

**Sistema Automatizado de Riego, Fertilización y Fumigado Para Cultivo de Habichuela  
Bajo Invernadero, Monitoreado Mediante Aplicación Móvil.**

**Elaborado por:**

**Cristian Camilo Quiroga Medina**

**Maicol Douglas Mora Perdomo**

**Luis Adolfo Cuellar Medina**

**Universidad Nacional Abierta y a Distancia (UNAD)**

**Escuela de Ciencias Básicas de la Tecnología E Ingeniería**

**Programa de Ingeniería Electrónica**

**Pitalito 2018**

**Sistema Automatizado de Riego, Fertilización Y Fumigado Para Cultivo de  
Habichuela Bajo Invernadero, Monitoreado Mediante Aplicación Móvil.**

**Elaborado por:**

**Cristian Camilo Quiroga medina**

**Maicol Douglas Mora Perdomo**

**Luis Adolfo Cuellar medina**

**Modalidad de grado de proyecto aplicado para optar título como ingeniero electrónico**

**Asesor:**

**Diego Fernando Nava Cuevas**

**Ingeniero electrónico**

**Universidad Nacional Abierta y a Distancia UNAD**

**Escuela de ciencias básicas de la tecnología e ingeniería**

**Programa de ingeniería electrónica**

**Pitalito 2018**

## Dedicatoria

### Dedicatoria Cristian Camilo Quiroga

*Con todo cariño para mis padres José y Melba, por ser el motor que me impulsa a proponerme grandes objetivos, cumplir valiosas metas y a esforzarme por ser cada vez mejor. A mis hermanas Valentina y Daniela y a mi familia por su apoyo incondicional, por creer en mí y tenerme presente siempre en sus oraciones. Camilo Quiroga*

## **Dedicatoria Maicol Douglas Mora**

*Primero Que todo agradezco a Dios  
por darme la vida y este triunfo de  
desarrollo profesional, Dedico este  
trabajo de grado a todos quienes me  
han apoyado en este camino  
universitario, en especial a mi familia,  
amigos, compañeros, tutores y en  
general a todos los que estuvieron junto  
a mí, también a la Universidad  
Nacional abierta y a distancia por el  
espacio de aprendizaje. Maicol  
Douglas*

## **Dedicatoria Luis Adolfo Cuellar**

*Este proyecto va especialmente  
dedicado a mis padres, María Gladis y  
Luis Antonio por tener la valentía de  
hacerme la persona que soy hoy en día,  
también a mi hermana María José,  
porque muchos de mis logros que he  
tenido son gracias a ustedes sin ustedes  
no había podido lograrlos, pero gracias  
a su apoyo incondicional ya que son el  
motor de mi vida; este no será el último  
les seguiré dedicando más logros, son  
solo y especialmente para ustedes*

*Gracias padres y hermana. Luis  
Cuellar*

## **Agradecimiento**

En nombre de los autores gracias a Dios por permitirnos la existencia y la alegría de disfrutar tantos logros, a la Universidad Nacional Abierta y a Distancia por brindarnos un medio alternativo para nuestra superación y a cada uno de los docentes, especialmente a quienes han estado cerca de nuestro proceso educativo ofreciendo innumerables asesorías y apoyo, gracias a ellos por su incansable labor y sus múltiples ideas generadoras de pensamiento, igualmente gracias a cada una de nuestras familias por brindarnos su apoyo incondicional, que nos ha permitido paso a paso culminar esta tesis.

## Tabla de contenido

<i>Dedicatorias</i> .....	3
Dedicatoria Cristian Camilo Quiroga .....	3
Dedicatoria Maicol Douglas Mora .....	4
Dedicatoria Luis Adolfo Cuellar.....	5
<i>Agradecimiento</i> .....	6
<i>Listado de figuras</i> .....	13
<i>Listado De Anexos</i> .....	16
<i>Listado de tablas</i> .....	16
<i>Resumen</i> .....	17
<i>Abstract</i> .....	18
<i>Introducción</i> .....	19
<i>Planteamiento Del Problema</i> .....	21
<i>Justificación</i> .....	22
<i>Objetivos</i> .....	24
Objetivo General.....	24
Objetivos Específicos .....	24
<i>Marco Referencial</i> .....	25

Marco teórico .....	25
Tecnologías para el agro .....	25
Agricultura de precisión.....	25
<i>Marco conceptual</i> .....	27
¿Qué es un invernadero?.....	27
Tipos de invernadero .....	27
Humedad relativa (UR).....	29
Suelo .....	30
Riego .....	32
Fertilización .....	33
Fumigación .....	34
Cultivos de habichuelas .....	35
Automatización.....	36
Internet de las cosas .....	36
Señales digitales.....	37
Señales analógicas .....	38
Optoacoplador.....	38
Relé .....	40
Modulo relé.....	41
Especificaciones: .....	43
<i>Electroválvula</i> .....	46
<i>Compresor</i> .....	46

<i>Arduino</i> .....	47
<i>Nodemcu</i> .....	50
Protocolo Modbus protocolo serie .....	51
<i>App inventor</i> .....	52
Aplicación móvil.....	53
<i>Thingspeak</i> .....	54
<i>Sensor DHT11</i> .....	55
Antecedentes .....	56
Antecedes internacionales.....	56
<i>Antecedentes locales</i> .....	58
<i>Diseño metodológico</i> .....	62
Análisis y levantamiento de información: .....	62
Diseño .....	62
Desarrollo.....	63
Implementación .....	63
Evaluación del sistema y resultados obtenidos.....	63
<i>Ingeniería Del Proyecto</i> .....	64
Invernadero a automatizar .....	64
Diagrama general del sistema automatizado. ....	66
Diagrama de comportamientos .....	67

Diagrama de componentes.....	68
Sistema de fumigación.....	69
Fertilización .....	69
Humedad relativa.....	71
Humedad del suelo.....	72
Sistema de riego.....	72
Etapa de potencia .....	74
Sistema de apagado automático .....	74
<i>Elección de Arduino</i> .....	76
Reloj de tiempo real (RTC) .....	77
Sistema de comunicación.....	77
Node Mcu Esp8266 .....	78
Thingspeak.....	78
Aplicación para monitoreo.....	79
Entorno de desarrollo App Inventor .....	79
Diagrama de flujo funcionamiento de la aplicación: .....	81
Componentes y conexiones .....	81
Diagrama conexionado de Arduino .....	83
<i>Desarrollo del sistema automatizado</i> .....	84
Circuitos.....	84
Sistema de apagado automático .....	84

Descripción de software y comunicación .....	85
Comunicación Arduino – Nodemcu .....	85
Desarrollo de aplicación en app inventor .....	86
Código principal Arduino .....	88
Código envío de datos a NODEMCU ESP8266.....	89
Código de programación para sistema de fumigación.....	89
Código de programación para sistema de fertilización.....	89
Código funciones para Bitmap .....	89
Código de programación para recolección de datos de sensores.....	90
Código de programación para recepción de datos y envío a thingspeak .....	90
Visualización no remota mediante TFT LCD.....	90
Sistema de fumigación.....	91
Sistema de fertilización.....	93
Control de humedad del suelo .....	94
Sensor de humedad del suelo HL-69 .....	94
Adecuación de electroválvula.....	95
Implementación de riego por goteo .....	96
Adecuación de sensor de humedad relativa y temperatura (DHT11).....	97
<i>Funcionamiento del sistema</i> .....	97
Visualización no remota .....	97
Fumigación .....	98
Riego.....	98

