elaboración propia.

El monitoreo de la humedad volumétrica con el sensor GS1 es durante largos periodos de tiempo mayor a un año, porque es un sensor robusto, capaz de operar bajo tierra y no se ve afectado por los niveles de sal del suelo o los niveles de fertilizante (Williams 2016).

El sistema implementado ayuda a controlar el riego, conservando al suelo a capacidad de campo y evitando que ocurra un estrés hídrico (Mahzabin et al. 2016).

4.7. COMPARACIÓN DEL SISTEMA AUTOMATIZADO Y MANUAL

La parcela demostrativa consta de doce subunidades de igual cantidad de laterales y goteros, de las cuales cuatro han sido automatizados. Se comparó el volumen de agua suministrado para un mismo tiempo, registradas por la válvula volumétrica dando como resultado un volumen de agua para las subunidades automatizadas de 30 litros y para las subunidades de operación manual de 38 litros. El control de encendido y apagado fue mediante el aplicativo, las electroválvulas al culminar el tiempo de encendido, se cerraron evitando que el flujo de agua siga circulando por los laterales, sin embargo para las subunidades de operación manual el agua sigue circulando, hasta que se cierre la válvula plason. En los Cuadros 12 y 13 se visualiza el registro de las válvulas volumétricas para las subunidades automatizadas y manuales.

Cuadro 12: Registro del volumen de agua para las subunidades automatizadas

N°	L.I.VAL (M3)	L.F.VAL (M3)	DIFERENCIA (M3)	Vol. Aplicado (Litros)
1	48.0997	48.13	0.0303	30.3
2	26.7005	26.73015	0.029650	29.6

FUENTE: Elaboración propia.

Cuadro 13: Registro del volumen de agua para las subunidades de operación manual

N°	L.I.VAL (M3)	L.F.VAL (M3)	DIFERENCIA (M3)	Vol. Aplicado (Litros)
1	23.4682	23.50645	0.03825	38.25
2	27.61395	27.6517	0.037750	37.8

FUENTE: Elaboración propia.

Para un tiempo de 25 minutos se realizó la prueba siendo el caudal teórico de la subunidad 73.6 l/h, por lo tanto un volumen teórico de 31 litros. Con este volumen se calcula la eficiencia del sistema automatizado y el manual siendo 99% y 79% respectivamente. La eficiencia de aplicación puede ser controlada debido a que el sensor de flujo de agua, además de mostrar el caudal, puede contabilizar el volumen de agua suministrado.

V. CONCLUSIONES

- El sistema automatizado de control y monitoreo de riego ha sido implementado y funciona satisfactoriamente.
- Los sensores de flujo de agua YF-S201, FS400A, presión HK3022, humedad GS1, ultrasonido HC-204, actuadores y módulos implementados en el sistema de riego están disponibles en el mercado, son compatibles con el Arduino Mega 2560.
- Se hallaron ecuaciones y factores de conversión que pueden ser utilizados para los sensores de las mismas características y modelo, para la conversión de la señal eléctrica al parámetro físico, operando dentro del rango de error según ficha técnica de cada dispositivo, los códigos de programación de los sensores y actuadores se integraron, elaborando un solo código, donde los sensores reciben información y los actuadores envían acciones, además de alertas para el control del nivel de agua en el tanque. El código implementado al microcontrolador se vincula mediante Bluetooth con el aplicativo “Ardunalm”, además se realizaron manuales para el uso de cada sensor.
- El aplicativo “Ardunalm” y el sistema de riego automatizado, permiten eliminar el control del riego manual y monitorea los parámetros de caudal, presión, humedad de suelo y nivel de agua en tiempo real, además permite el ahorro de tiempo en la manipulación del sistema (apertura y cierre de válvulas), aumentado la eficiencia de aplicación.

VI. RECOMENDACIONES

- Para un futuro trabajo en este proyecto, se recomienda implementar un microcontrolador con WiFi, además de un datalogger.
- Reemplazar la fuente de suministro de energía por sistemas alternativos, tales como paneles solares o un conjunto de baterías.
- Desarrollar una interfaz gráfica para los datos de sensor de humedad, para visualizar el comportamiento en el suelo.
- Contemplar la posibilidad de extender el área de terreno que se ha considerado en este proyecto, porque los sensores que se han escogido pueden operar en dimensiones más grandes.
- El sensor de presión HK3022 como se mencionó en los resultados presentaba una descalibración paulatina, por ello según experiencia en campo si se desea usar este sensor económico, realizar calibraciones mensuales, además de proteger a la exposición de los rayos solares y el rocío del agua que oxida la parte externa del sensor.
- Crear una librería en el software IDE para el uso del sensor GS3.

VII. REFERENCIAS BIBLIOGRAFICAS

- Almaraz, F; Maz Machado, A; López Esteban, C. 2015. Tecnología móvil y enseñanza de las matemáticas:: una experiencia de aplicación de App Inventor. Epsilon: Revista de la Sociedad Andaluza de Educación Matemática " 32(91):77-86.
- Arduino. 2017. Arduino Mega 2560 Rev3 (en línea, sitio web). Consultado 9 ene. 2018. Disponible en <https://store.arduino.cc/usa/arduino-mega-2560-rev3>.
- _____. 2017. Compare board specs (en línea, sitio web). Consultado 11 oct. 2017. Disponible en <https://www.arduino.cc/en/Products/Compare>.
- Cervantes, W. 2016. Diseño e implementación de sistema de riego automatizado y controlado por una placa arduino para la finca “ La Lucia ”. s.l., Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López. 10-11 p.
- Cifuentes Meza, ATC. 2016. Requerimiento de agua para el césped americano (*Stenotaphrum secundatum*) empleando riego por goteo subterráneo, durante el otoño en la UNALM. s.l., Universidad Nacional Agraria La Molina. 36-39 p.
- Crespo, E. 2014. UART y USB en Arduino | Aprendiendo Arduino (en línea, sitio web). Consultado 5 oct. 2017. Disponible en <https://aprendiendoarduino.wordpress.com/2016/11/09/uart-y-usb-en-arduino/>.
- David, J; Martín, QP; Bravo, D; Yaime, A. 2015. Monitoreo remoto a sistemas de riego por bombeo eléctrico a cultivos de arroz en el departamento del Huila. Revista Ingeniería y Región 14(2):55-63.
- Edaphic, scientific. 2004. GS1 Soil Moisture Sensor Operator’s Manual. Decagon Devices 1:3-10.
- Herrero Herranz, José Carlos Sánchez Allende, J. 2015. Tecnología y desarrollo. Tecnología y desarrollo 13(0):21.
- INEI. 2012. IV Censo Nacional Agropecuario. Resultados Definitivos. IV Censo Nacional Agropecuario 1:62.

