

**UNIVERSIDAD NACIONAL AGRARIA**

**LA MOLINA**

**FACULTAD DE INGENIERÍA AGRÍCOLA**



**“DISEÑO DE UN PROTOTIPO DE RIEGO AUTOMATIZADO PARA  
PEQUEÑOS HUERTOS AGROECOLÓGICOS UTILIZANDO  
ARDUINO EN ZONAS DESÉRTICAS URBANAS DE LIMA”**

**TESIS PARA OPTAR TÍTULO DE  
INGENIERO AGRÍCOLA**

**LUIS MIGUEL HUAMÁN GUTIÉRREZ**

**LIMA – PERÚ**

**2023**

# DISEÑO DE UN PROTOTIPO DE RIEGO AUTOMATIZADO PARA PEQUEÑOS HUERTOS AGROECOLÓGICOS UTILIZANDO ARDUINO EN ZONAS DESÉRTICAS URBANAS DE LIMA

## INFORME DE ORIGINALIDAD

12%

INDICE DE SIMILITUD

12%

FUENTES DE INTERNET

3%

PUBLICACIONES

5%

TRABAJOS DEL ESTUDIANTE

## FUENTES PRIMARIAS

1	<a href="http://repositorio.lamolina.edu.pe">repositorio.lamolina.edu.pe</a> Fuente de Internet	1%
2	<a href="http://hdl.handle.net">hdl.handle.net</a> Fuente de Internet	1%
3	<a href="http://www.faecta.coop">www.faecta.coop</a> Fuente de Internet	1%
4	<a href="http://repositorio.utn.edu.ec">repositorio.utn.edu.ec</a> Fuente de Internet	<1%
5	<a href="http://aprenderly.com">aprenderly.com</a> Fuente de Internet	<1%
6	Submitted to Universidad San Ignacio de Loyola Trabajo del estudiante	<1%
7	<a href="http://sistemamid.com">sistemamid.com</a> Fuente de Internet	<1%
8	<a href="http://www.ugt.es">www.ugt.es</a> Fuente de Internet	<1%

**UNIVERSIDAD NACIONAL AGRARIA LA MOLINA**

**FACULTAD DE INGENIERIA AGRICOLA**

**“DISEÑO DE UN PROTOTIPO DE RIEGO AUTOMATIZADO PARA  
PEQUEÑOS HUERTOS AGROECOLÓGICOS UTILIZANDO  
ARDUINO EN ZONAS DESÉRTICAS URBANAS DE LIMA”**

TESIS PARA OPTAR EL TITULO DE:

**INGENIERO AGRÍCOLA**

Presentado por:

**BACH. LUIS MIGUEL HUAMÁN GUTIÉRREZ**

Sustentado y aprobado por el siguiente jurado:

Dr. DAVID RICARDO ASCENCIOS TEMPLO

Presidente

Mg. Sc. JUVENAL VIVIANO GARCÍA ARMAS

Asesor

Ing. SARAY SIURA CÉSPEDES

Miembro

Mg. Sc. LUIS RAMÓN RÁZURI RAMÍREZ

Miembro

Dr. GONZALO RAMCÉS FANO MIRANDA

Co-Asesor

LIMA – PERU

2023

## **DEDICATORIA**

Este trabajo está dedicado a mi amiga y madre la señora Carmina Gutiérrez Prieto quien siempre estuvo apoyándome incondicionalmente tanto de manera emocional como económica. Gracias a sus consejos y alientos pude salir adelante y superarme a lo largo de mi etapa académica, la verdad me cuesta trabajo pensar qué hubiera sido de mí sin mi madre.

## **AGRADECIMIENTOS**

Un agradecimiento especial a la familia Silverio Huamán y esposa por su desinteresado apoyo en la ejecución de la presente tesis brindándome todas las facilidades posibles.

Un agradecimiento a la señorita Carmen Rosa Gamboa quien fue una gran compañera y ayudante en la instalación y puesta en práctica del trabajo de investigación.

También agradecerles a los ingenieros Fano Miranda y Juvenal García quienes se tomaron el tiempo y la molestia de guiarme, darme sus consejos y pautas para sacar adelante este trabajo.

Y en general agradecer a todas las personas entre amigos y colegas que me brindaron alguna forma de apoyo.

## ÍNDICE GENERAL

I.	INTRODUCCIÓN.....	1
1.1.	OBJETIVO.....	2
1.1.1.	Objetivo general .....	2
1.1.2.	Objetivos específicos.....	2
II.	REVISIÓN DE LA LITERATURA.....	3
2.1.	Agricultura familiar.....	3
2.2.	Agroecología.....	3
2.3.	Hortalizas .....	4
2.3.1.	Tolerancia del pH y la salinidad en hortalizas .....	5
2.4.	Sistema tecnificado riego por goteo .....	6
2.4.1.	Componentes del sistema de riego.....	6
2.5.	Diseño agronómico.....	8
2.6.	Diseño hidráulico .....	10
2.7.	Automatización del sistema de riego.....	10
2.7.1.	Arduino.....	11
2.7.2.	Sensores.....	13
2.7.3.	Actuadores.....	14
2.7.4.	Almacenamiento de datos.....	15
2.7.5.	Fuentes de energía .....	15
2.7.6.	Programación.....	15
III.	METODOLOGÍA .....	17
3.1.	Materiales y equipos.....	17
3.2.	Descripción del área experimental de estudio .....	17
3.3.	Metodología aplicada .....	18
3.3.1.	Recopilación de información.....	18
3.3.2.	Diseño agronómico.....	27
3.3.3.	Diseño hidráulico.....	29
3.3.4.	Automatización del sistema.....	32
3.3.5.	Calibración de sensores .....	35
3.3.6.	Programación en el microcontrolador.....	36
3.3.7.	Aplicativo ArduinoDroid.....	39
3.3.8.	Recopilación y almacenamiento de datos .....	39

3.3.9.	Encendido manual .....	40
3.4.	Fase de campo .....	40
3.4.1.	Instalación del sistema de riego .....	40
3.4.2.	Conexión e instalación electrónica y eléctrica del sistema automatizado.....	40
3.4.3.	Prueba de coeficiente de uniformidad (Cu) .....	44
3.4.4.	Rotación de cultivos .....	45
3.4.5.	Operatividad y manejo del sistema de riego .....	48
3.4.6.	Presupuesto, costo unitario y metrado de la instalación .....	48
IV.	RESULTADOS Y DISCUSIÓN .....	50
4.1.	Diseño del sistema de riego por goteo.....	50
4.1.1.	Diseño agronómico.....	50
4.1.2.	Diseño hidráulico.....	54
4.1.3.	Fuente de agua.....	58
4.1.4.	Caseta de control de riego.....	59
4.1.5.	Tuberías .....	61
4.1.6.	Arcos de riego.....	61
4.1.7.	Porta goteros .....	61
4.2.	Instalación electrónica del riego automatizado.....	62
4.2.1.	Sensores de humedad de suelo .....	62
4.2.2.	Sensor de flujo de agua.....	63
4.2.3.	Sensor de ultra sonido.....	63
4.2.4.	Válvulas solenoides .....	64
4.2.5.	Bomba de agua .....	65
4.2.6.	Módulo de lector de micro SD.....	66
4.3.	Instalación eléctrica del riego automatizado.....	68
4.3.1.	Cableado .....	68
4.3.2.	Fuente de energía.....	68
4.4.	Control manual de sistema de riego .....	69
4.4.1.	Control de bomba de agua .....	69
4.4.2.	Control de válvulas solenoides .....	69
4.5.	Calibración de sensores .....	70
4.5.1.	Sensor de humedad de suelo.....	70
4.5.2.	Sensor de flujo.....	72
4.6.	Programación .....	73
4.7.	Pruebas de funcionamiento.....	78
4.7.1.	Software Arduino .....	78
4.7.2.	Sensores de humedad de suelo y flujo de agua .....	79
4.7.3.	Aplicativo Arduinodroid.....	81
4.7.4.	Coefficiente de uniformidad .....	81

4.8.	Operatividad .....	82
4.8.1.	Funcionamiento manual .....	83
4.8.2.	Funcionamiento automatizado .....	83
4.9.	Presupuesto.....	85
V.	CONCLUSIONES.....	89
VI.	RECOMENDACIONES .....	90
VII.	BIBLIOGRAFÍA.....	91
VIII.	ANEXOS.....	96

## ÍNDICE DE TABLAS

Tabla 1:	Diez elementos principales de la agroecología.....	4
Tabla 2:	Niveles de tolerancia de pH en las hortalizas .....	5
Tabla 3:	Rendimiento de los cultivos frente a la salinidad (CEe).....	7
Tabla 4:	Análisis de la calidad de agua .....	18
Tabla 5:	Análisis textural del suelo .....	19
Tabla 6:	Promedio mensual de evapotranspiración potencial de los últimos 10 años .....	20
Tabla 7:	Tolerancia a la Salinidad (CEe) y al Boro .....	21
Tabla 8:	Criterios de selección utilizados en los grupos de cultivos .....	24
Tabla 9:	Características agronómicas de los cultivos seleccionados para el proyecto.....	26
Tabla 10:	Tipos de variables utilizadas en programación.....	39
Tabla 11:	Conexión humedad con puerto Arduino.....	41
Tabla 12:	Conexión Flujo con puerto Arduino.....	41
Tabla 13:	Conexión ultrasonido con puerto Arduino .....	41
Tabla 14:	Conexión relay-válvula con puerto Arduino .....	42
Tabla 15:	Conexión relay-bomba con puerto Arduino .....	42
Tabla 16:	Conexión de pines micro SD con puerto Arduino.....	43
Tabla 17:	Clasificación por porcentaje de CU.....	45
Tabla 18:	Coefficiente de cultivo de cultivos propuestos .....	50
Tabla 19:	Eficiencia de riego según textura y profundidad .....	52
Tabla 20:	Coefficiente de uniformidad según separación, pendiente y clima.....	52
Tabla 21:	Resumen de cálculos hidráulicos según necesidades (por tramo) .....	55
Tabla 22:	Puertos utilizados en la placa Arduino Leonardo .....	67
Tabla 23:	Mediciones con sensor de humedad n°1 .....	70
Tabla 24:	Mediciones con sensor de humedad n°2.....	71
Tabla 25:	Mediciones con sensor de humedad n°3.....	71
Tabla 26:	Mediciones con sensor de humedad n°4.....	71
Tabla 27:	Mediciones con sensor de flujo n°1 .....	72
Tabla 28:	Mediciones con sensor de flujo n°2.....	72
Tabla 29:	Mediciones con sensor de flujo n°3.....	73
Tabla 30:	Mediciones con sensor de flujo n°4.....	73
Tabla 31:	Volumen y caudal por Líneas de goteros y posición de los goteros .....	82
Tabla 32:	Materiales definitivos utilizados en el proyecto de tesis .....	85

## ÍNDICE DE FIGURAS

Figura 1: Interfaz IDE de Arduino.....	11
Figura 2: Sensores compatibles con Arduino.....	14
Figura 3: Actuadores compatibles con Arduino.....	14
Figura 4: Interfaz de ArduinoDroid .....	16
Figura 5: Ubicación del área experimental de trabajo.....	17
Figura 6: Calendario de siembra y cosecha para un ciclo de vegetación .....	23
Figura 7: Distribución de los cultivos seleccionados en el área de estudio.....	25
Figura 8: Esquema general de la automatización.....	32
Figura 9: Resumen de sensores y actuadores controlados por Arduino .....	34
Figura 10: Procedimiento de escritura y lectura del código.....	36
Figura 11: Localización de goteros para prueba de uniformidad .....	44
Figura 12: Calendario anual de cultivo .....	46
Figura 13: Rotación de los cultivos por época de siembra entre los grupos .....	47
Figura 14: Grupos de rotación de los cultivos propuestos .....	49
Figura 15: Curva de descarga .....	53
Figura 16: Distribución de los implementos de la automatización y riego .....	56
Figura 17: Accesorios y partes de la fuente de agua.....	59
Figura 18: Caseta de control .....	60
Figura 19: Partes del arco de riego .....	61
Figura 20: Porta goteros auto compensantes de 2L/h .....	62
Figura 21: Conexiones de los sensores de humedad con el Arduino Leonardo .....	62
Figura 22: Conexiones de los sensores de flujo con el Arduino Leonardo .....	63
Figura 23: Conexiones del sensor de ultrasonido con el Arduino Leonardo .....	64
Figura 24: Conexión del relay y válvulas con el Arduino Leonardo .....	65
Figura 25: Conexión del transistor mosfet y bomba con el Arduino Leonardo .....	66
Figura 26: Conexión del lector micro SD con el Arduino Leonardo .....	66
Figura 27: Diagrama de la instalación eléctrica.....	68
Figura 28: Conexión eléctrica de bomba-interruptor .....	69
Figura 29: Conexión válvula solenoide-interruptor.....	70
Figura 30: Código de la programación en software Arduino.....	76
Figura 31: Procedimiento de la programación del código en la placa Arduino Leonardo .....	77
Figura 32: Lectura monitor serial área 1-estado seco-encendida .....	78
Figura 33: Humedad de suelo en el estrato de saturación .....	79

Figura 34: Caudal en litros por minuto .....	80
Figura 35: Litros consumidos .....	80
Figura 36: Encendido utilizando el aplicativo Android del celular.....	81
Figura 37: Caudales de la prueba de CU.....	82
Figura 38: Esquema secuencial de la instalación manual .....	83
Figura 39: Esquema secuencial de la instalación automatizada.....	84

## ÍNDICE DE ANEXOS

Anexo 1: Caracterización física del suelo .....	96
Anexo 2: Resumen de las propiedades físicas del suelo .....	96
Anexo 3: Determinación de la textura al tacto .....	97
Anexo 4: Caracterización mediante la tabla Munsell.....	97
Anexo 5: Análisis de agua -Rutina.....	98
Anexo 6: Diagrama de clasificación para las aguas de riego .....	99
Anexo 7: Límites tolerantes del boro el agua de riego.....	99
Anexo 8: Coeficiente de fricción en función del material de la tubería .....	100
Anexo 9: Coeficiente de Christiansen según el número de salidas .....	100
Anexo 10: Especificaciones técnicas placa Arduino Leonardo .....	100
Anexo 11: Especificaciones técnicas de sensor capacitivo .....	101
Anexo 12: Especificaciones técnicas de sensor de flujo .....	101
Anexo 13: Especificaciones técnicas del sensor ultrasónico.....	102
Anexo 14: Especificaciones técnicas de modulo relay.....	102
Anexo 15: Especificaciones técnicas Driver Mosfet IRF520N.....	102
Anexo 16: Especificaciones técnicas de válvula solenoide.....	103
Anexo 17: Especificaciones técnicas Bomba de agua.....	103
Anexo 18: Especificaciones técnicas de Micro SD.....	104
Anexo 19: Especificaciones del Aplicativo ArduinoDroid .....	104
Anexo 20: Código de lectura de módulo micro SD .....	104
Anexo 21: Código de lectura de electroválvula-válvula solenoide .....	105
Anexo 22: Código de lectura para la bomba de agua.....	105
Anexo 23: Código para la calibración de sensor de humedad de suelo .....	106
Anexo 24: Código para la lectura del sensor de humedad de suelo .....	106
Anexo 25: Código para la calibración del sensor de flujo.....	107
Anexo 26: Código para la lectura del sensor de flujo .....	108
Anexo 27: Código para la lectura del sensor de ultrasonido .....	109
Anexo 28: Código ejecutado con el aplicativo Arduinodroid en el prototipo riego automatizado .....	109
Anexo 29: Código ejecutado con laptop (monitor serial) en el prototipo riego automatizado .....	116
Anexo 30: Conexión general de los actuadores y sensores con el microcontrolador Arduino Leonardo ....	124
Anexo 31: Resumen técnico de las placas Arduino más usadas .....	125
Anexo 32: Registro de datos almacenados en la tarjeta del módulo micro SD .....	126
Anexo 33: Compilación (verificación del código en celular) y cargada a placa Arduino Leonardo.....	126

Anexo 34: Alternativas de uso del sistema de riego automatizado con los distintos modelos de Arduino ..	127
Anexo 35: Área de ejecución del proyecto de tesis.....	128
Anexo 36: Instalación de las tuberías principales y secundarias.....	129
Anexo 37: Instalación de arcos de riego .....	129
Anexo 38: Instalación de porta laterales y bigotes de riego .....	130
Anexo 39: Pruebas de funcionamiento y calibración.....	131
Anexo 40: Fertilizantes utilizados en la agricultura urbana .....	132
Anexo 41: Alternativas de solución frente a suelos salinos .....	133
Anexo 42: Fuente de materia orgánica.....	133

## GLOSARIO

AC: Son las siglas en inglés de Alternating Current que en español significa Corriente Alterna y es un tipo de flujo de corriente que circula en dos sentidos de un polo y alterna en el otro polo.

bit: Es la unidad de medida de la velocidad de transmisión de datos, si se hace referencia a la memoria, un bit viene a ser una celda de la memoria de un microcontrolador donde se puede albergar datos.

byte: Viene a ser 8 veces un bit y en Arduino vendría ser 8 bits, con esto se quiere decir que con un byte se tiene 8 celdas en la memoria que pueden albergar datos.

CEe: Siglas de la conductividad eléctrica en el extracto de saturación del suelo.

CEw: Siglas de la conductividad eléctrica del agua de riego.

DC: Son las siglas en inglés de Direct Current que en español significa Corriente Directa y es un tipo de flujo de corriente que circula en un solo sentido de un polo a otro polo.

FAO: Organización de las naciones unidas para la alimentación y la agricultura o más conocido como Food and Agriculture Organization.

GND: Es la abreviación de la palabra en inglés (EE.UU.) Ground que significa tierra y es tomado como un punto de referencia que significa 0V.

HARDWARE: Son las partes en físico de un equipo, por ejemplo: un celular, una placa Arduino, computadora.

ICSP: Son las iniciales de la palabra In-Circuit Serial Programming que traducido significa Programación serie en circuito y sirve para poder programar un dispositivo (microcontrolador) mientras están instalados en un sistema completo.

IDE: Entorno de desarrollo integrado.

MEMORIA EEPROM: EEPROM de sus siglas en inglés Electrically Erasable Programmable Read Only Memory que significa memoria de solo lectura programable y borrable

eléctricamente, cuya función es permitir el almacenamiento de datos para poder recuperar en caso de pérdidas de alimentación en nuestro dispositivo.

**MEMORIA FLASH:** Es la memoria propia del programa, es de tipo no volátil o sea si se pierde la alimentación los datos permanecerán en ésta memoria. Es aquí donde se guarda el sketch que se cargan y tiene una vida útil de unos 100000 ciclos de escritura.

**MEMORIA SRAM:** De sus siglas en inglés Static Random Access Memory que significa Memoria estática de acceso aleatorio. Sirve de espacio para que los programas almacenen y manipulen variables que se ejecutarán, cabe mencionar que ésta memoria es eliminada cuando el Arduino pierde su alimentación.

**MICROCONTROLADOR:** Es un circuito integrado, que funciona como automatizador y procesador de información, prácticamente es una computadora (mini PC) pues incluye elementos de salida y entrada para el control.

**PWM:** Son las siglas en inglés de Pulse Width Modulation que significa modulación por ancho de pulso. Es utilizada para transmitir una señal ya sea analógica o digital o también se puede utilizar para controlar la cantidad de energía que se envía a una carga.

**RESET:** Traducida vendría a ser Reinicio, es utilizado en Arduino para restablecer al inicio la placa. Ésta se puede ejecutar tanto desde el Software como el hardware Arduino.

**VCC:** Es el símbolo de la entrada de alimentación de 5V que hay en las placas Arduino.

**VIN:** Es el símbolo de la alimentación externa para el Arduino, en un rango de 12 a 6 v de corriente directa. También se puede utilizar como una fuente de salida de voltaje.

**SOFTWARE:** Son los conjuntos de programas o aplicaciones, comandos y reglas de informática que sumado hacen que el equipo(hardware) funcione. Ejemplo: sistema Android, IDE Arduino, sistema Windows.

## **RESUMEN**

Éste trabajo busca la complementariedad entre tecnología y agricultura, para lograr mejora en la distribución del agua, dosificación según necesidades de los cultivos, selección de materiales idóneos, integración de tecnología barata y sencilla en cuanto a hardware, códigos y ejecución y ser atractiva para su masificación. Para esto se realizó un prototipo automatizado de sistema de riego para huertos aplicando principios agroecológicos para su selección, diversificación, distribución y cuidado; realizando estudios de clima, suelo y agua obteniendo los potenciales cultivos. Luego se realizó el diseño agronómico e hidráulico hallando las necesidades hídricas del cultivo y el dimensionamiento de materiales a instalar. Después se planteó su automatización mediante un microcontrolador Arduino conectándolo con sensores y actuadores, controlando parámetros de humedad de suelo, flujo de agua, niveles de agua, aperturas y cierres de válvulas y bomba. En cuanto a los datos arrojados por los sensores de suelo y flujo, fueron validados a través de su calibración. Para la conexión entre el microcontrolador, los sensores y actuadores se desarrolló la programación del código en el IDE del software Arduino, donde se añadieron librerías, funciones y variables para su automatización. Iniciando la programación, tomando lecturas del sensor de humedad de suelo para las cuatro áreas, donde si el porcentaje de lectura es menor al 50% se abrirá la electroválvula del área leída, dejando pasar el flujo de agua y si es mayor o igual al 50% se cerrará la electroválvula; para el caso que todas las áreas tengan una lectura del sensor de humedad mayor al 50% éste emitirá una señal al Arduino que apagará la bomba y cerrará las electroválvulas. Por último, se habilitó los encendidos, automatizado con App ArduinoDroid y manual del sistema de riego, para el caso de fallo en la placa Arduino o dificultad en el uso de la automatización.

Palabras claves: Automatización, Arduino Leonardo, ArduinoDroid, riego tecnificado, hortalizas.

## **ABSTRACT**

This work seeks the complementarity between technology and agriculture, to achieve improvement in the distribution of water, dosage according to the needs of the crops, selection of suitable materials, integration of cheap and simple technology in terms of hardware, codes and execution and to be attractive for its use. massification. For this, an automated prototype of an irrigation system for orchards was made applying agroecological principles for their selection, diversification, distribution and care; conducting studies of climate, soil and water obtaining potential crops. Then the agronomic and hydraulic design was carried out, finding the water needs of the crop and the sizing of materials to be installed. Afterwards, its automation was proposed by means of an Arduino microcontroller connecting it with sensors and actuators, controlling parameters of soil humidity, water flow, water levels, opening and closing of valves and pump. Regarding the data provided by the soil and flow sensors, they were validated through their calibration. For the connection between the microcontroller, the sensors and actuators, the programming of the code was developed in the IDE of the Arduino software, where libraries, functions and variables were added for its automation. Starting the programming, taking readings from the soil moisture sensor for the four areas, where if the reading percentage is less than 50% the solenoid valve of the area read will open, allowing the flow of water to pass and if it is greater than or equal to 50 % the solenoid valve will close; In the event that all areas have a humidity sensor reading greater than 50%, it will issue a signal to the Arduino that will turn off the pump and close the solenoid valves. Finally, the ignitions were enabled, automated with the ArduinoDroid App and manual of the irrigation system, in case of failure in the Arduino board or difficulty in using the automation.

**Keywords:** Automation, Arduino Leonardo, ArduinoDroid, technical irrigation, vegetables.

## I. INTRODUCCIÓN

Uno de los problemas que aquejan a las zonas periféricas a la ciudad capital, como en el caso del AA.HH. Palmeras de Huarangal, es la escasez de alimentos ricos en vitaminas, así también la escasa disponibilidad de agua, haciendo que los pobladores se vean afectados directa o indirectamente con enfermedades ligadas a la falta de vitaminas y nutrientes.

Sin embargo, la ventaja que tienen estos habitantes, es que disponen de espacios con áreas entre 40 y hasta 500 m<sup>2</sup> los que se adecuarían para formar huertos agroecológicos para el cultivo de hortalizas (zanahoria, vainita, apio, lechuga, rábano, tomate, acelga, ají, betarraga, espinaca, pallar, nabo y pimiento, etc.), dado que estos “son cultivos que necesitan poco espacio para crecer, tienen breves ciclos vegetativos y pueden suministrar vitaminas y nutrientes que no se obtiene fácilmente de otras fuentes alimentarias, por lo tanto contribuirían a evitar las carencias de micronutrientes”(FAO, 1996). Esta situación asociada a un manejo óptimo de la escasa disponibilidad de agua, sería una solución para superar estas carencias.

En tal sentido en este trabajo se pretende fomentar una agricultura agroecológica de intercambio vecinal, que aproveche el manejo de los escasos recursos de agua y suelo, mediante un sistema de riego tecnificado, haciendo uso de software y hardware libre de bajo costo y de fácil manejo.

Por lo que se plantea un diseño agroecológico e hidráulico automatizado, utilizando un microcontrolador Arduino con dispositivos electrónicos asociados a él (sensores y actuadores), que ayudarán a una distribución homogénea y oportuna de agua en cada una de las subunidades.

## **1.1. OBJETIVO**

### **1.1.1. Objetivo general**

Diseño de un prototipo de riego automatizado para pequeños huertos agroecológicos utilizando Arduino en zonas desérticas urbanas de Lima.

### **1.1.2. Objetivos específicos**

- Diseño agronómico para huertos agroecológicos.
- Diseño hidráulico del sistema de riego.
- Diseño del sistema de automatización para el control del riego.

## **II. REVISIÓN DE LA LITERATURA**

### **2.1. Agricultura familiar**

La agricultura familiar abarca diversos conceptos a lo largo del mundo, a pesar de que analizando cada una de éstas, se puede ver muchas diferencias las cuales dificultan tener un eje comparativo entre ellas. Sin embargo, se tiene tres destacadas similitudes en dichos conceptos: a) la mano de obra para la explotación es predominantemente familiar (también se puede contratar por ausencia de familiares o por ampliación de la operación de explotación); b) administración de la economía y producción por parte del jefe(a) de la explotación (qué hacer con los activos y destinación de mano de obra); c) el tamaño de explotación y/o producción (referidas a fincas con áreas pequeñas (Maletta,2011)) (FAO, 2014).

Por lo consiguiente como definición de agricultura familiar, la FAO utiliza los principios anteriormente descritos y nos dice “la Agricultura familiar (incluye todas las actividades agrícolas basadas en la familia) es una forma de organizar la agricultura, ganadería, silvicultura, pesca, acuicultura y pastoreo, que administrada y operada por una familia y sobre todo que depende preponderadamente del trabajo familiar, tanto de mujeres como hombres. La familia y la granja están vinculadas, co-evolucionan y combinan funciones económicas ambientales, sociales y culturales.”(FAO, 2014).

### **2.2. Agroecología**

Es una ciencia dedicada al estudio de la estructura y función de los agroecosistemas en cuanto a su relación ecológica y cultural, también tiene enfoques políticos los cuales son considerados como instrumentos de cambio social (dejan de ser un fin), con lo cual se convierte en una estrategia para alcanzar una transformación rural y urbana, logrando que los pueblos creen relaciones sociales y comerciales justas y equitativas.(Comunidad Andina, 2011).

También puede expresarse como una “disciplina científica que reúne, sintetiza y aplica conocimientos de la agronomía, la ecología, la sociología, la etnobotánica y otras ciencias afines, con una óptica holística y sistémica y un fuerte componente ético, para generar conocimientos, validar y aplicar estrategias adecuadas para diseñar, manejar y evaluar agroecosistemas sustentables (Sardón, 2002b)” (Sarandón & Flores, 2014).

La FAO en su afán de establecer y brindar pautas para que transformen sus sistemas agrícolas, alimentarios e integren la agricultura sostenible a gran escala y así poder lograr el reto del hambre cero, estableció 10 elementos fundamentales de la agroecología(FAO, 2018).

**Tabla 1: Diez elementos principales de la agroecología**

Elemento	Descripción
Diversidad	Fundamental en la transición agroecológica para garantizar la seguridad alimentaria y la nutrición y al mismo tiempo conservar, proteger y mejorar los recursos naturales
Creación e intercambio de conocimientos	Las innovaciones agrícolas responden mejor a los desafíos locales cuando se crean conjuntamente mediante procesos participativos.
Sinergias	Crear sinergias(cooperación) potencia las principales funciones de los sistemas alimentarios, lo que favorece la producción y múltiples servicios ecosistémicos.
Eficiencia	Las prácticas agroecológicas innovadoras producen más utilizando menos recursos externos.
Reciclaje	Reciclar más significa una producción agrícola con menos costos económicos y ambientales.
Resiliencia	Mejorar la resiliencia(adaptabilidad) de las personas, comunidades y los ecosistemas es fundamental para lograr sistemas alimentarios y agrícolas sostenibles.
Valores humanos y sociales	Proteger y mejorar los medios de vida, la equidad y el bienestar social es fundamental para lograr sistemas alimentarios y agrícolas sostenibles.
Cultura y tradiciones humanas	Mediante el apoyo a unas dietas saludables, diversificadas y culturalmente apropiadas, la agroecología contribuye a la seguridad alimentaria y la nutrición al tiempo que mantiene la salud de los ecosistemas.
Gobernanza responsable	Para lograr una alimentación y una agricultura sostenible es necesario adoptar mecanismos de gobernanza responsables y eficaces a diferentes escalas, de la local a la nacional y mundial.
Economía circular y solidaria	Que reconectan a los productores y consumidores ofreciendo soluciones innovadoras para vivir dentro de los límites de nuestro planeta y, al mismo tiempo, afianzan las bases sociales para el desarrollo inclusivo y sostenible.

FUENTE:(FAO, 2018)

### 2.3. Hortalizas

Las hortalizas son plantas herbáceas consumidas principalmente por los humanos y son cultivadas y cosechadas en estado silvestre, en bruto o en una forma mínimamente

procesada. Cabe mencionar que en ésta referencia las raíces y tubérculos como las patatas, batatas; las leguminosas (legumbres); los cereales, el maíz; los frutos secos, los cocos, las nueces, las semillas de girasol; té, café, cacao y los productos procesados y ultra procesados hechos de vegetales no son considerados dentro de ésta definición(FAO & CIRAD, 2021).

Por otro lado se resalta que las hortalizas son cultivos de ciclo de vida anual o bianual, excepcionalmente perennes, periodos de cultivos variados (40 días maduración/espinaca); tamaño de vegetación en estado de madurez diversos (betarraga 10 cm y tomate 100 cm de diámetro respectivamente) y altos rendimientos en policultivos y asociaciones entre cultivos de distintas variedades(Ugás et al., 2000).

### 2.3.1. Tolerancia del pH y la salinidad en hortalizas

Si bien es cierto que estos cultivos son flexibles en su adaptación respecto del suelo, cabe mencionar que se encontrará algunos cultivos que tienen una mejor respuesta a un determinado pH como se menciona en la Tabla 2, ya sea un rango ácido o básico, lo cual incrementa su rendimiento(Cerna, 2011).

**Tabla 2: Niveles de tolerancia de pH en las hortalizas**

Tolerancia		
Ligera	Moderada	Extrema
(pH 6.8 - 6.0)	(pH 6.8 – 5.5)	(pH 6.8 – 5.0)
acelga, apio berro, beterraga, brócoli, cebolla, col, col china, coliflor, espárrago, espinacas, lechuga, melón, okra, poro, soya	ajo, arveja, berenjena, col de Bruselas, maíz, choclo, mostaza, nabo, pallar, pepinillo, perejil, pimiento, rabanito, tomate, vainita, zanahoria, zapallito	achicoria, camote, diente de león, hinojo, papa, ruibarbo, sandía

FUENTE: (Ugás et al., 2000)

Sin embargo, no solo el pH está relacionado con el rendimiento de algunos cultivos hortícolas “según Becerra de la Flor (1970) tiene una mayor influencia la salinidad presente en los suelos, logrando disminuciones en el porcentaje de producción de hasta 50% de lo normal”(Cerna, 2011).

Además, si la tolerancia a la salinidad está contemplada como una capacidad que tienen los cultivos para soportar la sal presente en los suelos sin mantener efectos perjudiciales en su desarrollo y/o producción. Esto no implica que no sufran algunos problemas, pues la

disponibilidad de agua se verá disminuida por la presencia de sales con la cual se reflejará en el rendimiento y producción agrícola(Vásquez et al., 2017) (Tabla 3).

Cabe precisar que los problemas por la salinidad afectan al aspecto fisiológico y al metabolismo de la planta, ocasionando un desequilibrio iónico y estrés osmótico. Con lo cual la planta para recuperar el equilibrio perdido por la presencia salina afecta su actividad en: a) control de crecimiento, el cual debe reanudarse, pero a una tasa menor; b) control de la homeostasis iónica y osmótica; c) prevenir o reparar el daño o detoxificación(Alcaraz, 2012).

#### **2.4. Sistema tecnificado riego por goteo**

Éste tipo de sistema se basa en una frecuente dosificación de agua en pequeñas cantidades, “utilizando para esto tuberías a presión y emisores, esto se da cercano a la planta y puede ser superficial o subterráneo”(Fernández et al., 2010).

Por otro lado, éste sistema es muy adecuado para “la incorporación simultánea de agua con fertilizantes además de poder adicionar productos para el control de plagas y enfermedades” (Siura, 2007).

Cabe mencionar que “éste sistema, tiene un elevado costo en su instalación, pero esto dependerá del cultivo, la calidad de agua, su exigencia del filtrado, la instalación de la automatización, etc.”(Fernández et al., 2010). Pero esto es compensado con un mejor manejo en su automatización, menor costo en la mano de obra, menor consumo de agua y disponibilidad de control en el flujo de agua.

##### **2.4.1. Componentes del sistema de riego**

La mayoría de sistemas están compuestos por diversos componentes o simplemente pueden tener lo básico, cuyas “partes son: pozo o fuente de agua, bomba de agua, válvulas de paso y de no retorno, un filtro de malla, tubería principal, válvula de aire, válvula de lavado, líneas secundarias, laterales de riego, goteros y nudos finales”(Briceño et al., 2012).

**Tabla 3: Rendimiento de los cultivos frente a la salinidad (CEe)**

Cultivo	Salinidad máxima sin disminución del rendimiento ds/m <sup>2</sup>	Disminución en % del rendimiento (% por ds/m)
Cultivos sensibles (dS/m)		
Vainita	1	19
Zanahoria	1	14
Fresa	1	33
Cebolla	1.2	16
otros: berenjena, arveja		
Cultivos moderadamente sensibles		
Nabo	0.9	9
Rabanito	1.2	13
Lechuga	1.3	13
Pimiento	1.5	14
Camote	1.5	11
Haba	1.6	10
Maíz	1.7	12
Papa	1.7	12
Col	1.8	10
Apio	1.8	6
Espinaca	2	8
Pepinillo	2.5	13
Tomate	2.5	10
Brócoli	2.8	9
Zapallo	3.2	16
Otros: alcachofa, ajo, coliflor, melón, ají		
Cultivos moderadamente tolerantes		
Beterraga	4	9
Zapallito	4.7	9
Otros: espárrago		
Cultivos tolerantes		
algodón	7.7	5
cebada	8	5

Nota:  $1 \text{ dS/m}^2 = 1 \text{ mmho/cm}$

FUENTE: (Ugás et al., 2000)

## **2.5. Diseño agronómico**

Una de las partes más importantes de los sistemas de riego es el diseño agronómico, en ésta se determinará la cantidad de agua que se necesitará para satisfacer la demanda hídrica del cultivo.

Para Fernández “lo que realmente interesa conocer es la demanda de agua para el periodo en el cual el cultivo requiera la mayor cantidad de agua”(2010).

### **Evaporación**

Es un proceso en la cual el agua líquida pasa a estado de vapor retirándose de la superficie evaporante, las cuales pueden ser lagos, ríos, suelos y vegetación mojada. Para éste proceso es necesario que la molécula de agua reciba energía proveniente de la radiación o de la temperatura del aire con la cual es posible el paso de líquido a vapor (Estudio FAO Riego y Drenaje 56, 2006).

### **Transpiración**

Es el proceso por el cual el agua contenida en los tejidos de la planta es removida hacia la atmósfera, principalmente desde los estomas. Ésta depende de la radiación, temperatura del aire, humedad atmosférica y el viento (Estudio FAO Riego y Drenaje 56, 2006).

### **Evapotranspiración (ET)**

Es el proceso que se da en conjunto entre la evaporación del agua del suelo y la transpiración de las plantas hacia la atmósfera, ésta depende de elementos climáticos (temperatura del aire, humedad atmosférica, velocidad del viento), morfológicos, fisiológicos (tipo de cultivo, variedad y etapa de desarrollo) y del nivel de humedad del suelo (Estudio FAO Riego y Drenaje 56, 2006).

### **Evapotranspiración de cultivo de referencia (ET<sub>o</sub>)**

Es la evapotranspiración de un área de referencia, donde se tiene un cultivo hipotético denso, como el pasto y se mantiene con ciertas características específicas. Éste concepto es introducido para el estudio de la evapotranspiración por parte de la atmósfera sin tomar en cuenta el desarrollo del cultivo y el manejo que se le da al suelo, bajo las condiciones de abundante agua (Estudio FAO Riego y Drenaje 56, 2006).

Ésta es calculada de muchas maneras, entre ellas la recomendada por la FAO, el método de Penman-Monteith.

### **Evapotranspiración del cultivo (ETc)**

Es un tipo de evapotranspiración, está dado bajo condiciones óptimas de suelo, fertilidad y suministro de agua, exceptuado de todo tipo de enfermedades. La ETc también es conocida como “uso consuntivo del agua” y según Vásquez et al.(2017) se calcula como:

$$ETc = ETo * K \text{ (mm/día) .....(0)}$$

Donde:

- K: coeficiente que relaciona los efectos de agua-suelo-planta, que proviene de la relación de  $Kc * Ks * Kh$ , donde Ks y Kh se asumirá como 1 por tener un suelo fértil y con disponibilidad de agua. Entonces se tendrá que  $K = Kc$ , donde Kc es el coeficiente propio del cultivo.
- ETo: evapotranspiración de referencia en mm/día

### **Coficiente del cultivo (Kc)**

También conocido como factor de cultivo, que indica el grado de cobertura o desarrollo del suelo por el cultivo. Entre los factores que determinan el Kc tenemos las características propias de cada cultivo, el calendario de siembra, duración del periodo vegetativo, condiciones climáticas, frecuencia de riego y lluvia(Vásquez et al., 2017).

Si tomamos la fórmula (0) ya mencionada y se despeja el K se tendrá,  $K = ETc/ETo$ , lo cual como ya se mencionó es lo mismo que  $Kc = ETc/ETo$ , entonces se puede entender ésta relación como la comparación entre las necesidades de agua de un cultivo en específico y las necesidades de agua de un cultivo de referencia bajo condiciones estándares.

De forma numérica esto se entiende bajo el valor de la unidad, si el valor obtenido es menor que uno, esto quiere decir que las necesidades de agua son menores que las estimadas con el cultivo de referencia. Y si el valor es mayor que la unidad esto quiere decir que las necesidades hídricas del cultivo planteado son mayores a las necesidades calculadas con el cultivo de referencia.

### **Coficiente de uniformidad (Cu)**

Es la magnitud que mide el grado de uniformidad que tiene los emisores en el riego, para lograr que toda el área designada dosifique el mínimo recomendado. Entre los factores que

intervienen en el Cu, entre los más comunes en el diseño se tiene los factores constructivos (por la fabricación) e hidráulicos (presencia de distintas presiones y caudales)(Pizarro, 1996).

Se calcula como:

$$CU = q_{25}/q_a \dots \dots \dots (1)$$

Donde:

- $q_{25}$ : Caudal medio del 25% de los emisores de menor caudal
- $q_a$ : Caudal medio de todos los emisores

## **2.6. Diseño hidráulico**

Es el cálculo que se realiza para que se pueda obtener las dimensiones de los diferentes componentes como tuberías primarias, secundarias, laterales, emisores, bombas para casos de bajas cargas, etc. Todo con la finalidad de obtener el mayor rendimiento en el funcionamiento del sistema, teniendo en cuenta las condiciones de topografía, suelo y cultivo(Mendoza, 2013).

## **2.7. Automatización del sistema de riego**

Resulta más conveniente la automatización del sistema de riego, ya que con esto se puede programar y manejar operaciones en forma secuencial, teniendo así más control y decidiendo cuándo inicia o finaliza el riego (Lliota et al., 2015).

La automatización se puede realizar por distintos métodos (tiempo de riego, volumen de riego, etc.), para éste caso se presentará en el método de automatización por volúmenes de riego.

