



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA DE INGENIERÍA EN TELEINFORMÁTICA

TRABAJO DE TITULACIÓN
PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERA EN TELEINFORMÁTICA

ÁREA
ELECTROMAGNETISMO

TEMA
“PROTOTIPO DE INVERNADERO INTELIGENTE CON
USO DE RASPBERRY PARA CULTIVOS DIVERSOS”.

AUTORA
GUARICELA CASTRO ALEXANDRA CRISTINA

DIRECTOR DEL TRABAJO
ING. ELECT. PARRA LOPEZ RODOLFO ANTONIO, MG.

GUAYAQUIL, ABRIL 2021



ANEXO XI.- FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN

FACULTAD DE INGENIERÍA INDUSTRIAL CARRERA INGENIERÍA EN TELEINFORMÁTICA



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN			
TÍTULO Y SUBTÍTULO:			
Prototipo de invernadero inteligente con uso de Raspberry para cultivos diversos.			
AUTOR(ES) (apellidos/nombres):		Guaricela Castro Alexandra Cristina	
REVISOR(ES)/TUTOR(ES) (apellidos/nombres):		Ing. Gallegos Zurita Diana Ercilia, Mg / Ing. Parra López Rodolfo Antonio, Mg	
INSTITUCIÓN:		Universidad de Guayaquil	
UNIDAD/FACULTAD:		Facultad de ingeniería Industrial	
MAESTRÍA/ESPECIALIDAD:			
GRADO OBTENIDO:		Ingeniería en Teleinformática	
FECHA DE PUBLICACIÓN:		24 de septiembre de 2021	No. DE PÁGINAS: 127
ÁREAS TEMÁTICAS:		Electromagnetismo	
PALABRAS CLAVES/ KEYWORDS:		Automatización, Invernadero, Monitoreo, Raspberry pi, Sensores // Automation, Greenhouse, Monitoring, Raspberry pi, Sensors.	
RESUMEN/ABSTRACT (100-150 palabras): <p>Resumen:</p> <p>Este proyecto de tesis hace una propuesta de un invernadero para multicultivos haciendo uso de la tecnología de sensores y RASPBERRY PI, el prototipo muestra valores en tiempo real a través de una interfaz dinámica y amigable con su usuario.</p> <p>Dando a conocer las bondades de producción que un invernadero inteligente podría brindar a las personas que se interesen en hacer una plantación o cultivo en casa o a mayor escala, debido a que este sistema de control y riego haría el trabajo de monitoreo de los parámetros ambientales (temperatura y humedad) en el momento adecuado y así evitar la pérdida del cultivo. Uno de los principales objetivos de este trabajo es dar una solución práctica a los problemas que presentan los invernaderos.</p> <p>Específicamente se centró en los problemas que tienen los agricultores para usar la tecnología y llevar a la automatización los invernaderos, así como el excesivo consumo</p>			

de energía eléctrica, la mala lectura de las principales variables, así como el desperdicio de agua.

Abstract

This thesis Project proposes a greenhouse for multiple raising of crops using sensor technology and RASPBERRY Pi device. The prototype shows numbers in real time through a dynamic and user-friendly interface.

Letting to know the production benefits that a smart greenhouse can offer to people that are interested in crop raising at home or at major scale. Due to this irrigation and control system makes monitoring labor of environmental parameters (temperature and humidity) in the accurate times and avoiding crop loss. One of the main objectives of this project is to provide a practical solution to the problems that greenhouses represent.

Specifically, it focused on the agriculturists' problems for using technology and bring the automatization to the greenhouses, therefore the excessive electricity consumption, the incorrect lecture of the main variables, and so water waste.

ADJUNTO PDF:	SI X	NO
CONTACTO CON AUTOR/ES:	Teléfono: 0939779660	E-mail: guarialexa@hotmail.com
CONTACTO CON LA INSTITUCIÓN:	Nombre: Ing. Ramón Maquilón Nicola, MG.	
	Teléfono: 593-2658128	
	E-mail: direccionTi@ug.edu.ec	



**ANEXO XII.- DECLARACIÓN DE AUTORÍA Y DE
AUTORIZACIÓN DE LICENCIA GRATUITA**



**INTRANSFERIBLE Y NO EXCLUSIVA PARA EL USO NO COMERCIAL DE LA
OBRA CON FINES NO ACADÉMICOS**

**FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**

LICENCIA GRATUITA INTRANSFERIBLE Y NO COMERCIAL DE LA OBRA
CON FINES NO ACADÉMICOS

Yo, **GUARICELA CASTRO ALEXANDRA CRISTINA**, con C.C. No. **0950839209**, certifico que los contenidos desarrollados en este trabajo de titulación, cuyo título es **“PROTOTIPO DE INVERNADERO INTELIGENTE CON USO DE RASPBERRY PARA CULTIVOS DIVERSOS”** son de mi absoluta propiedad y responsabilidad, en conformidad al Artículo 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN*, autorizo la utilización de una licencia gratuita intransferible, para el uso no comercial de la presente obra a favor de la Universidad de Guayaquil.

A handwritten signature in blue ink, which appears to read "Alexandra Guarcela", enclosed within a blue oval.

GUARICELA CASTRO ALEXANDRA CRISTINA
C.C. No. 0950839209



ANEXO VII.- CERTIFICADO PORCENTAJE DE SIMILITUD

FACULTAD DE INGENIERÍA INDUSTRIAL CARRERA INGENIERÍA EN TELEINFORMÁTICA



Habiendo sido nombrado Ing. Rodolfo Parra López, Mg., tutor del trabajo de titulación certifico que el presente trabajo de titulación ha sido elaborado por Guaricela Castro Alexandra Cristina, C.C.: 0950839209, con mi respectiva supervisión como requerimiento parcial para la obtención del título de “INGENIERA EN TELEINFORMÁTICA”.

Se informa que el trabajo de titulación: “PROTOTIPO DE INVERNADERO INTELIGENTE CON USO DE RASPBERRY PARA CULTIVOS DIVERSOS”, ha sido orientado durante todo el periodo de ejecución en el programa Antiplagio (URKUND) quedando el 3% de coincidencia.



Document Information

Analyzed document	PROTOTIPO DE INVERNADERO INTELIGENTE CON USO DE RASPBERRY PARA CULTIVOS DIVERSOS.docx (D112258410)
Submitted	9/8/2021 11:13:00 PM
Submitted by	
Submitter email	alexandra.guaricelac@ug.edu.ec
Similarity	3%
Analysis address	ingrid.garcia@ug@analysis.orkund.com

<https://secure.orkund.com/view/106960671-226623-390238>



Firmado electrónicamente por:

**RODOLFO
ANTONIO PARRA
LOPEZ**

Ing. Rodolfo Parra López, Mg.
Tutor de trabajo de titulación
C.C. 0909770448

FECHA: 10 de septiembre de 2021



**ANEXO VI. - CERTIFICADO DEL DOCENTE-TUTOR DEL
TRABAJO DE TITULACIÓN
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



Guayaquil, 10 de septiembre de 2021

Sr (a).

Ing. Annabelle Lizarzaburu Mora, MG.

Director (a) de Carrera Ingeniería en Teleinformática / Telemática

**FACULTAD DE INGENIERÍA INDUSTRIAL DE LA UNIVERSIDAD DE
GUAYAQUIL**

Ciudad. -

De mis consideraciones:

Envío a Ud. el Informe correspondiente a la tutoría realizada al Trabajo de Titulación “Prototipode invernadero inteligente con uso de RASPBERRY para cultivos diversos” de la estudiante Guaricela Castro Alexandra Cristina, indicando que ha (cumplido con todos los parámetros establecidos en la normativa vigente:

- El trabajo es el resultado de una investigación.
- El estudiante demuestra conocimiento profesional integral.
- El trabajo presenta una propuesta en el área de conocimiento.
- El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se adjunta el certificado de porcentaje de similitud y la valoración del trabajo de titulación con la respectiva calificación.

Dando por concluida esta tutoría de trabajo de titulación, **CERTIFICO**, para los fines pertinentes, que el (los) estudiante (s) está (n) apto (s) para continuar con el proceso de revisión final.

Atentamente,



Firmado electrónicamente por:

**RODOLFO
ANTONIO PARRA
LOPEZ**

Ing. Rodolfo Parra López, Mg.

Tutor de trabajo de titulación

C.C. 0909770448

FECHA: 10 de septiembre de 2021



ANEXO VIII.- INFORME DEL DOCENTE REVISOR
FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA



Guayaquil, 23 de septiembre del 2021

Sr (a).

Ing. Annabelle Lizarzaburu Mora, Mg.

Director (a) de Carrera Ingeniería en Teleinformática / Telemática

FACULTAD DE INGENIERÍA INDUSTRIAL

UNIVERSIDAD DE GUAYAQUIL

Ciudad. -

De mi consideración:

Envío a usted el informe correspondiente de la REVISIÓN FINAL del Trabajo de Titulación titulado: ***“PROTOTIPO DE INVERNADERO INTELIGENTE CON USO DE RASPBERRY PARA CULTIVOS DIVERSOS”*** del estudiante, ***GUARICELA CASTRO ALEXANDRA CRISTINA***. Las gestiones realizadas me permiten indicar que el trabajo fue revisado considerando todos los parámetros establecidos en las normativas vigentes, en el cumplimiento de los siguientes aspectos:

Cumplimiento de requisitos de forma:

El título tiene un máximo de 11 palabras.

La memoria escrita se ajusta a la estructura establecida.

El documento se ajusta a las normas de escritura científica seleccionadas por la Facultad.

La investigación es pertinente con la línea y sublíneas de investigación de la carrera.

Los soportes teóricos son de máximo 5 años.

La propuesta presentada es pertinente.

Cumplimiento con el Reglamento de Régimen Académico:

El trabajo es el resultado de una investigación.

El estudiante demuestra conocimiento profesional integral.

El trabajo presenta una propuesta en el área de conocimiento.

El nivel de argumentación es coherente con el campo de conocimiento.

Adicionalmente, se indica que fue revisado, el certificado de porcentaje de similitud, la valoración del tutor, así como de las páginas preliminares solicitadas, lo cual indica el que el trabajo de investigación cumple con los requisitos exigidos.

Una vez concluida esta revisión, considero que el estudiante está apto para continuar el proceso de titulación.

Particular que comunicamos a usted para los fines pertinentes.

Atentamente,



Firmado electrónicamente por:
DIANA ERCILIA
GALLEGOS
ZURITA

Ing. Diana Gallegos Zurita, Mg.

Docente Tutor Revisor

C.C. 1204926313

FECHA: 23 de septiembre de 2021

Dedicatoria

A Dios, mi familia por su amor y apoyo incondicional desde el inicio de esta travesía.

Mi madre que me ha inculcado desde pequeña que a pesar de las adversidades que se puedan presentar en el camino que siempre con esfuerzo y perseverancia se podrán lograr nuestras metas.

A mis hermanos por el sacrificio para lograr mi meta.

Agradecimiento

A Dios, por brindarme sabiduría, salud y fuerzas para culminar mi carrera.

A mi familia, por su sacrificio y confianza puesta en mí.

A los amigos y compañeros con los que compartí y nos apoyamos en nuestra vida universitaria.

Índice del contenido

N°	Descripción	Pág.
	Introducción	1

Capítulo 1

	El problema	2
--	-------------	---

N°	Descripción	Pág.
1.1	Planteamiento del problema	2
1.2	Situación Conflicto. Nudos Críticos	2
1.2.1	Causas y consecuencias del Problema	3
1.3	Metodología del proyecto	3
1.4	Justificación e importancia	3
1.5	Objetivos	4
1.5.1	Objetivo general	4
1.5.2	Objetivo específico	4
1.6	Hipótesis prospectiva	4
1.6.1	Variables e indicadores	5
1.7	Preguntas de investigación	6
1.8	Alcance del proyecto	6

Capítulo II

Marco teórico

N°	Descripción	Pág.
2.1	Antecedentes del estudio	9
2.2	Fundamentación teórica	9
2.2.1	Invernadero	9
2.2.2	Sistema de riego	14
2.2.3	Comparación entre los sistemas de riego más comunes	17
2.3	Programación del Sistema De Riego	22
2.3.1	Frecuencia del Riego	22
2.3.2	Determinar el Agotamiento de la Humedad del Suelo (SMD)	22
2.3.3	Determinar el Índice de Aplicación del Sistema (AR)	24
2.3.4	Determinar la Duración del Riego	25

N°	Descripción	Pág.
2.4	Raspberry Pi	26
2.5	Desarrollo Web	27
2.5.1	Raspberry Pi OS	27
2.5.2	PHP	27
2.5.3	Python	27
2.5.4	Bootstrap	27
2.5.5	Xampp	28
2.5.6	Db Forge	28
2.5.7	AdminLTE	28
2.5.8	HTML5	29
2.6	API	29
2.7	Base de datos	29
2.8	Temperatura	29
2.9	Humedad Relativa	29
2.10	Sensores	30
2.10.1	Sensor DHT 22	30
2.10.2	Fototransistor	31
2.9.3	Sensor de humedad YL-69	32
2.11	CD4069	33
2.12	Definiciones conceptuales	32
2.13	Fundamentación legal	34

Capítulo III

Metodología

N°	Descripción	Pág.
3.1	Propuesta	35
3.2	Metodología del proyecto	36
3.2.1	Fases del PMI	36
3.3	Descripción	37
3.3.1	Factibilidad técnica	37

N°	Descripción	Pág.
3.3.2	Factibilidad legal	45
3.3.3	Factibilidad económica	46
3.3.4	Factibilidad operacional	47
3.4	Esquema general del proyecto	48
3.5	Sistema de Riego	49
3.6	Servidor XAMPP	50
3.7	Estructura Base de Datos	51
3.7.1	Detalle de la Base de datos	51
3.7.2	Modelo de la Base de datos	51
3.7.3	Información de ambiente de Programación	52
3.8	Procedimiento	53
3.8.1	Prueba de funcionalidad	53
3.8.2	Aplicación Web	53
3.8.3	Prueba	56
3.9	Conclusiones y recomendaciones	60
3.9.1	Conclusiones	60
3.9.2	Recomendaciones	60
	Anexos	104
	Bibliografía	105

Índice de tablas

Nº	Descripción	Pág.
1	Conceptualización y operacionalización de las variables	5
2	Indicadores	5
3	Invernadero plano sus ventajas y desventajas	12
4	Invernadero túnel sus ventajas y desventajas	13
5	Invernadero capilla sus ventajas y desventajas	14
6	Invernadero gótico sus ventajas y desventajas	15
7	Adaptación a los cultivos de los diferentes sistemas de riego	18
8	Adaptación a las características del terreno de los diferentes sistemas de Riego	18
9	Consumo de agua de los diferentes sistemas de Riego	19
10	Calidad de agua de los diferentes sistemas de Riego	19
11	Eficiencia de los diferentes sistemas de riego	19
12	Control de agua aplicados en los diferentes sistemas de riego	20
13	Mano de obra utilizada en los diferentes sistemas de riego	20
14	Comportamiento de diferentes sistemas de riego según la sostenibilidad del sistema de producción	21
15	Propiedades Físicas para Diferentes Texturas	23
16	Valores Óptimos en el análisis del suelo	26
17	Especificación del Software utilizado para el sistema	38
18	Recurso Hardware- Servidor	38
19	Datos del sensor	39
20	Muestreo de bomba	42
21	Duración de Riego	42
22	Características de la placa Raspberry Pi	44
23	Recursos técnicos	47
24	Recursos Humanos	47
25	Presupuesto total de recursos empleados	47
26	Descripción de la Base de datos	51
27	Estructura del ambiente de Programación	52

Índice de figuras

Nº	Descripción	Pág.
1	Invernadero	10
2	Invernadero Plano	11
3	Invernadero Túnel	12
4	Invernadero Capilla	12
5	Invernadero Gótico	13
6	Sistema de riego por aspersión	14
7	Sistema de riego por goteo	15
8	Riego por inundación	15
9	Sistema Microaspersión	16
10	Sistema de Riego	16
11	Sistema de riego fertirrigación	20
12	Determinando la Humedad del Suelo por medio del Tacto y Apariencia	22
13	Riego por Goteo para Cultivos de Hileras	24
14	Uniformidad de Distribución	24
15	Placa de Raspberry Pi	25
16	Php	26
17	Python	26
18	Bootstrap	27
19	Xampp	27
20	Db Forge	27
21	Vista de AdminLTE	28
22	Diferentes tipos de sensores	29
23	Sensor DHT 22	30
24	Sensor de humedad-YL-69	31
25	Circuito integrado CD4096	32
26	Esquema del hardware del prototipo	34
27	Sensor DTH 22 a utilizar en el prototipo	38

N°	Descripción	Pág.
28	Sensor de humedad-YL-69	39
29	Bomba de Agua aire	40
30	Placa del microcontrolador a utilizar en el prototipo (Raspberry Pi4)	42
31	Software Phyton	44
32	Simulación de conexión	44
33	Estructura Backend	47
34	Estructura Backend del sistema capaz de controlar todos los sensores	47
35	Sistema por Goteo	48
36	Levantamiento de Servidor MySQL	49
37	Levantamiento de la base de datos en MySQL	49
38	Modelo de Base de Datos en dbForge	50
39	Configuración del sistema web con la base de datos	51
40	Limpieza de Cache	51
41	Inicio de sesión para el Invernadero Inteligente	53
42	Formulario de registro de usuarios	53
43	Vista y registro de los perfiles de los usuarios en el sistema	54
44	Dashboard del sistema	54
45	Registro de las mediciones tomadas por los sensores	55
46	Elección del terreno para el invernadero	55
47	Construcción del invernadero	56
48	Prueba del sensor YL-69 y DTH22	57
49	Registro del sensor YL-69 en la página web	57
50	Agenda para programar el riego	58
51	Conexión del sistema de Riego	58
52	Registro del sistema de riego en la página web	58



ANEXO XIII.- RESUMEN DEL TRABAJO DE TITULACIÓN (ESPAÑOL)

FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA



“PROTOTIPO DE INVERNADERO INTELIGENTE CON USO DE RASPBERRY PARA CULTIVOS DIVERSOS”

Autor: Guaricela Castro Alexandra Cristina

Tutor: Ing. Parra López Rodolfo Antonio, Mg

Resumen

Este proyecto de tesis hace una propuesta de un invernadero para multicultivos haciendo uso de la tecnología de sensores y RASPBERRY PI, el prototipo muestra valores en tiempo real a través de una interfaz dinámica y amigable con su usuario.

Dando a conocer las bondades de producción que un invernadero inteligente podría brindar a las personas que se interesen en hacer una plantación o cultivo en casa o a mayor escala, debido a que este sistema de control y riego haría el trabajo de monitoreo de los parámetros ambientales (temperatura y humedad) en el momento adecuado y así evitar la pérdida del cultivo. Uno de los principales objetivos de este trabajo es dar una solución práctica a los problemas que presentan los invernaderos.

Específicamente se centró en los problemas que tienen los agricultores para usar la tecnología y llevar a la automatización los invernaderos, así como el excesivo consumo de energía eléctrica, la mala lectura de las principales variables, así como el desperdicio de agua.

Palabras Claves: Automatización, Invernadero, Monitoreo, Raspberry pi, Sensores.



**ANEXO XIV.- RESUMEN DEL TRABAJO DE
TITULACIÓN (INGLÉS)**

**FACULTAD DE INGENIERÍA INDUSTRIAL
CARRERA INGENIERÍA EN TELEINFORMÁTICA**



**“SMART GREENHOUSE PROTOTYPE WITH RASPBERRY USAGE FOR
DIVERSE CROP RAISING”**

Author: Guaricela Castro Alexandra Cristina

Advisor: Ing. Parra López Rodolfo Antonio, Mg

Abstract

This thesis Project proposes a greenhouse for multiple raising of crops using sensor technology and RASPBERRY Pi device. The prototype shows numbers in real time through a dynamic and user-friendly interface.

Letting to know the production benefits that a smart greenhouse can offer to people that are interested in crop raising at home or at major scale. Due to this irrigation and control system makes monitoring labor of environmental parameters (temperature and humidity) in the accurate times and avoiding crop loss. One of the main objectives of this project is to provide a practical solution to the problems that greenhouses represent.

Specifically, it focused on the agriculturists' problems for using technology and bring the automatization to the greenhouses, therefore the excessive electricity consumption, the incorrect lecture of the main variables, and so water waste.

Keywords: Automation, Greenhouse, Monitoring, Raspberry pi, Sensors.

Introducción

Raspberry es una serie de ordenadores reducida, con el objetivo de poner en manos de las personas de todo el mundo el poder de la informática y la creación digital.

Contando con un software de código abierto haciendo que se facilite el acceso a las nuevas tecnologías inteligentes, las cuales ayudan a distintas áreas en el campo laboral.

En esta tesis se expone la resolución de un problema de la vida real, que tiene como objetivo ofrecer una solución eficaz a través de la automatización de un invernadero inteligente. Esta tesis está dirigida al sector agrícola, directamente al cultivo de plantas en invernaderos. Para la automatización del invernadero y llevar a cabo diferentes tareas sin intervención humana.

Por ende, un invernadero automatizado tiene un microclima óptimo para el desarrollo de un cultivo o varios a la vez.

Partiendo de un análisis técnico de microclimas para el desarrollo de las múltiples especies a cultivar, se necesita obtener la temperatura, humedad relativa, luminosidad, riego y ventilación para conseguir la alta productividad a bajo precio y menos tiempo, sin daño ambiental.

Capítulo I.- Tiene relación con la introducción de los temas de los invernaderos y además se expone los objetivos a conseguir, la justificación del asunto. Así como una breve descripción del contenido de la tesis.

Capítulo II.- Se presentan las propiedades y los elementos de los varios tipos de invernaderos para la formación de los circuitos para el invernadero.

Capítulo III.- En este capítulo es considerada la factibilidad económica, como en la operativa y técnica. Con el fin de tener una aceptación del prototipo por parte de la muestra elegida, obteniendo conclusiones y recomendaciones, tomando en cuenta las opiniones para mejorar el prototipo.

Capítulo I

El Problema

1.1 Planteamiento del problema

En la actualidad la integración de la tecnología en la vida diaria da una gran variedad de posibilidades y facilita las actividades o tareas, las cuales eran complicadas por algún factor y ahora para las nuevas generaciones son simples por ende para la mayoría de los agricultores o personas particulares cuya función principal es el cultivo de hortalizas ya sea para el autocosumo o para la venta, abordan la problemática del bajo nivel de automatización en el manejo de los cultivos y el poco conocimiento en el uso de fuentes alternativas de energía en la agricultura.

Entre los sistemas más usados por los agricultores está el sistema por riego por aspersión y riego por goteo los cuales ya tienen más de 10 años en el mercado ecuatoriano que si se lo maneja en un contexto ambiental hace que se desperdicie el recurso hídrico.

Actualmente, existen productos comerciales para automatizar invernaderos, debido a que la autonomía del proceso de cultivo facilita la eficiencia, así como también la velocidad del trabajo que se ejecuta, sin embargo, el acceso a estas tecnologías de automatización, suelen ser muy costosas para el productor y generalmente son empleadas para la producción a gran escala.

Los sistemas actuales de riego presentan varios problemas en los cuales podemos destacar sus elevados costos que oscilan entre 500 a 3000 dólares los cuales hacen su adquisición un poco costosa para el productor.