- Ingale, HT; Kasat, NN. 2012. Automated Irrigation System. International Journal of Engineering Research and Development 4(11):51-54.
- Kalfountzos, D; Vyrlas, P. 2002. Water Saving and Yield Increase of Sugar Beet With Subsurface Drip Irrigation. Revista Global Nest 4(2):85-91.
- Lion, GC. 2004. Automatización del riego. Universidad Politencnica de Madrid :202-205.
- López, EM; Choez, JL. 2016. Sistema de riego por goteo automático utilizando una red de sensores inalámbricos. s.l., s.e., vol.7. p. 5-9.
- López López, Y. 2015. Herramienta móvil para captura y procesamiento geográfico y estadístico para asociaciones de productores en invernadero. s.l., Universidad Autónoma del Estado de México. 38-45 p.
- Lucia, D; Garces, M; Alfonso, J; Nuñez, Y. 2017. Estructura teórica entorno al sistema microcontrolado de riego por goteo. Universidad de La Guajira :7.
- Mahzabin, A; Taziz, CA; Amina, MH; Gloria, M; Zishan, MSR. 2016. Design and Implementation of an Automatic Irrigation System Based on Monitoring Soil Moisture. Electrical & Information Engineering Department, Covenant University, Nigeria 3(10):159-162.
- MaxElectronica. 2018. Sensor de Flujo de Agua 30L/min Efecto Hall Modelo YF-S201 (en línea, sitio web). Consultado 26 jun. 2018. Disponible en <http://www.maxelectronica.cl/sensores/121-sensor-de-flujo-de-agua-30lmin-efecto-hall-modelo-yf-s201.html>.
- Mecatronium. 2015. Tutorial sensor de flujo YF-S201 - Mecatronium Chips (en línea, sitio web). Consultado 7 ene. 2018. Disponible en <http://chips.mecatronium.com/tutorial-sensor-de-flujo-yf-s201/>.
- _____. 2017. Módulo Relay, 4 canales 5VDC (en línea, sitio web). Consultado 9 oct. 2017. Disponible en <http://www.naylampmechatronics.com/drivers/152-modulo-relay-4-canales-5vdc.html>.
- Mechatronics, N. 2017. Sensor de presión HK3022 0.5MPa (en línea, sitio web). Consultado 9 oct. 2017. Disponible en <http://www.naylampmechatronics.com/sensores-liquido/318-sensor-de-presion-hk3022->

05mpa.html?search_query=sensor+de+presion+agua&results=6.

_____. 2017. Válvula solenoide 1/2" 12V DC (en línea, sitio web). Consultado 9 oct. 2017.

Disponible en <http://www.naylampmechatronics.com/valvulas/314-valvula-solenoide-12-12v.html>.

Moreno García, F; Bedoya Fierro, J; López Martínez, G arturo. 2013. Modelo a escala de un sistema de riego automatizado , alimentado con energía solar fotovoltaica : nueva perspectiva para el desarrollo agroindustrial colombiano. Universidad Distrtial Francisco José de Caldas 17:33-47.

Murillo, ÁM; Paredes, KR; Esquicha, J-D. 2015. Sistema Didáctico de Enseñanza Introductoria a la Programación usando Scratch , PSeInt , Processing y Arduino. 13th LACCEI Annual International Conference: "Engineering Education Facing the Grand Challenges, What Are We Doing?" :1-2.

Naylamp, M. 2018. Sensor Ultrasonido HC-SR04 - Naylamp Mechatronics - Perú (en línea, sitio web). Consultado 14 abr. 2018. Disponible en http://www.naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html?search_query=ultrasonico+hc-sr04&results=4.

Patra, S. 2017. Serial Communication between Arduino and Bluetooth Module (Hc05) via Android Device. Grenze Scientific Society 1:7-10.

Peters, T. 2016. Use of Soil Moisture Sensors for Irrigation Scheduling. United States Department of Agriculture :1-6.

Recktenwald, GW; State, P; Hall, DE; Tech, L. 2011. Using Arduino as a Platform for Programming , Design. American Society for Engineering Education 1:3.

Sinobas, LR; Rodríguez, MG. 2005. A Review of Subsurface Drip Irrigation and Its Management. Technical University of Madrid 1:25.

Sood, R; Kaur, M; Lenka, H. 2013. Design and Development of Automatic Water Flow Meter. International Journal of Computer Science, Engineering and Applications (IJCSEA) 3(3):49-59.

Torrente Artero, Ó. 2013. Arduino - Curso práctico de formación. *In Alfaomega Grupo Editor, SA (ed.)*. México, s.e. p. 7,61-65,129.

- Wang, S; Jia, L; He, Y. 2016. Intelligent irrigation system based on S3C6410. Revista de la Facultad de Ingenieria 31(6):219-232.
- Wei, Y; Yildirim, P; Weekly, ICTI; Henley, WE. 2014. Aurduino Mega 2560. Revista Impatto Zero 33(2):81-87.
- Williams, D. 2016. Managing Active Forestry Lands for Increased Water Retention: a New Approach for Protecting Summer Water Supplies in the Western United States. s.l., The Evergreen State College. 23-52 p.
- Zieris, H; Gerstberger, H; Müller, W. 2015. Using Arduino -Based Experiments to Integrate Computer Science Education and Natural Science. Dept. of Physics Education, Dept. of Media Education and Visualization Univ. of Education 1:381-389.

VIII. ANEXOS

ANEXO 1: MANUAL DE CALIBRACIÓN DE LOS SENSORES DE PRESIÓN

El procedimiento de calibración del sensor de presión consistió en instalar en un collarín de doble entrada en una tubería, cada sensor con un punto de prueba, en este último se colocaba el manómetro para comparar la presión en metros con la señal en Bits que arroja el sensor, de esta manera correlacionando los datos en un análisis de regresión simple, hallando una ecuación lineal que pueda asociar la presión en metros con la señal analógica del sensor.

Como primer paso fue instalar el sensor en el banco de pruebas del laboratorio de Riego y Drenaje, posterior se conectó el sensor al microcontrolador y este al computador para realizar el código de programación que se mostrará a continuación. El diseño electrónico de la conexión se encuentra en la Figura 36.

```
int sensorPresion=analogRead(A1); //Se declara la variable sensorVal y se asocia
a la lectura del pin analógico A1
void setup() {
    Serial.begin(9600); //Se activa la comunicación del puerto serial
}
void loop(){
    Serial.print("Sensor Value: "); //La frase que esta entre comillas se mostrara
en el monitor serial
    Serial.println(sensorPresion); //Se imprimirá la lectura del sensor de presión
en monitor serial
    delay(1000); //Intervalo para mostrar los datos en el monitor serial de
1000ms
}
```

Este código nos muestra los valores leídos por el sensor en el monitor serial del software IDE, el cual se apreciara en la Figura 54.