### **Automatización por volúmenes – programación electrónica**

Es un método en el cual se mide el agua añadida al riego para alcanzar el volumen necesario que satisfaga el cultivo, una vez alcanzado se interrumpe de manera automática el paso del agua. Éste método subsana el error del método por tiempo de riego que en ocasiones no coincide con lo calculado(Pizarro, 1996).

Éste método involucra tres elementos fundamentales, los contadores de agua con sistemas de transmisión de datos, los programadores de riego y las electroválvulas, que suelen tener un elevado costo(Pizarro, 1996).

Por otro lado, se debe evaluar otras alternativas ya que para este caso se busca que el sistema de riego y automatización sea: lo más barato posible, componentes que sean fáciles de adquirir, flexibilidad para adaptarse a cualquier terreno, sistema inteligente con un requerimiento mínimo de interacción entre el sujeto y el terreno(Domingo, n.d.).

### 2.7.1. Arduino

Es una plataforma electrónica de código abierto basada en hardware y software libre, cuya facilidad de uso y sus múltiples funciones la hace cerebro de proyectos cotidianos hasta proyectos complejos y científicos. Para la utilidad del Arduino se tiene que utilizar el lenguaje de programación Arduino (basado en Wiring) y el software Arduino (IDE) basado en Processing(Hernández et al., 2011).

Ventajas que Arduino ofrece: económico, multiplataforma, entorno de programación simple y claro, software y hardware de código abierto y extensible (Arduino, 2018).

#### Entorno de desarrollo integrado (IDE)

Es un conjunto de herramientas software que permite escribir y ensayar, teniendo la capacidad de editar los programas creados por uno mismo si en caso se haya cometido algún error en el sketch (que es la plataforma de edición), con lo cual si es correcta la edición se procede a grabar en la memoria del microcontrolador de la placa Arduino, logrando que la placa se convierta en ejecutor autónomo del programa(Torrente, 2013). En la figura 1 se describe las partes del IDE.

Pestañas	Descripción
Archivo	Se puede: Crear, abrir un archivo, abrir archivos utilizados, ejemplos de programas de Arduino, elegir letra, tamaño, idioma etc.
Editar	Se puede: Copiar, pegar, comentar código; aumentar sangría y fuente de código.
Programar	Se puede: Verificar escritura de código, cargar el código a la placa; buscar y descargar librerías
Herramientas	Se puede: Seleccionar tipo de placa Arduino, tipo de puesto de conexión, visualizar el monitor serial.
Ayuda	Se puede: Solicitar ayuda en línea

**Figura 1: Interfaz IDE de Arduino**

FUENTE: Software Arduino

## **Lenguaje de Arduino**

Es un conjunto de instrucciones de los lenguajes C y C++ que están diseñadas de manera simplificada para el desarrollo de programas para microcontroladores. Ya que los lenguajes C y C++ que utiliza Arduino son de difícil aprendizaje y dominio, tanto por su contenido amplio como por sus detalles que llegan a ser innecesarios para el programador entusiasta que quiera ejecutar iniciativas de proyectos con Arduino. Por lo mencionado se puede decir que el “lenguaje” Arduino sirve de medio para agilizar el entendimiento y evitar la complejidad de los lenguajes C y C++ (Torrente, 2013).

## **Librería de Arduino**

Es un conjunto de funciones que facilitan la programación y abstracción de los detalles técnicos de un hardware para que sea fácil y sencillo el trabajo en el sketch. Funciones como: permitir el control de pantallas LCDs, leer y escribir datos en la memoria EEPROM y en una tarjeta SD, conectar Arduino, comunicar la placa con los programas ejecutados, etc.(Torrente, 2013).

## **Funciones en Arduino**

Una función es una pequeña parte del código al cual se identificará con un nombre, el cual está definido o el entusiasta programador puede crearlo con el nombre más conveniente que crea, haciendo fácil de identificar y llamar hacia el sketch. Con dicha función se podrá ejecutar todo lo incluido dentro de ella, tan solo escribiendo el nombre en el lugar deseado del código(Torrente, 2013).

Las ventajas de utilizar funciones: sirven de organizadoras del código, codifican una acción en el lugar deseado y facilitan la reutilización de los códigos en otros programas(Torrente, 2013).

## **Entrada y salida digital**

Para Arduino la entrada y salida digital es el pin donde irá conectada un sensor o actuador que detecte algún cambio o variación del mundo real, los valores detectados están entre 0V y 5V.

Se le llama señal digital porque tiene valores concretos y diferenciados, es decir, son señales de tipo todo/nada, si/no, verdadero/falso, 1/0.

## **Entrada y salida analógica**

Para Arduino una entrada analógica puede tomar cualquier valor dentro de un intervalo si se compara con la entrada digital que toma valores exactos (0V o 5V), esta toma valores decimales, es decir si una entrada analógica puede tomar un valor de 2.72 V o 3.85 V en un intervalo de 0V a 5V.

Las señales analógicas generalmente son de tipo continua con lo cual se tiene que necesitar un convertidor analógico a digital para poder leerlas. Estos valores son emitidos por sensores de temperatura, presión, caudal, humedad, intensidad de corriente, intensidad lumínica, etc.

## **Arduino Leonardo**

Es una placa de la familia Arduino, esta tarjeta utiliza el microcontrolador ATmega32u4 (Atmel/ Microchip), con características de poseer 20 entradas/salidas digitales de los cuales 7 pueden usarse como PWM (3, 5, 9, 10, 11, 12 y 13) entradas analógicas (A0-A5 y los pines digitales 4, 6, 8, 9, 10 y 12) un resonador cerámico de 16 MHz, conexión con reconocimiento dispositivo USB, conector de alimentación, conector ICSP y un botón de Reset(Anexo 10).

### **2.7.2. Sensores**

Son dispositivos que tienen la capacidad de detectar magnitudes físicas o químicas a la cual se le llama variables de instrumentación para posteriormente transformarlas en variables eléctricas que puedan ser leídas.

VARIABLES FÍSICAS MEDIBLES, se tiene temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. que dependerán del tipo de sensor a utilizar para saber si conectarla a una entrada digital o a una analógica.

Algunas características que se toman en cuenta en la selección de un sensor: rango de medida (en la cual se pueda aplicar), precisión (para saber cuál es el error máximo esperado), sensibilidad (para saber si es un sensor de entrada a salida y su variación de la magnitud de entrada), rapidez de respuesta (esto dependerá de la magnitud a medir y de la capacidad que tendrá el sistema para seguir), repetitividad (para saber qué error se esperaría al realizar la misma medición una y otra vez).

En la figura 2 se muestran los principales parámetros a medir y los instrumentos electrónicos que se utilizan en la programación con Arduino (Anexos 11, 12 y 13).

Tipo	Descripción	Imagen
Humedad de suelo	Compatible con Arduino, Mide la humedad en función a capacitancia, además posee una capa anticorrosiva que alarga la vida útil.	
Flujo de agua	Compatible con Arduino, cuantifica el flujo volumétrico de manera aproximada en L/min, L/h o m3/h; se coloca directamente en las tuberías.	
Ultra sonido	Compatible con Arduino, mide la distancia a un rango de 2 a 450 cm por medio de señales de sonido.	

**Figura 2: Sensores compatibles con Arduino**

### 2.7.3. Actuadores

Son dispositivos encargados de controlar (flujo de agua), regular (la energía entrante y saliente) y energizar (abriendo o cerrando el flujo de electricidad) los aparatos para su funcionamiento. En la figura 3 se muestran los principales dispositivos controlados compatibles con Arduino (Anexos 14, 15, 16 y 17).

Tipo	Descripción	Imagen
Módulo Relé	Dispositivo permite controlar el voltaje reduciéndolo para poder ser utilizado por el Arduino	
Driver mosfet	Modulo que funciona como interruptor encendiendo o apagando la señal	
Válvula solenoide	Electroválvula que funciona dejando pasar o cerrando el flujo, por medio de los impulsos eléctricos emitidos desde un microcontrolador	
Bomba de agua	Bomba tipo centrífuga compatible con Arduino, con entrada y salida tipo rosca, silenciosa y un bajo consumo de energía	

**Figura 3: Actuadores compatibles con Arduino**

#### **2.7.4. Almacenamiento de datos**

##### **Lector de MicroSD**

Módulo lector de memoria micro SD card con comunicación vía SPI, alimentación mejorada de hasta 5V, gracias a la incorporación de un regulador de voltaje, soporta tarjetas micro SDHC (Anexo 18).

#### **2.7.5. Fuentes de energía**

##### **Fuente de alimentación conmutada AC/DC**

Esta fuente se encarga de convertir la corriente alterna (AC) en corriente continua (DC), mediante transistores en conmutación, en un tamaño reducido, compacto y a una alta eficiencia (81%). Este tipo de fuente es de una potencia máxima de 250W, voltaje de entrada 200V-240V AC, voltaje de salida de 12V DC, corriente máxima que transita 20A, ideal para alimentar equipos como motores DC, servomotores, luces led, etc.(Naylamp Mechatronics, 2021c).

##### **Fuente de alimentación tipo cargador AC/DC**

Esta fuente se encarga de convertir la corriente alterna (AC) en corriente continua (DC), mediante transistores en conmutación. Esta fuente tiene voltaje de entrada de 100v – 240 AC, voltaje de salida de 12V DC, corriente máxima que circula de 1A, eficiencia de 80% en salida, tiene enchufe AC tipo A(IEC), con cable de 90 cm. Ideal para encendido de Arduino, válvulas solenoide DC, mini bomba de 12V, etc.(Naylamp Mechatronics, 2021d).

#### **2.7.6. Programación**

##### **ArduinoDroid**

ArduinoDroid - Arduino/ESP8266/ESP32 IDE es una aplicación para celular, cuya función se asemeja al software Arduino. Esta tiene funciones de escritura, compilación y carga de códigos, uso de bibliotecas de Arduino o ESP8266/ESP32 todo a través de USB o WIFI desde celulares con dispositivo Android(Anton Smirnov, 2021)(Anexo 19).

##### **Interfaz de ArduinoDroid**

El aplicativo cuenta con pestañas similares al software Arduino, con lo cual se puede utilizar sketch creados en Arduino y abrirlos, editarlos, compilarlos y cargarlos en ArduinoDroid.

En la Figura 4 se muestra el interfaz de navegación del aplicativo, en el cual el sketch puede abrir archivos con códigos, editar el mismo archivo con el teclado del aplicativo, verificar si está correctamente escrito, cargar el código al aplicativo y verificar si el código se ejecuta según lo establecido.

Por otro lado, cada pestaña descrita en la Figura 4 tiene comandos para buscar y subir ejemplos propios del aplicativo, buscar librerías que ayudan en la digitación de la programación, poseer un solucionador de problemas y ayuda en línea.

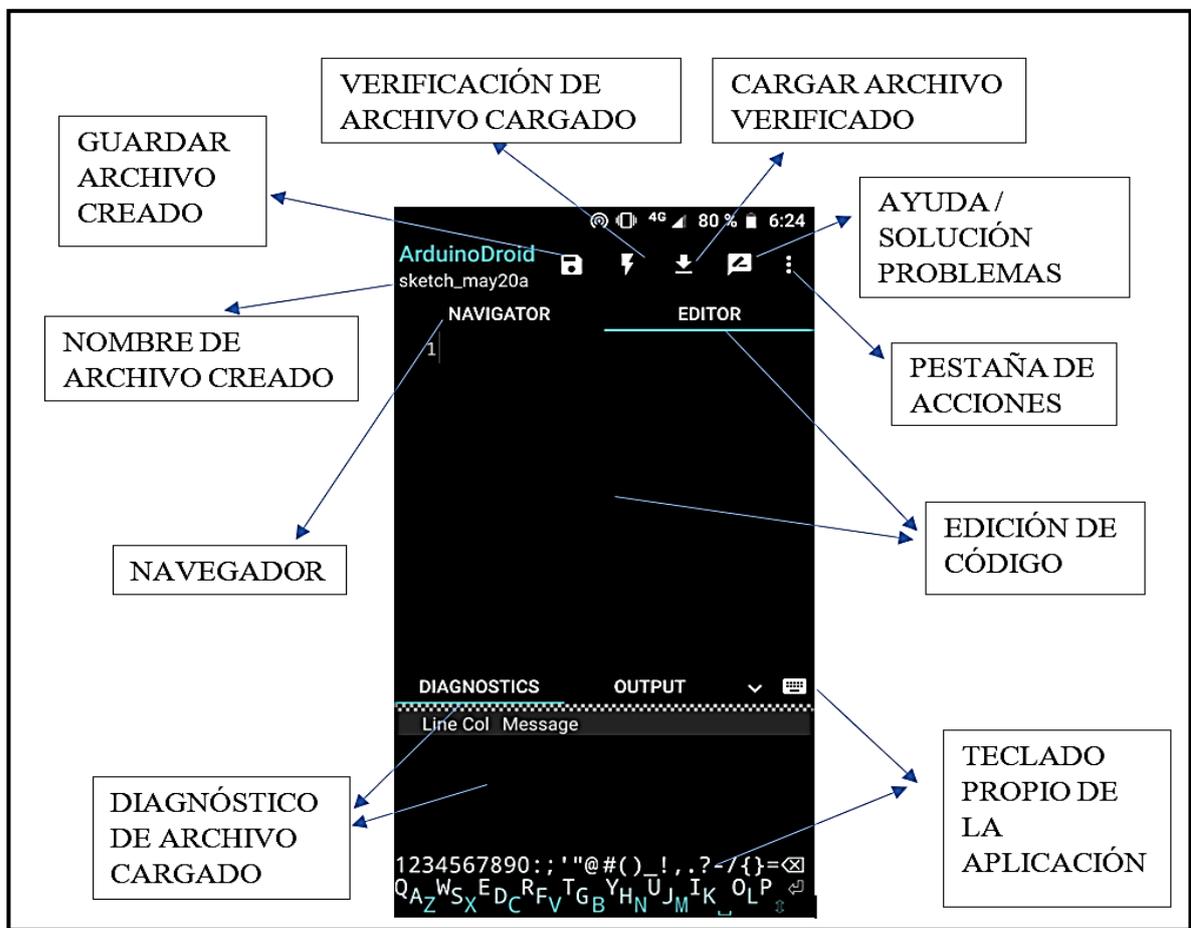


Figura 4: Interfaz de ArduinoDroid

### III. METODOLOGÍA

#### 3.1. Materiales y equipos

Los materiales utilizados fueron básicamente para la preparación del terreno (pico, lampa, rastrillo); para la delimitación del terreno (troncos, alambre, clavos, malla reshell, pabilo); para la instalación (sierra, pegamento, tuberías, conexiones, implementos de riego, reservorio de agua); para el control (software libre Arduino, sensores de humedad, caudalímetro, electroválvula, válvulas, filtro, interruptores, complementos para el hardware y caseta de control); para la recopilación de información (base de datos (grilladas, homogenizadas e interpoladas), hoja de cálculo Excel, tarjeta micro SD) y para redacción y expresión gráfica (hoja Word y AutoCAD).

#### 3.2. Descripción del área experimental de estudio

El estudio se realizó en el asentamiento humano Palmeras de Huarangal ubicado en las coordenadas  $12^{\circ}16'23.35''S$  -  $76^{\circ}51'31.42''O$ . El área neta del proyecto es de  $50\text{ m}^2$  donde  $38\text{ m}^2$  son de áreas cultivables y el periodo de inicio de la investigación se dio entre los meses de enero del 2021 a agosto del 2022.



**Figura 5: Ubicación del área experimental de trabajo**

FUENTE: (Google Maps, 2021)

### 3.3. Metodología aplicada

#### 3.3.1. Recopilación de información

##### Análisis de la calidad de agua

Para el presente proyecto se analizó la composición del fluido hídrico de la fuente que provee al área de estudio, el cual proviene del pozo subterráneo que abastece de agua a los pobladores de Palmeras de Huarangal con una dosificación diaria de 160L/h en las mañanas a 180 L/h por la noche, de donde se tomó como muestra 1.0 L, que sirvió para el análisis en el laboratorio de Agua, Suelos, Medio Ambiente y Fertirriego de la UNALM (Anexo 5), dando los resultados siguientes:

**Tabla 4: Análisis de la calidad de agua**

PARÁMETROS	RESULTADO	UNIDADES
CE	3.34	mmhos/cm
pH	7	-
Calcio	16.61	meq/L
Magnesio	3.93	meq/L
Sodio	11.74	meq/L
Potasio	0.24	meq/L
CATIONES		
TOTAL	32.53	meq/L
Cloruro	19.97	meq/L
Sulfato	7.41	meq/L
Bicarbonato	5.17	meq/L
Nitratos	0.46	meq/L
Carbonatos	0	meq/L
ANIONES		
TOTAL	33.01	meq/L
SAR	3.66	-
Clasificación	C4-S2	-
Boro	0.88	mg/L

Con lo cual se tiene que la interpretación de la muestra de agua es: Agua con una calidad muy alta de salinidad, se usa para riego en suelos permeables con adecuado drenaje y con riego en exceso para mantener niveles de sales adecuados en el suelo (CE: Clase C4) ; Agua con calidad media en sodio, apto para utilizarse en suelos de textura gruesa o suelos orgánicos de alta permeabilidad (SAR/RAS: S2) (Anexo 6) y en cuanto a los niveles de boro

hallados nos indicaron que su utilización está condicionada para cultivos sensibles, semi tolerantes y tolerantes en un rango de 0.33 a 3.75 ppm (mg/L = ppm; 0.33mg/L a 3.75mg/L)(Anexo 7).

En resumen, se concluye que el agua tiene un nivel medio para su uso y en función a esto se seleccionaron los potenciales cultivos más adecuados para su producción.

### **Caracterización física de suelo**

Para el área neta de estudio de 40 m<sup>2</sup> se extrajo una muestra de 5 kg de suelo a una profundidad de 30 y 40 cm, el cual fue extraído en la intersección de las diagonales del área. La muestra por ser de un área de menor extensión se determinó realizar una caracterización física dando como resultado los siguientes aspectos:

**Tabla 5: Análisis textural del suelo**

Parámetros	Resultados	Método
Textura	Arenosa - Franca (AF)	Tacto
Estructura	Sin estructura / granos simples	Simple inspección
Poros	Alto contenido /infiltración	Simple inspección
Consistencia	Baja/Suelta	Simple inspección
plasticidad	No plástica	Tacto
MO <sub>a</sub>	Escasa	Simple inspección
Aireación	Excelente	Simple inspección
Color	2.5Y-7/2	Tabla de Munsell
Densidad aparente	1.5 gr/cm <sup>3</sup>	Teórica

*Nota: a = materia orgánica*

FUENTE: (Pastor, 2005).

En el cual la textura y la plasticidad se realizaron al tacto, siguiendo el procedimiento: a) se tomó medio puñado de suelo, b) se añadió agua hasta su saturación, c) se comenzó a realizar bolas con el suelo y de acuerdo al Anexo 3 se establecerá su tipo de textura; la estructura, los poros, la aireación, MO (materia orgánica) y la consistencia se establecieron por simple inspección, por ser arena suelta con escasas partículas de limo y arcilla; para el color se utilizó la tabla de Munsell (Anexo 4).

Con estos resultados se puede afirmar teóricamente que el suelo posee una alta permeabilidad, pobre estructuración, excelente aireación, escasa retención de agua lo que implica una baja capacidad de campo, plasticidad baja, escasa reserva de nutrientes y actividad biológica y carece de propiedades coloidales (arcilla, humus, sílice)(Pastor, 2005). Por lo cual para la siembra se prevé un mejoramiento del suelo con materia orgánica, tierra y arena (Anexo 42).

## Datos climáticos

La base de datos obtenidos en la Tabla 6 para las coordenadas 12°16'23.35"S - 76°51'31.42"O existentes actualmente, comprenden entre los años 2007-2016, las cuales son extraídas de las estaciones meteorológicas del Perú, cuyos resultados pasaron un control de calidad automático y visual, para luego ser completados y homogenizados con variables satelitales del producto ERA5-Land, luego de tener estos datos completados se interpolan con variables grilladas, para finalmente con estos poder obtener la Eto mediante la ecuación de Penman-moneteith FAO-56 a escala diaria es decir en mm/día. Cabe mencionar que estos resultados muestran mejores rendimientos en zonas costeras y andes occidentales/orientales del Perú(Huerta et al., 2022).

**Tabla 6: Promedio mensual de evapotranspiración potencial de los últimos 10 años**

Año/ mes	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Set	Oct	Nov	Dic
2007	3.64	3.94	3.77	3.91	3.21	2.43	1.93	1.94	2.32	2.64	3.35	3.22
2008	3.38	3.79	3.91	3.91	2.70	2.17	2.29	2.16	2.61	2.63	3.38	3.50
2009	3.69	3.68	3.73	4.03	3.17	2.42	2.14	2.16	2.51	2.83	3.35	3.08
2010	3.17	3.70	4.11	4.07	3.11	2.50	2.09	2.35	2.79	2.59	3.14	3.26
2011	3.75	3.71	3.99	3.84	3.12	2.57	2.14	2.12	2.71	2.70	3.59	3.41
2012	4.16	3.81	3.87	4.11	3.20	2.65	2.42	2.18	2.65	2.65	3.35	3.27
2013	4.24	4.00	3.75	4.10	2.99	2.40	2.10	2.17	2.75	2.80	3.25	3.67
2014	3.67	3.93	3.68	3.88	2.76	2.68	2.20	2.31	2.62	2.89	2.96	3.26
2015	3.83	3.35	3.81	3.74	3.05	2.74	2.28	2.39	3.02	2.99	3.14	3.34
2016	3.91	3.93	4.22	4.04	3.40	2.69	2.34	2.33	2.95	2.88	3.70	3.61
Promedio (mm/día)	3.74	3.78	3.88	3.96	3.07	2.53	2.19	2.21	2.69	2.76	3.32	3.36

FUENTE: (Huerta et al., 2022)

## Cultivos propuestos

Según los resultados obtenidos del análisis de agua y la caracterización física del suelo que arrojaron elevada salinidad (CE: 3.34 mmhos/cm = CE<sub>w</sub>), tolerancia media al boro, suelo arenoso, buena permeabilidad y escaso contenido de materia orgánica. Se elaboró una lista de cultivos tolerantes y con tolerancia media con respecto a los parámetros de CE<sub>e</sub> y Boro: Cabe precisar que la CE<sub>w</sub> tiene relación con la conductividad en el extracto de saturación del suelo (CE<sub>e</sub>), en donde la CE<sub>e</sub> = 1.4\*CE<sub>w</sub> (donde se toma 1.5 por tener una textura franca arenosa y frecuencia de riego fue diaria, este valor es tomado entre los valores 1.3 a 1.5 dependiendo de la textura del suelo y frecuencia de riego)(Cropaia, 2023).

Entonces se obtiene la relación:

**Tabla 7: Tolerancia a la Salinidad (CEe) y al Boro**

	Valores de CEe (mmhos/cm) para un determinado % de producción de cultivo					Tolerancia de Boro mg/L
	100	90	75	50	0	
Hortalizas	100	90	75	50	0	mg/L
Beterraga	4	5.1	6.8	9.6	15	4
Tomate	2.5	3.5	5	7.6	12.5	2
Espinaca	2	3.3	5.3	8.6	15	-
Apio	1.8	3.4	5.8	9.9	18	2.0-4.0*
Col /Repollo	1.8	2.8	4.4	7	12	2.0-4.0*
Ají	1.5	2.2	3.3	5.1	8.5	2
Lechuga	1.3	2.1	3.2	5.2	9	4

FUENTE: (Pizarro, 1996); \*(Vásquez et al., 2017).

De la Tabla 7 presentada, se tomó como referencia un máximo del 50 % en la reducción de la producción del cultivo por presencia salina, dicho valor se tomó en función del resultado de hallar la salinidad del suelo 4.67 mmhos/cm, es decir la producción de lechuga se verá disminuida hasta en un 50 %, debido a la presencia de las sales, por otro lado, cultivos como la beterraga son más tolerantes a la salinidad por lo que se verá afectado hasta en un 10% en la disminución de su producción.

En la Tabla 9 se apreció las principales características agronómicas de los cultivos propuestos, con los cuales se realizaron las distribuciones en cada una de las áreas de estudio planteadas. Siguiendo los principales criterios de asociación expuestos en la Tabla 8, lo cual se buscó que los cultivos fueran distribuidos para generar una mayor diversidad y asociación entre las plantas, propiciar un aumento en la productividad, evitar competencia entre la misma familia de cultivos y competencia por los rayos de sol, etc.

Es así que se aplicó los principios básicos de la agroecología citado por Altieri & Nicholls (2019) al diversificar las especies cultivadas con 7 cultivos en todo el área neta a cultivar; dentro del total de las áreas se planteó en cada sub unidad la distribución y la complementariedad de los cultivos evitando competencia por nutrientes y rayos solares; la optimización del control biológico al diversificar los cultivos, ya que estos actúan como repelentes e inhibidores de plagas del cultivo asociado; al propiciar condiciones favorables al suelo generando la rotación de los cultivos; planteando biofertilizantes foliares y radiculares (Anexo 40), los cuales favorecen una interacción biótica entre las plantas y los organismos vivientes en ellas lo cual genera hábitat apropiado.

En la Figura 7 se visualizó la distribución de las hortalizas donde se tomaron en cuenta la ubicación de cada uno de los cultivos en cada grupo según la familia a la que pertenecen, ciclo vegetativo, parte del cultivo comestible, tamaño en altura y diámetro en el área de cultivo. Para generar una mayor diversidad en cuanto a las necesidades y sensibilidad a las plagas, tener cultivos de periodos cortos y a la vez dejar tiempo para los de periodos largos, cultivos de mayor tamaño que dejen espacio para cultivar otros de menor diámetro de plantación, con la finalidad de utilizar al máximo toda el área para generar una mayor producción.

Así por ejemplo para el grupo 4 (Figura 7) se colocó a los costados laterales del área, el cultivo de betarraga que pertenece a la familia de las *Chenopodiaceae*, la cual tiene un diámetro de ocupación del cultivo de 10 cm, una altura de 30 cm, un periodo de vegetación entre 50 y 60 días y una época de siembra de todo el año. A este cultivo lo asociamos con el apio de la familia *Apiaceae*, cuyo diámetro de cultivo es de 30 cm, altura de 50 cm, periodo de vegetación 90 días y época en verano, otoño e invierno, siendo mayor al de la betarraga en los tres primeros aspectos. También están asociados estos dos al cultivo de tomate de la familia *Solanaceae*, con un diámetro de cultivo de 100 cm, una altura entre 50 a 80 cm, un periodo vegetativo entre 80 y 120 días y una época de siembra de todo el año. De igual manera es como se asociaron los cultivos de los grupos 1, 2 y 3.

En cuanto a la siembra y cultivo, se planteó el calendario de la Figura 6 donde se muestra los meses de siembra según la época, la cantidad de días hábiles para la cosecha y el distanciamiento entre plantas en cada hilera de cada sub unidad/ sub área de cultivo.

Cabe precisar que se planteó la siembra para el mes de julio del 2023, según el calendario tentativo, se dará inicio con la siembra de los cultivos como: col, espinaca, lechuga, brócoli y tomate al mismo tiempo, y se distribuirá según el diseño planteado en la figura 7, para el primer mes de la primavera se empezará con la siembra de los cultivos de ají, betarraga y tomate.

En cuanto a la cosecha, la primera que se dará será las cultivadas en el mes de julio, dando la primera producción de espinacas a los 40 días de su siembra en los grupos 1 y 3, la última de la época de invierno será la col a unos 90 días de su siembra en los grupos 1 y 2; ají, tomate y betarraga se cosecharán a fines y mediados de primavera según la Figura 6.

En la segunda campaña lo que se realizará es la rotación de cultivos entre los grupos 1 y 4, y los grupos 2 y 3 de tal manera que el cultivo cosechado va directamente a la siembra en la misma posición de la hilera del cultivo del grupo a rotar.

Grupo	Cultivo	Época de siembra	Siembra												T	Cosecha	
			Verano		Otoño			Invierno			Primavera						
			E	F	M	A	M	J	J	A	S	O	N	D			Días
2	Ají	P / V														120	50
4	Apio	V/O/I														90	7
2,4	Betarraga	TA														60	30
1,2	Brócoli	TA														60	30
1,2	Col/repollo	TA														90	30
1,3	Espinaca	O / I														40	todo a la vez
1,3	Lechuga	O / I / P														70	25
3,4	Tomate	TA														120	30

nota: T= Tiempo de siembra en días

**Figura 6: Calendario de siembra y cosecha para un ciclo de vegetación**

**Tabla 8: Criterios de selección utilizados en los grupos de cultivos**

Criterio	Rango	Comentario
Distancia entre plantas	0.1 - 0.8 m	En cada hilera se colocó cultivos de mayor separación y se rellenó los espacios con cultivos de menor separación entre plantas.
Diámetro de planta	0.1 - 1.3 m	En cada hilera se colocó cultivos con mayor diámetro de desarrollo, se rellenó los espacios con cultivos de menor diámetro para generar una mayor producción.
Familia	más de 1	Se procuró tener cultivos de diferentes familias entre los grupos y no se colocaron juntos en una misma hilera dos cultivos que tengan la misma familia, generando mayor diversidad, necesidades variables y no ser sensibles a las mismas plagas.
Periodo del cultivo	40- 120 días	Se seleccionó los cultivos de periodos largos y se juntó con los cultivos de periodos cortos, de tal manera que los segundos van produciendo uno o dos cosechas mientras van madurando los primeros.
Parte comestible	Fruto Hoja	Se trató de poner cultivos con frutos y otros cultivos con hojas, debido a que tendrán necesidades distintas de nutrientes y también los espacios de cultivo variados.
Altura de cultivo	0.2 - 0.9 m	Se buscó que cultivos de mayor tamaño estén ubicados en las partes centrales o extremas contrarias al recorrido del sol, de tal manera que no cubran los rayos solares y no provoquen sombra a los cultivos de menor tamaño.
Época de siembra	I-O-V-P Todo el año	Se colocaron cultivos con épocas de siembra de todo el año y se asociaron con cultivos que tengan épocas únicas o varias épocas, con lo cual, por ejemplo, se puede cultivar betarraga todo el año, pero solo en primavera o verano se cultivará ají teniendo hasta 4 cosechas de betarraga antes de cultivar ají.
Ciclo de vida	Anual Bianual	Se buscó que los cultivos que duren dos años estén en cada grupo y sean completadas con cultivos anuales para tener siempre cultivos en desarrollo.

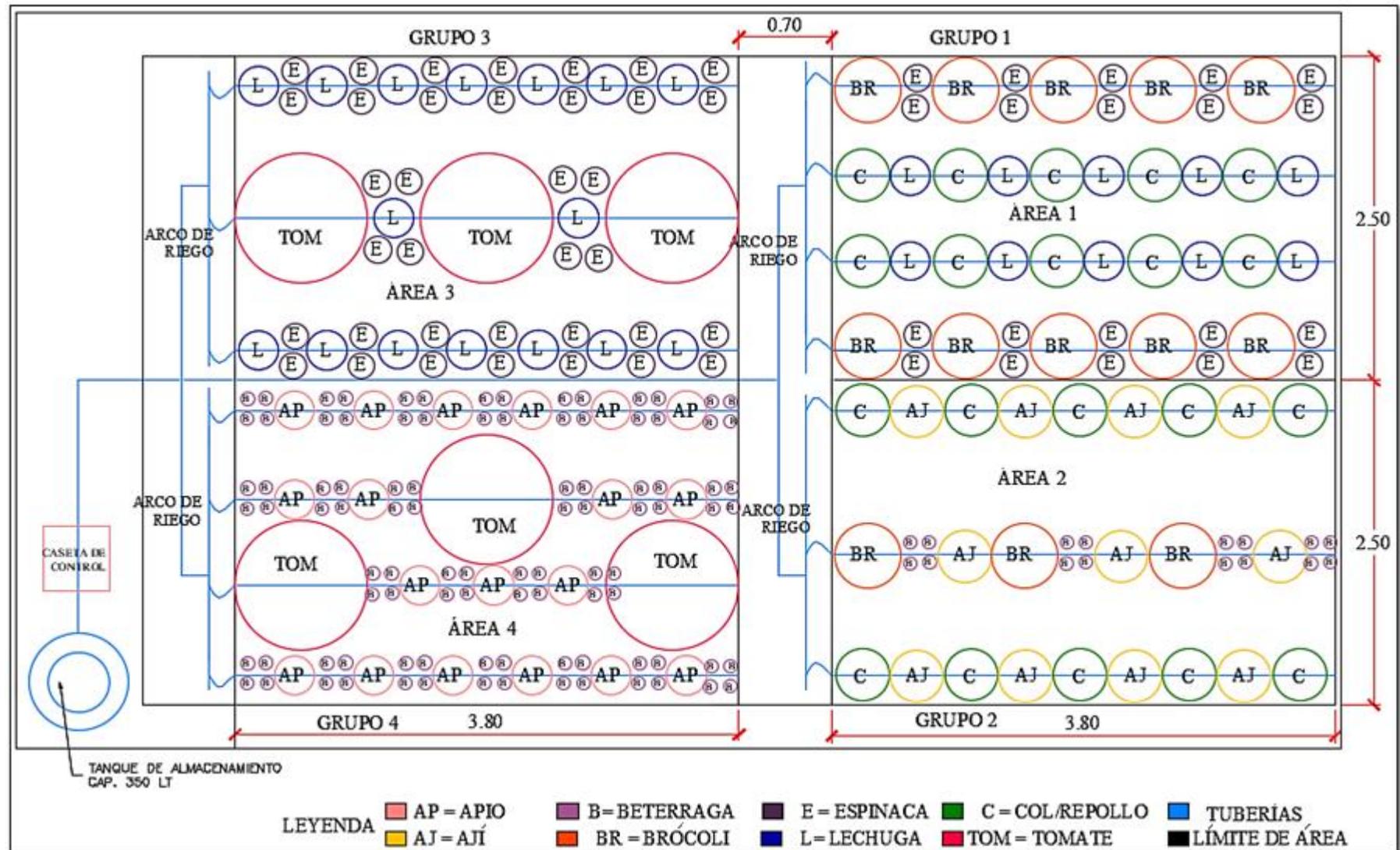


Figura 7: Distribución de los cultivos seleccionados en el área de estudio

**Tabla 9: Características agronómicas de los cultivos seleccionados para el proyecto**

CULTIVO		PERIODO DEL CULTIVO (DÍAS)		CLIMA	ZONAS DE PRODUCCIÓN **	TIPO DE SUELO		TAMAÑO DE LA PLANTA		DISTANCIAMIENTO (m)		ÉPOCA DE SIEMBRA	RENDIMIENTO	CICLO DE VIDA
NOMBRE	FAMILIA	MADURACIÓN	COSECHA			MO/ salinidad	pH	ALTURA (m)	DIÁMETRO (m)	PLANTAS	SURCOS			
AJÍ	SOLANACEAE	80-120	40-60	Cálido-Templado 16 a 24°C	Lima (Rímac, Chillón, Lurín) Áncash, barranca	salinidad y acidez media	5.5 - 7.0	0.6-1.2	0.4-0.6	0.5	0.8-1.2	P / V	1.2-1.6 kg/m2 2.0-3.0 kg/m2	Anual
APIO	APIACEAE	90	7	Templado, 16-21°C	Lima (Rímac, Chillón), Junín (Tarma) Cañete	francos/Ricos en MO	6.5 - 6.8	0.5	0.3	0.15-0.2	0.7	V / O / I	3 atados /m2 4 atados /m2	Bianual
BETARRAGA	CHENOPODIACEAE	50-60	30	Templado, cálido, 14 - 22° C	Lima (Rímac, Chillón), Cañete, Junín (Tarma)	MO/Tolerancia salinidad y acidez	6.0 - 6.8	0.3	0.1	0.1-0.8	0.6-0.7	TA	2.5 atados /m2	Bianual
BRÓCOLI	BRASSICACEAE	50-70	20-30	Templado, tolera heladas, 14-18° C	Lima (Rímac, Chillón), Cañete, Huaral, Ica	MO/Tolerancia salinidad y acidez	6.0 - 7.5	0.6-0.9	0.5	0.4-0.5	0.7-0.8	TA	0.8 - 1.2 kg/m2	Anual
COL /REPOLLO	BRASSICACEAE	70-100	20-30	Templado, 15-20° C	Lima (Rímac, Chillón), Arequipa, Junín, Huaral	MO/Tolerancia salinidad y poca acidez	6.0 - 7.5	0.3-0.4	0.5-0.7	0.4-0.6	0.7-0.8	TA	1.8 frutos/m2	Bianual
ESPINACA	CHENOPODIACEAE	40	todo a la vez	Templado, toleran heladas 13-18° C	Lima (Rímac, Chillón), Cañete, Junín (Huancayo)	MO/Tolerancia salinidad y poca acidez	6.0 - 6.8	0.2	0.2	0.1	0.6-0.8	O / I	1.0-1.5 kg/m2	Anual
LECHUGA	ASTERACEAE	60-80	15-25	Templado, 10-20° C	Lima (Rímac, Chillón), Cañete, Junín (Tarma)	MO/Tolerancia salinidad y poca acidez	6.0 - 6.8	0.2	0.3	0.3	0.8	O / I / P	6.0 frutos/m2	Anual
TOMATE	SOLANACEAE	80-120	30	Cálido, no toleran heladas, 18-28° C	Lima (Rímac, Chillón), Mala, Ica, Loreto, Cañete	MO/Tolerancia salinidad y acidez	5.5 - 6.8	0.5-0.8	1.0-1.5	0.2-0.4	2.2-2.5	TA	5-8 kg/m2	Anual

Nota: O (otoño), I (invierno), P (primavera), V (verano); \* (crecimiento rastrero); \*\* (zonas de reproducción más comunes cercanas a la ciudad de Lima); TA: todo el año

FUENTE: (Ugás et al., 2000)

### 3.3.2. Diseño agronómico

#### Balance hídrico

Es el cálculo de las necesidades totales de lámina de riego (Nt) por parte del cultivo, para este caso se utilizó el procedimiento descrito en el esquema 6.1 de Pizarro (1996) donde se describe el cálculo final tomando en cuenta que es un balance para un riego localizado de alta frecuencia.

Obteniendo la primera relación:

$$N_n = E_{Trl} - P_e - G_w - \Delta w \dots \dots \dots (2)$$

donde:

- $N_n$ : necesidades netas de lámina de riego.
- $E_{Trl}$ : es la evapotranspiración corregida según las condiciones locales, obtenidas de  $E_{Tc} * K_L$ , donde a su vez  $K_L$  es el coeficiente de localización y según Pizarro esta se obtendrá en función del área sombreada ( $A_t = (\pi * (D)^2) / (4 * (E_p * E_s))$ ), donde  $E_p$  espaciado entre plantas,  $E_s$  espaciado entre surco y  $D$  diámetro del cultivo). Y tomando como referencia el promedio de las fórmulas de Decroix.....  $K_L = 0.1 + A_t$  y Hoare.....  $K_L = A_t + 0.5 * (1 - A_t)$  (1996).

Y finalmente este valor  $E_{Tc} * K_L$  obtenido se le multiplicó el criterio de corrección de Hernández Abreu por condiciones locales climáticas que es un valor entre 1.15 a 1.2, tomando el valor de 1.2 para riego localizado de alta frecuencia.

Por otro lado, también se realizó la corrección por advección, pero esta se desestima por el tamaño de terreno utilizado  $q$  es inferior al 0.01 ha.

- $P_e$ : precipitación efectiva, nula para el caso de zonas desérticas o con escasa lluvia
- $G_w$ : aporte capilar, para casos con Napa freática próxima.
- $\Delta w$ : variación de almacenamiento de agua, para cálculos de mayores necesidades no se toma en cuenta.

Tomando como referencia el punto (2), mayormente se cumplirá que

$$N_n = E_{Trl} \dots \dots \dots (3)$$

Como segunda relación se tendrá el (A) que en este caso será:

$$A = N_n + P_p \dots \dots \dots (4)$$

Donde:

- A: agua aplicada al riego
- Pp: pérdida de agua por percolación, en riego localizado son debidas a percolación en su mayoría.

