



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

DISEÑO DE SISTEMA DE RIEGO POR GOTEO AUTOMATIZADO UTILIZANDO UNA PLACA PROGRAMABLE RASPBERRY PI PICO Y MÓDULO ESP8266, EN LA FINCA MORALFA, DEPARTAMENTO DE QUETZALTENANGO, MUNICIPIO DE COATEPEQUE

Josué David Oliva España

Asesorado por el Ing. Kenneth Issur Estrada Ruiz

Guatemala, noviembre de 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

DISEÑO DE SISTEMA DE RIEGO POR GOTEO AUTOMATIZADO UTILIZANDO UNA PLACA PROGRAMABLE RASPBERRY PI PICO Y MÓDULO ESP8266, EN LA FINCA MORALFA, DEPARTAMENTO DE QUETZALTENANGO, MUNICIPIO DE COATEPEQUE

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

JOSUÉ DAVID OLIVA ESPAÑA

ASESORADO POR EL ING KENNETH ISSUR ESTRADA RUIZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, NOVIEMBRE DE 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. José Francisco Gómez Rivera (a. i.)
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Ing. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Armando Alonso Rivera Castillo
EXAMINADOR	Ing. Kenneth Issur Estrada Ruiz
EXAMINADOR	Ing. Natanael Jonathan Requena Gómez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE SISTEMA DE RIEGO POR GOTEO AUTOMATIZADO UTILIZANDO UNA PLACA PROGRAMABLE RASPBERRY PI PICO Y MÓDULO ESP8266, EN LA FINCA MORALFA, DEPARTAMENTO DE QUETZALTENANGO, MUNICIPIO DE COATEPEQUE

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha febrero de 2023.


José David Oliva España



Guatemala, 30 de mayo de 2023.
REF.EPS.DOC.249.05.2023.

Ing. Oscar Argueta Hernández
Director Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Argueta Hernández.

Por este medio atentamente le informo que como Asesor-Supervisor de la Práctica del Ejercicio Profesional Supervisado (E.P.S.), del estudiante universitario **Josué David Oliva España** de la Carrera de Ingeniería Electrónica, Registro Académico No. **201603019** y **CUI 2905 33104 0101**, procedí a revisar el informe final, cuyo título es **“DISEÑO DE SISTEMA DE RIEGO POR GOTEO AUTOMATIZADO UTILIZANDO UNA PLACA PROGRAMABLE RASPBERRY PI PICO Y MODULO ESP8266, EN LA FINCA MORALFA, DEPARTAMENTO DE QUETZALTENANGO, MUNICIPIO DE COATEPEQUE”**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todos”

Ing. Kenneth Issur Estrada Ruiz
ASESOR - SUPERVISOR DE EPS
Unidad de Prácticas de Ingeniería y EPS
Asesor-Supervisor de EPS
Área de Ingeniería Electrónica

c.c. Archivo
KIER/ra

Universidad de San Carlos de
Guatemala



Facultad de Ingeniería
Unidad de EPS

Guatemala 30 de mayo de 2023.
REF.EPS.D.183.05.2023.

Ing. Armando Alonso Rivera Carrillo
Director Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Presente

Estimado Ingeniero Rivera Carrillo.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **"DISEÑO DE SISTEMA DE RIEGO POR GOTEO AUTOMATIZADO UTILIZANDO UNA PLACA PROGRAMABLE RASPBERRY PI PICO Y MODULO ESP8266, EN LA FINCA MORALFA, DEPARTAMENTO DE QUETZALTENANGO, MUNICIPIO DE COATEPEQUE"** que fue desarrollado por el estudiante universitario, **Josué David Oliva España**, quien fue debidamente asesorado y supervisado por el Ing. Kenneth Issur Estrada Ruiz.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor - Supervisor de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"

Ing. Oscar Argueta Hernández
Director Unidad de EPS

/ra

REF. EIME 40.2023.

El director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del director de EPS, del asesor, con el Visto Bueno del Coordinador de Área, al Informe final de EPS del estudiante Josué David Oliva España: "**DISEÑO DE SISTEMA DE RIEGO POR GOTEO AUTOMATIZADO UTILIZANDO UNA PLACA PROGRAMABLE RASPBERRY PI PICO Y MODULO ESP8266, EN LA FINCA MORALFA, DEPARTAMENTO DE QUETZALTENANGO, MUNICIPIO DE COATEPEQUE**", procede a la autorización respectiva.



Ing. Armando Alonso Rivera Carrillo

Guatemala, 24 de agosto de 2023.



Decanato
Facultad de Ingeniería
24189101- 24189102
secretariadecanato@ingenieria.usac.edu.gt

LNG.DECANATO.OI.745.2023

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DE SISTEMA DE RIEGO POR GOTEO AUTOMATIZADO UTILIZANDO UNA PLACA PROGRAMABLE RASPBERRY PI PICO Y MÓDULO ESP8266, EN LA FINCA MORALFA, DEPARTAMENTO DE QUETZALTENANGO, MUNICIPIO DE COATEPEQUE**, presentado por: **Josué David Oliva España**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:



Ing. José Francisco Gómez Rivera

Decano a.i.

Guatemala, noviembre de 2023

JFGR/gaoc

ACTO QUE DEDICO A:

- Dios** Quien me ha acompañado en los momentos más difíciles y siempre ha estado ahí para mí.
- Mis padres** Mario Alfredo Oliva y Sandra Maritza España, por su apoyo incondicional en todo momento.
- Mis hermanos** Jorge Mario y Óscar Daniel Oliva, por ser unos muy buenos hermanos en los momentos de convivencia.
- Mis compañeros** Jorge Raúl Zelada, Jorge Andrés Rivera y Elman Obed Pelén, por su apoyo en las actividades dentro y fuera del aula.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala Por ser mi casa de estudios.

Facultad de Ingeniería Por formarme como ingeniero electrónico.

Ing. Kenneth Issur Estrada Ruiz Por la ayuda en este trabajo de graduación.

Finca Moralfa Por permitirme utilizar sus instalaciones para mi trabajo de graduación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS.....	XI
GLOSARIO.....	XIII
RESUMEN.....	XIX
OBJETIVOS.....	XXI
INTRODUCCIÓN.....	XXIII
1. ANTECEDENTES DE LA INSTITUCIÓN.....	1
1.1. Historia.....	1
1.2. Visión.....	1
1.3. Misión.....	1
1.4. Organigrama.....	2
2. MARCO TEÓRICO.....	5
2.1. Aguacate.....	5
2.1.1. Definición.....	6
2.1.2. Tipos de aguacate.....	8
2.1.2.1. Floración tipo A.....	9
2.1.2.2. Floración tipo B.....	9
2.1.3. Proceso de cultivo del aguacate.....	10
2.2. Riego.....	11
2.2.1. Definición.....	12
2.2.2. Características.....	13
2.2.3. Tipos de riego y sus usos.....	13
2.2.4. Componentes.....	15

2.2.4.1.	Tuberías	15
2.2.4.1.1.	Estándar de tuberías ANSI/ASME B31.9.....	16
2.2.4.2.	Equipos de control	17
2.2.5.	Monitoreo	17
2.2.5.1.	Control de flujo	19
2.2.5.2.	Regulador de voltaje	19
2.3.	Placas de desarrollo.....	19
2.3.1.	Elementos básicos que conforman una placa de desarrollo	20
2.3.1.1.	Módulo de alimentación	20
2.3.1.2.	Microprocesador	20
2.3.1.3.	Memoria RAM.....	21
2.3.1.4.	Memoria ROM	21
2.3.1.5.	Puertos de entrada y salida (GPIO)	21
2.3.1.6.	Buses de datos	22
2.3.1.7.	Módulo de <i>display</i>	22
2.3.2.	Tipos de placas programables.....	23
2.3.2.1.	Módulo ESP.....	23
2.3.2.1.1.	ESP-01	24
2.3.2.1.2.	ESP-05	24
2.3.2.1.3.	ESP-12	24
2.3.2.1.4.	ESP-201	25
2.3.2.1.5.	Esp-8266	25
2.3.2.1.6.	Esp-8266 WeMos D1 mini	25
2.3.2.1.7.	NodeMCU	26
2.3.2.2.	Raspberry Pi.....	27

2.3.2.3.	Módulos programables de la Raspberry Pi.....	27
2.3.2.3.1.	Pico <i>display</i>	27
2.3.2.4.	Tipos de Raspberry Pi.....	27
2.3.2.4.1.	Raspberry Pi 1 A	28
2.3.2.4.2.	Raspberry Pi 1 B	28
2.3.2.4.3.	Raspberry Pi 2 B	28
2.3.2.4.4.	Raspberry Pi 3 B	28
2.3.2.4.5.	Raspberry Pi 3 B+	29
2.3.2.4.6.	Raspberry Pi 3 A+	29
2.3.2.4.7.	Raspberry Pi 400	29
2.3.2.4.8.	Raspberry Pi Pico.....	30
2.3.2.4.9.	Raspberry Pi Zero	30
2.4.	Lenguajes de programación	31
2.4.1.	Definición	31
2.4.2.	Usos de un lenguaje de programación	32
2.4.3.	Entornos de desarrollo	32
2.4.3.1.	Thonny	33
2.4.4.	Lenguajes de programación más comunes en microcontroladores	34
2.4.4.1.	Lenguaje C.....	34
2.4.4.2.	Ensamblador	35
2.4.4.3.	MicroPython	35
2.4.4.3.1.	Convención PEP8	36
2.5.	Sensores	37
2.5.1.	Definición	38
2.5.2.	Tipos de sensores.....	39
2.5.2.1.	Sensores ópticos.....	39
2.5.2.2.	Sensores de distancia	39

	2.5.2.3.	Sensores de humedad.....	40
	2.5.2.4.	Sensores de presión.....	40
	2.5.2.5.	Sensores sónicos	40
	2.5.2.6.	Sensores de temperatura	41
	2.5.2.7.	Sensores magnéticos	41
2.6.		Actuadores.....	42
	2.6.1.	Tipos de actuadores	42
	2.6.1.1.	Válvulas	42
		2.6.1.1.1. Válvulas mecánicas	43
		2.6.1.1.2. Válvulas hidráulicas	44
		2.6.1.1.3. Válvulas neumáticas	44
		2.6.1.1.4. Válvulas electrónicas	44
	2.6.1.2.	Relevador	45
		2.6.1.2.1. Módulo de relé	46
2.7.		Comunicación inalámbrica	46
	2.7.1.	Características	47
	2.7.2.	Ventajas y desventajas.....	48
	2.7.3.	Ruido.....	49
	2.7.4.	Estándares más utilizados.....	50
		2.7.4.1. Estándar IEEE 802.11	51
	2.7.5.	Ondas electromagnéticas	52
	2.7.6.	Internet de las cosas	52
2.8.		Software para diseño	53
	2.8.1.	Tinkercad.....	53
	2.8.2.	Autodesk Eagle	53
	2.8.3.	Lucidchart.....	54
	2.8.4.	Microsoft Office	54

3.	MARCO METODOLÓGICO	57
3.1.	Dispositivos y materiales por utilizar	57
3.1.1.	Placa Raspberry Pi Pico	57
3.1.2.	Pantalla Pico <i>display</i> OLED	59
3.1.3.	Módulo ESP-8266 WeMos D1 mini	63
3.1.4.	Válvula electrónica	66
3.1.5.	Sensores.....	67
3.1.5.1.	Botón pulsador	67
3.1.5.2.	Sensor de humedad de suelo YL-69	67
3.1.5.3.	Sensor de flujo YF-B6	69
3.2.	Dispositivos hidráulicos por utilizar	70
3.3.	Lenguaje de programación por utilizar	71
3.3.1.	Micro-Python.....	71
3.4.	Diseño	72
3.5.	Programa de simulación	74
3.5.1.	Tinkercad	74
4.	DISEÑO PROPUESTO	75
4.1.	Diseño propuesto.....	75
4.1.1.	Diseño de estructura	75
4.1.2.	Diseño de un sistema electrónico	76
4.1.3.	Diagrama de relación	79
4.2.	Código de programación.....	79
5.	MANUAL DE USUARIO	91
5.1.	Descripción.....	91
5.2.	Funcionamiento	91

6.	MANUAL DE MANTENIMIENTO	93
6.1.	Mantenimiento preventivo	93
6.2.	Mantenimiento correctivo	94
7.	ANÁLISIS FINANCIERO.....	97
7.1.	Análisis de costos	97
	CONCLUSIONES	99
	RECOMENDACIONES	101
	REFERENCIAS.....	103
	APÉNDICES	107
	ANEXOS	111

ÍNDICE DE ILUSTRACIONES

FIGURAS

Figura 1.	Organigrama de la finca.....	2
Figura 2.	Finca en perspectiva desde Google Earth	3
Figura 3.	Aguacate.....	6
Figura 4.	Aguacate en alimento casero.....	7
Figura 5.	Planta del aguacate recién trasplantada	11
Figura 6.	Riego convencional.....	12
Figura 7.	Riego por aspersión.....	14
Figura 8.	Ejemplo de tubería.....	16
Figura 9.	Ejemplo de <i>display</i>	23
Figura 10.	Esp-8266 WeMos D1 mini	26
Figura 11.	Raspberry Pi Pico	30
Figura 12.	IDE Thonny.....	33
Figura 13.	Código de programación en lenguaje C	34
Figura 14.	Código de programación en lenguaje ensamblador	35
Figura 15.	Código de programación en lenguaje MicroPython	36
Figura 16.	Sensor ultrasónico HC-SR04	38
Figura 17.	Válvula de agua	43
Figura 18.	Relevador o relé	46
Figura 19.	Comunicación inalámbrica entre dos teléfonos celulares	47
Figura 20.	Ruido en señal de cable en un televisor.....	50
Figura 21.	Imagen de los pines de la Raspberry Pi Pico	59
Figura 22.	Pantalla OLED estableciendo conexión	61
Figura 23.	Pantalla OLED iniciada	62

Figura 24.	Pantalla OLED en modos de operación.....	62
Figura 25.	Pantalla OLED del sistema en un modo de operación.....	63
Figura 26.	Diagrama de pines del ESP8266 WeMos mini D1	64
Figura 27.	ESP funcionando como servidor	65
Figura 28.	ESP funcionando como cliente.....	66
Figura 29.	Sensor de humedadYL-69.....	68
Figura 30.	Sensor de flujo YF-B6	70
Figura 31.	Texto de pantalla OLED funcionando en IDE Thonny	72
Figura 32.	Caja de protección IP66	73
Figura 33.	Diseño de la estructura.....	75
Figura 34.	Diseño de la estructura vista superior.....	76
Figura 35.	Conexiones de la Raspberry Pi Pico y ESP8266 del área A.....	77
Figura 36.	Conexiones de la ESP8266 en el área B.....	78
Figura 37.	Diagrama de relación del sistema.....	79
Figura 38.	Variabes de la Raspberry Pi Pico	80
Figura 39.	Pines de entrada y salida	80
Figura 40.	Texto en la pantalla OLED	81
Figura 41.	Lectura de datos recibidos de la ESP8266	81
Figura 42.	Comunicación con la ESP8266	82
Figura 43.	Declaración de funciones para control de flujo I	82
Figura 44.	Declaración de funciones para control de flujo II	83
Figura 45.	Declaración de funciones para control de flujo III	83
Figura 46.	Tiempos de apertura I	84
Figura 47.	Tiempos de apertura II	84
Figura 48.	Tiempos de apertura III	85
Figura 49.	Librerías y puertos.....	85
Figura 50.	Funciones de lectura y escritura.....	86
Figura 51.	Comunicación inalámbrica	87
Figura 52.	Activación de pines y librerías por utilizar	88

Figura 53.	Conexión con <i>socket</i> existente	88
Figura 54.	Condición de continuación y valor de humedad	89

TABLAS

Tabla 1.	Propuesta de documentación del mantenimiento preventivo	94
Tabla 2.	Propuesta de documentación del mantenimiento correctivo	96
Tabla 3.	Costos del material eléctrico y electrónico	97
Tabla 4.	Costos de materiales de riego y tubería.....	98
Tabla 5.	Costos totales	98

LISTA DE SÍMBOLOS

Símbolo	Significado
B	Byte
Vcc	Colector de voltaje
gal	Galón
C°	Grados Celsius
L	Litros
m	Metro
mm	Milímetro
GHz	Mil millones de hercios
Min	Minuto
Pa	Pascales
Mb	Un millón de <i>bits</i>
Mbps	Un millón de <i>bits</i> por segundo
MB	Un millón de <i>bytes</i>
MHz	Un millón de hercios
MPa	Un millón de pascales
V	Voltaje

GLOSARIO

ADC	Convertidor de señal analógica a digital.
ANSI	Instituto Nacional Estadounidense de Estándares.
Antena	Dispositivo diseñado con el objetivo de emitir o recibir ondas electromagnéticas.
ARM	Empresa fabricante de procesadores ARM.
ARM procesador	Procesador con menos consumo eléctrico y más económico que un procesador convencional.
ASME	American Society of Mechanical Engineers.
<i>Bit</i>	Unidad básica de la informática.
<i>Buzzer</i>	Dispositivo que convierte la energía eléctrica en sonido.
<i>Byte</i>	Unidad de información estándar utilizada en informática conformada de 8 <i>bits</i> .
DAC	Convertidor de señales digitales a señales analógicas.

<i>Datasheet</i>	Hoja de datos del fabricante que contiene las características del dispositivo.
Derivación	Conducto secundario que se obtiene de conectar un conducto al principal.
<i>Display</i>	Dispositivo que muestra información visual, como números, letras o gráficos.
ESP	Acrónimo de Expressif Systems Platform.
GND	Punto de referencia en un circuito eléctrico.
GPIO	Pin programable para distintos propósitos.
HDMI	Interfaz multimedia de alta definición.
Hercio	Unidad de frecuencia del Sistema internacional de Unidades.
I2C	Protocolo de comunicación serial.
IDE	Es un entorno de desarrollo que ofrece un conjunto de herramientas integradas para facilitar el proceso de creación y depuración de software.
IoT	<i>Internet of Things</i> (Internet de las cosas en español).

Hertz	Unidad de frecuencia del Sistema internacional de Unidades.
IEEE	Acrónimo de Instituto de Ingenieros Eléctricos y Electrónicos.
IoT	Interconexión de dispositivos y objetos a través de una red.
LCD	Pantalla de cristal líquido.
Led	Dispositivo semiconductor que genera luz.
Lixiviación	Disolución de un sólido mediante un líquido y extraer el líquido resultante.
MicroSD	Memoria SD de menor tamaño.
OLED	Un diodo orgánico de emisión de luz en una capa electroluminiscente, formada por una película de componentes orgánicos.
PCB	Placa de circuito impreso.
PVC	Resina usada frecuentemente para aislamientos y tuberías.
PWM	Técnica en la que se modifica el ciclo de trabajo de una señal periódica.

QSPI Flash	Tipo de SPI.
RAM	Memoria de acceso aleatorio.
Reflexión	Cambio de dirección de una onda al entrar en contacto con una superficie.
Relé	Interruptor controlado mediante un circuito eléctrico.
Resistencia	Oposición de un componente al paso de la corriente a través de él.
SCL	Línea de pulsos para sincronizar los dispositivos.
SD	Dispositivo de almacenamiento electrónico.
SDA	Línea de comunicación de datos.
SDRAM	Memoria volátil de alta velocidad sincrónica.
UART	Dispositivo que convierte datos en serie a paralelo y viceversa.
UPS	Dispositivo que almacena energía por si se interrumpe el suministro eléctrico.
USB	Dispositivo que permite la conexión de distintos periféricos.

WEP	Protocolo más antiguo de seguridad entre redes inalámbricas.
WPA	Protocolo de seguridad entre redes inalámbricas sucesoras de WEP.
WPA2	Protocolo de seguridad entre redes inalámbricas sucesoras de WPA.

RESUMEN

El presente trabajo corresponde al anteproyecto del Ejercicio Profesional Supervisado (EPS) y se titula: *Diseño de sistema de riego por goteo automatizado utilizando una placa programable Raspberry Pi Pico y módulo ESP8266, en la finca Moralfa, departamento de Quetzaltenando, municipio de Coatepeque.*

En este diseño se plantea un sistema de riego tipo goteo automatizado, el cual está destinado para plantación de aguacate. Este sistema apoyará la conservación del agua, debido a que el sistema de riego por goteo ayuda a conservar una mayor cantidad de la misma, ya que es un recurso que no está disponible todo el día en todos los lugares del país.