Fertilización .....	99
Monitoreo.....	99
Etapas de crecimiento y producción .....	100
Germinación.....	100
Crecimiento.....	102
Desarrollo.....	103
Producción .....	104
Producto final.....	105
<i>Análisis e Interpretación de Resultados</i> .....	105
<i>Impactos</i> .....	108
<i>Verificación de Objetivos</i> .....	109
Investigar parámetros y condiciones para cultivo de habichuelas bajo invernadero .	109
Diseñar sistema de control que incluye circuitos electrónicos y actuadores .....	109
Implementar el sistema de control de humedad, fertilización y fumigación. ....	109
Verificar el correcto funcionamiento del Sistema .....	110
Aportes realizados en el proceso de producción.....	110
<i>Conclusiones</i> .....	112
<i>Recomendaciones</i> .....	113
<i>Bibliografía</i> .....	113
<i>Anexos</i> .....	120

Códigos de programación.....	120
------------------------------	-----

### **Listado de figuras**

Figura 1.	Invernadero asimétrico (Technology, 2018) .....	28
Figura 2.	Invernadero Almería Type (Technology, 2018).....	28
Figura 3.	Invernadero Multicapilla (Technology, 2018).....	29
Figura 4.	Internet de las cosas .....	37
Figura 5.	Ilustración diseño típico Opto acoplador (itlalaguna.edu, 2018) .....	39
Figura 6.	Optoacoplador (autoría propia) .....	40
Figura 7.	Modulo relé (Tienda8, 2018).....	42
Figura 8.	Diseño esquemático de modulo relé controlado mediante Arduino.	

Realizado en proteus 8, se describen las conexiones para su correcto funcionamiento.

(autoría propia)	45	
Figura 9.	Electroválvulas .....	46
Figura 10.	Compresor .....	47
Figura 11.	Placa Arduino mega (Arduino, Arduino, 2018) .....	48
Figura 12.	Placa de desarrollo Nodemcu ESP8266- 12E (Llamas, 2018) .....	51
Figura 13.	Protocolo de comunicación modbus (logicbus, 2018).....	52
Figura 14.	MIT App inventor (Appinventor, 2018).....	53
Figura 15.	Aplicación móvil (Multiplicalia, 2018).....	54
Figura 16.	Esquema Thingspeak (Ning, 2017) .....	55

Figura 17.	Sistema de inyección de fertilizante con bombas de paletas (Peña Peralta, 2010)	57
Figura 18.	Creación de clúster de controles e indicadores (Ruiz canales & molina martinez, (2010).	60
Figura 19.	Invernadero automatizado .....	65
Figura 20.	Diagrama general. Se describen de forma general por partes los bloques, que se requieren para cumplir con lo propuesto en el proyecto. (autoría propia) ..	66
Figura 21.	Diagrama de comportamientos. En está sé describen los comportamientos de cada uno de los componentes que se encuentran dentro del invernadero, dependiendo de las condiciones en las que se encuentre (autoría propia) .....	67
Figura 22.	Diagrama de componentes (autoría propia).....	68
Figura 23.	Sistema de fumigación (autoría propia).....	69
Figura 24.	Diseño de sistema de fertilización (autoría propia).....	70
Figura 25.	Diseño sistema de fertilización (autoría propia).....	71
Figura 26.	Ubicación de sensores de humedad sobre invernadero (autoría propia)	72
Figura 27.	Diseño de sistema de riego por goteo (autoría propia).....	73
Figura 28.	Esquemático Control de alimentación por fotorresistencia (autoría propia)	74
Figura 29.	Esquemático Circuito con relé digital. (autoría propia) .....	76
Figura 30.	Módulo RTC (autoría propia).....	77
Figura 31.	Node Mcu Esp8266 (autoría propia) .....	78
Figura 32.	Envío de datos a Thingspeak (autoría propia) .....	79
Figura 33.	Diseño de aplicación móvil (Autoría propia) .....	80

Figura 34.	Diagrama de flujo MY GREEN HOME. (Autoría propia) .....	81
Figura 35.	Diagrama conexionado de Arduino (autoría propia) .....	83
Figura 36.	Montaje Real Control de alimentación por fotorresistencia.....	84
Figura 37.	Montaje Real relé digital (autoría propia). .....	85
Figura 38.	Código pantalla principal App inventor (autoria propia) .....	86
Figura 39.	Código pantalla humedad del suelo, relativa y estado del riego App inventor (autoria propia).....	87
Figura 40.	Programación de fertilización app inventor (autoria propia).....	87
Figura 41.	Programación de códigos en Arduino IDE (autoría propia).....	88
Figura 42.	Pantalla TFT para visualización de datos .....	90
Figura 43.	Sistema de desplazamiento para fumigación.....	91
Figura 44.	Tarjeta encargada de controlar el motor paso a paso.....	92
Figura 45.	Boquilla para fumigación .....	92
Figura 46.	Electroválvula para fumigación.....	93
Figura 47.	Estante de almacenamiento de fertilizante .....	93
Figura 48.	Electroválvula para fertilización.....	94
Figura 49.	Sensor de humedad instalado (autoría propia) .....	95
Figura 50.	Implementación de modulo relé en invernadero (autoria propia) .....	95
Figura 51.	Electroválvula para riego (autoría propia).....	96
Figura 52.	Pruebas de funcionamiento de sistema de riego por goteo (autoria propia). Desarrollo y funcionamiento del sistema automatizado para invernadero .....	97
Figura 53.	Adecuación de sensor DHT11 .....	97
Figura 54.	Proceso de germinación planta de habichuela.....	101
Figura 55.	Germinación de planta de habichuelas .....	101

Figura 56.	Planta de habichuela recién germinada .....	101
Figura 57.	Planta de habichuela recién germinada .....	102
Figura 58.	Planta en crecimiento.....	102
Figura 59.	Planta en crecimiento.....	103

### **Listado De Anexos**

Anexo.1	Código principal Arduino Mega .....	120
Anexo.2	Código envío de datos a NodeMcu (ESP8266).....	126
Anexo.3	Código de programación para sistema de fumigación .....	126
Anexo.4	Código de programación proceso de fertilización.....	128
Anexo.5	Código funciones para bitmap.....	128
Anexo.6	Código de programación para recolección de datos de sensores. ....	133
Anexo.7	Código programación para NodeMcu (ESP8266) .....	135

### **Listado de tablas**

Tabla.1	Según Teslabem (2018), se describen las características y especificaciones en las que se describen voltajes de trabajo y voltajes soportados. ....	42
Tabla.2	Características sensor temperatura y humedad relativa DHT11. ....	55

## Resumen

Dentro del diseño y desarrollo de este proyecto aplicado, se elaboró un sistema totalmente automatizado en los aspectos de riego, fertilización y fumigado para un cultivo de habichuelas, donde se utilizó sistemas de medición de temperatura y humedad relativa para garantizar que el cultivo se encuentre siempre en óptimas condiciones.

Se implementó un sistema capaz de realizar las tareas de fertilización y fumigado en el cultivo, reduciendo tiempo y esfuerzo del agricultor y aumentando la eficiencia y calidad del cultivo, adicionalmente este sistema fue monitoreado remotamente haciendo uso de una aplicación móvil.

Este proyecto se enfocó en la importancia que se le deben dar a cada una de las fases de desarrollo de los cultivos, en el cual se aplicó de manera automatizada, el riego, la fertilización y la fumigación, con los cuales se logró producir una cosecha de calidad, además de disminuir las infecciones y enfermedades en las plantas.