En este proyecto se desarrolla una opción más económica, que sea fácil de implementar y de usar por el productor en la automatización de invernaderos, además de que pueda estar al alcance de personas que deseen cosechar en menor escala. Este trabajo se llevó a cabo usando la plataforma de código abierto Raspberry Pi, utilizando la placa Raspberry Pi 4 junto con otros componentes electrónicos para poder realizar el diseño del sistema de control y de riego. Además, se considera el diseño de una página web, la cual está creada en Laravel. Por medio de esta página web, se visualiza el monitoreo de las variables involucradas en el invernadero, como son: temperatura, humedad relativa, humedad del suelo, intensidad de luz. Asimismo, se pueden establecer condiciones para el control

automatizado, así como comprobar el buen funcionamiento del sistema. El cálculo de un sistema de riego también se lleva a cabo, por el interés de considerar el buen uso del recurso hídrico en el proceso que se requiera en este proyecto.

1.2 Situación Conflicto. Nudos Críticos

El presente proyecto busca aplicar la tecnología a la agricultura, para ayudar a impulsar y fortalecer la agricultura orgánica para una alimentación sana y saludable y solucionar los problemas que se presentan por: inundaciones, plagas y sequías que provocan pérdidas de los cultivos y económicas en la Parroquia General Vernaza del Cantón Salitre, pero debido a los acontecimientos que están ocurriendo por la pandemia no se pudo realizar la implementación en el lugar destinado por ese motivo la implementación se lo realizará en la ciudad de Guayaquil.

1.2.1 Causas y consecuencias del Problema

Causas:

- Nulidad en implementación tecnológica
- Deforestación masiva.

Consecuencias:

- Pérdida de los cultivos.
- Desperdicio del recurso hídrico.

1.3 Justificación e importancia

El presente proyecto de tesis surge ante la necesidad de mejorar el proceso de multicultivos de una manera automatizada, debido a los cambios climáticos que se producen en el país. Se busca desarrollar un prototipo que desempeñe mediante funciones específicas un buen monitoreo, que genere una mayor seguridad y tranquilidad a los agricultores o personas particulares.

Gracias al avance de nuevas tecnologías digitales, tanto en hardware como en software nos facilitan la obtención y transmisión de datos, de una gran variedad de sensores digitales capaces de detectar las mediciones que se evidencian al controlar el interior del invernadero según la programación realizada. Con el uso de estos componentes electrónicos se puede obtener un control de forma automatizada las variables de temperatura, humedad e intensidad de luz y riego del suelo al interior del invernadero, logrando proporcionar un mayor mantenimiento y control de los cultivos

1.4 Objetivos

1.4.1 Objetivo general.

Diseñar e implementar un prototipo de un sistema inteligente, para controlar de forma eficiente un invernadero, así como también por medio de una página web, se podrán visualizar las mediciones que se obtengan de los sensores para monitorear y garantizar el buen funcionamiento del sistema.

1.4.2 Objetivos específicos.

- Identificar las variables climáticas y diferentes procesos que intervienen en los cultivos.
- Seleccionar e implementar los sensores y materiales adecuados para obtener el registro de mediciones del invernadero a controlar, con el desarrollo de un código fuente de una plataforma con una interfaz que permitirá al usuario revisar su historial de mediciones.
- Explicar la metodología empleada para el desarrollo del prototipo del invernadero inteligente que cumple con los requerimientos técnicos ajustados a las necesidades de los cultivos.
- Realizar las pruebas de funcionalidad del invernadero con sensores que muestren las medidas en una muestra, con el fin de confirmar el correcto funcionamiento del invernadero en varios cultivos.

1.5 Hipótesis prospectiva

Se elabora un invernadero de bajo costo que mida y controle en tiempo real las condiciones en las que se encuentra un cultivo de plantas, lo que reduce el riesgo de pérdida en los cultivos, teniendo de esta manera mejores resultados en el mantenimiento de un invernadero.

1.5.1 Variables e indicadores

1.5.1.1 Variable independiente

Desarrollo sistema de riego automatizado para invernaderos.

Causa: desarrollar un prototipo de invernadero inteligente para ayudar al desarrollo de un cultivo en tiempo real y de esta manera prevenir daños, evitando perder los cultivos.

1.5.1.2 *Variable dependiente*

Tipo de mediciones que detecta.

Consecuencia: Toma de temperatura, luminosidad, humedad, etc a una distancia máxima de 2m, logrando tener una idea aproximada de la posición en la que se encuentran los cultivos, consiguiendo así evitar una posible pérdida.

Conceptualización y operacionalización de las variables

Tabla 1

Conceptualización y operacionalización de las variables.

Variable	“Desarrollo sistema de riego automatizado para invernaderos.”	“Tipo de mediciones que detecta”
Definiciones conceptuales	El desarrollo de un prototipo es una etapa previa a la obtención de un producto final, es decir un modelo de trabajo o una versión no pulida de un producto, en este caso un invernadero inteligente que ayuda al desarrollo de un cultivo.	Al referirnos a un invernadero Inteligente, hablamos de automatizar el entorno y procesos que conlleva sembrar las diferentes plantas que se desee el agricultor o personas que gusten de la agricultura.
Definiciones operacionales	El desarrollo de este prototipo engloba aspectos específicos que puntualizan su correcto funcionamiento. Es así como el radio de cobertura refiriéndose al espacio en el cual se va a leer la las mediciones propone una media	Para el desarrollo del prototipo se utiliza una Raspberry, con sensores DTH-22, encargado de medir la temperatura y el sensor de humedad-YL-69 relativa del suelo en el invernadero, brindando señales para ser mostradas al usuario.

Indicadores

Tabla 2.

Indicadores.

Variables	Indicadores
-----------	-------------

“Desarrollo sistema de riego automatizado para invernaderos.	<ul style="list-style-type: none"> - Presupuesto - Diseño - Invernadero (plantas)
“Tipo de mediciones que detecta”	<ul style="list-style-type: none"> - Temperatura del invernadero - Humedad del suelo - Luminosidad del invernadero

1.6 Preguntas de investigación

¿Cuáles son los principales problemas que presentan los cultivos a lo largo de su desarrollo?

¿Qué beneficios otorgaría a los agricultores automatizar sus invernaderos?

¿Cuál es el microclima y los parámetros apropiados para el desarrollo de un cultivo?

¿Cuán asequible sería el prototipo si surgiera en el mercado?

1.7 Alcance del proyecto

Se diseñará un prototipo funcional de invernadero para el cultivo de plantas comestibles en ambientes controlados y se controlará de forma automatizada las variables de temperatura, iluminación y humedad del sustrato o suelo al interior del invernadero.

Los objetivos específicos que rigen el adecuado proceso de esta investigación se complementan con técnicas que permitan acceder a información que conformarán el marco teórico, así mismo dichas técnicas permiten la descripción de las metodologías usadas. El método empleado en este proceso responde al tipo hipotético deductivo, facilitando la realización de las pruebas de correcta funcionabilidad.

Además de no tener un costo demasiado elevado como lo son otras herramientas lo cual los hace pensar mucho en si están en posibilidades de conseguirlo.

Capítulo II

Marco teórico

2.1 Antecedentes del estudio

Una de las profesiones más antiguas que ha desarrollado el ser humano, es la agricultura, a lo largo de la historia, desde entonces el hombre ha tenido que luchar contra los factores climáticos donde ya se daban cultivos para conseguir alimento.

Las personas con el paso del tiempo tuvieron la necesidad de desarrollar varios métodos para proteger los cultivos dando como resultado los invernaderos.

En cuanto a la estructura agraria ecuatoriana, permanece la división entre la Agricultura Empresarial (AE) y la Agricultura Familiar Campesina (AFC) con graves distorsiones. A saber, la Agricultura Empresarial concentra 80% de la tierra en un 15% de las Unidades de Producción Agrícolas (UPAs), utiliza el 63% del agua para riego y hace un uso indiscriminado de agroquímicos y energía para la agroexportación. La Agricultura Familiar representa el 84,5% de las UPAs con una concentración de 20% de la tierra, cuenta con 37% del agua para riego y se dedica principalmente a la producción para la satisfacción de las necesidades básicas. (agricultura, 2016)

En la Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingeniería y diseño Industrial de Madrid en el año 2015 el estudiante Barroso García Andrés, realizó el Control y monitorización de un invernadero a través de una aplicación móvil, de las variables del sistema de medición de una forma rápida y cómoda. (Barroso García Andrés, 2019)

En la Fundación universitaria los Libertadores ubicada en Bogotá en el año 2017 los estudiantes Julio Enrique Barón Alarcón y Diego Fernando Pérez Orozco realizaron un invernadero de riego y temperatura controlado remotamente desde el web basado en tecnologías móviles, la aplicación web toma los valores o estados de los campos de la base de datos y los muestra por pantalla en un entorno gráfico donde el usuario visualiza las acciones que el sistema está haciendo en el invernadero. (Julio Barón & Diego Pérez, 2017)

Este proyecto tiene como objetivo principal crear un sistema inteligente, para el controlar de manera eficiente un invernadero, así como también por medio de una página web, se podrán visualizar las mediciones que se obtengan de los sensores para monitorear y garantizar el buen funcionamiento del sistema que ayude a los agricultores o terceras personas en sus cultivos, aumentando la producción y economía.

La automatización de procesos busca sustituir tareas o actividades que son realizadas por humanos con la capacidad de dispositivos mecánicos, electrónicos y de control.

2.2 Fundamentación teórica

2.2.1 Invernadero

Se entiende por invernadero un lugar cerrado, estático y dotado habitualmente de una cubierta translúcida de vidrio o de plástico exterior, destinado al cultivo de plantas dentro del cual se puede obtener un microclima mediante el cual se puede tener el control de la temperatura, humedad y de otros factores ambientales, además, se pueden proporcionar sistemas automáticos de riego y ventilación, lo cual se utiliza para la producción y de cultivos de forma controlada. (INSST, s.f.)



Figura 1. Invernadero.

Fuente: Información adaptada de la página web freepik.

2.2.1.1 Tipos de invernaderos

Para el proyecto se hará uso de un prototipo de invernadero túnel representado en una maqueta y el cual se describe a continuación.

2.2.1.1.1 Invernadero Plano

Este tipo de invernadero es usado en zonas poco lluviosas, su construcción no es muy aconsejable debido a que su ventilación es mala y su infraestructura envejece rápidamente. (Horticultivos, 2017)

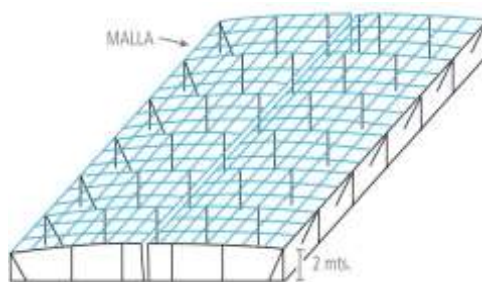


Figura 2. Invernadero Plano.

Fuente: Información adaptada de la página web horticultivos.

2.2.1.1.2 Ventajas y desventajas de un invernadero plano

Tabla 3.

Invernadero plano sus ventajas y desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"> • Su costo en la construcción es económico. • Gran adaptabilidad a la geometría del terreno. • Mayor resistencia al viento. • Aprovecha el agua de lluvia en periodos secos. • Presenta una gran uniformidad luminosa. 	<ul style="list-style-type: none"> • Poco volumen de aire y/o mala ventilación. • La instalación de ventanas cenitales es bastante difícil. • Demasiada especialización en su construcción y conservación. • Rápido envejecimiento de la instalación • Es poco o nada aconsejable en lugares lluviosos.

Fuente: Información tomada de la página Horticultivos.

2.2.1.1.3 Invernadero Túnel

Este invernadero es diseñado especialmente para cultivos en zonas pequeñas. Son invernaderos generalmente económicos debido a su estructura que, a pesar de ser simple es muy resistente para el control de los factores climáticos y de fácil traslado. (Horticultivos, 2017)

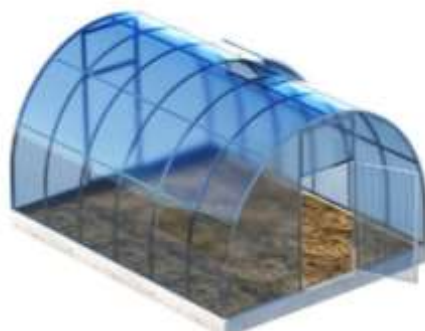


Figura 3. Invernadero Túnel.

Fuente: Información adaptada de la página web horticultivos

2.2.1.1.4 Ventajas y desventajas de un invernadero túnel

Tabla 4.

Invernadero túnel sus ventajas y desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"> • Alta resistencia a los vientos y fácil instalación (recomendable para productores que se inician en el cultivo protegido). • Alta transmisión de la luz solar. • Apto tanto para materiales de cobertura flexibles como rígidos. 	<ul style="list-style-type: none"> • No aprovecha el agua de lluvia.

Fuente: Información tomada de la página Horticultivos.

2.2.1.1.5 Invernadero Capilla

Este tipo invernadero es apto para climas templados, sin embargo, al reforzar las estructuras del invernadero puede ser apto para cualquier tipo de clima.

El invernadero brinda una óptima transmisión de luz, buena ventilación y, además permite una buena movilidad dentro del invernadero. (Agricultoras, 2016)



Figura 4. Invernadero Capilla.

Fuente: Información adaptada de la página web Novagric.

2.2.1.1.6 Ventajas y desventajas de un invernadero capilla

Tabla 5.

Invernadero capilla sus ventajas y desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"> • Buena estanqueidad a la lluvia y al aire. • Permite la instalación de ventilación cenital, así como ventilación perimetral • Buen reparto de la luminosidad en el interior del invernadero. • Fácil instalación. • Permite realizar labores agrícolas mecanizadas en su interior. 	Ninguna relevante

Fuente: Información tomada de la página web Novagric.

2.2.1.1.7 Invernadero Gótico

Este tipo de invernadero permite proporcionar un mayor volumen de aire, lo que permite un mejor microclima e iluminación.

Su diseño está adaptado para climas extremos, todo tipo de cultivos y soportar grandes cargas además de exigir ciertos cuidados y condiciones ambientales para el cultivo. (Horticultivos, 2017)

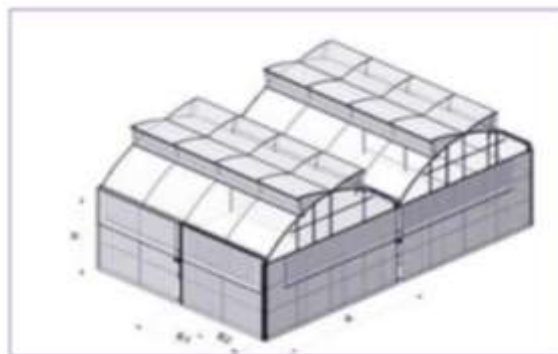


Figura 5. Invernadero Gótico.

Fuente: Información adaptada de la página web Novagric.

2.2.1.1.8 Ventajas y desventajas de un invernadero Gótico

Tabla 6.

Invernadero gótico sus ventajas y desventajas

Ventajas	Desventajas
<ul style="list-style-type: none"> Alta duración y resistencia a la corrosión. Eficacia de fijación del plástico de cubierta (buena hermeticidad). Montaje rápido y sin soldaduras. Mayor distancia de la ventilación al cultivo 	Ninguna relevante

Fuente: Información tomada de la página web Horticultivos

2.2.2 Sistema de Riego

Es un conjunto de estructuras, que se usan para determinar qué área puede ser cultivada y aplicar el agua necesaria a las plantas. Constando de varios componentes y dichos componentes dependerán del tipo de sistema de riego empleado. (Sepúlveda, 2018)

2.2.2.1 Riego aspersión

El sistema de riego por aspersión es uno de los múltiples métodos de riego de cultivos que existen en la actualidad. Consiste en aplicar el agua imitando la lluvia, es decir, mediante un chorro de agua pulverizada en gotas. El mecanismo funciona a través de una red de tuberías que transporta el agua hasta los aspersores, los cuales utilizan presión para

dispararla. El riego como tal es potenciado a través de un sistema de bombeo. (Agropinos, 2020)



Figura 6. Sistema de riego por aspersión.

Fuente: Información tomada de la página agricultureros.

2.2.2.2 Riego por goteo

De forma muy general, se puede definir el Riego por Goteo como Riego Localizado. El riego por goteo o riego gota a gota es un método de irrigación que permite una óptima aplicación de agua y abonos en los sistemas agrícolas de las zonas áridas. (Maherelectronica, 2018)

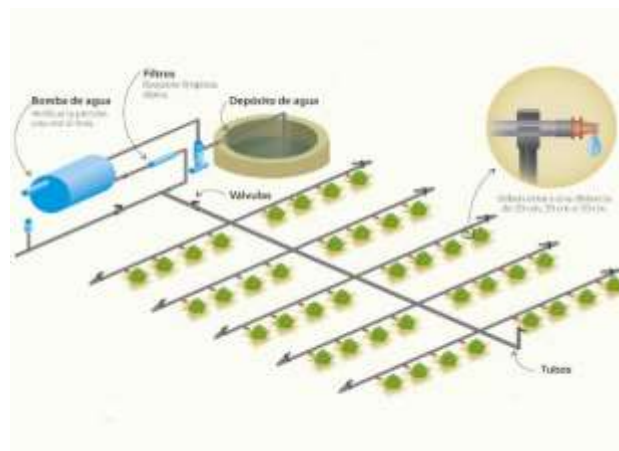


Figura 7. Sistema de riego por goteo.

Fuente: Información tomada de la página web gestiriego.

2.2.2.3 Inundación

El agua es distribuida superficialmente sobre el terreno de regadío, inundándolo totalmente o en partes. (FAO, 2016)



Figura 8. Riego por inundación

Fuente: Información tomada de la página web Agroquímicos-Arca-SA.

2.2.2.4 Microaspersión

Es una modificación del sistema de aspersión tradicional que permite asperjar el agua a poca distancia de la planta y de manera localizada. (Agriculturas, 2016)



Figura 9. Sistema Microaspersión.

Fuente: Información tomada de la página hidrocultiva.

2.2.3 Comparación entre los sistemas de riego más comunes

Una comparación entre los diferentes sistemas de riego, con relación a variables de carácter técnico-agronómico, social y económico, puede ayudar a la toma de decisión acerca de qué tipo de sistema implementar en un determinado proyecto. (FAO, 2016)



Figura 10. Sistema de Riego.

Fuente: Información adaptada de FAO

2.2.3.1 Adaptación a los cultivos

Tabla 7.

Adaptación a los cultivos de los diferentes sistemas de riego

Inundación	Se adapta mejor a los cultivos que permiten la inundación total (por ejemplo: arroz) y a aquellos cuyo espaciamiento entre hileras posibilita la construcción del camellón y el surco.
Aspersión	Su adaptación a aquellos cultivos que ocupan toda el área del terreno o que poseen espaciamiento pequeño, cuyos sistemas radicales ocupan todo el volumen de suelo sembrado. Ejemplo: pastos. Es poco eficiente para aquellos cultivos sembrados en espalderas o ramadas, porque estas estructuras no permiten la distribución uniforme del agua. No es un sistema adecuado para los cultivos altos y frondosos.
Microaspersión	Se adapta mejor a aquellos cultivos de espaciamiento amplio, cuyos sistemas radicales no ocupan todo el volumen de suelo y, por lo tanto, el riego localizado es más eficiente. Ejemplo: frutales arbóreos o no arbóreos, güisquil, loroco, etc
Goteo	Se adapta mejor a aquellos cultivos de espaciamiento amplio, cuyos sistemas radicales no ocupan todo el volumen de suelo y, por lo tanto, el riego localizado es más eficiente. La forma y tipo de crecimiento de los cultivos sean altos y frondosos o bajos y de escaso follaje, no tiene importancia para el riego por goteo. Donde el agua es escasa y el diferencial de altura pequeño, es mejor que el sistema de micro-aspersión.

Fuente: Información tomada de la página web FAO

2.2.3.2 Adaptación a las características del terreno

Tabla 8.

Adaptación a las características del terreno de los diferentes sistemas de Riego

Inundación	Se adapta mejor en caso de terrenos planos o semi planos, normalmente aluviones a las orillas de ríos y quebradas. En áreas altas y con más pendiente, las dificultades para conducir y distribuir adecuadamente el agua son muy grandes. Además, los riesgos de erosión son elevados.
Aspersión	Se adapta mejor a terrenos planos o semi planos. A medida que aumenta la pendiente, se incrementa el riesgo de erosión debido al desprendimiento de partículas de suelo que causa el impacto de las gotas.
Microaspersión Y Goteo	Por igual, ambos sistemas se adaptan bien a terrenos de cualquier pendiente.

Fuente: Información tomada de la página web FAO.

2.2.3.3 Consumo de agua

Tabla 9.

Consumo de agua de los diferentes sistemas de Riego

Inundación	Es el que más consume agua, por lo tanto, con excepción de las vegas de los ríos, donde no hay altura para instalar otros sistemas por gravedad, los sistemas por inundación deberían ser evitados, principalmente donde el agua es escasa.
Aspersión	Utiliza menos agua que el anterior, pero por lo menos el doble que el sistema de goteo.
Microaspersión Y Goteo	Por igual, ambos sistemas se adaptan bien a terrenos de cualquier pendiente.

Fuente: Información tomada de la página web FAO

2.2.3.4 Calidad del agua

Tabla 10.

Calidad de agua de los diferentes sistemas de Riego

Inundación	Puede trabajar con agua de mala calidad física (sedimentos, algas, detritos, etc.).
Aspersión	La presencia de detritos sólidos suele bajar la eficiencia del sistema, porque taponan los picos de los aspersores, cambiándoles el caudal de distribución. Además, partículas duras, como arena, desgastan las boquillas, afectando la uniformidad de distribución del agua a mediano plazo.
Microaspersión Y Goteo	Ambos sistemas son muy susceptibles a la presencia de sólidos en suspensión, porque taponan los orificios de los micro aspersores o goteros. Inclusive, la presencia de algunas sales puede taponarlos, al precipitarse en las boquillas cuando cesa el riego.

Fuente: Información tomada de la página web FAO

2.2.3.5 Eficiencia de riego

Tabla 11.

Eficiencia de los diferentes sistemas de riego

Inundación	40 - 65%
Aspersión	80 - 85%
Microaspersión	85 - 90%
Goteo	90 - 95%

Fuente: Información tomada de la página web FAO

(Relación del agua benéficamente utilizada en los cultivos y el agua total utilizada en la práctica del riego)

2.2.3.6 Control del agua aplicada

Tabla 12.

Control de agua aplicados en los diferentes sistemas de riego

Inundación	Aunque es posible controlar la cantidad de agua aplicada a través del uso de compuertas o sifones, es muy difícil saber exactamente qué cantidad de agua se está aplicando en cada punto del terreno. Para que haya un humedecimiento uniforme del suelo es necesario que la superficie sea bien uniforme para que no se produzcan pozas y lugares secos.
Aspersión	Se puede controlar bien la cantidad de agua aplicada a través de pluviómetros sencillos o por la relación caudal/tiempo de los aspersores. Sin embargo, si la presión en los aspersores es baja o desigual entre ellos, la distribución del agua también es desigual dentro del perímetro regado; por lo tanto, se aplica más agua en algunas áreas que en otras.
Microaspersión Y Goteo	En ambos sistemas se puede controlar bien la cantidad de agua aplicada a través de la relación caudal/tiempo de los micro aspersores o goteos o por el consumo en la estructura de almacenamiento (pila, barriles, etc.).

Fuente: Información tomada de la página web FAO

2.2.3.7 Utilización de mano de obra

Tabla 13.

Mano de obra utilizada en los diferentes sistemas de riego

Inundación	Los requerimientos de mano de obra para la construcción y mantenimiento de canales suelen ser altos. Igualmente, en la preparación del terreno y la construcción de camellones y surcos, aunque esta labor puede ser realizada con la ayuda de tracción animal. En la operación del riego, se requiere de mano de obra permanente para las operaciones de distribución y control del agua.
Aspersión	El mayor uso de mano de obra está en la operación del sistema y, más específicamente, en la rotación de los equipos en el terreno en cada turno de riego. En todo caso, esta labor es menos ardua que la requerida para el manejo de los sistemas de inundación, principalmente por surcos.