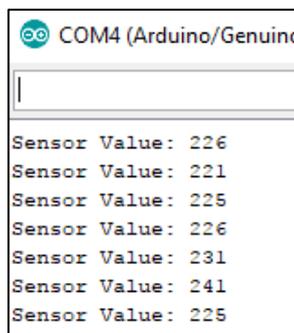


Figura 54: Valores de la señal analógica para el sensor de la subunidad 4

Obteniendo los análisis de regresión simple de cada sensor, el cual compara las presiones tabuladas con la señal analógica recibida en el monitor serial, estos análisis son mostrados en las figuras siguientes, teniendo como coeficiente de correlación para todas estas mayores a 0.99.

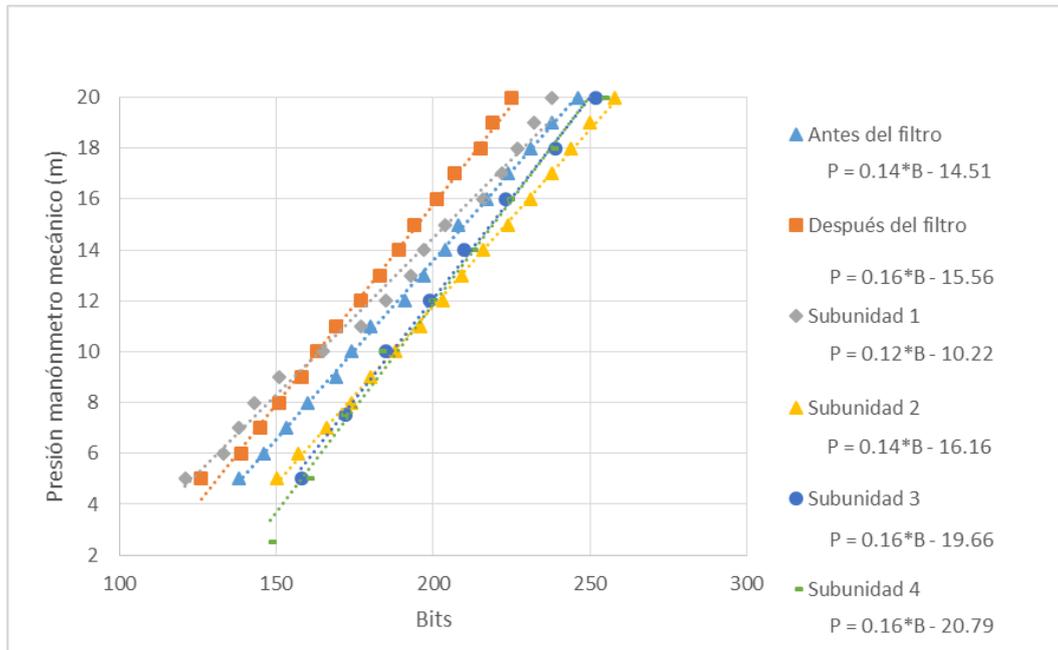


Figura 55: Ecuación de ajuste de los transductores de presión instalados.

ANEXO 2: MANUAL DE CALIBRACION DE LOS SENSORES FLUJO DE AGUA (CAUDAL).

El sensor de flujo de agua fue instalado en el banco de pruebas, en una línea de tubería con salida de manguera, para que sea depositado el volumen de agua en un recipiente de 20 litros. Se marcó el recipiente cada 5 litros, para tomar la lectura de la frecuencia o número de pulsos en ese instante.

Se procedió a conectar el sensor al microcontrolador y se realizó el código de programación mostrado a continuación, el cual tiene como finalidad contar el número de pulsos para un volumen determinado. El diseño electrónico de la conexión se encuentra en la Figura 37.

```
volatile long NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 2; //Sensor conectado en el pin 2

//---Función que se ejecuta en interrupción-----
void ContarPulsos ()
{
    NumPulsos++; //incrementamos la variable de pulsos
}
```

```

//---Función para obtener frecuencia de los pulsos-----

void setup()
{
  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0,ContarPulsos,RISING);//(Interrupción 0(Pin2),función,Flanco
de subida)
  interrupts(); //Habilitamos las interrupciones
}

void loop ()
{
  //-----Enviamos por el puerto serie-----
  Serial.print ("Numero de Pulsos = ");
  Serial.println (NumPulsos);
  delay(170);//Las lecturas tendrán un intervalo de 170ms, debido al tiempo
acumulado que genera el código de programación final sea de 1000 ms
}

```

El programa solo muestra la cantidad de pulsos del sensor como se visualiza en la Figura 61, para realizar una muestra de pulsos y volumen es necesario resetear y a continuación empezar a llenar el recipiente, en otras palabras verificar que cuando se empiece a medir el agua la cantidad de pulsos debe estar en cero.

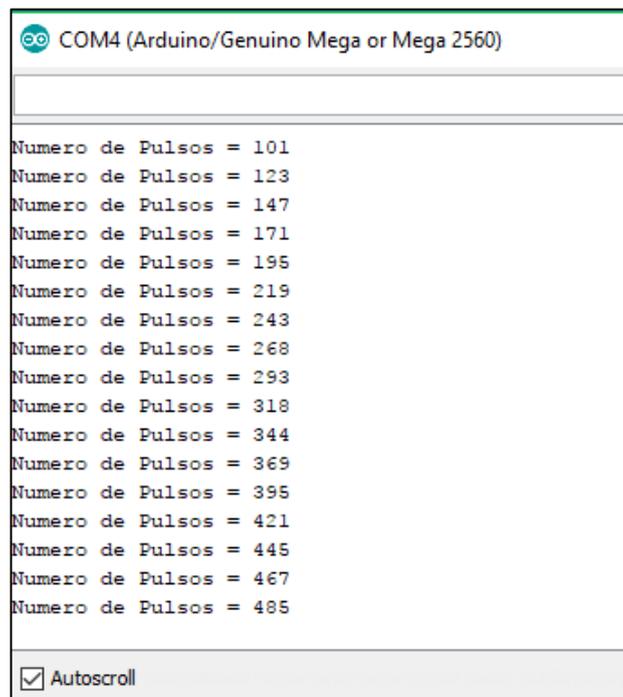


Figura 56: Monitoreo de la frecuencia de pulsos para la calibración del sensor de flujo de agua.