De acuerdo con la aplicación de este sistema, se comprobó que el uso de la tecnología en la industria agrícola permite la obtención de mejores resultados, en cuanto a la producción, la calidad, la disminución del tiempo y trabajo dedicado al cultivo y el control técnico de las fases necesarias para la producción.

**Palabras clave:** actuadores, aplicaciones móviles, comunicación vía WI-FI, Control, IOT, sensores, sistema embebido.

## **Abstract**

Within the design and development of this applied project, a fully automated system was developed in the aspects of irrigation, fertilization and fumigation for a bean crop, where temperature and relative humidity measurement systems were used to ensure that the crop is always in optimal conditions.

A system was implemented that was able to carry out the tasks of fertilization and fumigation in the crop, reducing time and effort of the farmer and increasing the efficiency and quality of the crop, additionally this system was monitored remotely using a mobile application.

This project focused on the importance that he should be given to each of the phases of crop development, in which it was applied in an automated way, the irrigation, fertilization and fumigated, with which it was possible to produce a quality harvest, additionally to reducing infections and diseases in plants.

In accordance with the application of this system, it was found that the use of technology in the agricultural industry allows obtaining better results, in terms of production, quality, decrease of time and work dedicated to cultivation and technical control of the phases necessary for production.

**Keywords:** actuators, mobile applications, communication via WI-FI, Control, IOT, sensors, embedded system.

## Introducción

Desde la creación e inclusión de las tecnologías de comunicación, ya sea para la transmisión de voz, datos, videos o el mismo internet, estos medios de comunicación han sido bien aceptados e incluso innovados con el fin de hacerlos cada vez más rápidos y eficaces, posterior a la invención de esto se ha hecho uso de tecnología de punta y avances para ser aplicados en diferentes campos.

Hablando de la automatización en el ámbito agropecuario es el sector que más presenta rezagos, Colombia y específicamente el departamento del Huila que es el entorno más cercano, la economía depende de la agricultura, existe una oportunidad de avanzar en este sector, como ha sucedido en otros sectores como el petrolero y minero que han adecuado diversas tecnologías para automatizar sus procesos.

El presente trabajo que se ha nombrado “Sistema Automatizado De Riego, Fertilización Y Fumigado para cultivo de Habichuela Bajo Invernadero, Monitoreado Mediante Aplicación Móvil”, en el cual a través de tecnologías de Arduino se implementa un prototipo que permite por medio de sensores, actuadores y protocolos de comunicación, la automatización de las tareas comprometidas en la producción de habichuelas, además del monitoreo de este proceso.

Este proyecto es orientado a la automatización de procesos en el sector agropecuario, más exactamente al cultivo de habichuela bajo invernadero, teniendo en cuenta que, para todos los procesos de fertilización, fumigado y riego, dentro de un cultivo de habichuela depende de la intervención humana; con la implementación del prototipo de automatización

es posible reducir la presencia del agricultor en el cultivo, además de mejorar el proceso; el cultivo sobre el que se trabajara tiene una serie de requerimientos en la producción para obtener un producto final de calidad, como lo es la humedad relativa, los nutrientes del suelo y los herbicidas necesarios.

## Planteamiento Del Problema

El incremento de la demanda y la competitividad en el mercado ha encaminado a pequeños y grandes agricultores optar por mejorar la producción agrícola, la calidad de los productos, además considerando que existen una serie de rezagos en la adecuación de tecnologías en pro de mejorar la producción agraria del territorio colombiano.

Teniendo en cuenta la importancia de la agroindustria, existen muchas regiones de Colombia con terrenos fértiles, donde aún se mantiene una visión retrógrada del campo y se siguen empleando métodos rústicos para su explotación. Esto casusa un evidente impacto ambiental y socioeconómico deficiente; las prácticas agrícolas convencionales tienen en cuenta que los cultivos son terrenos homogéneos que brindan la misma cantidad de recursos en cada fracción del suelo, por lo que la aplicación de nutrientes en el suelo se haga de forma uniforme de acuerdo con los conocimientos empíricos del agricultor, hecho que provoca el desaprovechamiento de los insumos y el recurso hídrico (Castro, 2015). Siendo conscientes de esto y teniendo en cuenta la constante evolución tecnológica y la variedad de aplicaciones que ofrecen en diferentes sectores de producción, se desea implementar un sistema de control automatizado para el cultivo de habichuelas bajo invernadero que pueda ser monitoreado mediante aplicación móvil orientado al concepto de IOT o internet de las cosas.

En contexto a esto: ¿se puede implementar un sistema automatizado de riego, fertilización y fumigado en un cultivo de habichuela monitoreado mediante aplicación móvil?

## Justificación

Por estos tiempos las sociedades enfrentan, un marco global de constante crecimiento de la población en un ámbito de rápido cambio global. Las demandas de más y mejores alimentos se vuelven el centro de políticas y estrategias locales regionales e internacionales, con miras a alcanzar la seguridad alimentaria, brindar la calidad de los productos y la sostenibilidad de los ecosistemas implicados (Medina, de Obschatko, & Preciado Patiño, 2013). Los cultivos en invernadero han brindado gran variedad de beneficios a la hora de cultivar, ofreciendo mayor rendimiento y productos de calidad, que permiten cultivar en cualquier época del año en cualquier clima.

El hecho de añadir agua y nutrientes a los cultivos dependiendo de los requerimientos del suelo se ha convertido no solo en una opción sino en una necesidad, ya que permite no desperdiciar los insumos y mejorar la calidad de los productos, por ende, se hace uso de tecnologías para el cuidado y automatización de los cultivos.

La implementación del sistema de control automatizado en un cultivo de habichuela capaz de realizar las labores de riego, fertilización y fumigado en el cultivo, podría aumentar la eficiencia hasta en un 65%, reducirá el tiempo de producción, tendrá múltiples ventajas sobre los cultivos tradicionales permitiéndole a los agricultores emplear menos tiempo en la revisión y cuidado del cultivo, siendo una interfaz cómoda para el usuario que permanentemente estará monitoreando el cultivo y enviándole la información en tiempo real sobre el estado de este. (Castelblanco, 2015)

Esto permite mejorar el proceso de cultivación de las habichuelas, tratando de mantener la humedad relativa adecuada, la cantidad de fungicida y fertilizantes que requiere. Dando la oportunidad de minimizar el gasto de estos.

La finalidad de este proyecto es evaluar cada una de las características que hacen productivo y sano el cultivo, con base a ello se diseña un sistema que controle los tiempos de riego, fertilización y fumigado que las plantas requieren, para así mejorar la calidad de la cosecha y evitar el daño del cultivo ya sea por el exceso o falta de agua, fertilizantes y/o químicos.

El principio de funcionamiento es un sistema de automatización embebido que permite captar mediciones de sensores, procesar la información, tomar una decisión y activar los actuadores dependiendo de las condiciones que se deseen alterar.

El proyecto está orientado al uso de las IOT (internet de las cosas), que realizan la interconexión digital entre objetos cotidianos con internet, haciendo posible el envío y recepción de información de forma remota posibilitando el monitoreo de sistemas embebidos desde cualquier lugar y con solo contar con acceso a internet, lo que por hoy en día es algo muy común y de fácil acceso.

El principal beneficiario de la implementación del proyecto pilotos como el presente son los agricultores, debido a que la finalidad es automatizar las tareas más importantes, permitiendo obtener producciones de calidad y con menos presencia requerida por parte del agricultor sobre el cultivo.

## **Objetivos**

### **Objetivo General**

- Diseñar e implementar un sistema automatizado para un cultivo de habichuela monitoreado desde aplicación móvil.