Asimismo, se aplicarán los conceptos teóricos para poder acoplarse a las placas de desarrollo a Raspberry Pi Pico y ESP8266 en las áreas rurales.

En el anteproyecto se detallan los fundamentos teóricos, el plan de trabajo, los recursos humanos y materiales con los que se contará, así como el tipo de placas programables y el lenguaje de programación a utilizar, además de las características de este tipo de riego y de los dispositivos por utilizar.

OBJETIVOS

General

Diseñar un sistema de riego por goteo automatizado utilizando una placa programable Raspberry Pi Pico y módulo ESP8266, en la finca Moralfa, departamento de Quetzaltenando, municipio de Coatepeque.

Específicos

1. Determinar los componentes electrónicos a utilizar en el diseño de acoplamiento del sistema de riego de acuerdo con las dimensiones de la finca.
2. Definir el tipo de tubería a utilizar y los costos para la implementación.
3. Diseñar un esquema del funcionamiento.
4. Desarrollar el código fuente en el lenguaje de programación.
5. Desarrollar los manuales de usuario y mantenimiento del diseño propuesto.

INTRODUCCIÓN

El presente trabajo corresponde al anteproyecto del Ejercicio Profesional Supervisado (EPS), el mismo se desarrollará para el diseño de un sistema de automatización de riego por goteo para mejorar el sistema existente en la finca Moralfa.

Se utilizando placas programables y se genera un código para la implementación y automatización de un sistema de riego tipo goteo. Además, se utilizarán válvulas de control de flujo del agua, controladas electrónicamente, junto con algunas manuales para la interrupción de emergencia. Se diseñará un sistema que se adapte de la mejor manera al cultivo de aguacate, junto con sensores de humedad, para aprovechar el agua de la mejor manera posible y no dañar las plantas al suministrarles demasiada agua.

Este proyecto se realizará con el fin de incrementar la eficiencia del uso del agua en la finca y así mejorar los riegos tradicionales. Además, al automatizar los cultivos, se puede mejorar la calidad de estos, controlar de mejor manera el crecimiento de las plantas y finalmente acercar la tecnología a las áreas rurales, para que así puedan verse los beneficios de la implementación tecnológica en una de las principales actividades del país: la agricultura.

1. ANTECEDENTES DE LA INSTITUCIÓN

1.1. Historia

La finca Moralfa fue fundada en el año 1792 en el municipio de Quetzaltenango y perteneció al expresidente Manuel Lisandro Barillas. Posee una extensión territorial aproximada de 3.5 caballerías, de las cuales una está dedicada en su totalidad a la siembra y cultivo de aguacates Hass y café. Posee hoy en día un promedio de 9,000 árboles de aguacate, con una distancia entre ellos de 6 metros equidistantes. También pertenece al grupo de ANACAFÉ, como la tercera finca con mejor producción de café en la categoría volcánica San Marcos, en la región I. En el ciclo 2021-2022 tuvo una de las mejores producciones de café en la región.

1.2. Visión

Brindar productos de alta calidad y sostenibles al mercado nacional e internacional, con el fin de contribuir al desarrollo económico y social de la región y del país en su conjunto. Nos comprometemos a trabajar de manera responsable y ética, promoviendo la conservación del medio ambiente y el bienestar de nuestros trabajadores y comunidades aledañas. (Finca Moralfa, 2023, párr. 2)

1.3. Misión

Tenemos la visión de convertirnos en líderes en la producción de café y aguacate Hass sostenibles, reconocidos por la calidad y la excelencia de

nuestros productos a nivel nacional e internacional. Buscamos ser una empresa innovadora y comprometida con la preservación del medio ambiente y el desarrollo económico y social de nuestras comunidades aledañas. Además, aspiramos a ser una empresa rentable y sostenible a largo plazo, que genere empleo y contribuya al bienestar de nuestra región y del país. (Finca Moralfa, 2023, párr. 3)

1.4. Organigrama

En la siguiente figura se presenta detalladamente el organigrama según el cual se trabaja en la finca Moralfa.

Figura 1.

Organigrama de la finca



Nota. Diagrama que muestra el organigrama de la finca. Elaboración propia, realizado con Microsoft Word.

Figura 2.

Finca en perspectiva desde Google Earth



Nota. Finca en perspectiva aérea desde. Obtenido de Google Earth (2022). *Fotografía panorámica (aérea) de la finca Moralfa.* (<https://earth.google.com/web/@14.8328868,-91.5134779,2325.14336969a,153.02944921d,35y,0h,45t,0r/data=CIEaTxJJiUweDg1OGU5N2ZIM2ZjODIkM2Y6MHgzYmE3ZmYwMTFmMGYwMDBmGdLFppVCtC1AlXegTnn04VbAKg5RdWV0emFsdGVuYW5nbxgCIAE>), consultado el 12 de diciembre del 2022. De dominio público.

2. MARCO TEÓRICO

2.1. Aguacate

El aguacate es una fruta única que ha ganado popularidad en los últimos años debido a su versatilidad, valor nutricional y significado cultural. Según la definición botánica del aguacate, este es conocido científicamente como *Persea Americana*; es una fruta que se originó en América Central y se ha cultivado durante miles de años.

Por su parte, el árbol de aguacate es un miembro de la familia *Lauraceae*, que también incluye canela y hojas de laurel. A diferencia de la mayoría de las frutas, el aguacate es alto en grasas saludables y bajo en azúcar. La anatomía de una fruta de aguacate consiste en una semilla grande, rodeada por una pulpa verde cremosa que es rica en nutrientes. La fruta del aguacate es única porque no madura hasta que se cosecha, lo que la convierte en un desafío para los productores comerciales.

Figura 3.

Aguacate



Nota. Aguacate comercial. Elaboración propia.

2.1.1. Definición

El aguacate se ha convertido en un ingrediente popular en muchas cocinas de todo el mundo, debido a su textura y sabor. Se usa comúnmente en la cocina mexicana, guatemalteca, chilena, entre otras. El valor nutricional del aguacate es otra razón de su popularidad. Tiene un alto contenido de grasas saludables, fibra y vitaminas. Además, el aguacate se puede preparar de muchas maneras diferentes, como asar a la parrilla, freír y hornear.

Se ha convertido en un alimento indispensable en todas las cocinas: ensaladas, guacamole, tostadas, entre otras comidas. Son infinitas las recetas que se pueden elaborar con el aguacate. Es el ingrediente ideal para una dieta saludable y equilibrada (Sigfrido Fruit, 2021).

Figura 4.

Aguacate en alimento casero



Nota. Aguacate machacado conocido como guacamole. Elaboración propia.

Almacene los aguacates a temperatura ambiente hasta que estén completamente maduros. Si desea reducir la maduración, guarde inmediatamente los aguacates en el refrigerador, ya que no continuarán madurando cuando estén refrigerados. Aguacates enteros y maduros se mantendrán durante dos o tres días en el refrigerador, mientras que los aguacates cortados se mantendrán durante uno o dos días. La carne de aguacate se oscurece cuando se expone al aire. (Viveros Grajera, 2019, s.p.)

Sin embargo, la industria del aguacate también se ha enfrentado a controversias en los últimos años, debido a preocupaciones ambientales y laborales. La alta demanda de aguacates ha provocado deforestación y escasez de agua en algunas regiones, y ha habido informes de explotación laboral en las fincas de aguacate.

El aguacate es una fruta única que ha ganado popularidad debido a su versatilidad, valor nutritivo y significado cultural. Al explorar sus aspectos botánicos, culinarios y culturales, es posible obtener una mejor comprensión de lo que hace que el aguacate sea tan especial. Si bien existen preocupaciones sobre los impactos ambientales y sociales de la industria del aguacate, está claro que este seguirá siendo una parte importante de muchas cocinas y culturas en todo el mundo.

2.1.2. Tipos de aguacate

Según el momento en que la flor abra como femenina o masculina, las variedades de aguacates se clasifican en dos grupos florales: tipo A y tipo B. En los de tipo A se incluyen: Hass, Lamb Hass, Pinkerton y Reed; y en los de tipo B: las variedades son Fuerte, Bacon y Zutano (Junta de Andalucía, 2022).

- Aguacate Hass: el tipo de aguacate más común. Es el tipo más extendido y en forma ovalada. Sus semillas son pequeñas y gruesas. En invierno y primavera, el exportador principal de aguacate Pinkerton siempre ha sido Israel, por lo que se puede decir que este es un aguacate característico en estas áreas (Sigfrido Fruit, 2021).
- Pinkerton es una especie de híbrido de Guatemala-México con buena calidad de pulpa, que madura entre la primavera y el verano. Proviene del cruce entre Rincón y Hass, que son dos variedades de prestigio altos. Su fruto está aislado, delgado y de color verde oscuro. La piel es mediana y estricta, fácil de pelar. La carne es de color crema, y la calidad es muy buena. El tamaño esquelético es pequeño y el rango de fruta es de entre 250 y 500 gramos. En comparación con la carne provista de otros

aguacates específicos, las semillas son muy pequeñas, y tienen una buena calidad (Junta de Andalucía, 2022).

- Bacon es una variedad del aguacate con tonos oscuros y de pequeño tamaño originario de California en Estados Unidos. Posee un alto contenido de aceite a comparación de otras variedades de aguacate, por lo que su sabor viene acompañado de una sensación muy suave y cremosa. Lo más destacado del aguacate de México es que su piel es comestible, al igual que las hojas del árbol que lo produce y que se utilizan como condimento (Sigfrido Fruit, 2021).

Para incrementar la frecuencia de fecundación y cuaje de los frutos de una mejor manera, es necesario llevar a cabo una polinización cruzada. Esto se da al plantar en conjunto variedades que presenten una floración complementaria y que permita la coexistencia de flores de ambos sexos en el mismo momento, flores de una variedad tipo A y tipo B (Viveros Brokaw, 2022).

2.1.2.1. Floración tipo A

Las flores de tipo A son aquellas que abren en estado femenino por la mañana del primer día y se cierran al medio día del mismo día, y no se vuelven a abrir hasta la tarde del siguiente día, y, cuando lo hacen, lo hacen en estado masculino. Algunas de las variedades del tipo A son, por ejemplo, Hass y Lamb-Hass (Viveros Brokaw, 2022).

2.1.2.2. Floración tipo B

Las flores de variedad tipo B también se abren en estado femenino por primera vez, pero durante la tarde y en estado masculino durante la mañana del

día siguiente. Algunos ejemplos de variedades tipo B son Zutano, Bacon y Fuerte (Viveros Brokaw, 2022).

2.1.3. Proceso de cultivo del aguacate

El proceso de cultivo de aguacate generalmente comienza con la selección de semillas de frutas maduras y saludables. Estas semillas se limpian y se colocan en un lugar cálido y húmedo para iniciar la germinación. Después de un tiempo, se pueden trasplantar las plántulas a macetas más grandes y se colocan en un ambiente cálido y soleado.

Cuando las plántulas alcanzan una altura de aproximadamente 30 cm, se pueden trasplantar al campo. El sitio de cultivo debe ser elegido cuidadosamente, ya que los árboles de aguacate requieren un clima cálido y húmedo y un suelo bien drenado. Durante los primeros meses, los árboles requieren un riego regular para asegurarse de que las raíces tengan suficiente humedad.

Después de unos años, los árboles de aguacate comenzarán a dar frutos. La cosecha se realiza durante la temporada de producción, que varía según la ubicación geográfica. Durante la cosecha, se debe tener cuidado de no dañar los árboles y se debe recolectar los frutos con cuidado para evitar dañarlos.

Figura 5.

Planta del aguacate recién trasplantada



Nota. Planta de aguacate recién trasplantada en un terreno en San Lucas Sacatepéquez. Elaboración propia.

2.2. Riego

En general, cuando se habla de riego, se dice que la pregunta principal considera cuándo y cómo regar. Debe conocerse si el suelo que se va a regar posee problemas, ya sea de falta de nutrientes o si es muy seco. También es necesario saber cómo aplicar dependiendo del tipo de suelo en el que se define el tiempo y cuando se define ese tiempo. Todos estos datos son para el uso

general de agua y aprovechar la oportunidad de tener la máxima eficiencia de esta y obtener el máximo rendimiento de producción (Redactor de Revista Irrinews, 2017).

Por último, se debe tener en cuenta que la cantidad y la forma de riego dependen del clima y tipo de planta.

Figura 6.

Riego convencional



Nota. Riego de un árbol en un patio con una manguera convencional. Elaboración propia.

2.2.1. Definición

La definición y el objetivo del riego en general incluyen la aplicación artificial de agua en el suelo para que las plantas (cultivos) puedan satisfacer las

necesidades de la humedad requerida para su desarrollo. Los objetivos del riego son:

- Proporcionar la humedad necesaria para los cultivos.
- Proporcionar nutrición en la solución.
- Hay que asegurar que el cultivo se oponga a la sequía a corto plazo.
- Refrigerar el suelo y la atmósfera para mejorar el medio ambiente de las plantas.
- Reducir el contenido de la sal en el suelo.

2.2.2. Características

Elegir el sistema de riego que sea más adecuado para el cultivo es una de las decisiones más importantes en la agricultura. Conocer las diferentes alternativas, antes de elegir el tipo de regadío, ayuda a conocer más las necesidades de los cultivos y de qué manera es posible potenciar y cuidar su crecimiento (Citi-Sa, 2023).

Una de las principales necesidades de la agricultura es el agua, el alimento de las plantas. El agua hace posible que estas se desarrollen y crezcan bien, siendo así el papel de los sistemas de riego trasladar el agua de un lugar a otro en terrenos muy extensos, de forma rápida, constante y completa (SAS, 2022).

2.2.3. Tipos de riego y sus usos

El riego de espolvoreado incluye un mecanismo que descarga agua, haciendo que el agua parezca llover en la tierra en algunos lugares específicos.

Figura 7.

Riego por aspersión



Nota. Riego por método de aspersión en jardín casero. Elaboración propia.

Se llama riego por goteo a un sistema de tubería dispuesto a lo largo del suelo o en la superficie a lo largo del jardín de plantación. Se le denomina riego gota a gota a la clase de riego que es utilizada en zonas en las que el agua escasea y optimiza ese recurso de una manera considerable. La idea de funcionamiento es distribuir el agua a través de goteros, que humedecerán la zona de las raíces de cada una de las plantas.

Finalmente, en el campo más grande, generalmente se usa el riego por inundación. En este sistema el agua alcanza los cultivos de una manera similar a la definición de la lluvia local. La unidad que formó el sistema es un grupo de

bombeo, con supervisores hidráulicos de incendios, algunas tuberías y lanzamiento (tubo perforado, boquilla, rociador) (Novagric, 2022).

2.2.4. Componentes

Algunos de los principales componentes de un sistema de riego son los siguientes:

2.2.4.1. Tuberías

Una tubería es un conducto hueco que se utiliza para transportar líquidos, gases u otros fluidos de un lugar a otro. Las tuberías pueden estar hechas de diversos materiales, como hierro fundido, acero, cobre, PVC, entre otros, y su diámetro y grosor varían según la cantidad y la presión del fluido que se va a transportar.

Las tuberías se utilizan en una amplia variedad de industrias y aplicaciones, desde la distribución de agua potable y la conducción de petróleo y gas, hasta la ventilación de edificios y el transporte de productos químicos. Las tuberías también poseen distintos tamaños, en general, el tamaño de la tubería debe seleccionarse cuidadosamente para garantizar un flujo de fluido adecuado y una presión de transporte óptima para la aplicación específica. Por ejemplo, las tuberías más pequeñas, con diámetros de menos de 1 pulgada (25.4 mm), se utilizan comúnmente en aplicaciones de plomería residencial y comercial, mientras que las tuberías más grandes, con diámetros de varias pulgadas o más (100 mm o más), se utilizan en aplicaciones industriales, de energía y de tratamiento de agua.

Figura 8.

Ejemplo de tubería



Nota. Imagen tomada de tubería en una construcción. Elaboración propia.

2.2.4.1.1. Estándar de tuberías ANSI/ASME B31.9

Los estándares de tuberías se refieren a las especificaciones y normas que se utilizan para diseñar, fabricar y ensamblar tuberías para diferentes aplicaciones, tales como la distribución de fluidos, el transporte de gas y petróleo, la industria química, la industria alimentaria, entre otras. El estándar ANSI/ASME B31 especifica estándares para el diseño, construcción, inspección y pruebas de tuberías y sistemas de tuberías para diferentes aplicaciones. Específicamente en el área 31.9 se puntualizan tuberías al servicio de fábricas, industrias, comercio y unidades multifamiliares.

2.2.4.2. Equipos de control

Son todos los dispositivos que ayudan a controlar todo el proceso de riego, desde contadores hasta electroválvulas. Todos estos elementos estarían dentro de un diseño estándar, pero, por ejemplo, si el sistema de riego fuera por aspersión, se cambiarían las tuberías por aspersores (Novagric, 2022).

Su función es controlar todo el sistema de riego. Se sitúa al inicio de la instalación, e incluye todos los equipos y materiales empleados para controlar el sistema de riego, medir el volumen de agua que entra procedente del embalse (contador), filtrarla, regular su presión, dosificar los fertilizantes y programar el riego (Prakor, 2019).

Los componentes de un sistema de riego por aspersión son: equipo de bombeo, línea principal, líneas secundarias, sistema de aspersores e inyectores para fertirrigación. Los fertilizantes deben estar a bajas concentraciones para evitar quemaduras en el follaje, además, después de aplicar el agua con fertilizantes, se debe hacer un riego sin fertilizantes con la finalidad de lavar residuos de estos sobre las hojas, de tal forma que lleguen a la zona de raíz de la planta. Hay que considerar que, al aplicar mayor volumen de agua, es probable que exista lixiviación de fertilizantes, aumentando los costos del insumo al ser utilizados con menor eficiencia.

2.2.5. Monitoreo

Consejos de mantenimiento y solución de problemas para tuberías y equipos de control de riego:

- Barro en el emisor: otro escenario que puede dificultar el funcionamiento adecuado del sistema de riego es la presencia de barro seco en los emisores, debido a que es un componente que puede obstruir parcial o completamente el paso del agua. Para solucionar esto bastará con realizar una limpieza de los emisores y asegurarse de abrir el riego luego de que llueva, para eliminar los restos de barro antes de que se sequen (Novagric, 2022).
- Problemas de presión: otro de los problemas comunes que presentan los sistemas de riego es la falta de presión en el agua. Esto puede suceder debido al exceso de emisores de riego para el sistema actual, un suministro de agua insuficiente, fallas en el funcionamiento del sistema, entre otros factores. Para solucionar este problema es necesario rectificar si el sistema de riego no tiene presión debido a un exceso de emisores para ese sistema en particular o si existe algún otro agente externo que pueda ocasionar el problema. Si se descubre que el inconveniente está en el mecanismo del sistema, lo mejor es acudir a especialistas (Novagric, 2022).
- En caso de comprobar que ningún artefacto ajeno al cultivo afecta la presión del agua, lo más indicado es comunicarse con el proveedor de servicios públicos, ya que el problema se encuentra a lo largo de la línea de agua y puede ser causado por nuevas construcciones o trabajos de mantenimiento en el área (Novagric, 2022).
- Presión del agua: otro problema común planteado por el sistema de riego es la carencia de presión de agua. Este es el caso que puede suceder porque hay demasiados transmisores de riego en el sistema actual, el suministro de agua y la falla del sistema. Para resolver este problema es

necesario corregir si el sistema de riego no tiene presión debido a transmisores excesivos del sistema específico o si hay otros agentes externos que pueden causar problemas (Novagric, 2022).

2.2.5.1. Control de flujo

El control de flujo de agua se refiere a la capacidad de regular el caudal de agua que fluye a través de una tubería o sistema de riego. Puede ser realizado mediante el uso de válvulas, compuertas, reguladores de presión, sensores de flujo y otros dispositivos que permiten ajustar y controlar la cantidad de agua que se suministra a una zona de riego o a un cultivo específico. El control de flujo de agua es importante para garantizar que las plantas reciban la cantidad adecuada de agua, evitando el desperdicio y maximizando la eficiencia del riego.