Para hallar el factor de conversión de cantidad de pulsos a volumen de agua se usó la siguiente formula: $K = \frac{n^{\circ}Pulsos}{Volumen*60}$

Para tener un valor más exacto es mejor tomar varias muestras y finalmente quedarnos con el promedio de estas, en nuestro caso nuestros resultados son los siguientes:

Cuadro 14: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 1

CALIBRACION SENSOR SUBUNIDAD 1		
Volumen (litros)	Cantidad de pulsos (170 ms)	k(170ms)
5	391	1.30
10	758	1.26
15	1126	1.25
20	1493	1.24
5	378	1.26
10	741	1.24
15	1102	1.22
20	1475	1.23
5	376	1.25
10	740	1.23
15	1116	1.24
20	1473	1.23
	Promedio	1.25

Cuadro 15: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 2

CALIBRACION SENSOR SUBUNIDAD 2		
Volumen (litros)	Cantidad de pulsos (170 ms)	k(170ms)
5	332	1.11
10	653	1.09
15	972	1.08
20	1304	1.09
5	340	1.13
10	660	1.1
15	978	1.09
20	1308	1.09
5	331	1.1
10	651	1.08
15	982	1.09
20	1313	1.09
	Promedio	1.1

Cuadro 16: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 3

CALIBRACION SENSOR SUBUNIDAD 3		
Volumen (litros)	Cantidad de pulsos (170 ms)	k(170ms)
5	329	1.1
10	663	1.11
15	986	1.1
20	1307	1.09
5	326	1.09
10	659	1.1
15	981	1.09
20	1302	1.08
5	329	1.1
10	661	1.1
15	982	1.09
20	1300	1.08
	Promedio	1.09

Cuadro 17: Resultado del factor de conversión (k) para el sensor de flujo de la subunidad 4

CALIBRACION SENSOR SUBUNIDAD 4		
Volumen (litros)	Cantidad de pulsos (170 ms)	k(170ms)
5	363	1.21
10	695	1.16
15	1024	1.14
20	1352	1.13
5	352	1.17
10	681	1.13
15	1018	1.13
20	1343	1.12
5	361	1.20
10	689	1.15
15	1015	1.13
20	1339	1.12
	Promedio	1.15

Cuadro 18: Resultado del factor de conversión (k) para el sensor de flujo del cabezal de riego

CALIBRACION SENSOR CABEZAL		
Volumen (litros)	Cantidad de pulsos (170 ms)	k(170ms)
5	222	0.74
10	430	0.72
15	638	0.71
20	856	0.71
5	227	0.76
10	439	0.73
15	650	0.72
20	859	0.72
5	218	0.73
10	426	0.71
15	640	0.71
20	854	0.71
	Promedio	0.72

Posterior a la obtención del factor de conversión se añade este al siguiente código, para el monitoreo del caudal.

```
volatile int NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 2; //Sensor conectado en el pin 2
float factor_conversion=1.15; //para convertir de frecuencia a caudal

//---Función que se ejecuta en interrupción-----
void ContarPulsos ()
{
  NumPulsos++; //incrementamos la variable de pulsos
}

//---Función para obtener frecuencia de los pulsos-----
int ObtenerFrecuencia()
{
  int frecuencia;
  NumPulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(170); //muestra de 170 milisegundo
  noInterrupts(); //Desabilitamos las interrupciones
  frecuencia=NumPulsos; //Hz(pulsos por segundo)
  return frecuencia;
}

void setup()
{
  Serial.begin(9600);
  pinMode(PinSensor, INPUT); //Sensor como dispositivo de entrada
}
```

```

    attachInterrupt(digitalPinToInterrupt(2), ContarPulsos, RISING);
    //(Interrupcion 0(Pin2),funcion,Flanco de subida)
}

void loop ()
{
    float frecuencia=ObtenerFrecuencia(); //obtenemos la Frecuencia de los pulsos
    en Hz
    float caudal_L_m=frecuencia/factor_conversion; //calculamos el caudal en L/m
    float caudal_L_s=caudal_L_m/60; //calculamos el caudal en L/h

    //-----Enviamos por el puerto serie-----
    Serial.print ("FrecuenciaPulsos: ");
    Serial.print (frecuencia,0);
    Serial.print ("Hz\tCaudal: ");
    Serial.print (caudal_L_m,3);
    Serial.print (" L/m\t");
    Serial.print (caudal_L_s,3);
    Serial.println ("L/s");
}

```

ANEXO 3: MANUAL DE CALIBRACION DEL SENSOR DE HUMEDAD GS1

El proceso de calibración de este sensor comprendió de 19 días extrayendo siete muestras de suelo, fue calibrado in situ. El diseño electrónico de la conexión se encuentra en la Figura 35. Su código de programación para la lectura de la señal analógica se mostrará a continuación.

```

int sensorGS1;//Declaración de la variable
void setup() {
    Serial.begin(9600);//Activación de la comunicación serial
}
void loop() {
    sensorGS1= analogRead(A0);//Lectura del sensor analógico
    Serial.print("Señal analógica sensor GS1 =");
    Serial.println(sensorGS1);//Impresión de los datos del sensor
    delay(1000);
}

```

Los valores que el sensor muestra por el monitor serial se visualiza en la Figura 57.

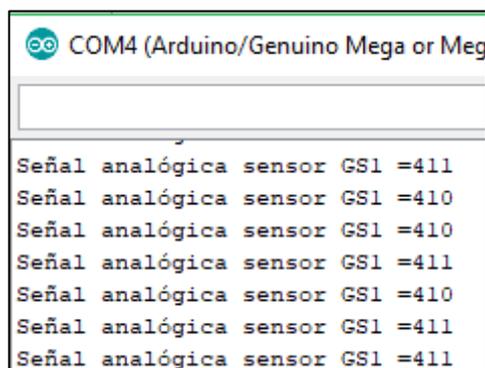


Figura 57: Valores de la señal analógica para el sensor de humedad GS1

Se tomaron tres muestras de suelo a 10 cm alrededor del sensor, para obtener un promedio y sea comparado con la señal de salida del sensor. Estos datos están en el Cuadro 19.

Cuadro 19: Porcentaje de humedad de suelo y señal analógica del sensor de humedad GS1.