### **Objetivos Específicos**

- Investigar parámetros y condiciones para cultivo de habichuelas bajo invernadero
- Diseñar sistema de control que incluye circuitos electrónicos y actuadores
- Implementar el sistema de control de humedad, fertilización y fumigación.
- Verificar el correcto funcionamiento del Sistema

## **Marco Referencial.**

### **Marco teórico**

#### **Tecnologías para el agro**

Las tecnologías del agro funcionalmente han sido creadas como herramienta mediadora que facilita las tareas entre el hombre y la naturaleza. El papel básico que cumple es la transformación de la naturaleza para beneficio de las personas que viven en el campo.

El proceso de explotación de los suelos se supone que es de forma lineal, en el que lo de uso tradicional es remplazado por la tecnología. Causando diversas situaciones económicas y socioculturales en el cual varios pensadores han trabajado en determinar los impactos e implicaciones de forma social. De esta forma, el concepto de tecnología se entiende como conocimientos y procesos para construir un nuevo panorama y resolver los problemas presentes. (Jasso, 1999)

#### **Agricultura de precisión**

Un concepto que soporta teóricamente el desarrollo de esta tesis es la agricultura de conservación o agricultura de precisión (AP), el fin de está es la aplicación localizada de los insumos agrícolas, que se añaden de acuerdo con los requerimientos de los suelos para un cultivo. Este hecho que añade tecnología al agro está siendo implementada a nivel mundial principalmente en países desarrollados, con el fin de distribuir de forma eficiente los recursos, incrementar la calidad de los productos y con el objetivo de transformar la agricultura en un sector más rentable, siendo la agricultura uno de los sectores más significativas en cuanto a la generación de empleo y por ende una de las más importantes actividades económicas. (SAC, 2010)

Esta base teórica fundamenta de forma importante este proyecto, ya que esta organización colombiana expone la importancia de la agricultura de precisión, que trata de añadir los insumos necesarios al suelo y a las plantas solamente cuando se requieren, la implementación del sistema automatizado de riego, fertilización y fumigación para un cultivo de habichuelas, trata de aplicar el concepto de agricultura de precisión, a través de la tecnificación o especificación de las condiciones que se pueden manipular en el cultivo que en este caso se encuentra bajo un invernadero.

El agua es un recurso vital en la agrícola de carácter limitado, su distribución óptima es de suma importancia (LÓPEZ, 2012). En la actualidad, se implementan diferentes formas de riego que carecen de un sistema que en conjunto permita medir la humedad de suelo, controlar el tiempo de riego e identificar la parte de suelo que requiere ser regada (DEAQUIZ, 2014). Por lo tanto, esto deja de ser solo una opción tecnológica a ser una necesidad para, satisfacer de forma efectiva y precisa el riego en un cultivo (VEGA, 2013)

Existen diversos lugares en Colombia en los que el acceso agua para los cultivos es limitado, por eso es sumamente importante la distribución de este recurso de forma precisa, por esta razón, el hecho de aplicar la necesaria, no se ha convertido en una opción, sino que además representa una necesidad dentro del proceso de la explotación del suelo.

## **Marco conceptual**

### **¿Qué es un invernadero?**

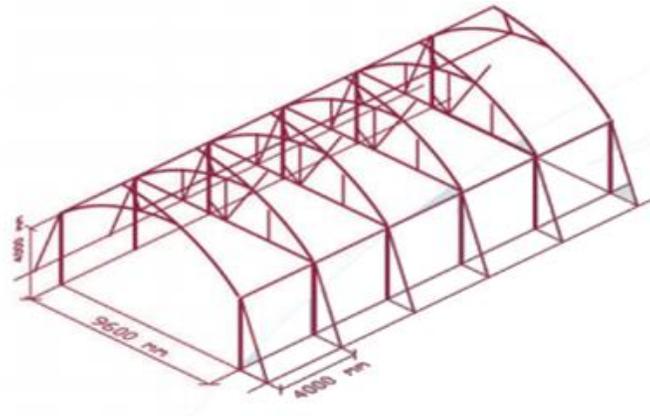
Un invernadero es cualquier tipo de construcciones o estructuras que están cubiertas por materiales transparentes o semitransparentes, en la que se pueden tener un microclima de forma artificial, para con ello realizar el proceso de cultivación de una planta específica en óptimas condiciones y fuera de la época.

Este es un sistema eficiente y de bajo coste para utilizar los rayos del sol a favor de las plantas. Existen diversas opciones para utilizar invernaderos y a la hora de especificar el tipo de invernadero se debe tener en cuenta estos factores, requerimientos ambientales del cultivo (clima), las condiciones climáticas de la zona, disponibilidad de recurso humano e insumos criterios de eficiencia y funcionalidad. (Miserendino, 2014)

### **Tipos de invernadero**

#### *Asimétrico*

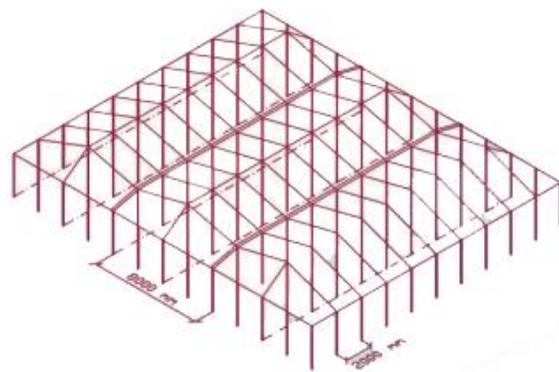
El tipo de invernadero asimétrico está diseñado para realizar cultivación en clima tropical, el cruce de los semiarcos permite una corriente de aire cenital. (Technology, 2018)



**Figura 1. Invernadero asimétrico** (*Technology, 2018*)

### *Almería Type*

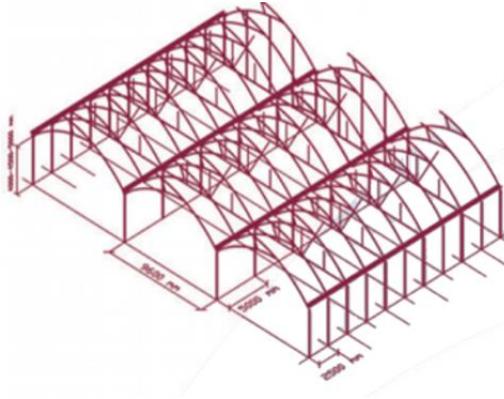
El diseño de invernadero Almería es una construcción por medio de tubos encofrados con entramado de alambres y cordón acerado. Su contorno es plástico y es básicamente adecuado para cualquier tipo de cultivo. En este tipo de invernadero el aprovechamiento del espacio es máximo y es adaptable al terreno, de bajo coste y puede llevar ventilación cenital automática. (*Technology, 2018*)



*Figura 2. Invernadero Almería Type* (*Technology, 2018*)

## **Multicapilla**

Esta clase de invernadero está construido para cualquier tipo de invernadero, estos pueden ser cubiertos con plástico, mallas, policarbonato. Permite variedad de tipos de ventilación. (Technology, 2018)



*Figura 3. Invernadero Multicapilla (Technology, 2018)*

## **Humedad relativa (UR)**

El vapor es formado por la evaporación del agua que se encuentra en el suelo, este vapor generado es directamente transmitido al aire, en cuantías proporcionales a las circunstancias del ambiente, causando así un aumento en el contenido de humedad en el aire. Esta condición regularmente es nombrada “cantidad de saturación” y está aumentada por el nivel de temperatura.