2.2.5.2. Regulador de voltaje

Un regulador de voltaje es un componente electrónico que mantiene una salida de voltaje constante, a pesar de las variaciones en la entrada de voltaje o la carga eléctrica. Su función principal es proteger los dispositivos electrónicos sensibles de daños por fluctuaciones en el suministro de energía eléctrica.

2.3. Placas de desarrollo

Las placas de desarrollo son dispositivos que cuentan con un microcontrolador (*microchip*) reprogramable, el cual puede ejecutar instrucciones para un fin específico. Generalmente estas placas cuentan con entradas y salidas análogas y/o digitales para permitir la comunicación con sensores externos, haciendo uso de la electrónica digital (Placas de Desarrollo, 2022).

Las placas de desarrollo son una computadora simple de placa única que realiza tareas simples tomando entradas de componentes electrónicos externos como sensor de luz y generador de salida como iluminación de luz.

2.3.1. Elementos básicos que conforman una placa de desarrollo

Existen muchos componentes que conforman una placa de desarrollo programable, pero algunos de los elementos comunes que se encuentran en estas son:

2.3.1.1. Módulo de alimentación

Un módulo de alimentación es un dispositivo que se utiliza para suministrar energía eléctrica a un sistema o dispositivo electrónico. Los módulos de alimentación pueden variar en tamaño y forma, desde unidades pequeñas que se utilizan para suministrar energía a dispositivos electrónicos portátiles, hasta grandes sistemas que se utilizan para alimentar servidores y otros equipos informáticos

2.3.1.2. Microprocesador

Un microprocesador es un circuito integrado que contiene la unidad central de procesamiento (CPU) de un ordenador. El microprocesador es el cerebro del ordenador y es responsable de ejecutar las instrucciones de los programas y realizar las operaciones lógicas y aritméticas necesarias para procesar los datos. El microprocesador se compone de millones de transistores que se organizan en una o varias unidades de procesamiento para ejecutar las instrucciones del programa.

2.3.1.3. Memoria RAM

La memoria RAM (*random access memory* o memoria de acceso aleatorio) es un tipo de memoria volátil de un ordenador que se utiliza para almacenar temporalmente los datos y programas en uso. La RAM permite al procesador acceder rápidamente a los datos que necesita para llevar a cabo las operaciones del sistema y las aplicaciones en ejecución. Cuando se apaga el ordenador, la memoria RAM pierde los datos almacenados en ella.

2.3.1.4. Memoria ROM

La memoria ROM (*read only memory* o memoria de solo lectura) es un tipo de memoria no volátil de un ordenador que se utiliza para almacenar datos que no cambian o que no deben ser modificados. La memoria ROM es esencialmente un chip en el que se escriben los datos durante el proceso de fabricación y que no puede ser alterado una vez que se ha grabado. Se utiliza para almacenar programas que son necesarios para que el computador funcione correctamente.

2.3.1.5. Puertos de entrada y salida (GPIO)

Los puertos GPIO (*general purpose input/output*) son un conjunto de pines de entrada/salida que se encuentran en algunos dispositivos electrónicos y que se pueden utilizar para conectar y controlar dispositivos externos. Los pines GPIO se pueden configurar para funcionar como entradas o salidas y se pueden utilizar para leer señales de sensores o para controlar el encendido o apagado de dispositivo. La configuración de los pines GPIO y su funcionalidad depende del dispositivo en el que se encuentran.

2.3.1.6. Buses de datos

Los buses de datos son un conjunto de cables o pistas eléctricas que se utilizan para transmitir datos entre los componentes de un sistema informático, como la CPU, la memoria, los dispositivos de entrada/salida y otros dispositivos. Los buses de datos se utilizan para transportar información digital, como instrucciones de programa, datos de entrada/salida y direcciones de memoria, entre los diferentes componentes del sistema. Los buses de datos pueden variar en tamaño y velocidad.

2.3.1.7. Módulo de *display*

Un módulo de *display* es un componente electrónico que se utiliza para mostrar información visual en una pantalla. Los módulos de *display* pueden ser de varios tipos, incluyendo LCD, OLED y led, y se utilizan en una amplia variedad de dispositivos electrónicos, desde relojes y calculadoras hasta televisores y paneles de control. El módulo de *display* convierte la información digital en una imagen visual que se puede ver en la pantalla.

Figura 9.

Ejemplo de display



Nota. Pantalla OLED de microcontrolador. Elaboración propia.

2.3.2. Tipos de placas programables

Existe una gran diversidad de placas programables y también varios fabricantes en todo el mundo. Algunos son los siguientes:

2.3.2.1. Módulo ESP

El módulo ESP (o ESP module) es una serie de módulos de desarrollo de hardware y software creados por Espressif Systems. Estos módulos utilizan un microcontrolador de bajo costo y bajo consumo energético, combinado con un módulo Wi-Fi integrado, lo que los hace muy populares en proyectos de IoT (Internet de las cosas).

El módulo ESP se ha convertido en un elemento básico en la creación de proyectos electrónicos, debido a su bajo costo, facilidad de uso y gran cantidad de documentación y recursos disponibles en línea. Algunos de los módulos más conocidos de la serie ESP son el ESP8266 y el ESP32, que han sido

ampliamente utilizados en proyectos de automatización del hogar, monitoreo de sensores, control de dispositivos y muchos otros proyectos.

2.3.2.1.1. ESP-01

El ESP-01 tiene un tamaño compacto y es fácilmente integrable en proyectos electrónicos. Tiene un conector de 2x4 pines, que permite la conexión de cables y otros componentes, y también cuenta con un chip *flash* incorporado que permite la actualización del *firmware*. Sin embargo, el ESP-01 tiene algunas limitaciones, como una cantidad limitada de pines de entrada/salida y una memoria *flash* limitada. A pesar de estas limitaciones, el ESP-01 sigue siendo muy popular debido a su bajo costo y facilidad de uso.

2.3.2.1.2. ESP-05

El ESP-05 es otro modelo del módulo ESP8266 que, al igual que el ESP-01, combina un microcontrolador y un módulo wifi en un solo chip. Sin embargo, a diferencia del ESP-01, el ESP-05 tiene un factor de forma diferente y cuenta con una antena externa para mejorar la señal wifi

2.3.2.1.3. ESP-12

El ESP-12 es otro modelo del módulo ESP8266 que, al igual que el ESP-01 y el ESP-05, combina un microcontrolador y un módulo wifi en un solo chip. Sin embargo, a diferencia de los otros dos modelos, el ESP-12 tiene más pines de entrada/salida y más memoria, lo que lo hace más versátil y adecuado para proyectos más complejos.

2.3.2.1.4. ESP-201

Es un modelo de módulo ESP8266 que también combina un microcontrolador y un módulo wifi en un solo chip. Es similar al ESP-12 en términos de funcionalidad. Es menos común que otros modelos de la serie ESP y puede ser más difícil de encontrar y trabajar con él, debido a su factor de forma y disposición de los pines.

2.3.2.1.5. Esp-8266

El módulo ESP8266 es un microcontrolador de bajo costo y alta conectividad inalámbrica, diseñado por la empresa china Espressif Systems. El módulo incluye un chip wifi integrado que permite a los dispositivos conectarse a redes inalámbricas y comunicarse con otros dispositivos a través de Internet.

El ESP8266 es muy popular en la comunidad de electrónica y programación, debido a su bajo costo, su capacidad para conectarse a Internet y su compatibilidad con muchos lenguajes de programación populares, como Python y C++. El módulo ESP8266 también es fácil de programar y configurar, lo que lo hace ideal para proyectos de IoT (Internet de las cosas), como sensores, dispositivos de monitoreo y control remoto.

2.3.2.1.6. Esp-8266 WeMos D1 mini

El WeMos D1 mini es una placa de desarrollo basada en el módulo ESP8266 que ofrece una amplia compatibilidad con diferentes lenguajes de programación y bibliotecas, lo que la hace ideal para proyectos de IoT y prototipado rápido.

Figura 10.

Esp-8266 WeMos D1 mini



Nota. Fotografía tomada al módulo ESP-8266 WeMos D1 mini desde la parte frontal y sin conexiones. Elaboración propia.

2.3.2.1.7. NodeMCU

NodeMCU es una plataforma de desarrollo basada en el microcontrolador ESP8266 de Espressif Systems. NodeMCU se enfoca en la creación de proyectos de Internet de las cosas (IoT, por sus siglas en inglés) y permite la programación en distintitos lenguajes de programación. Además, NodeMCU cuenta con conectividad wifi integrada, lo que lo hace ideal para proyectos de IoT que requieren una conexión a internet.

NodeMCU ha ganado popularidad entre los desarrolladores debido a su facilidad de uso, bajo costo y amplia comunidad de usuarios y desarrolladores. Además, NodeMCU es compatible con muchos de los IDE populares, como

Arduino IDE y MicroPython, lo que hace que sea más fácil para los desarrolladores trabajar con la plataforma.

2.3.2.2. Raspberry Pi

Raspberry Pi es una serie de placas de desarrollo de bajo costo y tamaño reducido que se utilizan para realizar una variedad de proyectos de electrónica, robótica e IoT. Actualmente, existen varios modelos de Raspberry Pi disponibles en el mercado, cada uno con características y especificaciones ligeramente diferentes.

2.3.2.3. Módulos programables de la Raspberry Pi

Son los componentes adicionales que se pueden utilizar para realizar tareas específicas que no trae por defecto la Raspberry Pi.

2.3.2.3.1. Pico *display*

Es una pantalla que permite realizar proyectos y visualizar si se desea proyectar algo y hacer más fácil de comprender lo deseado.

2.3.2.4. Tipos de Raspberry Pi

Existen diversos tipos de Raspberry y cada uno tiene características que cambian levemente entre unas, pero grandemente con otras. Pueden ser desde muy simples para bajo consumo eléctrico y aplicaciones pequeñas, hasta equipos completos para utilizarlos como computador.

2.3.2.4.1. Raspberry Pi 1 A

El Raspberry Pi 1 modelo A es uno de los primeros modelos de la serie Raspberry Pi. Cuenta con un procesador ARM11 de 700 MHz, 256 MB de RAM, una ranura para tarjeta SD y un puerto USB. A diferencia del modelo B que se verá más adelante, el modelo A no cuenta con un puerto Ethernet y solo tiene un puerto USB. El modelo A es más adecuado para proyectos que requieren bajo consumo de energía y no necesitan conectividad de red. A pesar de que es uno de los modelos más antiguos, el Raspberry Pi 1 model A todavía se utiliza en algunos proyectos debido a su bajo costo y tamaño reducido.

2.3.2.4.2. Raspberry Pi 1 B

Raspberry Pi 1 model B: el primer modelo de Raspberry Pi lanzado en 2012. Cuenta con un procesador ARM11 de 700 MHz, 512 MB de RAM y una salida HDMI.

2.3.2.4.3. Raspberry Pi 2 B

Raspberry Pi 2 model B: este modelo cuenta con un procesador ARM cortex-A7 de cuatro núcleos a 900 MHz, 1 GB de RAM y una salida HDMI.

2.3.2.4.4. Raspberry Pi 3 B

Raspberry Pi 3 model B: este modelo cuenta con un procesador ARM cortex-A53 de cuatro núcleos a 1,2 GHz, 1 GB de RAM y conectividad inalámbrica Wi-Fi y Bluetooth integrada.

2.3.2.4.5. Raspberry Pi 3 B+

Raspberry Pi 3 model B+: este modelo cuenta con un procesador ARM cortex-A53 de cuatro núcleos a 1,4 GHz, hasta 1 GB de SDRAM y un puerto HDMI.

2.3.2.4.6. Raspberry Pi 3 A+

Tiene un factor de forma similar al Raspberry Pi 3 modelo B+, pero es más pequeño y económico. Cuenta con un procesador ARM cortex-A53 de cuatro núcleos a 1.4 GHz, 512 MB de RAM, una ranura para tarjeta microSD, conectividad inalámbrica Wi-Fi y Bluetooth integrada, y una única conexión USB. El Raspberry Pi 3 modelo A+ es adecuado para proyectos donde se necesita una computadora pequeña y potente, pero no se requiere conectividad Ethernet.

2.3.2.4.7. Raspberry Pi 400

Es una computadora compacta que integra una Raspberry Pi 4 dentro de un teclado, lo que la hace muy conveniente para ser utilizada como una computadora de escritorio. La Raspberry Pi 400 cuenta con un procesador Broadcom BCM2711 de cuatro núcleos a 1.8 GHz, 4 GB de RAM, conectividad inalámbrica Wi-Fi y Bluetooth, y puertos USB 3.0 y Gigabit Ethernet. También cuenta con puertos HDMI para conectar un monitor y un puerto para tarjeta microSD para almacenamiento. La Raspberry Pi 400 se puede utilizar para una amplia variedad de aplicaciones, desde programación y desarrollo de software hasta educación, automatización del hogar y proyectos de electrónica (Raspberry Pi, 2022).

2.3.2.4.8. Raspberry Pi Pico

El Raspberry Pi Pico es un microcontrolador de placa única lanzado por Raspberry Pi en 2021. Es el primer dispositivo de este tipo producido por la compañía y cuenta con un procesador RP2040 de doble núcleo a 133 MHz, 264 KB de RAM y 2 MB de almacenamiento *flash*. También tiene conectividad USB, UART, SPI e I2C y soporta PWM, ADC, DAC y pines GPIO programables. El Raspberry Pi Pico es compatible con diversos lenguajes de programación como MicroPython, C++ y CircuitPython, lo que lo hace adecuado para una amplia gama de proyectos de electrónica y robótica (Raspberry Pi, 2022).

Figura 11.

Raspberry Pi Pico



Nota. Fotografía tomada a la Raspberry Pi Pico desde la parte frontal y sin conexiones. Elaboración propia.

2.3.2.4.9. Raspberry Pi Zero

El Raspberry Pi Zero es un microordenador de placa única lanzado en 2015. Es una versión más pequeña y económica del Raspberry Pi original, con

un procesador Broadcom BCM2835 de un solo núcleo a 1 GHz, 512 MB de RAM, una ranura para tarjeta microSD y un puerto mini-HDMI para la salida de video. También tiene un puerto micro-USB para la alimentación y la comunicación con otros dispositivos, así como una conexión para una cámara Raspberry Pi. El Raspberry Pi Zero es ideal para proyectos donde se requiere una computadora de bajo costo y tamaño reducido, como dispositivos IoT, reproductores de música o robots.

2.4. Lenguajes de programación

El lenguaje de programación es un lenguaje informático. Los programadores se utilizan para comunicar y desarrollar programas de software, aplicaciones, páginas web, *scripts* u otros conjuntos de instrucciones.

2.4.1. Definición

Un lenguaje de programación es un conjunto de reglas y símbolos que se utilizan para escribir programas informáticos. Es una herramienta que los programadores utilizan para comunicarse con la computadora y decirle qué hacer. Los lenguajes de programación pueden ser interpretados por el equipo o compilados antes de su ejecución.

El tipo de lenguaje de programación se puede clasificar en niveles bajos y lenguajes avanzados de programación. Aunque son simples en comparación con el lenguaje humano, son más complicados que los niveles bajos. Al mismo tiempo, en comparación con los compañeros de bajo nivel, el lenguaje avanzado tiene una mayor legibilidad, y su explicación requiere conocimiento profesional en la arquitectura de la computadora (Rudeus, 2022).

2.4.2. Usos de un lenguaje de programación

Para elegir el lenguaje de programación adecuado para un proyecto, se deben considerar diferentes factores, como el rendimiento, el tipo de aplicación, la seguridad, entre otros.

Se debe tener un propósito u objetivo claro en mente antes de elegir un lenguaje de programación para un proyecto. La elección del idioma depende en gran medida del tipo de aplicación. ¿Qué tipo de aplicación se va a construir? ¿Es una aplicación web o una aplicación móvil o su firmware integrado, entre otros? Este es el primero y uno de los factores más importantes que se debe considerar antes de elegir cualquier lenguaje de programación para un proyecto (Rudeus, 2022).

El lenguaje de programación para un proyecto debe basarse en las necesidades del negocio. Tal vez se considera que, dado que es el desarrollador, debe depender totalmente del usuario y de su elección para implementar cualquier idioma, y que se tiene la libertad de elegir la tecnología que se desee, pero esto no funciona en una organización de TI y puede ser contraproducente, hay que pensar en cómo todo se va a unir posteriormente, en que depende un sistema de otro, y esto aplica a muchos proyectos (Rudeus, 2022).

2.4.3. Entornos de desarrollo

Un entorno de desarrollo es un conjunto de herramientas y recursos que se utilizan para crear, depurar y probar software. Estos entornos proporcionan un conjunto integrado de herramientas, como un editor de código, compilador, depurador y otros, que permiten a los desarrolladores trabajar de manera eficiente y productiva (Pressman y Maxim, 2020).

2.4.3.1. Thonny

Thonny es un entorno de desarrollo integrado (IDE) para Python que está diseñado para ser fácil de usar y de aprender, especialmente para principiantes. Thonny cuenta con una interfaz de usuario simple e intuitiva que permite escribir, ejecutar y depurar código de Python de manera rápida y sencilla. Cuenta, además, con un editor de código integrado, un depurador, una consola de comandos interactiva y un administrador de paquetes, y que también es compatible con microcontroladores.

Figura 12.

IDE Thonny



Nota. Captura de pantalla del IDE Thonny en una plantilla en blanco. Elaboración propia.

2.4.4. Lenguajes de programación más comunes en microcontroladores

Existen varios lenguajes utilizados en microcontroladores. Algunos son preferidos ya sea por su optimización de recursos o por su facilidad de aprendizaje. Algunos de los más utilizados son:

2.4.4.1. Lenguaje C

Este es uno de los lenguajes más comunes para programar microcontroladores. Es un lenguaje de bajo nivel que proporciona un control preciso sobre el hardware del microcontrolador y es altamente eficiente en términos de velocidad y uso de memoria.

Figura 13.

Código de programación en lenguaje C

```
int main()
{
    int n, suma=0;
    printf("ingrese un numero: \n");
    scanf("%i", &n);

    for(int a=1; a<=n; a++){
        int x=0;
        for(int div=1; div<=a; div++){
            if(a%div==0 && a%1==0)
                x++;
            if(x==2){suma=suma+a;
            }
        }
    }
    printf("la suma de los numero primos antes de %i es: %i",n,suma);
    return 0;
}
```

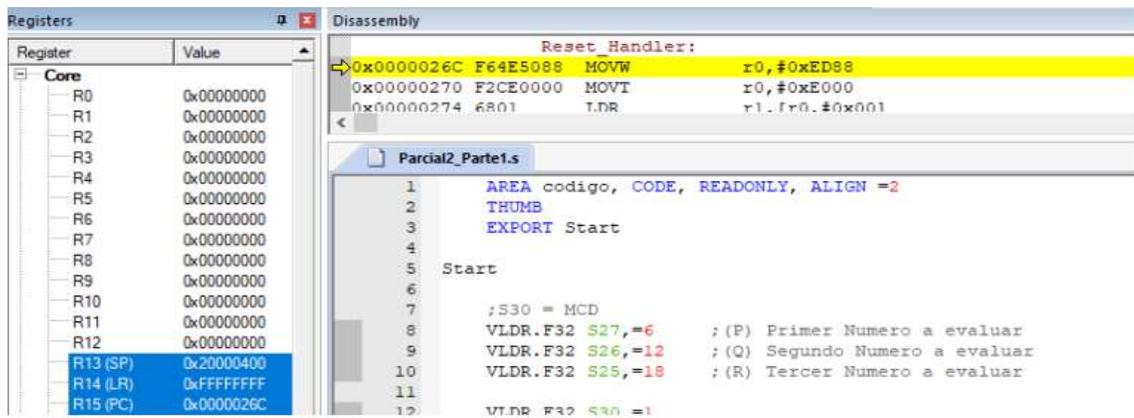
Nota. Programación realizada en código C. Elaboración propia, realizado con Thonny.