Fecha	Muestra	W húmedo	W seco	% Humedad volumétrica	PROMEDIO	Bits
09/04/2018	M1	111.3	92	33.57%	32.4%	460
	M2	108.6	90.6	31.79%		
	M3	85.5	71.3	31.87%		
16/04/2018	M1	83.66	68.6	35.13%	32.2%	455
	M2	74.21	63	28.47%		
	M3	84.99	70.5	32.89%		
18/04/2018	M1	56.9	48	29.67%	31.9%	449
	M2	116.9	96.3	34.23%		
	M3	67.5	56.3	31.83%		
19/04/2018	M1	119.2	97.8	35.01%	31.3%	443
	M2	76.6	62.3	36.73%		
	M3	59.3	52.1	22.11%		
23/04/2018	M1	72	59.9	32.32%	30.4%	425
	M2	50.7	42	33.14%		
	M3	47.8	41.2	25.63%		
26/04/2018	M1	61.3	51.2	31.56%	30.0%	410
	M2	104	87.6	29.95%		
	M3	107.9	91.6	28.47%		
02/05/2018	M1	42.5	36.2	27.85%	28.2%	386
	M2	143	121.6	28.16%		
	M3	61.6	52.3	28.45%		

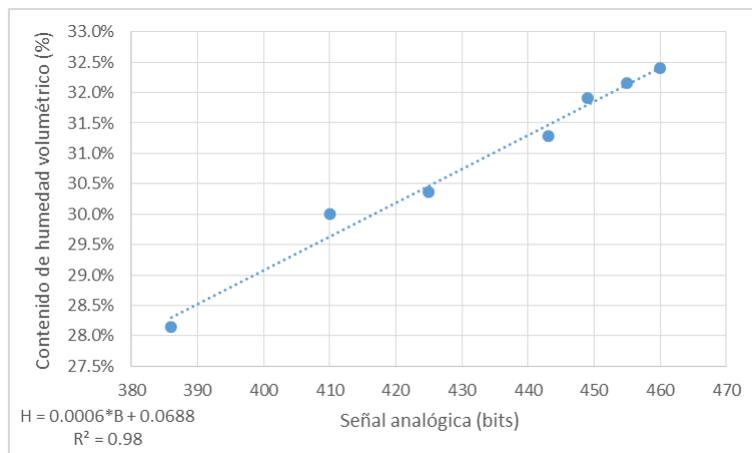


Figura 58: Análisis de regresión simple para la ecuación de ajuste del sensor de humedad GS1

En la Figura 58 se visualiza el análisis de regresión simple con los datos obtenidos de humedad de suelo y señal analógica del sensor siendo el R^2 igual a 0.98.

ANEXO 4: MANUAL DE CALIIBRACION DEL SENSOR DE ULTRASONIDO

El proceso de calibración de este sensor comprendió de la programación en base a la velocidad del sonido, para obtener la medida impresa en el monitor serial y corroborada con una regla milimétrica. El diseño electrónico de la conexión se encuentra en la Figura 38. Su código de programación se mostrará a continuación.

```
int trigger = 10; //Asignación del puerto Trigger
int echo = 8; // Asignación del puerto Echo
float tiempo, distancia; // Creación de las variables tiempo y distancia

void setup() {
  Serial.begin(9600); //Activación del puerto serial
  pinMode(trigger, OUTPUT); //Modo salida pin Trigger
  pinMode(echo, INPUT); //Modo entrada pin Echo
}

void loop() {
  digitalWrite(trigger, LOW); //Estado apagado pin Trigger
  delayMicroseconds(2);
  digitalWrite(trigger, HIGH); //Estado alto pin Trigger
  delayMicroseconds(10);
  digitalWrite(trigger, LOW); // Estado apagado pun Trigger

  tiempo = pulseIn(echo, HIGH); //Medición del tiempo de recepción del pin Echo
  distancia = (tiempo / 2) / 29.1; // Medición de la distancia

  Serial.print("Distancia= "); // Impresión de los resultados
  Serial.print(distancia);
  Serial.println("cm");

  delay(500);
}
```

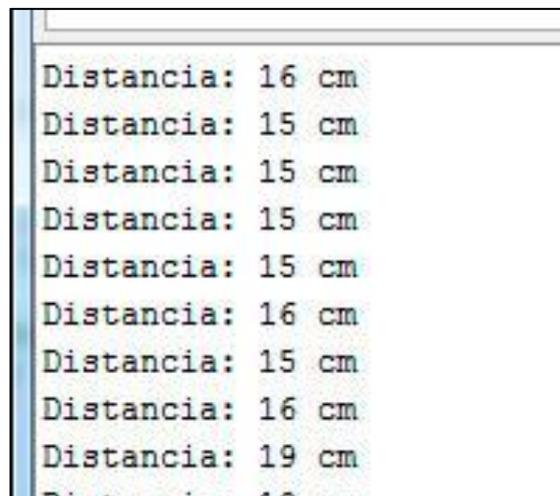


Figura 59: Monitor serial del sensor de ultrasonido

ANEXO 5: CÓDIGO DE PROGRAMACIÓN PARA EL PRENDIDO Y APAGADO DE UN RELÉ

La programación para el prendido o apagado de un relé está basado en condicionales, asignando una letra. El circuito que debe usarse para el código que se mostrara a continuación es normalmente abierto. El diseño electrónico de la conexión se encuentra en la Figura 39.

```
int rele = 8; //Asocia el rele al pin 8
char valor; //Se crea la variable valor
void setup() {
  Serial.begin(9600); //Se activa el puerto serial
  pinMode(rele, OUTPUT); //Se habilita el rele como variable de salida
  digitalWrite(rele, HIGH); //Se inicia el rele en estado apagado
}
void loop() {
  if (Serial.available())
  {
    valor=Serial.read(); //Se activa la lectura del valor
    if(valor=='A') //Si recibe la letra A
    {
      digitalWrite(rele, LOW); //Se prende el rele
    }
    if(valor=='B') //Si se recibe la letra B
    {
      digitalWrite(rele, HIGH); //Se apaga el rele
    }
  }
  {
    delay(1000);
  }
}
```