Ahora se calcula la eficiencia del riego en la aplicación de agua (Ea) se tendrá

$$Ea = Nn/A.....(5)$$

Entonces remplazando y despejando (5) en (4) se obtiene

$$Pp = A*(1-Ea).....(6)$$

Ahora a esto se tiene que adicionar una cierta cantidad R de agua por necesidades de lavado con lo cual para este caso se asume momentáneamente que no hay percolación, se tendrá entonces que el agua a regar A será:

$$A = Nn + R.....(7)$$

Además, se deduce el coeficiente de necesidades de lavado (LR) como:

$$LR = R/A.....(8)$$

Donde para cálculos rápidos, LR se obtiene de:

$$LR = CEw/2CEe$$

Donde: CEw, será la conductividad eléctrica del agua y CEe, será la conductividad eléctrica del extracto de saturación del suelo que dependerá del cultivo. Entonces si se reemplaza y despeja (8) en (7) se tendrá:

$$A = Nn + A*LR.....(9)$$

Para homogenizar las relaciones obtenidas en (9), (6) y (4) podemos escribir:

$$A = Nn + A*K.....(10)$$

Donde K es (1-Ea) en caso de pérdidas por percolación y LR para los casos de necesidad de lavado. Cumpliendo que si  $Pp > R$  entonces  $(1-Ea) > LR$  con lo que se controlará el nivel de salinidad mínima debido a que se pierde más de lo que se pretende perder. Para el caso contrario se habrá que incitar a una percolación o un desnivel para que haya la pérdida necesaria.

Después se obtuvo

$$A = Nn / (1-K).....(11)$$

Donde se tomó para el caso donde K es mayor, se adicionó el coeficiente de uniformidad (Cu) para calcular el Nt dando así:

$$N_t = A/C_u \dots \dots \dots (12)$$

Y finalmente relacionado el punto (11) en (12) se tuvo

$$N_t = N_n / ((1-K) * C_u) \dots \dots \dots (13)$$

Que fue el volumen neto de agua a adicionar.

Por otro lado, cabe mencionar que se tuvo que calcular el caudal del sistema en función a las necesidades halladas.

Para esto el  $N_t$  (mm/día) será transformado en  $Q$  (caudal del sistema) con lo cual se tiene que recordar que 1mm es equivalente a 1 L/m<sup>2</sup> y por conversión y multiplicando el  $N_t$  por el área total utilizada tendremos el  $Q$  en L/s. Ya obtenido el total de caudal como máximo para el sistema él se utilizó para obtener la potencia de la bomba (Ecuación 22).

Por último “se debe distribuir los goteros buscando la uniformidad del riego, donde el espaciamiento óptimo entre goteros debe ser de 80% del diámetro de bulbo húmedo”(Mendoza, 2013). Para hallar “el diámetro aproximado del bulbo húmedo se utiliza la ecuación de Zazueta” (Sepúlveda & Zazueta, 2004).

$$D_b = 0.01 * (q * r / I)^{1/3} \dots \dots \dots (14)$$

donde:

- $D_b$ : diámetro del bulbo húmedo, distancia que dependerá de la raíz del cultivo en m.
- $q$ : caudal de descarga del emisor en L/hr
- $r$ : profundidad de la raíz en m
- $I$ : tasa de infiltración básica en m/s

### 3.3.3. Diseño hidráulico

Es el cálculo que se realizó para que se pueda obtener las dimensiones de los diferentes componentes como tuberías primarias, secundarias, laterales, emisores, bombas para el caso de una baja carga, etc. Todo con la finalidad de obtener el mayor rendimiento en el funcionamiento del sistema. Teniendo en cuenta las condiciones de topografía, suelo y cultivo(Mendoza, 2013).

Para el buen diseño se procedió realizando las dimensiones de tuberías siguiendo la relación

$$A = \pi * D^2 / 4 \dots \dots \dots (15)$$

Donde:

- A: el área de la sección de la tubería medida en m<sup>2</sup>
- D: diámetro del interior de la tubería en m

para casos comunes se relaciona el diámetro con el caudal, para efectos de cálculo se representa como

$$V=Q/A$$

Donde:

- V: velocidad del fluido en m/s
- Q: caudal en m<sup>3</sup>/s

Por otro lado, se tuvo que considerar las pérdidas por rozamiento, pérdidas por carga, pérdidas por accesorios. La metodología utilizada para este caso fue:

1. Se realizó un cálculo rápido para hallar el diámetro económico mediante la fórmula conservadora de Bresse, 1860 (Bernis, 2020).

$$\text{Aplicando Bresse: } D= 1.5 Q^{0.5} \dots\dots(16)$$

2. Con el caudal del sistema se seleccionó el diámetro de la tubería que cumplió y que está disponible en el mercado comercial.

$$Dt (Q, D) \dots\dots\dots(17)$$

3. Se calculo la pérdida de manera general como:

$$Hf = J*L \dots\dots\dots(18)$$

Donde:

- Hf: pérdida de energía o carga producto de la fricción en m.
- J: gradiente o pérdida de carga por cada metro de tubería m/m.
- L: longitud de la tubería en m.

4. Para el cálculo de la Hf la fórmula que se utilizo fue la de Hazen y Williams:

$$Hf = 1.131*10^9*(Q/C)^{1.852}*(1/D)^{4.87}*L \dots\dots\dots(19)(Mendoza, 2013).$$

Donde:

- C: coeficiente de fricción propio de cada material (adimensional) (Anexo 8)
- D: diámetro interno de la tubería en mm.
- L: longitud de la tubería en m.

- Q: caudal circulante en m<sup>3</sup>/hora
5. Se considero el número de salidas para dar la pérdida total, esta se conoce como el coeficiente de Christiansen que afecta a la ecuación de Hazen y Williams, tomando en cuenta que no es aplicado para tuberías de flujo cerrado, se tiene:

$$H_{ft} = f * H_f \dots\dots(20)$$

Donde:

- H<sub>ft</sub>: pérdida de carga en la tubería afectado el coeficiente de Christiansen en m.
  - f: coeficiente de Christiansen (Anexo 9)
  - H<sub>f</sub>: pérdida de carga con la ecuación de Hazen y Williams en m.
6. Se calculó la velocidad con la relación (15) y (16), y corroborar que este resultado este entre 0.6 a 2.4 m/s, sabiendo que los valores más utilizados son 1 a 2 m/s(Briceño et al., 2012). Cabe mencionar que, de no cumplir los cálculos, se tuvieron que volver a modificar el diámetro y proseguir con los pasos anteriores.
7. Con los calculos anteriormente hallados, se calculó la presión a la entrada de la lateral con la relación:

$$PEL = P_g + \frac{3}{4} H_{ft} + AE \dots\dots\dots(21)$$

donde:

- PEL: presión a la entrada de la lateral
  - P<sub>g</sub>: presión del gotero
  - H<sub>ft</sub>: pérdida de carga en la tubería afectado el coeficiente de Christiansen en m.
  - AE: altura del elevador
8. Se añadió a la pérdida por fricción en la tubería (pérdidas primarias) la pérdida por accesorios (pérdidas secundarias) que están entre 10 a 25 % de las pérdidas primarias(Mendoza, 2013).
9. Se calculó la carga dinámica total con la cual se pudo hallar la potencia necesaria que requerirá del sistema.

$$CDT = PEL + PPC + \Delta SD + PS \dots\dots\dots(22)$$

donde:

- CDT: carga dinámica total

- PEL: presión en la entrada de la lateral
- PPC: pérdida en la tubería principal más perdida en la tubería de conducción
- ΔSD: desnivel entre la succión y la descarga
- PS: pérdidas secundarias que están entre 10 a 25% de las pérdidas principales(Mendoza, 2013).

10. Con lo que finalmente se halló la potencia de la bomba con la relación:

$$HPb = (Q*CTD) / (75*Ef).....(23)$$

- HPb: potencia de la bomba en Hp
- Q: caudal del sistema en L/s
- CTD: carga dinámica total en m
- Ef: eficiencia de la bomba que esta entre 0 y 1(Mendoza, 2013).

### 3.3.4. Automatización del sistema

El control se realizó utilizando Arduino, en el cual se acoplaron sensores y capacitadores que permitieron medir y almacenar los datos de temperatura y humedad ambiental, humedad del suelo, caudal de la fuente y permitió por medio de una señal encender o apagar la dosificación de agua según las necesidades presentadas.

En la figura 8 se muestra el esquema general del procedimiento de la automatización.

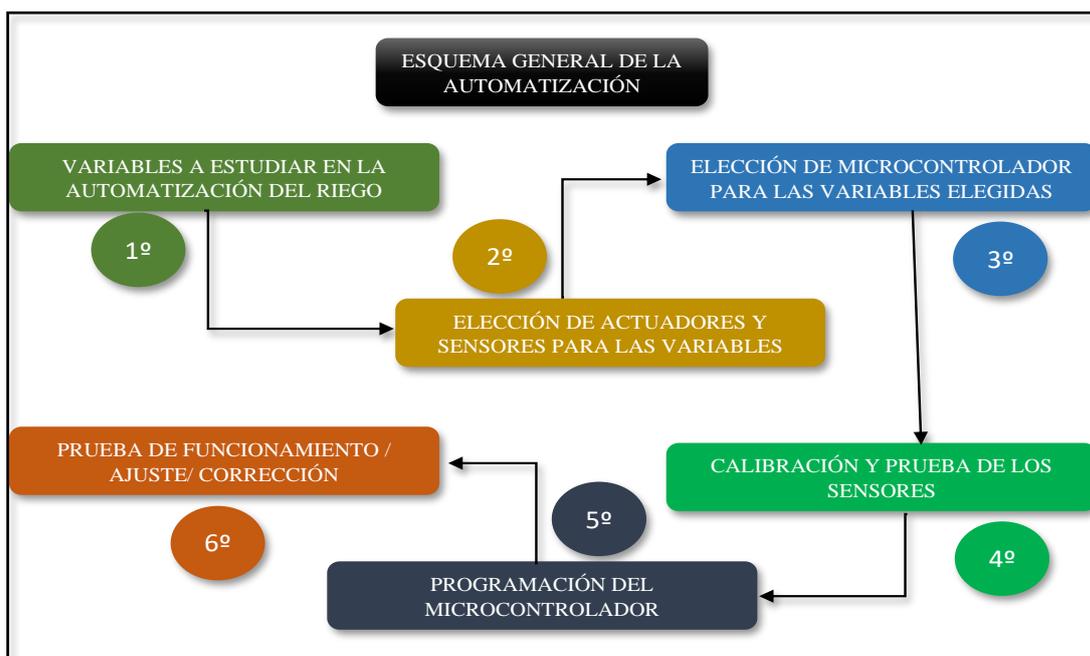


Figura 8: Esquema general de la automatización

## **Variables monitoreadas y dispositivos a programar**

- Cantidad de agua: Éste parámetro fue medido con un sensor de flujo o caudalímetro en l/min e instalado en cada arco de riego. Con este sensor se pudo estimar la cantidad de agua que están consumiendo cada uno de los grupos de cultivos propuestos.
- Humedad del suelo: Éste parámetro se midió con un sensor de humedad de suelo capacitivo en porcentaje (%) y conforme hubo presencia o no de humedad fue variando el porcentaje medido, su medición es fundamental para el encendido o apagado de las electroválvulas.
- Distancia: Éste se midió para controlar el nivel óptimo de agua en el tanque, asegurando que en todo momento esté encendido la bomba siempre que el nivel del agua sea el adecuado. La medición arrojó en centímetros con una precisión de  $\pm 3$  mm y un alcance de hasta 4.50 metros.
- Válvula solenoide: Se controló la apertura y cierre del flujo de agua de cada una de las electroválvulas en cada arco de riego del área experimental. Con esto aseguramos el paso del flujo de agua al área que lo necesita, según la medición que obtengamos del sensor de humedad de suelo.
- Bomba de agua: Se controló el encendido y apagado de la bomba, la cual impulsa el fluido aumentando la presión a través de la red del sistema, dependiendo de lo que iba marcando el sensor de humedad de suelo. Con esto aseguramos que el caudal llegue a través de los conductos de las tuberías hasta cada uno de los grupos de cultivos destinados.

## **Elección de actuadores y sensores**

- Sensor de humedad de suelo: Se utilizó sensor de tipo capacitivo cuya cubierta es anticorrosiva lo cual prolongará su vida útil, su compatibilidad con Arduino, su simpleza en la redacción del código y su fácil conexión. Además, poseen un valor intermedio (S/ 25-30) a comparación de los sensores resistivos (S/ 7-10) y sensores de mayor precisión como el sensor de humedad GS1 (S/ 650-750) los cuales son utilizados en proyectos de riego.
- Sensor de flujo: Se utilizó el modelo YF-S201, compatible con Arduino y de fácil conexión, el cual se encargó de medir el flujo de agua en L/min y la cantidad consumida.

- Sensor de ultra sonido: Se utilizó el modelo HS-SR04 que utiliza pines digitales, conexión fácil (4 pines) y es comercial. Con el que se midió el nivel de agua existente en el tanque en un rango de medición de 2 hasta 450 cm.
- Válvula solenoide: Se utilizó válvula de plástico, enroscado tipo macho de 12VDC, con un driver Relay compatible con Arduino, controlando el flujo y poniéndolo en estado abierto o cerrado dependiendo de lo que lea el microcontrolador Arduino.
- Bomba de agua: Se utilizó bomba tipo centrífuga, 800 L/h, 19 W y 5 mca, con lo cual se cubrieron los 240 L/h que se necesitaba en funcionamiento para el diseño de todo el sistema.

### Selección de micro controlador

- Para la lectura y funcionamiento del código digitado de cada uno de los sensores y actuadores elegidos, se trabajó con un modelo de Arduino capaz de integrarlos en uno solo de tal manera que la integración permitió una programación del sistema de riego coherente y fluida. En este caso se utilizó Arduino Leonardo en el cual se utilizaron 12 pines para señales digitales y 4 pines para señales analógicas, poniendo en funcionamiento 15 dispositivos, con lo cual los pines sobrantes fueron dos pines analógicos y dos pines digitales, aprovechando así la mayor cantidad de pines del Arduino.

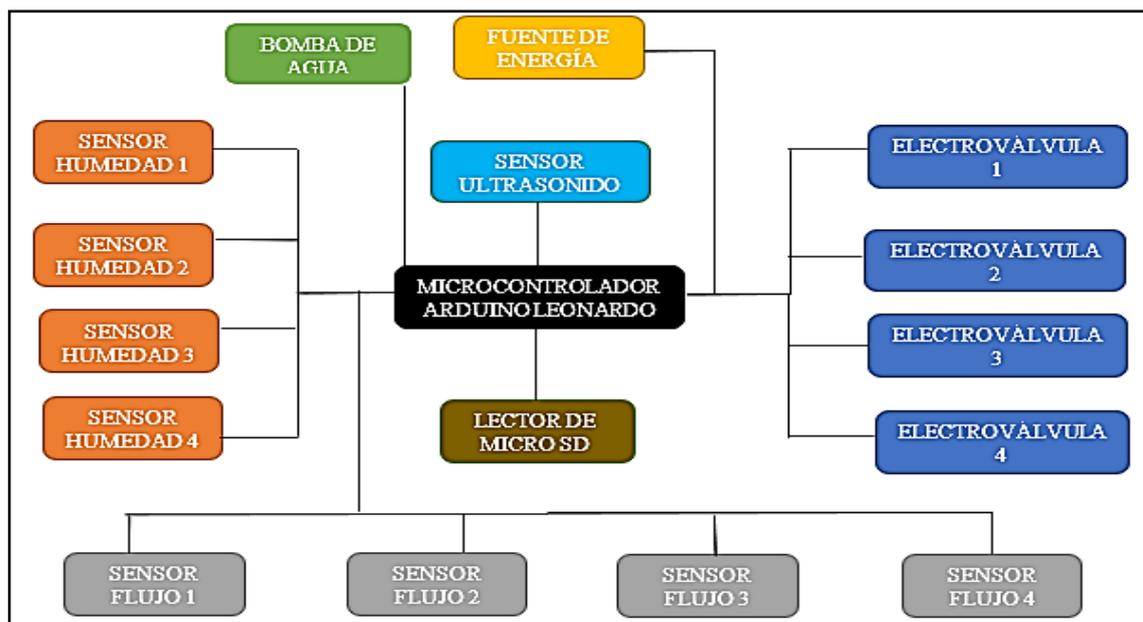


Figura 9: Resumen de sensores y actuadores controlados por Arduino

### 3.3.5. Calibración de sensores

#### Sensor de humedad

Más que una calibración lo que se realizó para la correcta medición de los datos son pruebas de medida rápida tanto para el valor máximo y como para el valor mínimo del sensor. El procedimiento que se siguió fue.

- Realizar las conexiones entre el sensor y el micro controlador según la Tabla 12.
- Se digitó el código de programación en el sketch, según el Anexo 23, para comenzar con las lecturas.
- Se tomó lecturas de el sensor al ambiente (humedad del suelo al aire), esto fue hasta lograr estabilizar el valor en el monitor serial.
- Luego se tomó lecturas sumergido al agua (humedad del suelo en el agua), hasta lograr estabilizar los valores en el monitor serial.
- Los valores encontrados fueron introducir en el código de programación descrito en el Anexo 24 y luego con eso se empezará a realizar la toma real.

#### Sensor de caudal

Para este sensor la calibración se resumió en obtener un Factor “K” (factor de conversión de frecuencia) propio de cada sensor, ya que existe una relación entre la frecuencia de los pulsos y el caudal. Con lo cual se pudo plantear la siguiente relación.

$$K = f(\text{Hz}) / Q(\text{L}/\text{min})$$

FUENTE: (Naylamp Mechatronics, 2021j)

Entonces se realizará el procedimiento para la calibración:

- Primero se conectó el sensor al micro controlador según la Tabla 13.
- Se digitó el código en el sketch de Arduino con el cual se realizaron medidas de la cantidad de pulsos por volumen circulado (Anexo 25).
- Se registraron los volúmenes y los pulsos de la prueba (los que se visualizaron en el monitor serial).
- Se realizaron 5 mediciones para dos tipos de volumen (0.8 litro y 1.6 litros)
- Se utilizó un recipiente graduado en mililitros

- Finalizado la toma de datos, se utilizó el promedio de los factores hallados por cada medición K. Este procedimiento se realizó para los demás sensores de flujo.

### 3.3.6. Programación en el microcontrolador

#### Software Arduino

Se realizó la programación de cada uno de los sensores, actuadores, módulo almacenamiento de datos y fuentes de alimentación siguiendo el esquema descrito en la Figura 10. La estructura del IDE está compuesta de una sección donde se realizó la declaratoria de variables, funciones, se adicionaron librerías al código y códigos globales tanto para sensores como para actuadores de tal manera que tenga un orden y se entienda la programación; en la sección (Void setup) donde se definió la forma de trabajar de los pines conectados si serán INPUT (pin como entrada de información) u OUTPUT (pin de salida de información), cabe mencionar que esta sección solo se ejecutó una vez en la programación; por último en la sección (Void loop) de ejecución continua, se repitió la función hasta que se desconecte la fuente de alimentación y se ejecutaron todos los sensores y actuadores según fueron programando desde el inicio hasta el final.

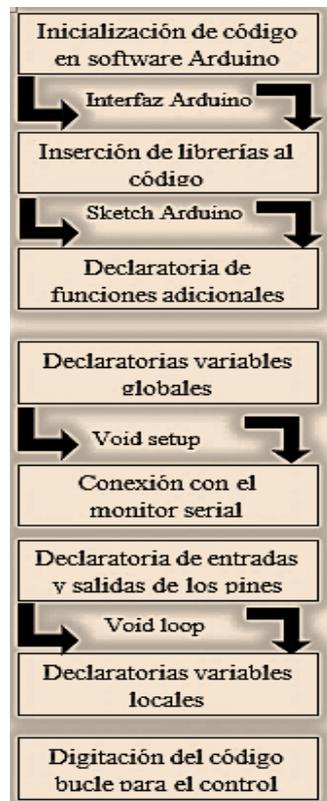


Figura 10: Procedimiento de escritura y lectura del código

## **Librerías y funciones a utilizar**

### **Librería SPI**

Esta librería permitió la comunicación de Arduino con dispositivos externos mediante el protocolo SPI. Esta comunicación en el hardware se realizó a través de los pines MOSI( que es la línea de envío de datos entre el Arduino y el dispositivo externo), MISO(línea de envío de datos entre el dispositivo externo y el Arduino) , SCK(línea de reloj) y pin SS( línea de selección del dispositivo a comandar)(Torrente, 2013).

### **Librería SD**

Esta librería permitió la comunicación entre el Arduino y la tarjeta SD mediante el protocolo SPI a través de los pines MOSI, MISO, SCK y SS. Son ideales para almacenar videos, imágenes y textos obtenidos de cualquier sensor(Torrente, 2013).

Cabe recalcar que para el uso de SS se puede designar cualquier pin que tenga esta funcionalidad el cual se tiene que configurar antes de su uso. Por otro lado, también se menciona que esta librería solo admite creaciones de ficheros con 8 caracteres como máximo y cada nombre creado de los ficheros deben de contener “(.TXT), (.txt) o (.tXt)” en su extensión(Torrente, 2013).

### **Función pinMode()**

Esta función sirvió para configurar el pin elegido ya sea para actuar como entrada o salida. La sintaxis de esta función es: pinMode(pin, modo) en la cual el pin puede ser cualquiera número y el modo puede ser INPUT, OUTPUT o INPUT\_PULLUP(Prometec, n.d.).

### **Función digitalWrite()**

Esta función activa o desactiva un pin digital con un valor alto (HIGH con un voltaje de 5V) o bajo (LOW con un voltaje de 0V (tierra)) sucesivamente. La sintaxis de esta función es :“digitalWrite” (pin, valor) donde se pone el número del pin digital y el valor es HIGH o LOW(Prometec, n.d.).

### **Función analogRead()**

Esta función se encargó de leer el valor analógico en el número de pin digitado. Esto significa que con esta función se puede leer rangos de voltajes entre 5V a 0V cuyos valores enteros

están entre 0 a 1023. La sintaxis de la función es: “analogRead(pin)” pines analógicos para el Arduino leonardo son cualquiera de los A0 hasta A12 en el pin digital 12(Prometec, n.d.).

### **Función millis ()**

Esta función devolvió el número en milisegundos (ms) transcurridos desde que la placa Arduino se empezó a ejecutar en el sketch actual. El número arrojado se reseteará pasado aproximadamente unos 50 días cuando el valor máximo “unsigned long” sea sobrepasado. La sintaxis de esta función es: millis () la cual no tiene parámetros(Torrente, 2013).

### **Función map ()**

Esta función modifica el valor de una escala en otra. Con esto se quiere decir que el primer parámetro que está en un rango (mínimo – máximo) pase en otro rango (mínimo- máximo) de la manera más proporcional posible. La sintaxis de esta función es: “map (valor, rango mínimo actual, rango máximo actual, rango mínimo deseado, rango máximo deseado).

### **Función attachInterrupt ()**

Esta función determino la rutina del programa que se ejecutó cuando se presentó interrupciones a través de los pines 3, 2, 0, 1 de Arduino leonardo. Esta función es muy importante para que se pueda automatizar, por ejemplo: activar Arduino solo cuando suceda un evento en específico(Prometec, n.d.).

Consejos para el uso de las IRS (rutinas de interrupción)(Prometec, n.d.):

a) No usan parámetros y no devuelven nada al usarlas; b) Deben ser cortas en su duración, c) Las funciones delay () y millis () no funcionaran correctamente se deberá utilizar Delay microseconds (), d) Las variables utilizadas deberán declararse como “volatile”.

La sintaxis es “attachInterrupt (interrupt, ISR, mode)” donde se “interrupt” es el número de interrupciones, tipo “int”, el IRS es la rutina que será llamada cuando ésta interrupción se active y el “mode” se define cuando se activa la interrupción (para este caso se utilizará el RISING(que activa la interrupción cuando el pin pasa de baja a alta (LOW >> HIGH))) (Prometec, n.d.).

### **Estructura, Variables y tipos de datos usados**

Para la programación se utilizaron: Estructuras de control que permitieron tomar decisiones y realizar repeticiones (if ( ): cuyo uso es de tipo condicional ya que se ejecutará la

instrucción siempre y cuando sea cierta la afirmación ; for() y while(): cuyos usos hacen que se repita el ciclo según las condiciones establecidas dentro de los paréntesis); operadores de comparación (== (igual que), >< ( mayor y menor), >= <= (mayor y menor o igual que )) ; operadores matemáticos ( +, -, / , \* ) y tipos de datos descritos en la Tabla 10 (void, int, long, float, unsigned long, unsigned char)(Hernández et al., 2011).

**Tabla 10: Tipos de variables utilizadas en programación**

Tipo	Descripción	Ejemplo
void	Reservado para la declaración de funciones sin valor de retorno	void setup () void loop ()
int	numero entero entre 32767 y - 32768 (16 bits)	int lucho= 28927
long	numero entero entre 2147483647 y - 2147483648 (32 bits)	long luis= 67876
float	número real con decimales entre 3.4028325E+38 y - 3.4028325E+38	float luchito= 3.56
unsigned long	numero natural positivo de 32 bits entre 0 y 4294967296	unsigned long luch=7866546
unsigned char	se utiliza para datos de 0 hasta 255 (1 byte)	unsigned char carmen = 36

FUENTE:(Prometec, n.d.)

### 3.3.7. Aplicativo ArduinoDroid

Este App se utilizó por sus similitudes con la versión original de Software Arduino, disponibilidad en lugares donde no se posea aparatos con el Software y su facilidad en la descarga, ya que es un aplicativo móvil para Android gratis y sin necesidad de conexión a internet para su funcionamiento.

Con este App se realizaron las pruebas en campo del funcionamiento del sistema automatizado, como la verificación y cargado del código, modificaciones en el sketch de código original (modificación en variables locales, cambios en el control de las válvulas solenoides, cambio en el control de los sensores de humedad y flujo, etc.) y pruebas de funcionamiento de los actuadores de cada área.

### 3.3.8. Recopilación y almacenamiento de datos

La información de las mediciones de los cuatro grupos (áreas experimentales) fueron recopiladas en una tarjeta micro SD de 16 GB, con ayuda del módulo lector de micro SD.

La información que se recaudó fue el caudal, la cantidad de agua consumida y el porcentaje de humedad de suelo que hay en función del estrato de saturación del mismo.

### **3.3.9. Encendido manual**

La automatización del sistema de riego depende en este caso del microcontrolador, dicho hardware tuvo un código digitado que sigue una línea de lectura dentro de su programación. Para no generar una dependencia única que perjudique el normal funcionamiento, el encendido o apagado se instalaron interruptores luminosos led manuales para cada actuador (electroválvula y bomba de agua). De tal manera que si en caso de avería del micro controlador o la programación, este siga funcionando con los interruptores (luz prendida/ luz apagada).

## **3.4. Fase de campo**

### **3.4.1. Instalación del sistema de riego**

Luego de la realización de los cálculos necesarios en gabinete, planteado el diseño definitivo y adquirido los materiales según el diseño. Se procedió a la instalación del sistema, primeramente, realizando las conexiones de la tubería principal y secundaria con sus conectores, seguidamente los arcos de riego con los bigotes de la porta lateral y se realizó el enterrado junto a las tuberías eléctricas de los sensores y actuadores. Concluido esto se realizó la colocación del porta goteros de las cuatro áreas y se empalmó la tubería principal con la bomba de agua y el tanque, por último, se colocaron los implementos para la captación de agua de la fuente hacia el tanque.

Cabe mencionar que el cableado de las instalaciones eléctricas fue enterrado dentro de tuberías de polietileno y junto con las tuberías de conexión de agua.

### **3.4.2. Conexión e instalación electrónica y eléctrica del sistema automatizado**

#### **Conexiones en el microcontrolador**

Se conectaron al microcontrolador los sensores de caudal, humedad del suelo, los actuadores como la electroválvula y la bomba y otros accesorios como los leds, módulo lector micro SD, Drivers y releés.

1. Sensor de humedad suelo: La conexión con el microcontrolador fue mediante cables jumper color negro, amarillo y rojo propios del sensor, para la conexión de la señal generalmente se utiliza el A0, pero en la práctica se pudo utilizar cualquier pin analógico,

para GNV y VCC del sensor se conectarán al GNV y VCC del Arduino tal como se describe en la Tabla 11.

**Tabla 11: Conexión humedad con puerto Arduino**

Sensor de humedad suelo capacitivo v1.2	Arduino leonardo
AOUT (cable amarillo)	A0*
VCC (cable rojo)	VCC
GND (cable negro)	GND

*Nota: \* pines analógicos desde A0 hasta A5*

2. Sensor de caudal: La conexión con el microcontrolador fue mediante cables jumper color negro, amarillo y rojo propios del sensor, para la conexión de la señal se utilizan pines digitales con interrupciones 3, 2, 0, 1, 7, para GNV y VCC del sensor se conectarán al GNV y VCC del Arduino como se especifica en la Tabla 12.

**Tabla 12: Conexión Flujo con puerto Arduino**

Sensor de flujo 1/2 YF-S201	Arduino Leonardo
cable amarillo	3*
cable rojo	VCC
cable negro	GND

*Nota\* para pines digitales 3,2,0,1,7*

3. Sensor de ultra sonido: La conexión con el microcontrolador esta descrita en la Tabla 13, para la conexión fue mediante cables jumper y para la señal se utilizó cualquier pin digital desde pin 0 hasta el pin 13, para GNV y VCC del sensor se conectarán al GNV y VCC de Arduino.

**Tabla 13: Conexión ultrasonido con puerto Arduino**

Sensor de ultra sonido HC-SR04	Arduino leonardo
TRIG	8*
ECHO	9*
GND	GND
VCC	VCC

*Nota: \*cualquier pin digital (0-13)*

4. Válvula solenoide: La conexión con el microcontrolador fue mediante un relay, en el cual se conectó un terminal de la válvula al puerto NO (normalmente abierto) y el otro polo de la válvula se conectó al puerto negativo de la fuente de energía de 12 V. El relay a su vez se conectó con el puerto COM (puerto común) al polo positivo de la fuente, se cerró la conexión conectando el puerto In a cualquier pin digital y GNV y VCC del relay se conectarán al GNV y VCC del Arduino como en la Tabla 14.

**Tabla 14: Conexión relay-válvula con puerto Arduino**

Válvula solenoide 1/2"	Modulo Relay	Fuente 12V	Arduino Leonardo
SC *	COM	polo +	SC*
cualquier polo	SC*	polo -	SC*
cualquier polo	NO	SC*	SC*
SC *	In1	SC*	5**
SC *	GND	SC*	GND
SC *	VCC	SC*	VCC

*Nota: \*sin conexión; \*\* cualquier pin digital*

5. Bomba de agua: La conexión con el microcontrolador fue mediante un drive, en el cual se conectó el terminal positivo y negativo de la bomba al puerto positivo y negativo del drive. El drive a su vez se conectó con la salida VIN al polo positivo de la fuente, cerrando la conexión entre el GND del Drive con el polo negativo de la fuente, por último se conectó la señal del drive a cualquier pin digital y GNV al GND del Arduino especificado en la Tabla 15.

**Tabla 15: Conexión relay-bomba con puerto Arduino**

Bomba 12V,800 L/h	Driver Mosfet	Fuente 12V	Arduino Leonardo
Polo negativo	V-	SC*	SC*
Polo positivo	V+	SC*	SC*
SC *	GND	Polo negativo	SC*
SC *	VIN	Polo positivo	SC*
SC *	SIG	SC*	6**
SC *	GND	SC*	GND
SC *	VCC	SC*	***NC

*Nota: \*Sin conexión; \*\* Cualquier pin digital\*\*\* No se conecta*

6. Módulo lector de tarjeta micro SD: La conexión con el microcontrolador fue mediante seis cables jumper de cualquier color, para la conexión CS de la señal se utilizó cualquier pin digital desde 0 hasta 13, para las otras conexiones se utilizaron cinco pines de la conexión ICSP excepto el pin nº5 reset, así como se menciona en la Tabla 16.

**Tabla 16: Conexión de pines micro SD con puerto Arduino**

Módulo lector de micro SD	Arduino Leonardo
CS	4*
SCK	3-ICSP
MOSI	4-ICSP
MISO	1-ICSP
VCC	2-ICSP
GND	6-ICSP

*Nota: \* cualquier pin digital*

### **Instalación del sistema automatizado**

Se realizaron las conexiones respectivas entre microcontrolador Arduino con los sensores, actuadores, leds y lector de micro SD. Se utilizó cablecillos multifilares a los que se adaptó los conectores tanto hembra como macho dependiendo la conexión microcontrolador-actuador / sensor, también se empalmaron cables para lograr la conexión. Todos estos aparatos fueron ubicados en un área de control, accesible, segura y protegida contra agentes externos y ambientales.

### **Instalación del cableado en el área experimental**

Se contabilizó el consumo de la intensidad de corriente del proyecto, en función de los resultados se compró el cable más adecuado para el tendido eléctrico, se instaló una llave termomagnética independiente para proteger los aparatos que se utilizaron y se enterraron a 30 cm de profundidad las tuberías de agua junto con las tuberías del cableado de los sensores y actuadores.

### **Implementación y delimitación del terreno**

Previo a la conclusión de la instalación de las tuberías, implementos de riego, automatización y antes de la realización de las pruebas del funcionamiento y uniformidad del sistema de riego, se cercó toda el área de estudio con troncos de madera, malla reshell y alambre

galvanizado con la finalidad de generar las condiciones que aíslen al área de estudio de posibles agentes externos que podrían dañar o alterar los resultados a obtener. Por otro lado, la implementación de ésta modesta estructura generaría una disminución de la velocidad del viento evitando que ésta pueda dañar las plántulas en sus inicios de desarrollo que es donde son más susceptibles al daño, también se logra una disminución de la radiación directa y generará un microclima dentro del área cercada favoreciendo a los cultivos.

### 3.4.3. Prueba de coeficiente de uniformidad (Cu)

Esta prueba se realizó con la finalidad de obtener una distribución adecuada del agua en cada planta, evitando de esta manera errores futuros de goteros secos o con excesiva dosis. Para su cálculo, se seleccionó emisores que han sido distribuidos uniformemente dentro de la subunidad de riego de estudio (para este caso tomaremos el área extrema más lejana a la fuente de agua del proyecto). Seguidamente, según Merriam y Keller (1978) recomendaron seleccionar 16 emisores para calcular el coeficiente (Pizarro, 1996). En la Figura 11 se aprecia la distribución de los emisores dentro de la sub unidad de estudio y para realizar el procedimiento se siguieron los siguientes pasos: escoger el primer lateral cercano al inicio de la porta laterales y el último lateral, escoger las laterales que ocupa el tercio y las dos terceras partes del total de las laterales; para los emisores se escogió de igual manera, el primero, el último emisor y el emisor que ocupa el tercio y dos terceras partes del total de emisores.

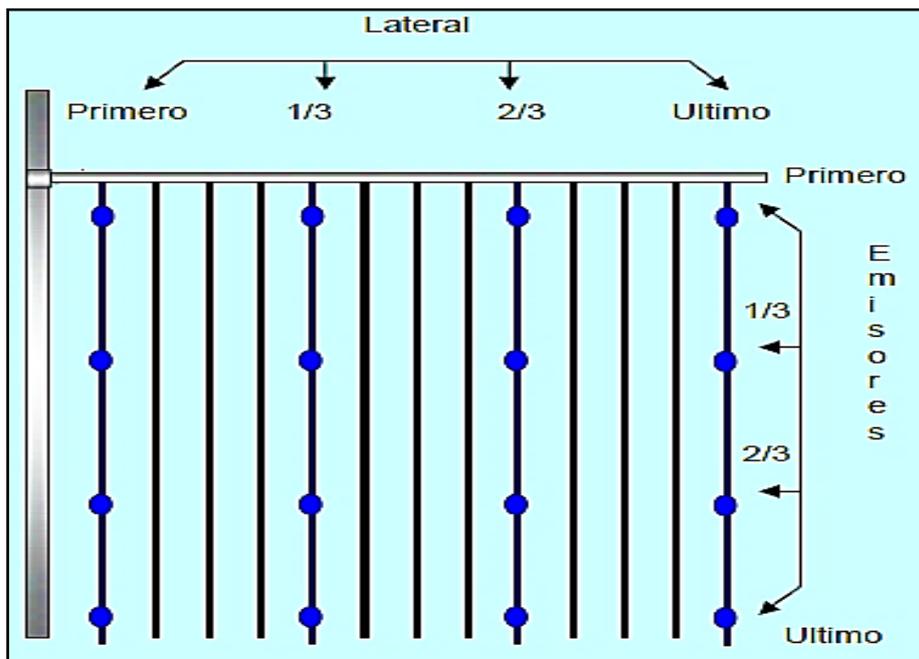


Figura 11: Localización de goteros para prueba de uniformidad

FUENTE:(Mendoza, 2013)

Una vez seleccionados y ubicados los emisores se procede a registrar mediante un recipiente graduado los caudales en cada uno de ellos. Según Mendoza (2013) los caudales se registrarán con un tiempo entre 3 a 5 minutos para goteros. Los resultados de los 16 goteros se registraron y con ellos se calculó el  $q_m$  que fue el caudal promedio de las mediciones hechas, también se calculó  $q_{25}$  que es el promedio de caudal del 25% de caudales más bajos registrados en los emisores(Mendoza, 2013). Para luego clasificarlos según la Tabla 17.

$$Cu = q_{25} / q_m$$

**Tabla 17: Clasificación por porcentaje de CU**

CUC	Calificación
90% - 100%	Excelente
80% - 90%	Buena
70% - 80%	Aceptable
<70%	Inaceptable

FUENTE: (Mendoza, 2013)

#### **3.4.4. Rotación de cultivos**

Para evitar el mono cultivo en las áreas trabajadas se realizó un intercambio cruzado entre los cultivos de un grupo con los cultivos de otro (Figura 14), esto se planteó ya que entre los vegetales agrupados se tuvieron cultivos de diferentes familias, tamaños y periodos. Esta asociación y su posterior rotación se realizarán con la finalidad de generar una mayor diversidad en cada área destinada y que en cada rotación se tuviera para el siguiente periodo vegetativo un cultivo de otras características y diferente familia, propiciando la permanencia de la variabilidad entre los cultivos de un grupo.

La rotación se realizó entre los grupos 1 y 4 y los grupos 2 y 3 de manera cíclica en el periodo anual. Se inició la propuesta con la fecha de inicio de la siembra para el mes de julio, tomando en cuenta el calendario de cultivo descrito en la Figura 12, donde se describe los cultivos por cada grupo, la cantidad de veces que se cultivarán en un periodo de un año, el inicio de cada cultivo por periodo y el tiempo de maduración hasta el tiempo estimado para su cosecha.

Grupo	Cultivo	Época de siembra	Siembra-maduración												T	Cosecha		
			Verano		Otoño			Invierno			Primavera							
			E	F	M	A	M	J	J	A	S	O	N	D			Días	
2	Ají	P / V	2	2	2	2							1	1	1	1	120	50
4	Apio	V/O/I	2	2	3	3	3			1	1	1				2	90	7
2,4	Betarraga	TA	3	3	4	4	5	5	6	6	1	1	2	2			60	30
1,2	Brócoli	TA	3	3	4	4	5	5	6	6	1	1	2	2			60	30
1,2	Col/repollo	TA	2	2	3	3	3	4	4	4	1	1	1	2			90	30
1,3	Espinaca	O / I			3	4	5			1	2						40	todo a la vez
1,3	Lechuga	O / I / P			4	4	5	5	1	1	2	2	3	3			70	25
3,4	Tomate	TA	2	2	3	3	3	3	1	1	1	1	2	2			120	30

*Nota: TA: todo el año, P: primavera, V: verano, I: invierno, O: otoño; T: tiempo total de maduración; 1,2,3,4,5,6: n° de veces cultivadas*

**Figura 12: Calendario anual de cultivo**

Para el caso del grupo 1, se sembró en julio espinaca y lechuga con un periodo de cultivo de 40 y 70 días respectivamente; para el grupo 4 se sembró tomate y apio con periodos más largos de 90 y 120 días. Esto implicó que para agosto el cultivo de espinaca (grupo 1) sea el primero en rotar y ser sembrado en el grupo contrario, quedando el grupo 4 para este mes con cultivos de tomate, apio y espinaca. Para setiembre en el grupo 1 se sembró brócoli, col y la lechuga que había en este grupo se cosechó y se volvió a sembrar en el grupo 4, quedando en él, apio, tomate, betarraga y lechuga. Para los meses siguientes se procedió de la misma manera, según lo descrito en la Figura 13, siguiendo la metodología de que la cosecha se realizará una vez finalizado su periodo de maduración y el cultivo que finaliza su periodo en uno, inicia su siembra en el otro, siempre y cuando se respete la época de siembra y los meses planteado en el calendario de la Figura 12.