Microaspersión Y Goteo	Como se trata de sistemas fijos, gran parte de la mano de obra es utilizada en la instalación del equipo. La operación consiste en una visita diaria para inspeccionar el sistema y, en particular, el funcionamiento de los micro aspersores o goteros.
-------------------------------	--

Fuente: Información tomada de la página web FAO

Análisis aproximado del comportamiento de diferentes sistemas de riego según algunos indicadores relacionados a los atributos de sostenibilidad del sistema de producción.

Tabla 14.

Comportamiento de diferentes sistemas de riego según la sostenibilidad del sistema de producción.

Indicadores de sostenibilidad	Sistema de Riego		
	Inundación por surcos	Aspersión	Goteo
Utilización de mano de obra en la operación	Alta	Mediana	Baja
Consumo de agua	Alto	Mediano	Bajo
Control de consumo de agua	Alto	Bajo	Bajo
Riesgo de erosión en zonas de laderas	Alto	Mediano	Bajo
Posibilidades de generar conflicto por el agua	Alto	Mediano	Bajo

2.2.3.8 Riego fertirrigación

La fertirrigación es una técnica de riego mediante la cual se incorporan los nutrientes minerales que necesitan los cultivos a través del agua. Su uso está muy extendido en sistemas de riego localizado, ya que estos sistemas son los que más óptimamente aportan los nutrientes a las raíces, minimizando las pérdidas por lixiviación. (Agrícolas, 2016)



Figura 11. Sistema de riego fertirrigación.

Fuente: Información tomada de la página Novagric.

Para el proyecto se hará uso del sistema de **fertirrigación** en riego por goteo debido a que al haber comparado con los sistemas de riego más comunes cumplen con las variables técnico-agronómico, social y económico.

2.3 Programación del Sistema De Riego

2.3.1 Frecuencia del Riego

La cantidad de tiempo entre los riegos.

La frecuencia del riego se determina por el uso de la cantidad o índice de agua de cultivo en relación con el agua disponible en el suelo. El uso del agua de cultivo es afectado por el tamaño de la planta y las condiciones climáticas.

Entre más grande sea la planta, junto con las condiciones climáticas cálidas, secas y el viento; la cantidad de absorción será más grande. Las plantas más jóvenes necesitan menos agua pero también necesitan ser regadas más frecuentemente que las plantas maduras porque tienen raíces poco profundas y se secan rápidamente. Las plantas extraen agua del suelo a una proporción medida en pulgadas por día. El agua almacenada en el suelo se mide en pulgadas de agua por las pulgadas de la profundidad del suelo.

2.3.2 Determinar el Agotamiento de la Humedad del Suelo (SMD)

La cantidad de agua que tiene que ser repuesta desde el último riego es el Agotamiento de la Humedad del Suelo (SMD). El Agotamiento de la Humedad del Suelo se puede determinar utilizando estos métodos:

- Calcular el Agotamiento de la Humedad en la zona radical efectiva utilizando la gráfica/guía del USDA para determinación de la humedad.
- Determinar la pérdida de agua evapotranspiración (ET) desde el último riego. La curva del crecimiento de la planta y/o el porcentaje de cobertura del follaje se pueden utilizar para determinar la evapotranspiración del cultivo (ETc) durante las diferentes etapas del desarrollo conforme las plantas llegan a su madurez. A su madurez, las fresas tienen una evapotranspiración de cultivo (ETc) de 0.85

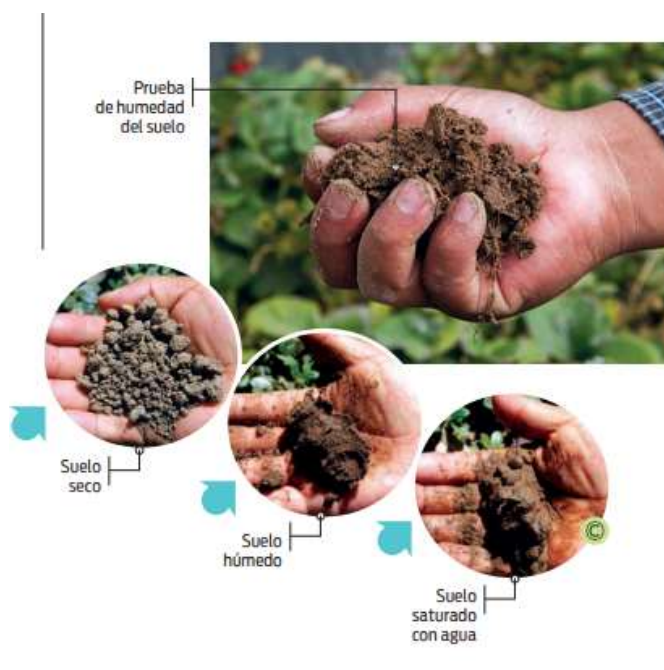


Figura 12. Determinando la Humedad del Suelo por medio del Tacto y Apariencia.

Fuente: Información tomada de la página CRCD.

Elaboración: Terri Lajda

Determinar el SMD utilizando la guía del USDA para determinar la humedad del campo

El Agotamiento de la Humedad del Suelo (SMD) en pulgadas/pie (in/ft) cuando la apariencia y tacto del suelo son una de las siguientes

Tabla 15.

Propiedades Físicas para Diferentes Texturas

	Arena Fina y Franco Arenoso Fino	Franco Arenoso y Migajón Arenoso Fino	Franco Arenoso de Arcilla y Arcilla	Arcilla, Franco Arcilloso
La humedad del campo 50-75%	Húmeda, forma una bola débil. Granos sueltos de arena y juntos permanecen en los dedos, manchas oscuras de agua. 0.2 - 0.6 (in/ft)	Húmeda, forma una bola con pocos granos de suelo juntos quebrándose. Manchas claras de agua, color oscuro. 0.3-0.9 (in/ft)	Húmeda, forma una bola firme con marcas de los dedos bien definidas, capa irregular de suelo/agua en los dedos. Color oscuro y flexible. 0.4-1.1 (in/ft)	Húmeda, forma una bola suave con marcas definidas de los dedos, pocos o sin granulados en los dedos. Flexible, cintas entre el dedo pulgar y el dedo

				índice. 0.4-1.2 (in/ft)
La humedad del campo 75-100%	Mojada, forma una bola débil, granos sueltos de arena y agregados procedentes de capas desiguales sobre los dedos. 0.0-0.3 (in/ft)	Mojada, forma una bola, sin agua, aparece en la superficie cuando se aprieta o sacude. Capas irregulares de suelo/agua sobre los dedos. 0.0 -0.4 (in/ft)	Mojada, forma una bola sin agua aparece sobre la superficie del suelo cuando se aprieta o se sacude. Capa irregular de suelo/agua sobre los dedos. 0.0 - 0.5 (in/ft)	Mojada, forma suelo blando puede brillar después de ser apretada o sacudida. Capa de suelo/agua de ligera a pesada sobre los dedos, cintas fáciles. 0.0 - 0.6 (in/ft)
La humedad del campo 100%	Mojada, forma una bola débil. Sin brillo de agua, brevemente sobre la superficie cuando se sacude. Contorno mojado en la mano después de ser apretada. 0.0 (in/ft)	Mojada, forma una bola blanda, sin agua apareciendo brevemente sobre la superficie del suelo cuando se aprieta o sacude. Capa irregular de suelo/agua sobre los dedos. 0.0 (in/ft)	Mojada, forma un pedacito de suelo blando con agua con brillo de agua sobre la superficie después de apretarse o sacudirse. Suelo grueso. 0.0 (in/ft)	Mojada, forma un pedacito de suelo blando. Capa gruesa de suelo/ agua sobre los dedos. El suelo brilla, resbaloso y pegajoso, no se forman cintas. 0.0 (in/ft)

Fuente: Información tomada de la página web USDA

2.3.3 Determinar el Índice de Aplicación del Sistema (AR)

El índice de aplicación es la profundidad del agua aplicada por el sistema riego sobre cierta área por hora, por lo general se mide en pulgadas por hora

Si el índice de aplicación (AR) es de 0.2 pulgadas por hora, éso quiere decir que por cada hora que el sistema está prendido, equivale a un promedio de 0.2 pulgadas de lluvia.

Índice de Aplicación por Goteo

Utilizar la tabla para un índice de aplicación aproximado o utilizar el índice de aplicación provisto en el reporte del laboratorio móvil de riego.

Ancho de la Cama [pulgadas]	Índice de Descarga de la Cinta [galones/minuto por 100 pies]				
	0.30	0.35	0.40	0.45	0.50
24	0.14	0.17	0.19	0.22	0.24
28	0.12	0.14	0.17	0.19	0.21
32	0.11	0.13	0.14	0.16	0.18
36	0.10	0.11	0.13	0.14	0.16
40	0.09	0.10	0.12	0.13	0.14
44	0.08	0.09	0.11	0.12	0.13
48	0.07	0.08	0.10	0.11	0.12
52	0.07	0.08	0.09	0.10	0.11
56	0.06	0.07	0.08	0.09	0.10
60	0.06	0.07	0.08	0.09	0.10
64	0.05	0.06	0.07	0.08	0.09
68	0.05	0.06	0.07	0.08	
72	0.05	0.06	0.06	0.07	0.08
76	0.05	0.05	0.06	0.07	0.08

Figura 13. Riego por Goteo para Cultivos de Hileras.

Fuente: Información tomada de la página CRCD.

Elaboración: Terri Lajda

2.3.4 Determinar la Duración del Riego

Ajustar para la Uniformidad de Distribución (DU)

Se necesita aplicar más agua cuando el agua de riego no se está distribuyendo uniformemente para que las plantas que reciben menos agua puedan hacerlo. Los sistemas de riego no son 100% eficientes al distribuir el agua. Cuando un sistema ha sido diseñado y mantenido óptimamente:

El riego por aspersión puede lograr típicamente 75% (0.75) de Uniformidad de Distribución

El riego por goteo puede lograr típicamente 85% (0.85) de Uniformidad de Distribución. La Uniformidad de Distribución se reduce cuando no hay suficiente presión, presión irregular, obstrucciones, fugas y otros problemas de diseño y mantenimiento. Utilizar Uniformidad de Distribución aproximada o utilizar la Uniformidad de Distribución provista en el reporte del laboratorio móvil de riego.



Figura 14. Uniformidad de Distribución.*Fuente:* Información tomada de la página CRCD.*Elaboración:* Terri Lajda**Formula**

$$\text{Duración del Riego} = \frac{\text{Agotamiento de la Humedad del Suelo (SMD)}}{\text{Índice de Aplicación (AR) * Uniformidad de Distribución (DU)}}$$

Valores óptimos de elementos mayores en análisis de suelo**Tabla 16.**

Valores Óptimos en el análisis del suelo

NITROGENO N	ANHIDRIDO FOSFORICO P2O2	OXIDO DE POTASIO K2O
Clima frio 0,25 – 0,50%	>50 ppm	20 -40 moo/100g
Clima medio 0,15 – 0,25%		

Fuente: Información tomada de la página web manual de producción en Coalcomán Michoacán.*Elaboración:* Mora Amanda.**2.4 Raspberry Pi**

La Raspberry Pi es un ordenador de bajo coste y tamaño reducido, tanto es así que cabe en la palma de la mano, pero puedes conectarle un televisor y un teclado para interactuar con ella exactamente igual que cualquier otra computadora. (Xakata, 2018)

**Figura 15.** Placa de Raspberry Pi.*Fuente:* Información tomada de la página Xataka.

Especificaciones: Las Raspberry Pi poseen una cabecera GPIO 40 pines, cada uno de ellos está identificado y se utiliza para diferentes funciones.

GND: Abreviatura de “GROUND”, indica conexión a tierra y se puede encontrar varios de ellos en las placas.

5V: 5V como lo indica, posee 5 voltios de energía. Ya que la mayoría de los elementos con los que trabaja la Raspberry Pi se encuentran en el rango de los 5 voltios.0

2.5 Desarrollo Web

2.5.1 Raspberry Pi OS

Raspberry Pi OS es una distribución del sistema operativo GNU/Linux basado en Debian, y por lo tanto libre para la SBC Raspberry Pi, orientado a la enseñanza de informática.

2.5.2 PHP

Es un lenguaje de programación de código abierto que sirve para desarrollo web y que puede ser desarrollado en HTML. (PHP, 2016)



Figura 16. Php.

Fuente: Tomada de la página DevMagazine.

2.5.3 Python

Es un lenguaje de programación interpretado cuya principal filosofía es que sea legible por cualquier persona con conocimientos básicos de programación. (Visus, 2020)



Figura 17. Python.

Fuente: Tomada de la página Blogthikgbig. Bootstrap

Bootstrap es un paquete de herramientas de código abierto que sirve para desarrollos web responsive con HTML, CSS y JavaScript. En la cual se puede dar forma a un sitio web a través del uso de sus librerías CSS y JavaScript. (Fontanela, 2020)



Figura 18. Bootstrap.

Fuente: Tomada de la página [getbootstrap](https://getbootstrap.com/).

2.5.4 Xampp

Es una distribución de Apache gratis muy fácil de instala y usar que contiene MariaDB, PHP y Perl. (FriendsApache, 2021)



Figura 19. Xampp.

Fuente: Tomada de la página [AuriboxTraining](https://www.auriboxtraining.com/).

2.5.5 Db Forge

DbForge Studio para MySQL es un entorno integrado que sirve para el desarrollo, administración y gestión de bases de datos en MySQL y MariaDB. El IDE le posibilita producir y llevar a cabo consultas, desarrollar y depurar rutinas almacenadas, automatizar la gestión de objetos de la base de datos, examinar datos de tablas por medio de una interfaz intuitiva. (Devart, 2018)

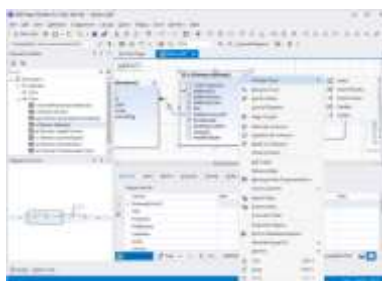


Figura 20. Db Forge.

Fuente: Información tomada de la página [Db Forge](https://www.devart.com/dbforge/).

2.5.6 AdminLTE

AdminLTE es un panel de control que sirve para Bootstrap realizado a través del estudio Almsaeed. Es un recurso de suministro libre basado principalmente en un diseño modular que da paso a una creaciones y personalizaciones fluidas. La iniciativa es que todos los activos establecidos sean un complemento o un widget mediante el cual se construye la interfaz del cliente tanto para las partes de front-end como de back-end. (BBVA, 2016)



Figura 20. Vista de AdminLTE.

Fuente: Información tomada de AdminLTE

2.6 HTML5

Siendo la última versión de HTML que aún en desarrollo y en conjunto con CSS3, tiene nuevos modelos para el desarrollo web que ayudan a rediseñar el código resolviendo problemas y actualizando nuevas necesidades. (Mozilla, 2015)

2.7 API

Las API como se conoce comúnmente en la actualidad sirven para usos exclusivos de aportes en empresas grandes para la elaboración de software a medida en menos tiempos como beneficio únicamente de ser la interfaz que comunica entre sistemas. (Chilán, 2019)

2.8 Temperatura

La temperatura es una intensidad física que sugiere la energía interna de un cuerpo humano, de un objeto o ambiental generalmente, medida por un termómetro. Dicha energía interna se expresa en términos de calor y frío, siendo el primero asociado con una

temperatura más alta, mientras que el frío se asocia con una temperatura más baja.
(Significados, 2019)

2.9 Humedad Relativa

Es la relación que existe entre la cantidad de vapor existente en el aire y la que debería tener para saturarse a igual temperatura. Esto quiere decir que cuando se habla de una humedad relativa del 50%, quiere decir que de la totalidad de vapor de agua que puede abarcar el aire a esa temperatura, solo tiene el 50%. (Conceptodefinicion, 2019)

2.10 Sensores

Son dispositivos electrónicos que poseen la capacidad de detectar variaciones como: temperatura, iluminación, movimiento, presión, pH, entre otras. Estos sensores están controlados por un sistema, obedeciendo a una previa programación que determina su funcionamiento. Son llamados elementos primarios debido a que son los encargados de dar una respuesta requerida sobre la variable que se está censando. (Nicolás Avellaneda, s.f.)

“El sensor traduce la información que le llega del exterior en un impulso eléctrico, normalmente digital (pasa o no pasa corriente), que puede ser analizado y procesado por la unidad de control del sistema.” (Quimis & Cantos, 2018)

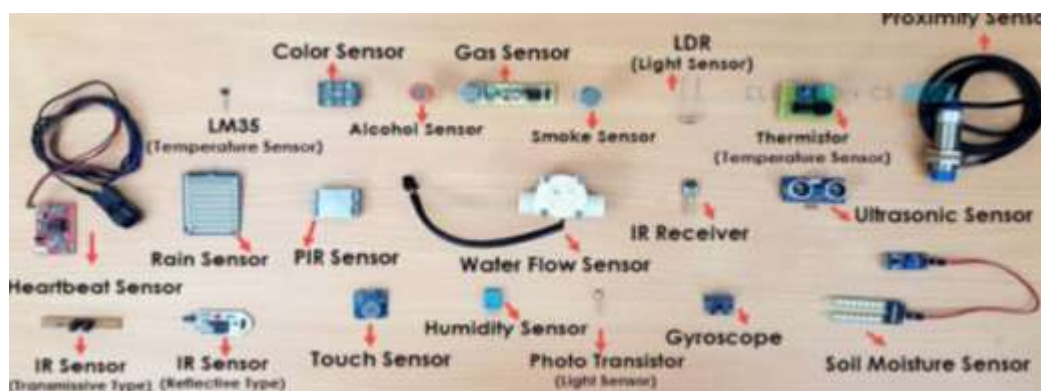


Figura 21. Diferentes tipos de sensores.

Fuente: Información tomada de la página paolaguimerans.

2.10.1 Sensor DHT 22

Es un sensor básico digital de medición de temperatura y humedad relativa de buen rendimiento y bajo precio. Este sensor muestra los datos por medio de una señal digital en el pin de datos debido a que no tiene una salida analógica. Usado en aplicaciones de control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más.

Utilizar el sensor DHT22 con las plataformas Arduino/Raspberry Pi/Nodemcu es muy sencillo tanto a nivel de software como hardware. A nivel de software se dispone de librerías para Arduino con soporte para el protocolo "Single bus". (Naylamp, 2016)

Según indica Naylamp las características de este sensor son:

- Voltaje de Operación: 3V - 6V DC
- Rango de medición de temperatura: -40°C a 80°C
- Precisión de medición de temperatura: $\leq \pm 0.5^{\circ}\text{C}$
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: De 0 a 100% RH
- Precisión de medición de humedad: 2% RH
- Resolución Humedad: 0.1% RH
- Tiempo de sensado: 2s
- Interface digital: Single-bus (bidireccional)
- Modelo: AM2302
- Dimensiones: 20*15*8 mm
- Peso: 3 gr.
- Carcasa de plástico blanca



Figura 22. Sensor DHT22.

Fuente: Información tomada de la página Aosong.

2.10.2 Fototransistor

El fotorresistor es un resistor que está hecho de material semiconductor, y la conductancia cambia con variación de luminancia. El fotorresistor se puede fabricar con diferentes figuras e iluminar área basada en esta característica. El fotorresistor es

ampliamente utilizado en muchas industrias, como juguetes, lámparas, cámara, etc.

(Alcubierre, 2014)

Según (electronicaplugandplay, s.f.) Indica las siguientes características

Rango de variación: 10 K Ω - 1 M Ω .

Tiempo de respuesta: De 20 a 30 milisegundos.

Longitud de onda Respuesta Máxima: 540 nm.

Voltaje Máximo: 150 V.

Potencia Máxima: 100 mW.

Fabricante: Senba Optical.

2.10.3 Sensor de humedad- YL-69

Este sensor puede medir la humedad presente en el suelo que lo bordea. Siendo un sensor de baja gama tecnológica, pero sigue siendo ideal para hacer un seguimiento en un jardín urbano. Utilizando las dos sondas en la cual una servirá para pasar corriente a través del suelo, y luego leer la resistencia para obtener el nivel de humedad. Mientras más agua hace que el suelo conduzca la electricidad con mayor facilidad (menos resistencia), mientras que el suelo seco es un mal conductor de la electricidad (más resistencia).

(Electronilab, 2016)

Según indica (Geekfactory, s.f.) Las características de este sensor son:

- Salida no calibrada, el usuario debe encontrar los puntos de funcionamiento óptimos para su aplicación.
- Voltaje de alimentación: 5 VDC.
- Sensor de humedad en suelo basado en conductividad eléctrica.
- Ideal para mediciones en pequeñas macetas.
- Apto para proyectos escolares, pruebas de concepto o maquetas.
- Basado en el comparador LM393.
- Salida analógica y digital.



Figura 23. Sensor de humedad-YL-69.

Fuente: Información tomada de la página Geekfactory

2.11 CD4069

Es un inversor CMOS hexagonal en el encapsulado DIP de 14 pines. Se compone de seis circuitos CMOS inversor. El CD4069UBE está destinado a todas las aplicaciones del inversor de propósito general donde no se requieren capacidades de conversión de unidad TTL potencia media y de nivel lógico. Las aplicaciones típicas incluyen la inversión de la lógica, la formación de impulsos, osciladores y amplificadores de alta impedancia de entrada. (Electronica, 2021)



Figura 24. Circuito integrado CD4096.

Fuente: Tomada de la página Carrod Electrónica.

Según indica (Electronica, 2021) sus especificaciones son las siguientes:

- Tipo: CMOS
- Familia: CD4000
- Tipo circuito lógico: Inversor
- Corriente de salida: 6.8 mA
- Número de entradas: 1
- Tensión de alimentación mínima: 3 V
- Tensión de alimentación máxima: 18 V
- Temperatura de funcionamiento mínima: -55 ° C
- Temperatura de funcionamiento máxima: 125 ° C
- Encapsulado DIP
- 14 pines

2.12 Definiciones conceptuales

Microcontrolador: “Microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento que puede guardar programas y sus cambiantes (flash y RAM) pareciéndose a una pequeña PC que incluyen los sistemas para el control de recursos de entrada/salida.” (Heptro, 2017)

Software: “Es el soporte lógico e inmaterial que permite a la computadora realizar tareas inteligentes, dirigiendo los componentes físicos o equipamiento con instrucciones y datos a través de diferentes tipos de programas.” (Buzón, 2020)

Servidor: “Un servidor es un equipo informático que forma parte de una red y provee servicios a otros equipos cliente.” (Systems, 2015)

Base de datos: “Es una colección ordenada de datos, comúnmente almacenados en un marco de computadora en la cual luego puede acceder el usuario para administrar, modificar, actualizar, controlar y organizar de manera más fácil los datos que se encuentren en la base.” (Oracle, 2016)

MariaDB: Es un sistema que administra una base de datos relacional.

Xampp: “Es un paquete de instalación de servidor web independiente, que gestiona primordialmente la base de datos MySQL, el servidor web Apache y los intérpretes para idiomas de script: PHP y Perl.” (Zepeda, 2018)

Laravel: “Es uno de los frameworks de código abierto más fáciles de relacionar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.” (Garcia, 2015)

JavaScript: Lenguaje de programación ligero, que interpreta y compila funciones complejas en una consola.