Cuando la cantidad de vapor que hay en un volumen de aire determinado, aumenta con la temperatura proporcionalmente, este vapor es condensado hasta cambiar al estado líquido. (Instruments, 2018)

La humedad relativa resulta ser una variable ambiental difícil de controlar dentro de invernaderos, inclusive los equipos para control ambiental no son capaces de controlar

eficazmente el nivel de humedad en invernadero. El nivel de humedad en el aire puede afectar la temperatura y además las plantas transpiran y añaden vapor al aire regularmente. (Peery, 2018)

En algunas áreas climáticas del norte, existen una serie de desafíos que se multiplican debido a varios factores, por ejemplo, que el aire más seco es muy seco para realizar variaciones en el aire. El ambiente húmedo es agente directo a problemáticas, como enfermedades en raíces y en hojas, secado retrasado del sustrato, entres en las plantas, baja calidad en la producción, etc. Por lo tanto, se requiere de la utilización de más pesticidas para el control de enfermedades y las plantas tendrán un crecimiento deficiente y estirado, lo que las hará menos atractivas. (Peery, 2018).

### **Suelo**

El suelo es la superficie de la tierra conformada por sólidos, líquidos y gases, en los que es posible que se desarrollen las raíces de las plantas, captando de ahí los insumos necesarios para su desarrollo. (Sierra, 2016)

Para que un suelo sea ideal debe tener una repartición uniforme de los sólidos, como minerales y materia orgánica y poros necesarios para el tráfico de agua y aire. (Sierra, 2016)

### ***Propiedades físicas del suelo***

Es suelo está principalmente conformado por compuestos sólidos; como minerales de rocas y residuos de plantas. (Sierra, 2016)

Las características físicas del suelo son perceptibles al tacto, se pueden oler y/o medir y están correlacionadas con su conformación, textura color y su capacidad para retener agua, de esto depende si el suelo es apto para la cultivación. (Sierra, 2016)

### ***Capacidad para retener el agua***

EL suelo posee un ciclo con respecto a la retención de líquido el cual empieza con su llegada por medio de precipitaciones o irrigación, el fluido se filtra por el suelo y se evapora. La conservación del agua principalmente depende los poros disponibles; los suelos que son ideales para la cultivación poseen una capacidad pareja para que se transporte el agua, puesto que el exceso de aire pudre los cultivos, mientras que el exceso de líquido puede disminuir la fuerza y vida de las plantas. (Sierra, 2016)

### ***Propiedades químicas del suelo***

Las características químicas del suelo son cambiantes y es necesario realizar un estudio con precisión para establecer cuales están presentes, cuales carecen y cuales están de forma excesiva. (Sierra, 2016)

### ***pH***

El pH del suelo es una medida sobre la acidez o alcalinidad del suelo. El índice varía de 0 a 14, con un nivel neutro en el 7. Desde el 0 al 7 son niveles ácidos y por encima de 7 son alcalinos. La mayor disponibilidad de nutrientes para casi todas las plantas se encuentra entre los niveles 6.5 y 7.5 y los niveles peligrosos fuera de este rango se encuentran por debajo de los 5 y mayores a 8.

La escala del pH es logarítmica, cada unidad representando un incremento de 10 niveles en el índice de acidez o alcalinidad. (Sierra, 2016)

### ***Fertilidad***

Este factor está relacionado a la capacidad del suelo para mantener el abastecimiento de componentes químicos específicos para que se aprovechado por las plantas. Se encuentran dos tipos de nutrientes:

*Macronutrientes:* Estos tienen que encontrarse en amplias cantidades en el suelo para puedan ser por las plantas; los elementos químicos principales son: Nitrógeno, fósforo, potasio y magnesio.

*Micronutrientes:* Estos son inversos a los macronutrientes puesto que están en menor cantidad, más sin embargo se requieren en el suelo para garantizar el adecuado crecimiento de las plantas. Los principales son: Hierro, manganeso, zinc, boro, cobre, molibdeno, cloro y azufre. (Sierra, 2016)

### **Riego**

Desde las últimas décadas se ha incluido el sistema de riego por gravedad en la cultivación de la habichuela, este es el más aconsejables sobre terrenos planos, cuando se disponga de bastante agua y pueda ser añadida mediante los surcos en los cultivos; se tienen una serie de ventajas de la utilización del riego por goteo en las que se encuentran:

Poca mano de obra para su operación

Es adaptable a terrenos planos o con poca pendiente.

En ocasiones donde haya poca disponibilidad de agua para riego, es un sistema eficiente, ya que se puede aprovechar al máximo este recurso.

Se tiene la posibilidad de añadir fertilizantes mezclados con el agua de riego. (DANE, 2016)

### ***Riego por goteo***

El riego por goteo ha sido añadido a la producción agrícola desde hace ya hace varios años y este fue adaptado gracias a su eficiencia, debido a que este sistema permite evitar a la infiltración profunda y algo importante, disminuye el escurrimiento superficial, permitiendo que se añada solamente el líquido que el cultivo requiere para su desarrollo y producción. (Zeballos, 2005).

### ***Que es un sistema de riego por goteo***

Según (Zeballos, 2005)“Es un método de riego localizado donde el agua es aplicada en forma de gotas a través de emisores, comúnmente denominados “goteros”. La descarga de los emisores fluctúa en el rango de 2 a 4 litros por hora por gotero”. Este sistema añade a intervalos frecuentes pequeñas cantidades de agua a la superficie o específicamente a las raíces de cada planta, por medio de delgados tubos, Esta forma de riego ha sido utilizada en varios países, ya que permite una perdida reducida de agua por evaporación o filtración y es aceptado para cualquier cultivo.

### **Fertilización**

Para el correcto suministro de nutrientes en el suelo es un factor importante para la búsqueda de una alta productividad en diversidad de producciones agrícolas, aún más si los cultivos están en suelos que cuentan con una gran capacidad natural para suministrarlos. Por esta razón, los fertilizantes son utilizados para entregar a las plantas complementos nutritivos que el sustrato no tiene. (Vélez, 2014)

### ***Potasio***

Una función elemental de este compuesto químico es el sostenimiento de la turgencia de la planta y además tolo que esto conlleva (Hanke, 2008). El nivel de concentración tiene

similitud al del nitrógeno. Este elemento medio en diferentes procesos fisiológicos y bioquímicos, en los que están la síntesis de la proteína, neutralización de aniones y grupos de ácidos de macromoléculas orgánica, aceleración de enzimas y en regulación de la presión osmótica de la planta trabajando en la apertura y cierre de estomas. El uso del fertilizante es importante en la producción agrícola relacionada al transporte de la foto asimilador desde las hojas al resto de la planta. (Wild, 1992).

El potasio interfiere en la absorción y disminución de los nitrados, mejora la formación de fibras en la planta, tratando de evitar el volcamiento de las cosechas y brinda protección contra los ataques de plagas y enfermedades que se puedan frecuentar y su consumo produce azúcares, almidones y aceites en diferentes cultivos. (Hanke, 2008)

### *Calcio*

El calcio es un nutriente muy importante para la deshidratación produciendo su contracción, esto fomenta la llamada evapotranspiración. Debido a esto este elemento es netamente importante y antagónico al potasio en los cultivos, por eso el equilibrio de estos dos compuestos es indispensable para el crecimiento de las plantas (Hanke, 2008). Además en la membrana celular es empleado como enlace de los fosfolípidos o proteínas de la membrana, lo que permite la formación de paredes y membranas celulares fuertes. (Norton, 2013)

### **Fumigación**

Para la cultivación de habichuelas la competencia de arvenses por el agua y nutrientes es perjudicial para el desarrollo de las plantas. Así que, es sumamente importante realizar un adecuado control de malezas, especialmente durante la etapa inicial del cultivo, haciendo

uso de herbicidas y escardando el cultivo, de acuerdo con el tipo de maleza que se encuentre.