De igual manera se procedió con la rotación de los cultivos del grupo 2, en la cual se planteó su inicio el mes de setiembre de tal manera que coincidan el inicio de siembra de los cultivos de ají, betarraga, brócoli y col. Con lo cual para este grupo se tuvo toda el área cubierta de vegetación y conforme fue avanzando de la siembra a la maduración se fue cosechando y rotando con los cultivos del grupo 3. Como pasa en el caso del brócoli que su primera cosecha se dió a fines el mes de octubre y pasará a siembra en el mes de noviembre, pero en el grupo 3.

Para el grupo 3 se dió inicio a la siembra de cultivos en el mes de julio con la espinaca, lechuga y tomate con lo cual para el mes siguiente la espinaca será cultivada en el grupo 2 por un periodo promedio de 40 días con lo que en la primera semana de setiembre solo quedó el tomate (cuya cosecha se realizará en octubre) en este grupo ya que la lechuga fue vuelta a cultivar a partir de noviembre. Entonces para los siguientes meses se siguió realizando este mismo procedimiento descrito en la Figura 13 siempre respetando los periodos de siembra y maduración, cultivo debe rotar o en su defecto debe ser vuelto a cultivar en el mismo lugar en un periodo mínimo de 1 mes y que la cosecha se realizó en un solo día.

Cabe mencionar que los cultivos seleccionados son cultivados y producidos en la zona de Lurín, así como lo menciona Ugás et al., (2000) en el manual hortalizas: datos básicos. Por otro lado, estas hortalizas tienen un coeficiente de cultivo parecido en etapa de maduración (Tabla 9), también están en un suelo suelto, permeable y el tiempo de riego está basado en una programación automatizada que dependerá del sensor de humedad el cual tiene un déficit del 50 % con lo cual se asegura que no haya sobresaturación en la superficie de los cultivos.

Época/ mes	Verano		Otoño			Invierno			Primavera			
	E	F	M	A	M	J	J	A	S	O	N	D
Grupo 1	AP, TOM, BR	AP, TOM, BR	L, B, E, C	L, B, C	BR, E, C	BR	E, L	L	BR, C	BR, C	C, TOM, L, B	L, B, AP, TOM
Grupo 4	B, C	B, C	BR, TOM, AP	BR, TOM, AP, E	TOM, L, AP, B	L, TOM, B, C	TOM, AP	TOM, AP, E	B, TOM, AP, L	B, TOM, L	BR	BR, C
Grupo 2	TOM, B, BR	TOM, B, BR	C, L	C, E, L	C, B, BR	B, BR		E	BR, B, C, AJ	BR, B, C, AJ	AJ, C, TOM	AJ, TOM
Grupo 3	C, AJ	C, AJ	TOM, AJ, E, B	TOM, AJ, B	TOM, E	TOM, C	E, L, TOM	L, TOM	TOM	TOM	L, B, BR	C, L, B, BR

*Nota: AP: apio, TOM: tomate, BR: brócoli, L: lechuga, E: Espinaca, B: betarraga, C: col, AJ: ají*

**Figura 13: Rotación de los cultivos por época de siembra entre los grupos**

### **3.4.5. Operatividad y manejo del sistema de riego**

Para este trabajo se destinaron funcionamiento manual y automático. Con respecto al funcionamiento manual se compuso de tres principales pasos, con un conocimiento medio para su entendimiento y ejecución: a) conexiones fuente de agua-goteros, b) conexión cableado fuente de energía-actuadores c) control y monitoreo; para el funcionamiento automatizado debido al elevado grado de conocimientos que se necesita con respecto a la programación se compuso de: a) conexiones de tuberías, b) conexiones de sensores y actuadores, c) programación, d) control y monitoreo

### **3.4.6. Presupuesto, costo unitario y metrado de la instalación**

El proyecto tuvo cinco principales componentes de gastos, la primera es el sistema de riego tecnificado, el segundo es el sistema de automatización, el tercero sistema eléctrico, los servicios y las herramientas, mano de obra e insumos adicionales. En este proyecto se planteó una propuesta de ahorro en cuanto a materiales (diseño hidráulico) y automatización básica, haciendo a esta propuesta atractiva para ser replicada en zonas aledañas.

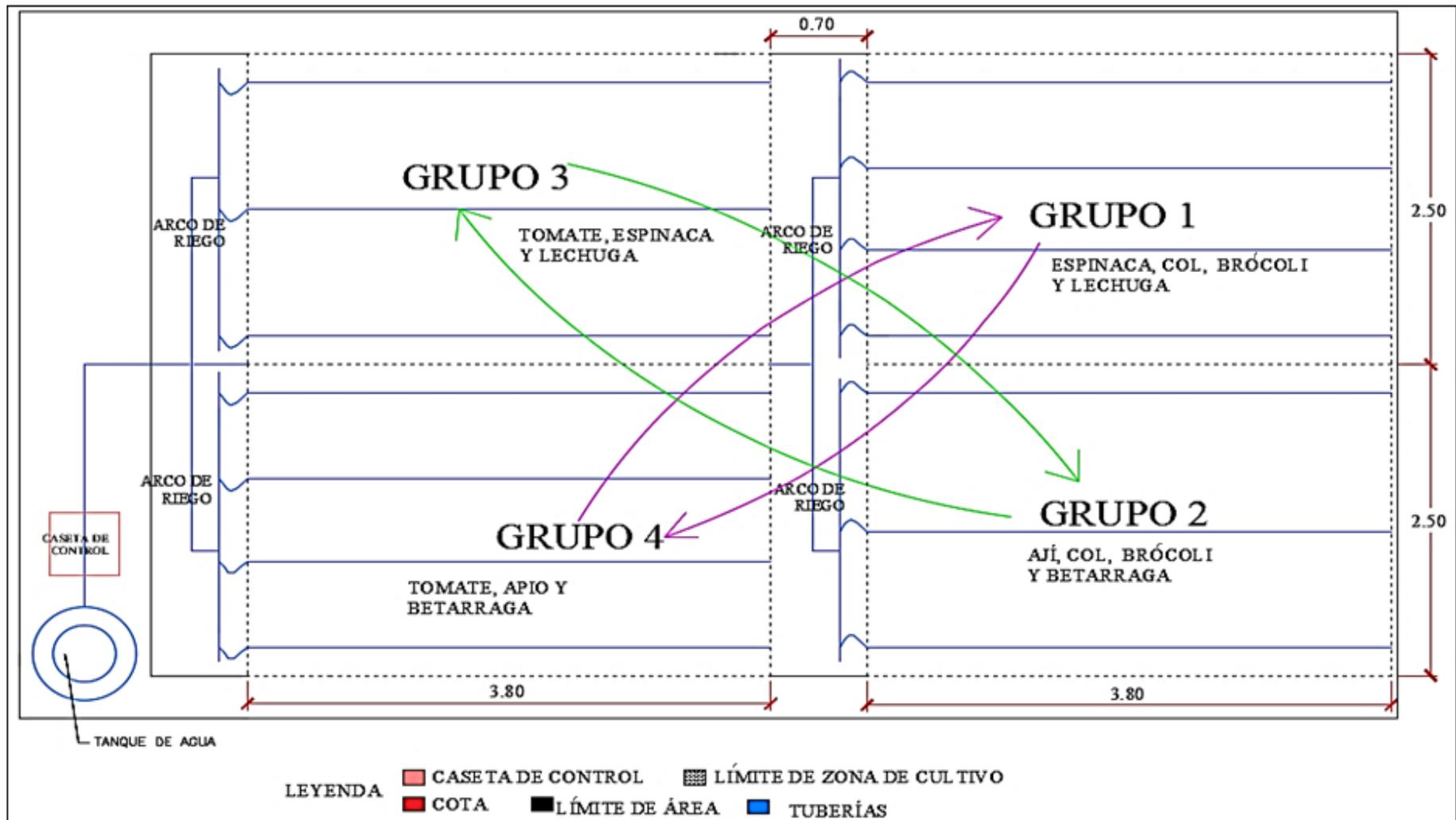


Figura 14: Grupos de rotación de los cultivos propuestos

## IV. RESULTADOS Y DISCUSIÓN

### 4.1. Diseño del sistema de riego por goteo

#### 4.1.1. Diseño agronómico

##### Evapotranspiración de referencia (ET<sub>o</sub>)

Para este diseño se tomó el valor de la evapotranspiración para el periodo más crítico, el cual se encuentra en el mes abril, con una ET<sub>o</sub> de 3.96 mm/día. Este dato es tomado de un promedio diario por mes en un intervalo de 10 años, el cual según la Organización Mundial de Meteorología (2018) se necesita un mínimo de 10 años de observación y recaudación de datos diarios para la elaboración de éste entre otros parámetros de referencia.

##### Coefficiente de cultivo

Para diseño agronómico se utilizó el coeficiente de cultivo referencial tomado de bibliografías como la FAO, manuales de cultivos y riego tecnificado. Donde, de manera controlada para distintos parámetros y condiciones lograron hallar el K<sub>c</sub> de los cultivos.

**Tabla 18: Coeficiente de cultivo de cultivos propuestos**

Hortaliza	Kc de Cultivos		Profundidad de Raíces
	Periodo		
	Maduración	Cosecha	Rango
Beterraga	1	0.9	0.6-1.0
Ají**	1.05	1.01	0.5-1.0
Espinaca	0.95	0.9	0.3-0.5
Tomate	1.05	0.6	0.3-0.6
Brócoli	1.05	0.95*	-
Col/Repollo	1.05	0.95*	-
Lechuga	0.95	0.9	0.3-0.5
Apio	1	0.7	0.3-0.5

FUENTE:(Vásquez et al., 2017);\*

(Estudio FAO Riego y Drenaje 56, 2006);\*\*(Quispe et al., 2016).

Por ser una agricultura familiar en la cual cultivarán diversos vegetales y frutas, se tomaron para el cálculo de la máxima necesidad hídrica, el K<sub>c</sub> para el periodo y el cultivo más

crítico. Para este caso se tomó como referencia el Kc del cultivo de Brócoli, siendo este 1.05 (resultando lo mismo para Kc del Ají, Tomate o Col) de tal manera que con esto aseguramos una dotación de agua mínima para el cultivo que requiera más y reducimos algún estrés hídrico a futuro.

**Necesidades totales de lámina de riego (Nt)**

Esta lámina se calculó siguiendo el procedimiento dado por Pizarro (1996) en su manual de RLAF (riego localizado de alta frecuencia) para el cálculo de las necesidades.

1. Calculamos la ETc:

$$ETc = ETo * Kc = 3.96 \text{ mm/día} * 1.05 = 4.16 \text{ mm/día}$$

2. Calculamos ETrl (Evapotranspiración corregida según condiciones locales):

$$ETrl = ETc * KL$$

Previamente se halla KL: coeficiente de localización, el cual hallamos con el promedio de los cálculos siguientes.

$$\text{Decroix} \dots \dots \dots KL = 0.1 + A = 0.1 + (\pi * (D)^2) / (4 * (Ep * Es))$$

$$\text{Decroix} \dots \dots \dots KL = 0.1 + (\pi * 0.2^2) / (4 * (0.2 * 0.3)) = 0.62$$

Con Ep=0.2, Es= 0.3, D = 0.2, que son datos de para el cultivo de espinaca, asumiendo que el espacio que dejará será menor a 0.2 m

$$\text{Hoare} \dots \dots \dots KL = A + 0.5 * (1 - A) = A + 0.5 - 0.5 * A = 0.5 + 0.5 * A$$

$$\text{Hoare} \dots \dots \dots KL = 0.5 + 0.5 * (\pi * 0.2^2) / (4 * (0.2 * 0.3)) = 1.02$$

$$KLp \text{ (promedio)} = (0.62 + 1.02) / 2 = 0.82$$

Entonces:  $ETrl = ETc * KLp = 4.16 \text{ mm/día} * 0.82 = 3.41 \text{ mm/día}$

Seguidamente se realiza la corrección por condiciones locales

$$ETrl = Nn = 3.41 \text{ mm/día} * 1.2 = 4.1 \text{ mm/día}$$

Se concluyó para esta parte que Nn (necesidades netas) son 4.1 mm/día

3. Calculamos K en función de la Ea (eficiencia de aplicación)

$$K = 1 - Ea = K = 1 - 0.90 = 0.10$$

La eficiencia para riego por goteo es de 90 %, según la Tabla 19 para profundidades menores de 0.75 y textura arenosa.

**Tabla 19: Eficiencia de riego según textura y profundidad**

Profundidad de raíces (m)	Muy porosa (grava)	Textura		
		Arenosa	Media	Fina
< 0.75	0.85	0.90	0.95	0.95
0.75 - 1.50	0.90	0.90	0.95	1.00
> 1.50	0.95	0.95	1.00	1.00

FUENTE:(Pizarro, 1996)

4. Calculamos el  $N_t = N_n / ((1-K) * C_u) = 4.1 \text{ mm/día} / ((1-0.1)*0.85) = 5.35 \text{ mm/día}$

Tomando como referencia el  $C_u$  de la Tabla 20 para pendientes menores de 2 % y cultivos espaciados a menos de 2.5m.

5. Luego se calcula  $Q$  en L/min, si  $1\text{mm} = 1\text{L/m}^2$  entonces se tendrá que  $N_t = 5.35 \text{ L/ m}^2*\text{día}$ , para un área de cultivo de  $38\text{m}^2$  se tendrá:

$$Q = (5.35 \text{ L/ m}^2 * \text{día}) * 38\text{m}^2 = 203.3 \text{ L/día} = 8.5 \text{ L/hora} = 0.14 \text{ L/min}$$

**Tabla 20: Coeficiente de uniformidad según separación, pendiente y clima**

Emisores	pendiente (i)	CU	
		Clima árido	Clima húmedo
Emisores espaciados más de 4 m en cultivos permanentes	Uniforme (i < 2 %)	0.90 - 0.95	0.80 - 0.85
	Uniforme (i > 2 %) u ondulada	0.85 - 0.90	0.75 - 0.80
Emisores espaciados menos de 2.5 m en cultivos permanentes o semipermanentes	Uniforme (i < 2 %)	0.85 - 0.90	0.75 - 0.80
	Uniforme (i > 2 %) u ondulada	0.80 - 0.90	0.70 - 0.80
Mangueras o cintas de exudación en cultivos anuales	Uniforme (i < 2 %)	0.80 - 0.90	0.70 - 0.80
	Uniforme (i > 2 %) u ondulada	0.70 - 0.85	0.65 - 0.75

FUENTE:(Pizarro, 1996)

Una vez culminado los pasos anteriores se distribuyeron los goteros buscando la uniformidad del riego, donde el espaciamiento óptimo entre goteros fue del 80% del diámetro de bulbo húmedo(Mendoza, 2013). Para hallar el bulbo húmedo utilizamos la fórmula de(Sepúlveda & Zazueta, 2004).

6. Finalmente hallamos el diámetro del bulbo húmedo

$$D = 0.01 * (q * r / I)^{1/3}$$

Con un caudal de gotero q: de 2Lt/h; r: profundidad de raíz de 0.4 m en promedio; I: tasa de infiltración básica de = 30 mm/hora dato extraído del Anexo 2. Tomando en cuenta la caracterización física del suelo franco arenoso, con alta permeabilidad y el cual con la adición de compost y tierra de chacra se logró asimilar a suelo franco.

Entonces se tiene:

$$D = 0.01 * (2 * 0.4 * 3600 / 30 * 10^{-3})^{1/3} = 0.46 \text{ m de diámetro}$$

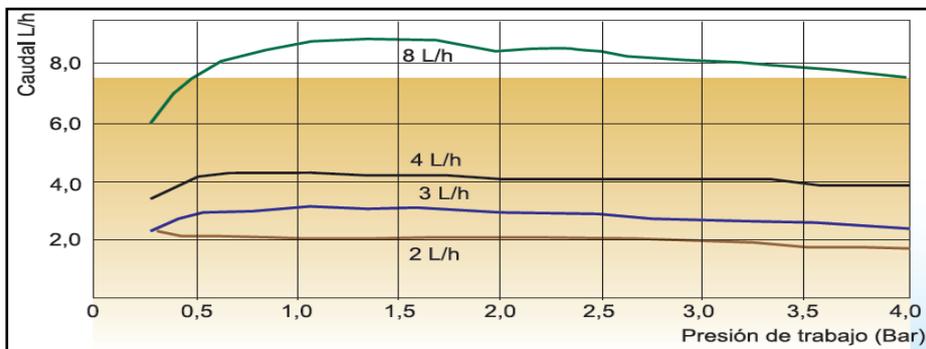
En conclusión, el espaciamiento entre los goteros fue el 80% de D

Espaciamiento entre goteros =  $36.8 = 0.36 \text{ m}$

Con esto se logró un traslape mínimo de 10 cm lo cual beneficiará a los cultivos que estén ubicados entre los goteros.

### Elección de los goteros

Para este caso se optó por el gotero de botón MicroFlapper, el cual es un emisor de bajo caudal, compensador de presión y auto limpiado, está hecho de silicona y tiene un amplio rango de presión (0.6 bar a 4.0 bar) como se visualiza en la Figura 15, para el diseño se utilizó un gotero de 2 L/h color marrón.



**Figura 15: Curva de descarga**

FUENTE:(ORBES AGRÍCOLA SAC, 2020)

Tomando en cuenta el precio que es de 0.45 soles y ventaja de ser de autolavado asegurando que no sea fácil de obturar, en comparación con otros goteros con precios desde 0.6 -0.8 soles y de no contar con autolavado. Resulta más conveniente adquirir un gotero MicroFlapper.

### Frecuencia y tiempo de riego

Para este caso se obvió la fórmula convencional en función de la infiltración básica, la lámina neta y evapotranspiración de cultivo cuya relación es:

Frecuencia riego = lámina neta/ ETr y Tiempo riego = lámina neta/ infiltración básica

Pero al no ser éste un diseño de un riego tecnificado convencional no se tomó en cuenta dichas relaciones.

Entonces para encontrar estos valores se procedió de la siguiente manera:

Si las necesidades totales de lámina de riego arrojaron 5.35 mm de agua por día y sabiendo que 1 mm = 1 L/m<sup>2</sup>. Entonces se tendrá 5.35 L en un metro cuadrado de área por día, el planteamiento inicial era de un área de influencia de 0.3\*0.2m = 0.06 m<sup>2</sup>, con lo cual ajustando se tuvo una nueva lámina de riego por área de influencia (Ntn)

$$Ntn = 5.35 \text{ L/m}^2 * 0.06 \text{ m}^2 = 0.32 \text{ L}$$

Entonces se tuvieron 0.32 L en un área de 0.06 m<sup>2</sup> por día, si se eligió el gotero microflapper de 2 L/ h, con lo que se necesitó de unos minutos de riego de éste gotero para retribuir al suelo el volumen de agua de la necesidad calculada.

$$\text{Tiempo de riego} = 9.6 \text{ min} = 10 \text{ min aproximadamente}$$

Cabe precisar que arrojó este dato debido a que se consideró las necesidades del cultivo, pero para este caso se utilizó una bomba de agua que brindó 13L/min (0.22 L/s) a una potencia de 19W con una carga calculada de 8 mca aprox, sabiendo que el sistema necesita que fluya 4.67 L/min (0.077 L/s) por la tubería y para que cada gotero pueda dosificar según sus especificaciones un total de 2L/h, con lo cual al tener un menor caudal solicitado la presión aumentará haciendo que los goteros emitan más agua en menos tiempo. Es así que, para éste caso con un riego de 5 min, periódicamente por un intervalo de 4 horas en las mañanas y diariamente fue lo adecuado.

#### **4.1.2. Diseño hidráulico**

Para el diseño hidráulico primero se calculó los diámetros de las tuberías y seguidamente las pérdidas por fricción en función del material, el caudal que fluye y la longitud de las tuberías. Los cálculos se realizaron desde la emisión (el gotero) hasta la fuente (salida de agua de la mini bomba centrífuga) en el orden mencionado.

#### **Tubería porta emisores de polietileno**

Se realizó un cálculo rápido para hallar el diámetro económico mediante Bresse:

$$D = 1.5 Q^{0.5} = 1.5 * (5.56 * 10^{-6})^{0.5} = 3.53 * 10^{-3} \text{ m} = 3.53 \text{ mm}$$

Para  $Q = 0.02 \text{ m}^3/\text{h} = 5.56 \cdot 10^{-6} \text{ m}^3/\text{s}$  del caudal del porta gotero. Con lo que tuvimos que recurrir a otra medida comercial con un diámetro mayor.

Entonces el diámetro comercial encontrado fue de 16 mm para tubería de polietileno. (nota: diámetro de 12 mm sería el más adecuado, pero es menos comercial).

### **Pérdida por fricción en la tubería (Hf)**

Para un C (coeficiente propio del material) =150, material plástico; longitud de 4 m; un coeficiente de Christiansen de 0.402 (10 emisores)(Mendoza, 2013). Se calculó la pérdida de carga para éste tramo 5-4 de la Figura 16, mediante la fórmula de Hazen-Williams.

$$J = 1.131 \cdot 10^9 \cdot (Q/C)^{1.852} \cdot (1/D)^{4.87}$$

$$J = 1.131 \cdot 10^9 \cdot (0.02/150)^{1.852} \cdot (1/16)^{4.87}$$

$$J = 1.03 \cdot 10^{-4} \text{ m/m}$$

$$H_f = J(\text{m/m}) \cdot L = 1.32 \cdot 10^{-4} \cdot 4.0 = 5.28 \cdot 10^{-4} \text{ m}$$

Valor obtenido por cada metro de longitud, entonces para nuestra longitud dada y con el coeficiente incluido, el Hft será:

$$H_{ft} = J (\text{m/m}) \cdot L \cdot f = 1.32 \cdot 10^{-4} \cdot 4.0 \cdot 0.402 = 1.03 \cdot 10^{-4} \text{ m} = 0.0002 \text{ m}$$

Con el resultado hallado se puede concluir que es casi nulo la pérdida en el porta gotero.

Para los siguientes tramos del cálculo hidráulico se elaboró la Tabla 21 en la cual se resume los cálculos para los tramos 4-3,3-2,2-1 del recorrido más crítico del sistema ilustrado en la Figura 16.

**Tabla 21: Resumen de cálculos hidráulicos según necesidades (por tramo)**

CÁLCULOS HIDRÁULICOS								
Tramo	C (Plástico)	D(mm)	Q(m <sup>3</sup> /h)	L(m)	J(m/m)	f(n)	Hft(m)	V(m/s)
5-4	150	16	0.020	4	0.0001	0.402	0.0002	0.028
4-3	150	19.05	0.08	2.5	0.0005	0.486	0.001	0.078
3-2	150	12.7	0.08	2.5	0.004	1	0.01	0.175
2-1	150	12.7	0.28	7	0.042	1	0.29	0.614
Total, pérdidas Hf(m)							0.31	

La Tabla 21 está en función a las necesidades del cultivo tomado, en el cual el caudal para cada cultivo fue suministrado por un gotero de 2 L/h y a su vez éste llegó por la lateral que conduce un caudal de 20 L/h (0.02 m<sup>3</sup>/h).

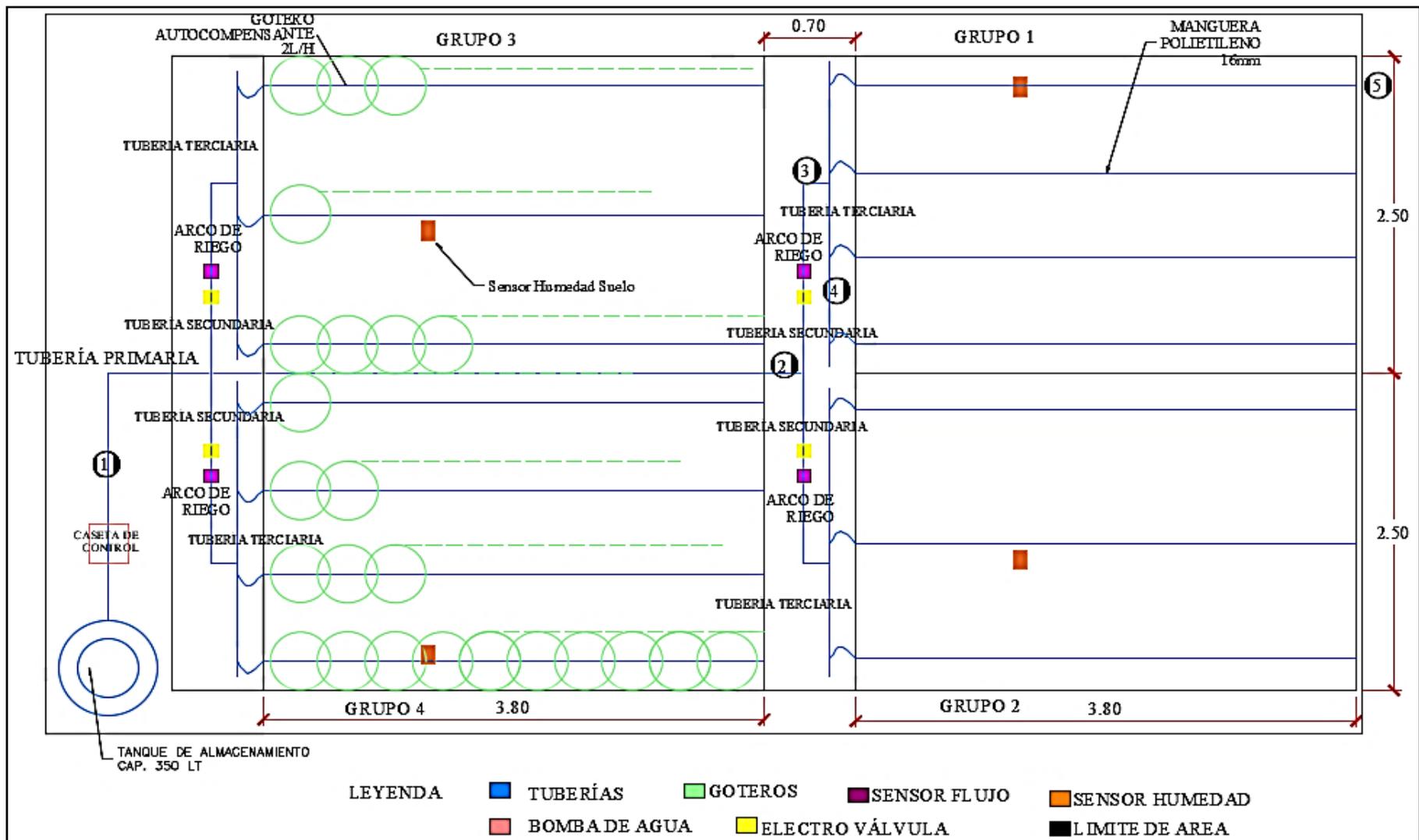


Figura 16: Distribución de los implementos de la automatización y riego

### **Presión a la entrada de la lateral (PEL)**

Para éste cálculo se tuvo la relación

$$PEL = P_g + \frac{3}{4} H_f + AE$$

Donde:  $P_g = 0.6 \text{ bar} = 6.12 \text{ m}$  (aproximadamente  $1 \text{ bar} = 10.2 \text{ mca}$ ), presión mínima de trabajo del gotero, tomada como referencia de la Figura 31;  $H_f = 0.0002 \text{ m}$ , para lo cual se concluyó que era casi nula (tramo 5-4);  $AE = 0.3 \text{ m}$  por debajo del nivel del suelo. Con lo cual se planteó que:

$$PEL = 6.11 + \frac{3}{4} * (0.0002) + 0.3 = 6.42 \text{ m}$$

### **Presión en la entrada de la porta lateral PPL**

Para éste cálculo se tuvo la relación:

$$PPL = PEL + \frac{3}{4} H_f$$

donde  $H_f = 0.001 \text{ m}$  que fue la pérdida sufrida en la tubería de la porta lateral (tramo 4-3), entonces se tiene:

$$PPL = 6.42 + \frac{3}{4} * 0.001 = 6.427 \text{ m}$$

### **Carga dinámica total (CDT)**

Para este cálculo se tuvo la relación:

$$CDT = PEL + PPC + \Delta SD + PS$$

donde  $PPC = 0.31 \text{ m}$ , pérdida de la tubería principal + pérdida de la tubería de conducción (tramo 3-1);  $\Delta SD = 0$ , no hay presencia de desnivel entre la succión y descarga;

$PS = 0.15 * (0.31 \text{ (suma de pérdidas de todo el tramo crítico (5-1))})$  (Mendoza, 2013).

$$CDT = 6.43 + 0.31 + 0 + 0.05 = 6.79 \text{ m}$$

### **Potencia de la bomba (HPb)**

Finalmente, para éste cálculo se llegó a la relación:

$$HPb = (Q * CTD) / (75 * E_f)$$

donde  $Q = 0.28 \text{ m}^3/\text{h} = 0.0778 \text{ L/s}$ , caudal que recorrió el sistema (según las necesidades de los goteros) y hallado del supuesto que todos los goteros están emitiendo agua en un mismo instante;  $E_f = 0.80$ , eficiencia tomada entre el rango  $0 - 1$  (Mendoza, 2013).

$$HPb = (0.0778 * 6.79) / (75 * 0.8) = 0.0088 \text{ HP}$$

sabiendo que  $1\text{HP} = 745.7\text{W}$  entonces se tuvo que utilizar una bomba

$$\text{HPb} = 0.0088 * 745.7 = 6.56 \text{ W}$$

Como resultado se eligió una bomba compatible con Arduino cuya potencia nominal indica 19W con lo cual quedan cubiertas las necesidades del sistema.

Cabe mencionar que para las necesidades del proyecto no se optó por otro tipo de bomba, ya que en el mercado otras bombas de corriente continua (AC) compatibles con el software Arduino en su mayoría sirven para proyectos pequeños, para peceras o proyectos escolares de movimiento de agua.

Ahora si se hubiera optado por una bomba de corriente alterna, la menor potencia encontrada es 0.5 HP, con una carga máxima de 30 mca y una descarga y succión de 1 pulgada. Con lo cual dificultaría muchas conexiones y tendría que adecuar una fuente que funcione con esta corriente especialmente, sin contar con añadir un diferencial y un contactor para el encendido y apagado.

#### **4.1.3. Fuente de agua**

La fuente de agua que abasteció al área viene extraída del pozo comunitario del AA. HH Palmeras de Huarangal, el cual brinda agua las 24 horas, comúnmente no hay interrupciones, pero en ocasiones la presión disminuye o hay días de mantenimiento y el flujo de agua no llega con continuidad, siendo su máxima presión de agua en horas de la noche con un caudal de 180 L/h.

Debido a la incertidumbre de flujo de agua en la red, para poder automatizar y utilizar continuamente el agua, se instaló un tanque de agua FarPlast de 4 capas. La capacidad de éste tanque se estimó teniendo como referencia el consumo máximo emitido por los goteros de 2 L/h, ya que si las cuatro áreas están encendidas el consumo para los 140 goteros sería un total de 280 L/h con lo que un tanque de una capacidad de 300 litros cumpliría los requisitos.

Por otro lado, para asegurar un excedente del consumo predestinado y por ser el volumen comercial existente se optó por adquirir un tanque de 350 litros capacidad con un costo aceptable en comparación con otras opciones. En la Figura 17 se aprecia que se adicionó al tanque, un filtro de malla Mesh 130 micras antes del ingreso del agua hacia el tanque y una

válvula de ingreso de agua tipo bola, asegurando así un flujo libre de partículas que podrían obturar los goteros.



**Figura 17: Accesorios y partes de la fuente de agua**

Cabe mencionar que se colocó el filtro de mallas antes del ingreso de agua para el llenado del tanque, con la finalidad de evitar alguna pérdida que pueda ocasionar éste accesorio si iría después de la descarga (salida de agua del tanque o bomba).

#### **4.1.4. Caseta de control de riego**

Ésta área construida alberga al microcontrolador; las conexiones de los actuadores, los sensores, la fuente de energía, los interruptores y el cableado desde donde se recibirán y emitirán las señales para su posterior funcionamiento.

En la Figura 18 se observa la caseta de control construida y forrada con triplay y madera tornillo, posee un techo de planchas acanaladas de policarbonato para proteger contra

lloviznas. También está dividida en dos compartimientos, en su segundo piso se encuentra la caja del microcontrolador y sus implementos, las fuentes eléctricas y los interruptores desde donde se dotó de energía eléctrica al micro controlador, válvulas solenoides y bomba de agua mediante un transformador de corriente alterna (AC) a corriente continua (DC). Y en el primer piso está las conexiones que vienen desde el tanque, válvulas, unión universal y pasan por la bomba hasta salir de ella hacia los arcos de riego.



Figura 18: Caseta de control

#### 4.1.5. Tuberías

Para las tuberías primaria y secundaria de PVC el menor diámetro comercial encontrado fue de ½” y para los porta goteros se utilizó diámetro de ¾”, éste último se utilizó por facilidad con la conexión de conector de línea (conector inicial + empaque).

El enterrado de las tuberías se realizó a 30 cm de la superficie y por el centro del área de estudio.

#### 4.1.6. Arcos de riego

Se instaló en cada una de las cuatro áreas, arcos de riego junto a los sensores de flujo (para registrar el flujo de agua), el cableado para el sensor de humedad de suelo (que será el indicador para abrir o cerrar la válvula según las necesidades de agua presente en el suelo) y las válvulas solenoides, como se ilustra en la Figura 19.

Este arco está a una elevación de 25 cm a partir de la superficie del suelo y se conecta con la porta lateral.

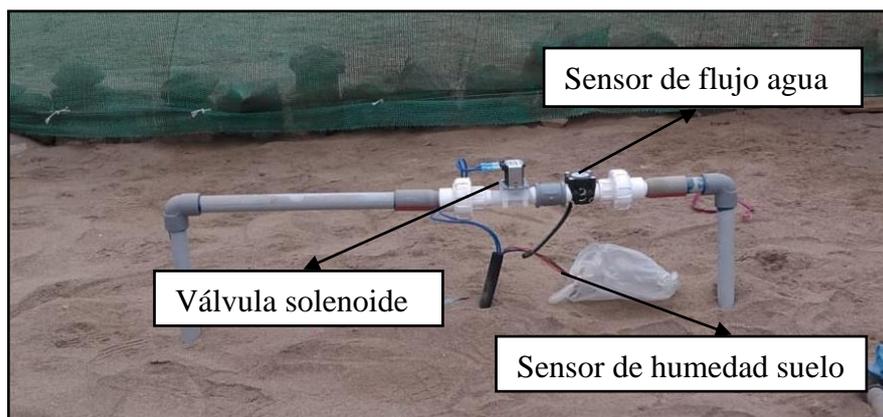


Figura 19: Partes del arco de riego

#### 4.1.7. Porta goteros

Según el diseño hidráulico el diámetro de los porta goteros debería ser de 3.53 mm, pero éste diámetro no es comercial, con lo cual el diámetro más próximo e ideal sería el de 12.00mm de diámetro, pero éste no es muy común en los centros de venta y se optó por utilizar un diámetro comercial de 16 mm para los porta goteros.

Se colocaron los goteros en cada lateral según el diámetro que humedecen, que es de 46 cm, distribuyéndolos a una distancia de 36 cm entre goteros como se visualiza en la Figura 20, siguiendo la recomendación de la separación entre goteros del 80 % del diámetro

humedecido. En total se colocaron 10 goteros por cada porta gotero, con lo cual en total se colocaron 140 goteros auto compensantes en toda el área.

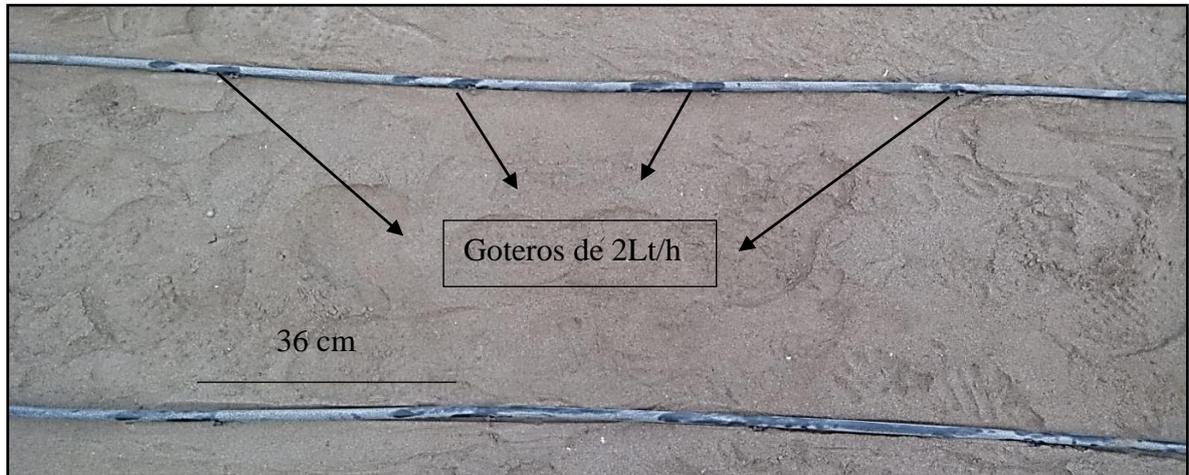


Figura 20: Porta goteros auto compensantes de 2L/h

## 4.2. Instalación electrónica del riego automatizado

Se instalaron todos los actuadores y sensores al microcontrolador Arduino Leonardo tal como se muestra en el Anexo 30.

### 4.2.1. Sensores de humedad de suelo

Se instalaron cuatro sensores de humedad capacitivos al Arduino Leonardo según la Figura 21, utilizando los pines analógicos A0, A1, A3, A5 del mismo, para medir el por ciento de humedad en función al estrato de saturación del suelo. Estos sensores fueron ubicados en cada suelo de la parte intermedia de porta goteros.

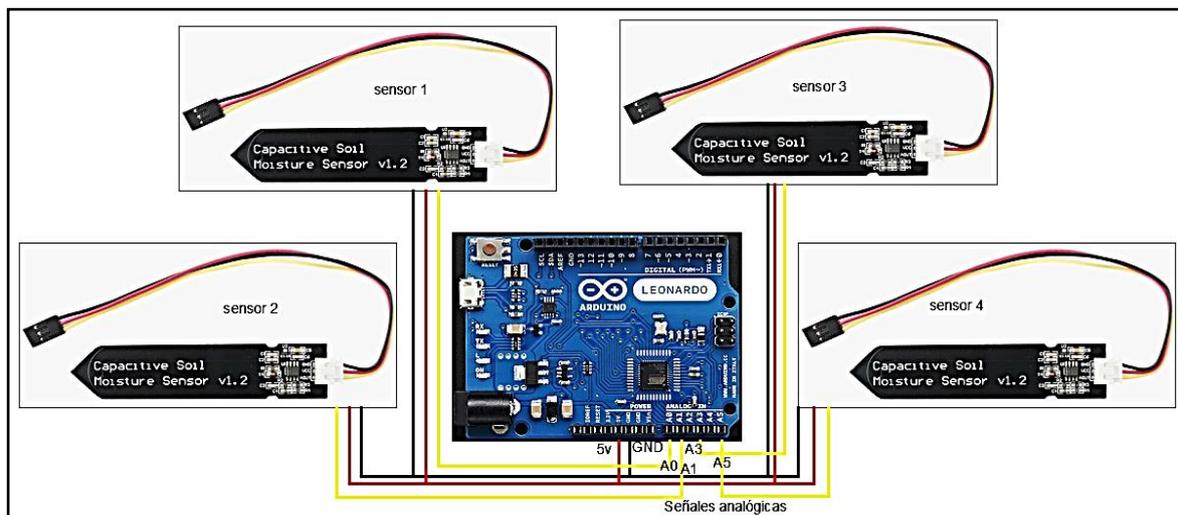


Figura 21: Conexiones de los sensores de humedad con el Arduino Leonardo

Se hace el hincapié que estos sensores capacitivos son más resistentes que los sensores resistivos, además de tener un mayor tamaño con el cual se puede alcanzar más profundidad de medición de la humedad en el suelo.

#### 4.2.2. Sensor de flujo de agua

Se instalaron cuatro sensores de flujo de agua de ½” al Arduino Leonardo según la Figura 22, utilizando los pines digitales que admiten interrupciones 3, 2, 0, 1 del mismo, para medir el caudal en L/min que fluye y el consumo en litros por parte de cada área(grupo). Estos sensores fueron ubicados seguidamente después de la válvula solenoide en cada uno de los arcos de riego.