PHP: “Es un lenguaje de programación de código abierto que sirve para el desarrollo web y ser introducido en HTML.” (PHP, 2016)

Relé: “Es un aparato eléctrico accionado eléctricamente, que funciona como un interruptor, abriendo y cerrando el paso de la corriente eléctrica.” (Areatecnologia, 2021)

2.13 Fundamentación legal

Para complementar el análisis que se realiza en este proyecto se deben expresar las bases legales. Por eso surge la necesidad de citar los artículos de la constitución de la república del ecuador que fundamentan el proyecto que se realizara. Ver anexo 1.

.

CAPITULO III

3.1 PROPUESTA

Antes de plasmar el análisis de las factibilidades técnicas y operacionales del prototipo del invernadero inteligente se presenta el siguiente diseño técnico y esquemático por medio del cual se realizaron previas pruebas de funcionalidad, mediante un software de simulación. Se usa una Raspberry Pi como parte importante del diseño final del prototipo, introduciendo en la programación condiciones y variables para así lograr que los sensores todo el tiempo estén detectando las necesidades del cultivo, automatizar el proceso normal de un invernadero y guardar un registro diario de los parámetros obtenidos por los sensores.

A continuación, se detalla el esquema del hardware del prototipo, realizando simulaciones en software fritzing.

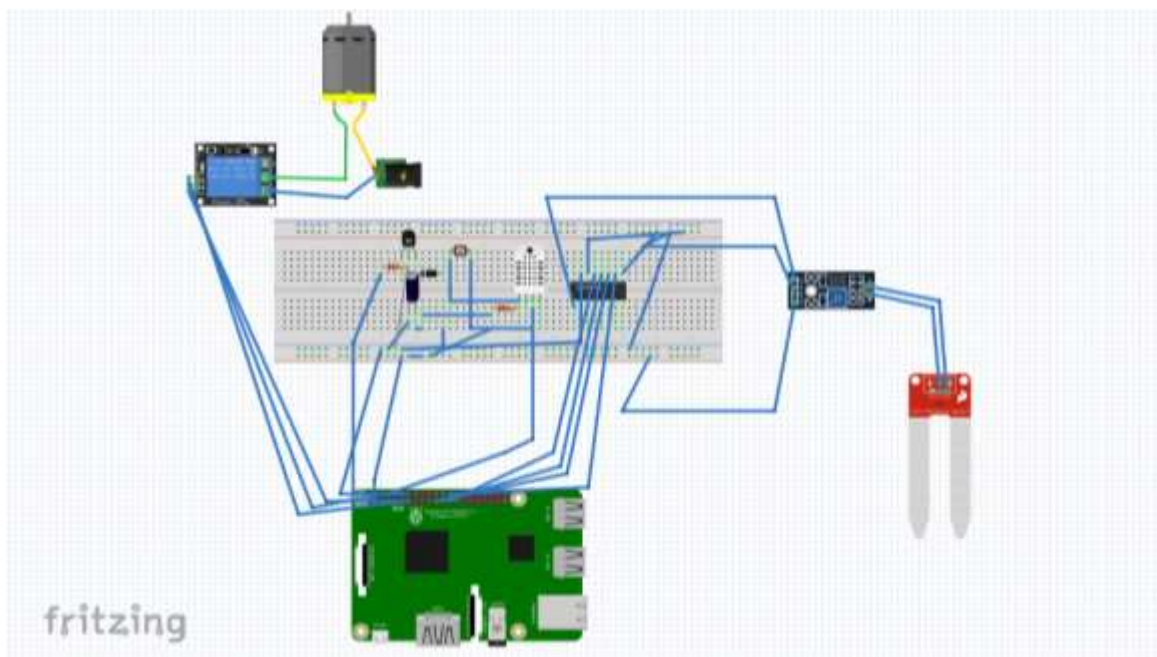


Figura 25. Esquema del hardware del prototipo.

3.2 Metodología del proyecto

El método utilizado en este estudio responde al PMI (Project Management Institute), “Es una de las asociaciones profesionales de miembros más grandes del mundo que cuenta con medio millón de miembros e individuos titulares de sus certificaciones en 180 países. Es una organización sin fines de lucro que avanza la profesión de la dirección de proyectos a través de estándares y certificaciones reconocidas mundialmente, a través de comunidades de colaboración, de un extenso programa de investigación y de oportunidades de desarrollo profesional”. (PMI, 2018)

En el primer capítulo se especifica las fases del PMI, a continuación se detallará cada una de las fases a realizar:

3.2.1 Fases del PMI

Fase de inicio: como se detalla en el capítulo uno de este proyecto, se define la ejecución y su posterior entrega, con el propósito de dejar claro una problemática y brindar una resolución específica.

Fase de planificación: en esta fase se deja claro la parte investigativa del proyecto, pudiendo de esta manera realizar el diseño del prototipo del invernadero inteligente. Se elabora un esquema del proceso que se efectúa según su actividad, como se ve a continuación.

Fase de ejecución: en esta tercera fase se realizan las configuraciones de cada uno de los dispositivos que conforman nuestro prototipo, las actividades que se ejecutaron son:

- Configurar las funciones que va a desempeñar la Raspberry Pi mediante software de uso libre que lleva el nombre de Python y Laravel.
- Condicionar en este mismo software la distancia de medición de cada uno de los sensores, a una distancia máxima de 2m que es el tamaño del invernadero en el prototipo.
- Ensamblar cada uno de los elementos hardware en el invernadero, albergando cada dispositivo en su respectiva posición.

Fase de monitoreo y control: en esta etapa del proyecto se realizaron las pruebas necesarias y pertinentes para verificar y constatar el correcto funcionamiento del prototipo del invernadero inteligente. El monitoreo y control se realizó en cada una de las pruebas tanto en hardware como en software en el prototipo.

- **Fase de cierre:** luego de llevar a cabo con éxito cada una de las fases, obteniendo resultados del prototipo del invernadero inteligente a continuación, se brinda el código fuente del proyecto:

3.3 Descripción

Desarrollo de un prototipo del invernadero inteligente con sensores de temperatura, luminosidad y humedad buscando aplicar la tecnología a la agricultura, para ayudar a impulsar y fortalecer la agricultura.

De acuerdo con las características del prototipo antes mencionadas, la presente tesis mostrará su funcionalidad de forma física, ya que en esta propuesta de diseño se pretende utilizar un microcontrolador de bajo costo como Raspberry Pi, previo a especificaciones analizadas, sensor de temperatura DTH-22, sensor de luminosidad, sensor de humedad-YL-69 y un sistema de riego automatizado, facilitando el crecimiento de las plantas.

3.3.1 Factibilidad técnica

Cumple con la factibilidad técnica porque el hardware reúne los requisitos necesarios para poder cumplir con el propósito del invernadero inteligente como lo es el control de temperatura, humedad, luminosidad, etc., los dispositivos electrónicos de hardware propuestos para la elaboración del invernadero inteligente son económicos y asequibles en el mercado local, por lo tanto se llega a la conclusión de que se considera factible el proyecto técnicamente, los software que se utilizan al ser de código libre facilitan la descarga de librerías y la utilización de guía para poder desarrollar a cabalidad la programación.

El uso de estas herramientas básicamente compone el cuerpo de la aplicación Web y Laravel que permite mostrar al front end (el lado del cliente) es decir que sea amigable para el usuario. Uno de los componentes XAMPP que facilita el manejo a nivel de servidor y gestión de datos de la base de datos de la aplicación Web.

Laravel que es un intermedio de comunicación entre la parte del back end (el lado del servidor) y el front end, que usara los datos que guarda y los mostrara en una página basada en HTML.

Tabla 17.

Especificación del Software utilizado para el sistema.

Herramientas	Característica
SO (Sistema Operativo)	Multisistema portable
Base de Datos	MariaDB
Servidor de Aplicaciones	Apache
Framework	Laravel
Lenguaje de Programación	JavaScript/Php

Una de las ventajas de usar este tipo de tecnologías de código abierto es lo fácil que puede ser de interpretar para el usuario y los beneficios de su escalabilidad a nivel estructural facilitan como anteriormente en la tabla 15 sobre el recurso del software. El gestor de base de datos amigable adaptable a múltiples IDE para el manejo de información.

En la tabla 16 se detalla uno de los beneficios a nivel de hardware que brinda como servidor.

Tabla 18.

Recurso Hardware- Servidor

Características	Especificaciones
Procesador	Core i3 6ma Generación
Memoria	4 Gb RAM
Disco Duro	1 TB
Monitor "15"	Portátil
Periféricos	Teclado, Mouse, Cámara Web

Hardware para utilizar

A continuación, se detallan cada uno de los elementos a utilizar para el desarrollo del prototipo del invernadero inteligente.

3.3.1.1 *Sensor DTH 22*

Son sensores digitales de Temperatura y Humedad, fáciles de implementar con cualquier microcontrolador (Raspberry pi 4). Usando un termistor para medir el aire que lo rodea y un sensor capacitivo de humedad y solo un pin para la lectura de los datos. Tal vez la desventaja de estos es la velocidad de las lecturas y el tiempo que se debe esperar para tomar nuevas lecturas (nueva lectura después de 2 segundos), pero debido a que la temperatura y humedad son variables que no cambian muy rápido en el tiempo.



Figura 26. Sensor DTH 22 a utilizar en el prototipo

Tabla 19.

Datos del sensor

Características	Detalles
Alimentación	$3.3V_{dc} \leq V_{cc} \leq 6V_{dc}$
Rango de medición de temperatura	$-40^{\circ}C$ a $+80^{\circ}C$
Precisión de medición de temperatura	error de $\pm 0.5^{\circ}C$
Resolución Temperatura	$0.1^{\circ}C$
Rango de medición de humedad	De 0 a 100% RH
Precisión de medición de humedad:	2% RH
Resolución Humedad	0.1%RH
Tiempo de sensado	2s

Señal de salida (Echo)	Pulso en estado lógico algo, tiempo proporcional en uS
------------------------	--

Fuente: Tomada de la página koalab.

3.3.1.2 Sensor YL-69

Es sensor de humedad de suelo resulta ser otro módulo que utiliza la conductividad entre dos terminales para determinar ciertos parámetros relacionados a agua, líquidos y humedad

Antes se ha hablado de los sensores de humedad relativa, como el DHT22. Este nos devuelve la humedad en el ambiente, pero no es capaz de medir la humedad en el suelo. Para este propósito se usa el Módulo YL-69, un sensor de humedad de suelo.



Figura 27. Sensor de humedad-YL-69.

Fuente: Tomada de la página Geekfactory.

3.3.1.3 Bomba Agua aire

Independientemente de sus características o potencia, se puede controlar un equipo de bombeo mediante un procesador, siendo de hecho frecuente que estén controlados por un autómata. Raspberry Pi, podemos encender cualquier tipo de bomba de agua mediante las salidas digitales y el uso de una salida por relé. (Llamas, 2016)

Según (EC, 2020) las características de esta bomba son:

Máximo rango de bombeo de 2m.

Máxima cabeza de 3m.

Maneja líquidos con temperaturas de hasta 80°C

Flujo máximo entre 1 - 3 L/min.

Voltaje de trabajo 6V - 12V DC

Corriente: 0.5A - 0.7A

Horas de servicio: 2500h



Figura 28. Bomba de Agua aire.

Fuente: Tomada de la página Talos Electronics.

Función de la bomba

Una bomba de agua consta de un actuador, la mayoría de las veces un motor eléctrico, que está conectado a un elemento giratorio llamado rodete. El rodete está formado por palas que giran para transferir parte de la energía al fluido que pasa. Normalmente, la hoja permanece curvada para formar una guía para las partículas. Su forma define la relación entre la energía transferida a un fluido y la cantidad distribuida entre velocidad o presión. (Llamas, 2016)

¿Cómo se calcula el caudal?

Para poder calcular el caudal, hay que saber que se debe dividir el volumen en un tiempo o sea es la cantidad de agua que es capaz de entregar una bomba en un tiempo determinado. (SA, 2014)

Fórmula

$$Q = V/t \text{ (min)}$$

Q = Caudal **V** = Volumen **T** = tiempo

Tabla 20.

Muestreo de bomba

MUESTREO BOMBA						
Interacción	Fecha	Recipiente (cant)	Tiempo (min)	Bomba	Litros	Voltaje
1	04/06/2021	1 Ltr	1	1	1,12	10,52V
2	05/06/2021	1 Ltr	1.20	2	1,43	10,53V
3	06/06/2021	1 Ltr	1.40	3	2,39	10,54V
4	07/06/2021	1 Ltr	2.50	1	1,44	10,55V
5	08/06/2021	1 Ltr	1.51	2	2,40	11,56V
6	09/06/2021	1 Ltr	2.51	3	2,40	12,57V
7	10/06/2021	1 Ltr	1.52	2	1,45	13,58V
8	11/06/2021	1 Ltr	2.52	1	2,41	10,59V
9	12/06/2021	1 Ltr	1.53	2	2,41	10,6V
10	13/06/2021	1 Ltr	2.59	3	2,90	11,2V
					2,03	11,224
					5	

3.3.1.4 Calcular el duración de riego para el cultivo de fresas y ajos

$$\text{Duración del Riego} = \frac{\text{Agotamiento de la Humedad del Suelo (SMD)}}{\text{Índice de Aplicación (AR) * Uniformidad de Distribución (DU)}}$$

Tabla 21.

Duración de Riego

Muestreo para la duración del Riego			
Interacción	Fecha	Bomba	Tiempo (min)
1	04/06/2021	1	180
2	05/06/2021	1	120
3	06/06/2021	1	140
4	07/06/2021	1	250

Cálculos para los Fertilizantes

Fertilizantes Líquidos

Los fertilizantes líquidos deben ser convertidos primero a galones. La etiqueta del fertilizante líquido debería indicar la densidad en libras por galón. Este valor puede utilizarse para convertir el análisis garantizado en la etiqueta a libras de fertilizante y luego a libras de N, P y K aplicado.

¿Cuántas libras de N –P – K en 100 galones de 5-10-10 fertilizante líquido, con una densidad de 11.65 lbs/gal?

Primero convertir el volumen del líquido a una masa (libras aplicadas):

$$100 \text{ galones} \times 11.65 \text{ lbs/galón} = 1,165 \text{ lbs de fertilizante}$$

Segundo hacer sus calculaciones como se indica abajo:

- $1,165 \text{ lbs de fertilizante} \times 0.05 \text{ (recuerde que este es el 5\%)} = 58.25 \text{ lbs de N}$
- $1,165 \text{ lbs de fertilizante} \times 0.10 = 116.5 \text{ lbs P}_2\text{O}_5$ $116.5 \times 0.44 = 51.3 \text{ lbs de P}$
- $1,165 \text{ lbs de fertilizante} \times 0.10 = 116.5 \text{ lbs K}_2\text{O}$ $116.5 \times 0.83 = 96.7 \text{ lbs de K}$

3.3.1.5 *Raspberry Pi*

La Raspberry Pi es un ordenador de bajo coste y tamaño reducido, tanto es así que cabe en la palma de la mano, pero puedes conectarle un televisor y un teclado para interactuar con ella exactamente igual que cualquier otra computadora. (Xakata, 2018)



Figura 29. Placa del microcontrolador a utilizar en el prototipo (Raspberry Pi4).

Fuente: Xakata.

3.3.1.6 *Cómo funciona Raspberry Pi*

La Raspberry Pi es la placa de un ordenador simple compuesto por un SoC, CPU, memoria RAM, puertos de entrada y salida de audio y vídeo, conectividad de red, ranura

SD para almacenamiento, reloj, una toma para la alimentación, conexiones para periféricos de bajo nivel, reloj, prácticamente lo mismo la parte de atrás de la torre de un ordenador, porque la Raspberry es un ordenador. Eso sí, no tiene interruptor para encenderlo o apagarlo.

Para ponerlo en marcha se tiene que conectar periféricos de entrada y salida para poder interactuar como una pantalla, un ratón y un teclado y grabar un sistema operativo para Raspberry Pi en la tarjeta SD. Finalmente se conecta a la corriente y estaría listo para funcionar. (Xakata, 2018)

Tabla 22.

Características de la placa Raspberry Pi

	Características
Sistema en un chip	Broadcom BCM2711
CPU	Procesador de cuatro núcleos a 1,5 GHz con brazo Cortex-A72
GPU: VideoCore VI	VideoCore VI
Memoria	1/2/4GB LPDDR4 RAM
Conectividad	802.11ac Wi-Fi / Bluetooth 5.0, Gigabit Ethernet
Vídeo y sonido	2 x puertos micro-HDMI que admiten pantallas de 4K@60Hz a través de HDMI 2.0, puerto de pantalla MIPI DSI, puerto de cámara MIPI CSI, salida estéreo de 4 polos y puerto de vídeo compuesto.
Puertos	2 x USB 3.0, 2 x USB 2.0
Alimentación	5V/3A vía USB-C, 5V vía cabezal GPIO
Expansión: Cabezal GPIO de 40 pines	Cabezal GPIO de 40 pines
Sistema en un chip: Broadcom BCM2711	Sistema en un chip: Broadcom BCM2711
CPU	Procesador de cuatro núcleos a 1,5 GHz con brazo Cortex-A72

Fuente: Xakata.

3.3.1.7 *Phyton*

Es el lenguaje usado para el desarrollo del sistema de riego y poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo y menores inconvenientes como una menor velocidad. (Alvarez, DesarrolloWeb, 2015)



Figura 30. Software Phyton.

Fuente: Tomada de la página Blogthikbig.

3.3.1.8 Simulación de conexión

Se simula como se debe establecer la conexión entre cada uno de los elementos.

En el invernadero se encuentran los sensores y el sistema de riego escogido, estos a su vez van conectados hacia la placa de Raspberry Pi y dicho microcontrolador va conectado a una salida de 5V que es el encargado de suministrar energía para hacer funcional el prototipo.

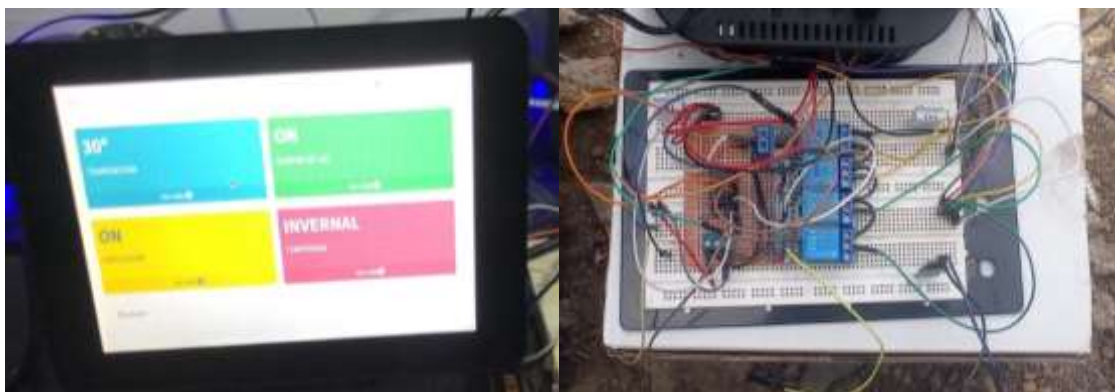


Figura 31. Simulación de conexión.

3.3.2 Factibilidad legal

Para cumplir con la factibilidad legal se toma en consideración la ley vigente en la Constitución de la República del Ecuador, de igual manera se encuentra respaldado por la Ley Orgánica de Educación superior y el Código Orgánico De La Economía Social De Los Conocimientos Creatividad E Innovación, basándose a las leyes y reglamentos que mantiene vigente el estado ecuatoriano, el presente proyecto no afecta de manera directa o indirecta dicho marco legal.

Por tal motivo, el prototipo del invernadero inteligente se acomoda a las leyes vigentes

ya descritas previamente en el Anexo 1. Culminando de esta manera, que no hay afectación o impedimento a terceras personas involucradas en el funcionamiento del dispositivo.

3.3.3 Factibilidad económica

El prototipo del invernadero inteligente cumple con criterios de factibilidad económica, dado que los dispositivos tanto hardware como software empleados en el desarrollo de este son de fácil acceso y costos módicos, utilizando tecnologías open source, permitiendo así que su realización sea posible y de fácil acceso económico a agricultores o personas que gustan de la agricultura, brindando de esta manera un aporte tecnológico, y crear nuevas alternativas para los invernaderos.

Tabla 23.

Recursos técnicos

Recurso técnico			
Descripción	Cantidad	C. Unitario	C. Total
Sensor DHT 22	2	\$6.00	\$ 12,00
Raspberry Pi 4	1	\$55,00	\$ 55,00
Sensor de humedad- YL-69	2	\$2,00	\$ 4,00
Fotoceldas	3	\$0,15	\$0,45
Kit cables variados (macho-macho, macho-hembra, hembra-hembra)	1	\$2,50	\$ 2,50
Ventilador	1	\$5,50	\$ 5,50
Mini bomba de agua	3	\$3,00	\$9,00
Gastos varios (estructura del invernadero)	1	\$ 30,00	\$ 30,00
Total			\$118,45

Tabla 24.

Recursos Humanos

Recurso humano			
Descripción	Cantidad	C. Unitario	C. Total
Mano de obra	1	\$60,00	\$60,00
Total			\$60,00

Tabla 25.

Presupuesto total de recursos empleados

Recurso técnico			
Descripción	Cantidad	C. Unitario	C. Total
Recursos técnicos	1	\$118,45	\$118,45
Recursos humanos	1	\$60	\$60,00
Total			\$178,45

3.3.4 Factibilidad operacional.

El prototipo está diseñado para beneficio de las personas que se dedican a la agricultura con el objetivo de ayudar a mejorar la calidad de sus plantaciones de una manera automática haciendo que el prototipo sea amigable y de fácil uso.

Cumple con los criterios de factibilidad operacional puesto que, este diseño se ha desarrollado en respuesta a la necesidad de los agricultores y personas que les gusta la agricultura sirviéndoles como instrumento de apoyo en sus actividades agrícolas.

La factibilidad operacional se ve orientada además por lo expuesto en el primer capítulo de este documento. Así también, al ser un diseño de prototipo su fácil uso lo convierte en un instrumento cómodo.

¿Existe aceptación de la propuesta por parte de la comunidad a la que va dirigido, muestra o beneficiarios directos?

¿La forma que se usará para verificar el correcto funcionamiento del prototipo es aceptada por los usuarios?

¿Los usuarios han participado en la planeación y en el desarrollo del proyecto?

3.4 Esquema general del proyecto.

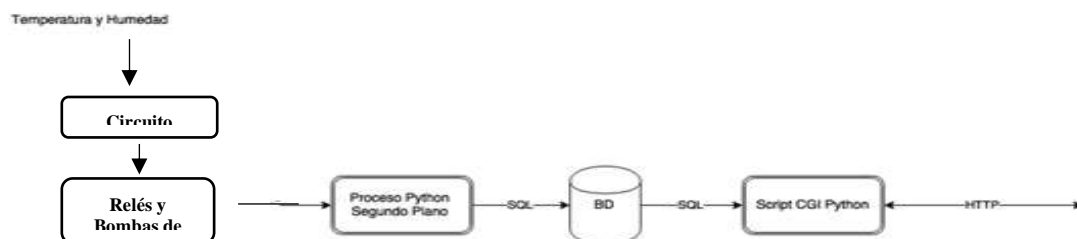


Figura 32. Estructura Backend

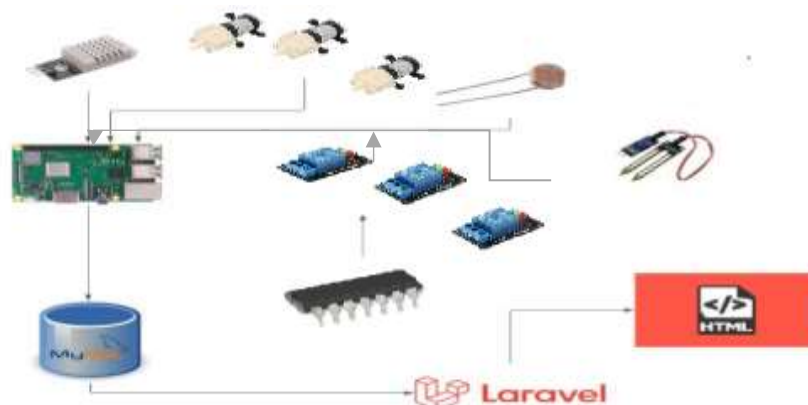


Figura 33. Estructura Backend del sistema capaz de controlar todos los sensores

3.5 Sistema de Riego

Se usa el esquema anterior (Figura 34) para realizar todas las conexiones. Se debe ubicar el sistema en un lugar cercano al invernadero que se va a regar y clavar las dos puntas del sensor de humedad en la tierra de la planta, cerca del tallo.