Por lo general se requiere hacer una o dos desyerbas en el ciclo de cultivación de habichuelas, las malezas más frecuentes son hospederas de una gran diversidad de insectos chupadores, principalmente las gramíneas, con el fin de evitar la infestación de plagas en los cultivos. (DANE, 2016)

### **Cultivos de habichuelas**

La habichuela es un cultivo de gran importancia económica, es mundialmente reconocido por sus propiedades alimenticias y usos industriales; siendo posible usar tanto la vaina como el grano. En la actualidad, este cultivo se encuentra ampliamente distribuido por diferentes lugares de los trópicos, subtrópicos y en regiones de clima templado, de esta forma es una de las legumbres más importante tanto en América Latina como en diferentes partes de África. (Hernández, Hernández, & Soto, 2010)

Se ha determinado a América como sitio de origen de la habichuela, especialmente países como México, Guatemala y Perú. En el caso de Colombia no se cuenta con variedades propias; se destacan solo las de tipo voluble, entre los que se sobresalen la variedad Blue Lake o Lago Azul, la cual se estima ocupar aproximadamente el 90% del área sembrada en el país (DANE, 2016). Cabe resaltar que la habichuela cultivada como verdura tiene su origen en Europa y siendo resultado de mutaciones genéticas de frijol común, fue introducido al continente Americano luego de su descubrimiento. Posteriormente, el arquetipo fue introducido a Norteamérica, donde

se han logrado grandes avances en calidad y rendimiento de nuevas variedades (DANE, 2016).

En el aspecto taxonómico, la habichuela pertenece al grupo de las leguminosas, familia papilionaceae, género phaseolus y especie vulgaris; al igual que el fríjol común, debido a que es una mutación de este, por lo que también se le ha denominado como fríjol verde o judía. De acuerdo al comportamiento agronómico de estas dos variedades también son muy similares, diferenciándose solo por la textura de la vaina, la cual es carnosa y no forma fibra, lo que evita que se abra, esto la hace apta para el consumo en verde. El contenido nutricional de la habichuela se caracteriza por ser rico en vitaminas A, vitamina C, tiamina, riboflavina, niacina y elementos como Ca, Mg, Na, P y K. (DANE, 2016)

### **Automatización**

Es un proceso donde se obtiene hacer tareas de manera automática, que antes se realizaban de forma manual, esto trae consigo una serie de ventajas como lo es, aumento en la eficiencia en el proceso de producción, reducción de costes y tiempos de obtención de un producto, pero es necesario hacer mantenimiento y el respectivo control a los equipos de automatización verificando su correcto funcionamiento (Lapiente, 2015) La automatización de tareas ha sido aplicada en diversidad de áreas de la producción y además también se ha implementado de forma digital, automatizando tareas en la nube.

### **Internet de las cosas**

La interconexión de objetos cotidianos con internet es el principal objetivo de las IOT, Con objetos cotidianos se hace referencia a todo tipo de productos como heladeras, licuadoras, automóviles, etc. En general cualquier dispositivo o

electrodoméstico que interactúe de alguna forma con una persona. Otro nuevo concepto que se empieza a escuchar es el Internet de Todo o como se lo conoce por sus siglas en inglés IOE (Internet of Everything). Este concepto, hace referencia no solo a la conexión inteligente de las cosas, sino que también pretende incluir la comunicación de personas, procesos y datos. Es decir que a las tecnologías IoT se le incluiría la inteligencia necesaria para que todo el sistema pueda funcionar forma convergente y orquestada. (Zito, 2018)



*Figura 4. Internet de las cosas (Groups, 2018)*

### **Señales digitales**

Las señales digitales son señales eléctricas que varían en el tiempo transmitiendo información de acuerdo con parámetros previamente programados, para este tipo de señal hay dos símbolos universales que representa cada nivel eléctrico: 0 ó 1, V o F, etc. Los niveles respectivos dependen de la configuración del dispositivo usado. Por ejemplo, si se hace uso de componentes de la familia lógica TTL (transistor-transistor-logic) los niveles

son 0 V y 5 V, aunque cualquier valor por debajo de 0,8 V es interpretado correctamente como un 0 y cualquier valor por encima de 2 V es interpretado como un 1 (los niveles de salida están por debajo de 0,4 V y por encima de 2,4 V respectivamente). Para el caso de la familia CMOS (complementary metal-oxide-semiconductor), los valores dependen de la alimentación. Para alimentación de +5 V, los valores ideales son también 0 V y 5V, pero se reconoce un 0 hasta 2,25 V y un 1 a partir de 2,75 V. (Miyara, 2004)

### **Señales analógicas**

Las señales analógicas son señales eléctricas que cambian en el tiempo de forma analógica dependiendo de la transformación física de una variable, estas pueden representarse con corriente, tensión o una carga eléctrica, las señales oscilan de manera continua entre un límite inferior y un límite superior, cuando los límites coinciden con los aceptados por determinado dispositivo, se considera que la señal está normalizada. (Miyara, 2004), este tipo de señales se ven diversidad de dispositivos, desde reproductores de música, televisores, teléfonos, por lo general estas se procesan de forma inversa es decirse pueden captar de algún medio para procesarlas de forma digital, o bien pueden ser originadas para transmitir o generar comportamientos dependiendo del caso específico.

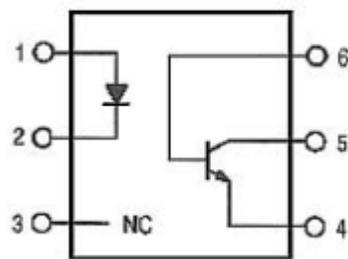
### **Optoacoplador**

Para el diseño del circuito electrónico que permitirá controlar la humedad y nutrientes del suelo, se hace uso de optoacopladores para el control de voltajes que exceden el permitido por los microcontroladores del Arduino, un ejemplo puntual es controlar las electroválvulas que permitirán activar el riego por goteo para el cultivo.

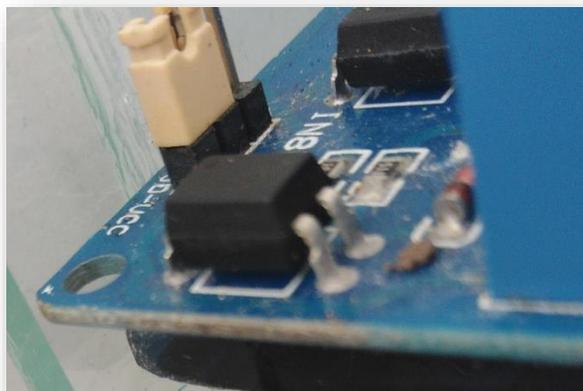
El optoacoplador es generado a partir de combinación una fuente óptica (generalmente un Led) con algún tipo de detector óptico (generalmente un semiconductor de sí) en un solo

dispositivo semiconductor o encapsulado, el resultado de esta combinación se llama opto acoplador u opto interruptor. Esta estructura permite el acoplamiento de señales de dos tipos de circuitos electrónicos independientes y totalmente aislados entre sí, según el tipo de encapsulado y componentes internos estos dispositivos pueden tener un aislamiento hasta de 3500 V. (itlalaguna.edu, 2018)

### ***Esquema Eléctrico***



*Figura 5. Ilustración diseño típico Opto acoplador (itlalaguna.edu, 2018)*



*Figura 6. Optoacoplador (autoría propia)*

El optoacoplador también llamado acoplador óptico es un dispositivo electrónico que permite a los diseñadores electrónicos un mayor grado de libertad en el diseño de circuitos y sistemas. Su funcionamiento está basado en la detección de luz emitida. La entrada del optoacoplador está conectada a un diodo emisor de luz o diodo led y la salida está conectada a un fotodetector. Estos dos elementos se encuentran separados por un aislante transparente y dentro de un encapsulado que lo aísla de la luz exterior (figura 1). Existen diferentes tipos de optoacopladores, todos ellos tienen con un diodo de luz infrarroja (LED), pero el detector puede ser fotodiodo (itlalaguna.edu, 2018). Estos dispositivos tienen variedad de aplicaciones, donde se manejan tensiones diferentes o mayores a las permitidas y que representan riesgo para los circuitos de control.