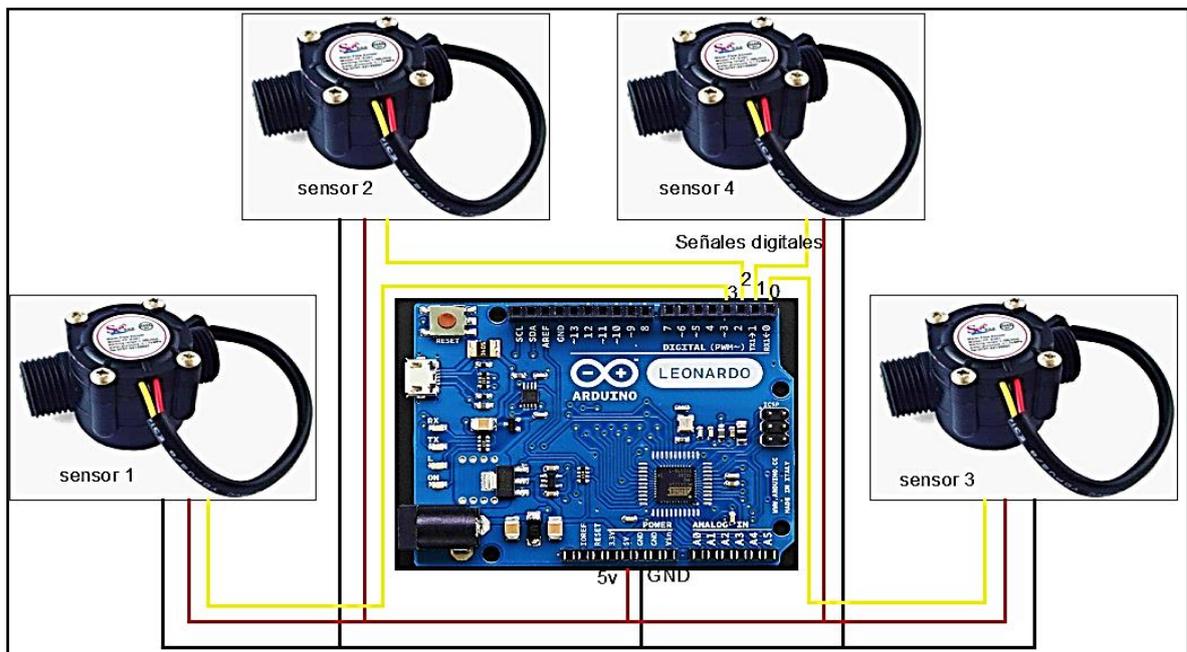


Figura 22: Conexiones de los sensores de flujo con el Arduino Leonardo

Estos sensores son buenos para contabilizar el flujo de agua en términos de cantidades importantes en varios litros, por otro lado, si se buscara tener una precisión en los datos a recabar, se hace la mención que se debe evitar su uso.

#### 4.2.3. Sensor de ultra sonido

Se conectó un sensor ultra sonido al Arduino Leonardo según la Figura 23, utilizando los pines digitales 8 y 9, con lo cual se controló el nivel de agua óptimo para su funcionamiento. Éste sensor se instaló en la parte superior de la tapa del tanque y colocada de manera vertical, con el cual se controló el nivel de agua a la que el sistema se enciende, siendo ésta menor a

129 cm y en nivel a la cual se apaga todo, nivel mayor a 130 cm. Asegurando así que la bomba del sistema trabaje con un flujo constante y no se dañe por problemas de flujo en seco o podría sufrir problemas de cavitación. El sensor de ultra sonido posee una buena precisión en el campo, pero se debe evitar confundir sus conexiones, ya que el cruce entre un cable de la señal con uno de tierra genera daño en sus componentes internos.

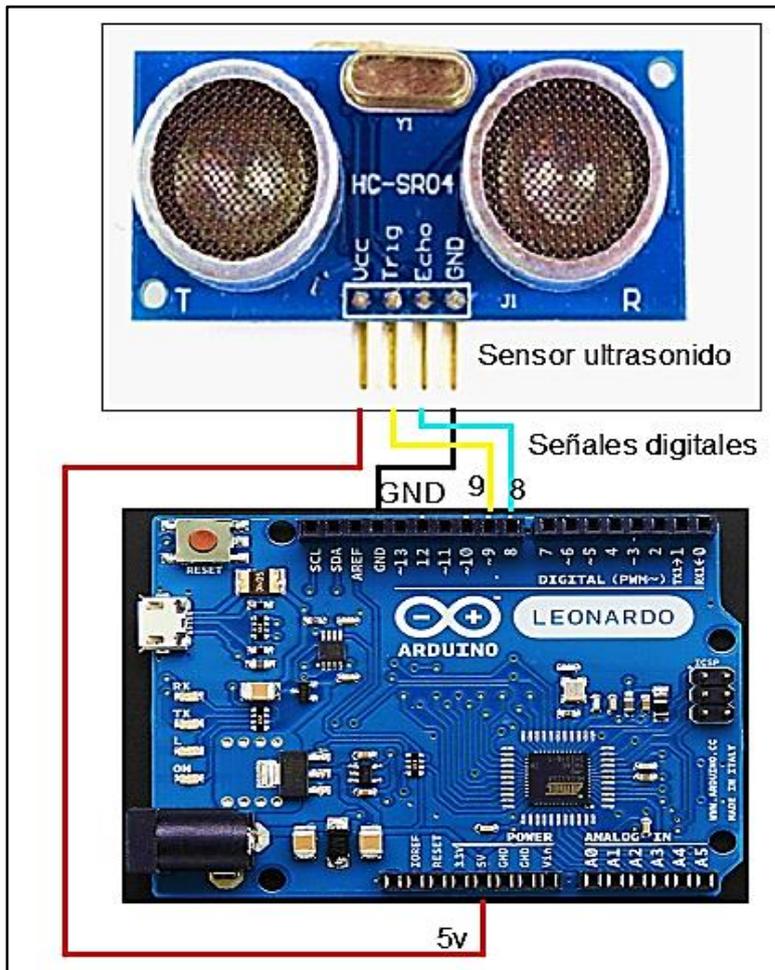


Figura 23: Conexiones del sensor de ultrasonido con el Arduino Leonardo

#### 4.2.4. Válvulas solenoides

Se instalaron cuatro electroválvulas de ½” al Arduino Leonardo según la Figura 24 por intermedio de un relay de cuatro canales, utilizando los pines digitales 10,11, 12, 5 del Arduino, para controlar el paso del flujo de agua en cada área (grupo). Estos actuadores fueron ubicados antes de los sensores de flujo en cada uno de los arcos de riego.

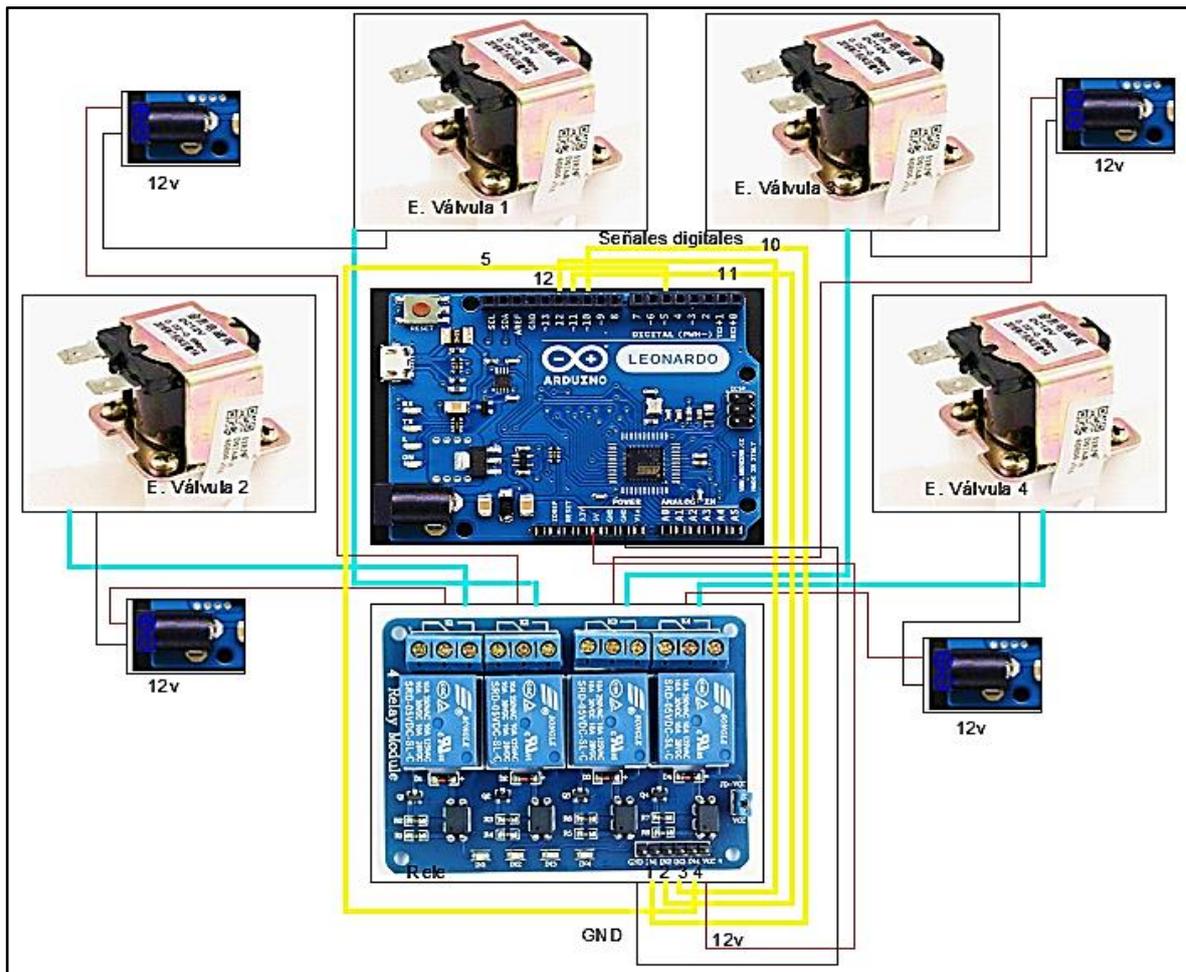


Figura 24: Conexión del relay y válvulas con el Arduino Leonardo

#### 4.2.5. Bomba de agua

Se conectó una bomba tipo centrífuga al Arduino Leonardo según la Figura 25, por intermedio de un transistor mosfet de un canal, utilizando el pin digital 6 del Arduino, para impulsar y hacer circular el flujo hacia cada área (grupo o sub unidad) según sea la necesidad.

Éste actuador fue ubicado después del tanque de reservorio y filtro de malla por donde pasa el agua hacia cada arco de riego.

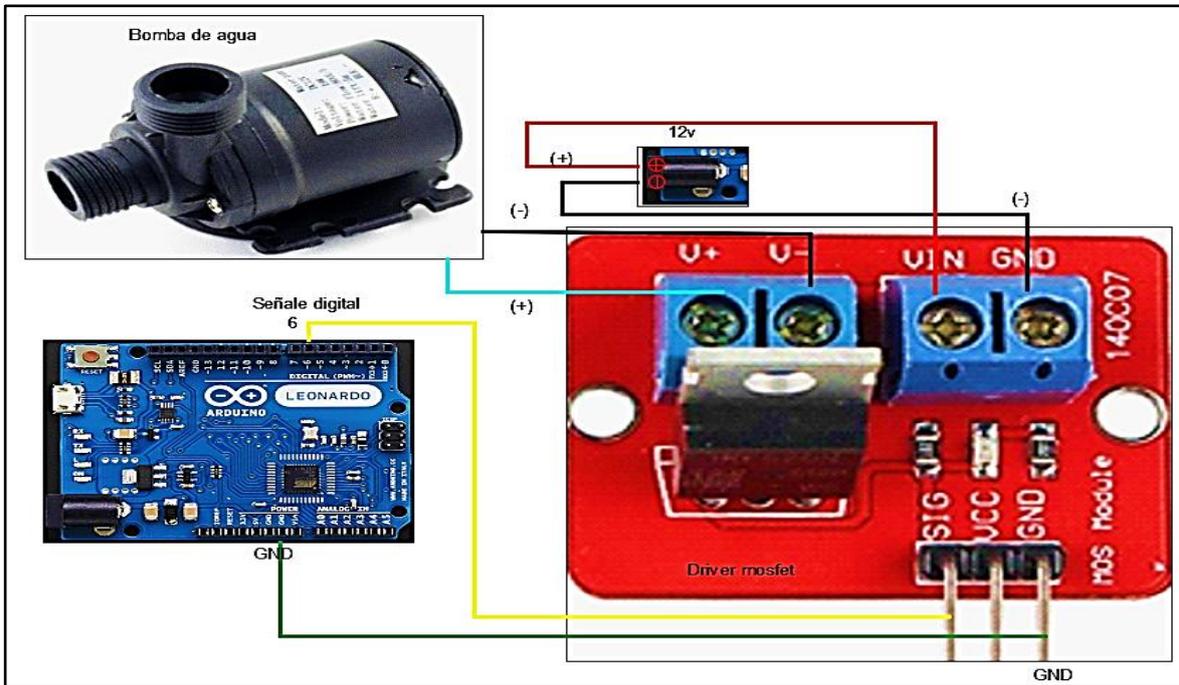


Figura 25: Conexión del transistor mosfet y bomba con el Arduino Leonardo

#### 4.2.6. Módulo de lector de micro SD

Se conectó el lector de micro SD a través de ICSP del micro controlador Arduino como se ve en la Figura 26, utilizando los pines 3, 4, 1, 2, 6 de la conexión ICSP y el pin digital 4 del Arduino. Éste módulo lector se utilizó para almacenar lecturas de humedad del suelo en porcentaje, caudal en L/min y consumo de agua en litros de los cuatro arcos de riego instalados, éste almacenamiento se archivó en cuatro documentos txt cada uno independiente.

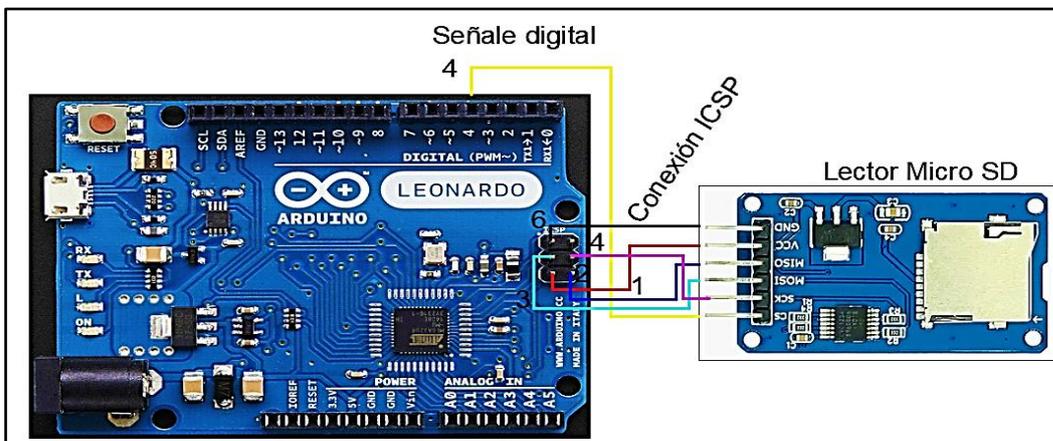


Figura 26: Conexión del lector micro SD con el Arduino Leonardo

En la Tabla 22 se muestra de manera resumida el tipo de conexión, el pin del microcontrolador y el parámetro o la acción que mide o ejecuta cada sensor o actuador.

**Tabla 22: Puertos utilizados en la placa Arduino Leonardo**

Sensor / Actuador	Pin de Arduino	Tipo de conexión	Parámetro a medir/ controlar
Sensor de flujo 1	3	entrada digital	cantidad de agua al área 1
Sensor de flujo 2	2	entrada digital	cantidad de agua al área 2
Sensor de flujo 3	0	entrada digital	cantidad de agua al área 3
Sensor de flujo 4	1	entrada digital	cantidad de agua al área 4
Válvula solenoide 1	5	salida digital	abrir o cerrar flujo de agua al área 1
Válvula solenoide 2	10	salida digital	abrir o cerrar flujo de agua al área 2
Válvula solenoide 3	11	salida digital	abrir o cerrar flujo de agua al área 3
Válvula solenoide 4	12	salida digital	abrir o cerrar flujo de agua al área 4
Bomba de agua	6	salida digital	impulsar o parar el flujo de agua de la fuente
Sensor ultrasónico	8	entrada digital	controla el nivel de agua del tanque de la fuente de agua
	9	salida digital	
Lector de micro SD	4	entrada digital	registra los datos de los sensores de las 4 áreas
	3,4,1,2,6	ICSP	
Sensor humedad suelo 1	A0	entrada analógica	registra el nivel de agua en el área 1
Sensor humedad suelo 2	A1	entrada analógica	registra el nivel de agua en el área 2
Sensor humedad suelo 3	A2	entrada analógica	registra el nivel de agua en el área 3
Sensor humedad suelo 4	A3	entrada analógica	registra el nivel de agua en el área 4

### 4.3. Instalación eléctrica del riego automatizado

#### 4.3.1. Cableado

Para esto se calculó un total de consumo de amperaje en corriente alterna de 0.3 A por adaptador de corriente que se utilizó para dar energía al sistema de corriente continua, en total se utilizaron 6 con lo que el consumo sería  $0.3 \times 6 = 1.8$  A en corriente alterna, con éste dato se podría utilizar cualquier cable AWG hasta el n° 22, pero por facilidad en la instalación, se escogió el cable n°16 con una resistencia de hasta 19 Amperios, lo cual se asegura una mejor manipulación y que el cable no se queme debido al consumo.

Se debe mencionar que el cableado más utilizado en el mundo de la programación con Arduino es los cablecillos multifilar n°22 o en su defecto cables Jumper, el problema con éstos es su sensibilidad, tienden a romperse con una fuerza leve y para realizar el tendido de los cables desde el micro controlador hasta cada sensor resulta una tarea dificultosa.

#### 4.3.2. Fuente de energía

La vivienda cuenta con un fluido eléctrico constante y estable el cual llega hasta el tablero general del domicilio, descrita en el diagrama de la figura 27, en donde se cuenta con una llave térmica general de 25 amperios, una llave diferencial de 25 Amperios, del cual se realizó el tendido del cable hasta una caja eléctrica independiente para el proyecto, el cual cuenta con una llave térmica de 16 amperios, desde donde se realizó el cableado hasta la caseta de control donde el microcontrolador, los sensores y actuadores tomarán la energía desde los adaptadores de fuente(DC) de 12v -1A instalados.

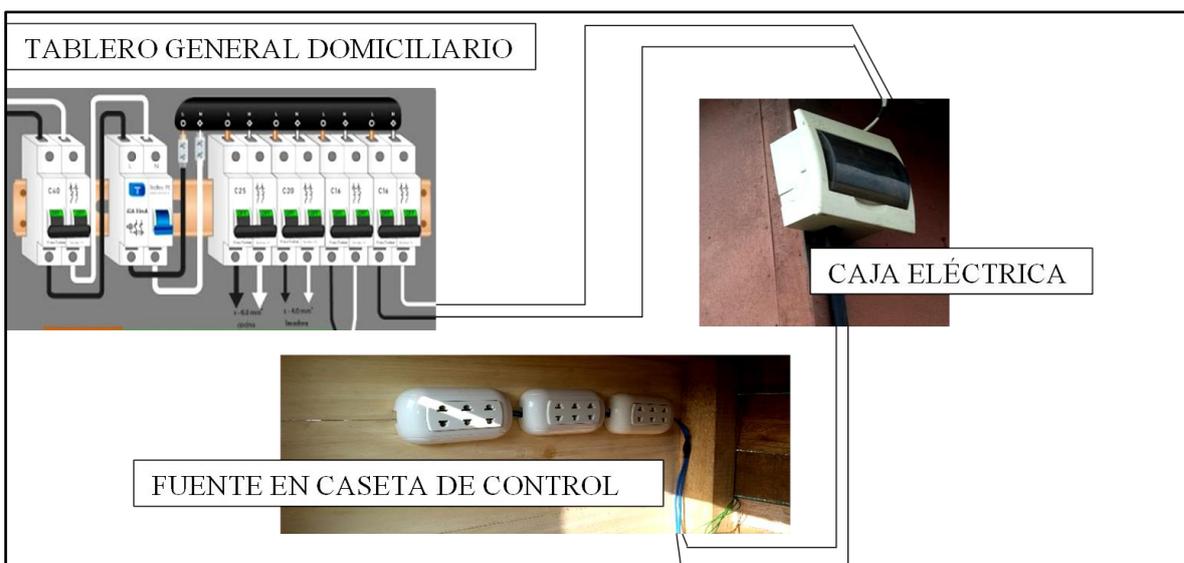


Figura 27: Diagrama de la instalación eléctrica

#### 4.4. Control manual de sistema de riego

##### 4.4.1. Control de bomba de agua

El encendido de la bomba se controló con un interruptor luminoso el cual al cerrar el circuito con la fuente de alimentación de 12V-1A se encendió el led y se dió inicio al encendido de la bomba.

Se debe precisar que el control de la bomba se realiza conjuntamente con el control de la(s) válvula(s) solenoide(s) ya que ésta bomba no puede funcionar en seco. Esto implica que se encendió la bomba es porque se abrió una válvula solenoide de las cuatro áreas.

En la figura 28 se visualiza su conexión.



Figura 28: Conexión eléctrica de bomba-interruptor

##### 4.4.2. Control de válvulas solenoides

Los controles de éstas válvulas también se realizaron con interruptores luminosos el cual al cerrar el circuito con la fuente de alimentación de 12V-1A se encendió el led y la válvula pasó de NC (normalmente cerrado) a NO (normalmente abierto).

Para éste caso se tuvieron que manejar cuatro interruptores para todas las sub unidades del área experimental. Éstas si se pueden abrir y cerrar independientemente si la bomba funciona o no.

En la figura 29 se visualiza su conexión



Figura 29: Conexión válvula solenoide-interruptor

#### 4.5. Calibración de sensores

##### 4.5.1. Sensor de humedad de suelo

Para cada sensor de humedad se realizaron cinco mediciones en agua y al aire, en un intervalo de tres minutos para estabilizar la medida, logrando registrar valores, en las Tablas 23, 24, 25 y 26, con los cuales se obtendrá el promedio que fue utilizado en la programación general del Arduino Leonardo.

- Sensor de humedad de suelo n°1

**Tabla 23: Mediciones con sensor de humedad n°1**

Sensor de humedad n° 1		
Nº medición	Ambiente	Agua
1era medición	554	323
2 da medición	555	321
3 era medición	553	324
4 ta medición	554	322
5 ta medición	555	325
Promedio	554	323

- Sensor de humedad de suelo n°2

**Tabla 24: Mediciones con sensor de humedad n°2**

Sensor de humedad n° 2		
Nº medición	Ambiente	Agua
1era medición	547	306
2da medición	546	304
3era medición	537	303
4ta medición	535	301
5ta medición	536	302
Promedio	540	303

- Sensor de humedad de suelo n°3

**Tabla 25: Mediciones con sensor de humedad n°3**

Sensor de humedad n° 3		
Nº medición	Ambiente	Agua
1era medición	546	335
2da medición	547	336
3era medición	546	334
4ta medición	547	335
5ta medición	547	336
Promedio	547	335

- Sensor de humedad de suelo n°4

**Tabla 26: Mediciones con sensor de humedad n°4**

Sensor de humedad n° 3		
Nº medición	Ambiente	Agua
1era medición	530	332
2da medición	528	326
3era medición	531	325
4ta medición	535	331
5ta medición	533	332
Promedio	531	329

Cabe mencionar que éste sensor tiende a variar en cuanto al valor de sus mediciones, ya que, si calibramos cada uno de los sensores con el microcontrolador en gabinete, nos arrojará otros valores distintos a que si calibráramos lo mismo en el campo. Ésto en parte es debido a los ruidos que ocasiona cuando se trasmite la señal a lo largo del cableado.

#### 4.5.2. Sensor de flujo

Para cada sensor de flujo se realizó seis mediciones, tres para 0.8 litros y tres para 1.6 litros, anotando los pulsos registrados, el tiempo de los pulsos, los Herz o sea cada pulso por segundo y finalmente se halló con el caudal el valor del coeficiente K propio de cada sensor, descritos en las Tablas 27, 28, 29 y 30. Con los resultados registrados se halló el promedio el cual fue utilizado para el cálculo del volumen y el caudal del sensor en la programación general del Arduino Leonardo.

- Sensor de flujo nº1

**Tabla 27: Mediciones con sensor de flujo nº1**

Sensor de flujo nº 1							
nº Pruebas	Volumen (L)	nº pulsos	Tiempo de prueba (s)	nº pulsos / segundo	Hz	Q(l/min)	K
1	1	916	9.6	95.4	95.4	6.3	15.3
2	1	974	9.75	99.9	99.9	6.2	16.2
3	1	921	9.61	95.8	95.8	6.2	15.4
4	1	945	9.45	100.0	100.0	6.3	15.8
5	1	951	9.28	102.5	102.5	6.5	15.9
6	1	940	9.58	98.1	98.1	6.3	15.7
Promedio		941	14.4	98.6	98.6	6.3	15.7

- Sensor de flujo nº2

**Tabla 28: Mediciones con sensor de flujo nº2**

Sensor de flujo nº 2							
nº Pruebas	Volumen (L)	nº pulsos	Tiempo de prueba (s)	nº pulsos / segundo	Hz	Q(l/min)	K
1	1	13455	9.28	1449.9	1449.9	6.5	224.3
2	1	9354	9.45	989.8	989.8	6.3	155.9
3	1	10854	9.18	1182.4	1182.4	6.5	180.9
4	1	9910	9.41	1053.1	1053.1	6.4	165.2
5	1	11544	9.54	1210.1	1210.1	6.3	192.4
6	1	10544	9.32	1131.3	1131.3	6.4	175.7
Promedio		10944	9.4	1169.4	1169.4	6.4	182.5

- Sensor de flujo nº3

**Tabla 29: Mediciones con sensor de flujo nº3**

Sensor de flujo nº 3							
nº Pruebas	Volumen (L)	nº pulsos	Tiempo de prueba (s)	nº pulsos / segundo	Hz	Q(l/min)	K
1	1	7850	9.21	852.3	852.3	6.5	130.8
2	1	6023	9.26	650.4	650.4	6.5	100.4
3	1	6875	9.45	727.5	727.5	6.3	114.6
4	1	6850	8.8	778.4	778.4	6.8	114.2
5	1	7324	9.05	809.3	809.3	6.6	122.1
6	1	7421	9	824.6	824.6	6.7	123.7
Promedio		7057	13.8	773.8	773.8	6.6	117.7

- Sensor de flujo nº4

**Tabla 30: Mediciones con sensor de flujo nº4**

Sensor de flujo nº 4							
nº Pruebas	Volumen (L)	nº pulsos	Tiempo de prueba (s)	nº pulsos / segundo	Hz	Q(l/min)	K
1	1	4493	8.2	547.9	547.9	7.3	74.9
2	1	4108	8.8	466.8	466.8	6.8	68.5
3	1	4794	8.1	591.9	591.9	7.4	79.9
4	1	4545	8.35	544.3	544.3	7.2	75.8
5	1	4454	9.3	478.9	478.9	6.5	74.2
6	1	4751	8.3	572.4	572.4	7.2	79.2
Promedio		4524	12.7	533.7	533.7	7.1	75.5

Éste sensor tuvo el mismo comportamiento que los sensores de humedad de suelo capacitivo, ya que la calibración en gabinete arroja números distintos a los arrojados en una calibración en campo como los presentados en ésta sección.

#### 4.6. Programación

Para la programación se utilizó el entorno integrado de desarrollo del Arduino Leonardo (IDE) que viene incorporado con la descarga del software libre Arduino. En esta ventana fueron digitando los códigos de actuadores, sensores y demás elementos electrónicos junto con sus variables globales, locales y librerías propias de cada elemento. El procedimiento seguido para la programación fue el descrito en la Figura 31 y fue ejecutado de la siguiente manera:

### **Digitación de variables globales y las librerías de:**

- Módulo lector de micro SD, se incluyeron librerías SPI y SD para la comunicación y facilitar el almacenamiento de los parámetros captados, se crearon cuatro archivos para guardar los resultados de los sensores instalados en cada uno de los arcos de riego y se designó la variable para el pin digital nº 4 (Anexo 20).
- Se designó variables para los sensores de humedad de suelo 1, 2, 3 y 4 para los pines analógicos A5, A1, A3 y A0 respectivamente (Anexo 24).
- Se designó variables para las electroválvulas 1, 2, 3 y 4 con los pines 5, 10, 11 y 12 (Anexo 21).
- Se designó variable para la bomba de agua en el pin nº 6 (Anexo 22).
- Para los sensores de flujo 1, 2, 3 y 4 se asignaron variables en los pines digitales con interrupción 2, 3, 0 y 1 en el orden en el cual se dan las interrupciones. También aquí se designó constante al tiempo de cada una de las interrupciones, al conteo de pulsos de cada sensor, al factor de coeficiente K propio de cada sensor, al volumen registrado de cada sensor y se inscribieron las funciones para cada uno de los sensores con los cuales se calcula dicho volumen (Anexo 26).
- Por último, se dió variables al sensor ultrasónico para sus dos conexiones en los pines digitales 8 (Trigger) y 9 (Echo) (Anexo 27).

### **En la estructura del void setup():**

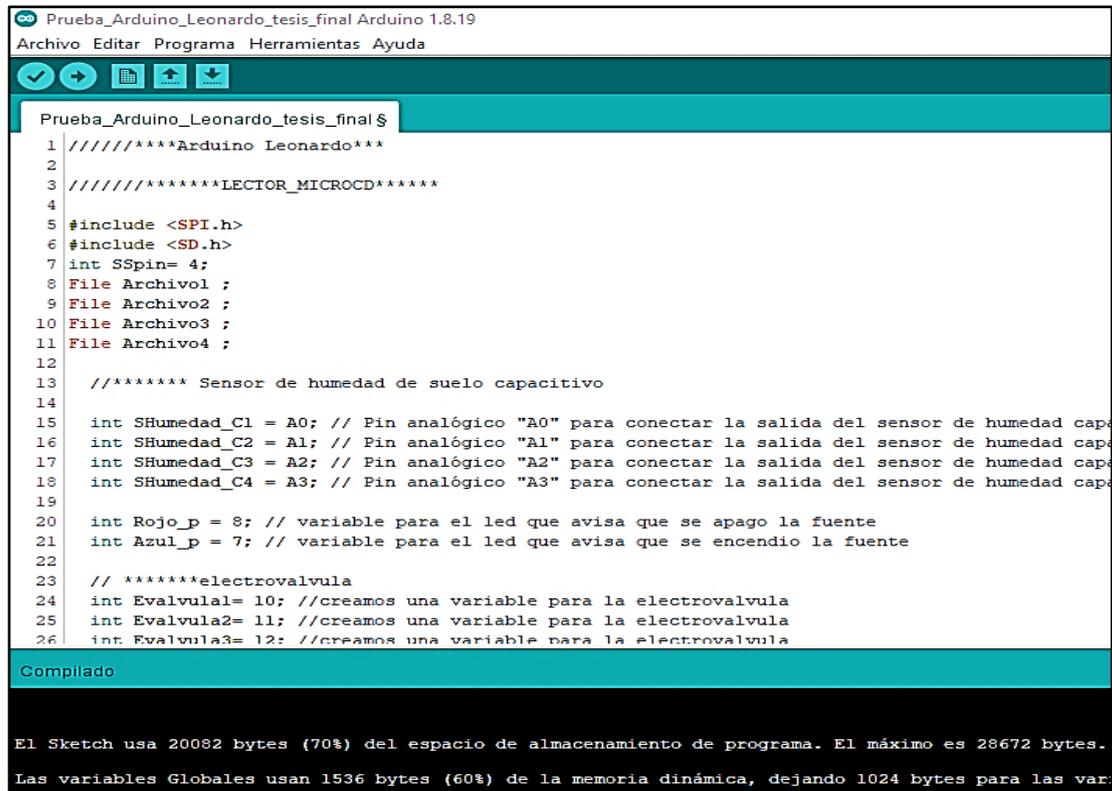
- Se declaró como salida a las válvulas solenoide, bomba y al pin “Trigger” del sensor de ultra sonido (por donde sale la señal del ultrasonido). Como entrada solo se declaró al pin “Echo” del sensor ultrasonido (por donde entra la señal del ultrasonido). Nota 1: no se declaró los sensores de flujo ni de sensor de humedad de suelo porque el software los interpreta como entradas implícitas.
- Se digitó función 4 interrupción para la conexión de los 4 sensores de flujo, el conteo de pulso y su estado (modo de acción).
- Se hizo un llamado a la inicialización correcta del lector de micro SD.

Se hace la acotación que para el caso del lector micro SD si hay una correcta inicialización el programa funcionará y empezará a leer el código, si no se arrojará en el monitor serial “fallo en la inicialización” deteniéndose todo el sistema.

#### **En la estructura del void loop():**

- Se habilita el tiempo en millis, se pone una condicional con respecto al tiempo del ultrasonido, se crea variables locales tipo “long” para el tiempo y distancia del sensor de ultrasonido, se envía la señal de ultrasonido que dura 1 micro segundos y luego se recepciona, con lo cual se calcula la distancia.
- Después se crea las condicionales que regirán el funcionamiento, la primera nos indicará que para registros mayores de distancia =100 cm, que arroje el sensor ultrasónico, se cumplirá que se desenergizará la bomba y las electroválvulas, con lo cual el flujo no circula indicando que no hay suficiente agua en el tanque y obligando a abrir la fuente para el llenado del mismo. El segundo, para valores menores de 100 cm se cumplirá que funcionará cada arco de riego según la humedad registrada (si el suelo tiene una humedad menor del 50% se abre la válvula solenoide y si su humedad supera el 50% se cerrará la válvula solenoide indicando que el suelo está en un punto de saturación adecuada)
- Con respecto al contenido de humedad superior del 50%, se digitó una condicional que apaga la bomba en caso todas las áreas registren porcentajes superiores al 50% y se mantendrá prendido la bomba registrando solo un área un 50% menor de humedad.
- Por otro lado, se digitó códigos para registrar con el sensor de flujo el caudal (en litros por minuto) y consumo de agua (en litros) en cada arco de riego. Para ésto se tuvo que crear funciones “void flujo ()” para cada sensor, en cada función se registró el conteo de pulsos de manera ascendente con lo cual ayudó a calcular el caudal y posteriormente el consumo de agua de cada área.
- Finalmente, todos estos datos de cada sensor en cada arco de riego fueron registrados y almacenados en las carpetas previamente creadas. Al cerrar el programa o al extraer la memoria micro SD los datos quedarán registrados en formato txt.
- Código completo de la programación se encuentra en el Anexo 28 y 29.

En la figura 30 se puede apreciar en el IDE de Arduino la digitación del código, el cual fue verificado, cargado y ejecutado.



```
Prueba_Arduino_Leonardo_tesis_final Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

Prueba_Arduino_Leonardo_tesis_final $
1  //*****Arduino Leonardo***
2
3  //*****LECTOR_MICROCD*****
4
5  #include <SPI.h>
6  #include <SD.h>
7  int SSpin= 4;
8  File Archivo1 ;
9  File Archivo2 ;
10 File Archivo3 ;
11 File Archivo4 ;
12
13 //***** Sensor de humedad de suelo capacitivo
14
15 int SHumedad_C1 = A0; // Pin analógico "A0" para conectar la salida del sensor de humedad cap
16 int SHumedad_C2 = A1; // Pin analógico "A1" para conectar la salida del sensor de humedad cap
17 int SHumedad_C3 = A2; // Pin analógico "A2" para conectar la salida del sensor de humedad cap
18 int SHumedad_C4 = A3; // Pin analógico "A3" para conectar la salida del sensor de humedad cap
19
20 int Rojo_p = 8; // variable para el led que avisa que se apago la fuente
21 int Azul_p = 7; // variable para el led que avisa que se encendio la fuente
22
23 // *****electrovalvula
24 int Evalvula1= 10; //creamos una variable para la electrovalvula
25 int Evalvula2= 11; //creamos una variable para la electrovalvula
26 int Evalvula3= 12; //creamos una variable para la electrovalvula

Compilado

El Sketch usa 20082 bytes (70%) del espacio de almacenamiento de programa. El máximo es 28672 bytes.
Las variables Globales usan 1536 bytes (60%) de la memoria dinámica, dejando 1024 bytes para las var
```

Figura 30: Código de la programación en software Arduino

Durante la programación se acota que no fue posible el uso del comando “delay”, ya que cada vez que se ejecute un código de la estructura “void loop” lo que hace ese comando es detener por “completo” la lectura de la programación por el tiempo establecido y recién se reanuda cuando acaba. Lo que dificultaría el funcionamiento de los demás comandos que dependen de funcionamiento individual, como por ejemplo el código de la lectura de sensor de humedad y flujo, la lectura del sensor ultrasónico y el código para guardar los datos en la tarjeta micro SD.

Por todo lo mencionado antes se optó por utilizar el comando “millis”, con el cual se aparenta una ejecución simultánea del código con lo cual hace funcionar todos los artículos conectados al Arduino(multifunciones).

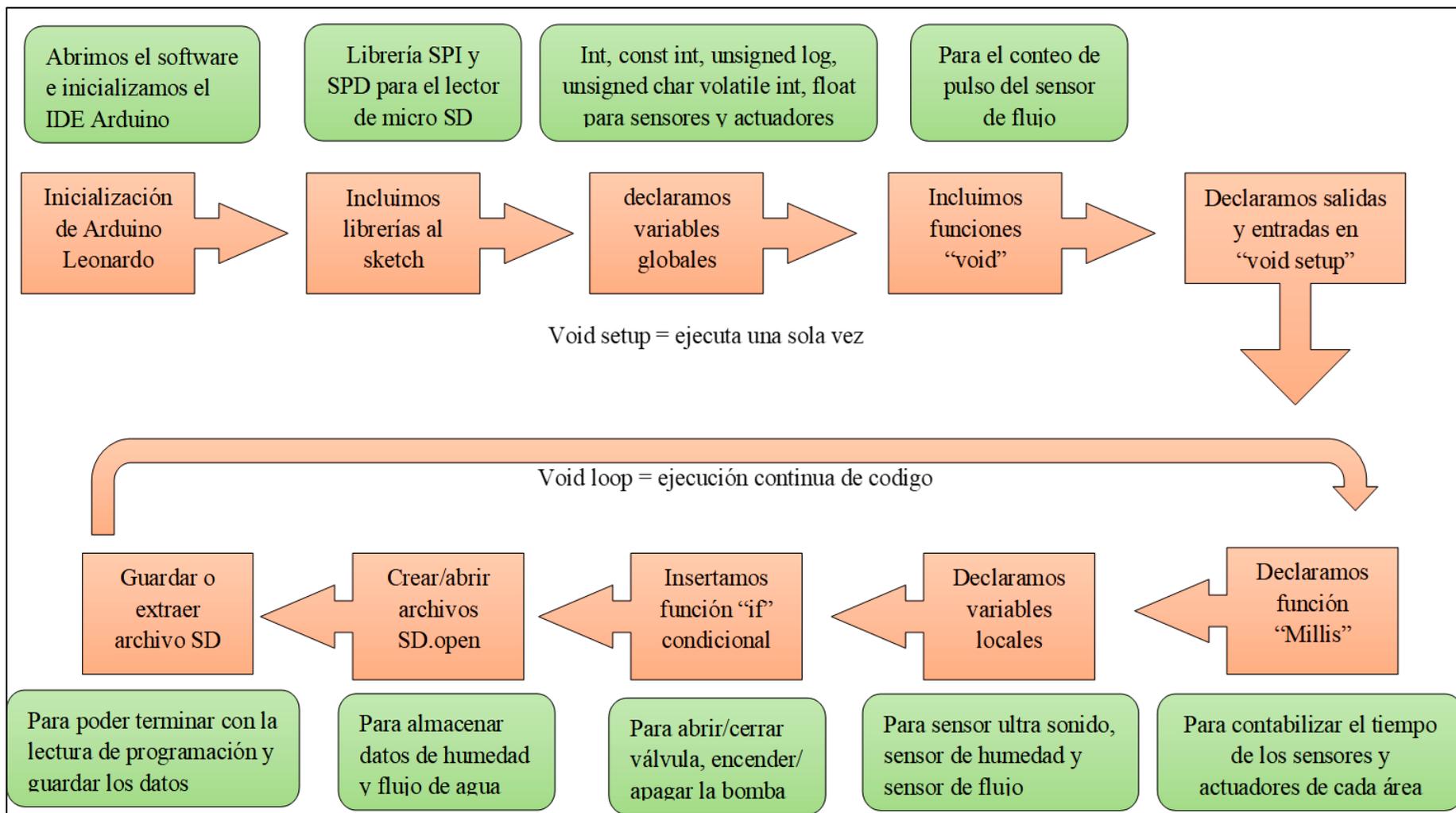


Figura 31: Procedimiento de la programación del código en la placa Arduino Leonardo

## 4.7. Pruebas de funcionamiento

Se realizaron pruebas de encendido y apagado para cada una de las áreas independientemente una de otra, tanto con el software Arduino como con el aplicativo ArduinoDroid y se visualizó el comportamiento de los sensores. También se realizaron las pruebas de funcionamiento de los goteros a través del cálculo del coeficiente de uniformidad CU.