El sistema de riego automático con Raspberry Pi regará siempre que detecte una serie de condiciones ambientales. Por ejemplo, cuando visualiza el monitoreo de las variables involucradas en el invernadero, como son: temperatura, humedad relativa, humedad del suelo, intensidad de luz, se configuran las validaciones en la Raspberry Pi y en ese momento se activaría el motor para regar la planta.

En verano según el experto en jardinería Carlos Herrera dice:

“Es recomendable regar las plantas de noche, en el momento de menor calor, ya que hacerlo durante los días intensos de calor podría perjudicar más que beneficiar...”

Y en invierno también sugiere el experto que no es recomendable regar las plantas sin sol, porque baja demasiado la temperatura. El agua se congela y, al cristalizarse las raíces, se rompe la estructura lo que vuelve más vulnerables a las plantas.”

A continuación, se realiza las conexiones para la mini bomba y los contenedores que se van a utilizar para aplicar el riego de fertirrigación por goteo en el cual se mezclarán los macronutrientes necesarios según los valores que presentan el análisis del suelo de los cultivos, en un contenedor y esta mezcla se unirá al agua que se encuentra en otro contenedor que se ha destinado para el riego. Se hacen pruebas previas para saber cuánto se necesita de agua y nutrientes.

Será necesario recordar dar mantenimiento al equipo cada cierto tiempo para que funcione de manera correcta.



Figura 34. Sistema por Goteo.

3.6 Servidor XAMPP

Para el backend es necesario el servidor como base de simulación en XAMPP, lo primero sería la instalación como se observa en el anexo 9, partiendo de la instalación se levanta el servidor de MySQL como se observa en la figura 18.

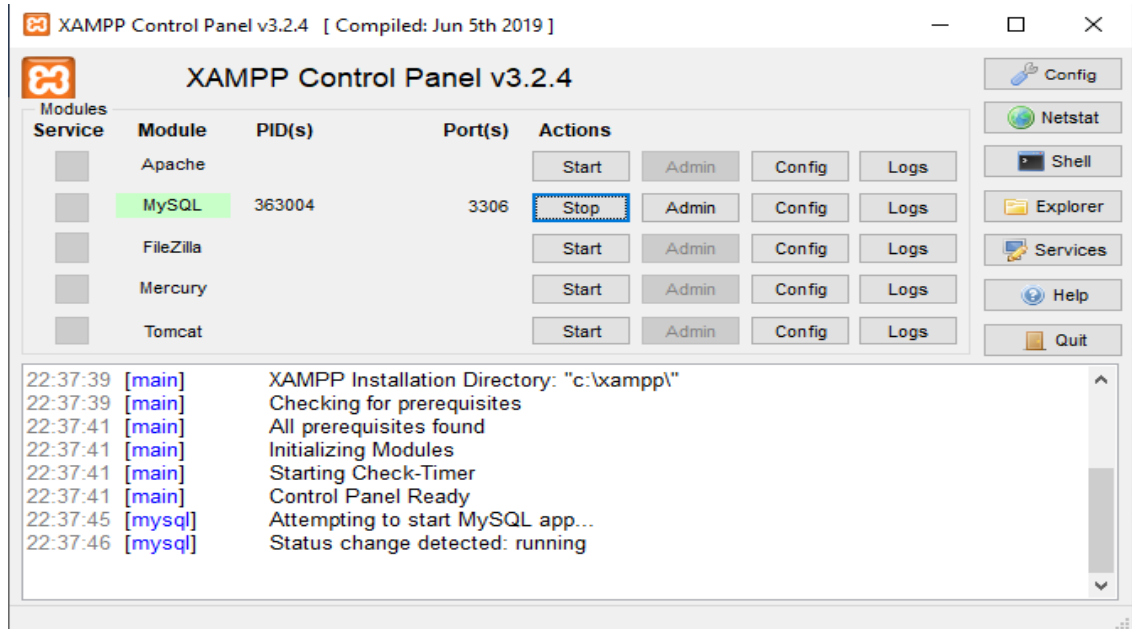


Figura 35. Levantamiento de Servidor MySQL

Después de haber hecho el levantamiento del servidor MySQL se inicia la creación de la base de datos con el nombre “invernadero” como se observa en la figura 19.

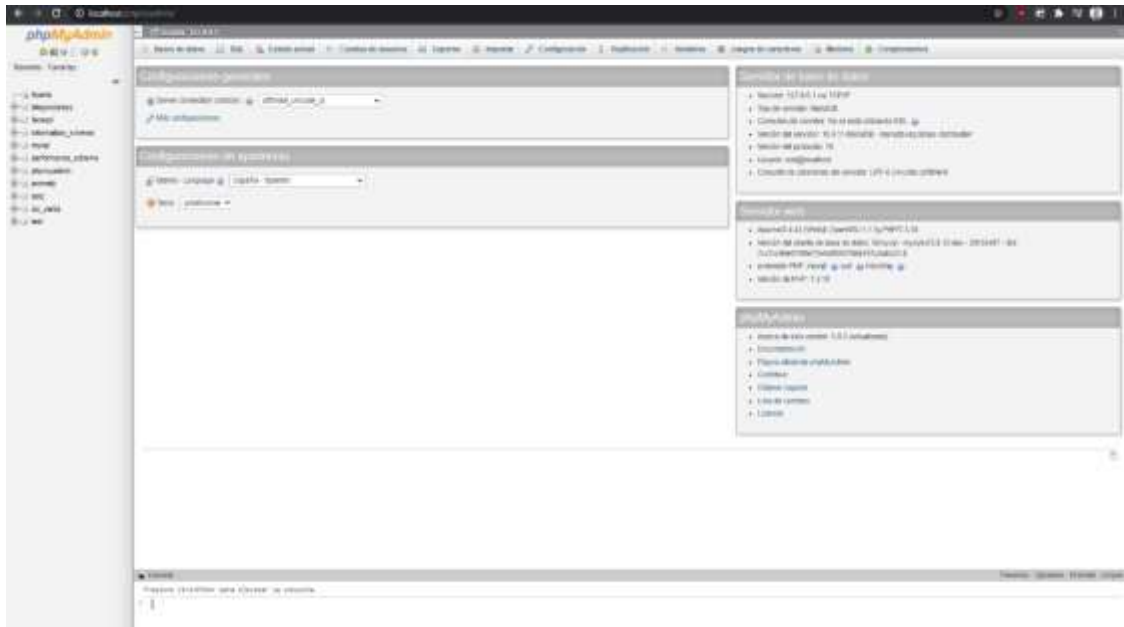


Figura 36. Levantamiento de la base de datos en MySQL

3.7 Estructura Base de Datos

Para el diseño de aplicación web la estructura de datos está definida con el control único de los datos principales del sistema que fueron receptados por los sensores como irá el detallado a continuación del modelo en MySQL.

3.7.1 Detalle de la Base de datos

Tabla 26.

Descripción de la Base de datos

Parámetro	Datos
IDE de la Base de datos	dbForge Studio
Dominio	127.0.0.1
Puerto	3400

3.7.2 Modelo de la Base de datos

Según lo mencionado con anterioridad al usar una base de datos el modelo lógico se establecería desde un IDE y gestor de base de datos denominados “dbForge Studio” y su uso es intuitivo para observar el diagrama de las tablas en la figura 22.

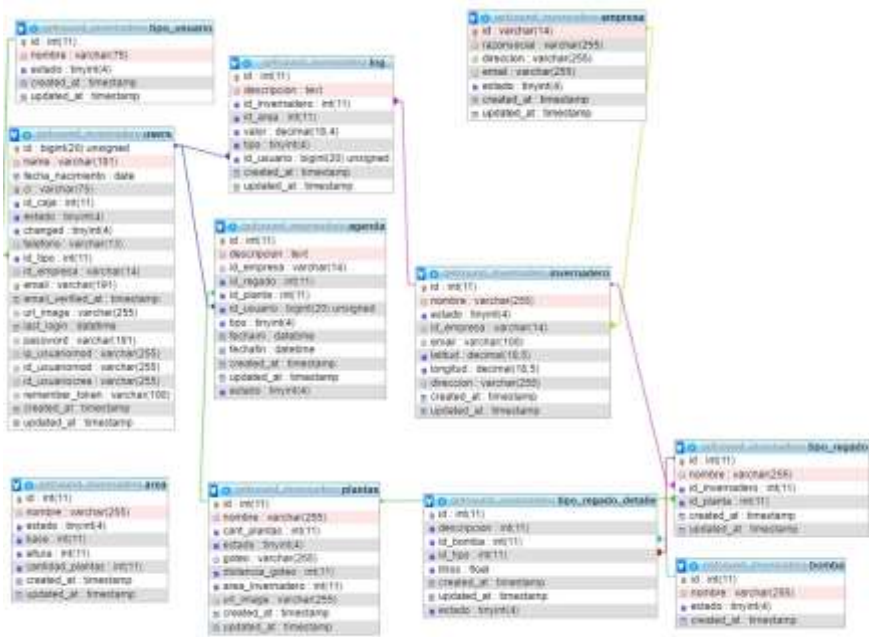


Figura 37. Modelo de Base de Datos en dbForge Studio.

3.7.3 Información de ambiente de Programación

Tabla 27.

Estructura del ambiente de Programación

Parámetro	Datos
Lenguaje de Programación	JavaScript. Python y SQL
Servidor Web	XAMPP
Puerto	8088, 3000
Framework	Laravel
JavaScript	Integrado

Configuración del sistema Web con la base de datos

En la aplicación de laravel configuramos el env con la configuración de la base de datos



Figura 38 Configuración del sistema web con la base de datos.

Luego limpiamos la cache de los archivos para que Laravel reconozca la base de datos.

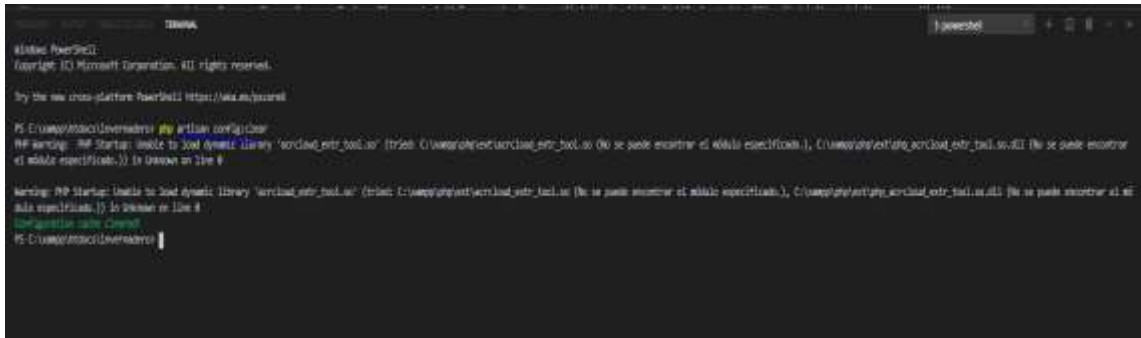


Figura 39. Limpieza de Cache.

3.8 PROCEDIMIENTO

Para comenzar a realizar las pruebas para la verificación y funcionamiento, se utilizó un protoboard para establecer las conexiones físicas de los diferentes sensores, Raspberry y las bombas de agua.

Para la conexión del sensor YL-69 que será el encargado de monitorear la humedad en el suelo del invernadero debemos conectarlo a un circuito integrado CD 4069 para la inversión del voltaje ya que este sensor tiene una medida análoga.

El sensor DTH22 medirá la temperatura y comiencen a registrar el estado en el que se encuentre el invernadero.

Luego se conectan las bombas de agua con los sensores a la Raspberry Pi para comprobar el funcionamiento físico de los componentes, utilizando el software (Python) para realizar la prueba de verificación del código en el programa.

En el código se agrega varias variables para los componentes que se están utilizando como los sensores para que detecten las condiciones en la que está el suelo y el ambiente del invernadero y las bombas que incluyendo la fórmula para medir el caudal que se van a ocupar las bombas para sistema de riego en los cultivos.

Se procede a establecer los pines de entrada y salida para mostrar por pantalla las lecturas dadas por el circuito.

A continuación, se accederá a la página web para agendar un riego a una hora determinada para observar el completo funcionamiento del prototipo.

Se acopla el circuito al invernadero y ser sometido a pruebas físicas con muestras.

3.8.1 Prueba de funcionalidad

En esta parte se va a demostrar de manera detallada cada una de las pruebas de funcionalidad, comprobando el correcto funcionamiento del proyecto y de los componentes que se están usando.

3.8.2 Aplicación Web

3.8.2.1 Inicio de sesión

Al iniciar sesión en el sistema se podrá visualizar las mediciones realizadas por los sensores, se pudo observar el código del login en el anexo 3.

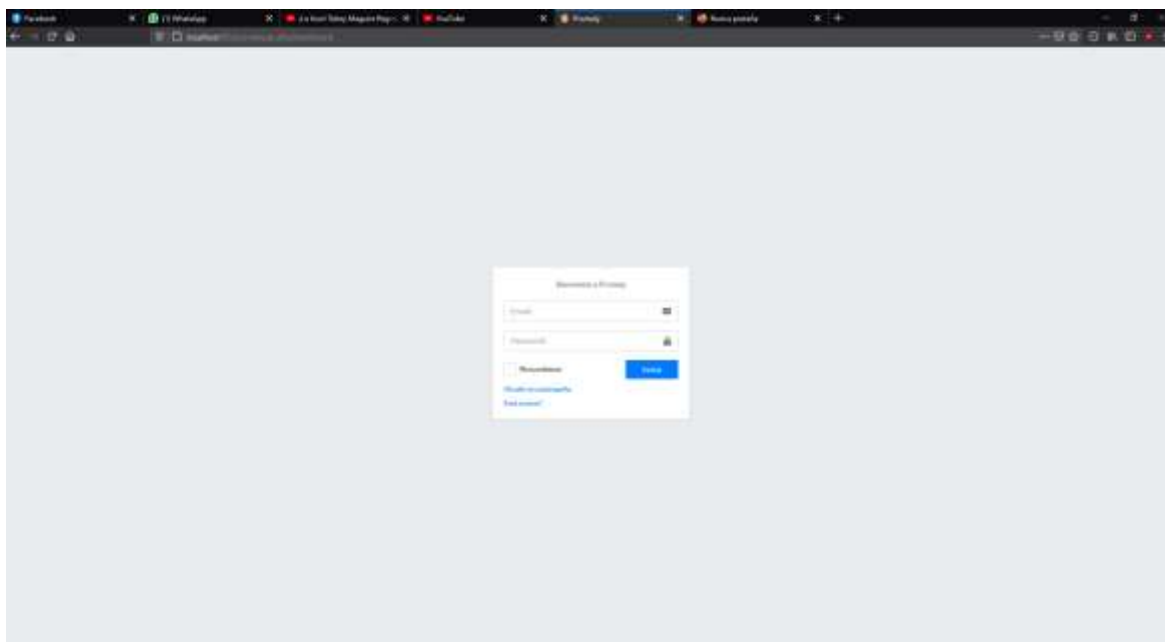


Figura 40 Inicio de sesión para el Invernadero Inteligente.

3.8.2.2 Agregar Usuarios

En este proyecto le permitirá la utilización de los módulos formados en el sistema, los campos que tiene son: los datos personales del usuario, contraseña, email y una imagen para su perfil. Se puede mirar lo comentado la figura 29.

Figura 41. Formulario de registro de usuarios.

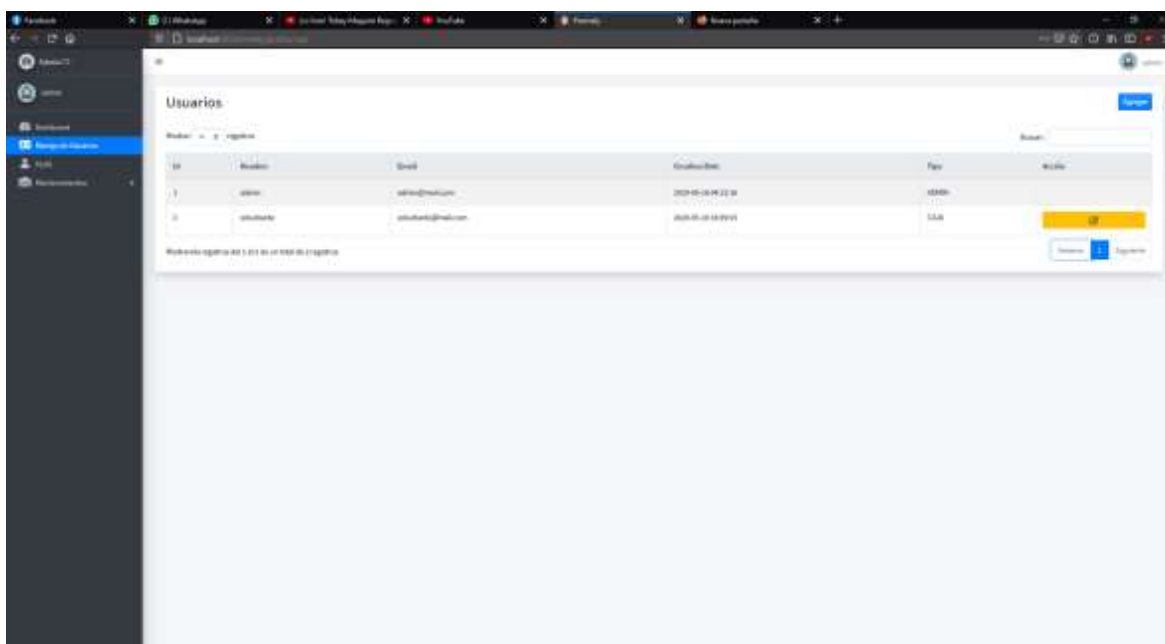


Figura 42. Vista y registro de los perfiles de los usuarios en el sistema.

3.8.2.3 Dashboard

En este módulo es esencial ya que aquí se encontrará los datos de las mediciones tomadas por los sensores.

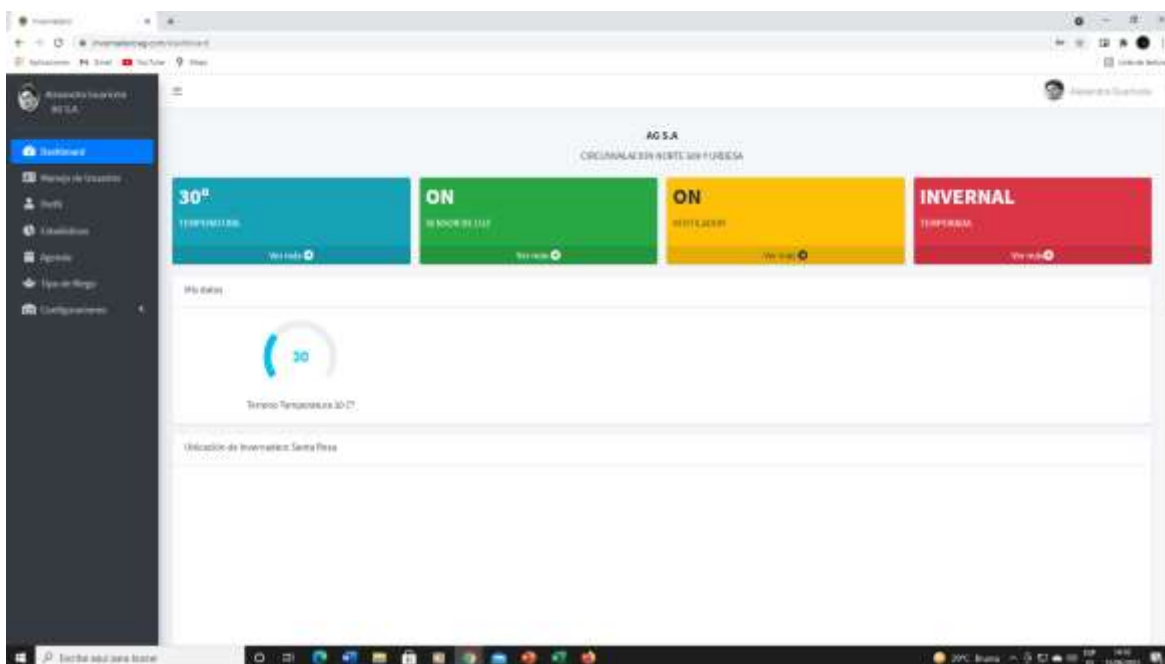


Figura 43. Dashboard del sistema.

#	Observación	Usuario	Fecha	Temperatura	Unidad
1	Rega	Alexandra Guzmán		20.00	°C
2	Rega	Alexandra Guzmán	2011-02-08 (05:17:26)	21.00	°C
3	Rega	Alexandra Guzmán	2011-02-08 (06:48:21)	24.00	°C
4	Rega	Alexandra Guzmán	2011-02-08 (07:38:45)	25.00	°C
5	Rega	Alexandra Guzmán	2011-02-08 (11:06:05)	26.00	°C
6	Rega	Alexandra Guzmán	2011-02-08 (17:20:04)	20.00	°C
7	Rega	Alexandra Guzmán	2011-02-08 (17:31:00)	21.00	°C

#	Observación	Usuario	Fecha	Estado
1	Inicio de la rega	Alexandra Guzmán	2011-02-08 (11:05:00)	ACTIVO
2	Procesamiento datos del invernadero	Alexandra Guzmán	2011-02-08 (12:00:04)	ACTIVO
3	Inicio de la rega	Alexandra Guzmán	2011-02-08 (13:00:00)	ACTIVO
4	Procesamiento datos del invernadero	Alexandra Guzmán	2011-02-08 (14:00:04)	ACTIVO
5	Inicio de la rega	Alexandra Guzmán	2011-02-08 (17:00:00)	ACTIVO
6	Procesamiento datos del invernadero	Alexandra Guzmán	2011-02-08 (17:00:04)	ACTIVO
7	Inicio de la rega	Alexandra Guzmán	2011-02-08 (17:30:00)	ACTIVO

Figura 44. Registro de las mediciones tomadas por los sensores.

3.8.3 Prueba

Escenario 1

Elección del terreno para la construcción del invernadero



Figura 45. Elección del terreno para el invernadero.

Se comienza con la construcción del invernadero tipo túnel.

En el invernadero se prepararon las camas de acuerdo con la distancia de siembra y separación de laterales formando un total de 18 unidades experimentales (fresas y ajos) y 2 camas.

Las características del área experimental son las siguientes: -

Tamaño del invernadero = 3 m², -

Área a utilizar = 1.50 m², -

Área útil = 1.50 m², -

Cantidad de plantas = 18 u, -

Cantidad de plantas por cama = 6 u y 12 u-

Cantidad de camas = 2 u, -

Cantidad de lisímetros = 48 u, -

Distancia entre líneas de riego = 0,50 cm; y, -

Separación entre goteros = 0,10 cm.