2.4.4.2. Ensamblador

Este es un lenguaje de programación de muy bajo nivel que se utiliza para programar microcontroladores de manera muy precisa y eficiente. Es un lenguaje muy poderoso, pero puede ser difícil de aprender y lleva más tiempo escribir programas.

Figura 14.

Código de programación en lenguaje ensamblador



Nota. Programación realizada en código ensamblador. Elaboración propia, realizado con Thonny.

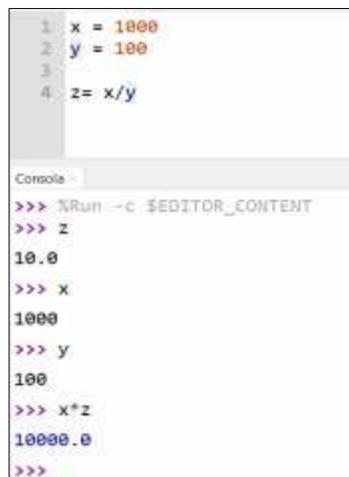
2.4.4.3. MicroPython

MicroPython es muy similar a Python, con la diferencia que MicroPython está diseñado específicamente para microcontroladores y se ha optimizado para ser más pequeño y eficiente en términos de uso de recursos de hardware, mientras que Python es un lenguaje de programación más general que se utiliza en una amplia variedad de aplicaciones de software. Aunque no es tan común como C/C++ o ensamblador, Python se está volviendo cada vez más popular

como un lenguaje de programación para microcontroladores. Es un lenguaje de programación de alto nivel que es fácil de aprender y escribir, y puede ser utilizado para una amplia variedad de aplicaciones.

Figura 15.

Código de programación en lenguaje MicroPython



```
1 x = 1000
2 y = 100
3
4 z = x/y

Console
>>> %Run -c $EDITOR_CONTENT
>>> z
10.0
>>> x
1000
>>> y
100
>>> x*z
10000.0
>>>
```

Nota. Programación realizada en código ensamblador. Elaboración propia, realizado con Thonny.

2.4.4.3.1. Convención PEP8

PEP8 es una normativa de estilo de codificación para el lenguaje de programación Python, que fue propuesta por Guido van Rossum, el creador de Python. La normativa PEP8 establece un conjunto de reglas y convenciones para el estilo de codificación en Python, con el objetivo de hacer que el código sea más legible, consistente y fácil de mantener.

Algunas de las principales directrices establecidas por la normativa PEP8 son las siguientes:

- El código debe seguir una sangría de 4 espacios.
- Las líneas no deben tener más de 79 caracteres.
- Las palabras en nombres de variables y funciones deben separarse con guiones bajos (dos_palabras).
- Las constantes deben nombrarse en mayúsculas.
- Los comentarios deben ser claros y concisos, y deben estar separados de las instrucciones de código por dos espacios.
- Debe haber una línea en blanco al final de cada archivo.

Además, la normativa PEP8 establece recomendaciones para el uso de espacios en blanco, la colocación de paréntesis, el uso de comillas y otros aspectos de la sintaxis de Python.

En resumen, la normativa PEP8 establece un conjunto de directrices y recomendaciones para el estilo de codificación en Python, con el objetivo de hacer que el código sea más legible, consistente y fácil de mantener. Seguir la normativa PEP8 puede ayudar a mejorar la calidad y la legibilidad del código en proyectos de Python.

2.5. Sensores

El sensor es un dispositivo que puede medir fenómenos físicos o químicos y convertirlos en electricidad. En otras palabras, los sensores pueden leer: temperatura, distancia, fuerza, presión, humedad, entre otros. Una vez que lea estas variables, las convertirá en señales eléctricas. Estas señales ayudan a tareas ilimitadas en el día a día (Mecafenix, 2022).

Figura 16.

Sensor ultrasónico HC-SR04



Nota. Ejemplo de un tipo de sensor. Elaboración propia.

2.5.1. Definición

Es un dispositivo que detecta o mide algún tipo de fenómeno físico o químico y convierte esta información en una señal eléctrica u otra forma de señal útil para su procesamiento y análisis. La digitalización de los productos para sensores tiene beneficios que exceden la capacidad de comunicarse fácilmente con los sistemas de control digitales. Como ejemplo, la mayoría de los sensores digitales incluyen una característica de modo de suspensión que apaga al sensor cuando no es necesario para las funciones del sistema (Arrow, 2015).

Al reducir la energía requerida, se mejora la vida útil de la batería de los sistemas remotos y móviles que dependen de la cosecha de energía o de la energía de la batería. Los sensores digitales también pueden aprovechar las rutinas de optimización de software que no están disponibles en el modo

analógico. Técnicas como el sobremuestreo y el filtrado digital pueden mejorar considerablemente la resolución del sensor y la discriminación de frecuencias. Y lo mejor de todo es que estas mejoras en el rendimiento no generan aumento en los costos, está todo en el software (Arrow, 2015).

2.5.2. Tipos de sensores

Dentro de los sensores existen diferentes tipos que utilizan los mismos principios de funcionamiento, pero se le agregan variaciones para ser utilizados en diferentes aplicaciones.

2.5.2.1. Sensores ópticos

Un sensor óptico es un tipo de sensor que utiliza la luz para detectar o medir un objeto o fenómeno. Estos sensores pueden utilizar diferentes principios ópticos, como la reflexión, la refracción, la absorción, la difracción, entre otros, para captar información sobre el objeto en cuestión. Algunos ejemplos comunes de sensores ópticos incluyen sensores de luz ambiental, sensores de proximidad, lectores de códigos de barras, entre otros.

2.5.2.2. Sensores de distancia

Un sensor de distancia es un tipo de sensor que se utiliza para medir la distancia entre un objeto y otro punto de referencia. Estos sensores pueden utilizar diferentes principios físicos para realizar estas mediciones, como el ultrasonido, la luz láser, la luz infrarroja, entre otros. La mayoría de los sensores de distancia funcionan emitiendo una señal y midiendo el tiempo que tarda en reflejarse desde el objeto de interés hasta el sensor. Esta información se puede utilizar para calcular la distancia del objeto con gran precisión. Los sensores de

distancia se utilizan comúnmente en aplicaciones como la robótica, la automatización industrial y la medición de distancias en topografía.

2.5.2.3. Sensores de humedad

Un sensor de humedad es un tipo de sensor que se utiliza para medir la cantidad de vapor de agua presente en el aire u otros gases. Estos sensores pueden utilizar diferentes principios físicos para realizar estas mediciones, como la capacitancia, la resistencia, la conductividad, entre otros. Los sensores de humedad se utilizan comúnmente en aplicaciones como el control de humedad en interiores, el monitoreo de la humedad en la agricultura y la horticultura, el control del clima en la producción de alimentos, entre otras actividades.

2.5.2.4. Sensores de presión

Son dispositivos que se utilizan para medir la presión de un fluido o gas en un sistema. Estos pueden utilizar diferentes principios físicos para realizar estas mediciones, como la deformación mecánica, la capacitancia, la resistencia eléctrica, entre otros. Algunos ejemplos comunes de sensores de presión incluyen manómetros, barómetros, transductores de presión, sensores piezoeléctricos, entre otros. Los sensores de presión se utilizan en una amplia variedad de aplicaciones, como la industria automotriz, la industria aeronáutica y la medición de presión arterial en la medicina.

2.5.2.5. Sensores sónicos

Un sensor sónico es un tipo de sensor que utiliza ondas sonoras para detectar la presencia de objetos y medir distancias. Estos sensores emiten una señal de sonido y miden el tiempo que tarda en reflejarse desde un objeto y volver

al sensor. Los sensores sónicos se utilizan comúnmente en aplicaciones como la detección de obstáculos en vehículos autónomos, la medición de niveles de líquidos, la medición de niveles de agua en ríos, entre otras actividades.

2.5.2.6. Sensores de temperatura

Un sensor de temperatura es un dispositivo que se utiliza para medir la temperatura de un objeto o ambiente. Algunos ejemplos comunes de sensores de temperatura incluyen termómetros digitales, termistores, termómetros infrarrojos, entre otros. Los sensores de temperatura se utilizan en una amplia variedad de aplicaciones, como la monitorización de la temperatura en sistemas de calefacción y refrigeración, en procesos industriales, la medición de la temperatura en aplicaciones médicas, incluso en los hogares, como en una estufa eléctrica.

2.5.2.7. Sensores magnéticos

Los sensores magnéticos son dispositivos que se utilizan para detectar campos magnéticos y medir la intensidad o dirección de estos. Estos sensores pueden utilizar diferentes principios físicos para realizar estas mediciones, como el efecto Hall, la magnetorresistencia, entre otros. Algunos ejemplos comunes de sensores magnéticos incluyen sensores de posición angular, sensores de velocidad de rotación e interruptores magnéticos. Los sensores magnéticos se utilizan en una amplia variedad de aplicaciones, como la navegación, la medición de campos magnéticos en la industria y la detección de la posición de piezas en la maquinaria.

2.6. Actuadores

Un actuador es un componente mecánico o electromecánico que convierte una señal de control en una acción física, como por ejemplo movimiento, fuerza o presión. Los actuadores se utilizan en una amplia variedad de aplicaciones industriales y de automatización, desde el control de válvulas y puertas hasta el control de motores y sistemas de posicionamiento (Equipo editorial de Revista Autocrash, 2015).

2.6.1. Tipos de actuadores

Los actuadores pueden ser de diferentes tipos y diseños, como por ejemplo actuadores hidráulicos, neumáticos, eléctricos, entre otros. Cada tipo de actuador tiene sus propias características y aplicaciones, y puede ser seleccionado en función de las necesidades específicas de la aplicación en la que se utilizará.

2.6.1.1. Válvulas

Una válvula es un componente mecánico utilizado para controlar el flujo de un fluido (líquido, gas o vapor) a través de un conducto o tubería. Básicamente, una válvula tiene un mecanismo que permite abrir o cerrar el flujo de un fluido, o bien regular su cantidad o presión. Las válvulas pueden ser de diferentes tipos y diseños.

Son parte de muchas máquinas que se usan todos los días y realizan múltiples funciones. Tres funciones comunes de las válvulas incluyen detener e iniciar el flujo, restringir el flujo y actuar como una válvula de retención para el flujo (Hidrocuta, 2018).

Figura 17.

Válvula de agua



Nota. Válvulas de control de agua tipo bola que aíslan una parte de la tubería y bloquean el paso de fluido al rotarse. Elaboración propia.

2.6.1.1.1. Válvulas mecánicas

Una válvula mecánica es una válvula que se acciona manualmente a través de una palanca, volante u otro mecanismo similar. Este tipo de válvula no utiliza ningún tipo de sistema hidráulico o electrónico para su funcionamiento y es comúnmente utilizada en aplicaciones industriales y domésticas.

2.6.1.1.2. Válvulas hidráulicas

Una válvula hidráulica es una válvula que se acciona mediante el flujo de un líquido, generalmente aceite o agua, a través de un sistema hidráulico. Este tipo de válvula puede ser accionada manualmente o mediante un sistema hidráulico más complejo que permite un mayor control de la apertura y cierre de la válvula.

2.6.1.1.3. Válvulas neumáticas

Una válvula neumática es una válvula que utiliza aire comprimido para su funcionamiento. Este tipo de válvula se utiliza comúnmente en aplicaciones industriales y de automatización, ya que permite un control preciso del flujo de aire comprimido y su distribución en diferentes partes de un sistema neumático. Las válvulas neumáticas se pueden accionar de diferentes formas, como por ejemplo mediante una palanca manual o una señal eléctrica.

2.6.1.1.4. Válvulas electrónicas

Una válvula electrónica es una válvula que se acciona mediante señales eléctricas, en lugar de un mecanismo manual o un sistema hidráulico. Este tipo de válvula es muy común en aplicaciones industriales y de automatización, ya que permite un mayor control y precisión en el funcionamiento de la válvula. Las válvulas electrónicas pueden ser programadas para realizar diferentes tareas, como la apertura y cierre de la válvula en función de ciertas condiciones, o la regulación de la cantidad de fluido que se permite pasar a través de la válvula.

2.6.1.2. Relevador

Es un dispositivo eléctrico que permite controlar un circuito eléctrico a través de otro circuito eléctrico separado. Funciona como un interruptor electromagnético que se activa mediante una corriente eléctrica que fluye por el circuito de control.

El relé consta de un núcleo magnético, un conjunto de contactos eléctricos y un resorte de retorno. Cuando se activa la corriente en el circuito de control, se genera un campo magnético que atrae al núcleo magnético, lo que hace que los contactos eléctricos se cierren y se establezca la conexión eléctrica en el circuito controlado. Al cortar la corriente en el circuito de control, el campo magnético desaparece y el resorte de retorno devuelve el relé a su posición inicial, lo que hace que los contactos eléctricos se abran y se interrumpa la conexión eléctrica en el circuito controlado.

Los relés se utilizan en una amplia variedad de aplicaciones, como el control de motores, la protección de circuitos, el control de iluminación y el control de temperatura, entre otras. También se utilizan en circuitos de automatización industrial y en sistemas de control de procesos.

Figura 18.

Relevador o relé



Nota. Fotografía de un relevador de baja potencia de un polo. Elaboración propia.

2.6.1.2.1. Módulo de relé

Es un módulo compuesto de uno o varios relevadores, que viene diseñado ya en una placa electrónica solamente para realizar el cableado e instalación, evitando tener que soldar el dispositivo, lo que lo hace versátil si es necesario retirarlo del lugar o reemplazarlo por otro distinto.

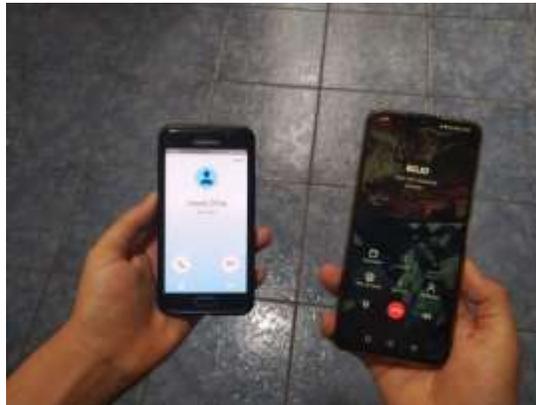
2.7. Comunicación inalámbrica

La comunicación inalámbrica es un método de transmisión de información que se realiza a través del aire, sin la necesidad de cables o conexiones físicas. En otras palabras, se trata de una forma de comunicación que utiliza ondas electromagnéticas para enviar y recibir datos a través del espacio. Las

comunicaciones inalámbricas pueden realizarse en diferentes frecuencias y bandas de radio, y pueden utilizarse para una amplia variedad de aplicaciones, desde la transmisión de datos a larga distancia hasta la comunicación entre dispositivos cercanos, como en el caso de las redes inalámbricas wifi o Bluetooth.

Figura 19.

Comunicación inalámbrica entre dos teléfonos celulares



Nota. Realización de una llamada telefónica entre dos teléfonos celulares. Elaboración propia.

2.7.1. Características

Los estándares existentes para transmisión inalámbrica de datos han sido y son ampliamente utilizados en entornos poco ruidosos y distancias próximas (viviendas, oficinas, entre otros ámbitos), pero su baja fiabilidad y su corto alcance han impedido su penetración masiva en el entorno industrial. No obstante, en aras de satisfacer las necesidades de la industria y sobre todo tras la aparición del IoT, en los últimos años se han desarrollado tecnologías inalámbricas que permiten su aplicación en la mejora industrial de los procesos y, por tanto, es necesario conocerlas para saber cuál aplicar según sean las necesidades inalámbricas.

2.7.2. Ventajas y desventajas

Algunas de las ventajas de la comunicación inalámbrica son la movilidad, flexibilidad y rentabilidad. La popularidad de las redes inalámbricas se debe principalmente a su comodidad, rentabilidad y facilidad de integración con otras redes y componentes de red. Las redes inalámbricas se han consolidado como una tecnología imprescindible. Con la llegada de la pandemia y la implementación del trabajo remoto, las redes inalámbricas se han convertido en una herramienta muy útil para los trabajadores remotos y para la continuidad de las compañías (Soto, 2019).

La tendencia a la movilidad hace que cada vez sean más utilizados los sistemas inalámbricos. Entre las distintas ventajas se encuentran las siguientes:

- Permiten una amplia libertad de movimientos.
- Permiten reubicar las estaciones de trabajo con facilidad.
- Evitan la necesidad de establecer cableado.
- La instalación de la red es mucho más rápida que con cables.
- Una red inalámbrica tiene menores costos, porque se ahorra lo que se gasta en cables.
- Permiten la cobertura en puntos de difícil conexión mediante cables.
- Permiten una ampliación de redes locales cableadas.
- Permiten la fácil expansión o limitación de usuarios en la red con solo añadir o quitar módulos.

Así como poseen varias ventajas, también poseen desventajas. Las principales desventajas de la comunicación inalámbrica son las interferencias, riesgos de seguridad y alcance limitado.

Son propensas a las interferencias: las redes inalámbricas son muy propensas a las interferencias; por lo tanto, la niebla, la radiación, las señales de radio o cualquier otra interferencia similar pueden causar un mal funcionamiento en una red inalámbrica. Además, cuando hay demasiados usuarios en la misma zona, la banda aérea por la que se transmiten las señales puede sobrecargarse (Soto, 2019).

Seguridad: si una red inalámbrica no se instala correctamente o no se mantiene correctamente, puede causar graves amenazas a la seguridad. Una red inalámbrica no requiere la conexión de componentes físicos como los cables, solo necesita un adaptador inalámbrico, lo cual aumenta automáticamente el riesgo de piratería informática, ya que se puede tener un acceso más fácil a la red (Soto, 2019).

2.7.3. Ruido

En telecomunicaciones, el ruido se refiere a cualquier señal indeseada o interferencia que afecta la calidad de una señal de comunicación. El ruido puede ser generado por diferentes fuentes, como por ejemplo la interferencia electromagnética de otros dispositivos, la degradación de la señal en el medio de transmisión (como por ejemplo la atenuación o el eco), o incluso por errores en el procesamiento de la señal. El ruido puede causar errores en la transmisión de datos, lo que puede afectar la calidad de la comunicación y limitar la velocidad y el alcance de la transmisión. Por lo tanto, la reducción del ruido es un objetivo importante en el diseño y la implementación de sistemas de telecomunicaciones.

Figura 20.

Ruido en señal de cable en un televisor



Nota. Televisor con interferencia o ruido. Elaboración propia.

2.7.4. Estándares más utilizados

Las comunicaciones inalámbricas tienen muchos usos y muchas aplicaciones, por lo que existen diversos estándares que las regulan.

2.7.4.1. Estándar IEEE 802.11

El estándar IEEE 802.11, también conocido como wifi, es un conjunto de especificaciones técnicas que definen las características y requerimientos de las redes inalámbricas de área local (WLAN). Este estándar establece las reglas para la comunicación inalámbrica entre dispositivos electrónicos, como computadoras, *smartphones*, tabletas, entre otros, y las redes de acceso inalámbricas.

El estándar IEEE 802.11 define diferentes tipos de tecnologías y características de redes inalámbricas, como las velocidades de transmisión, las frecuencias de radio utilizadas, la distancia de cobertura, la seguridad, entre otros. Estas tecnologías son mejoradas y actualizadas de manera regular, lo que permite que las redes wifi sean cada vez más rápidas y eficientes.

El estándar IEEE 802.11 se divide en diferentes subestándares, que se denominan con letras. Los subestándares más comunes son 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac y 802.11ax. Cada subestándar tiene diferentes características y especificaciones técnicas que lo hacen adecuado para diferentes tipos de aplicaciones y entornos.

El estándar IEEE 802.11 ha tenido un gran impacto en la tecnología inalámbrica y ha permitido el desarrollo de una amplia variedad de aplicaciones y dispositivos inalámbricos. Las redes wifi son utilizadas en hogares, empresas, lugares públicos y en muchas otras aplicaciones, lo que ha llevado a una mayor conectividad y acceso a Internet en todo el mundo.