### 4.7.1. Software Arduino

Para el caso del software Arduino se utilizó una computadora portátil en la cual se inicializó, abriendo los sketches de códigos para cada área y el sketch completo para todas las áreas juntas, en donde se compiló y se cargó los códigos digitados al microcontrolador, se abrió el monitor serial dando inicio a la tarjeta de memoria que almacenará estado de humedad del suelo, consumo y caudal de agua y nivel de agua en el tanque.

Se registraron los códigos del sensor de humedad, flujo y nivel de agua en el monitor serial de la Figura 32, para cada área. Como ejemplo se visualiza que para el área 1 se registró, estado seco 0%, estado medio húmedo 45% y estado húmedo con 69%, datos fluctuantes entre los valores mencionados.

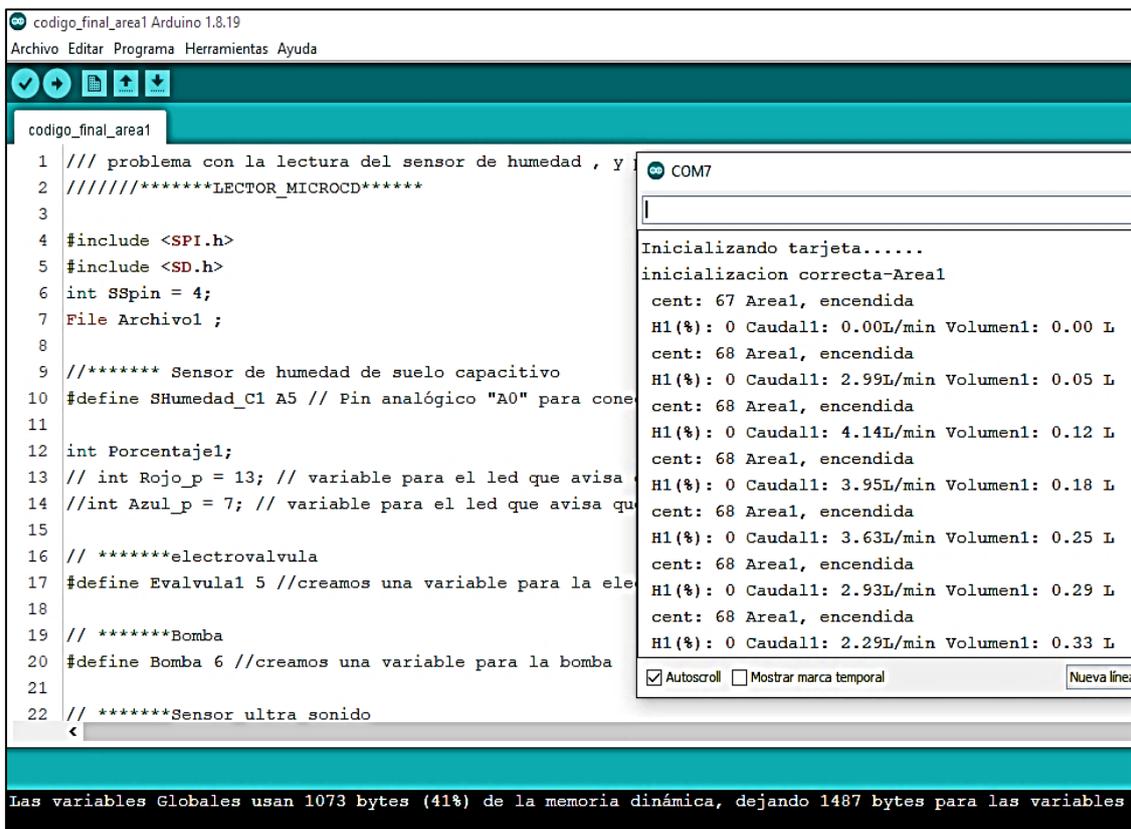


Figura 32: Lectura monitor serial área 1-estado seco-encendida

#### 4.7.2. Sensores de humedad de suelo y flujo de agua

Se probó los sensores durante el funcionamiento y se almacenaron los resultados en el módulo micro SD adicionado al micro controlador Arduino Leonardo, los cuales arrojaron comportamientos similares en cada una de las áreas en las que están ubicados los sensores.

El funcionamiento del sistema desde que se enciende con el celular o laptop hasta que sea apagado con el corte del fluido eléctrico, hace que éstos sensores capten las señales y sean remitidas al microcontrolador el cual seguirá la secuencia del código desarrollado en placa.

Para este caso, se tomó las lecturas emitidas por los sensores ubicados en el área 3.

- Sensor de humedad de suelo capacitivo

Una vez iniciado la secuencia del código desarrollado en Arduino, el sensor de humedad fue introducido al medio en el suelo del área 3 y cercano a uno de los goteros. En sus inicios el sensor registró un porcentaje de humedad de 0% en el estrato de saturación y rápidamente fue subiendo hasta niveles cercanos a 50% donde pasado éste valor, el sensor emitió una señal al micro controlador el cual indicó que el suelo estaba saturado en un nivel mayor al 50%, como lo muestra la Figura 33, con lo cual se procedió a cerrar automáticamente la electro válvula impidiendo el paso hasta que el agua presente en el suelo disminuya y el sensor capte ésta variación, con lo cual se abrirá la electroválvula y continúe con el ciclo.



Figura 33: Humedad de suelo en el estrato de saturación

- Sensor de flujo YF-S201

Durante el desarrollo de la programación, la bomba impulsó el fluido de agua a través de la red de tuberías hasta llegar al arco de riego donde está el sensor de flujo, el cual se encargó de convertir las revoluciones en pulsos y a su vez en cantidad de litros por minuto que pasan por el sensor. En la Figura 34 se ve el caudal que pasa en el mismo momento que el sensor de humedad está tomando la lectura de la cantidad de agua en el suelo y también en la Figura 35 se ve el consumo de agua que hay hasta ese momento. Cabe resaltar que cuando se cierra la electroválvula ya no se registrará el caudal y el consumo(litros) y se mantendrá y se irá acumulando cuando vuelva a abrirse la electroválvula dejando pasar el agua.

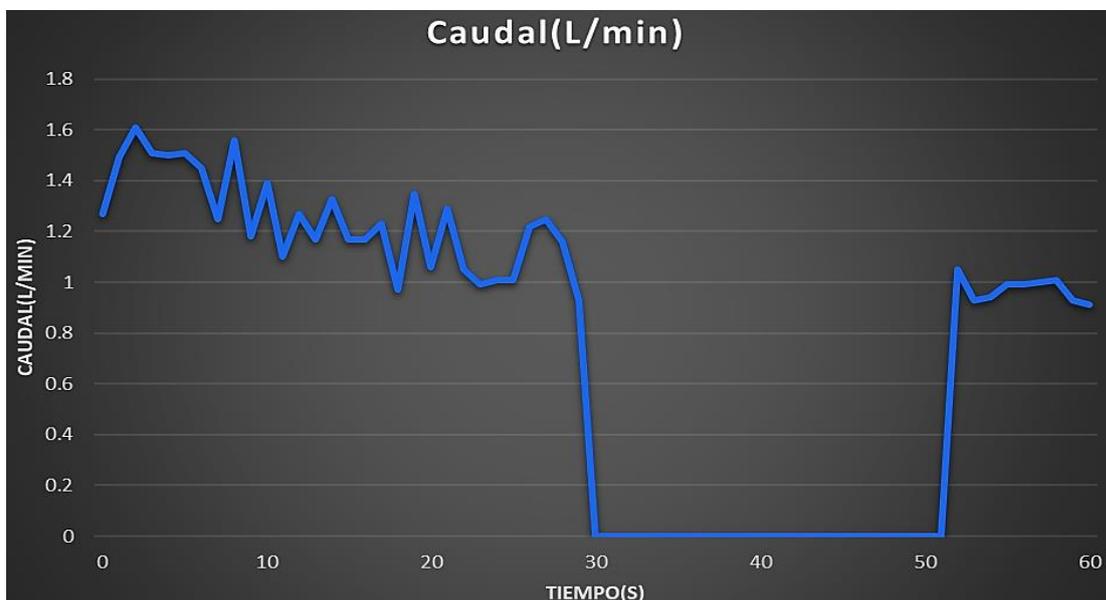


Figura 34: Caudal en litros por minuto



Figura 35: Litros consumidos

### 4.7.3. Aplicativo Arduinodroid

Se realizó la conexión entre el microcontrolador y el celular mediante un adaptador OTG-USB con el cual se logró cargar el programa al aplicativo ArduinoDroid y así pudiendo encender el sistema de riego automatizado.



Figura 36: Encendido utilizando el aplicativo Android del celular

Se debe aclarar que el aplicativo ArduinoDroid no posee una interfaz propia ni personalizada y tampoco cuenta con bluetooth para monitorear o encender, dicho aplicativo contrasta en ese aspecto con: Bluetooth Controller 8 Lamp, Remotexy, Control de Arduino, Virtuino, Blynk etc.

### 4.7.4. Coeficiente de uniformidad

Se realizó la prueba de uniformidad con ayuda de dos recipientes graduados en mililitros con un tiempo de 3 min y 3 repeticiones, para el área 1 que es la más alejada de la fuente de agua, registrando los valores y dando los siguientes resultados promedio de las 3 repeticiones en la Tabla 31.

Arrojando un coeficiente de uniformidad del 79%, así como se muestra en la figura 37, lo cual está en el rubro de aceptable con tendencias a ser bueno el CU.

**Tabla 31: Volumen y caudal por Líneas de goteros y posición de los goteros**

Gotero n°	Línea 4		Línea 3		Línea 2		Línea 1	
	V (mil)	Q (mil/min)	V (mil)	Q (mil/min)	V (mil)	Q (mil/min)	V (mil)	Q (mil/min)
1	315	105.0	300	100.0	345	115.0	300	100.0
2	320	106.7	345	115.0	325	108.3	300	100.0
3	290	96.7	310	103.3	300	100.0	278	92.7
4	215	71.7	265	88.3	250	83.3	200	66.7

Orden	Sin ordenar	Ordenado
	Q(mil/min)	Q(mil/min)
1	100.00	115.00
2	100.00	115.00
3	92.67	108.33
4	66.67	106.67
5	115.00	105.00
6	108.33	103.33
7	100.00	100.00
8	83.33	100.00
9	100.00	100.00
10	115.00	100.00
11	103.33	96.67
12	88.33	92.67
13	105.00	88.33
14	106.67	83.33
15	96.67	71.67
16	71.67	66.67
Qm (25%)		77.50
Qm (100%)		97.04
Cu (Qm (25%) /Qm (100%))		79.86

**Figura 37: Caudales de la prueba de CU**

#### 4.8. Operatividad

Para lograr un mejor entendimiento, con una información simplificada, procedimientos claves, continuos y detallados y que se pueda tomar como referencia éste trabajo para su posterior replicación en otros lugares, se elaboró unos procedimientos para su instalación, funcionamiento y automatizado.

#### 4.8.1. Funcionamiento manual

En este caso es simple el funcionamiento, ya que esto se activa mediante interruptores lumínicos (Figura 27, 28 y 29) que indicaron según el área de riego, al cual se le está regando y el tiempo que uno desee al abrir o cerrar el interruptor (apaga y prende) manualmente.

Cabe resaltar que su instalación no es complicada y se resume en la Figura 38, en la cual se detalla los pasos seguidos. El entusiasta que desee replicar éste trabajo solo debe seguir el procedimiento, acotando que para espacios pequeños como éste caso(40m<sup>2</sup>) no es necesario cálculos de diámetros de tuberías (porque se utiliza la de ½” para la conducción de agua y ¾” para porta goteros), pérdidas por rozamiento (que se demostró que es casi nula) y para los goteros dependerá de los cultivos que se vayan a tener (si se quiere aprovechar al máximo todo el espacio se tendría que utilizar gotero de 2L/h a 36 cm como máximo) y para la bomba, cualquiera con entrada de ½” de corriente continua y que emita un caudal superior a 300 L/h se considera correcto.

Se hace mención que para éste procedimiento no se requiere de conocimientos avanzados y solo de una adecuada explicación gráfica y un buen entendimiento.

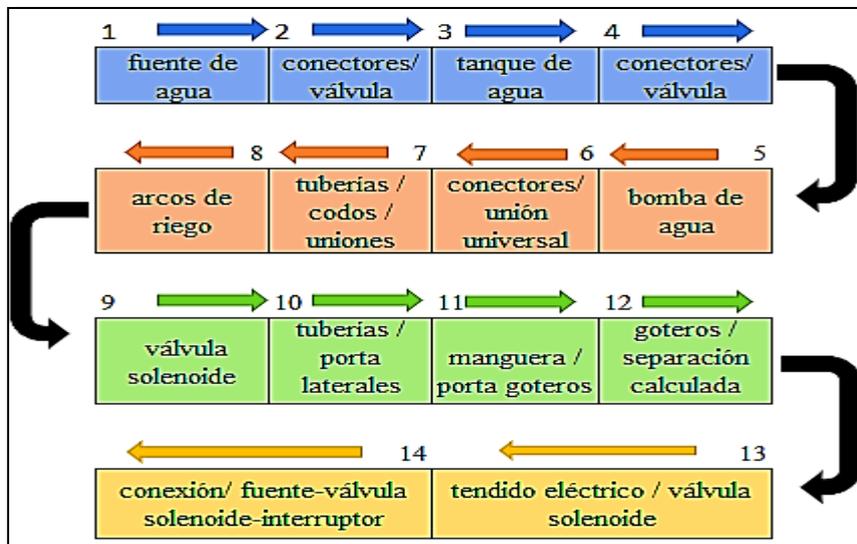


Figura 38: Esquema secuencial de la instalación manual

#### 4.8.2. Funcionamiento automatizado

Éste caso fue más complejo y se requirió alimentar el microcontrolador con una fuente de 9V-1A y ejecutar la inicialización de la secuencia con una computadora portátil con el software Arduino o en su defecto abrir el archivo del código completo e inicializar mediante el aplicativo ArduinoDroid desde el celular Android.

Por otro lado, la dificultad de éste funcionamiento radica en que el entusiasta debe tener un nivel avanzado de entendimiento de automatización, micro controladores, sensores, actuadores, programación y uso de aplicativo ArduinoDroid para ejecutar el sistema. Por eso se pretende que el entusiasta que desee replicar éste proyecto copie los códigos realizados en éste trabajo. Comenzando con los códigos para la calibración descritos en los Anexos 23 y 25. En cuanto a la inicialización de la secuencia el código para el uso de una computadora portátil se encuentra descrito en el Anexo 29 y para el uso de celular con Android el código está descrito en el Anexo 28 siempre y cuando antes se tiene que descargar el aplicativo y guardar el archivo del código.

Se precisa que la instalación sigue el procedimiento de la Figura 39, con la diferencia que en éste punto se tiene que tender cables para arrastrar la señal de los actuadores y sensores hasta el microcontrolador donde se tienen que ensamblar pin por pin tal cual está descrito en el código de la programación secuencial.

Por otro lado, se cumple que, para espacios pequeños, no es necesario cálculos de diámetros de tuberías, pérdidas por rozamiento, para los goteros dependerá de los cultivos que se vayan a tener y para la bomba, cualquiera con entrada y salida de ½” de corriente continua y que emita un caudal superior a 300 L/h se considera correcto.

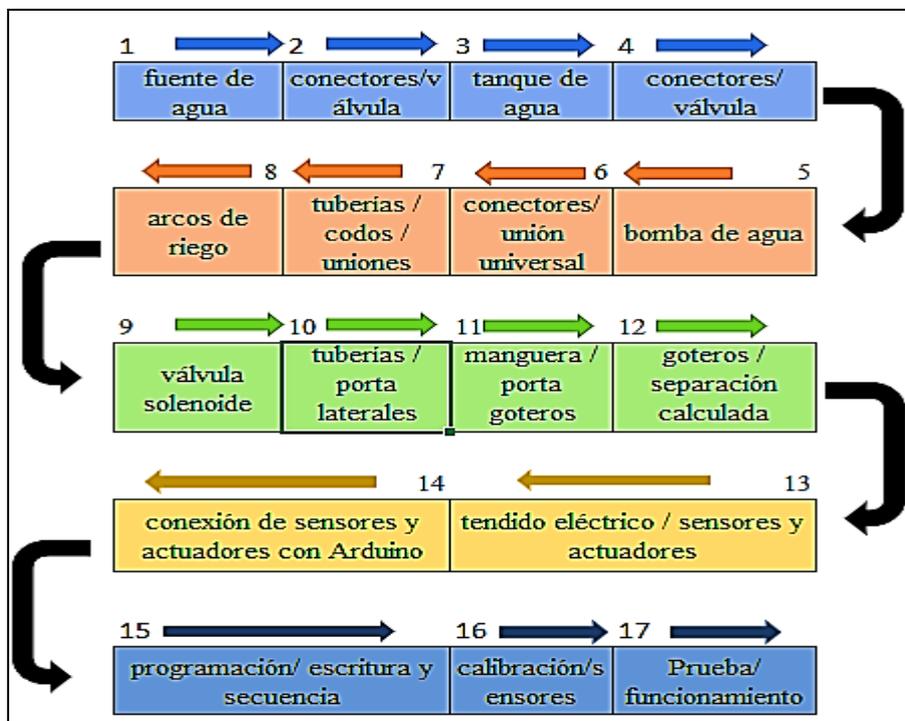


Figura 39: Esquema secuencial de la instalación automatizada

#### 4.9. Presupuesto

En el presupuesto general se contemplaron aquellos artículos y aparatos que solo se han definido e instalado en el sistema de riego automatizado, los cuales salieron como resultados del diseño y el planteamiento del prototipo automatizado. Cabe mencionar también que se han utilizado otros artículos compatibles con Arduino que en el camino del desarrollo del proyecto se han ido desestimando.

**Tabla 32: Materiales definitivos utilizados en el proyecto de tesis**

PRESUPUESTO GENERAL DEL PROTOTIPO DEL SISTEMA AUTOMATIZADO					
Componentes	Descripción	Cantidad	Unidad	Precio unitario	Precio total S/.
Sistema de riego	Tubería 1/2"	15	Metros	3.6	54
	Tubería 3/4"	10	Metros	5.6	56
	Manguera ciega de polietileno 16mm	100	Metros	0.8	80
	Cinta teflón	7	Rollo	1.9	13.3
	Pegamento	0.5	Litros	32	16
	Codo 90° 1/2"	21	Unidad	1.5	31.5
	Conexión T 1/2"	1	Unidad	1.8	1.8
	Conexión T 3/4"	5	Unidad	2.1	10.5
	Goteros micro Flipper 2l/h	140	Unidad	0.4	56
	Válvula ramal 16mm	14	Unidad	2.3	32.2
	Unión universal de 1/2"	10	Unidad	8.5	85
	Abrazadera zincada 16mm	17	Unidad	0.6	10.2
	Válvula esférica de 1/2"	3	Unidad	5	15
	Reductor de 3/4" a 1/2" SP	4	Unidad	2.5	10
	Adaptador 1/2"	18	Unidad	1	18
	Adaptador hembra-hembra de 1/2"	5	Unidad	1	5
	Tapones para tubo 3/4"	8	Unidad	0.6	4.8
	Conector inicial 16mm	14	Unidad	0.6	8.4
	Terminal de línea tipo 8	14	Unidad	0.5	7
	Punzón de gotero	1	Unidad	5	5
	Filtro de malla	1	Unidad	25	25
	Bushin tipo campana 1/2" a 3/4"	6	Unidad	2.5	15
	Adaptador de 1/2 "a 3/4"	4	Unidad	2	8
	Tanque 350 Lt	1	Unidad	380	380
	Hoja de sierra	2	Unidad	7	14
	Niple	1	unidad	2	2
	Bushin tipo campana 1/2" a 1/2"	2	unidad	0.8	1.6

PRESUPUESTO GENERAL DEL PROTOTIPO DEL SISTEMA AUTOMATIZADO						
Componentes	Descripción	Cantidad	Unidad	Precio unitario	Precio total S/.	
	Conexión tipo cruz de 1/2" pvc	1	unidad	8	8	
Sistema automatizado	Sensor de humedad de suelo	4	unidad	8	32	
	Sensor de flujo	4	unidad	20	80	
	Válvula solenoide	4	unidad	25	100	
	Arduino Leonardo	1	unidad	65	65	
	Modulo relay 4 CH	1	unidad	24	24	
	Módulo Driver mosfet	1	unidad	6	6	
	Módulo lector de micro SD	1	unidad	8	8	
	Memoria micro SD 16GB	1	unidad	25	25	
	bomba de agua	1	unidad	70	70	
	Protoboar 830	1	unidad	12	12	
	Sensor de ultra sonido	1	unidad	7	7	
Sistema eléctrico	Cablecillo multifilar 1m	180	metro	0.3	54	
	Cables jumper (M-M,H-H,M-H)	80	unidad	0.25	20	
	Bornera aérea 3A	2	unidad	2.5	5	
	Molex de 1 pin	2	docena	3	6	
	Rollo de cable n°16	100	metro	1	100	
	Fuente conmutada 12V 20A	1	unidad	47	47	
	Fuente de 9 V 1 A, AC - DC	1	unidad	10	10	
	Fuente de 12 V 1 A, AC - DC	6	unidad	10	60	
	Conector plug DC con bornera hembra	6	unidad	1.5	9	
	Manga termo contraible	2	metro	1.5	3	
	Terminal aislado hembra	2	docenas	2	4	
	Pernos y tornillos	2	docenas	3	6	
	Caja eléctrica	1	unidad	15	15	
	Terminales para cable	2	docena	3	6	
	Llave térmica	1	unidad	16	16	
	Tomacorriente doble	2	unidad	6	12	
	Interruptor	5	unidad	2.5	12.5	
	Rollo de cinta aislante	2	rollo	6	12	
	Servicios	Análisis de agua	1	muestra	66	66
		Caracterización del suelo	1	muestra	0	0
Mano de obra por Instalación del sistema de riego, sistema eléctrico y automatización		1	proyecto	0	0	
Asesoramiento en software y hardware		1	proyecto	0	0	
Construcción de caseta		1	unidad	0	0	
Movilidad		1	proyecto	350	350	

PRESUPUESTO GENERAL DEL PROTOTIPO DEL SISTEMA AUTOMATIZADO					
Componentes	Descripción	Cantidad	Unidad	Precio unitario	Precio total S/.
	Malla reshell 50%	54	metro	1	54
	Troncos para cercos	15	metro	2	30
	Alambre galvanizado n° 14	2	kilo	10	20
	Abono	5	costal	5	25
	Tierra de chacra	10	costal	3	30
Otros	Bisagra	1	par	3.5	3.5
	Candado	1	unidad	6	6
	Planchas 2x1m	2	unidad	10	20
	Maderas 5.0m	1	unidad	15	15
	Clavos de 1/2"	0.5	kilo	8.5	4.25
	Clavos de 1 1/2"	0.5	kilo	12	6
	Cuerda	19	metro	1	19
<b>GASTO TOTAL</b>					<b>2448.55</b>

Para la ejecución de éste proyecto no se contó con personal técnico o de ingeniería, pero si un personal de apoyo para el enterrado de las tuberías, para el cercado y para el acondicionamiento del área a cultivar. En cuanto a la construcción e instalación, sistema eléctrico, electrónico y sistema de riego por goteo junto con sus accesorios, se utilizó los conceptos básicos adquiridos en el periodo universitario en la carrera de ingeniería agrícola, complementado con cursos virtuales y materiales académicos gratuitos en YouTube y Google de libre descarga.

Por otro lado, se hace mención que los materiales utilizados en éste trabajo en cuanto al sistema de riego son el resultado de las necesidades halladas y cálculos hidráulicos, adquiriendo tuberías, accesorios de riego, conexiones, válvulas, tanque y filtros lo más idóneos posibles y siempre guardando la relación con el aspecto económico. De nada serviría utilizar materiales de marca o con resistencias o capacidades excesivas, si éste resulta muy costoso.

En cuanto a la automatización se hizo la adquisición de los actuadores y sensores en función de la utilidad y relevancia del producto con respecto a lo planteado, no es lo mismo utilizar un sensor de humedad de suelo de alta precisión para saber cuánto porcentaje relativo de agua hay en el suelo, si con solo saber un rango o margen de humedad presente en el estrato de saturación bastaría para activar o desactivar la bomba Arduino, pero el caso del sensor de

humedad resistivo el problema radicaría en la poca durabilidad que posee, oxidándose rápidamente, por eso se opta por un sensor capacitivo.

Quizás la diferencia hubiera significado en conocer con exactitud la cantidad de agua que consume cada área de riego, pero en estos casos ya no se utiliza un sensor de flujo si no un contador digital de agua con un precio entre 10 a 20 veces más dependiendo del grado de precisión, por otro lado, está el hecho que no son compatibles con el software Arduino.

También para cercar o delimitar el terreno se hubiera preferido el uso de cercos vivos, cultivos arbustivos o en su defecto cactus, el detalle recae en la practicidad, ya que estos cultivos si se cultivan son de lento crecimiento, si se compra listo para cultivar aumentaría el costo del proyecto, sin contar con el tiempo que se tendría que esperar para poder recién delimitar el área.

## V. CONCLUSIONES

1. El diseño agronómico agroecológico utilizó principios básicos de la agroecología para la selección, diversificación, distribución y complementariedad de los cultivos en el área planteada, para el Kc del cultivo al ser un huerto con policultivos se tomó el que tiene mayores necesidades hídricas. En cuanto a la lámina de riego se tuvo 5.35 mm/ día y para un área neta de cultivo de 38 m<sup>2</sup> dió 203.3 L/día de necesidades brutas de agua a adicionar al suelo. El tiempo de riego será de 10 min a un intervalo estimado de 6 horas de riego con 12 horas hábiles al día, por temas de control de salinidad.
2. Para el diseño hidráulico se obtuvo una pérdida de 0.31 m en el recorrido crítico del sistema para tuberías de 1/2" y manguera de 16mm, hallando una carga dinámica total de 6.79m con lo cual se tiene un requerimiento de bomba de 6.56 watts a lo que se utilizó una bomba tipo centrífuga de corriente continua compatible con Arduino y una carga de 8m. En cuanto al funcionamiento se realizó la prueba de coeficiente de uniformidad (Cu) el cual arrojó 79% de uniformidad con lo que se asegura una distribución uniforme del agua en cada gotero y una humedad mínima en el área de los cultivos.
3. El prototipo automatizado utilizando el microcontrolador Arduino Leonardo cumplió con su funcionamiento, encendiendo las áreas cuando el porcentaje de agua presente en el estrato de saturación está por debajo del 50% y apagando cuando está por encima del 50% de humedad del suelo, esto gracias al sensor de humedad de suelo capacitivo por su fácil desarrollo del código y su calibración, también se pudo conocer el consumo de agua en litros de cada una de las áreas del proyecto, con el uso de sensor de flujo, el cual se calibró para lograr una mayor precisión para finalmente ser almacenado en un micro SD una vez finalizado el funcionamiento del sistema. En cuanto a su operación en campo, resulta práctico el uso del aplicativo ArduinoDrid en el celular ya que facilita el encendido y modificación del código sin el uso de conexión a internet. Por otra parte, resultó fundamental la implementación del sistema de riego de encendido manual ya que facilita el uso en caso de fallar el microcontrolador o en caso de dificultar el uso del sistema automatizado.

## **VI. RECOMENDACIONES**

1. Replicar éste sistema en áreas mayores, optimizando la capacidad de la mini bomba y extendiendo la longitud de los porta goteros.
2. Adaptar el diseño para uso de cultivos en macetas y en cultivos verticales (Anexo 34).
3. Trabajar con longitudes de cableado menores o iguales a 10 metros en la conexión del microcontrolador con los sensores.
4. Realizar en campo la calibración del sensor de humedad de suelo capacitivo, ya que las señales emitidas por éste sensor provienen de una señal analógica las cuales son muy susceptibles a cambios y fluctuaciones en las mediciones dadas.
5. Realizar en campo la calibración del sensor de flujo de agua, tomando en cuenta el caudal emitido por la fuente (bomba).
6. Utilizar fuente de alimentación externa de acuerdo a las especificaciones del consumo de energía de los componentes utilizados en el sistema, para una adecuada operación.
7. Tener en cuenta la lista de biofertilizantes orgánicos del Anexo 40 para el abonamiento.
8. Ejecutar alternativas para mantener los niveles recomendables de salinidad (Anexo 41).
9. Incorporar al mejoramiento del suelo planteado, materia orgánica y fuentes naturales de abono (Anexo 42).

## VII. BIBLIOGRAFÍA

- Alcaraz, F. (2012). Salinidad y vegetación. In *Geobotánica* (p. 11). Universidad de Murcia.  
<https://www.um.es/docencia/geobotanica/ficheros/tema18.pdf>
- Altieri, M., & Nicholls, C. (2019). Agroecología Urbana: Diseño de Granjas Urbanas Biodiversas, Productivas y Resilientes. *Celia Boletín Científico*, 2(2), 1–28.  
<https://doi.org/10.4206/agrosur.2018.v46n2-07>
- Anton Smirnov. (2021, March 18). *ArduinoDroid - Arduino/ESP8266/ESP32 IDE - Apps en Google Play*.  
[https://play.google.com/store/apps/details?id=name.antonsmirnov.android.arduinoroid&hl=es\\_PE&gl=US](https://play.google.com/store/apps/details?id=name.antonsmirnov.android.arduinoroid&hl=es_PE&gl=US)
- Arduino. (2018). *¿Qué es Arduino? | Arduino*.  
<https://www.arduino.cc/en/Guide/Introduction>
- Bernis, F. (2020). *Diámetro óptimo en impulsiones . Optimal diameter in drives . 19*.  
[https://www2.uned.es/ca-tortosa/Biblioteca\\_Digital/Biblio/JM\\_Franquet/Disen\\_economico.pdf](https://www2.uned.es/ca-tortosa/Biblioteca_Digital/Biblio/JM_Franquet/Disen_economico.pdf)
- Briceño, M., Álvarez, F., & Barahona, U. (2012). MODULO 5: MANUAL DE RIEGO Y DRENAJE. In P. Abelino & P. Valladares (Eds.), *Riego y drenaje*. Escuela Agrícola Panamericana, Carrera de Ciencias y producción agropecuaria El Zamorano, Honduras, Centroamérica. <https://doi.org/10.1785/0120050052>
- Cerna, L. (2011). *Manual de Olericultura* (Primera ed). Fondo editorial UPAO.  
[https://descarga.upao.info/manual\\_olericultura](https://descarga.upao.info/manual_olericultura)
- Comunidad Andina. (2011). Agricultura Familiar Agroecológica Campesina en la Comunidad Andina. In *Comunidad Andina* (p. 54). Comunidad Andina.  
[https://www.comunidadandina.org/StaticFiles/2011610181827revista\\_agroecologia.pdf](https://www.comunidadandina.org/StaticFiles/2011610181827revista_agroecologia.pdf)
- Courel, G. (2019). Guía de estudio. Suelos Salinos y Sódicos. *Journal of Chemical Information and Modeling*, 53(9), 1–9.  
<https://webcache.googleusercontent.com/search?q=cache:bjR8L0Fx46AJ:https://www>

[.edafologia.org/app/download/9026474176/Suelos%2BSalinos%2BBy%2Bs%25C3%25B3dicos%2B2019.pdf%3Ft%3D1563476239+%&cd=1&hl=es-419&ct=clnk&gl=pe](https://edafologia.org/app/download/9026474176/Suelos%2BSalinos%2BBy%2Bs%25C3%25B3dicos%2B2019.pdf%3Ft%3D1563476239+%&cd=1&hl=es-419&ct=clnk&gl=pe)

Cropaia. (2023). *La Conductividad Eléctrica del Agua y Suelo*. Google. <https://cropaia.com/es/blog/conductividad-electrica-del-agua-y-suelo/>

Domingo, J. (n.d.). *Sistema electrónico para el control de riego automático*. 62. Retrieved August 7, 2023, from <https://biblus.us.es/bibing/proyectos/abreproy/12418/fichero/PFC-2418-DAMIGO.pdf>

Estudio FAO Riego y Drenaje 56. (2006). *Evapotranspiración del cultivo* (56th ed.). FAO. <http://www.fao.org/3/x0490s/x0490s00.htm>

FAO. (1996). *Estado Mundial de la Agricultura y La Alimentación 1996* (Nº29). Colección FAO; Agricultura, Nº29. <http://www.fao.org/3/a-w1358s.pdf>

FAO. (2014). *Agricultura Familiar En América Latina y El Caribe: Recomendaciones de Política* (S. Salomón & L. Guzmán, Eds.). FAO. <https://www.fao.org/3/i3788s/i3788s.pdf>

FAO. (2018). *LOS 10 ELEMENTOS DE LA AGROECOLOGIA: Guía para la transición hacia sistemas alimentarios y agrícolas sostenibles* (p. 15). FAO. <https://www.fao.org/publications/card/es/c/I9037ES>

FAO, & CIRAD. (2021). *FRUTAS Y HORTALIZAS: Oportunidades y desafíos para la agricultura sostenible a pequeña escala* (J. Marshall, Ed.). FAO-CIRAD. <https://doi.org/https://doi.org/10.4060/cb4173es>

Fernández, Rafael., Yruela, M. del Carmen., Milla, Mercedes., García, Juan., & Oyonarte, Nicolas. (2010). *Manual de Riego para Agricultores Módulo 4: Riego Localizado* (Junta de A). [http://www.juntadeandalucia.es/export/drupaljda/1337160941RIEGO\\_BAJA.pdf](http://www.juntadeandalucia.es/export/drupaljda/1337160941RIEGO_BAJA.pdf)

Google Maps. (2021). *Área geográfica del distrito de Lurín*. <https://goo.gl/maps/2BeTqLRPKoso8uJY8>

Hernández, I., Fajardo, J., Vanegas, O., & Jimmy, B. (2011). *Guía básica de Arduino* (p. 120). [https://www.maristashuelva.es/webinfo/tecnologia/arduino/Libro\\_kit\\_Basico.pdf](https://www.maristashuelva.es/webinfo/tecnologia/arduino/Libro_kit_Basico.pdf)

Huerta, A., Bonnesoeur, V., Cuadros-Adriazola, J., Gutiérrez, L., Ochoa-Tocachi, B. F., Román-Dañobeytia, F., & Lavado-Casimiro, W. (2022). PISCOe\_o\_pm, a reference

- evapotranspiration gridded database based on FAO Penman-Monteith in Peru. *Scientific Data*, 9(1), 1–18. <https://doi.org/10.1038/s41597-022-01373-8>
- Lliota, Mario., Carrión, Rolando., Ciancaglini, Nicolas., & Olguin, Alfredo. (2015). MANUAL DE CAPACITACION: RIEGO POR GOTEO. *Inta*, 15. [https://inta.gob.ar/sites/default/files/inta\\_manual\\_riego\\_por\\_goteo.pdf%0Afile:///C:/Users/SURI/OneDrive/Documentos/inta\\_manual\\_riego\\_por\\_goteo.pdf](https://inta.gob.ar/sites/default/files/inta_manual_riego_por_goteo.pdf%0Afile:///C:/Users/SURI/OneDrive/Documentos/inta_manual_riego_por_goteo.pdf)
- López, C. (n.d.). Estudios de los suelos salinos y su rehabilitación en la costa peruana. *Universidad Nacional Agraria La Molina*, 1, 1–16.
- Mendoza, A. (2013). *RIEGO POR GOTEO*. [https://www.academia.edu/7433003/Riego\\_por\\_Goteo\\_2013](https://www.academia.edu/7433003/Riego_por_Goteo_2013)
- Munsell. (n.d.). *Tablas de color Munsell*. Microsoft Edge. Retrieved January 30, 2023, from [http://edafologia.ugr.es/programas\\_suelos/practclas/abcsol/comun/munsells.htm](http://edafologia.ugr.es/programas_suelos/practclas/abcsol/comun/munsells.htm)
- Naylamp Mechatronics. (2021a). *Bomba de agua D1/2" 12VDC 5m 800L/h*. <https://naylampmechatronics.com/bombas-de-agua/446-bomba-de-agua-d12-12vdc-5m-800lh.html>
- Naylamp Mechatronics. (2021b). *Driver Mosfet IRF520 6A*. <https://naylampmechatronics.com/drivers/239-driver-mosfet-irf520-6a.html>
- Naylamp Mechatronics. (2021c). *Fuente de alimentación Conmutada 12v 20a*. Google. <https://naylampmechatronics.com/fuentes-switching-ac-dc/628-fuente-de-alimentacion-conmutada-acdc-250w-12v-20a-wode.html>
- Naylamp Mechatronics. (2021d). *Fuente de alimentación DC 12v 1a*. Google. <https://naylampmechatronics.com/fuentes-switching-ac-dc/831-fuente-de-alimentacion-dc-12v-1a.html>
- Naylamp Mechatronics. (2021e). *Módulo lector de memoria Micro SD Card - SPI*. <https://naylampmechatronics.com/almacenamiento/104-modulo-lector-de-memoria-micro-sd-card.html>
- Naylamp Mechatronics. (2021f). *Módulo Relay, 4 canales 5VDC*. <https://naylampmechatronics.com/drivers/152-modulo-relay-4-canales-5vdc.html>
- Naylamp Mechatronics. (2021g). *Sensor de flujo de agua 1/2" YF-S201*. <https://naylampmechatronics.com/sensores-liquido/108-sensor-de-flujo-de-agua-12-yf-s201.html>

- Naylamp Mechatronics. (2021h). *Sensor de Humedad de Suelo Capacitivo*. <https://naylampmechatronics.com/sensores-temperatura-y-humedad/538-sensor-de-humedad-del-suelo-capacitivo.html>
- Naylamp Mechatronics. (2021i). *Sensor Ultrasonido HC-SR04*. <https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- Naylamp Mechatronics. (2021j). *Tutorial sensor de flujo de agua*. Naylampmechatronics.Com. <https://naylampmechatronics.com/blog/47-tutorial-sensor-de-flujo-de-agua.html>
- Naylamp Mechatronics. (2021k). *Válvula solenoide 1/2" 12VDC (NC)*. <https://naylampmechatronics.com/valvulas/314-valvula-solenoide-1p2-pulg-12vdc-nc.html>
- OMM. (2018). *Guía de prácticas climatológicas* (n°100). Organización Mundial de Meteorología. [https://library.wmo.int/doc\\_num.php?explnum\\_id=10027](https://library.wmo.int/doc_num.php?explnum_id=10027)
- ORBES AGRÍCOLA SAC. (2020). *MicroFlapper* (p. 1). <https://orbesagricolasac.com/tienda/riego-por-goteo/gotero-microflapper-autocompensado/>
- Pastor, R. (2005). *PROPIEDADES FISICAS DEL SUELO* (p. 95).
- Pizarro, F. (1996). *Riegos Localizados de Alta Frecuencia* (3ra, Edici ed.). 1986, Ediciones Mundi-Prensa. <https://es.scribd.com/document/494675214/312379749-Pizarro-Riegos-Localizados-de-Alta-Frecuencia-1-Copia#>
- Prometec. (n.d.). *GUIA A LA PROGRAMACION DEL ARDUINO | Funciones*. Retrieved July 2, 2022, from <https://www.prometec.net/funciones1/>
- Quispe, A., Montalvo, N., & Ugás, R. (2016). *International pepper conference 23rd-2016*. 26.
- Sarandón, S., & Flores, C. (2014). *Agroecología: bases teóricas para el diseño y manejo de Agroecosistemas sustentables* (S. Sarandón & C. Flores, Eds.; primera). Editorial de la Universidad de La Plata. [http://sedici.unlp.edu.ar/bitstream/handle/10915/37280/Documento\\_completo.pdf?sequence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/37280/Documento_completo.pdf?sequence=1&isAllowed=y)