Figura 46. Construcción del invernadero.

Escenario 2

Para realizar las pruebas del buen funcionamiento se realizó de la toma de la temperatura de humedad del suelo con el sensor YL-69 y la temperatura del ambiente con el sensor DTH22.

Constatamos que el sensor funciona de manera correcta observando sus registros en la página web.

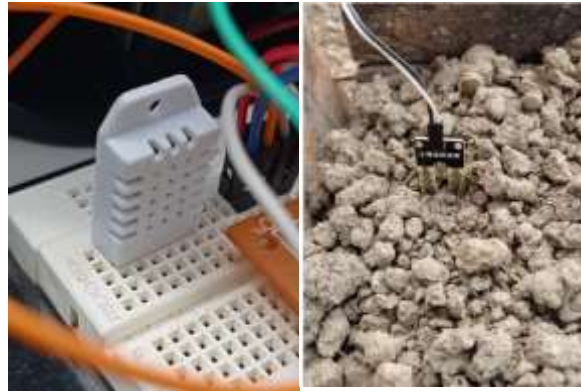


Figura 47. Prueba del sensor YL-69 y DTH22.

Fecha	Temperatura	Unidad
	26.00	C°
2021-03-06 10:57:20	21.00	C°
2021-03-06 10:58:23	24.00	C°
2021-03-06 10:58:41	20.00	C°
2021-03-06 17:36:26	30.00	C°
2021-03-06 17:36:44	26.00	C°
2021-03-06 17:37:02	25.00	C°

Figura 48. Registro del sensor YL-69 en la pag web.

Escenario 3

Se ejecutan unas pruebas básicas para el sistema de riego.

Se realiza un agendamiento para que el sistema de riego se active en un tiempo determinado, en este caso el área de las fresas se activara durante 30 min según los cálculos obtenidos en la duración de riego habiéndose tomado en cuenta el agotamiento de la humedad (SMD), el índice de aplicación y la uniformidad de distribución (DU) que se encuentra en el invernadero.

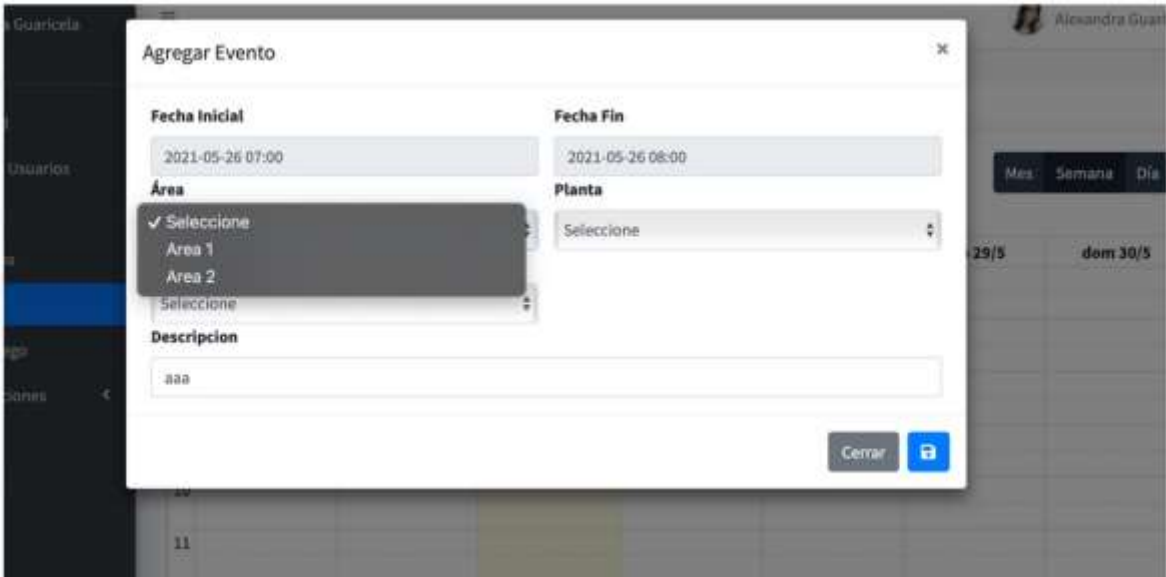


Figura 49. Agenda para programar el riego.

Constatando el funcionamiento de manera correcta observando sus registros en la página web.



Figura 50. Conexión del sistema de Riego.

Log				
#	Observación	Usuario	Fecha	Estado
1	Revisando el riego	Alexandra Guarcata	2021-03-08 11:00:34	ACTIVO
2	Procesando datos del invernadero	Alexandra Guarcata	2021-03-08 11:00:34	ACTIVO
3	Revisando el riego	Alexandra Guarcata	2021-03-08 11:00:34	ACTIVO
4	Procesando datos del invernadero	Alexandra Guarcata	2021-03-08 11:00:34	ACTIVO
5	Revisando el riego	Alexandra Guarcata	2021-03-08 17:00:34	ACTIVO
6	Procesando datos del invernadero	Alexandra Guarcata	2021-03-08 17:00:34	ACTIVO
7	Revisando el riego	Alexandra Guarcata	2021-03-08 17:00:34	ACTIVO

Figura 51. Registro del sistema de riego en la pág web.

3.9 Conclusiones y Recomendaciones

3.9.1 Conclusiones

- Mediante el análisis de la problemática y de los requerimientos de las personas que a pesar de su falta de conocimiento en tecnología buscan nuevas opciones que faciliten su trabajo, por tal motivo se desarrolló un dispositivo que permite detectar las necesidades que tiene un invernadero en tiempo real.
- Se determinó que los sensores de temperatura y humedad son un elemento de bajo costo y gran utilidad, con la ayuda de herramientas de código libre, estando al alcance de las personas de bajos recursos económicos, siendo idóneo para el desarrollo del prototipo del invernadero.
- Se explicó la metodología PMI (Project Management Institute), obteniendo de manera organizada y estructurada los pasos a seguir la cual describe fase a fase el desarrollo del prototipo del invernadero, solventando las necesidades de las personas en el proceso de desarrollo de sus cultivos.

3.9.2 Recomendaciones

Dentro de las recomendaciones y consideraciones de uso del dispositivo se detallan los siguientes puntos:

- Debido al uso de código libre en el diseño, se podría implementar mejoras en respecto a software mediante implementación de librerías que mejoren el proceso de control de las plantas.
- Se recomienda la posibilidad de añadir nuevas funciones que el usuario crea necesarias para el control el invernadero.
- Se recomienda que para el correcto funcionamiento del mecanismo de riego es necesario mantenerlo siempre conectado a una fuente de agua potable.

ANEXOS

Anexo 1

Constitución de la República del Ecuador

Título I

Elementos Constitutivos del Estado

Art. 3.- Son deberes primordiales del estado:

1.- Garantizar sin discriminación alguna el efectivo goce de los derechos establecidos en la constitución y en los instrumentos internacionales, en particular la educación, la salud, la alimentación, la seguridad social y el agua para sus habitantes.

Título II

Derechos

Capítulo II

Derechos del Buen Vivir

Sección III

Comunicación e información

Art 16.- Todas las personas, en forma individual o colectiva, tienen derecho a:

2.- El acceso universal a las tecnologías de información y comunicación

4.- El acceso y uso de todas las formas de comunicación visual, auditiva, sensorial y a otras que permitan la inclusión de personas con discapacidad.

Art 17.- El estado fomentará la pluralidad y la diversidad en la comunicación, y al efecto:

1.- Facilitará la creación y el fortalecimiento de medios de comunicación públicos, privados y comunitarios, así como el acceso universal a las tecnologías de información y comunicación en especial para las personas y colectividades que crezcan de dicho acceso o lo tengan de forma limitada.

Título VII

Régimen del Buen Vivir

Capítulo I

Sección VIII

Ciencia, Tecnología, Innovación y saberes ancestrales

Art. 385.- El sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida y la soberanía, tendrá como finalidad:

1.- Generar, adaptar y difundir conocimientos científicos y tecnológicos.

3.- Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del

buen vivir.

Art. 387.- Será responsabilidad del estado:

1.- Facilitar e impulsar la incorporación a la sociedad del conocimiento para alcanzar los objetivos del régimen de desarrollo.

2.- Promover la generación y producción de conocimiento, fomentar la investigación científica y tecnológica, y potenciar los saberes ancestrales, para así contribuir a la realización del buen vivir, al sumak kawsay.

Art. 388.- El estado destinará los recursos necesarios para la investigación científica, el desarrollo tecnológico, la innovación, la formación científica, la recuperación y desarrollo de saberes ancestrales y la difusión del conocimiento. Un porcentaje de estos recursos se destinará a financiar proyectos mediante fondos concursables. Las organizaciones que reciban fondos públicos estarán sujetas a la rendición de cuentas y al control estatal respectivo.

“Ley orgánica de educación superior”

Título I

Ámbito, objeto fines y principios del sistema de educación superior

Capítulo II

Fines de la Educación Superior

Art. 3.- Fines de la Educación Superior. - La educación superior de carácter humanista, cultural y científica constituye un derecho de las personas y un bien público social que, de conformidad con la Constitución de la República, responderá al interés público y no estará al servicio de interés individuales y corporativos.

Art. 8.- Serán fines de la Educación Superior. - La educación superior tendrá los siguientes fines:

- a) Aportar al desarrollo del pensamiento universal, al despliegue de la producción científica y a la promoción de las transferencias e innovaciones tecnológicas;
- b) Fortalecer en las y los estudiantes un espíritu reflexivo orientado al logro de la autonomía personal, en un marco de libertad de pensamiento y de pluralismo ideológico;
- c) Contribuir al conocimiento, preservación y enriquecimiento de los saberes ancestrales y de la cultura nacional;
- d) Formar académicos y profesionales responsables, con conciencia ética y solidaria, capaces de contribuir al desarrollo de las instituciones de la República, a la vigencia del orden democrático, y a estimular la participación social;
- e) Aportar con el cumplimiento de los objetivos del régimen de desarrollo previsto

en la Constitución y en el Plan Nacional de Desarrollo;

f) Fomentar y ejecutar programas de investigación de carácter científico, tecnológico y pedagógico que coadyuven al mejoramiento y protección del ambiente y promuevan el desarrollo sustentable nacional;

g) Constituir espacios para el fortalecimiento del Estado Constitucional, soberano, independiente, unitario, intercultural, plurinacional y laico; y,

h) Contribuir en el desarrollo local y nacional de manera permanente, a través del trabajo comunitario o extensión universitaria.

Título VI

Pertinencia

Capítulo I

Del principio de Pertinencia

Art. 107.- Principio de pertinencia.- El principio de pertinencia consiste en que la educación superior responda a las expectativas y necesidades de la sociedad, a la planificación nacional, y al régimen de desarrollo, a la prospectiva de desarrollo científico, humanístico y tecnológico mundial, y a la investigación y actividades de vinculación con la sociedad, a la demanda académica, a las necesidades de desarrollo local, regional y nacional, a la innovación y diversificación de profesiones y grados académicos, a las tendencias del mercado ocupacional local, regional y nacional, a las tendencias demográficas locales, provinciales y regionales; a la vinculación con la estructura productiva actual y potencial de la provincia y la región, y las políticas nacionales de ciencia y tecnología.

“Código orgánico de la economía social de los conocimientos creatividad e innovación”

Capítulo II

Acceso y soberanía del conocimiento en entornos digitales e informáticos

Art. 39.- Acceso universal, libre y seguro al conocimiento en entornos digitales.- El acceso al conocimiento libre y seguro en entornos digitales e informáticos, mediante las tecnologías de la información y comunicaciones desarrolladas en plataformas compatibles entre sí; así como el despliegue en infraestructura de telecomunicaciones, el desarrollo de contenidos y aplicaciones digitales y la apropiación de tecnologías, constituyen un elemento transversal de la economía social de los conocimientos, la creatividad y la innovación y es indispensable para lograr la satisfacción de necesidades y el efectivo goce de derechos. El acceso universal, libre y seguro al conocimiento en entornos digitales es un derecho de las

y los ciudadanos.

Capítulo III **De los derechos de autor**

Sección II

Objeto

Art. 104.- Obras susceptibles de protección. - La protección reconocida por el presente Título recae sobre las obras literarias, artísticas y científicas, que sean originales y que puedan reproducirse o divulgarse por cualquier forma o medio conocido o por conocerse.

Las obras susceptibles de protección comprenden, entre otras, las siguientes:

- 7.- Proyectos, planos, maquetas y diseños de obras arquitectónicas y de ingeniería.
- 12.- Software.

Decreto ejecutivo N° 10014

Del software libre

Art. 2.- Se entiende por Software Libre, a los programas de computación que se pueden utilizar y distribuir sin restricción alguna, que permitan su acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas.

- Estos programas de computación tienen las siguientes libertades:
- Utilización del programa con cualquier propósito de uso común.
- Distribución de copias sin restricción alguna.
- Estudio y modificación del programa (requisito: código fuente disponible).
- Publicación del programa mejorado (Requisitos: código fuente disponible).

Art. 3.- Las entidades de la Administración Pública previa la instalación del software libre en sus equipos, deberán verificar la existencia de capacidad técnica que brinde el soporte necesario para el uso de este tipo de software.

Art. 4.- Se faculta la utilización de software propietario (no libre) únicamente cuando no exista una solución de Software Libre que supla las necesidades requeridas, o cuando esté en riesgo la seguridad nacional, o cuando el proyecto informático se encuentre en un punto de no retorno.

Para efectos de este decreto se comprende como seguridad nacional, las garantías para la supervivencia de la colectividad y la defensa de patrimonio nacional.

Para efectos de este decreto se entiende por un punto de no retorno, cuando el sistema o proyecto informático se encuentre en cualquiera de estas condiciones:

- Sistemas en producción funcionando satisfactoriamente y que en un análisis de costo beneficio muestre que no es razonable ni conveniente una migración a Software Libre.
- Proyecto en estado de desarrollo y que un análisis de costo- beneficio muestre que no es conveniente modificar el proyecto y utilizar Software Libre.

Periódicamente se evaluarán los sistemas informáticos que utilizan software propietario con la finalidad de migrarlos a Software Libre.

Ley de Propiedad Intelectual

Sección V

Disposiciones especiales sobre ciertas obras

Párrafo primero

De los programas del ordenador

Art. 28.- Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporadas en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por maquina (código objeto), ya sean por programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura secuencian y organización del programa.

Art. 29.- Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Dicho titular esta además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación.

El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo.

Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor.

Anexo 2

Instalación de paquetes de servidores XAMPP

1. Se comienza con la instalación de XAMPP mediante la página oficial.



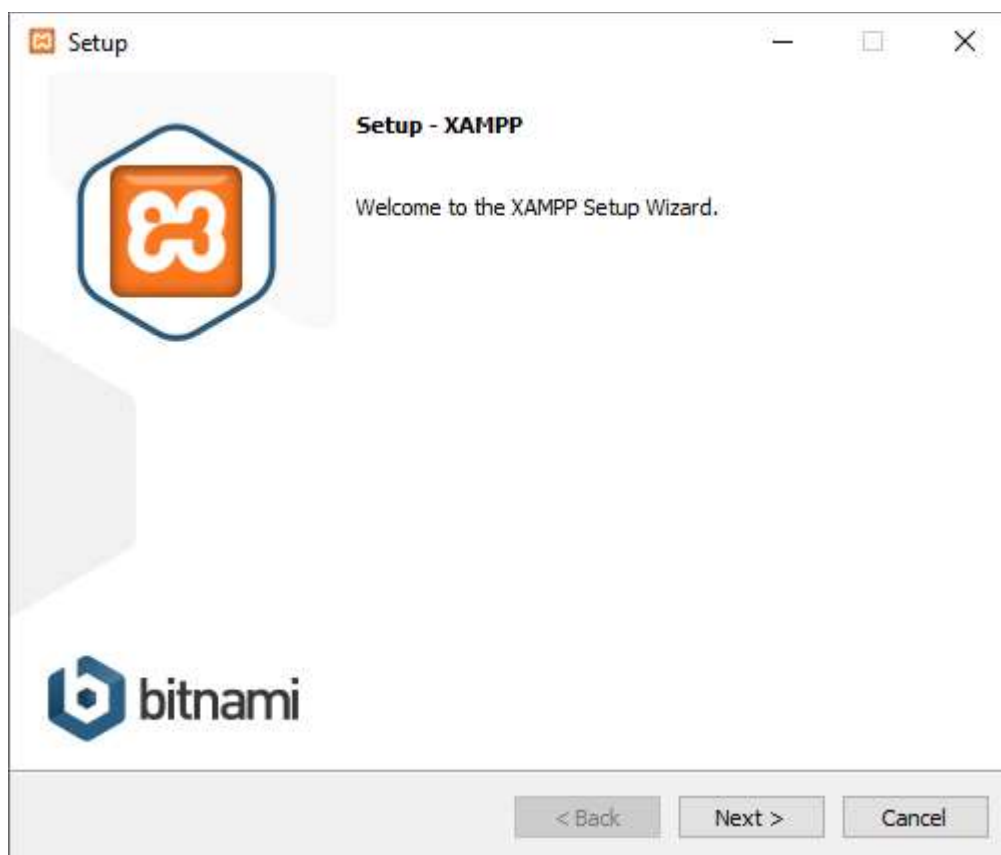
2. Se le da al botón de descargar según el sistema operativo que se esté usando.



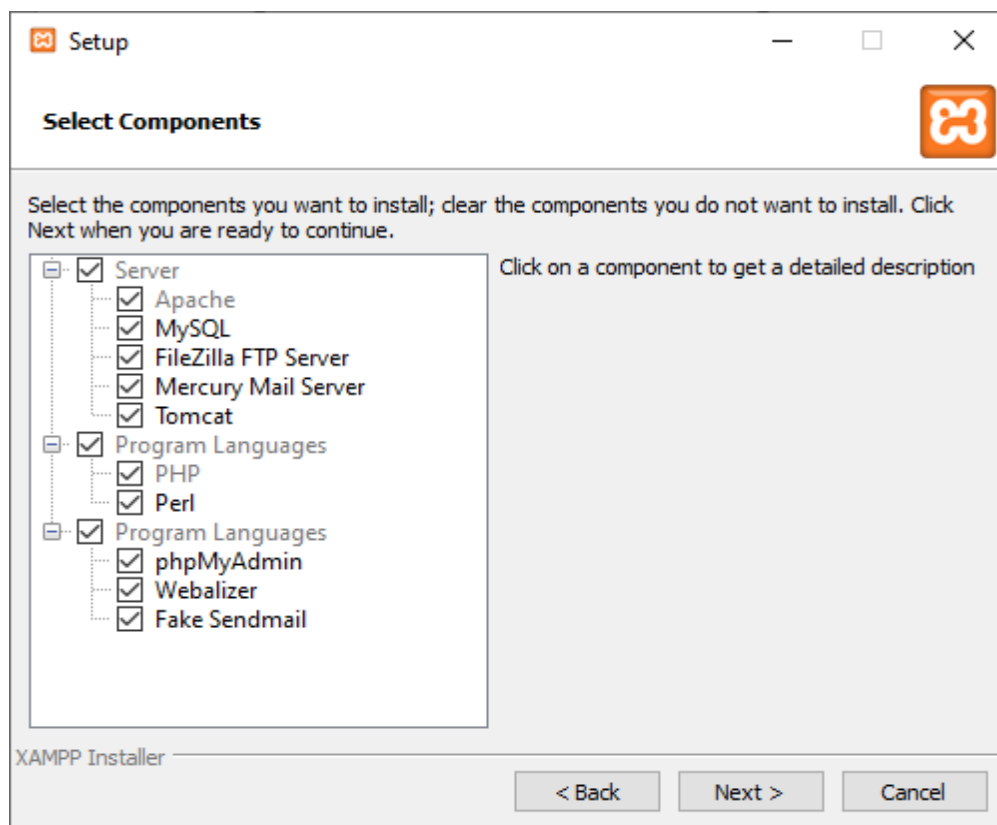
3. Luego del click guardar el archivo por preferencia en nuestro equipo.



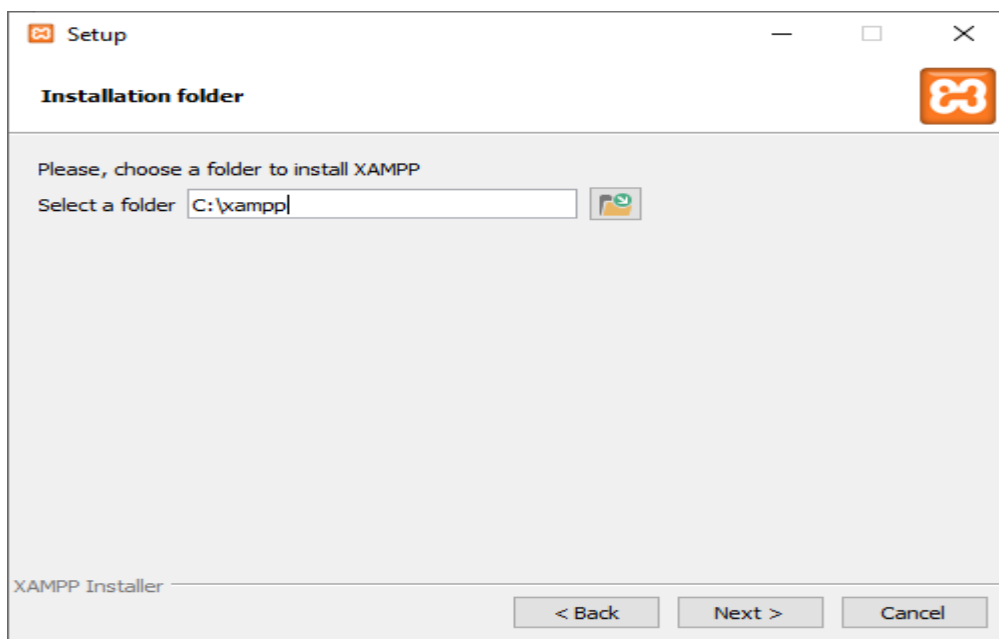
4. Se comienza con la instalación (En este caso es Windows 10)



5. Se selecciona los componentes a utilizar.



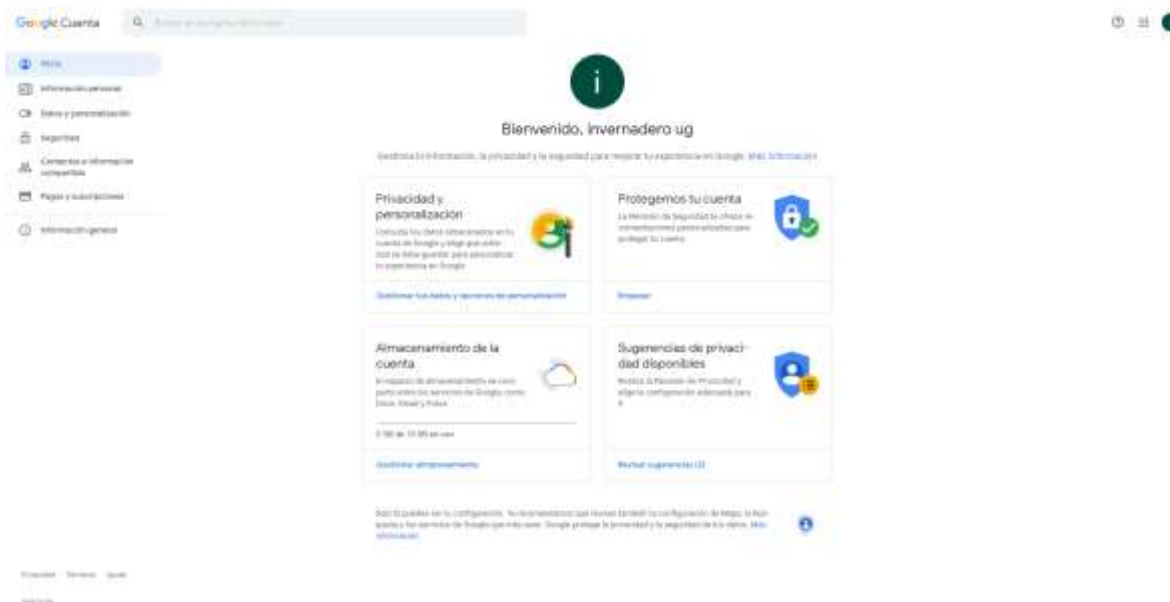
6. Selecciona la carpeta donde se guarde y listo.