Los dispositivos wifi utilizan dos bandas de frecuencia: 2.4 GHz y 5 GHz. La banda de 2.4 GHz es más popular debido a su mayor alcance, mientras que la banda de 5 GHz tiene menos interferencias y es más adecuada para

aplicaciones de alta velocidad. El máximo rango de alcance teórico de la especificación 802.11b es de 100 metros en interiores y 140 metros en exteriores, aunque la distancia real de cobertura puede variar dependiendo de diversos factores ambientales y de interferencia.

El estándar IEEE 802.11 especifica varios protocolos de seguridad, incluyendo WEP, WPA y WPA2, que proporcionan diferentes niveles de protección para las redes inalámbricas.

2.7.5. Ondas electromagnéticas

Son una forma de energía que se propaga a través del espacio en forma de ondas que combinan campos eléctricos y magnéticos oscilantes perpendiculares entre sí y en dirección de propagación. Estas ondas no necesitan un medio material para propagarse y pueden viajar a través del vacío.

2.7.6. Internet de las cosas

IoT son las siglas de *Internet of Things* (Internet de las cosas, en español), y se refiere a la conexión de dispositivos y objetos cotidianos a Internet, permitiendo que estos puedan recopilar y compartir datos de manera remota.

Esto significa que los objetos cotidianos, desde electrodomésticos hasta vehículos, dispositivos médicos y sensores industriales, pueden estar conectados a la red y ser controlados y monitoreados a través de Internet.

El objetivo principal de IoT es aumentar la eficiencia y reducir costos en muchos aspectos de la vida cotidiana y en la industria, así como permitir el desarrollo de soluciones innovadoras y nuevas oportunidades de negocio.

2.8. Software para diseño

El software de diseño es un tipo de programa informático que se utiliza para crear o modificar diseños, ya sea en 2D o en 3D. Estos programas permiten a los usuarios crear modelos digitales, dibujos técnicos, diagramas, animaciones, entre otros, y suelen ser utilizados en una amplia variedad de campos, como la arquitectura, la ingeniería, el diseño gráfico, la moda, entre otros.

2.8.1. Tinkercad

Tinkercad es una herramienta de diseño y modelado 3D en línea gratuita y fácil de usar que permite a los usuarios crear diseños y modelos 3D en su navegador web. Es una aplicación de modelado 3D basada en la nube, lo que significa que no es necesario descargar ni instalar ningún software para usarlo. Además, Tinkercad se integra con otras herramientas y plataformas de diseño 3D, lo que hace que la transferencia de archivos y el trabajo colaborativo sean más sencillos (Tinkercad, 2023).

2.8.2. Autodesk Eagle

Es un software de diseño de circuitos electrónicos que permite la creación de esquemas y la disposición de componentes electrónicos en una placa de circuito impreso (PCB, por sus siglas en inglés). El software cuenta con una amplia biblioteca de componentes electrónicos, lo que facilita la selección y disposición de los componentes en el diseño de la PCB. También cuenta con herramientas de enrutamiento de pistas que permiten la conexión de los componentes en la placa de circuito impreso.

2.8.3. Lucidchart

Es una herramienta en línea de diagramación y visualización de ideas que permite crear diagramas, mapas mentales, diagramas de flujo, organigramas y otros tipos de diagramas para comunicar de manera clara y eficiente ideas y conceptos. Lucidchart cuenta con una interfaz gráfica fácil de usar, lo que la hace accesible para personas con diferentes niveles de experiencia en diagramación y visualización.

Ofrece una interfaz gráfica fácil de usar, tiene funciones colaborativas y se integra con otras herramientas populares para hacerla una opción versátil y adaptable para diferentes entornos de trabajo.

2.8.4. Microsoft Office

Microsoft Office es una *suite* de aplicaciones de software empresarial que incluye programas como Microsoft Word, Excel, PowerPoint, Outlook, Access, entre otros. Cada programa de Office está diseñado para realizar tareas específicas, como procesamiento de textos, hojas de cálculo, presentaciones, correo electrónico, gestión de bases de datos y toma de notas.

Microsoft Office es utilizado por millones de personas en todo el mundo para mejorar la eficiencia y la productividad en el trabajo, la escuela y la vida personal. Las aplicaciones de Office son compatibles con una amplia gama de dispositivos y plataformas, incluidas las computadoras de escritorio, portátiles, tabletas y teléfonos móviles.

Además, Microsoft Office ofrece características y herramientas avanzadas, como la colaboración en tiempo real, la integración con otros

servicios en línea y la capacidad de crear y editar documentos en línea, lo que permite a los usuarios trabajar de forma remota y colaborar con otras personas en diferentes ubicaciones.

3. MARCO METODOLÓGICO

3.1. Dispositivos y materiales por utilizar

El primer apartado de este tercer capítulo de la investigación se encarga de describir puntualmente los distintos e importantes dispositivos y materiales que se utilizan en las tareas llevadas a cabo en el proyecto en estudio.

3.1.1. Placa Raspberry Pi Pico

La placa programable principal es el núcleo del proyecto, por lo que la elección de la ideal para el proyecto es de suma importancia. Si bien varias pueden ser utilizadas para distintos proyectos, siempre existen criterios a utilizar para escoger entre algún grupo selecto para adecuarlos al proyecto deseado.

En esta ocasión se seleccionó la placa Raspberry Pi Pico, debido a su alto desempeño y bajo costo de energía. Al ser un lugar que posee energía eléctrica por medio de un sistema tipo presa, que se basa en convertir la energía potencial en energía eléctrica, se debe optimizar el uso de esta. Además de esto, la Raspberry Pi Pico es de un tamaño muy pequeño y así no supondrá ningún problema donde se instale, y tiene módulos que se le pueden adaptar si se desea integrar algún añadido extra en un futuro.

El diseño contempla que la Raspberry Pi Pico se conecte directamente a un UPS para evitar que si no hay energía eléctrica se apague el sistema, sino que siga funcionando a pesar de esto, el tiempo que seguirá funcionando depende de la capacidad de la batería de emergencia. La Raspberry Pi Pico va

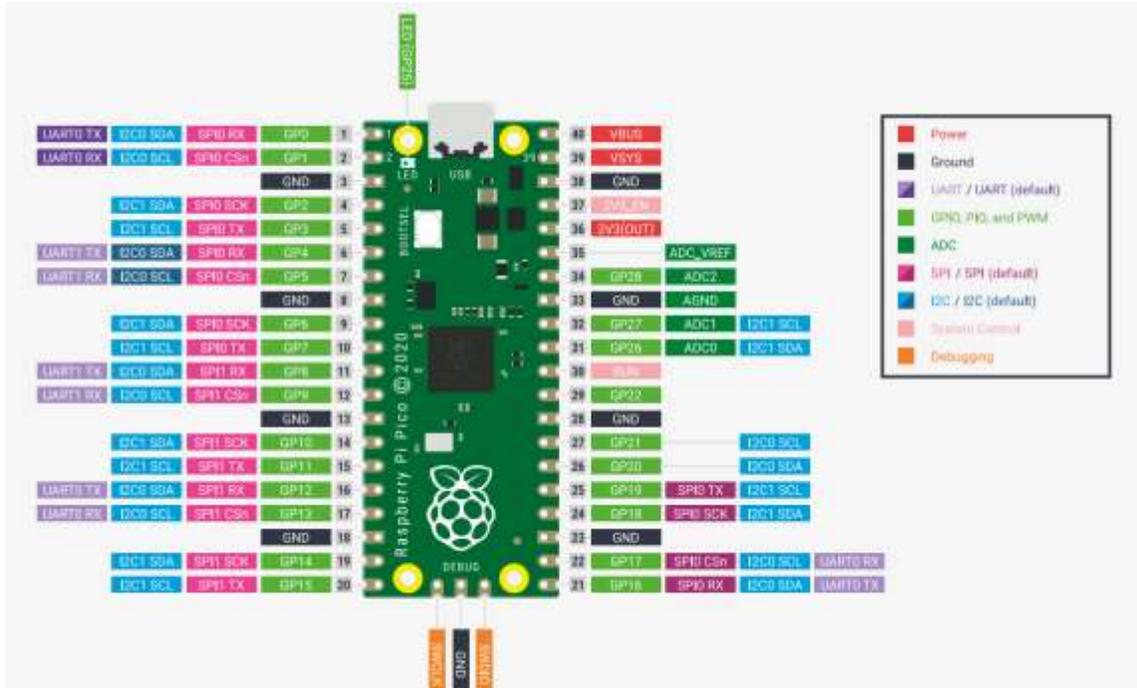
conectada directamente con un módulo Esp8266 al *display*, al sensor de flujo y al módulo del relé que activa una electroválvula.

Las características principales de la Raspberry Pi Pico son:

- Chip principal basado en ARM Cortex M0+ de doble núcleo corriendo a 133 MHz.
- Memoria SRAM de 254 KB.
- 2MB de QSPI Flash.
- Voltaje de trabajo de 1.8V – 5.5V.
- Temperatura de trabajo de -20°C – 84°C.
- 26 pines GPIO.
- 4 puertos de ADC.

Figura 21.

Imagen de los pines de la Raspberry Pi Pico



Nota. La imagen muestra los pines y sus funciones. Obtenido de Raspberry Pi (2023). *Raspberry Pi Pico datasheet.* (<https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>), consultado el 10 de febrero del 2023. De dominio público.

Los puertos GPIO se utilizaron en pull up, que es una configuración de resistencias, para evitar el uso de resistencias adicionales y así utilizar únicamente los módulos deseados, esto genera que el pulso de activación se realice mediante la detección de un “0”.

3.1.2. Pantalla Pico *display* OLED

Los seres humanos pueden realizar tareas que se indican en un manual o paso a paso, pero siempre es más sencillo un trabajo si, además de esto, se

agrega una imagen para visualizar y comprender mejor un procedimiento, por lo que se utilizará esta pantalla para ayudar a quien seleccione el modo de riego y la cantidad de agua que llegue a las plantas a visualizar más fácilmente lo que se está realizando.

La pantalla se programó en microPython para poder visualizar de manera sencilla si se estaba utilizando el modo 1 o 2 de configuración de riego, y se puede seleccionar con botones a los lados cuál utilizar. También se configuró para que con botones pulsadores a los lados se pueda aumentar de manera manual, por si no es suficiente o si se desea bajar más el ritmo de riego.

Se utilizó la librería SSD1306 para habilitar la pantalla OLED, se asignó el pin GPI 19 para la conexión del SCL y el pin GPI 18 para la conexión del SDA. Se asignaron en la conexión I2C1. En esta se colocan las dimensiones de la pantalla OLED, en este caso se colocó 128*64píxeles (figura 38).

Se diseñaron distintas pantallas para indicar los distintos procesos, incluyendo una pantalla inicial y de establecimiento de conexión que muestra que se están conectando las dos ESP-8266 entre ellas.

Figura 22.

Pantalla OLED estableciendo conexión



Nota. Estableciendo conexión entre los ESP-8266. Elaboración propia.

Hay pantallas de funcionamiento y opciones, para seleccionar si se desea el tipo de riego de la modalidad 1 o la modalidad 2. Estos se seleccionan mediante un botón de opciones y posteriormente, dependiendo la modalidad deseada, se presiona el botón correspondiente. Además, la pantalla también muestra la humedad externa a los cultivos, que es la que indica si hay lluvia o una fuga en el área donde se ubica el sensor de humedad.

Figura 23.

Pantalla OLED iniciada



Nota. Figura que muestra la pantalla OLED durante el funcionamiento inicial, sin haber seleccionado un modo de riego. Elaboración propia.

Figura 24.

Pantalla OLED en modos de operación



Nota. Figura que muestra la pantalla OLED en los modos de operación. Elaboración propia.

Figura 25.

Pantalla OLED del sistema en un modo de operación



Nota. Figura que muestra la pantalla OLED funcionando en un modo de operación seleccionado y el porcentaje de humedad del sensor. Elaboración propia.

3.1.3. Módulo ESP-8266 WeMos D1 mini

Debido a que el número de sensores en el proyecto propuesto no es muy elevado, se utilizó el módulo Esp-8266 WeMos D1 mini, se prefirió debido a su sencillez y rapidez en implementar junto a otros dispositivos, y como en este caso no será el núcleo central, sino la Raspberry Pi Pico, no se requiere mucho procesamiento por parte del ESP8266, y así es posible ahorrar consumo eléctrico. El módulo ESP8266 se conecta a un regulador de voltaje que se conecta directamente en la pared y en configuración de 3.3V.

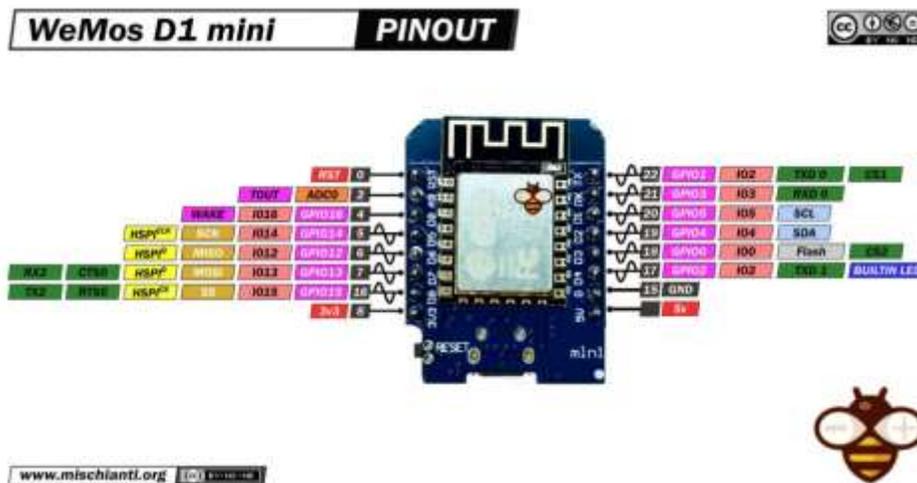
Las características principales del Esp-8266 WeMos D1 mini son:

- Módulo ESP-12F o ESP-12E

- Voltaje de operación entre 2.5V-3.6V
- 11 pines GPIO, con puertos de UART y ADC
- 32 Mbit de memoria *flash* SPI
- Modo de bajo y muy bajo consumo
- Transmisión de hasta 11Mbps 802.11b y 54Mbps en 802.11g
- Soporta entre -40°C hasta 125°C
- Antena en PCB y transmisión de 2.4Ghz y de igual recepción

Figura 26.

Diagrama de pines del ESP8266 WeMos mini D1



Nota. Imagen que muestra los pines de la ESP8266 WeMos mini D1. Obtenido de Mischianti (2021). *Pines ESP8266 We Mos mini D1.* (<https://www.mischianti.org/2021/04/26/wemos-d1-mini-high-resolution-pinout-and-specs/>), consultado el 12 de febrero del 2023. De dominio público.

La conexión del Esp-8266 se conectó a la Raspberry Pi Pico con el fin que se pueda dar la unión del control de válvulas que no se encuentran en la misma área y que también son accionadas por otro módulo Esp-8266 WeMos, pero que controlan otra tubería. El módulo Esp-82266 WeMos se conecta mediante

conexión inalámbrica al otro módulo Esp-8266 que se encuentra en otra área. El tipo de conexión inalámbrica que se utilizó fue la wifi que se rige por el estándar 802.11b.

Figura 27.

ESP funcionando como servidor

```
MicroPython v1.20.0 on 2023-04-26; ESP module with ESP8266
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

#22 ets_task(4020f57c, 29, 3fff9508, 10)
servidor iniciado, esperando conexion
('192.168.4.3', 39754)
GET / HTTP/1.1
Host: 192.168.0.9
Connection: keep-alive
Cache
-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: es-419,es;q=0.9,en;q=0.8
```

Nota. Módulo ESP8266 iniciando un servidor privado y listo para la comunicación. Elaboración propia, realizado con Thonny.

Una vez se configura el servidor, se realiza la conexión del cliente. La siguiente imagen muestra cómo se conectó de manera efectiva el cliente al servidor.

Figura 28.

ESP funcionando como cliente

```
MicroPython v1.20.0 on 2023-04-26; ESP module with ESP8266
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Dispositivo conectado a la red wifi
('192.168.58.49', '255.255.255.0', '192.168.58.97', '192.168.58.97')
```

Nota. Elaboración propia, realizado con Thonny.

3.1.4. Válvula electrónica

La válvula es el actuador principal en este diseño. Se conecta a los módulos de relé, ya que, si se conecta directamente a las placas programables, ya sea a la Raspberry Pi Pico o al módulo Esp-8266 WeMos, no lo soportan, por lo que estas válvulas se alimentan de manera independiente con los módulos de relé y solo dependen de ser accionadas por los módulos correspondientes.

Debido a que las válvulas no regulan la cantidad de litros que pasan en cierto tiempo, estas se activarán y desactivarán constantemente para mantener el goteo constante en los árboles deseados.

La electroválvula por utilizar es de $\frac{3}{4}$ de pulgada y a 12V en corriente continua. No hay problema si se utiliza una de corriente alterna, solo que el módulo de relé deberá ser correctamente adaptado a una conexión de corriente alterna. Es útil que no sea muy bajo el tamaño de la salida de la válvula o puede afectar a la cantidad de riego en cada planta.

3.1.5. Sensores

En los proyectos de automatización debe tomarse algún valor de medida para tener referencia durante el trabajo o para tomar en cuenta algún fenómeno físico a medir, ya que hay muchos tipos de sensores y se utilizan para diversas tareas e incluso no necesariamente para lo que fueron diseñados. Se pueden utilizar algunos sensores diseñados para otro objetivo y se pueden adaptar a una tarea distinta a la original.

3.1.5.1. Botón pulsador

Un botón es el actuador más sencillo que hay, ya que con solo presionarlo o soltarlo, dependiendo del tipo de botón, realiza una acción. Desde las teclas en un computador, encender un bombillo en el hogar o hasta un botón de emergencia de paro en la industria, los botones son muy utilizados. En este caso se utilizaron botones para navegar entre las configuraciones del sistema final y utilizar las modalidades de riego 1 y 2.

3.1.5.2. Sensor de humedad de suelo YL-69

Las plantas son seres vivos que, al igual que los humanos, requieren de agua para vivir, pero un exceso de esta puede provocar problemas en ellas. Con este motivo se ha decidido utilizar para el diseño el sensor de humedad de suelo YL-69, de esta manera el sensor capta si hay un exceso de humedad o si está seco donde se coloca el sensor. Esto ayuda a decidir al operario si regar o no la planta, independientemente si hubo lluvia o no.

El sensor de humedad se cablea de tal manera que la cantidad de cableado sea la suficiente para que el módulo Esp-8266 se encuentre en un área

cubierta de la lluvia, ya que solamente se necesita que el sensor de humedad se exponga.

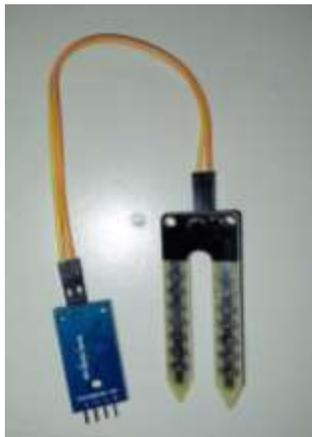
Debido a que el sensor de humedad está midiendo constantemente mediante el envío de pulsos eléctricos, con el paso del tiempo se deteriora más rápido, por lo que para evitar esto se diseñó para estar conectado a un módulo relé y se programaron los tiempos de activación del relé para que solo esté activo cierto periodo de tiempo y luego se encuentre apagado.

Las características principales del sensor de humedad YL-69 son:

- Voltaje de entrada de 3.3V-5V
- Voltaje de salida de 0-4.2V
- Sensibilidad ajustable mediante potenciómetro en el módulo en conjunto
- Led indicador de encendido y conmutación

Figura 29.

Sensor de humedad YL-69



Nota. Imagen del sensor seleccionado. Elaboración propia.

3.1.5.3. Sensor de flujo YF-B6

Es un dispositivo utilizado para medir la cantidad de líquido que fluye a través de una tubería en un sistema de riego o en otras aplicaciones industriales. Este sensor detecta movimientos en un componente interno para calcular la velocidad del flujo y generar una señal de pulso que es proporcional a la cantidad de líquido que fluye. Cada pulso corresponde a una cantidad específica de líquido que pasa a través del sensor.