ANEXO 6: CÓDIGO FUENTE ARDUINO IMPLEMENTADO EN EL MONITOREO Y CONTROL DE RIEGO

```
int sensorP1; //Inicializa la variable para los sensores de presión
int sensorP2;
int sensorP3;
int sensorP4;
int sensorP5;
int sensorP6;
volatile int NumPulsos2; //variable para la cantidad de pulsos recibidos
volatile int NumPulsos3;
volatile int NumPulsos4;
volatile int NumPulsos5;
volatile int NumPulsos6;
int sensorQ2 = 2; //Inicializa la variable para los sensor de flujo de agua y
se le asigna el pin digital
float factor_conversion2=0.72; // Coeficiente de corrección para pasar de Hz a
Caudal
int sensorQ3 = 3;
float factor_conversion3=1.25;
```

```

int sensorQ4 = 18;
float factor_conversion4=1.1;
int sensorQ5 = 19;
float factor_conversion5=1.09;
int sensorQ6 = 20;
float factor_conversion6=1.15;

int bombapren = 50; //Inicializa las variables para la bomba y válvulas con sus
pines respectivos
int bombaapa = 51;
int valv1 = 24;
int valv2 = 26;
int valv3 = 28;
int valv4 = 30;
int sensorGS1; //Inicializa la variable del sensor de humedad
int trigger = 42; //Pin digital 13 para el Trigger del sensor
int echo = 43; //Pin digital 12 para el echo del sensor
int led = 7;

float tiempo, distancia;//se crea las variables tiempo y distancia
char valor; //Esta función recibe un parámetro char como y enciende / apaga el
LED dependiendo de su valor

void ContarPulsos2 () // Se crea una variable para contar los pulsos del sensor
de flujo de agua
{
    NumPulsos2++; //incrementamos la variable de pulsos
}
int ObtenerFrecuencia2() //Se crea una variable, llamandola ObtenerFrecuencia
{
    int frecuencia2; // Se crea una variable, llamandola frecuencia
    NumPulsos2 = 0; //Ponemos a 0 el número de pulsos
    interrupts(); //Habilitamos las interrupciones
    delay(170); //muestra de 170 milisegundos, esto debido a que los delays se
suman y llegan al promedio del segundo
    noInterrupts(); //Desabilitamos las interrupciones
    frecuencia2=NumPulsos2; //Hz(pulsos por segundo)
    return frecuencia2; //Termine una función y devuelva un valor de una función
a la función de llamada
}

void ContarPulsos3 () //Se repita para todos los sensores de flujo de agua
{
    NumPulsos3++; //incrementamos la variable de pulsos
}
int ObtenerFrecuencia3()
{
    int frecuencia3;
    NumPulsos3 = 0; //Ponemos a 0 el número de pulsos
    interrupts(); //Habilitamos las interrupciones
    delay(170); //muestra de 170 milisegundo
    noInterrupts(); //Desabilitamos las interrupciones
    frecuencia3=NumPulsos3; //Hz(pulsos por segundo)
    return frecuencia3;
}

void ContarPulsos4 ()
{

```

```

    NumPulsos4++; //incrementamos la variable de pulsos
}
int ObtenerFrecuencia4()
{
    int frecuencia4;
    NumPulsos4 = 0; //Ponemos a 0 el número de pulsos
    interrupts(); //Habilitamos las interrupciones
    delay(170); //muestra de 170 milisegundo
    noInterrupts(); //Desabilitamos las interrupciones
    frecuencia4=NumPulsos4; //Hz(pulsos por segundo)
    return frecuencia4;
}

void ContarPulsos5 ()
{
    NumPulsos5++; //incrementamos la variable de pulsos
}
int ObtenerFrecuencia5()
{
    int frecuencia5;
    NumPulsos5 = 0; //Ponemos a 0 el número de pulsos
    interrupts(); //Habilitamos las interrupciones
    delay(170); //muestra de 170 milisegundo
    noInterrupts(); //Desabilitamos las interrupciones
    frecuencia5=NumPulsos5; //Hz(pulsos por segundo)
    return frecuencia5;
}

void ContarPulsos6 ()
{
    NumPulsos6++; //incrementamos la variable de pulsos
}
int ObtenerFrecuencia6()
{
    int frecuencia6;
    NumPulsos6 = 0; //Ponemos a 0 el número de pulsos
    interrupts(); //Habilitamos las interrupciones
    delay(170); //muestra de 170 milisegundo
    noInterrupts(); //Desabilitamos las interrupciones
    frecuencia6=NumPulsos6; //Hz(pulsos por segundo)
    return frecuencia6;
}
void setup() {
    pinMode(bombapren,OUTPUT); //Definimos las variables de bomba y válvulas como
    salida, enviamos información.
    pinMode(bombaapa,OUTPUT);
    pinMode(valv1,OUTPUT);
    pinMode(valv2,OUTPUT);
    pinMode(valv3,OUTPUT);
    pinMode(valv4,OUTPUT);
    pinMode(trigger, OUTPUT); // pin como salida
    pinMode(echo, INPUT); // pin como entrada
    Serial.begin(9600); //Activamos el monitor serial
    digitalWrite(bombapren,HIGH); //Cuando el microcontrolador se encienda las
    variables de valvulas y bomba esten apagadas.
    digitalWrite(bombaapa,HIGH);
    digitalWrite(valv1,HIGH);
    digitalWrite(valv2,HIGH);
}

```

```

digitalWrite(valv3,HIGH);
digitalWrite(valv4,HIGH);

pinMode(7, INPUT);//alarma de la distancia(encenderá el led)

    pinMode(sensorQ2, INPUT); //Definimos las variables de sensor de flujo de
agua como entrada, recibimos información.
    attachInterrupt(digitalPinToInterrupt(2),ContarPulsos2,RISING); //activamos
las interrupciones, se define el pin del sensor, la variable y el modo cuando
se va activar la interrupción.
    pinMode(sensorQ3, INPUT);
    attachInterrupt(digitalPinToInterrupt(3),ContarPulsos3,RISING);
    pinMode(sensorQ4, INPUT);
    attachInterrupt(digitalPinToInterrupt(18),ContarPulsos4,RISING);
    pinMode(sensorQ5, INPUT);
    attachInterrupt(digitalPinToInterrupt(19),ContarPulsos5,RISING);
    pinMode(sensorQ6, INPUT);
    attachInterrupt(digitalPinToInterrupt(20),ContarPulsos6,RISING);
}

void loop() {
if (Serial.available()) //Si el monitor serial esta activado
{
valor = Serial.read();
if (valor == 'A') //Si el microcontrolador recibe la letra 'A'
{
digitalWrite(bombapren,LOW); //El rele NO asociado al contactor se prende.
digitalWrite(bombaapa,HIGH); //El rele NC asociado al rele termico se apaga
}
if (valor == 'B') //Si el microcontrolador recibe la letra 'B'
{
digitalWrite(bombaapa,LOW); //El rele NO asociado al contactor se apaga.
digitalWrite(bombapren,HIGH);//El rele NC asociado al rele termico se
prende
}
if (valor == 'C') //Si el microcontrolador recibe la letra 'C'
{
digitalWrite(valv1,LOW); //La valvula 1 se prende
}
if (valor == 'D') //Si el microcontrolador recibe la letra 'D'
{
digitalWrite(valv1,HIGH); //La valvula 1 se apaga
}
if (valor == 'E')//Si el microcontrolador recibe la letra 'E'
{
digitalWrite(valv2,LOW);//La valvula 2 se prende
}
if (valor == 'F')//Si el microcontrolador recibe la letra 'F'
{
digitalWrite(valv2,HIGH);//La valvula 2 se apaga
}
if (valor == 'G')//Si el microcontrolador recibe la letra 'G'
{
digitalWrite(valv3,LOW);//La valvula 3 se prende
}
if (valor == 'H')//Si el microcontrolador recibe la letra 'H'
{
digitalWrite(valv3,HIGH);//La valvula 3 se apaga
}
}
}