### **Relé**

El relé es un dispositivo análogo que funciona como interruptor automático y permite ser controlado mediante pulsos eléctricos. Los relés actuando como interruptores permiten abrir o cerrar un circuito eléctrico sin la intervención de la mano humana.

#### *Para que sirve un relé*

Según (tecno12-18, 2013) las aplicaciones más comunes son las siguientes.

Automatismos: permite el control para activar o detener los motores de una puerta automática, las luces de un semáforo, un ascensor, etc.

Control de motores eléctricos industriales: los relés se utilizan para arrancar, parar o cambiar el sentido de giro del motor, también permiten aumentar o disminuir la velocidad en los motores eléctricos usados en máquinas industriales de cualquier tipo.

Los primeros ordenadores funcionaban con relés: en los primeros ordenadores se usaban relés como base para la realización de los cálculos matemáticos. Posteriormente fueron sustituidos por las válvulas de vacío y más tardes por los transistores de silicio.

### ***Relés Electromagnéticos***

El inicio de la actividad de los relés electromagnéticos queda establecida en los fenómenos de seducción e incitación electromagnética. Algunos de estos radican en un disco de inducción, un núcleo electromagnético, un muelle en forma espiral, una bobina secundaria o de sombra y de una unidad de sello indicadora. (Lobato, 2018)

### **Modulo relé**

EL módulo Relé de 8 Canales Opto acoplado. Es una tarjeta de relevadores usada para la conmutación de cargas de potencia. Los relevadores están diseñados para conmutar cargas de hasta 10A y 250VAC (o 30VDC), usualmente se recomienda usar tensiones por debajo de estos límites. Las entradas del módulo se encuentran aisladas con opto acopladores para minimizar el ruido mientras se realiza la conmutación de la carga. La señal de control puede provenir de cualquier fuente de alimentación o circuito de control TTL o CMOS como puede ser un microcontrolador. Este módulo es propicio cuando se quiere conmutar cargas de corriente alterna conectadas a la red eléctrica. Este módulo Soporta todos los microcontroladores, aplicaciones industriales, control del PLC, entre otros. (teslabem, 2018)

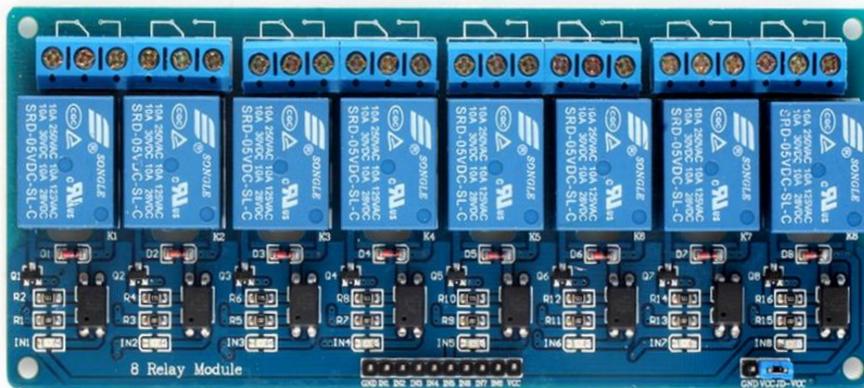


Figura 7. Modulo relé (Tienda8, 2018)

Tabla.1 Especificaciones modulo relé

Características	Especificaciones
<b>8 canales de interfaz de relé.</b>	Voltaje entrada: 5 V
<b>Cada uno necesita 15-20mA del controlador.</b>	Voltaje de control: 3.3 ~ 9 V
<b>Equipado con relé de alta corriente: 10<sup>a</sup> DC30V AC250V 10<sup>a</sup>.</b>	Voltaje de salida: 250 VCA o 30 VDC
<b>Los 8 canales están aislados ópticamente, seguro, anti-interferencias fiable.</b>	Corriente a la salida: 10 A

---

**Se puede seleccionar a través de relés puente y TTL o tierra.** Dimensiones: 139 x 55 x 17 mm

**Con el indicador de energía de carreteras, 8 vía tiene un indicador de estado.**

**Con una bobina de relé para absorber el diodo de protección.**

**El tamaño de la placa de circuito**  
**140mm X 55mm**

**Medida orificios 133mm X 40mm**

---

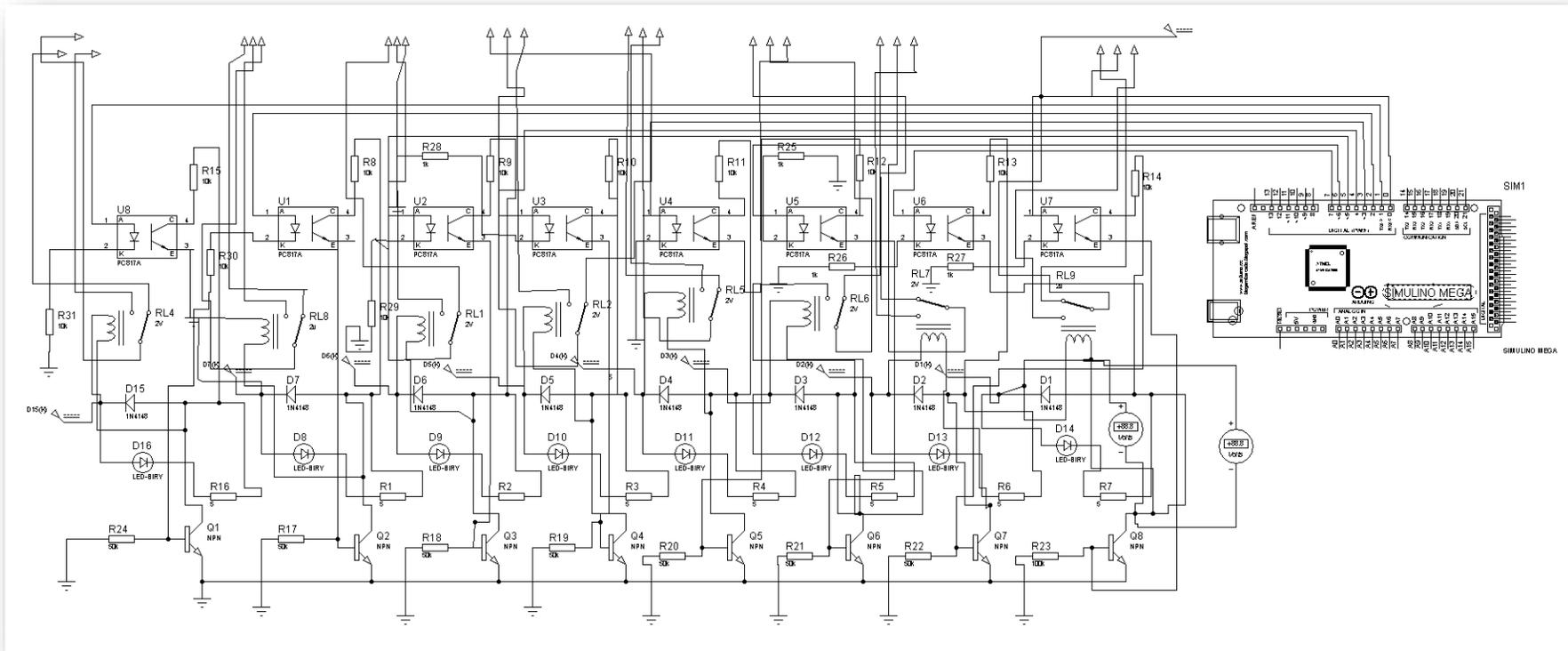
**Nota.** Recuperado de (teslabem, 2018)Características del módulo relé de 8 canales.