La frecuencia de los pulsos se puede utilizar para determinar el caudal o la velocidad del flujo. El sensor YF-B6 es fácil de instalar y se puede conectar a un microcontrolador o a otros dispositivos electrónicos para controlar y monitorear el flujo de líquido en tiempo real. Las características del sensor de flujo YF-B6 son las siguientes:

- Voltaje de trabajo entre 5V-15V
- Control de flujo entre 1-30L/min
- Temperatura de trabajo entre 0°C -80°C
- Sensor con efecto *hall*
- Presión de trabajo <1.75Mpa
- Diámetro 25mm

Figura 30.

Sensor de flujo YF-B6



Nota. Imagen del sensor de flujo seleccionado. Elaboración propia.

3.2. Dispositivos hidráulicos por utilizar

Para el transporte del agua hacia una dirección deseada se utilizan distintos métodos, uno de ellos es la distribución por medio de tuberías. En este caso se tiene la tubería principal, acoples para distribución de ramificaciones, reductor de presión, goteros, entre otros.

Debido a que la presión puede ser demasiada en ciertas ocasiones, se requiere un regulador y después controlar si efectivamente está funcionando el reductor previo. En este caso se seleccionó el reductor de presión Riegolux 291081, que se puede encontrar en páginas de venta en internet, no es necesario utilizar este mismo, pero sí poder regular a menos de 4 atmósferas, que es la resistencia máxima de estos tubos según el fabricante.

El gotero penetra directamente en la tubería de polietileno de distribución. Estos se colocan lo más cercano a las plantas, y se puede regular la cantidad de agua que se desea desenroscando la tapadera. A partir de una entrada se puede distribuir en dos o más direcciones distintas. Se seleccionó principalmente para la ramificación principal un acople tipo t, para así poder distribuir en las distintas locaciones.

Se colocan válvulas manuales en puntos de separación para aislar las distintas secciones de las tuberías en caso de realizar un cambio o mantenimiento a estas.

3.3. Lenguaje de programación por utilizar

En un proceso de automatización se requiere de algún tipo de programación en uno o varios dispositivos, esto dependerá del tamaño del proyecto a realizar. También, dependiendo del tipo de proyecto y consecuentemente del tipo de dispositivo a utilizar, dependerá el tipo de configuración o programación a realizar en ello. En este caso, como se utilizaron las placas programables Raspberry Pi Pico y Esp-8266 WeMos D1 mini, se utilizaron lenguajes de programación relacionados a cada uno de ellos.

3.3.1. Micro-Python

Ya que las placas programables permitían el uso de un mismo lenguaje de programación, fue otro punto por el cual se realizó el diseño con estos dispositivos. En este caso se utilizó MicroPython, que es una versión de Python, pero optimizada para su uso en microcontroladores.

Se simplifica el que ambos dispositivos utilicen el mismo lenguaje de programación, ya que se programan de una manera muy parecida. El diseño del lenguaje se rigió mediante la convención PEP8 para facilitar la comprensión de la programación.

Figura 31.

Texto de pantalla OLED funcionando en IDE Thonny

```
>>> %Run -c $EDITOR_CONTENT
INICIANDO SISTEMA
  Estableciendo conexión...
Conexión establecida...
botón de configuración presionado
entrando a configuraciones

botón para seleccion modo 1 presionado
modo 1 seleccionado
abriendo electroválvula
iniciando riego, 20 min.

botón de configuración presionado
entrando a configuraciones

botón para seleccion modo 2 presionado
modo 2 seleccionado
abriendo electroválvula
iniciando riego, 30 min.
```

Nota. Ejemplo de texto de pantalla OLED. Elaboración propia, realizado con Thonny.

3.4. Diseño

Todo equipo debe estar situado físicamente en un espacio adecuado para evitar que se dañe por malas prácticas, por accidentes, entre otros. Debido a que las placas programables son sensibles a movimientos bruscos, al agua, polvo, climas extremos, entre otros, se debe comprar o diseñar una base para estos, de

tal manera que sean protegidos contra lo que no están diseñados a soportar. En este caso se planteó el uso de cajas protectoras IP66.

Figura 32.

Caja de protección IP66



Nota. Fotografía de cajas protectoras para el sistema electrónico. Elaboración propia.

Se optó por esta opción para un fácil transporte, precio accesible y fácil instalación, en caso hubiese una falla, se realizarán modificaciones físicas a la finca o algún otro caso ajeno al sistema.

La manguera principal se separa cada seis metros, ya que las plantas en esta finca tienen esa separación, por lo que, además, se coloca un gotero en cada planta. La tubería que viene de la electroválvula pasa por un regulador previo en el caso del área “A” y después en el caso del área “B”, posteriormente la tubería debe ser de PVC y de un diámetro más grande que el diámetro de la manguera principal de riego, ya que, de igual manera, la manguera de riego es flexible y se coloca a presión, al igual que sus adaptadores de separación, en

este caso tipo T. Finalmente deben colocarse válvulas en las ramificaciones para poder separarlas en caso de necesitar alguna reparación o solo aislar la tubería de esa sección.

3.5. Programa de simulación

Las simulaciones son una parte esencial en muchos proyectos, ya que antes de llevar a cabo un proyecto en ocasiones es obligatorio observar cómo funciona el diseño propuesto antes de ponerlo en marcha. También es útil para observar desde una perspectiva completa o distinta cómo se ve el resultado y si hay algo que no está implementado, si el diseño tiene falencias, si no funciona o incluso solamente para poder observar de una manera más agradable y sencilla el resultado propuesto.

3.5.1. Tinkercad

Luego de realizar la selección de las placas programables a utilizar, los componentes mecánicos a implementar y la manera en que se relacionan unos con los otros, se procede a realizar el diseño del circuito, para visualizar cómo quedarán conectados los componentes del proyecto completo. Para esto se decidió utilizar el software llamado Tinkercad, ya que es muy fácil de utilizar y permite ver de manera agradable a la vista cómo queda el esquema general de la implementación a grandes rasgos y así tener una idea más clara de lo propuesto.

4. DISEÑO PROPUESTO

4.1. Diseño propuesto

En el presente capítulo, por medio de varios incisos específicos para cada elemento y subtema, se explica detalladamente el diseño que se propone.

4.1.1. Diseño de estructura

El diseño específico de la estructura propuesta para dar solución al problema de que se encarga este proyecto se presenta en la siguiente figura:

Figura 33.

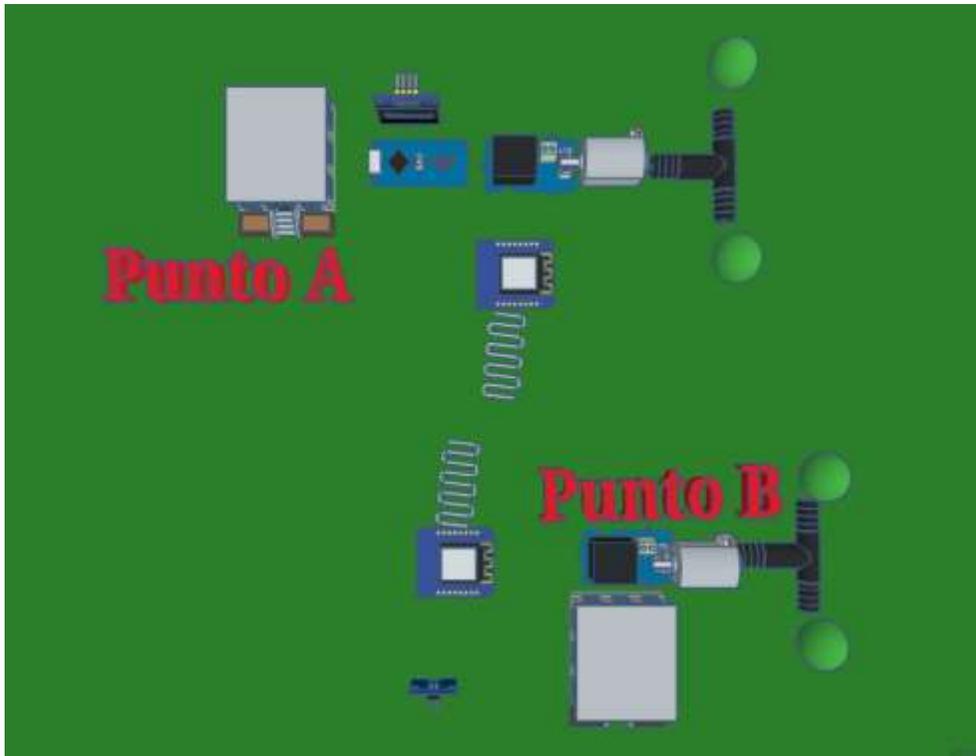
Diseño de la estructura



Nota. Diseño de modelo en 3D realizado en Tinkercad para una mejor comprensión del prototipo. Elaboración propia, realizado con Tinkercad.

Figura 34.

Diseño de la estructura, vista superior



Nota. Diseño de modelo en 3D realizado en Tinkercad para una mejor comprensión del prototipo desde perspectiva superior. Elaboración propia, realizado con Tinkercad.

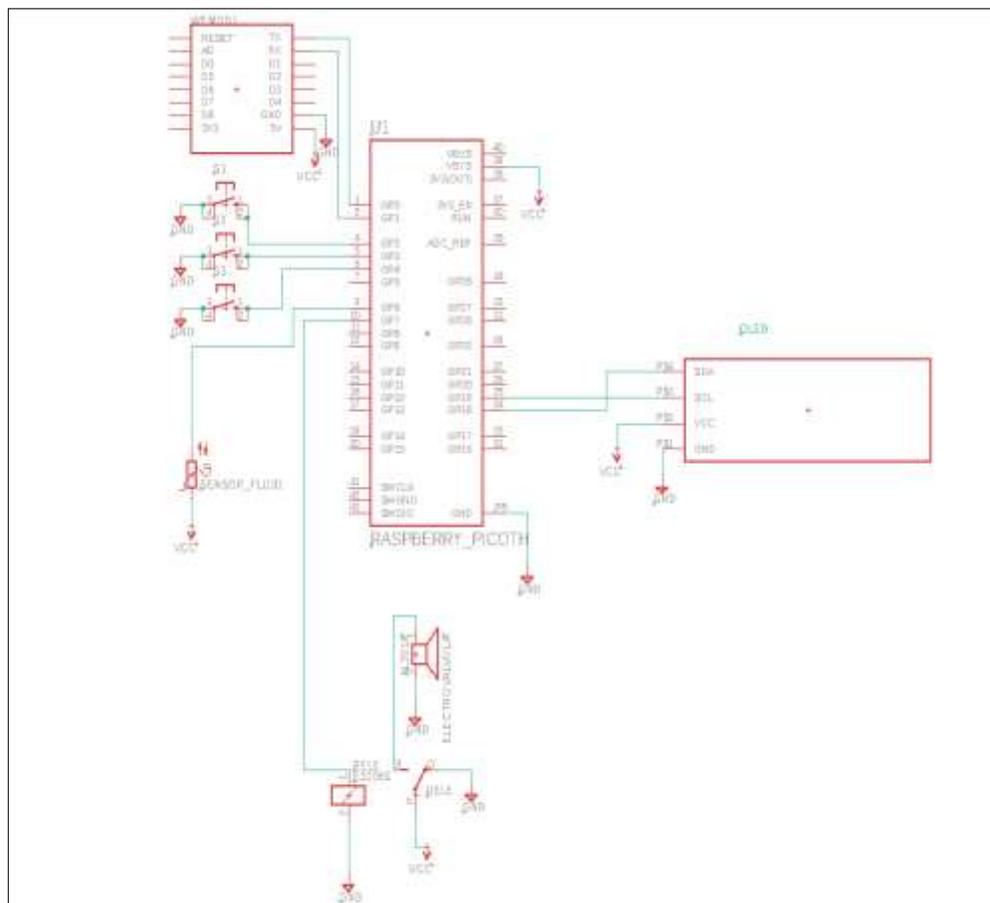
4.1.2. Diseño de un sistema electrónico

Se diseñaron distintos esquemas para poder apreciar las conexiones necesarias y utilizadas en el diseño propuesto. Se utilizó un relé convencional en el diseño, ya que no afecta al diseño final, debido a que lo único que se requiere es la activación de este. De igual manera, se utilizaron *buzzers* en el diseño en lugar de electroválvulas, ya que solo se quiere ejemplificar la conexión de estas, y que serán tomadas del relé y su activación dependerá de las placas

programables utilizadas, en este caso de la Raspberry Pi Pico y de la ESP8266 WeMos mini D1.

Figura 35.

Conexiones de la Raspberry Pi Pico y ESP8266 del área A



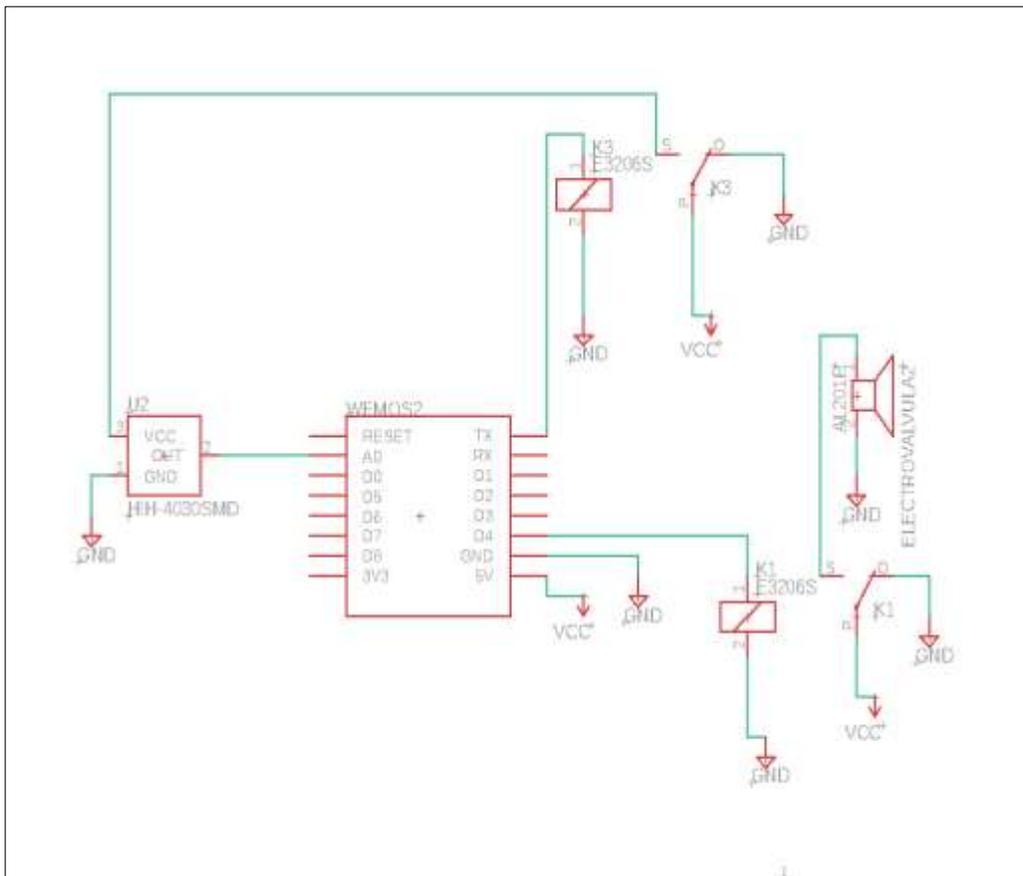
Nota. Conexiones de la Raspberry Pi Pico y ESP8266. Elaboración propia, realizado con Eagle.

Al igual que el esquema anterior, se utilizó un *buzzer* para representar la electroválvula y un relé simple para representar al módulo del relé. Este tiene la particularidad de que el sensor de humedad se representa con un sensor de humedad de ambiente, pero representa al sensor de humedad de suelo. Este se

puede observar que se activa con el relé para no estar constantemente en medición, como se había detallado anteriormente.

Figura 36.

Conexiones de la ESP8266 en el área B



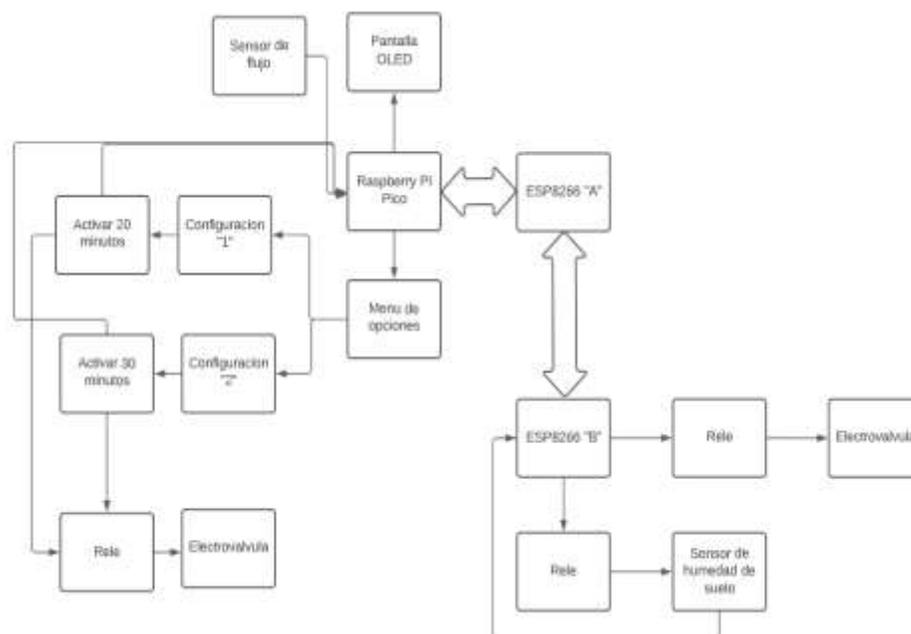
Nota. Conexiones de ESP8266 en el área B. Elaboración propia, realizado con Eagle.

4.1.3. Diagrama de relación

Se realizó un diagrama de relación de componentes para comprender cómo se relacionan los componentes entre sí de mejor manera.

Figura 37.

Diagrama de relación del sistema



Nota. Se ilustra el diagrama de relación del sistema. Elaboración propia, realizado con Lucidchart.

4.2. Código de programación

A continuación se detalla la programación realizada para la Raspberry Pi Pico, sus puertos de entrada y salida, la pantalla OLED y la comunicación con la ESP8266 del área "A".

Figura 38.

Variables de la Raspberry Pi Pico

```
1 from machine import UART, Pin, I2C
2 from ssd1306 import SSD1306_I2C
3 import time
4 import utime
5 import framebuf
6
7 #----- variables para el tamaño de la pantalla OLED
8 WIDTH = 128
9 HEIGHT = 64
10
11 confi = False #variable para saber si estamos dentro de configuraciones
12
13 #-----variables para control de tiempo
14 mod1 = False
15 mod2 = False
16
17 #para conexión UART con la ESP8266
18 baudrate=9600
19 uart0 = UART(0, baudrate, tx=Pin(0), rx=Pin(1))
20
21 #-----PARA LA PANTALLA I2C-----#####
22 i2c = I2C(1, scl=Pin(19), sda=Pin(18)) #pines para comunicación I2C
23
24 oled = SSD1306_I2C(WIDTH, HEIGHT, i2c) #para llamar a la función de la i2c
25
26
```

Nota. Variables de la Raspberry Pi Pico. Elaboración propia, realizado con Thonny.

Figura 39.

Pines de entrada y salida

```
28 #-----CONFIGURACION DE PINES DE ENTRADA-----
29 configuracion = Pin(2, Pin.IN, Pin.PULL_UP) #boton para entrar en configuracioens
30 modo1 = Pin(3, Pin.IN, Pin.PULL_UP) #boton para seleccionar el modo 1
31 modo2 = Pin(4, Pin.IN, Pin.PULL_UP) #boton para seleccionar el modo 2
32
33 sensor_flujo = Pin(6, Pin.IN, Pin.PULL_UP) # entrada para el sensor de flujo
34
35
36 #-----CONFIGURACION PINES DE SALIDA
37 electroValvula = Pin(7, Pin.OUT) # salida para electrovalvula
38
39
40 electroValvula.off() #apagamos la electrovalvula por si estaba encendida
41
42 print("INICIANDO SISTEMA \n Estableciendo conexión...")
43
44
```

Nota. Pines de entrada y salida. Elaboración propia, realizado con Thonny.