- Sepúlveda, E., & Zazueta, F. (2004). Dimensiones del Bulbo Húmedo en Riego Localizado. *IFAS EXTENSION*, 2. <https://ufdcimages.uflib.ufl.edu/IR/00/00/15/35/00001/AE26400.pdf>
- Siura, S. (2007). *Pequeña Agricultura y Riego Tecnificado* (p. 32). <https://idoc.pub/download/pequea-agricultura-y-riego-tecnificado-lurin-19n086jxqk4v>
- Terrile, R., & IPES-RUAF. (2010). *Biopreparados para el manejo sostenible de plagas y enfermedades en la agricultura urbana y periurbana* (1ra Edicion). <https://www.ipes.org/index.php/publicaciones/download/2-agricultura-urbana/53-biopreparados-para-el-manejo-sostenible-de-plagas-y-enfermedades-en-la-agricultura-urbana-y-periurbana>
- Torrente, O. (2013). *ARDUINO: Curso práctico de formación* (Primera ed). Alfaomega Grupo Editor, S.A. De C.V., Mexico. <https://www.latecnicalf.com.ar/descargas/material/proyectointegrador/Arduino%20Curso%20Pr%C3%A1ctico%20de%20formaci%C3%B3n.pdf>
- Ugás, R., Siura, S., Delgado de la Flor, F., Casas, A., & Toledo, J. (2000). Hortalizas: Datos básicos. In *Programa de Hortalizas* (1Ra, Edici ed., Vol. 1, Issue 3). Programa de Hortalizas-Universidad Nacional Agraria La Molina. <http://www.lamolina.edu.pe/hortalizas/Datosbasicos.html>
- Vásquez, A., Vásquez, I., Vásquez, C., & Cañamero, M. (2017). Fundamentos de la Ingeniería de Riegos. In *Mop* (Q & P Impres, Vol. 1). Fondo Editorial- Universidad Nacional Agraria La Molina. <https://www.hidrosm.com/2018/05/fundamentos-de-la-ingenieria-de-riegos.html>
- Zacetrex Technologies. (2022). *Arduino Leonardo R3 ATMEGA32U4*. <https://zacetrex.com/product/arduino-due-r3-at91sam3x8e-cable-usb-copia/>

## VIII. ANEXOS

Anexo 1: Caracterización física del suelo



Extracción de muestra a 30 cm



Muestra a examinar



Determinación textural al tacto

Anexo 2: Resumen de las propiedades físicas del suelo

Textura del suelo	Total de espacio poroso o porosidad total (%)	Densidad aparente $D_{ap}$ (gr/cm <sup>3</sup> )	Capacidad de campo $\theta_{cc}$ (%)	Punto de marchitez permanente $\theta_{pmp}$ (%)	Humedad total aprovechable		
					Peso seco $(\theta_{cc} - \theta_{pmp})$ (%)	Volumen $(\theta_{cc} - \theta_{pmp}) \cdot D_{ap}$ (%)	cm/m
Arenoso	38 (32-42)	1.65 (1.55-1.80)	9 (6-12)	4 (2-6)	5 (4-6)	8 (6-10)	8 (6-10)
Franco arenoso	43 (40-47)	1.50 (1.40-1.60)	14 (10-18)	6 (4-8)	8 (6-10)	12 (9-15)	12 (9-15)
Franco	47 (43-49)	1.40 (1.35-1.50)	22 (18-26)	10 (8-12)	12 (10-14)	17 (14-20)	17 (14-20)
Franco arcilloso	49 (47-51)	1.35 (1.30-1.40)	27 (23-31)	13 (11-15)	14 (12-16)	19 (16-22)	19 (16-22)
Arcilloso arenoso	51 (49-53)	1.30 (1.25-1.35)	31 (27-35)	15 (13-17)	16 (14-18)	21 (18-23)	21 (18-23)
Arcilloso	53 (51-55)	1.25 (1.20-1.30)	35 (31-39)	17 (15-19)	18 (16-20)	23 (20-25)	23 (20-25)

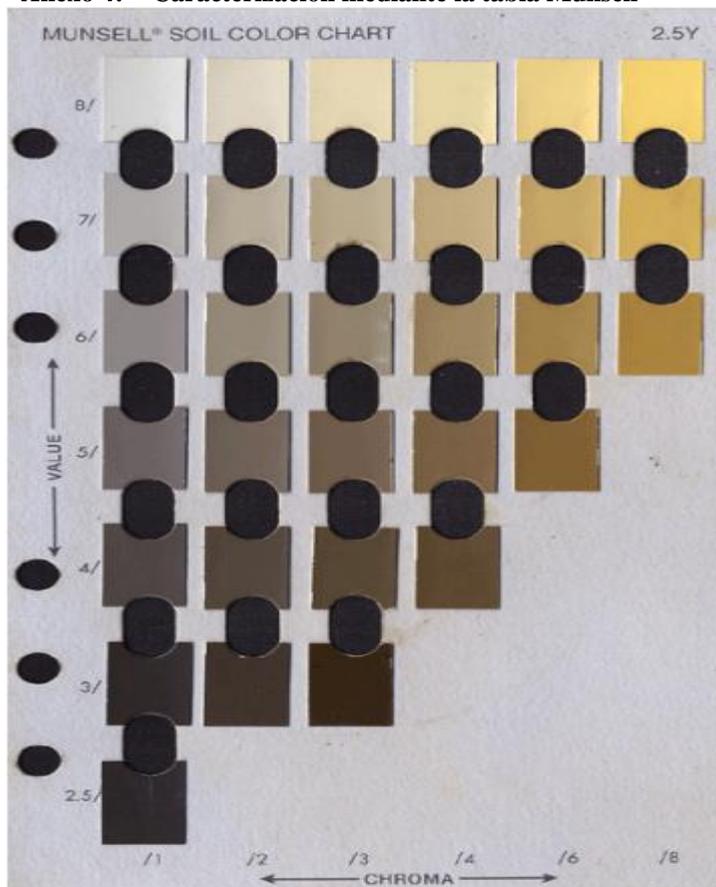
FUENTE: (Vásquez et al., 2017)

Anexo 3: Determinación de la textura al tacto

TEXTURA	SIMBOLO	CARACTERÍSTICAS
Arenosa	A	Al comprimirlo con los dedos se siente áspero.
Arenosa-Franca	AF	Áspero, forma bolas que desmenuzan fácilmente; mancha ligeramente los dedos.
Franco-arenosa	FA	Forma bolas poco resistentes; mancha ligeramente los dedos.
Franca	F	Forma bolas resistentes, mancha los dedos pero no forma cinta.
Franco-limosa	FL	Forma bolas que no se rompen y una cinta rizada
Limosa	L	Talcoso y jabonoso pero no pegajoso.
Franco-arcillo-arenosa	FArA	Algo pegajoso, plástico y mancha los dedos.
Franco-arcillosa	FAr	Pegajoso, mancha los dedos, forma bolas resistentes al manipuleo y cintas que se rompen fácilmente.
Franco-arcillosa-limosa	FArL	Algo plástico, forma una cinta rizada.
Arcillo-arenosa	ArA	Pegajoso, plástico y áspero.
Arcillo-limosa	ArL	Suave y liso.
Arcillosa	Ar	Forma bolas firmes, cintas delgadas y firmes.

FUENTE: (Pastor, 2005)

Anexo 4: Caracterización mediante la tabla Munsell



FUENTE: (Munsell, n.d.)

Anexo 5: Análisis de agua - Rutina

**UNIVERSIDAD NACIONAL AGRARIA LA MOLINA**  
**FACULTAD DE INGENIERÍA AGRÍCOLA**  
 DEPARTAMENTO DE RECURSOS HÍDRICOS DRH  
**LABORATORIO DE AGUA, SUELO, MEDIO AMBIENTE Y FERTIRRIEGO**  
 Av. La Molina s/n. Telf.: 614 7800 Anexo 226 / 349 3969 Lima. E-mail: las-fla@lamolina.edu.pe

## ANÁLISIS DE AGUA - RUTINA

SOLICITANTE : LUIS MIGUEL HUAMÁN GUTIÉRREZ  
 PROYECTO : DISEÑO DE UN SISTEMA DE RIEGO EN "PALMERAS DE HUARANGAL"  
 RESPONSABLE ANÁLISIS : Ing. Nore Arévalo Flores  
 FECHA DE ANÁLISIS : La Molina, 11 de febrero de 2022

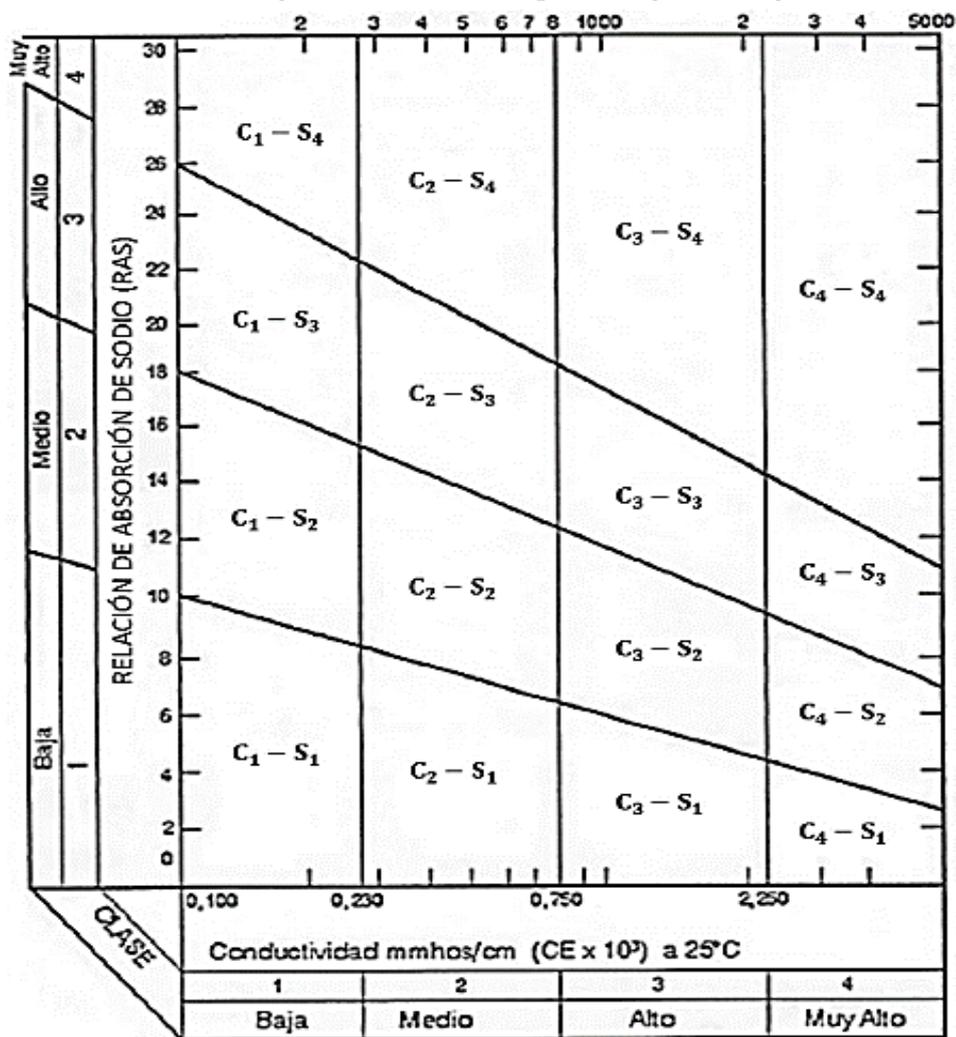
N° LABORATORIO	10124
N° DE CAMPO	Muestra N°1
CE mmhos/cm	3.34
pH	7.00
Calcio meq/l	16.61
Magnesio meq/l	3.93
Sodio meq/l	11.74
Potasio meq/l	0.24
SUMA DE CATIONES	32.53
Cloruro meq/l	19.97
Sulfato meq/l	7.41
Bicarbonato meq/l	5.17
Nitratos meq/l	0.46
Carbonatos meq/l	0.00
SUMA DE ANIONES	33.01
SAR	3.66
CLASIFICACION	C4-S2
Boro mg/L	0.88



Eusebio Ingol Blanco, PhD.  
 JEFE DE LABORATORIO



Anexo 6: Diagrama de clasificación para las aguas de riego



**Peligro de salinidad**

FUENTE:(Vásquez et al., 2017)

Anexo 7: Límites tolerantes del boro el agua de riego

CLASE DE AGUA POR CONCENTRACIÓN DE BORO	CULTIVOS		
	SENSIBLES (ppm)	SEMI TOLERANTES (ppm)	TOLERANTES (ppm)
1	0.33	0.67	1.00
2	0.33 - 0.67	0.67 - 1.33	1.00 - 2.00
3	0.67	1.33 - 2.00	2.00 - 3.00
4	1.25	2.50	3.75

FUENTE: (Vásquez et al., 2017)

**Anexo 8: Coeficiente de fricción en función del material de la tubería**

<b>Material Tubería</b>	<b>C</b>
Plástico	150
Asbesto - cemento	140
Aluminio (con o acoples)	130
Acero galvanizado	130
Acero nuevo	130

FUENTE:(Mendoza, 2013)

**Anexo 9: Coeficiente de Christiansen según el número de salidas**

<b>No de salidas</b>	<b>f</b>	<b>No de salidas</b>	<b>f</b>
1	1.000	18	0.379
2	0.639	19	0.377
3	0.535	20	0.376
4	0.486	22	0.374
5	0.457	24	0.372
6	0.435	26	0.370
7	0.425	28	0.369
8	0.415	30	0.368
9	0.409	35	0.365
10	0.402	40	0.364
11	0.397	50	0.361
12	0.394	60	0.359
13	0.391	80	0.357
14	0.387	100	0.356
15	0.384	150	0.354
16	0.382	300	0.353
17	0.380	>300	0.351

FUENTE:(Mendoza, 2013)

**Anexo 10: Especificaciones técnicas placa Arduino Leonardo**

Características – rangos y valores

- Microcontrolador: ATmega32u4
- Chip USB: ATmega16U2
- Conector USB: Micro-USB
- Voltaje de operación: 5V DC
- Voltaje de alimentación: 6V - 20V DC (7-12V recomendado)
- Pines digitales I/O: 20 (7 salidas PWM)
- Entradas analógicas: 12
- Corriente entrada/salida por pin: 40mA máx.
- Memoria FLASH: 32KB (4KB usados por el Bootloader)
- Memoria SRAM: 2.5KB
- Memoria EEPROM: 1KB
- Frecuencia de reloj: 16MHz
- Diseño Arduino original

- Incluye: Cable USB 50cm
  - Dimensiones: 68\*53\*13 mm y 20 g peso
- 

FUENTE:(Zacetrex Technologies, 2022)

#### **Anexo 11: Especificaciones técnicas de sensor capacitivo**

Características – rangos y valores

---

- Voltaje de alimentación: 3.3V - 5V DC
  - Corriente operación: 5mA
  - Voltaje de la señal de salida: 0 a 5V (Analógico)
  - Modelo: capacitive soil moisture sensor v1.2
  - Vida útil: 3 años mín.
  - Conector: PH 2.0-3P
  - Incluye: Electrodo y cable jumper hembra
  - Dimensiones: 98\*23 mm
  - Peso: 15 gramos
- 

FUENTE: (Naylamp Mechatronics, 2021h)

#### **Anexo 12: Especificaciones técnicas de sensor de flujo**

Características – rangos y valores

---

- Modelo: YF-S201
  - Voltaje de operación: 5V - 18V DC
  - Consumo de corriente: 15mA (5V)
  - Capacidad de carga: 10mA (5 VDC)
  - Salida: Onda cuadrada pulsante
  - Rango de Flujo: 1-30L/min
  - Volumen promedio por pulso: 2.25mL
  - Pulsos por litro: 450
  - Factor de conversión: 7.5
  - Rosca externa: 1/2" NPS
  - Presión de trabajo máx.: 1.75MPa (17 bar)
  - Temperatura de funcionamiento: -25°C a 80°C
  - Material: Plástico color negro
- 

FUENTE: (Naylamp Mechatronics, 2021g)

### **Anexo 13: Especificaciones técnicas del sensor ultrasónico**

---

#### Características – rangos y valores

---

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

FUENTE: (Naylamp Mechatronics, 2021i)

### **Anexo 14: Especificaciones técnicas de modulo relay**

---

#### Características – rangos y valores

---

- Voltaje de operación: 5V DC (bobina Relay)
  - Señal de Control: TTL (3.3V o 5V)
  - Para activar salida NO: 0 Voltios
  - N° de Relays (canales): 4 CH
  - Modelo Relays: Songle SRD-05VDC-SL-C
  - Capacidad máx: 10A/250VAC, 10A/30VDC
  - Corriente máx: 10A (NO), 5A (NC)
  - Tiempo de acción: 10 ms / 5 ms
  - Entradas Optoacopladas
  - Indicadores LED de activación
- 

FUENTE:(Naylamp Mechatronics, 2021f)

### **Anexo 15: Especificaciones técnicas Driver Mosfet IRF520N**

---

#### Características – rangos y valores

---

- Voltaje de salida: 0-24 VDC
  - Voltaje de control: 5V TTL
  - Corriente máxima: 9A
  - Corriente nominal: 6A
  - Para cargas mayor a 6A usar disipador
- 

FUENTE: (Naylamp Mechatronics, 2021b)

### **Anexo 16: Especificaciones técnicas de válvula solenoide**

#### Características – rangos y valores

---

- Voltaje de operación: 12V DC
  - Corriente de operación: 0.6A
  - Potencia consumo: 8W
  - Temperatura de funcionamiento: 5°C a 100°C
  - Presión de funcionamiento mínima: 0.02 MPa (0.2 Bar = 2.04 mca)
  - Presión de funcionamiento máximo: 0.8 MPa (8 Bar = 81.6 mca)
  - Tiempo de respuesta (apertura):  $\leq 0.15$  s
  - Tiempo de respuesta (cerrado):  $\leq 0.3$  s
  - Conector tubería: Rosca externa 1/2" NPS Macho
  - Reposo: Normalmente cerrado
  - Tipo de válvula: Diafragma
  - Adecuado para agua y fluidos de baja viscosidad
  - No se recomienda para aplicaciones que usan solo la gravedad, por la presión mínima de funcionamiento
  - Material cuerpo: Plástico ABS
  - Dimensiones: 85\*60\*26 mm
- 

FUENTE: (Naylamp Mechatronics, 2021k)

### **Anexo 17: Especificaciones técnicas Bomba de agua**

#### Características – rangos y valores

---

- Voltaje de Operación: 5-12V DC
  - Corriente máxima: 350mA
  - Potencia: 19W
  - Caudal máximo: 800L/h (13L/min)
  - Columna de agua máxima: 5m
  - Conexión rosca: G1/2"
  - Material ABS Negro
  - Protección: IP68
  - Ruido: <40dB a 0.5m
  - Temperatura del fluido: menor a 100°C
  - Tamaño 8x8x6.5cm
  - Peso: 220gr
- 

FUENTE:(Naylamp Mechatronics, 2021a)

### **Anexo 18: Especificaciones técnicas de Micro SD**

Características – rangos y valores

- Interfaz: SPI
- Voltaje de operación: 3.3 V-5.0 V
- Pines de conexión: MOSI, MISO, SCK,CS (SPI)
- Compatibilidad: Arduino, PIC

FUENTE:(Naylamp Mechatronics, 2021e)

### **Anexo 19: Especificaciones del Aplicativo ArduinoDroid**

Características generales

- Aplicación ocupa 500mb de memoria interna del celular
- Requiere Android versión 6.0 en adelante
- Contiene ejemplos y bibliotecas incluidas
- Contiene resaltador para la sintaxis de los códigos
- Contiene teclado de software
- Carga bocetos a través de cable USB
- Compatible con todas las placas de Hardware Arduino (uno, leonardo, mega, nano...etc.)
- Funciona sin conexión a internet
- Contiene monitor serie
- Contiene soporte Dropbox y Drive

FUENTE: (Anton Smirnov, 2021)

### **Anexo 20: Código de lectura de módulo micro SD**

```
////////*****LECTOR MICROCD*****
#include <SPI.h> // librería para la comunicación
#include <SD.h> // librería para la lectura
int SSpin= 4;
File Archivo;

void setup() {

    Serial.begin(9600); // iniciamos la comunicación serial entre el Arduino
y la microSD

    while (!Serial) {
        ; // esperar a que se conecte el puerto serial. Necesario solo para el
puerto usb nativo
    }
    Serial.print("Inicializando tarjeta....."); // imprimimos el mensaje en
el monitor serial
    delay(1000); // demora de 1 segundo
    if (!SD.begin(SSpin)) { // de no cumplirse, este emite una señal
contraria
        Serial.println("fallo en inicialización!");
        delay(1000);
        while (1); // obliga a que el ciclo se dé como verdadero
    }
    Serial.println("inicialización realizada"); // si el monitor serie
inicio correctamente
    delay(1000);
}
```

```

    if (SD.exists("example.txt")) { // se verifica que exista el archivo
        Serial.println("example.txt si existe."); // si existe el archivo en
la memoria se lee esto
        delay(1000);

    } else {
        Serial.println("example.txt no existe."); // si no existe el archivo
en la memoria se digita esto
        delay(1000);
    }
    Serial.println("creando archivo example.txt..."); // si no existe el
archivo, creamos un nuevo
    delay(1000);
    Archivo= SD.open("example.txt", FILE_WRITE); // abrimos el archivo
donde se guardarán los datos

    if(Archivo) { // si es verdadero se imprime esto, si es falso se
imprimirá el otro de abajo
        Archivo.println("probando 1, 2, 3, 4, 5, 6"); // datos se imprimirán en
la tarjeta
        Serial.println("escribimos en el archivo example.txt"); // datos se
imprimiran en el monitor serie

        Archivo.close(); // solo deseamos escribir una sola línea y cerrar

        Serial.println("escritura correcta");

    } else {
        Serial.println("error en apertura de example.txt"); // si es que hubo
error en la apertura del archivo

    }
}
}
void loop() {
    // nothing happens after setup finishes.
}

```

#### **Anexo 21: Código de lectura de electroválvula-válvula solenoide**

```

int Evalvula= 10; //creamos una variable para la electroválvula

void setup() {
    pinMode(Evalvula, OUTPUT); // declaramos como pin de salida electroválvula
}
void loop() {
    digitalWrite(Evalvula,HIGH); // se mantiene apagada la electroválvula
    delay (20000); // tiempo apagado 20 segundos

    digitalWrite(Evalvula,LOW); // se enciende la electroválvula
    delay (5000); // tiempo encendido 5 segundos
}

```

#### **Anexo 22: Código de lectura para la bomba de agua**

```

#include <SPI.h> // librería para la comunicación

int Bomba_agua= 6; //creamos una variable para la electroválvula

void setup() {

    pinMode(Bomba_agua, OUTPUT); // declaramos como pin de salida
electroválvula
}

```

```

}

void loop() {

analogWrite(Bomba_agua,255); // se mantiene encendido la bomba
delay (60000); // tiempo apagado 60 segundos

analogWrite(Bomba_agua,0); // se apaga la bomba
delay (60000); // tiempo encendido 60 segundos

}

```

#### **Anexo 23: Código para la calibración de sensor de humedad de suelo**

`int SHumedad_C = A0 ;` // variable entera para el Pin analógico "A0" donde conectaremos la salida del Sensor Humedad capacitivo

`int valor_sensor;` // Variable para almacenar los valores de salida del sensor de humedad capacitivo

```

void setup()
{
  Serial.begin(9600); // Comienzo de la comunicación con el monitor serie
del IDE de Arduino
}
void loop()
{
  // Leemos el valor de la salida analógica del sensor capacitivo,
conectada al pin analógico "A0"
  valor_sensor = analogRead(SHumedad_C);

  // Mostramos el valor de la salida analógica del sensor capacitivo a
través del monitor serie
  Serial.print("Valor del sensor de humedad capacitivo: ");
  Serial.println(valor_sensor);

  delay(1000); // Retardo de un segundo entre lecturas del sensor de
humedad capacitivo
}

```

#### **Anexo 24: Código para la lectura del sensor de humedad de suelo**

`int SHumedad_C = A0;` // variable entera para el Pin analógico "A0" donde conectaremos la salida del Sensor Humedad capacitivo

`const int Valor_Aire = 485;` // Valor calculado del Sensor aire  
`const int Valor_Agua = 207;` // Valor calculado del Sensor agua

`int Valor_Sensor = 0;` // Variable para almacenar los valores de salida del sensor de humedad capacitivo desde 0

`int porcentaje = 0;` // Variable para almacenar el porcentaje de humedad relativa del terreno desde 0

```

void setup()
{
  Serial.begin(9600); // se da inicio a la comunicación con el monitor
serie del IDE de Arduino
}
void loop()
{
  // Leemos el valor de la salida analógica del sensor capacitivo,
conectada al pin analógico "A0"
  Valor_Sensor = analogRead(SHumedad_C);

```

```

// Se calcula el porcentaje de humedad relativa teniendo en cuenta los
dos límites
porcentaje = map(Valor_Sensor, Valor_Agua, Valor_Aire, 100, 0);

if(porcentaje < 0) porcentaje = 0; // Evita porcentajes negativos en la
medida del sensor

if(porcentaje > 100) porcentaje = 100; // Evita porcentajes negativos
en la medida del sensor

// Se presenta el porcentaje a través del monitor serie
Serial.print("HUMEDAD: ");
Serial.print(porcentaje);
Serial.println("% HR");

// Se presentan los mensajes, a través del monitor serie, dependiendo
del porcentaje de humedad relativa
if(porcentaje <= 33)
{
  Serial.println("Suelo seco !!!");
}

if(porcentaje > 33 && porcentaje <= 66)
{
  Serial.println("Suelo húmedo !!!");
}

if(porcentaje > 66)
{
  Serial.println("Suelo con exceso de humedad !!!");
}

Serial.println(" ");

delay(1000); // Retardo de un segundo entre lecturas del sensor de
humedad capacitivo
}

```

#### Anexo 25: Código para la calibración del sensor de flujo

```

//***sensor flujo***
// calibración de sensor de flujo
volatile long NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 2; //Sensor conectado en el pin 2 o 3 que solo aceptan
interrupciones
int Evalvula4 = 5;
//*** bomba*****
int Bomba= 6; //creamos una variable para la electroválvula
//---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(Bomba, OUTPUT); // declaramos como pin de salida electroválvula
  pinMode ( Evalvula4, OUTPUT);
  pinMode(PinSensor, INPUT); // sensor como entrada
  attachInterrupt(digitalPinToInterrupt(PinSensor),ContarPulsos,RISING);//
  //(Interrupción 0(Pin2),función, Flanco de subida) // contar los pulsos
  desde 0
}

```

```

    interrupts(); //Habilitamos las interrupciones
}
void loop ()
{
    digitalWrite(Bomba,HIGH); // se enciende la electroválvula
    digitalWrite(Evalvula4, LOW);

    //-----Enviamos por el puerto serie-----
    Serial.print ("Numero de Pulsos = ");
    Serial.println (NumPulsos);
    delay(1000);
}

```

#### Anexo 26: Código para la lectura del sensor de flujo

```

//*****sensor de flujo
const int Sflujo = 3; // variable donde estará el pin del sensor de
flujo
const int measureInterval = 1000; // intervalo de las interrupciones
volatile int pulseConter; // variable para el conteo de pulsos

// YF-S201
const float factorK = 7.5; // factor de coeficiente propio de cada sensor,
sale de la calibración del sensor

// FS300A
//const float factorK = 5.5;

// FS400A
//const float factorK = 3.5;

float volumen = 0; // variable de tipo decimal para el volumen que empieza
desde cero

long t0 = 0; // variable de tiempo de tipo entera que empieza desde cero

void ISRCountPulse() // función para el conteo de pulsos
{
    pulseConter++; // conteo de pulsos de manera creciente que se va
sumando con el anterior
}

float GetFrequency()
{
    pulseConter = 0; // conteo de pulsos desde 0

    interrupts();
    delay(measureInterval); // toma de interrupciones cada 1 segundo
    noInterrupts();

    return (float)pulseConter * 1000 / measureInterval;
}

void SumVolumen(float dV) // función para el registro del volumen
{
    volumen += dV / 60 * (millis() - t0) / 1000.0;
    t0 = millis();
}

void setup()
{
    Serial.begin(9600); // comunicación serial con Arduino

```

```

    attachInterrupt(digitalPinToInterrupt(Sflujo), ISRCountPulse, RISING);
    t0 = millis();
}

void loop()
{
    float frequency = GetFrequency(); // variable para la frecuencia
    (obtener frecuencia en Hz )

    float flow_Lmin = frequency / factorK; // variable para calcular
    caudal L/min
    SumVolumen(flow_Lmin);

    Serial.print(" Caudal: "); // se lee el en el monitor serial el caudal
    y el volumen
    Serial.print(flow_Lmin, 3);
    Serial.print(" (L/min)\tConsumo:");
    Serial.print(volumen, 1);
    Serial.println(" (L)");
}

```

#### Anexo 27: Código para la lectura del sensor de ultrasonido

```

const int Trigger = 8; //Pin digital 2 para el Trigger del sensor
const int Echo = 9; //Pin digital 3 para el Echo del sensor

void setup() {
    Serial.begin(9600); //inicializamos la comunicación
    pinMode(Trigger, OUTPUT); //pin como salida
    pinMode(Echo, INPUT); //pin como entrada
    digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
}

void loop()
{
    long t; //tiempo que demora en llegar el eco
    long d; //distancia en centímetros

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

    t = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
    d = t/59; //escalamos el tiempo a una distancia en cm

    Serial.print("Distancia: ");
    Serial.print(d); //Enviamos serialmente el valor de la distancia
    Serial.print("cm");
    Serial.println();
    delay(1000); //Hacemos una pausa de 100ms
}

```

#### Anexo 28: Código ejecutado con el aplicativo Arduinodroid en el prototipo riego automatizado

```

//////// Arduino Leonardo ///////////
////////*****LECTOR_MICROCD*****

#include <SPI.h>
#include <SD.h>
int SSpin = 4;
File Archivol ;

```

```

File Archivo2 ; // Creamos archivos para almacenar los datos de humedad
de suelo y consumo de agua
File Archivo3 ; // de cada una de las 4 áreas del proyecto (1,2,3,4)
File Archivo4 ;

//***** Sensor de humedad de suelo capacitivo
#define SHumedad_C1 A5
#define SHumedad_C2 A1 // variables para los Pin analógico "A0,A1, A2,A3"
#define SHumedad_C3 A3 // para conectar la salida del sensor de humedad
capacitivo
#define SHumedad_C4 A0

int Porcentaje1;
int Porcentaje2; // variable para el porcentaje de humedad presente en el
suelo de cada área
int Porcentaje3;
int Porcentaje4;

// *****electroválvula
#define Evalvula1 5
#define Evalvula2 10 //variable para las electroválvulas
#define Evalvula3 11 // de cada una de las áreas
#define Evalvula4 12

// *****Bomba
#define Bomba 6 //creamos una variable para la bomba

// *****Sensor ultra sonido
const int Trigger = 8; // Pin digital 8 para el Trigger del sensor
const int Echo = 9; //Pin digital 9 para el Echo del sensor , pueden
estar conectados en cualquier pin digital

// ****variables para manejo de función millis****///
unsigned long tiempo = 0; //
unsigned long Evalvulatiempo1 = 0;
unsigned long Evalvulatiempo2 = 0;
unsigned long Evalvulatiempo3 = 0;
unsigned long Evalvulatiempo4 = 0;
unsigned long Area1 = 0;
unsigned long Area2 = 0;
unsigned long Area3 = 0;
unsigned long Area4 = 0;
int interval0 = 1000; // intervalo de tiempo para cada sensor
int interval_1 = 1000;
int interval1 = 1000;
int interval_2 = 1000; // tiempo transcurrido para la electroválvula 2
int interval2 = 1000;
int interval_3 = 1000;
int interval3 = 1000;
int interval_4 = 1000;
int interval4 = 1000;

volatile int conteo_pulsos1;
volatile int conteo_pulsos2;
volatile int conteo_pulsos3; // variable para la Cantidad de pulsos de
cada sensor de flujo.
volatile int conteo_pulsos4; //Como se usa dentro de una interrupción debe
ser volatile

unsigned long pulsosAcumulados1;

```

```

unsigned long pulsosAcumulados2; // Pulsos acumulados para cada sensor de
flujo
unsigned long pulsosAcumulados3;
unsigned long pulsosAcumulados4;

unsigned char sensorFlujo1 = 2;
unsigned char sensorFlujo2 = 3; // Pines a los que están conectados
unsigned char sensorFlujo3 = 0; // los sensores de flujo de cada área
unsigned char sensorFlujo4 = 1;

unsigned long tiempoUltrasonido; // Para calcular el tiempo transcurrido
en el sensor ultrasonido

float litrosPorMin1;
float litrosPorMin2; // cálculo de velocidad de flujo de cada sensor
float litrosPorMin3;
float litrosPorMin4;

float volumen1 = 0;
float volumen2 = 0; // Litros acumulados por cada sensor
float volumen3 = 0;
float volumen4 = 0;

void flujo1 ()
{
  conteo_pulsos1++;
}
void flujo2 () // Función de interrupción
{
  conteo_pulsos2++; // Simplemente sumar el número de pulsos
}
void flujo3 ()
{
  conteo_pulsos3++;
}
void flujo4 ()
{
  conteo_pulsos4++;
}
void setup()
{
  pinMode(Evalvula1, OUTPUT);
  pinMode(Evalvula2, OUTPUT); // declaramos como pin de salida la
electroválvula 2
  pinMode(Evalvula3, OUTPUT);
  pinMode(Evalvula4, OUTPUT);
  pinMode(Bomba, OUTPUT); // declaramos como pin de salida bomba
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT); //pin como entrada

  attachInterrupt(digitalPinToInterrupt(sensorFlujo1), flujo1, RISING);
  attachInterrupt(digitalPinToInterrupt(sensorFlujo2), flujo2,
RISING); // datos utilies para el sensor de flujo
  attachInterrupt(digitalPinToInterrupt(sensorFlujo3), flujo3, RISING);
  attachInterrupt(digitalPinToInterrupt(sensorFlujo4), flujo4, RISING);
  if (!SD.begin(SSpin)) {
    delay(1000);
    while (1);
  }
}
void loop ()

```

```

{
  tiempo = millis(); // habilitamos función millis
////////*****SENSOR ULTRASÓNICO*****////////

  if (tiempo - tiempoultrasonido >= intervalo)
  {
    tiempoultrasonido = tiempo;

    long tiem; //tiempo que demora en llegar el eco
    long dista; //distancia en centímetros

    digitalWrite(Triple, HIGH);
    digitalWrite(Triple, LOW);

    tiem = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
    dista = tiem / 59; //escalamos el tiempo a una distancia en cm

    if (dista >= 100) { // si la distancia medida por el sensor
ultrasónico es mayor de 100 cm
      digitalWrite(Bomba, LOW); // bomba se apaga, por q cuando no está
energizado la bomba no funciona
      return ;
    }
    if (dista < 100) { // si la distancia medida por el sensor ultrasónico
es menor de 100 cm

      if (Porcentaje1 || Porcentaje2 || Porcentaje3 || Porcentaje4 <= 50
)
// y si por el contrario ocurre este hecho entonces se procede así
      {
        digitalWrite(Bomba, HIGH); // bomba se enciende, mandamos una señal alta
      }
      if (Porcentaje1 && Porcentaje2 && Porcentaje3 && Porcentaje4 >50 ) // y
si por el contrario ocurre este hecho entonces se procede así
      {
        digitalWrite(Bomba, LOW); // bomba se apaga, mandamos una señal baja.
      }
    }
  }
}
////////*****ÁREA DE TRABAJO 1 *****////////

  if (tiempo - Areal >= (interval1)) // tiempo de control de millis para
el área 1( guardar en archivo, flujo de agua, humedad de suelo)
  {
    Areal = tiempo;

    Archivo1 = SD.open("ZONA_1.txt", FILE_WRITE);
    float factor_K1 = 15.7;
    pulsosAcumulados1 += conteo_pulsos1 ;
    litrosPormin1 = ((conteo_pulsos1 ) / factor_K1 ); // (Pulse frequency
x 60 min) / 7.5Q = flow rate in L/hour
    conteo_pulsos1 = 0;
    volumen1 = pulsosAcumulados1 * 1.0 / 941; //para el sensor 1° 941
pulsos

    int Valor_sensor1 = analogRead(SHumedad_C1); // variable local para
el valor registrado del sensor
    const int V_Aire1 = 554; // Valor calculado con el programa de
calibración con el sensor al aire
    const int V_Agua1 = 323; // Valor calculado con el programa de
calibración con el sensor sumergido en agua

```

```

    Porcentaje1 = map(Valor_sensor1, V_Agua1, V_Aire1, 100, 0); // Se
calcula el porcentaje de humedad relativa teniendo en cuenta los dos
límites
    if (Porcentaje1 < 0 ) Porcentaje1 = 0; // Evita porcentajes negativos
en la medida del sensor
    if (Porcentaje1 > 100) Porcentaje1 = 100; // Evita porcentajes
superiores al 100% en la medida del sensor

    if (Porcentaje1 <= 50 )          // si ocurre este hecho entonces se
procede así
    {
        if (tiempo - Evalvulatiempol >= (interval_1))
        {
            Evalvulatiempol = tiempo;
            digitalWrite(Evalvulal, LOW);          // Electvalvula se enciende,
por q está conectado en normalmente cerrado
        }
        Archivo1.print(" Caudal1: ");
        Archivo1.print(litrosPormin1, 2);
        Archivo1.print(" (L/min)\tConsumo:"); // se registran los datos en
la tarjeta SD
        Archivo1.print(volumen1, 3);
        Archivo1.print(" (L)");
        Archivo1.print(" H1(%): ");
        Archivo1.println(Porcentaje1);
    }
    if (Porcentaje1 > 50 )          // condicional , si ocurre este hecho
entonces se procede así
    {
        if (tiempo - Evalvulatiempol >= (interval_1))
        {
            Evalvulatiempol = tiempo;
            digitalWrite(Evalvulal, HIGH);        // Electroválvula se apaga
        }
        Archivo1.print(" H1(%): ");
        Archivo1.println(Porcentaje1); // se registran los datos en la
tarjeta SD
    }
}
Archivo1.close(); // se cierra los archivos guardados

////////*****AREA DE TRABAJO 2 *****////////
if (tiempo - Area2 >= (interval2))
{
    Area2 = tiempo;

    Archivo2 = SD.open("ZONA_2.txt", FILE_WRITE);
    float factor_K2 = 182.5;
    pulsosAcumulados2 += conteo_pulsos2 ;
    litrosPormin2 = ((conteo_pulsos2 ) / factor_K2 );
    conteo_pulsos2 = 0;
    volumen2 = pulsosAcumulados2 * 1.0 / 10944 ;

    int Valor_sensor2 = analogRead(SHumedad_C2);
    const int V_Aire2 = 540;
    const int V_Agua2 = 303;
    Porcentaje2 = map(Valor_sensor2, V_Agua2, V_Aire2, 100, 0);
    if (Porcentaje2 < 0 ) Porcentaje2 = 0;
    if (Porcentaje2 > 100) Porcentaje2 = 100;

    if (Porcentaje2 <= 50 )