```

```

    }
    if (valor == 'I')//Si el microcontrolador recibe la letra 'I'
    {
        digitalWrite(valv4,LOW);//La valvula 4 se prende
    }
    if (valor == 'J')//Si el microcontrolador recibe la letra 'J'
    {
        digitalWrite(valv4,HIGH);//La valvula 4 se apaga
    }
}

sensorP1=analogRead(A1); //Se asigna a los sensores de presion los pines
analógicos
sensorP2=analogRead(A2);
sensorP3=analogRead(A3);
sensorP4=analogRead(A4);
sensorP5=analogRead(A5);
sensorP6=analogRead(A6);
sensorGS1= analogRead(A7);

float presion1_m = (sensorP1*0.1405)-14.514; //Mediante la formula regresion
simple resultado de la calibracion de transforma bits a presion en metros

float presion2_m = (sensorP2*0.1569)-15.555;

float presion3_m = (sensorP3*0.1236)-10.22;

float presion4_m = (sensorP4*0.1396)-16.155;

float presion5_m = (sensorP5*0.1586)-19.66;

float presion6_m = (sensorP6*0.1633)-20.792;

float frecuencia2=ObtenerFrecuencia2(); //obtenemos la Frecuencia de los
pulsos en Hz, se añade un factor de correccion calculado por la variacion de
muestra de 170ms
float caudal_L_m2=frecuencia2/factor_conversion2; //calculamos el caudal en
L/m
float caudal_L_s2=caudal_L_m2/60;//calculas el caudal en Litros por segundo

float frecuencia3=ObtenerFrecuencia3(); //obtenemos la Frecuencia de los
pulsos en Hz
float caudal_L_m3=frecuencia3/factor_conversion3; //calculamos el caudal en
L/m
float caudal_L_s3=caudal_L_m3/60;

float frecuencia4=ObtenerFrecuencia4(); //obtenemos la Frecuencia de los
pulsos en Hz
float caudal_L_m4=frecuencia4/factor_conversion4; //calculamos el caudal en
L/m
float caudal_L_s4=caudal_L_m4/60;

float frecuencia5=ObtenerFrecuencia5(); //obtenemos la Frecuencia de los
pulsos en Hz
float caudal_L_m5=frecuencia5/factor_conversion5; //calculamos el caudal en
L/m
float caudal_L_s5=caudal_L_m5/60;

```

```

    float frecuencia6=ObtenerFrecuencia6(); //obtenemos la Frecuencia de los
pulsos en Hz
    float caudal_L_m6=frecuencia6/factor_conversion6; //calculamos el caudal en
L/m
    float caudal_L_s6=caudal_L_m6/60;
    float humedad=(0.0006*sensorGS1+0.0688)*100;//Ecuación de regresión formulada
en el codigo

    digitalWrite(trigger, LOW);
    delayMicroseconds(2); //Enviamos un pulso de 2us
    digitalWrite(trigger, HIGH);
    delayMicroseconds(10); //Enviamos un pulso de 10us
    digitalWrite(trigger, LOW);

    tiempo = pulseIn(echo, HIGH);
    distancia = (tiempo / 2) / 29.1; //formula para calcular la distancia,
teniendo como base la velocidad del sonido
    float volumenlitros=3.1416*55*55*(140-distancia)/1000;

if(volumenlitros<250) //si el volumen es menor a 250 litros encender la led
{
    digitalWrite(7,HIGH);
    digitalWrite(bombaapa,LOW); //El rele NO asociado al contactor se apaga.
    digitalWrite(bombapren,HIGH);//El rele NC asociado al rele termico se
enciende.
}
else
{
    digitalWrite(7,LOW);
}

{

    Serial.print(presion1_m); //se imprime los datos procesados por el
microcontrolador
    Serial.print(","); // se añade la separacion por coma entre cada variable
    Serial.print(presion2_m);
    Serial.print(",");
    Serial.print(presion3_m);
    Serial.print(",");
    Serial.print(presion4_m);
    Serial.print(",");
    Serial.print(presion5_m);
    Serial.print(",");
    Serial.print(presion6_m);
    Serial.print(",");
    Serial.print (caudal_L_s2,3);
    Serial.print(",");
    Serial.print (caudal_L_s3,3);
    Serial.print(",");
    Serial.print (caudal_L_s4,3);
    Serial.print(",");
    Serial.print (caudal_L_s5,3);
    Serial.print(",");
    Serial.print (caudal_L_s6,3);
    Serial.print(",");
}

```

```

Serial.print(humedad);
Serial.print(",");
Serial.print(distancia);
  Serial.print(",");
Serial.println(volumenlitros);

delay(170);
}
}

```

ANEXO 7: PRESUPUESTO PARA LA AUTOMATIZACIÓN

PRESUPUESTO DEL PROYECTO DE INVESTIGACION "IMPLEMENTACIÓN, CONTROL Y MONITOREO DE UN SISTEMA DE RIEGO POR GOTEO SUBTERRANEO CON MICROCONTROLADORES"				
	Descripción	Cantidad	P. unitario	P.total
1.	Sensores			S/. 1,410.00
1.1.	Sensor ultrasonido	2	S/. 10.00	S/. 20.00
1.2.	Sensor GS1	1	S/. 700.00	S/. 700.00
1.3.	Sensor de flujo de agua EFECTO HALL	6	S/. 35.00	S/. 210.00
1.4.	Sensor de presión HK3022	6	S/. 80.00	S/. 480.00
2.	Actuadores			S/. 173.00
2.1.	Módulo relay 4 canales	4	S/. 12.00	S/. 48.00
2.2.	Válvula solenoide	5	S/. 25.00	S/. 125.00
3.	Otros			S/. 1,605.00
3.3.	Protoboard blanco 830 puntos	1	S/. 12.00	S/. 12.00
3.4.	Cable Dupont 40 unid. Hembra Hembra	2	S/. 5.50	S/. 11.00
3.5.	Cable Dupont 40 unid. MACHO Hembra	2	S/. 5.50	S/. 11.00
3.6.	Cable Dupont 40 unid. MACHO MACHO	2	S/. 5.50	S/. 11.00
3.7.	Cable electrico #18 AWG rollo x 100m	6	S/. 60.00	S/. 360.00
3.8.	Global (materiales de pvc, pegamento,etc.)	1	S/. 800.00	S/. 800.00
3.9.	ARDUINO MEGA 2560 R3	2	S/. 70.00	S/. 140.00
3.12.	Transformador de 220 VAC a 12 VDC – 1.5A	5	S/. 52.00	S/. 260.00
	TOTAL			S/. 3,188.00