Anexo 3

Instalación de Google Maps API

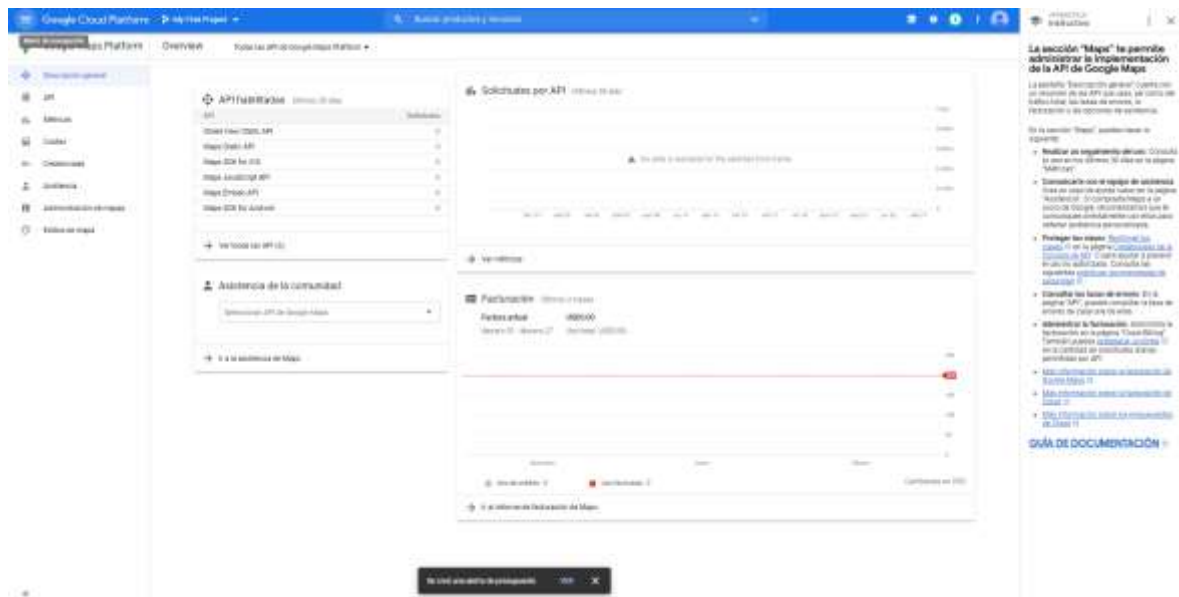
1. Creamos la cuenta de Google



2. Ingresamos a esta url: <https://console.cloud.google.com/google/maps-apis/welcome>



6. En Google cloud plataforma da un dashboard de analíticas de requisitos del api



7. Configuramos el acceso para url con http como localhost de la raspberry



8. Incluir el api token al script del código de inicio del sistema para la ubicación y se muestre el mapa

```

<div class="card">
  <div class="card-header">
    <h4 class="title d-d-inline">MI ubicación</h4>
  </div>
  <div class="card-body">
    <div id="map"></div>
  </div>
</div>
<div class="card">
</div>

<script>
  <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDK28E2OV9bcrdCumm_Eo8W2F2M0r0c&callback=initMap">
  </script>
  <script>
    <script>
      <script>
        function initMap() {
          const map = new google.maps.Map(document.getElementById("map"), {
            zoom: 10,
            center: { lat: -2.149393, lng: -79.921118 },
          });
          const image =
            "https://iib.co/2pwwbW";
          const beachMarker = new google.maps.Marker({
            position: { lat: -2.149393, lng: -79.921118 },
            map,
            icon: image,
          });
        }
      </script>
    </script>
  </script>
</div>
</div>

```

9. Ingresar a la url <https://es.imgbb.com/> para subir la imagen que usara en el marcador del mapa



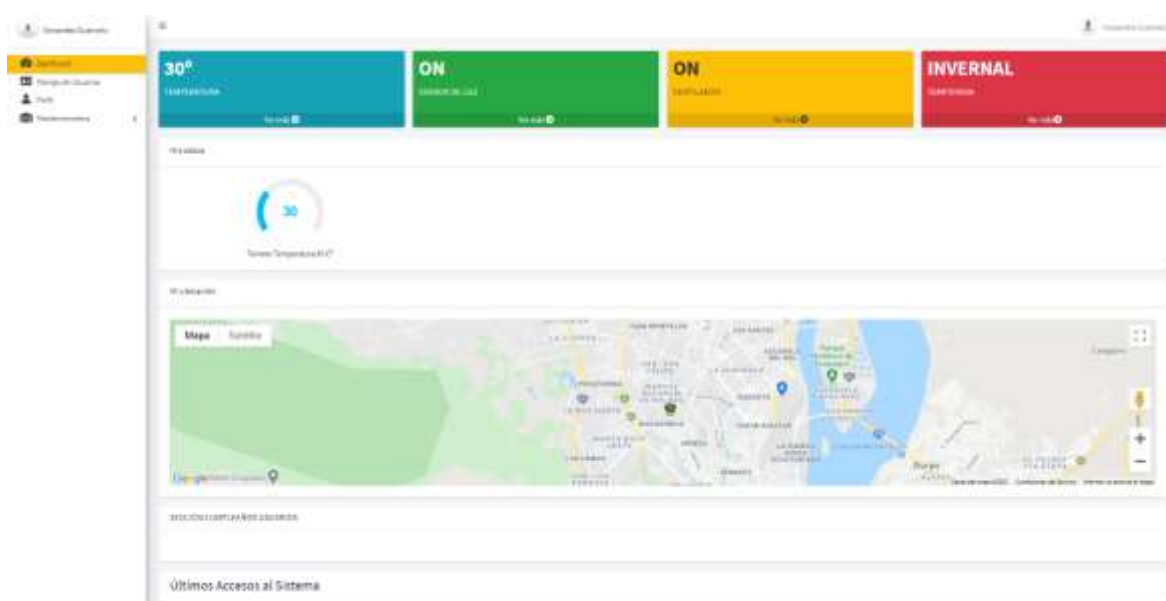
10. La página retorna una url de la imagen del repositorio online y se coloca en el marcador.

```

16 }
17
18 function initMap() {
19   const map = new google.maps.Map(document.getElementById("map"), [
20     zoom: 13,
21     center: { lat: -2.149393, lng: -79.921118 },
22   ]);
23   var icon = {
24     url: "https://i.ibb.co/VChmW2K/plant-1.png", // url
25     scaledSize: new google.maps.Size(25, 25), // scaled size
26     origin: new google.maps.Point(0,0), // origin
27     anchor: new google.maps.Point(0, 0) // anchor
28   };
29   const image =
30     "https://i.ibb.co/VChmW2K/plant-1.png";
31   const beachMarker = new google.maps.Marker({
32     position: { lat: -2.149393, lng: -79.921118},
33     map,
34     icon: icon,
35   });
36 }
37 </script>
38 @endsection

```

11. Se carga la imagen de la ubicación del invernadero



12. Se procede a configurar las unidades en el sistema

Agregar Tipo de Dato

Descripción:

Unidades:

[Guardar](#)

13. Se observa las unidades configuradas del sistema

Mis datos

30°

ON

ON

INVERNAL

Mapa

Datos de configuración

#	Descripción	Estado	Fecha	Responsable	Unidad
1	Temperatura externa	Activo	08/08/2018 10:00:00	Admin	°C

Anexo 4

Código del login

```

@extends('adminlte::master')

@section('adminlte_css_pre')
    <link rel="stylesheet" href="{{ asset('vendor/icheck-bootstrap/icheck-
bootstrap.min.css') }}">
@stop

@section('adminlte_css')
    @stack('css')
    @yield('css')
@stop

@section('classes_body', 'login-page')

@php( $login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login')
)
@php( $register_url = View::getSection('register_url') ?? config('adminlte.register_url',
'register') )
@php( $password_reset_url = View::getSection('password_reset_url') ?? config('adminl
te.password_reset_url', 'password/reset') )
@php( $dashboard_url = View::getSection('dashboard_url') ?? config('adminlte.dashbo
ard_url', 'home') )

@if (config('adminlte.use_route_url', false))
    @php( $login_url = $login_url ? route($login_url) : " )
    @php( $register_url = $register_url ? route($register_url) : " )
    @php( $password_reset_url = $password_reset_url ? route($password_reset_url) : " )
    @php( $dashboard_url = $dashboard_url ? route($dashboard_url) : " )
@else
    @php( $login_url = $login_url ? url($login_url) : " )
    @php( $register_url = $register_url ? url($register_url) : " )

```

```

@php( $password_reset_url = $password_reset_url ? url($password_reset_url) : " )
@php( $dashboard_url = $dashboard_url ? url($dashboard_url) : " )
@endif

@section('body')
<div class="login-box">
    <!--
    <div class="login-logo">
        <a href="{{ $dashboard_url }}">{!! config('adminlte.logo', '<b>Admin</b>LTE
E') !!}</a>
    </div>--->
    <div class="card">
        <div class="card-body login-card-body">
            <p class="login-box-msg">Bienvenido a Promely</p>
            <form action="{{ $login_url }}" method="post">
                {{ csrf_field() }}
                <div class="input-group mb-3">
                    <input type="email" name="email" class="form-control {{ $errors-
>has('email') ? 'is-
invalid' : '' }}" value="{{ old('email') }}" placeholder="{{ __('adminlte::adminlte.email'
) }}" autofocus>
                    <div class="input-group-append">
                        <div class="input-group-text">
                            <span class="fas fa-envelope"></span>
                        </div>
                    </div>
                    @if ($errors->has('email'))
                        <div class="invalid-feedback">
                            {{ $errors->first('email') }}
                        </div>
                    @endif
                </div>
                <div class="input-group mb-3">

```

```

        <input type="password" name="password" class="form-
control {{ $errors->has('password') ? 'is-
invalid' : '' }}" placeholder="{{ __('adminlte::adminlte.password') }}">
        <div class="input-group-append">
            <div class="input-group-text">
                <span class="fas fa-lock"></span>
            </div>
        </div>
        @if ($errors->has('password'))
            <div class="invalid-feedback">
                {{ $errors->first('password') }}
            </div>
        @endif
    </div>
    <div class="row">
        <div class="col-8">
            <div class="icheck-primary">
                <input type="checkbox" name="remember" id="remember">
                <label for="remember">Recuerdame</label>
            </div>
        </div>
        <div class="col-4">
            <button type="submit" class="btn btn-primary btn-block btn-flat">
                Iniciar
            </button>
        </div>
    </div>
</form>
@if ($password_reset_url)
    <p class="mt-2 mb-1">
        <a href="{{ $password_reset_url }}">
            {{ __('Olvidé mi contraseña') }}
        </a>
    </p>

```

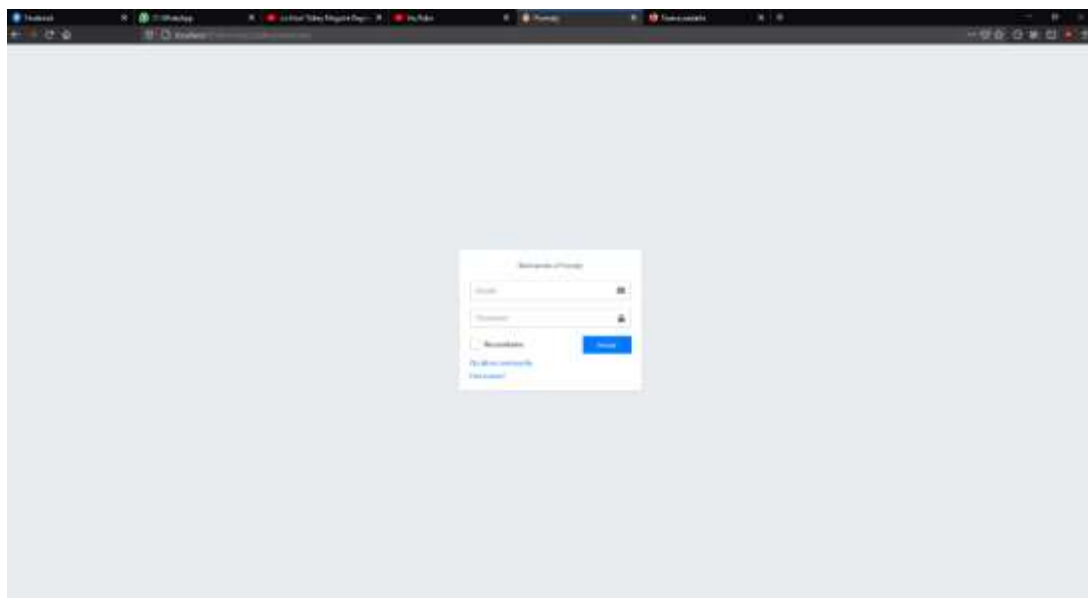
```
@endif

@if ($register_url)
    <p class="mb-0">
        <a href="{{ $register_url }}">
            Eres nuevo?
        </a>
    </p>
@endif

</div>
</div>
</div>
@stop

@section('adminlte_js')
    <script src="{{ asset('vendor/adminlte/dist/js/adminlte.min.js') }}"></script>
    @stack('js')
    @yield('js')
@stop
```

Visual del login de laravel



Anexo 5

Código Registrarse

```

@extends('adminlte::master')

@section('adminlte_css')
    @stack('css')
    @yield('css')
@stop

@section('classes_body', 'register-page')

@php( $login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login')
)
@php( $register_url = View::getSection('register_url') ?? config('adminlte.register_url',
'register') )
@php( $dashboard_url = View::getSection('dashboard_url') ?? config('adminlte.dashbo
ard_url', 'home') )

@if (config('adminlte.use_route_url', false))
    @php( $login_url = $login_url ? route($login_url) : " )
    @php( $register_url = $register_url ? route($register_url) : " )
    @php( $dashboard_url = $dashboard_url ? route($dashboard_url) : " )
@else
    @php( $login_url = $login_url ? url($login_url) : " )
    @php( $register_url = $register_url ? url($register_url) : " )
    @php( $dashboard_url = $dashboard_url ? url($dashboard_url) : " )
@endif

@section('body')
    <div class="register-box">
        <div class="register-logo">

        </div>

```

```

<div class="card">
  <div class="card-body register-card-body">
    <p class="login-box-msg">Registro </p>
    <form method="post" action="{ { route('users.store') } }" autocomplete="off" i
d="formulario">
      @csrf
      @if (session('status'))
        <div class="alert alert-success alert-
dismissible fade show" role="alert">
          {{ session('status') }}
          <button type="button" class="close" data-dismiss="alert" aria-
label="Close">

            <span aria-hidden="true">&times;</span>
          </button>
        </div>
      @endif
      <div class="form-group{{ $errors->has('name') ? ' has-danger' : " } }">
        <label class="form-control-label" for="input-
name">{{ __( 'Nombre' ) }}</label>
        <input type="text" name="name" id="input-name" class="form-
control form-control-alternative{{ $errors->has('name') ? ' is-
invalid' : " } }" placeholder="{{ __( 'Nombre' ) }}" required autofocus>

        @if ($errors->has('name'))
          <span class="invalid-feedback" role="alert">
            <strong>{{ $errors->first('name') }}</strong>
          </span>
        @endif
      </div>
      <div class="form-group{{ $errors->has('name') ? ' has-danger' : " } }">
        <label class="form-control-
label" for="email">{{ __( 'Email' ) }}</label>

```

```

<input type="email" name="email" id="email" class="form-
control form-control-alternative{{ $errors->has('name') ? ' is-
invalid' : '' }}" placeholder="{{ __('Email') }}" required autofocus>

```

```

@if ($errors->has('email'))
    <span class="invalid-feedback" role="alert">
        <strong>{{ $errors->first('email') }}</strong>
    </span>
@endif
</div>
<div class="form-group{{ $errors->has('name') ? ' has-danger' : '' }}">
    <label class="form-control-
label" for="cedula">{{ __('Ci') }}</label>
    <input type="text" onKeyPress="if (event.keyCode < 45 || event.
keyCode > 57) event.returnValue = false;" name="cedula" id="cedula" class="form-
control form-control-alternative{{ $errors->has('ci') ? ' is-
invalid' : '' }}" placeholder="{{ __('Ci') }}" required autofocus>

```

```

@if ($errors->has('ci'))
    <span class="invalid-feedback" role="alert">
        <strong>{{ $errors->first('ci') }}</strong>
    </span>
@endif
</div>
<div class="form-group{{ $errors->has('name') ? ' has-danger' : '' }}">
    <label class="form-control-
label" for="cedula">{{ __('Número de Telefono') }}</label>
    <input type="text" onKeyPress="if (event.keyCode < 45 || event.
keyCode > 57) event.returnValue = false;" name="phone_number" id="phone_number"
class="form-control form-control-alternative{{ $errors->has('phone_number') ? ' is-
invalid' : '' }}" placeholder="{{ __('No Telefono') }}" required autofocus>

```

```

@if ($errors->has('ci'))
    <span class="invalid-feedback" role="alert">

```

```

        <strong>{{ $Errors->first('ci') }}</strong>
    </span>
    @endif
</div>
<div class="form-group{{ $Errors->has('name') ? ' has-danger' : '' }}">
    <label class="form-control-
label" for="f_nacimiento">{{ __('Fecha Nacimiento') }}</label>
    <input type="date" name="f_nacimiento" id="f_nacimiento" clas
s="form-control form-control-alternative{{ $Errors->has('f_nacimiento') ? ' is-
invalid' : '' }}" required autofocus>

    @if ($Errors->has('f_nacimiento'))
        <span class="invalid-feedback" role="alert">
            <strong>{{ $Errors->first('f_nacimiento') }}</strong>
        </span>
    @endif
</div>
<div class="form-group{{ $Errors->has('password') ? ' has-
danger' : '' }}">
    <label class="form-control-label" for="input-current-
password">{{ __('Password') }}</label>
    <input type="password" name="password" id="input-current-
password" class="form-control form-control-alternative{{ $Errors-
>has('password') ? ' is-
invalid' : '' }}" placeholder="{{ __('Actual Password') }}" value="" required>

    @if ($Errors->has('password'))
        <span class="invalid-feedback" role="alert">
            <strong>{{ $Errors->first('password') }}</strong>
        </span>
    @endif
</div>
<div class="col-md-12" style="text-align: center">
    <button type="submit" class="btn btn-success">Guardar</button>

```


Anexo 6

Codigo perfil

```
@extends('adminlte::page')
```

```
@section('content')
```

```
<div class="container-fluid">
```

```
<div class="row">
```

```
<div class="col-xl-4 order-xl-2 mb-5 mb-xl-0">
```

```
<div class="card">
```

```
<div class="row">
```

```
<div class="col-md-12 col-xs-12" style="text-align: center;">
```

```
<a href="#" class="col-md-12">
```

```

```

```
</a>
```

```
</div>
```

```
</div>
```

```
<div class="card-body pt-0 pt-md-4">
```

```
<div class="row">
```

```
<div class="col">
```

```
<div class="card-profile-stats d-flex justify-content-center mt-md-5">
```

```
<div>
```

```
&nbsp;
```

```
</div>
```

```
<div>
```

```
&nbsp;
```

```
</div>
```

```
<div>
```

```
&nbsp;
```

```
</div>
```

```
</div>
```

```
</div>
```

```

</div>
<div class="text-center">
  <h3>
    {{ auth()->user()->name }}<span class="font-weight-
light"></span>
    <form action="{{ route('profile_photo') }}" method="POST" enctype="multipart/form-data" id="upload_photo">
      {{ csrf_field() }}
      <input type="file" name="photo" class="form-control" required>
    </form>
    <button type="button" onclick="guardar_foto()" class="btn btn-danger animation-on-hover btn-sm btn-block mb-3">Subir imagen</button>
  </h3>
</div>
</div>
</div>
<div class="col-xl-8 order-xl-1">
  <div class="card">
    <div class="card-header bg-white border-0">
      <div class="row align-items-center">
        <h3 class="col-12 mb-0">{{ __( 'Editar Perfil' ) }}</h3>
      </div>
    </div>
    <div class="card-body">
      <form method="post" action="{{ route('profile.update') }}" autocomplete="off">
        @csrf
        @method('put')

        <h6 class="heading-small text-muted mb-4">{{ __( 'Información del Usuario' ) }}</h6>

        @if (session('status'))

```

```

        <div class="alert alert-success alert-
dismissible fade show" role="alert">
            {{ session('status') }}
            <button type="button" class="close" data-dismiss="alert" aria-
label="Close">

                <span aria-hidden="true">&times;</span>
            </button>
        </div>
    @endif

    <div class="pl-lg-4">
        <div class="form-group{{ $errors->has('name') ? ' has-
danger' : '' }}">
            <label class="form-control-label" for="input-
name">{{ __('Nombre') }}</label>
            <input type="text" name="name" id="input-name" class="form-
control form-control-alternative{{ $errors->has('name') ? ' is-
invalid' : '' }}" placeholder="{{ __('Nombre') }}" value="{{ old('name', auth()->user()-
>name) }}" required autofocus>

            @if ($errors->has('name'))
                <span class="invalid-feedback" role="alert">
                    <strong>{{ $errors->first('name') }}</strong>
                </span>
            @endif
        </div>
        <div class="form-group{{ $errors->has('email') ? ' has-
danger' : '' }}">
            <label class="form-control-label" for="input-
email">{{ __('Email') }}</label>
            <input type="email" name="email" id="input-
email" class="form-control form-control-alternative{{ $errors->has('email') ? ' is-
invalid' : '' }}" placeholder="{{ __('Email') }}" value="{{ old('email', auth()->user()-
>email) }}" required>

```



```

        @if ($errors->has('email'))
            <span class="invalid-feedback" role="alert">
                <strong>{{ $errors->first('email') }}</strong>
            </span>
        @endif
    </div>

    <div class="text-center">
        <button type="submit" class="btn btn-success mt-
4">{{ __('Guardar') }}</button>
    </div>
</div>
</form>
<hr class="my-4" />
<form method="post" action="{{ route('profile.password') }}" autocomp
lete="off">

    @csrf
    @method('put')

    <h6 class="heading-small text-muted mb-
4">{{ __('Password') }}</h6>

    @if (session('password_status'))
        <div class="alert alert-success alert-
dismissible fade show" role="alert">
            {{ session('password_status') }}
            <button type="button" class="close" data-dismiss="alert" aria-
label="Close">

                <span aria-hidden="true">&times;</span>
            </button>
        </div>
    @endif