Figura 40.

Texto en la pantalla OLED

```
46 #mostrando para conexion
47 oled.fill(0) #con esto borramos toda la pantalla
48 time.sleep(1)
49 oled.text("INICIANDO", 0,0) #escribimos texto en x,y
50 oled.text('...', 0, 16)
51 oled.text('Estableciendo', 0, 32)
52 oled.text('conexion', 0, 48)
53 oled.show() #mostramos lo que escribimos
54 time.sleep(2)
```

Nota. Se establece el texto a mostrar en el transcurso de la conexión. Elaboración propia, realizado con Thonny.

Figura 41.

Lectura de datos recibidos de la ESP8266

```
#funcion para peticion de informacion de la UART
def lectura():

    global valor_humedad = ''

    if uart0.any() > 0:
        rxData = uart0.read()#leemos lo que envian por uart0
        convertido=rxData.decode('utf-8') #convertimos el dato a string
        print("dato recibido dentro de funcion lectura:\n" + convertido)
        valor_humedad = convertido

    if len(rxData) > 0:
        return valor_humedad
    else:
        print("no hay ningun dato para leer")
        return "0"
```

Nota. Se realiza la función de petición de datos de la UART. Elaboración propia, realizado con Thonny.

Figura 42.

Comunicación con la ESP8266

```
#-----funcion para comunicarnos con la ESP-----#
def escritura(infoSensor):
    uart0.write(infoSensor) #enviamos dato por uart0
    uart0.write(b"\r\n")
    print("FIN ESCRITURA: ")

print("Conexión establecida...")
oled.fill(0) #con esto borraremos toda la pantalla
time.sleep(1)
oled.text("SISTEMA DE RIEGO", 0,0) #escribimos texto en x,y
oled.text('FUNCIONANDO', 0, 16)
oled.text('...', 0, 32)
oled.text(valor_humedad, 0, 48)
oled.show() #mostramos lo que escribimos
time.sleep(2)
```

Nota. Comunicación con la ESP8266. Elaboración propia, realizado con Thonny.

Se establecen las funciones para el control del flujo del agua. Se realizó una función para determinar los movimientos en el sensor de flujo y, a partir de ahí, obtener la frecuencia, y esta a su vez es útil para el cálculo de la cantidad de volumen.

Figura 43.

Declaración de funciones para control de flujo I

```
#-----FUNCIONES PARA CONTROL DEL SENSOR DE FLUJO-----#
#funcion de interrupcion para el sensor de flujo
def encoder_handler(pin):
    global paso
    global confirmar
    try:
        if confirmar == "ok":
            paso += 1
        else:
            paso = 0
        #print(paso)
    except:
        None
```

Nota. Declaración de funciones para control de flujo I. Elaboración propia, realizado con Thonny.

Figura 44.

Declaración de funciones para control de flujo II

```
#funcion para calcular la frecuencia
def obtener_frecuencia():

    global paso
    global timer_start
    global frec_hz #variable solo para saber si está pasando flujo por el sensor

    timer_elapsed = utime.ticks_diff(utime.ticks_ms(), timer_start)
    if timer_elapsed >= 1000:
        state = disable_irq() #deshabilitamos las interrupciones
        Hz = paso #frecuencia de rotacion de las aspas
        paso = 0 #reiniciamos el contador de frecuencia
        enable_irq(state) #habilitamos las interrupciones nuevamente
        timer_start = utime.ticks_ms()
        return Hz
    else:
        return 0
```

Nota. Declaración de funciones para control de flujo II. Elaboración propia, realizado con Thonny.

Figura 45.

Declaración de funciones para control de flujo III

```
def cant_volumen(frecuencia):

    global t0
    global conteo_en_vol #variable para tener un conteo de cuantas veces se recurrió para sumar volumen

    if frecuencia != 0:

        galones_min = frecuencia/divisor # calculamos los galones por minuto
        gal_seg = galones_min/60 #calculamos los galones por segundo
        dt = utime.ticks_diff(utime.ticks_ms(), t0) #calculamos el tiempo que transcurrió, delta tiempo
        dt_seg = dt/1000 #paso de tiempo a segundos
        t0 = utime.ticks_ms() #reestablecemos el valor de la variable de conteo inicial
        print("galones_min: ", galones_min)
        print("gal_seg: ", gal_seg)
        print("tiempo pasado: ", dt_seg, " seg")
        volumen = volumen+gal_seg*(dt_seg) #calculamos el volumen en función del tiempo que ha pasado

    return volumen

#-----
#interrupcion del sensor de flujo
sensor_flujo.irq(trigger=Pin.IRQ_FALLING, handler=encoder_handler )

t_inicial = 0 #variable para control del tiempo
```

Nota. Declaración de funciones para control de flujo III. Elaboración propia, realizado con Thonny.

Figura 46.

Tiempos de apertura I

```
while True:
    if configuracion.value() == 0:
        electroValvula.off() #apagamos la electrovalvula por el estado encendido
        print("botón de configuración presionado")
        print("entrando a configuraciones\n")
        oled.fill(0) #con esto borramos toda la pantalla
        time.sleep(1)
        oled.text("    MODO 1", 0,0) #escribimos texto en x,y
        oled.text("20 Min.", 0, 16) #escribimos texto en x,y
        oled.text("-----", 0, 28) #escribimos texto en x,y
        oled.text("    MODO 2", 0,36) #escribimos texto en x,y
        oled.text("30 Min.", 0, 52) #escribimos texto en x,y
        oled.show() #mostramos lo que escribimos
        time.sleep(2)

        confi = True #establecemos que estamos dentro del apartado de configuraciones
        #vamos a configuraciones

    if modo1.value() == 0 and confi == True: #preguntamos si estamos dentro de configuraciones y si se presiona el botón
        electroValvula.on() #encendemos la electrovalvula por 20 min
        print("botón para seleccion modo 1 presionado")
        print("modo 1 seleccionado")
        print("abriendo electroválvula")
        print("iniciando riego, 20 min.\n")
        oled.fill(0) #con esto borramos toda la pantalla
        time.sleep(1)
        oled.text("SISTEMA DE RIEGO", 0,0) #escribimos texto en x,y
        oled.text('FUNCIONANDO', 0, 16)
        oled.text('MODO 1, OK', 0, 32)
        oled.text(valor_humedad, 0, 48)
        oled.show() #mostramos lo que escribimos
        time.sleep(2)
        confi = False #reiniciamos la variable confi a su estado inicial
        modi = True #variable para saber que se selecciono el modo 1
        mod2 = False #reiniciamos el valor
        t_inicial = utime_ticks_ms() #iniciamos variable para control del tiempo de apertura de electroválvula...
```

Nota. Se calculan los tiempos de apertura I. Elaboración propia, realizado con Thonny.

Figura 47.

Tiempos de apertura II

```
if modo2.value() == 0 and confi == True:
    electroValvula.on() #encendemos la electrovalvula por 30 min
    print("botón para seleccion modo 2 presionado")
    print("modo 2 seleccionado")
    print("abriendo electroválvula")
    print("iniciando riego, 30 min.\n")
    oled.fill(0) #con esto borramos toda la pantalla
    time.sleep(1)
    oled.text("SISTEMA DE RIEGO", 0,0) #escribimos texto en x,y
    oled.text('FUNCIONANDO', 0, 16)
    oled.text('MODO 2, OK', 0, 32)
    oled.text(valor_humedad, 0, 48)
    oled.show() #mostramos lo que escribimos
    time.sleep(2)
    t_inicial = utime_ticks_ms() #Iniciamos variable para control del tiempo de apertura de electroválvula
    mod2 = True #variable para saber que se selecciono el modo 2
    mod1 = False #reiniciamos el valor
    confi = False #reiniciamos la variable confi a su estado inicial
```

Nota. Se calculan los tiempos de apertura II. Elaboración propia, realizado con Thonny.

Figura 48.

Tiempos de apertura III

```
228     oled.text('HUMEDAD 50%', 0, 48)
229     oled.show()
230     time.sleep(1)
231
232
233     #tiempo por el selecciono modo 2, 30 min = 1800 segundos
234     if(tiempo_transcurrido >= 1800): #transcurrieron los 30 min de riego
235         if(mod2 == True):
236             electroValvula.off() #apagamos la electrovalvula
237             mod1 = False # reiniciamos valores para volver a iniciar con otro riego
238             mod2 = False
239             #pantalla inicio donde no hay ningún modo activo
240             oled.fill(0)
241             time.sleep(1)
242             oled.text("SISTEMA DE RIEGO", 0,0)
243             oled.text('FUNCIONANDO', 0, 10)
244             oled.text('...', 0, 32)
245             oled.text('HUMEDAD 50%', 0, 48)
246             oled.show()
247             time.sleep(1)
248
249     if configuracion.value() == 0: #analizo del contador de tiempo y podemos ir a configuraciones nuevamente
250         electroValvula.off() #apagamos la electrovalvula
251         mod1 = False
252         mod2 = False
253
```

Nota. Se calculan los tiempos de apertura III. Elaboración propia, realizado con Thonny.

Se deben importar las librerías necesarias, al igual que con la Raspberry Pi Pico, y se establecen las variables a utilizar para la ESP8266 del área “A”.

Figura 49.

Librerías y puertos

```
import network
from machine import Timer, UART, Pin
import usocket
import time

#-----variables para guardar la info del otro ESP8266
sensor_humedad = 0

#para conexion UART con la ESP8266
baudrate=9600
uart0 = UART(0, baudrate, tx=Pin(1), rx=Pin(3))
```

Nota. Se abordan las librerías y puertos. Elaboración propia, realizado con Thonny.

Figura 50.

Funciones de lectura y escritura

```
#funcion para peticion de informacion de la UART y enviarlo a la pico
def lectura():
    if uart0.any() > 0:
        rxData = uart0.read()#leemos lo que envian por uart0
        convertido=rxData.decode('utf-8') #convertimos el dato a string
        print("dato recibido dentro de funcion lectura:\n" + convertido)

    if len(rxData) > 0:
        return convertido
    else:
        print("no hay ningun dato para leer")
        return "0"

#-----funcion para comunicarnos con la Raspberry Pi PICO-----#
def escritura(infoSensor):
    uart0.write(infoSensor) #enviamos dato por uart0
    uart0.write(b"\r\n")
    print("FIN ESCRITURA: ")
```

Nota. Se crea una funcion de lectura y escritura para la comunicaci3n con la Raspberry Pi Pico. Elaboraci3n propia, realizado con Thonny.

Por 3ltimo, se crea la red wifi y se crea el *socket* de comunicaci3n entre los dos m3dulos ESP-8266.

Figura 51.

Comunicación inalámbrica

```
#creamos la red wifi
ap = network.WLAN(network.AP_IF) # create access-point interface
ap.active(True) # activate the interface
ap.config(ssid='ESP8266', channel = 11, authmode = 3, password = '12345678') # set the SSID of the access point

#creamos un socket para poder intercambiar información entre tarjetas ESP8266
s = usocket.socket()
s.bind(("192.168.0.9",80)) #dirección de socket para conectarse en el otro dispositivo, y el puerto
s.listen(5) #numero de dispositivos que se pueden conectar
print("servidor iniciado, esperando conexión")
continuar1 = True

while continuar1:
    (sc,addr) = s.accept() #accedemos a la info cuando envían un dato
    print(addr) #imprimimos la dirección del que se conecta
    continuar2 = True
    while continuar2:
        mensaje = sc.recv(64) #mensaje con la info del cliente, otro ESP8266
        if not mensaje:
            continuar2 = False
        elif mensaje.decode() == "fin": #para cancelar y cerrar el socket, esto es para pruebas
            continuar1 = False
            continuar2 = False
            print(mensaje.decode()) #imprimimos el mensaje
        else:
            sensor_humedad = mensaje.decode() #guardamos el valor del sensor de humedad recibido por el otro ESP8266
            escritura(sensor_humedad) #enviamos la info hacia la PICO
            #esperamos un momento para ver si enviamos la info de activar la electroválvula o no
            time.sleep(2)
            dato = lectura() #leemos lo que responde la pico y lo guardamos en dato
            if( dato == "activar"): #si la pico responde activar enviamos la info
                s.send("activar".encode()) #enviamos por el socket al ESP8266 de abajo que active la electroválvula

            print(mensaje.decode()) #imprimimos el mensaje recibido de la esp con el valor de la humedad del suelo
            sc.close() #cerramos el socket
    s.close() #cerramos el socket
print("finalización de programa")
```

Nota. Se aborda la comunicación inalámbrica. Elaboración propia, realizado con Thonny.

Finalmente se configura la ESP8266 del area "B".

Figura 52.

Activación de pines y librerías por utilizar

```
#programa que se conecta a la red creada por el otro ESP y envía la información del sensor de humedad,
#también recibe datos de si abre o cierra la electroválvula

from machine import Pin, ADC

import network
import socket
import time

#----configuración de pines

actHumedad = Pin(1, Pin.OUT)# activar el relay que alimenta el sensor de humedad
actHumedad = off() #desactivamos el relay

#pin para leer el sensor de humedad
sensHumedad = ADC(0)# unico ADC de la placa con resolución de 10bits

electroValvula = Pin(2, Pin.OUT) # salida para electrovalvula
electroValvula.off() #apagamos la electrovalvula por si estaba encendida

#-----credenciales para conectarme a la red wifi creada por la otra ESP8266
ssid = 'ESP8266'
password = '12345678'

station = network.WLAN(network.STA_IF)
station.active(True)
station.connect(ssid, password)

while station.isconnected() == False:
    pass

print('Dispositivo conectado a la red wifi')
print(station.ifconfig())
```

Nota. Se abordan los pines y librerías a utilizar. Elaboración propia, realizado con Thonny.

Figura 53.

Conexión con socket existente

```
#-----nos conectamos al socket creado por el otro ESP
s = socket.socket()
s = connect(("192.168.0.9",80))
print("dispositivo conectado al otro ESP, listo para la transmision de información")

continuar = True
```

Nota. Se aborda la conexión con el socket existente. Elaboración propia, realizado con Thonny.

Figura 54.

Condición de continuación y valor de humedad

```
while continuar:
    actHumedad.off()
    time.sleep(1200) #esperamos 20 min para activar el relay y leer la info del sensor de humedad
    actHumedad.on() #activamos el relay para alimentar al sensor de humedad
    time.sleep(2)

    humedad = sensHumedad.read()
    print(humedad) #imprimimos el valor del sensor de humedad
    # aplicamos un factor de conversión para obtenerlo en voltaje
    volts = humedad * (3.3/1023) #voltaje de la placa /resolución del ADC
    print(volts)

    #1023 -> no hay nada de humedad/seco
    #<100 hay exceso de humedad

    if(humedad <= 400): #se presenta demasiada humedad
        valor_humedad = "demasiado humedo"
        #enviamos la info a la otra ESP
        s.send(valor_humedad.encode()) #necesitamos codificarlo a binario para enviarlo, ya que no se puede enviar como string
    elif(humeda >= 600 and humeda <= 800):
        valor_humedad = "medio seco"
        s.send(valor_humedad.encode())
    else:
        valor_humedad = "seco"
        s.send(valor_humedad.encode())

    time.sleep(2) #esperamos respuesta del otro ESP para ver si activamos o no la electrovalvula
    (sc,addr) = s.accept() #accedemos a la info cuando envian un dato
    mensaje = sc.recv(64) #mensaje recibido en tamaño corto
    if(mensaje.decode() == "activar"):
        electroValvula.on() #activamos la electrovalvula
    elif(mensaje.decode() == "desconectar"):
        s.close() #cerramos el canal de escuche del socket... sirve para pruebas
    else:
        electroValvula.off()

s.close()
print("fin de la conexión")
```

Nota. Condición de continuación y valor de humedad. Elaboración propia, realizado con Thonny.

5. MANUAL DE USUARIO

5.1. Descripción

Este sistema es un riego automático por tiempos establecidos que se puede activar las veces deseadas en un día, simplemente seleccionando la modalidad deseada. Este sistema consta de dos áreas “A” y “B”, respectivamente, donde el área “A” es el área de visualización de la pantalla de selección de configuración y el control maestro.

El sistema consta de una pantalla que ayuda a visualizar si el sistema funciona correctamente y en qué tipo de modalidad se encuentra.

5.2. Funcionamiento

La instalación del módulo principal se puede acomodar dentro de la caja IP66, donde deben ir los siguientes componentes: 1. Raspberry Pi Pico. 2. ESP8266. 3. 1 pantalla OLED. 4. 1 módulo de relé, y 5. 3 botones de selección.

Se procede a colocarlo en un área cercana a un tomacorriente y a unos 2 metros de una tubería cercana. Luego se instala el sensor de flujo en la tubería cercana al punto “A”, se coloca la electroválvula en donde se desea iniciar la salida hacia el cultivo. Una vez termine el punto “A”, se procede a instalar el punto “B”. En el punto “B” viene un sensor de humedad, este se debe colocar lejos del área de cultivo si se desea saber qué tan húmeda está la tierra en general, o cerca del cultivo para controlar la humedad de los cultivos. Una vez instalados

ambos puntos, se procede a realizar el inicio de las modalidades y se regresa al punto “A”.

Se conecta el sistema del área “A” primeramente y se espera hasta que la pantalla indique que se encuentra estableciendo conexión. Una vez esta se encuentre estableciendo conexión, se procede a conectar el módulo del área “B”. Es posible observar cómo la pantalla en el área “A” indica que hay conexión exitosa. Hay 3 botones junto al dispositivo. Se presiona el botón superior para acceder al menú de opciones y posteriormente, si se desea iniciar el funcionamiento en la modalidad 1 (20 minutos), se presiona el segundo botón, si por el contrario se desea la modalidad 2 (30 minutos) se presiona el tercer botón. De esta manera el sistema iniciará un riego automático el tiempo deseado.

En la pantalla también se podrá observar un texto abajo del texto principal que indica la humedad del sensor donde se colocó. No es necesario colocarlo cerca de una planta, ya que indica si hay humedad en general, pero puede hacerlo.

Consideraciones:

- Evitar tocar un componente fuera de lo indicado.
- No permitir que los componentes entren en contacto con líquidos o pueden ocurrir daños eléctricos, a excepción del sensor de humedad, que puede soportar la humedad.
- Conectar los módulos como fue propuesto o se puede dañar un componente.
- Evitar dañar el cableado para incrementar el tiempo de vida y evitar fallos repentinos.

6. MANUAL DE MANTENIMIENTO

6.1. Mantenimiento preventivo

Realizar las siguientes acciones para preservar de mejor manera el funcionamiento del sistema y prolongar su tiempo de vida útil.

- Limpieza regular: es esencial para el mantenimiento preventivo de los dispositivos electrónicos. Se utiliza un paño suave y seco para limpiar la superficie de la caja IP66. Se evita el uso de limpiadores químicos fuertes, ya que pueden dañar la superficie.
- Evitar el polvo: los dispositivos electrónicos son sensibles al polvo y la suciedad, por lo que es importante mantenerlos alejados de cualquier lugar donde se pueda acumular polvo. No se debe dejar abierta la tapa de la caja IP66 que cubre los dispositivos y se debe limpiar las áreas cercanas donde están instalados.
- Revisar los cables y enchufes: los cables y enchufes pueden desgastarse con el tiempo o aflojarse por vibraciones o movimientos externos. Es útil revisar los cables y enchufes de los dispositivos electrónicos para asegurarse de que estén en buen estado. Si se observan signos de desgaste, se reemplazan los cables de inmediato.
- Revisar si hay fugas: se realiza una revisión ocasional para verificar la integridad de la caja protectora, para observar si existen fugas y, si es así,

se reemplaza la base o se tapa la fuga con un material que no sea afectado por el agua, también se evita dejar los dispositivos expuestos al sol.