ANEXO 8: PANEL FOTOGRÁFICO



Figura 60: Ensayo para el sensor de flujo de agua de 1"



Figura 61: Materiales para la instalación del sensor de ultrasonido



Figura 62: Calibración del sensor de ultrasonido



Figura 63: Delimitación de la zanja para los conductores eléctricos.



Figura 64: Excavación de zanjas para los conductores eléctricos.



Figura 65: Manguera de polietileno de 32 mm utilizada para conducir los cables eléctricos.



Figura 66: Instalación de la manguera de polietileno para conducir los cables eléctricos.



Figura 67: Instalación de la red eléctrica.

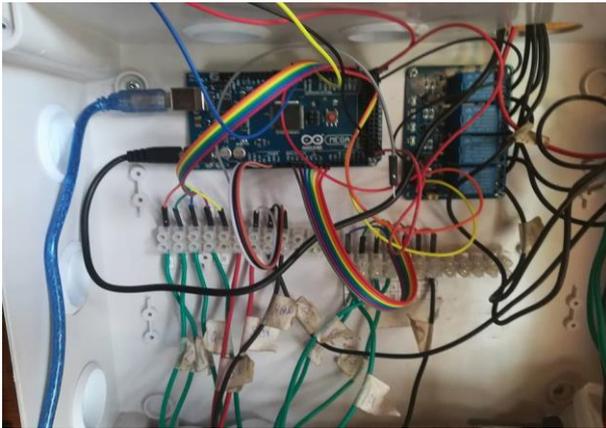


Figura 69: Instalación del tablero donde se alberga a los componentes electrónicos.



Figura 68: Instalación del tablero donde se alberga a los transformadores de 12V para las válvulas solenoides.



Figura 70: Instalación del arco de riego automatizado



Figura 71: Arco de riego manual y automatizado.



Figura 72: Instalación de los sensores de presión en las subunidades



Figura 73: Instalación del sensor de humedad GS1 a 10 cm de la superficie.



Figura 75: Toma de muestra de suelo para la calibración del sensor de humedad.



Figura 74: Instalación del cableado eléctrico para la automatización del tablero de la bomba.



Figura 76: Arco de riego de la subunidad 2 en funcionamiento.



Figura 77: Arco de riego de la subunidad 2 en funcionamiento.



Figura 78: Sensor de flujo de agua de 1" instalado en la tubería principal



Figura 79: Rejilla de PVC que protege al sensor de flujo de agua de 1"



Figura 80: Caja pase de 100x100mm que alberga al sensor de ultrasonido

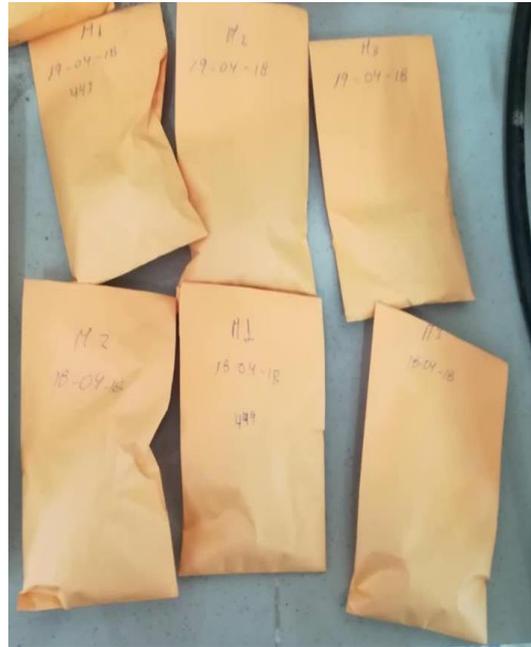


Figura 81: Muestras de suelo para la calibración del sensor de humedad GS1

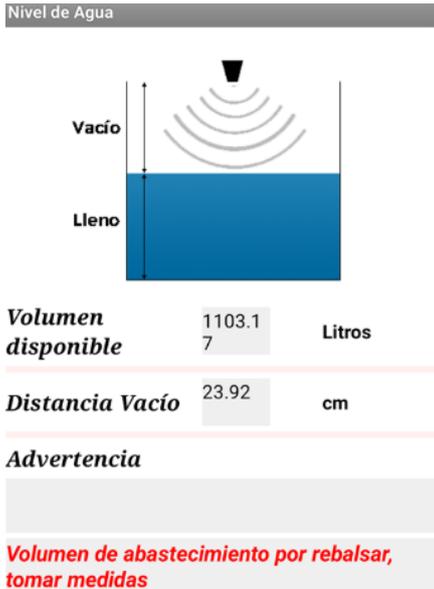


Figura 82: Alerta en el interfaz de nivel de agua



Figura 83: Registro de volumen en la válvula volumétrica



Figura 84: Sensor de caudal FS400 de 1" utilizado en el proyecto



Figura 85: Sensor de caudal YF-S 201 de 1/2" utilizado en el proyecto

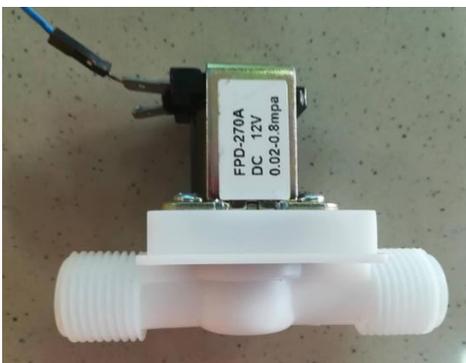


Figura 86: Válvula solenoide de 1/2" utilizado en el proyecto



Figura 87: Arduino Mega 2560 utilizado en el proyecto



Figura 88: Sensor de ultrasonido HC-SR04 utilizado en el proyecto



Figura 89: Transductor de presión HK3022 utilizado en el proyecto



Figura 90: Subunidades automatizadas