```

```

<div class="pl-lg-4">
  <div class="form-group{{ $Errors->has('old_password') ? ' has-
danger' : '' }}">
    <label class="form-control-label" for="input-current-
password">{{ __('Actual Password') }}</label>
    <input type="password" name="old_password" id="input-
current-password" class="form-control form-control-alternative{{ $Errors-
>has('old_password') ? ' is-
invalid' : '' }}" placeholder="{{ __('Actual Password') }}" value="" required>

    @if ($Errors->has('old_password'))
      <span class="invalid-feedback" role="alert">
        <strong>{{ $Errors->first('old_password') }}</strong>
      </span>
    @endif
  </div>
  <div class="form-group{{ $Errors->has('password') ? ' has-
danger' : '' }}">
    <label class="form-control-label" for="input-
password">{{ __('Nuevo Password') }}</label>
    <input type="password" name="password" id="input-
password" class="form-control form-control-alternative{{ $Errors-
>has('password') ? ' is-
invalid' : '' }}" placeholder="{{ __('Nuevo Password') }}" value="" required>

    @if ($Errors->has('password'))
      <span class="invalid-feedback" role="alert">
        <strong>{{ $Errors->first('password') }}</strong>
      </span>
    @endif
  </div>
  <div class="form-group">
    <label class="form-control-label" for="input-password-
confirmation">{{ __('Confirmar Nuevo Password') }}</label>

```

```

        <input type="password" name="password_confirmation" id="input-
password-confirmation" class="form-control form-control-
alternative" placeholder="{{ __('Confirm Nuevo Password') }}" value="" required>
    </div>

```

```

        <div class="text-center">
            <button type="submit" class="btn btn-success mt-4">{{ __('Cambiar password') }}</button>
        </div>
    </div>
</form>
</div>
</div>
</div>
</div>

```

```

</div>

```

```

<script type="text/javascript">
    function guardar_foto(){
        document.getElementById('upload_photo').submit();
    }
</script>

```

```

@endsection

```

Visual página de perfil

Editar Perfil

Información del usuario

Nombre

Email

Guardar

Password


Actual Password

Nuevo Password

Confirmar Nuevo Password

Guardar password

Perfil



Alexandra

[Eliminar perfil](#) [Eliminar perfil](#)

[Logout](#)

Visual de administración de usuarios

Usuarios

[Nuevo](#) [Eliminar](#)

ID	Nombre	Email	Fecha de creación	País	Acción
1	alexa	alexa@company.com	2023-05-18 14:22:38	USA	Ver
2	alexa	alexa@company.com	2023-05-18 14:22:38	USA	Eliminar

Mostrando registros de 1 de 1 de un total de 1 registros

Anexo 7

Controlador de la Agenda y Bomba de agua

```

<?php

namespace App\Http\Controllers;

use App\TipoRegado;
use App\TipoRegadoDetalle;
use App\Bomba;
use App\Agenda;
use App\Plantas;
use Mail;
use App\Area;
use App\Invernadero;
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;

class AgendaController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    private function rol(){
        $rolUsuario = Auth::user()->id_tipo;

        if(in_array($rolUsuario, array(1, 2)) == false){
            return true;
        }
    }
    public function index()
    {

```

```

        if($this->rol()){
            return response()->view('errors.404');
        }
        $agenda= Agenda::where('estado','1')->get();

        return view('calendario',['agenda'=>$agenda]);
    }
    public function modal(Request $request)
    {
        if($this->rol()){
            return response()->view('errors.404');
        }
        $planta= Plantas::all();
        $tipo= TipoRegado::all();
        $area= Area::all();
        return
        view('modaldate',['fechaini'=>$request['init'],'fechafin'=>$request['end'],'area'=>$area,'descripcion'=>$request['title'],'planta'=>$planta,'tipo'=>$tipo]);
    }
    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $idusuario = Auth::user()->id;
        $empresa = Auth::user()->id_empresa;
        $fechaini= $request['fechaini'];
        $fechafin= $request['fechafin'];
        $tipo= $request['tipo'];
        //aun falta agregarle el id_regado con la validacion
        Agenda::create([
            'descripcion'=>$request['descripcion'],

```

```

        'fechaini'=>$fechaini,
        'fechafin'=>$fechafin,
        'id_usuario'=>$idusuario,
        'id_empresa'=>$empresa,
        'estado'=>'1',
        'id_planta'=>$request['planta'],
        'id_regado'=>$tipo,
        'tipo'=>$tipo
    );
    return redirect()->route('agenda');
}

public function delete(Request $request)
{
    $idusuario = Auth::user()->id;
    $empresa = Auth::user()->id_empresa;
    $fechaini= $request['fechaini'];
    $fechafin= $request['fechafin'];
    //dd($request->all());
    $tipo= $request['tipo'];
    $agenda= Agenda::find($request['id']);
    $agenda->estado=0;
    $agenda->save();
    return response()->json("ok");
}

public function index_tipo(){
    $tipo= TipoRegado::all();
    return view("tiporegado.index",["tipo"=>$tipo]);
}

public function create_tipo(){
    $bombas= Bomba::all();
    $invernadero= Invernadero::all();
    $plantas= Plantas::all();

```

```

        return
    view("tiporegado.create",["bombas"=>$bombas,"invernadero"=>$invernadero,"plantas"=
    >$plantas]);
    }
    public function edit_tipo($id){
        $tipo= TipoRegado::find($id);
        $detalles = TipoRegadoDetalle::where('id_tipo',$id)->get();
        $bombas= Bomba::all();
        $invernadero= Invernadero::all();
        $plantas= Plantas::all();
        return
    view("tiporegado.edit",["tipo"=>$tipo,"bombas"=>$bombas,"invernadero"=>$invernadero
    ,"plantas"=>$plantas,'detalles'=>$detalles]);
    }
    public function store_tipo(Request $request){
        $id_tipo=TipoRegado::insertGetId([
            'nombre'=>$request['nombre'],
            'id_invernadero'=>$request['invernadero'],
            'id_planta'=>$request['planta']
        ]);
        for($i=0; $i<count($request['bomba']); $i++){
            if($request['bomba'][$i]!=null){
                TipoRegadoDetalle::create([
                    'id_tipo'=>$id_tipo,
                    'descripcion'=>$request['litraje'][$i],
                    'litros'=>$request['litraje'][$i],
                    'estado'=>'1',
                    'id_bomba'=>$request['bomba'][$i],
                ]);
            }
        }
        return redirect()->route('tiporegado.index');
    }
}

```



```

public function update_tipo($id,Request $request){
    $id_tipo=TipoRegado::find($id);
    $id_tipo->update([
        'nombre'=>$request['nombre'],
        'id_invernadero'=>$request['invernadero'],
        'id_planta'=>$request['planta']
    ]);
    $detais= TipoRegadoDetalle::where('id_tipo',$id)->delete();
    for($i=0; $i<count($request['bomba']); $i++){
        if($request['bomba'][$i]!=null){
            TipoRegadoDetalle::create([
                'id_tipo'=>$id,
                'descripcion'=>$request['litraje'][$i],
                'litros'=>$request['litraje'][$i],
                'estado'=>'1',
                'id_bomba'=>$request['bomba'][$i],
            ]);
        }
    }
    return redirect()->route('tiporegado.index');
}

public function getType(Request $request){
    if(!is_null($request['planta'])){
        $detalle= TipoRegado::where('id_planta',$request['planta'])->first();
        if(!is_null($detalle)){
            return response()->json(["id"=>$detalle->id,'data'=>'ok']);
        }
    }
    return response()->json(['data'=>'error']);
}

public function sendEmail(Request $request){
    $subject = "Empiezo riego";
    $for = "anthony.chilanp@gmail.com";

```

```
Mail::send('email',$request->all(), function($msj) use($subject,$for){
    $msj->from("soporte@invernaderoag.com","Soporte Invernadero");
    $msj->subject($subject);
    $msj->to($for);
});
return response()->json("ok");
}

}
```

Anexo 8

Código en Python

```
#!/usr/bin/env python3

## SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT
import RPi.GPIO as GPIO
import time
import Adafruit_DHT
import random
import mysql.connector
import requests
from datetime import datetime

DHT_SENSOR = Adafruit_DHT.DHT22
DHT_PIN = 4
channel = 17 #bomba1
channel2= 27 #bomba2
channel3= 22 #bomba3
url = 'https://invernaderoag.com/inverna'
urlvalid='https://invernaderoag.com/validAgenda'

# GPIO setup
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel, GPIO.OUT)

def motor_on(pin):
    print "encendido"
    GPIO.output(pin, GPIO.HIGH) # Turn motor on
    turn=1
    myobj = {'bomba':turn,'data':'bomba'}
    x = requests.post(url, data = myobj)
    print(x.text)
```

```

    #print(turn)
def motor_off(pin):
    print "apagado"
    GPIO.output(pin, GPIO.LOW) # Turn motor off
    turn=0
    myobj = {'bomba':turn,'data':'bomba'}
    x = requests.post(url, data = myobj)
    print(x.text)
    #print(turn)
def dht22():
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
    if humidity is not None and temperature is not None :
        print("Temp={0:0.1f}*C Humidity={1:0.1f}% ".format(temperature, humidity))
        myobj = {'temperatura':temperature,'humedad': humidity,'data':'dht22'}
        x = requests.post(url, data = myobj)
        print(x.text)
        print(temperature)
    else:
        temperature= random.randrange(20,30);
        humidity= random.randrange(20,30);
        myobj = {'temperatura':temperature,'humedad': humidity,'data':'dht22'}
        x = requests.post(url, data = myobj)
        print(x.text)
        print(temperature)
def density():
    temperature= random.randrange(10,100);
    myobj = {'density':temperature,'data':'density'}
    x = requests.post(url, data = myobj)
    print(x.text)
def yl69():
    temperature= random.randrange(20,30);
    myobj = {'yl69':temperature,'data':'yl69'}
    x = requests.post(url, data = myobj)
    print(x.text)

```

```

def askServer(bomba):
    print("Preguntando al servidor ...")
    date= datetime.today().strftime('%Y/%m/%d %H:%M:%S')
    print("Fecha del momento")
    print(date)
    if(bomba is not None):
        print(" Bomba")
        myobj = {'date':date,'bomba':bomba}
        x = requests.post(urlvalid, data = myobj)
        print(x.text)
        print("respuesta server")
        fin= x.text
        print(fin)
        if(fin=="ok"):
            motor_on(channel)
        else:
            motor_off(channel)
    else:
        bomba=1
        myobj = {'date':date,'bomba':bomba}
        x = requests.post(urlvalid, data = myobj)
        print(x.text)
        print("respuesta server")
        fin= x.text
        print(fin)
        if(fin=="ok"):
            motor_on(channel)
        else:
            motor_off(channel)

if __name__ == '__main__':
    try:
        while True:

```

```
motor_off(channel)
time.sleep(2)
dht22()
#motor_on(channel)
askServer(1)
time.sleep(2)
density()
yl69()
time.sleep(10)

#GPIO.cleanup()
except KeyboardInterrupt:
    GPIO.cleanup()
```

Anexo 9

Acrónimos

AE= Agricultura empresarial

AFC = Agricultura Familiar Campesina

UPAs = Unidades de Producción Agrícolas

PMI = Instituto de gestión de proyectos

GND = Ground – Tierra

IDE= Integrated Development Environment - Ambiente de Desarrollo Integrado

GPIO = General Purpose Input Output – Salida de entrada de uso general

GNU = Sistema Operativo libre

PHP = Hypertext Preprocessor- Preprocesador de hipertexto

SBC = Controlador de Borde de Sesión

HTML = HyperText Markup Language -Lenguaje de Marcas de Hipertexto

CSS = Cascading Style Sheets- Hojas de estilo en cascada

GUI = Interfaz gráfica de usuario

My SQL = My Structured Query Language - Lenguaje de Consulta Estructurado

API = Interfaz de programación de aplicaciones

Ph = Potencial Hidrógeno

DC = Corriente continua

RH = Humedad relativa

Gr = Gramos

K Ω = Kiloohmios

M Ω = Megaohmios

VDC =Voltaje de corriente directa

CMOS = Complementary Metal-Oxide-Semiconductor- Semiconductor de óxido de metal complementario

TTL = Transistor-transistor logic- lógica transistor a transistor

V = Voltios

PC = Personal Computer - computador personal

SI = Sistema informático

DBMS= Data Base Management System- Sistemas de Gestión de Bases de Datos

Q = Caudal

T = tiempo

PH = Potencial de Hidrogenios

RAM = Random Access Memory - Memoria de acceso aleatorio

Anexo 10

Glosario de términos

Código libre. Los usuarios pueden hacer su propia versión del módulo, ampliándolo y mejorándolo.

Sensor. Dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas.

Software. Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Microcontrolador. Es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida.

Prototipo. Un prototipo es un modelo fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas.

Servidor. Un servidor es un equipo informático que forma parte de una red y provee servicios a otros equipos cliente.

Base de datos. Una base de datos es una variedad coordinada de datos organizados, o información, normalmente guardada electrónicamente en un marco de PC.

JavaScript. Lenguaje de programación ligero, interpretado y compilado justo a tiempo al de una consola.

Invernadero. Es una construcción agrícola de estructura metálica, usada para el cultivo y/o protección de plantas, con cubierta de película plástica traslúcida que no permite el paso de la lluvia al interior

Lixiviación. Proceso hidrometalúrgico mediante el cual se provoca la disolución de un elemento desde el mineral que lo contiene para ser recuperado en etapas posteriores mediante electrólisis.

Bibliografía

- Agrícolas, N. (12 de Agosto de 2016). *Novagric*. Obtenido de Novagric:
<https://www.novagric.com/es/venta-invernaderos-novedades/tipos-de-invernaderos/invernadero-capilla>
- agricultura, O. d. (5 de Noviembre de 2016). *Organización de las Naciones Unidas para la alimentación y la agricultura*. Obtenido de Organización de las Naciones Unidas para la alimentación y la agricultura: <http://www.fao.org/ecuador/fao-en-ecuador/ecuador-en-una-mirada/es/>
- Agropinos. (12 de Agosto de 2020). *Agropinos*. Obtenido de Agropinos:
<https://www.agropinos.com/sistema-de-riego-por-aspersion>
- Alcubierre, D. (30 de Junio de 2014). *CEMAER*. Obtenido de CEMAER:
<https://www.cemaer.org/que-son-las-fotoceldas/>
- Alvarez, M. A. (19 de Noviembre de 2015). *DesarrolloWeb*. Obtenido de DesarrolloWeb:
<https://desarrolloweb.com/articulos/1325.php>
- Alvarez, M. A. (19 de Noviembre de 2015). *Desarrolloweb.com*. Obtenido de ¿Que es python?: <https://desarrolloweb.com/articulos/1325.php>
- Antonio. (2018). *proyectoarduino*. Obtenido de <https://proyectoarduino.com/sensor-de-ultrasonidos-medir-distancia-con-arduino/>
- Areatecnologia. (4 de agosto de 2021). *Areatecnologia*. Obtenido de Areatecnologia:
<https://www.areatecnologia.com/>
- Arendiendoarduino. (s.f.). *aprendiendoarduino.wordpress.com*. Recuperado el Agosto de 2020, de <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>
- ARIMETRICS. (14 de Julio de 2015). *ARIMETRICS*. Obtenido de ARIMETRICS:
<https://www.arimetrics.com/glosario-digital/google-maps>
- BBVA. (10 de Noviembre de 2016). *API_MARKET, BBVA*. Obtenido de <https://www.bbvaapimarket.com/es/mundo-api/plantillas-para-paneles-de-administracion-desarrolladas-con-el-framework-bootstrap/#:~:text=AdminLTE%20es%20un%20panel%20de,una%20construcci%C3%B3n%20y%20personalizaci%C3%B3n%20sencillas.&text=AdminLTE%20es%20t%C3%A>
- Buzón, M. (26 de Enero de 2020). *Profesional Review*. Obtenido de Profesional Review:
<https://www.profesionalreview.com/2020/01/26/definicion-software/>

- Chilán, A. (25 de octubre de 2019).
 Conceptodefinicion. (26 de Julio de 2019). *Conceptodefinicion*. Obtenido de
 Conceptodefinicion: <https://conceptodefinicion.de/humedad/>
- daissykaro344. (Noviembre de 2015). *slideshare*. Obtenido de
<https://es.slideshare.net/daissykaro344/uso-de-protoboard>
- Devart. (14 de Julio de 2018). *Devart*. Obtenido de
<https://www.devart.com/dbforge/mysql/studio/>
- EC, I. (14 de Diciembre de 2020). Obtenido de https://articulo.mercadolibre.com.ec/MEC-432317625-mini-bomba-agua-aire-6-12v-arduino-acuarios-nuevo-rye-_JM#position=6&search_layout=stack&type=item&tracking_id=e991b056-2a40-4c88-8204-f93a2c9deada
- Electronica, C. (12 de 06 de 2021). *Carrod Electronica*. Obtenido de Carrod Electronica:
<https://www.carrod.mx/products/cmos-seis-inversores-cd4069>
- electronicaplugandplay*. (s.f.). Obtenido de electronicaplugandplay:
<https://www.electronicaplugandplay.com/sensores-y-transductores/product/319-fotocelda-sensor-de-luz-ldr-gl5528>
- Electronilab. (17 de Marzo de 2016). *Electronilab*. Obtenido de Electronilab:
<https://electronilab.co/tienda/sensor-de-temperatura-y-humedad-dht11/>
- Environment, H. (20 de Septiembre de 2018). *Hydro Environment*. Obtenido de Hydro Environment:
https://www.hydroenv.com.mx/catalogo/index.php?main_page=page&id=44
- Fontanela, A. (14 de Octubre de 2020). *Raiola Networks*. Obtenido de Raiola Networks:
<https://raiolanetworks.es/blog/bootstrap/>
- FriendsApache. (12 de Julio de 2021). *Apache Friends*. Obtenido de Apache Friends:
<https://www.apachefriends.org/es/index.html>
- Garcia, J. B. (11 de 12 de 2015). *arsys*. Obtenido de arsys:
<https://www.arsys.es/blog/programacion/que-es-laravel/>
- gcfglobal. (2018). *gcfglobal*. Obtenido de <https://edu.gcfglobal.org/es/informatica-basica/que-es-hardware-y-software/1/>
- Geekfactory. (s.f.). *Geekfactory*. Recuperado el Agosto de 2020, de
<https://www.geekfactory.mx/tienda/sensores/hc-sr04-sensor-de-distancia-ultrasonico/>

- Guioteca. (Julio de 2015). *Guioteca*. Recuperado el Agosto de 2020, de <https://www.guioteca.com/educacion-para-ninos/que-es-el-ultrasonido-y-cuales-son-sus-aplicaciones/>
- Heptro. (12 de Noviembre de 2017). *hetpro-store*. Obtenido de <https://hetpro-store.com/TUTORIALES/microcontrolador/>
- Horticultivos. (26 de Julio de 2017). *Horticultivos*. Obtenido de Horticultivos: <https://www.horticultivos.com/featured/principales-tipos-invernaderos/>
- INSST. (s.f.). *Instituto Nacional de Seguridad y Salud en el Trabajo*. Obtenido de Instituto Nacional de Seguridad y Salud en el Trabajo: <https://www.insst.es/-/que-es-un-invernader-1>
- Keyence. (s.f.). *Keyence*. Obtenido de <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info>
- KNFBReader. (2018). *KNFBReader*. Recuperado el Agosto de 2020, de <https://knfbreader.com/>
- Koalab. (Abril de 2017). *koalab.tech*. Obtenido de <https://koalab.tech/aprende/componentes/sensor-ultrasonico-hc-sr04/>
- LATAM, M. (2018). *mecatronicalatam*. Obtenido de <https://www.mecatronicalatam.com/es/tutoriales/sensores/>
- Lazzus. (s.f.). *Lazzus*. Recuperado el Agosto de 2020, de <http://lazzus.com/es/#what>
- Llamas, L. (17 de Diciembre de 2015). *luisllamas*. Obtenido de <https://www.luisllamas.es/medir-distancia-con-Arduino-y-sensor-de-ultrasonidos-hc-sr04/>
- Llamas, L. (18 de Diciembre de 2016). Obtenido de <https://www.luisllamas.es/bomba-de-agua-con-arduino/>
- Maherelectronica. (24 de Enero de 2018). *Maherelectronica Smart Agorcontrollers*. Obtenido de Maherelectronica Smart Agorcontrollers: <https://maherelectronica.com/sistema-riego-automatico/>
- Mozilla. (2015). *Mozilla*. Obtenido de <https://developer.mozilla.org/es/docs/Web/Guide/HTML/HTML5>
- Naylamp. (15 de Julio de 2016). *naylampmechatronics*. Obtenido de [naylampmechatronics: https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html](https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html)

- Nicolás Avellaneda. (s.f.). *Sensores Industriales*. Recuperado el Agosto de 2020, de <https://sites.google.com/site/654sensoresindustriales/home/-quee-son-los-sensores>
- Oracle. (12 de Julio de 2016). Obtenido de <https://www.oracle.com/mx/database/what-is-database/>
- Oracle. (13 de Julio de 2016). *Oracle*. Obtenido de Oracle: <https://www.oracle.com/mx/database/what-is-database/>
- Organización Mundial de la Salud. (Agosto de 2014). *www.who.int*. Recuperado el Agosto de 2020, de <https://www.who.int/features/factfiles/blindness/es/>
- PHP. (16 de Agosto de 2016). Obtenido de PHP: <https://www.php.net/manual/es/intro-what-is.php>
- PMI. (2018). *PMI*. Obtenido de <https://pmi.org.py/index.php/pmi/que-es-el-pmi>
- Quimis, B., & Cantos, T. (2018). *PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICOS PARA EL MONITOREO DE HIPERTENSIÓN CON PACIENTES EN ESTADO DE GESTACIÓN*. Guayaquil: Universidad de Guayaquil.
- Robledano, Á. (24 de Septiembre de 2019). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-mysql/>
- Rodriguez, P. (10 de Febrero de 2020). *Futuro Eléctrico*. Obtenido de Qué es la resistencia eléctrica y para qué sirve: <https://futuroelectrico.com/resistencia-electrica-para-que-sirve/>
- SA, B. I. (4 de Julio de 2014). *Libro Datos Tecnico de Hidraulica de Bombas* . Obtenido de <https://www.benoit.cl/Bombas2.htm>
- Sepúlveda, A. (12 de Mayo de 2018). *Parques Alegres*. Obtenido de Parques Alegres: <https://parquesalegres.org/biblioteca/blog/sistema-riego-concepto/>
- Shashua, A., & Aviram, Z. (2018). *Orcam*. Recuperado el Agosto de 2020, de <https://www.orcam.com/es/myeye2/>
- Significados. (14 de Agosto de 2019). *Significados*. Obtenido de Significados: <https://www.significados.com/temperatura/>
- Systems, O. (30 de Enero de 2015). *Onyx Systems*. Obtenido de Onyx Systems: <https://www.onyxsystems.es/que-es-un-servidor.html>
- Technology Safety. (2019). <https://www.techsafety.org>. Recuperado el Agosto de 2020, de <https://www.techsafety.org/tecnologa-asistencial>
- Valdés, D. P. (26 de Octubre de 2015). *Maestrosdelweb*. Obtenido de Maestrosdelweb: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

Visus, A. (Octubre de 2020). *esic*. Obtenido de esic:

<https://www.esic.edu/rethink/tecnologia/para-que-sirve-python>

Xakata. (15 de Julio de 2018). *Xakata*. Obtenido de Xakata:

<https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi>

Xataka. (Julio de 2018). *Xataka*. Obtenido de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Xataka. (s.f.). *Xataka*. Recuperado el Agosto de 2020, de

<https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Yanchatuña Aguayo, L. (Fecbrero de 2016). *repositorio.uta.edu.ec*. Obtenido de

<https://repositorio.uta.edu.ec/handle/123456789/20348>

Zepeda, R. (19 de Septiembre de 2018). *El informático*. Obtenido de El informático:

<http://blogdelinformatico-reizer.blogspot.com/2015/11/que-es-xampp.html>