- Protege los dispositivos de golpes y caídas: los dispositivos electrónicos son frágiles y pueden dañarse fácilmente si se caen o se golpean. Si se realizará un movimiento de los componentes por cualquier motivo, se debe procurar llevarlos de una manera segura y cómoda para que no se resbalen durante el traslado y sufran daños en la estructura y los circuitos internos.
- Revisar regularmente las tuberías: es importante observar el suelo entre tramos de tubería para verificar si existe una fuga en un lugar no deseado. También es importante observar las uniones o separaciones de tuberías para corroborar si no hay fugas en ellas.

Tabla 1.

Propuesta de documentación del mantenimiento preventivo

Área	Equipo	Acción	Periodicidad	Realizado
1	Caja de IP66	Limpieza	Semanal	Sí
2	Sensor de humedad	Observar si hay oxidación	Mensual	Sí

Nota. Propuesta de documentación del mantenimiento preventivo. Elaboración propia, realizado con Microsoft Word.

6.2. Mantenimiento correctivo

Si existe un problema en el funcionamiento del sistema, se deben realizar los siguientes pasos:

- Identificar el problema: el primer paso para el mantenimiento correctivo es identificar el problema. Si un dispositivo electrónico no está funcionando correctamente, es importante determinar qué está causando el problema. Es útil asegurarse de examinar cuidadosamente el dispositivo y buscar cualquier signo de daño o mal funcionamiento.
- Observar detenidamente las uniones: antes de realizar una reparación de una tubería, es necesario verificar repuestos para esta y no dejar inhabilitado el sistema. Además, analizar previamente cómo se realizará el cambio y no mover de manera errónea una pieza.
- Realizar pruebas: antes de realizar cualquier reparación, debe haber seguridad de realizar pruebas para confirmar el problema. Se debe reiniciar los dispositivos, simplemente desconectando de la fuente de poder los mismos.
- Realizar la reparación: una vez se ha identificado la causa del problema, es hora de realizar la reparación. Si no hay seguridad de cómo realizar la reparación, es necesario buscar ayuda de un experto. Hay que asegurarse de tener las herramientas adecuadas antes de comenzar la reparación.
- Documentar el proceso: es importante documentar el proceso de mantenimiento correctivo. Esto incluye la identificación del problema, la investigación, las pruebas, la reparación y las pruebas adicionales. La documentación puede ayudar a solucionar problemas similares en el futuro y puede ser útil para otros usuarios.

Tabla 2.

Propuesta de documentación del mantenimiento correctivo

Área	Maquina/Equipo	Acción	Numero de mantenimiento	Resultado
1	Esp8266	Cambio de cables	1	Funcionando correctamente

Nota. Propuesta de documentación del mantenimiento correctivo. Elaboración propia, realizado con Microsoft Word.

Debe recordarse la importancia de tener cuidado al realizar reparaciones y buscar ayuda de un experto si no hay seguridad de cómo realizar una reparación.

7. ANÁLISIS FINANCIERO

7.1. Análisis de costos

A continuación se pueden observar tablas con los costos totales del sistema propuesto. Los costos de tubería se estimaron en que la separación entre árboles es de 6 metros y hay una distancia de 900 metros totales.

$$\text{Arboles a regar} = \frac{900m}{6m} = 150 \text{ arboles a regar}$$

$$\text{Cantidad de goteros} = \text{Arboles a regar}$$

Tabla 3.

Costos del materiales eléctrico y electrónico

No.	Material y equipo	Cantidad	Costo individual (Q)	Total (Q)
Equipo electrónico				
1	Raspberry Pi Pico	1	100.00	100.00
2	ESP8266 WeMos D1 mini	2	50.00	100.00
3	Pantalla OLED0.96 pulgadas	1	70.00	70.00
4	Electroválvula	2	120.00	240.00
5	Módulo relé	3	25.00	75.00
6	Sensor de humedad YL-69	1	25.00	25.00
7	Botones pulsadores	3	2.00	6.00
8	Sensor de flujo	1	130.00	130.00
9	Cableado en circuitos (m)	5	3.00	15.00
10	Cableado para exteriores (m)	15	5.00	75.00
			Total (Q)	836.00

Nota. Costos de materiales eléctrico y electrónico. Elaboración propia, realizado con Microsoft Word.

Tabla 4.*Costos de materiales de riego de tubería*

No.	Material y equipo	Cantidad	Costo individual (Q)	Total (Q)
Sistema de riego				
1	Manguera 305m de 16mm diámetro	3	894.00	2682.00
2	Goteros	150	3.50	525.00
3	Salidas tipo T	15	1.50	22.50
4	Válvulas manuales	15	7.00	105.00
5	Reductor de flujo	2	150.00	300.00
6	Tubería adicional previa	1	100.00	100.00
			Total (Q)	3,734.50

Nota. Costos de materiales de riego de tubería. Elaboración propia, realizado con Microsoft Word.

Tabla 5.*Costos totales*

Costos de equipo eléctrico y electrónico	8,36.00
Costos de materiales de riego y tubería	3,734.5
TOTAL (Q)	4,570.5

Nota. Costos totales del proyecto. Elaboración propia, realizado con Microsoft Word.

Estos costos pueden variar, ya que los precios de los dispositivos y componentes pueden cambiar dependiendo la localización de donde se consiguen o el alza de precios.

CONCLUSIONES

1. El diseño del sistema de riego por goteo automatizado, utilizando una placa programable Raspberry Pi Pico y módulo ESP8266, en la finca Moralfa, departamento de Quetzaltenango, municipio de Coatepeque, se completó satisfactoriamente.
2. Se determinan los componentes electrónicos a utilizar en el diseño del sistema de riego, de acuerdo con las dimensiones de la finca.
3. Se define el tipo de tubería a utilizar, de acuerdo con el sistema de riego y los costos asociados a este.
4. Se diseña un esquema del funcionamiento del sistema propuesto.
5. Se desarrolla el código fuente en el lenguaje de programación MicroPython.
6. Se desarrollan los manuales de usuario y mantenimiento del diseño propuesto.

RECOMENDACIONES

1. Utilizar un módulo de desarrollo que implemente Wi-Fi como placa de desarrollo principal y, de esta manera, controlar todos los periféricos cercanos desde la misma placa de desarrollo.
2. Incrementar la cantidad de dispositivos secundarios de recepción para tener mayor alcance y así incrementar el área de trabajo.
3. Obtener datos previos del comportamiento de la lluvia en el área, dependiendo de la estación, para incrementar el nivel de automatización y seguir mejorando el sistema.
4. Actualizar las placas programables cada 3 años, ya que la tecnología está en constante evolución, para así mejorar el sistema propuesto.

REFERENCIAS

- Agropinos. (febrero de 2023). *Cómo prevenir problemas en los sistemas de riego*.
<https://www.agropinos.com/blog/como-prevenir-problemas-en-los-sistemas-de-riego>.
- American Society of Mechanical Engineers. (2020). *Norma ASME 31.9-2020*. ASME.
- Arrow. (diciembre de 2015). *Sensores digitales: el camino al futuro*.
<https://www.arrow.com/es-mx/research-and-events/articles/te-digital-sensors-the-path-forward>.
- Cita-Sa. (enero de 2023). *Sistemas de riego*. <https://citi-sa.com/los-sistemas-de-riego-mas-recomendados-para-cada-tipo-de-cultivo/>.
- Club del Aguacate. (22 de septiembre de 2022). *El aguacate y sus variedades*. Aguacate Club. <https://aguacate.club/blog/aguacate-variedades/>.
- Equipo editorial. (noviembre de 2015). Principales actuadores en el vehículo y sus posibles fallos. *Revista Autocrash* 25(2), 2-14.
<https://www.revistaautocrash.com/electromecanica-principales-actuadores-en-el-vehiculo-y-sus-posibles-fallos/>.
- Finca Moralfa. (2023). *Información sobre la institución*. Moralfa.

Google Earth. (2023). *Fotografía panorámica (aérea) de la finca Moralfa.*

<https://earth.google.com/web/@14.8328868,-91.5134779,2325.14336969a,153.02944921d,35y,0h,45t,0r/data=CIEaTxJJCiUweDg1OGU5N2ZIM2ZjODIkM2Y6MHgzYmE3ZmYwMTFmMGYwMDBmGdLFppVCtC1AIXegTnn04VbAKg5RdWV0emFsdGVuYW5nbxgCIAE>

Hidrocucuta (marzo de 2018). *Tipos de válvulas, clasificación y funciones.*

<https://hidrocucuta.com/tipos-de-valvulas-clasificacion-y-sus-funciones/>.

Junta de Andalucía. (diciembre de 2022). *Agricultura: tipos de aguacate y cultivo.*

<https://agbaragriculture.com/tipos-de-aguacate-diferencias-cultivo-comercializacion/>.

Mecafenix. (octubre de 2022). *Información sobre sensores.*

<https://www.ingmecafenix.com/automatizacion/sensores/guia-sensores/>.

Mischiante. (junio de 2021). *Pines ESP8266 We Mos mini D1.*

<https://www.mischianti.org/2021/04/26/wemos-d1-mini-high-resolution-pinout-and-specs/>

Municipalidad de Colomba. (febrero de 2023). *Mapa de Colomba Costa Cuca.*

<https://259513-ENJXRBQTQK.pdf> (segeplan.gob.gt)

Novagric. (octubre de 2022). *Instalaciones de riego.*

<https://www.novagric.com/es/riego/servicios/instalaciones-de-riego>.

Placas de Desarrollo. (noviembre de 2022). *Información sobre placas de desarrollo.* <https://www.placasdedesarrollo.com/>.

Prakor. (febrero de 2019). *Componentes de un sistema de riego localizado*.
<https://prakor.com/componentes-de-un-sistema-de-riego-localizado/>.

Pressman, R. y Maxim B. (2020). *Software engineering: a practitioner's approach*.
McGraw-Hill.

Raspberry Pi. (agosto de 2022). *Productos Raspberry*.
<https://www.raspberrypi.com/products/raspberry-pi-400/>.

Raspberry Pi. (febrero de 2023). *Raspberry Pi Pico datasheet*.
<https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf>.

Redactor. (septiembre de 2017). Medidas y conceptos básicos sobre riego y drenaje. *Irrinews* 1(1), 1-5. <https://irrinews.com/2017/09/05/medidas-y-conceptos-basicos-sobre-riego-y-drenaje/>.

Rudeus, G. (5 de julio de 2022). *Cómo elegir un lenguaje de programación para un proyecto*. Barcelona Geeks. <https://barcelonageeks.com/como-elegir-un-lenguaje-de-programacion-para-un-proyecto/>.

SAS. (5 de diciembre de 2022). *Qué es un sistema de riego y tipos de sistemas de riego*. Fibras y Normas de Colombia. <https://blog.fibrasynormasdecolombia.com/que-es-un-sistema-de-riego-tipos-de-sistemas-de-riego/>.

Sigfrido Fruit. (agosto de 2021). *Variedades de aguacate*.
<https://sigfridofruit.com/cuales-son-las-variedades-del-aguacate/>.

Soto, S. (noviembre de 2019). *Redes inalámbricas*. Rom Mayer. <https://rom-mayer.cl/redes-inalambricas-2/>.

Tic Negocios. (enero de 2023). *Caminar con éxito hacia la industria*. <https://ticnegocios.camaravalencia.com/servicios/tendencias/caminar-con-exito-hacia-la-industria-4-0-capitulo-11-infraestructuras-i-redes-inalambricas/>.

Tinkercad. (febrero de 2023). *Información sobre Tinkercad*. <https://www.tinkercad.com/>.

Viveros Brokaw. (octubre de 2022). *Tipo de flor en cultivo de aguacate*. <https://www.viverosbrokaw.com/tipo-de-flor-en-cultivo-de-aguacate/>.

Viveros Grajera. (junio de 2019). *Clases de aguacates. Variedad Bacon. Características*. <https://viverosgrajera.com/clases-de-aguacates-variedad-bacon/>.

APÉNDICES

Apéndice 1.

Plantas de aguacate en la finca



Nota. Fotografía de la planta de aguacate en el lugar de estudio. Elaboración propia.

Apéndice 2.

Disposición de la plantación de aguacate en la finca



Nota. Fotografía de la disposición de la plantación de aguacate en la finca. Elaboración propia.

Apéndice 3.

Visita de campo

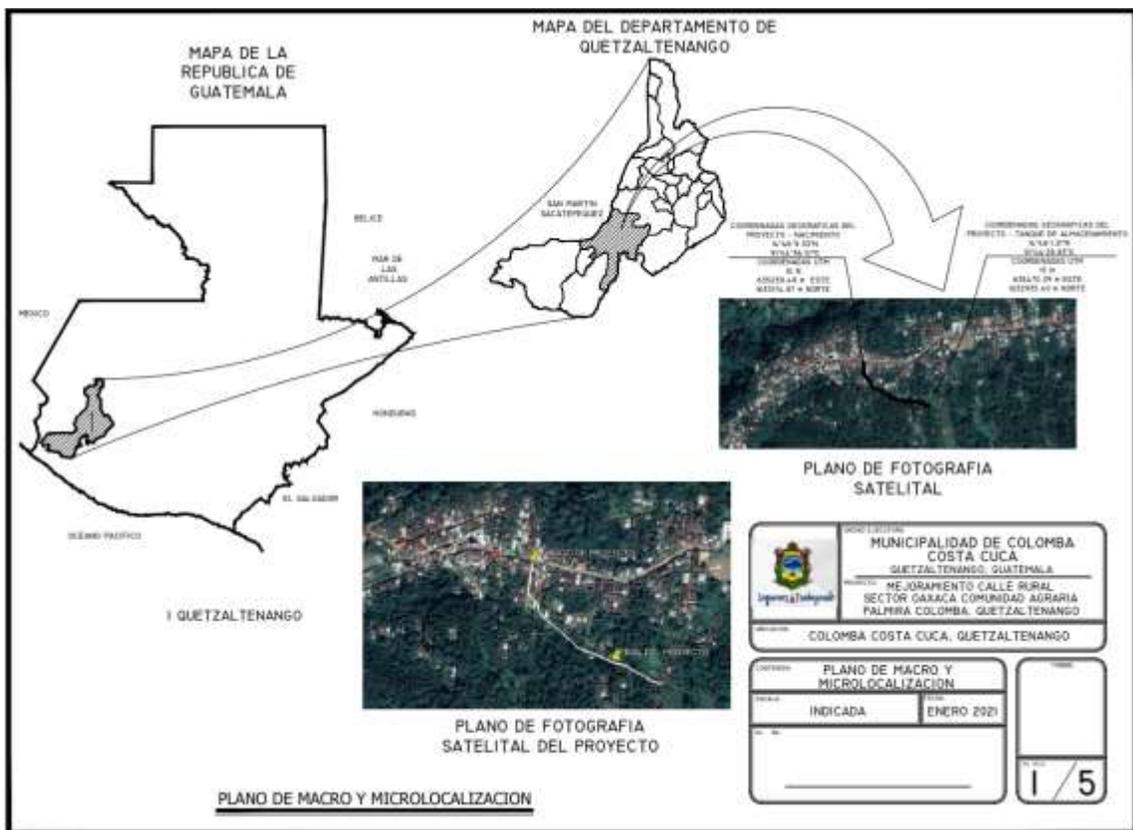


Nota. Fotografía de la visita de campo. Elaboración propia.

ANEXOS

Anexo 1.

Mapa de Guatemala, Quetzaltenango, Colomba Costa Cuca



Nota. Mapa de la región donde se ubica la finca Moralfa. Obtenido de Municipalidad de Colomba (2023). *Mapa de Colomba Costa Cuca* [mapa]. ([259513-ENJXRBTQK.pdf](#) ([segeplan.gob.gt](#)), consultado el 20 de febrero de 2023. De dominio público.

Anexo 2.

Límites de temperatura para tubería termoplástica

ASME B31.9-2020

Table I-2 Hydrostatic Design Stresses (HDS) and Recommended Temperature Limits for Thermoplastic Pipe

ASTM Spec. No.	Material	Recommended Temperature Limits [Notes (1), (2)]		Hydrostatic Design Stress, ksi, at		
		Minimum, °F	Maximum, °F	73°F [Note (3)]	100°F	180°F
D1627	ABS1210	-90	180	1.0	0.8	--
	ABS1316	-90	180	1.6	1.25	--
	ABS2112	-90	180	1.25	1.0	--
D2282	ABS1210	-90	180	1.0	0.8	--
	ABS1316	-90	180	1.6	1.25	--
	ABS2112	-90	180	1.25	1.0	--
D2513	ABS1210	-90	180	1.0	0.8	--
	ABS1316	-90	180	1.6	1.25	--
	ABS2112	-90	180	1.25	1.0	--
D2646 F441 F442	CPVC4120	0	210	2.0	1.6	0.5
	CPVC4120	0	210	2.0	1.6	0.5
	CPVC4120	0	210	2.0	1.6	0.5
D2618	PB2110	0	210	1.0	0.8	0.5
D2104	PE2306	-90	140	0.63	0.4	--
	PE3306	-90	160	0.63	0.5	--
	PE3406	-90	180	0.63	0.5	--
	PE3408	-90	180	0.80	0.5	--
D2239	PE2306	-90	140	0.63	0.4	--
	PE3306	-90	160	0.63	0.5	--
	PE3406	-90	180	0.63	0.5	--
	PE3408	-90	180	0.80	0.5	--
D2447	PE2306	-90	140	0.63	0.4	--
	PE3306	-90	160	0.63	0.5	--
	PE3406	-90	180	0.63	0.5	--
	PE3408	-90	180	0.80	0.5	--
D2513	PE2306	-90	140	0.63	0.4	--
	PE3306	-90	160	0.63	0.5	--
	PE3406	-90	180	0.63	0.5	--
	PE3408	-90	180	0.80	0.5	--
D2737	PE2306	-90	140	0.63	0.4	--
	PE3306	-90	160	0.63	0.5	--
	PE3406	-90	180	0.63	0.5	--
	PE3408	-90	180	0.80	0.5	--
D3033	PE2306	-90	140	0.63	0.4	--
	PE3306	-90	160	0.63	0.5	--
	PE3406	-90	180	0.63	0.5	--
D3033	PE3408	-90	180	0.80	0.5	--

55

Nota. Tabla que muestra los límites de temperatura. Obtenido de American Society of Mechanical Engineers (2020). Norma ASME 31.9-2020. (s.p.) ASME.

Anexo 3.

Tabla de módulo de elasticidad y coeficiente térmico de expansión

Table 919.3.1 Moduli of Elasticity and Thermal Expansion Coefficients (Cont'd)

Material	Modulus of Elasticity, psi × 10 ⁶	Coefficient of Thermal Expansion, in./in./°F × 10 ⁻⁶ [Note (1)]	Linear Thermal Expansion, in./100 ft, Between 70°F and Indicated Temperature, °F															
			0	25	50	70	100	125	150	175	200	225	250	275	300	325	350	375
PE3366	0.13	70.0	—	—	-1.68	0	2.52	—	—	—	—	—	—	—	—	—	—	—
PE3406	0.15	60.0	—	—	-1.44	0	2.16	—	—	—	—	—	—	—	—	—	—	—
PE3608	0.125	80.0	—	—	-2.16	0	3.24	—	—	—	—	—	—	—	—	—	—	—
PE3708	0.125	90.0	—	—	-2.16	0	3.24	—	—	—	—	—	—	—	—	—	—	—
PE3710	0.125	90.0	—	—	-2.10	0	3.24	—	—	—	—	—	—	—	—	—	—	—
PE4708	0.130	80.0	—	—	-1.92	0	2.88	—	—	—	—	—	—	—	—	—	—	—
PE4710	0.130	80.0	—	—	-1.92	0	2.88	—	—	—	—	—	—	—	—	—	—	—
FF1110	—	46.0	—	—	-1.15	0	1.73	—	—	—	—	—	—	—	—	—	—	—
FF1208	—	43.0	—	—	-1.03	0	1.55	—	—	—	—	—	—	—	—	—	—	—
FF2105	—	46.0	—	—	-0.96	0	1.44	—	—	—	—	—	—	—	—	—	—	—

RTSP Consult manufacturer

NOTE: (1) Average of the mean values over the temperature range for which data are shown.

Nota. Tabla que muestra datos del módulo de elasticidad y coeficiente térmico de expansión. Obtenido de American Society of Mechanical Engineers (2020). Norma ASME 31.9-2020. (s.p) ASME.