```

```

{
  if (tiempo - Evalvulatiempo2 >= (interval_2))
  {
    Evalvulatiempo2 = tiempo;
    digitalWrite(Evalvula2, LOW);
  }
  Archivo2.print(" Caudal2: ");
  Archivo2.print(litrosPormin2, 2);
  Archivo2.print(" (L/min)\tConsumo:");
  Archivo2.print(volumen2, 3);
  Archivo2.print(" (L)");
  Archivo2.print(" H2(%): ");
  Archivo2.println(Porcentaje2);
}
if (Porcentaje2 > 50 )
{
  if (tiempo - Evalvulatiempo2 >= (interval_2))
  {
    Evalvulatiempo2 = tiempo;
    digitalWrite(Evalvula2, HIGH);
  }
  Archivo2.print(" H2(%): ");
  Archivo2.println(Porcentaje2);
}
}
Archivo2.close();

////////*****AREA DE TRABAJO 3 *****////////
if (tiempo - Area3 >= (interval3))
{
  Area3 = tiempo;

  Archivo3 = SD.open("ZONA_3.txt", FILE_WRITE);
  float factor_K3 = 117.7;
  pulsosAcumulados3 += conteo_pulsos3 ;
  litrosPormin3 = ((conteo_pulsos3 ) / factor_K3 );
  conteo_pulsos3 = 0;
  volumen3 = pulsosAcumulados3 * 1.0 / 7057;

  int Valor_sensor3 = analogRead(SHumedad_C3);
  const int V_Aire3 = 547;
  const int V_Agua3 = 335;
  Porcentaje3 = map(Valor_sensor3, V_Agua3, V_Aire3, 100, 0);
  if (Porcentaje3 < 0 ) Porcentaje3 = 0;
  if (Porcentaje3 > 100) Porcentaje3 = 100;

  if (Porcentaje3 <= 50 )
  {
    if (tiempo - Evalvulatiempo3 >= (interval_3))
    {
      Evalvulatiempo3 = tiempo;

      digitalWrite(Evalvula3, LOW);
    }
    Archivo1.print(" Caudal3: ");
    Archivo1.print(litrosPormin3, 2);
    Archivo1.print(" (L/min)\tConsumo:");
    Archivo1.print(volumen3, 3);
    Archivo1.print(" (L)");
    Archivo1.print(" H3(%): ");
    Archivo1.println(Porcentaje3);
  }
}

```

```

}
if (Porcentaje3 > 50 )
{
  if (tiempo - Evalvulatiempo3 >= (interval_3))
  {
    Evalvulatiempo3 = tiempo;
    digitalWrite(Evalvula3, HIGH);
  }
  Archivo3.print(" H3(%): ");
  Archivo3.println(Porcentaje3);
}
}
Archivo3.close();

////////*****AREA DE TRABAJO 3 *****////////
if (tiempo - Area4 >= (interval4))
{
  Area4 = tiempo;

  Archivo4 = SD.open("ZONA_4.txt", FILE_WRITE);
  float factor_K4 = 75.5;
  pulsosAcumulados4 += conteo_pulsos4 ;
  litrosPormin4 = ((conteo_pulsos4 ) / factor_K4 );
  conteo_pulsos4 = 0;
  volumen4 = pulsosAcumulados4 * 1.0 / 4524;

  int Valor_sensor4 = analogRead(SHumedad_C4);
  const int V_Aire4 = 531;
  const int V_Agua4 = 329;
  Porcentaje4 = map(Valor_sensor4, V_Agua4, V_Aire4, 100, 0);
  if (Porcentaje4 < 0 ) Porcentaje4 = 0;
  if (Porcentaje4 > 100) Porcentaje4 = 100;

  if (Porcentaje4 <= 50 )
  {
    if (tiempo - Evalvulatiempo4 >= (interval_4))
    {
      Evalvulatiempo4 = tiempo;

      digitalWrite(Evalvula4, LOW);
    }
    Archivo4.print(" Caudal4: ");
    Archivo4.print(litrosPormin4, 2);
    Archivo4.print(" (L/min)\tConsumo:");
    Archivo4.print(volumen4, 3);
    Archivo4.print(" (L)");
    Archivo4.print(" H4(%): ");
    Archivo4.println(Porcentaje4);
  }
  if (Porcentaje4 > 50 )
  {
    if (tiempo - Evalvulatiempo4 >= (interval_4))
    {
      Evalvulatiempo4 = tiempo;
      digitalWrite(Evalvula4, HIGH);
    }
    Archivo4.print(" H4(%): ");
    Archivo4.println(Porcentaje4);
  }
}
Archivo4.close();

```

```
}
```

### Anexo 29: Código ejecutado con laptop (monitor serial) en el prototipo riego automatizado

```
////////// Arduino Leonardo ////////////  
//////////*****LECTOR_MICROCD*****  
  
#include <SPI.h>  
#include <SD.h>  
int SSpin = 4;  
File Archivo1 ;  
File Archivo2 ;  
File Archivo3 ;  
File Archivo4 ;  
  
//***** Sensor de humedad de suelo capacitivo  
#define SHumedad_C1 A5 // Pin analógico "A0" para conectar la salida del  
sensor de humedad capacitivo  
#define SHumedad_C2 A1 // Pin analógico "A1" para conectar la salida del  
sensor de humedad capacitivo  
#define SHumedad_C3 A3 // Pin analógico "A2" para conectar la salida del  
sensor de humedad capacitivo  
#define SHumedad_C4 A0 // Pin analógico "A3" para conectar la salida del  
sensor de humedad capacitivo  
  
int Porcentaje1;  
int Porcentaje2;  
int Porcentaje3;  
int Porcentaje4;  
  
// *****electroválvula  
#define Evalvula1 5  
#define Evalvula2 10 //creamos una variable para la electroválvula  
#define Evalvula3 11  
#define Evalvula4 12  
  
// *****Bomba  
#define Bomba 6 //creamos una variable para la bomba  
  
// *****Sensor ultra sonido  
const int Trigger = 8; //Pin digital 2 para el Trigger del sensor  
const int Echo = 9; //Pin digital 3 para el Echo del sensor  
  
// ****variables para manejo de funcion millis****///  
unsigned long tiempo = 0;  
unsigned long Evalvulatiempo1 = 0;  
unsigned long Evalvulatiempo2 = 0;  
unsigned long Evalvulatiempo3 = 0;  
unsigned long Evalvulatiempo4 = 0;  
unsigned long Area1 = 0;  
unsigned long Area2 = 0;  
unsigned long Area3 = 0;  
unsigned long Area4 = 0;  
unsigned long interval0 = 1000; // intervalo de tiempo para cada sensor  
unsigned long interval_1 = 1000; // tiempo transcurrido para la  
electroválvula  
unsigned long interval1 = 1000; // tiempo  
unsigned long interval_2 = 1000; // tiempo transcurrido para la  
electroválvula  
unsigned long interval2 = 1000;  
unsigned long interval_3 = 1000; // tiempo transcurrido para la  
electroválvula  
unsigned long interval3 = 1000;
```

```

unsigned long interval_4 = 1000; // tiempo transcurrido para la
electroválvula
unsigned long interval4 = 1000;

volatile int conteo_pulsos1;
volatile int conteo_pulsos2; // Cantidad de pulsos del sensor. Como se usa
dentro de una interrupcion debe ser volatile
volatile int conteo_pulsos3;
volatile int conteo_pulsos4;

unsigned long pulsosAcumulados1;
unsigned long pulsosAcumulados2; // Pulsos acumulados
unsigned long pulsosAcumulados3;
unsigned long pulsosAcumulados4;

unsigned char sensorFlujo1 = 2;
unsigned char sensorFlujo2 = 3; // Pin al que está conectado el sensor
unsigned char sensorFlujo3 = 0;
unsigned char sensorFlujo4 = 1;

unsigned long tiempoultrasonido; // Para calcular el tiempo transcurrido
en el sensor ultrasonido

float litrosPormin1;
float litrosPormin2;
float litrosPormin3; // litros/min
float litrosPormin4;

float volumen1 = 0;
float volumen2 = 0;
float volumen3 = 0; // Litros acumulados
float volumen4 = 0;

void flujo1 () // Función de interrupción
{
  conteo_pulsos1++;
}

void flujo2 () // Función de interrupción
{
  conteo_pulsos2++; // Simplemente sumar el número de pulsos
}
void flujo3 ()
{
  conteo_pulsos3++;
}

void flujo4 ()
{
  conteo_pulsos4++;
}

void setup()
{
  Serial.begin(9600);
  pinMode(Evalvula1, OUTPUT);
  pinMode(sensorFlujo1, INPUT);
  pinMode(Evalvula2, OUTPUT);
  pinMode(sensorFlujo2, INPUT);
  pinMode(Evalvula3, OUTPUT); // declaramos como pin de salida
electroválvula

```

```

    pinMode(sensorFlujo3, INPUT);
    pinMode(Evalvula4, OUTPUT); // declaramos como pin de salida
electroválvula
    pinMode(sensorFlujo4, INPUT);
    pinMode(Bomba, OUTPUT); // declaramos como pin de salida bomba
    pinMode(Triquer, OUTPUT); //pin como salida
    pinMode(Echo, INPUT); //pin como entrada
    // digitalWrite(Triquer, LOW); //Inicializamos el pin con 0

    attachInterrupt(digitalPinToInterrupt(sensorFlujo1), flujo1, RISING);
    attachInterrupt(digitalPinToInterrupt(sensorFlujo2), flujo2, RISING);
    attachInterrupt(digitalPinToInterrupt(sensorFlujo3), flujo3,
RISING); // datos utiles para el sensor de flujo
    attachInterrupt(digitalPinToInterrupt(sensorFlujo4), flujo4, RISING);

    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }
    Serial.println("Iniciando tarjeta.....");
    delay (1000);
    if (!SD.begin(SSpin)) {
        Serial.println("fallo en inicialización!");
        delay(1000);
        while (1);
    }
    Serial.println("inicialización correcta-Áreas 1, 2, 3 y 4");
    delay(1000);
}

void loop ()
{
    tiempo = millis();
    //////////*****SENSOR ULTRASONICO*****////////

    if ( tiempo - tiempoultrasonido >= intervalo)
    {
        tiempoultrasonido = tiempo;

        long tiem; //tiempo que demora en llegar el eco
        long dista; //distancia en centímetros
        digitalWrite(Triquer, HIGH);
        digitalWrite(Triquer, LOW);
        tiem = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
        dista = tiem / 59; //escalamos el tiempo a una distancia en cm
        Serial.print(" cent: ");
        Serial.print(dista);

        if (dista >= 100) {
            digitalWrite(Bomba, LOW); // bomba se apaga, por q
cuando no está energizado la bomba no funciona
            return ;
        }
        if (dista < 100) {
            if (Porcentaje1 || Porcentaje2 || Porcentaje3 || Porcentaje4 <= 50
) // y si por el contrario ocurre este hecho entonces se procede así
            {
                digitalWrite(Bomba, HIGH); // bomba se enciende, porque está
conectado en normalmente abierto
            }
            if (Porcentaje1 && Porcentaje2 && Porcentaje3 && Porcentaje4 > 50
) // y si por el contrario ocurre este hecho entonces se procede así

```

```

    {
        digitalWrite(Bomba, LOW); // bomba se enciende, porque está
conectado en normalmente abierto
    }
}
}
////////*****AREA DE TRABAJO 1 *****////////
if (tiempo - Areal >= (intervall))
{
    Areal = tiempo;

    Archivo1 = SD.open("ZONA_1.txt", FILE_WRITE);
    float factor_K1 = 15.7;
    pulsosAcumulados1 += conteo_pulsos1 ;
    litrosPormin1 = ((conteo_pulsos1 ) / factor_K1 ); // (Pulse frequency
x 60 min) / 7.5Q = flow rate in L/hour
    conteo_pulsos1 = 0; // Reset Counter
    volumen1 = pulsosAcumulados1 * 1.0 / 941;

    int Valor_sensor1 = analogRead(SHumedad_C1); //constante para el valor
registrado del sensor
    const int V_Aire1 = 554; // Valor calculado con el programa de
calibración con el sensor al aire
    const int V_Agua1 = 323; // Valor calculado con el programa de
calibración con el sensor sumergido en agua
    Porcentaje1 = map(Valor_sensor1, V_Agua1, V_Aire1, 100, 0); // Se
calcula el porcentaje de humedad relativa teniendo en cuenta los dos
límites
    if (Porcentaje1 < 0 ) Porcentaje1 = 0; // Evita porcentajes negativos
en la medida del sensor
    if (Porcentaje1 > 100) Porcentaje1 = 100; // Evita porcentajes
superiores al 100% en la medida del sensor

    if (Porcentaje1 <= 50 ) // y si por el contrario ocurre este hecho
entonces se procede asi
    {

        if (tiempo - Evalvulatiempol >= (interval_1))
        {
            Evalvulatiempol = tiempo;
            Serial.println(" Areal, encendida");
            digitalWrite(Evalvulal, LOW); // Electroválvula se enciende, por
q esta conectado en normalmente cerrado
        }

        Archivo1.print(" Caudal1: ");
        Archivo1.print(litrosPormin1, 2);
        Archivo1.print(" (L/min)\tConsumo:");
        Archivo1.print(volumen1, 3);
        Archivo1.print(" (L)");
        Archivo1.print(" H1(%): ");
        Archivo1.println(Porcentaje1);
        Serial.print(" H1(%): ");
        Serial.print(Porcentaje1);
        Serial.print(" ");
        Serial.print ("Caudal1: ");
        Serial.print (litrosPormin1, 2);
        Serial.print ("L/min Volumen1: ");
        Serial.print (volumen1, 3);
        Serial.println (" L");
    }
}

```

```

    if (Porcentaje1 > 50 )           // condicional , si ocurre este hecho
    entonces se procede asi
    {
        if (tiempo - Evalvulatiempo1 >= (interval_1))
        {

            Evalvulatiempo1 = tiempo;
            Serial.println(" Areal, apagada"); // palabras que se mostraran en
            el monitor serie
            digitalWrite(Evalvulal, HIGH);    // Electroválvula se apaga

        }
        Archivo1.print(" H1(%): ");
        Archivo1.println(Porcentaje1);

        Serial.print(" H1(%): ");
        Serial.print(Porcentaje1);
        Serial.println(" ");

    }
}
Archivo1.close();
////////*****AREA DE TRABAJO 2 *****////////
if (tiempo - Area2 >= (interval2))
{
    Area2 = tiempo;
    Archivo2 = SD.open("ZONA_2.txt", FILE_WRITE);
    float factor_K2 = 182.5;
    pulsosAcumulados2 += conteo_pulsos2 ;
    litrosPormin2 = ((conteo_pulsos2 ) / factor_K2 );
    conteo_pulsos2 = 0; // Reset Counter
    volumen2 = pulsosAcumulados2 * 1.0 / 10944;

    int Valor_sensor2 = analogRead(SHumedad_C2);
    const int V_Aire2 = 540;
    const int V_Agua2 = 303;
    Porcentaje2 = map(Valor_sensor2, V_Agua2, V_Aire2, 100, 0);
    if (Porcentaje2 < 0 ) Porcentaje2 = 0;
    if (Porcentaje2 > 100) Porcentaje2 = 100;

    if (Porcentaje2 <= 50 )
    {
        if (tiempo - Evalvulatiempo2 >= (interval_2))
        {
            Evalvulatiempo2 = tiempo;

            Serial.println(" Area2, encendida");
            digitalWrite(Evalvula2, LOW);
        }

        Archivo2.print(" Caudal2: ");
        Archivo2.print(litrosPormin2, 2);
        Archivo2.print(" (L/min)\tConsumo:");
        Archivo2.print(volumen2, 3);
        Archivo2.print(" (L)");
        Archivo2.print(" H2(%): ");
        Archivo2.println(Porcentaje2);
        Serial.print(" H2(%): ");
        Serial.print(Porcentaje2);
        Serial.print(" ");
        Serial.print ("Caudal2: ");
    }
}

```

```

    Serial.print (litrosPormin2, 2);
    Serial.print ("L/min Volumen2: ");
    Serial.print (volumen2, 3);
    Serial.println (" L");
}

if (Porcentaje2 > 50)
{
    if (tiempo - Evalvulatiempo2 >= (interval_2))
    {
        Evalvulatiempo2 = tiempo;
        Serial.println(" Area2, apagada");
        digitalWrite(Evalvula2, HIGH);
    }

    Archivo2.print(" H2(%): ");
    Archivo2.println(Porcentaje2);

    Serial.print(" H2(%): ");
    Serial.print(Porcentaje2);
    Serial.println(" ");
}
}
Archivo2.close();

////////*****AREA DE TRABAJO 3 *****//////////
if (tiempo - Area3 >= (interval3))
{
    Area3 = tiempo;

    Archivo3 = SD.open("ZONA_3.txt", FILE_WRITE);
    float factor_K3 = 117.7;
    pulsosAcumulados3 += conteo_pulsos3 ;
    litrosPormin3 = ((conteo_pulsos3 ) / factor_K3 );
    conteo_pulsos3 = 0; // Reset Counter
    volumen3 = pulsosAcumulados3 * 1.0 / 7057;

    int Valor_sensor3 = analogRead(SHumedad_C3);
    const int V_Aire3 = 547;
    const int V_Agua3 = 335;
    Porcentaje3 = map(Valor_sensor3, V_Agua3, V_Aire3, 100, 0);
    if (Porcentaje3 < 0 ) Porcentaje3 = 0;
    if (Porcentaje3 > 100) Porcentaje3 = 100;

    if (Porcentaje3 <= 50 )
    {
        if (tiempo - Evalvulatiempo3 >= (interval_3))
        {
            Evalvulatiempo3 = tiempo;

            Serial.println(" Area3, encendida");
            digitalWrite(Evalvula3, LOW);
        }

        Archivo1.print(" Caudal3: ");
        Archivo1.print(litrosPormin3, 2);
        Archivo1.print(" (L/min)\tConsumo:");
        Archivo1.print(volumen3, 3);
        Archivo1.print(" (L)");
    }
}

```

```

    Archivo1.print(" H3(%): ");
    Archivo1.println(Porcentaje3);
    Serial.print(" H3(%): ");
    Serial.print(Porcentaje3);
    Serial.print(" ");
    Serial.print("Caudal3: ");
    Serial.print(litrosPormin3, 2);
    Serial.print("L/min Volumen3: ");
    Serial.print(volumen3, 3);
    Serial.println(" L");
}

if (Porcentaje3 > 50 )
{
    if (tiempo - Evalvulatiempo3 >= (interval_3))
    {
        Evalvulatiempo3 = tiempo;
        Serial.println(" Area3, apagada");
        digitalWrite(Evalvula3, HIGH);
    }

    Archivo3.print(" H3(%): ");
    Archivo3.println(Porcentaje3);

    Serial.print(" H3(%): ");
    Serial.print(Porcentaje3);
    Serial.println(" ");
}
}
Archivo3.close();

////////*****AREA DE TRABAJO 4 *****////////
if (tiempo - Area4 >= (interval4))
{
    Area4 = tiempo;

    Archivo4 = SD.open("ZONA_4.txt", FILE_WRITE);
    float factor_K4 = 75.5;
    pulsosAcumulados4 += conteo_pulsos4 ;
    litrosPormin4 = ((conteo_pulsos4 ) / factor_K4 );
    conteo_pulsos4 = 0; // Reset Counter
    volumen4 = pulsosAcumulados4 * 1.0 / 4524;
    int Valor_sensor4 = analogRead(SHumedad_C4);
    const int V_Aire4 = 531;
    const int V_Agua4 = 329;
    Porcentaje4 = map(Valor_sensor4, V_Agua4, V_Aire4, 100, 0);
    if (Porcentaje4 < 0 ) Porcentaje4 = 0;
    if (Porcentaje4 > 100) Porcentaje4 = 100;

    if (Porcentaje4 <= 50 )
    {
        if (tiempo - Evalvulatiempo4 >= (interval_4))
        {
            Evalvulatiempo4 = tiempo;
            Serial.println(" Area4, encendida");
            digitalWrite(Evalvula4, LOW);
        }
        Archivo4.print(" Caudal4: ");
    }
}

```

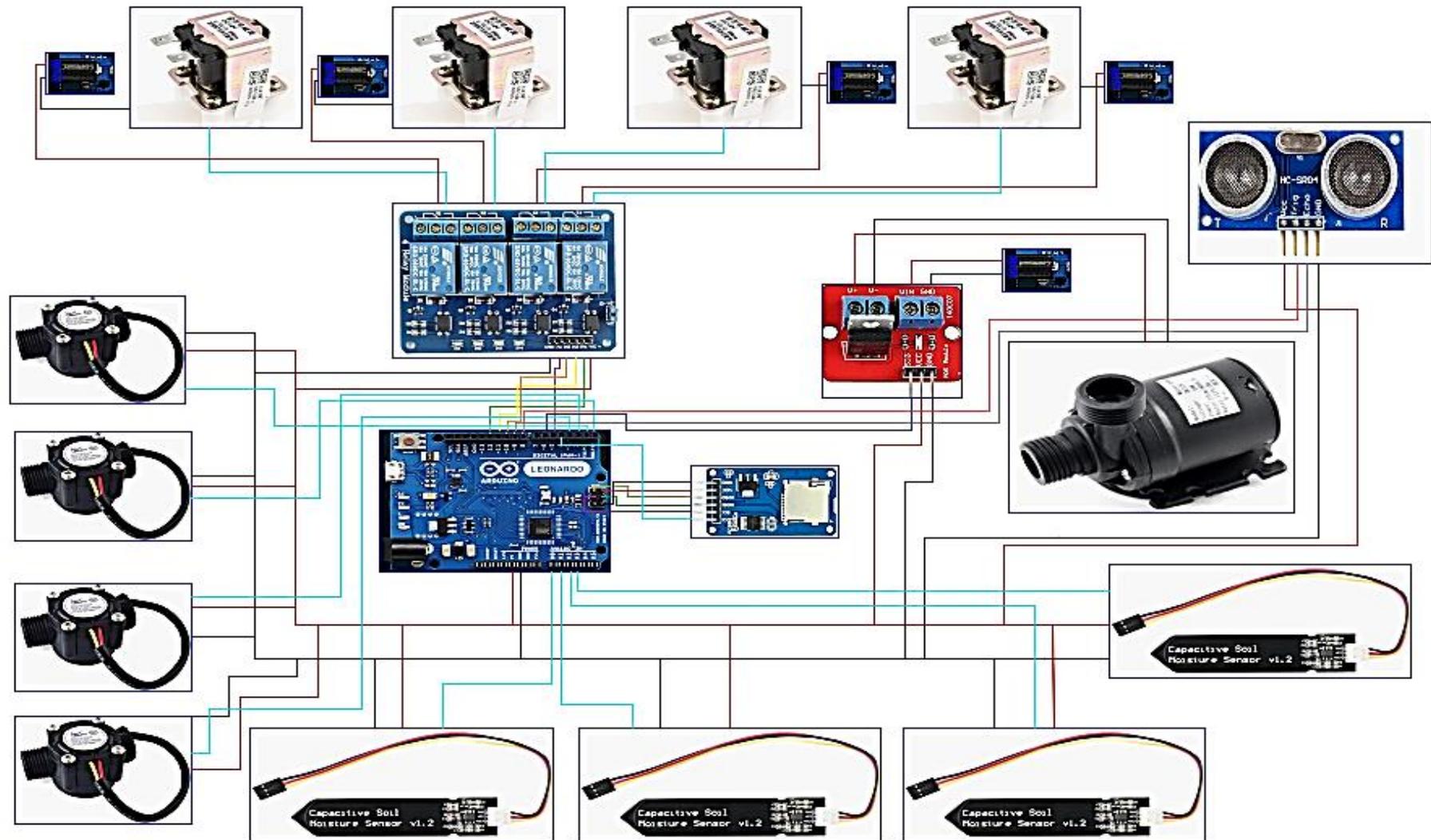
```

Archivo4.print(litrosPormin4, 2);
Archivo4.print(" (L/min)\tConsumo:");
Archivo4.print(volumen4, 3);
Archivo4.print(" (L)");
Archivo4.print(" H4(%): ");
Archivo4.println(Porcentaje4);
Serial.print(" H4(%): ");
Serial.print(Porcentaje4);
Serial.print(" ");
Serial.print("Caudal4: ");
Serial.print(litrosPormin4, 2);
Serial.print("L/min Volumen4: ");
Serial.print(volumen4, 3);
Serial.println(" L");
}
if (Porcentaje4 > 50 )
{
  if (tiempo - Evalvulatiempo4 >= (interval_4))
  {
    Evalvulatiempo4 = tiempo;
    Serial.println(" Area4, apagada");
    digitalWrite(Evalvula4, HIGH);
  }
  Archivo4.print(" H4(%): ");
  Archivo4.println(Porcentaje4);

  Serial.print(" H4(%): ");
  Serial.print(Porcentaje4);
  Serial.println(" ");
}
}
Archivo4.close();}

```

Anexo 30: Conexión general de los actuadores y sensores con el microcontrolador Arduino Leonardo



**Anexo 31: Resumen técnico de las placas Arduino más usadas**

Modelo /Tipo	N° Pines		Pines Comunicación				Micro controlador	Memoria			Velocidad			Voltaje		Tipo de conexión
	Digitales (Entrada /Salida)	Analógicos (Entrada)	I2C	SPI	pines de interrupción	n° salidas ( ) PWM		Flash	SRAM	EEPROM	reloj	trabajo	entrada	máximo		
Nano	22-22	8	A4(SDA), A5(SCK)	ICSP-(SS)10, 11(SI), 12(SO), 13(SCK)	2,3	6	ATmega328	32 KB (-2KB)	2.0KB	1KB	16MHz	5v	7-12v	6-20V	Mini-B USB	
UNO R3	14-14	6	A4(SDA), A5(SCK)	ICSP-(SS)10, 11(SI), 12(SO), 13(SCK)	2,3	6	ATmega328	32KB (-0.5KB)	2.0KB	1KB	16MHz	5v	7-12v	6-20V	USB-B	
Leonardo	20 - 20	12	2(SDA), 3(SCL)	ICSP	3,2,0,1,7	7	ATmega32U4	32 KB (-4KB)	2.5KB	1KB	16MHz	5v	7-12v	6-20V	MICROB-USB	
Mega 2560	54-54	16	20(SDA), 21(SCK)	ICSP-(SS)53, (SI)51, (SO)50, (SCK)52	2,3,21,20, 19,18	15	ATmega2560	256KB (-8KB)	8.0KB	4KB	16MHz	5v	7-12v	6-20V	USB-B	

### Anexo 32: Registro de datos almacenados en la tarjeta del módulo micro SD

The image shows four Notepad windows displaying data logs from a micro SD card. The logs are organized into four zones (ZONA\_1.TXT to ZONA\_4.TXT). Each entry typically includes flow rate (Caudal) in L/min, consumption (Consumo) in L, and humidity percentages (H1, H2, H3, H4) in %.

Zone	Flow Rate (L/min)	Consumption (L)	H1 (%)	H2 (%)	H3 (%)	H4 (%)
ZONA_1.TXT	0.25	1.202	50			
ZONA_1.TXT	1.34	1.224	20			
ZONA_1.TXT	4.08	1.292	7			
ZONA_1.TXT	4.39	1.366	5			
ZONA_1.TXT	3.57	1.425	2			
ZONA_1.TXT	4.14	1.494	1			
ZONA_1.TXT	3.50	1.553	50			
ZONA_1.TXT	3.44	1.610	30			
ZONA_1.TXT	2.87	1.658	0			
ZONA_1.TXT	3.38	1.714	47			
ZONA_1.TXT	2.48	1.756	0			
ZONA_2.TXT	4.76	1.837	28			
ZONA_2.TXT	5.63	1.931	25			
ZONA_2.TXT	5.44	2.021	27			
ZONA_2.TXT	5.54	2.114	26			
ZONA_2.TXT	5.43	2.204	28			
ZONA_2.TXT	5.32	2.293	27			
ZONA_2.TXT	5.08	2.378	25			
ZONA_2.TXT	4.75	2.457	25			
ZONA_2.TXT	4.39	2.530	23			
ZONA_2.TXT	4.54	2.606	23			
ZONA_2.TXT	3.58	2.665	23			
ZONA_2.TXT	4.67	2.743	22			
ZONA_2.TXT	4.48	2.818	21			
ZONA_2.TXT	4.65	2.896	22			
ZONA_2.TXT	4.65	2.973	24			
ZONA_2.TXT	4.49	3.048	33			
ZONA_2.TXT	4.58	3.124	32			
ZONA_3.TXT	1.06	0.931	1			
ZONA_3.TXT	1.14	0.950	1			
ZONA_3.TXT	1.16	0.970	0			
ZONA_3.TXT	1.21	0.990	0			
ZONA_3.TXT	1.19	1.009	8			
ZONA_3.TXT	1.10	1.028	0			
ZONA_3.TXT	0.88	1.042	7			
ZONA_3.TXT	1.21	1.062	10			
ZONA_3.TXT	1.33	1.085	5			
ZONA_3.TXT	1.04	1.102	8			
ZONA_3.TXT	1.21	1.122	0			
ZONA_3.TXT	1.05	1.140	3			
ZONA_3.TXT	0.93	1.155	3			
ZONA_3.TXT	0.94	1.171	6			
ZONA_3.TXT	0.99	1.187	0			
ZONA_3.TXT	0.99	1.204	0			
ZONA_4.TXT	0.45	2.202	29			
ZONA_4.TXT	0.52	2.211	21			
ZONA_4.TXT	0.62	2.221	11			
ZONA_4.TXT	0.61	2.231	12			
ZONA_4.TXT	0.45	2.239	21			
ZONA_4.TXT	0.68	2.250	21			
ZONA_4.TXT	0.54	2.259	0			
ZONA_4.TXT	0.52	2.268	18			
ZONA_4.TXT	0.60	2.278	0			
ZONA_4.TXT	0.50	2.286	20			
ZONA_4.TXT	0.34	2.292	0			
ZONA_4.TXT	0.45	2.300	17			

### Anexo 33: Compilación (verificación del código en celular) y cargada a placa Arduino Leonardo

```

1 //Arduino
2 //Leonardo
3 //*****LECTOR_MICROCD*****
4 #include <SPI.h>
5 #include <SD.h>
6 int SSpin = 4;
7 File Archivo1 ;
8 File Archivo2 ; // Creamos
archivos para almacenar los
datos de humedad de suelo y
consumo de agua
9 File Archivo3 ; // de cada una
de las 4 áreas del proyecto
(1,2,3,4)

```

Remove ads Get premium!

DIAGNOSTICS OUTPUT

Compilation finished

tesis\_codigo\_final.in monitor.serial.cpp.hex Binary sketch size: 21,408 bytes (of a 258,048 byte maximum)

```

1 //Arduino
2 //Leonardo
3 //*****LECTOR_MICROCD*****
4 #include <SPI.h>
5 #include <SD.h>
6 int SSpin = 4;
7 File Archivo1 ;
8 File Archivo2 ; // Creamos
archivos para almacenar los
datos de humedad de suelo y
consumo de agua
9 File Archivo3 ; // de cada una
de las 4 áreas del proyecto
(1,2,3,4)

```

Remove ads Get premium!

DIAGNOSTICS OUTPUT

Line Col Message

Sketch uploaded

**Anexo 34: Alternativas de uso del sistema de riego automatizado con los distintos modelos de Arduino**

ARDUINO	SENSORES			ACTUADORES Y MÓDULOS					CAMBIOS Y MODIFICACIONES			
	HUMEDAD DE SUELO CAPACITIVO	FLUJO DE AGUA YF-S201	ULTRA SONIDO HC-SR04	VÁLVULA SOLENOIDE 1/2"-(NC)	MINI BOMBA DE AGUA	RELÉ "X" CANALES	MOSFET IRF520	LECTOR MICRO SD	TOTAL PINES A/D	OPCIÓN 1	OPCIÓN 2	OPCIÓN 3
NANO	2 UNI; PINES: A0 Y A1	2 UNI; PINES: 2 Y 3	1 UNI; PINES: 0-21	2 UNI; PINES : 0-21	1 UNI; PINES : 0-21	2 UNI; PINES : 0-21	1 UNI; PINES : 0-21	1 UNI; PINES: (CS)4, (MOSI)11, (SCK)13, (MISO)12	2UNI / 12UNI	OJO: adicionar un shield nano de expansión para facilitar la conexión y la fuente de energía a través de un Jack DC (7v-12v)	reemplazar: lector SD, sensores de flujo. Adicionar 4 sensores de humedad suelo, 1 relé 4 canales y 4 válvulas solenoide	
UNO	2 UNI; PINES: A0 Y A1	2 UNI; PINES: 2 Y 3	1 UNI; PINES: 0-13	2 UNI; PINES : 0-13	1 UNI; PINES : 0-13	2 UNI; PINES : 0-13	1 UNI; PINES : 0-13	1 UNI; PINES: (CS)4, (MOSI)11, (SCK)13, (MISO)12	2UNI / 12UNI	reemplazar: lector SD, sensores de flujo. Adicionar 4 sensores de humedad suelo, 1 relé 4 canales y 4 válvulas solenoide	cambiar mini bomba de 800 lt/h por mini bomba de 240 l/h	cambiar sensores de humedad capacitivos por sensores resistivos FC-28 de un precio mucho menor
LEONARDO	4 UNI; PINES: A0, A1, A2, A3	4 UNI; PINES: 0, 1, 2, 3, 5, 7	1 UNI; PINES: 0-13	4 UNI; PINES : 0-13	1 UNI; PINES: 0-13	4 UNI; PINES: 0-13	1 UNI; PINES: 0-13	1 UNI; PINES: ICSP, 4	4UNI / 12UNI	reemplazar el lector SD y adicionar 1: sensor de flujo, humedad suelo, relé 1 canal y válvula solenoide	cambiar mini bomba de 800 lt/h por mini bomba de 240 l/h	
MEGA	6 UNI; PINES: A0-A5	6 UNI; PINES: 2, 3, 21, 20, 19, 18	1 UNI; PINES: 0-53	6 UNI; PINES: 0-53	1 UNI; PINES: 0-53	6 UNI; PINES: 0-53	1 UNI; PINES: 0-53	1 UNI; PINES: (CS)4, (MOSI)51, (SCK)52, (MISO)50	6UNI / 25UNI	reemplazar sensor de flujo y adicionar: 10 sensores de humedad suelo, 1 relé de 8 y 2 canales y 10 válvulas solenoide	fuentes externa para relé(5v) y para el Arduino(7v)	

Anexo 35: Área de ejecución del proyecto de tesis



Área/ terreno en limpio



Delimitación del terreno



Cercado del terreno



Enmallado del terreno

**Anexo 36: Instalación de las tuberías principales y secundarias**



Enterrado de tubería principal



Enterrado de tubería secundaria

**Anexo 37: Instalación de arcos de riego**



Arcos de riego



Enterrado de arcos de riego

**Anexo 38: Instalación de porta laterales y bigotes de riego**



Arcos de riego instalados en terreno



Bigotes de riego



Enterrado Porta laterales y Bigotes de riego



Porta laterales y Bigotes de riego

Anexo 39: Pruebas de funcionamiento y calibración



Prueba de purga



Prueba coeficiente de uniformidad



Uniformidad en la línea de goteros



Calibración en campo del sensor de flujo



Voltaje de llegada a los actuadores

**Anexo 40: Fertilizantes utilizados en la agricultura urbana**

BIOESTIMULANTES /BIOFERTILIZANTES									
Tipo	Utilidad	Efectos causados	Vol. Preparada*	Dosis de uso	Periodo(días)	Momento	Tiempo de vida	Rendimiento	validado
Bioestimulante dinamizado para enraizamiento	Favorece la germinación, crecimiento del tallo y la proliferación de raíces secundarias	Mineralizador del suelo, elimina los nematodos y sirve como repelente de salta hojas, pulgones e insectos trozadores	4.5Lt= 1kg estiércol+0.5kg ortiga+0.3kg de diente de león +0.1kg polvo de cuarzo+3lt agua	rociar en semillero 100ml antes y después de la siembra y hasta la aparición de las hojas	1 vez por semana hasta la verdadera hoja. Aplicación en la mañana o en días nublados	antes y después de la siembra y durante el crecimiento hasta la hoja	aproximadamente 3 meses debido a los componentes que posee	atomizador de 3 litros para 6 bandejas de semilleros de 30 alveolos con dos pasadas semanales, fumigador de 20 lt con 300ml de biopreparado para 200 plantas	Programa Terrazas Verdes Cemex-Agroambientalistas-ciudad de Bogotá-departamento de Cundinamarca-Colombia
Biol	Aplicado en todos los cultivos, estimula la floración y el fruto	Mejora la defensa frente ataque de plagas y enfermedades, se logra mayor floración, cuajado de frutos y producción.	150Lt= 40 kg estiércol+160 lt agua+ 5lt melaza de caña+1lt leche	en etapa de crecimiento o desarrollo para cultivos hortícolas con dosis ideales de 30% a 50 % de dilución, uso con atomizador o directamente al suelo.			hasta 4 meses, recipiente cerrado, oscuro y en lugar fresco	150 Lt diluidos al 50% alcanza para 0.5 ha	UNALM en huertos periurbanos de las comunidades de Mala (Lima)-Pucara (Huancayo)-Perú
Purín fermentado de ortiga	Vitaliza, estimula el crecimiento y previene enfermedades	Estimulador de crecimiento con alto contenido de N, P, K, Ca, Mg, Fe, Mo. Mejora la sanidad, el desarrollo de las raíces y el crecimiento de las plantas. También ayuda en la estructura del suelo y en la retención del agua	10Lt= 1 recipiente de 20 lt+1 kg de planta fresca o 200 gr de ortiga+ 10lt de agua (de lluvia o reposada)	preventivo: riego sobre la tierra (dosis común)	preventivo: cada 20 días, aporte extra de riego en etapas de brotación y floración, en sequías, heladas o cuando es atacada por plaga o enfermedad		6 meses en envases oscuros, cerrados y no metálicos	0.5 Lt rinde para 1 m2 de área cultivada	Programa de Agricultura Urbana de la municipalidad de Rosario CEPAR y el Programa Pro Huerta/ INTA en la ciudad de Rosario, Santa Fe, Argentina

*Nota: \*componentes predominantes \* cada biopreparado tiene sus componentes específicos*

FUENTE(Terrile & IPES-RUAF, 2010)

#### Anexo 41: Alternativas de solución frente a suelos salinos

PAUTAS	ALTERNATIVAS	LO QUE SE LOGRA
CÓMO MANEJAR LOS SUELOS SALINOS	Establecer cultivos tolerantes	*seleccionar cultivos glicófitos (adaptan) y halófilos
	Evitar usos excesivos de abonos y sobre todo los fertilizantes químicos	*debido a que algunos fertilizantes contienen niveles elevados de sales que aumentarán la salinidad del suelo
	Fracción extra de agua para el lavado	*para evitar la acumulación excesiva de sales en la zona radicular y mantener a raya los niveles de salinidad.
	Aumentar la frecuencia de riego que dependerá de la calidad de suelo y agua	*con el riego continuo se busca que el agua filtrada desplace las sales.
	Cultivar en suelos con un adecuado sistema de drenaje	promueven que las sales precipiten y generalmente se busca que los suelos sean arenosos y removidos.
	Aplicación de enmiendas con yeso o de naturaleza ácida	**mejora la estructura (floculación de la arcilla), el drenaje y aumenta las reservas útiles de agua en el suelo, previene la formación y disminución de costras en la superficie
	Construcción de camas elevadas (bancales)	ayuda a incursionar a la raíz al interior de la superficie, se genera un suelo más suelto ayudando a drenar el agua y precipita las sales
	Adicionar cubiertas vegetales	*previene la pérdida de agua por evaporación y así prevenir la acumulación de sales en las capas del suelo, también evita el ascenso de las mismas

FUENTE:(López, n.d.) \*, (Courel, 2019)\*\*

#### Anexo 42: Fuente de materia orgánica

MATERIAL	NITRÓGENO (% N)	FÓSFORO (% P <sub>2</sub> O <sub>5</sub> )	POTASIO (% K <sub>2</sub> O)	MATERIA SECA (%)	SALINIDAD (CE dS/m)
Ajinofer (líquido)	6.5	0.2	0.6		
Algas marinas	1	1	11		
Aserrín	0.2	0.1	0.2		
Biol (líquido)	1.6	0.2	1.5		
Cascarilla de arroz	0.5	0.04	4.5		
Ceniza de madera	0	1	5		
Compost	1.5	1.2	3	50	4
Gallinaza	6	5	3	30 - 40	9.2
Guano de caballo	1.6	0.4	2.5	20 - 30	2.1
Guano de cabra	1.4	1	3	40 - 50	11
Guano de cerdo	4	6.9	0.5	20 - 30	5.4
Guano de codorniz	1.5	0.2	1.2		20
Guano de conejo	0.5	1.2	0.5		
Guano de cuy	1.7	1.5	4		
Guano de islas	1.5-12	11-15	1.5-2	80	
Guano de llama	3.9	1.3	1.3	25 - 35	
Guano de alpaca	3.6	1.1	1.3	25 - 35	
Guano de oveja	4.2	2.5	6	25 - 35	
Guano de vaca (seco)	1.9	3.4	3.3	80	19
Guano de vaca (fresco)	2	2.9	1.4	20 - 40	36
Harina de pescado	10	6	—		
'Humus' de lombriz	2	1	0.6	60	3
Musgo	1	0.2	0.5		
Paja de frijol	1	0.2	4.5		
Purín (líquido)	0.25	0.1	0.35	5	
Sangre seca (sólido)	13	2	1		

FUENTE: (Ugás et al., 2000)