### **Especificaciones:**

El módulo relé es opto acoplado para cada uno de sus canales, esto con el fin de proteger los circuitos encargados de controlar y enviar pulsos digitales, recordar que los opto acopladores funcionan con emisiones de luz permitiendo aislar todo tipo de corriente que se desee controlar (itlalaguna.edu, 2018). A continuación, se describen los complementos y conexiones para cada uno respectivamente, el módulo es controlado por

Arduino, mediante pulsos digitales, específicamente para el manejo de fuentes de alimentación para actuadores, que requieren de alimentación mayor a la soportada por el microcontrolador.

- Relé 5v
- Opto acopladores
- Resistencias de 10K, resistencias de 50K
- Diodo emisor de luz
- Diodo 1N4148
- Transistor 2N3417,



*Figura 8. Diseño esquemático de módulo relé controlado mediante Arduino. Realizado en proteus 8, se describen las conexiones para su correcto funcionamiento. (autoría propia)*

## Electroválvula

Estos dispositivos construidos con el fin de controlar el flujo (on-off) de un fluido. Están diseñadas para ser empleadas con agua, gas, aire, gas combustible, vapor entre otros, estas suelen tener más de una vía asoporando hasta cinco.

Las electroválvulas de 2 vías, son implementadas frecuentemente las que funcionan en tres modalidades diferentes, esto dependiendo del uso que se les asigne; las tres modalidades pueden ser de acción directa, acción indirecta y acción mixta o combinada, también a cada una de estas modalidades o categorías se les conoce como normalmente cerrada (n/c.) o normalmente abierta (n/a.), depende del uso o la tarea que se le asigne se hace uso de electroválvulas que sean normalmente cerrada y cuando reciba el pulso o señal de control la solenoide abra durante unos segundos, o que esté abierta y cuando reciba la señal la solenoide corte el flujo. (Matarollo, 2014)

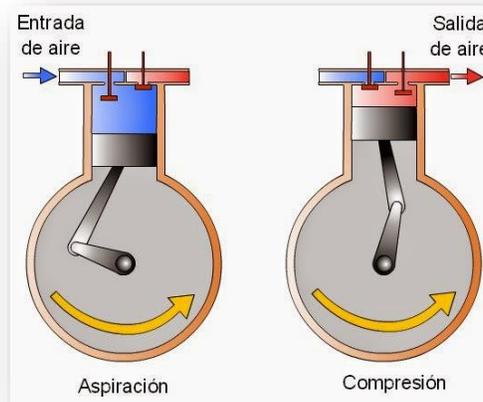


*Figura 9. Electroválvulas (Altec, 2018)*

## Compresor

El compresor es una maquina cuyo trabajo principal consiste en aumentar la presión de un fluido. El compresor eleva la presión en fluidos compresibles, lo más común es que el

elemento que se comprime sea aire, aunque también pueden comprimir gas, en los entornos industriales existen otros tipos de compresores que trabajan con diferentes tipos de gas según sea la necesidad de cada caso.



*Figura 10. Compresor alternativo de pistón (acondicionado, 2017)*

## **Arduino**

Arduino es una herramienta que surgió en Italia en el año 2005 para el desarrollo de productos y proyectos electrónicos en código abierto. Su facilidad en el uso de una tarjeta hardware y de un entorno de programación gratuito para el control de numerosas magnitudes físicas del alrededor (Arduino, Qué es arduino, 2018); Arduino es una placa desarrolladora sencilla y económica, cuenta con entradas y salidas analógicas y digitales y cuenta con un entorno de desarrollo simple que permite realizar su programación processing basado en java con una sencilla curva de aprendizaje, está herramienta facilita la creación de proyectos multidisciplinarios en la rama de la ingeniería electrónica.

## Arduino mega

El Arduino Mega 2560 es una tarjeta de desarrollo basado en el integrado ATmega2560 (datasheet). Tiene un total de 54 pines de entrada y salidas digitales, de estos 54 pines 14 pueden ser usados como salidas PWM, 16 como entradas analógicas, 4 pueden ser usadas para comunicación seria o UART, desarrollado con un oscilador de cristal de 16 MHz, conexión USB, conector de alimentación, encabezado ICSP y un botón de reset. Incluye todo lo necesario para soportar el microcontrolador; solo hace falta conectarlo a una computadora mediante cable USB plug and play o alimentarlo de forma externa haciendo uso de adaptadores de AC a DC o de algún tipo de batería. (Arduino, Arduino, 2018)

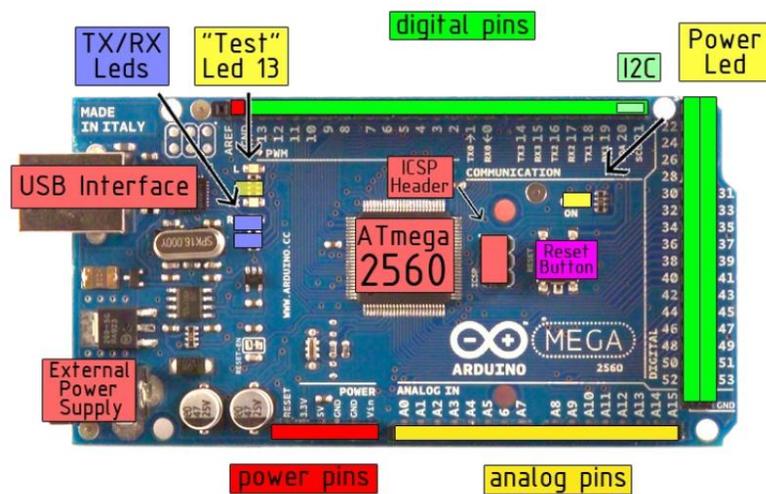


Figura 11. Placa Arduino mega (Arduino, Arduino, 2018)

### Especificaciones Técnicas Arduino

El Arduino Mega2560 puede ser alimentado mediante conexión USB o haciendo uso de una fuente de alimentación externa. Esta última es seleccionada automáticamente, La alimentación externa (no la USB) puede provenir de cualquier fuente ya sea adaptadores de

AC a DC o incluso de una batería. El adaptador puede ser conectado directamente a la placa Arduino mediante un conector de 2.1 mm en el pin Vin de alimentación de la placa. Al igual que con una batería puede insertarse los conectores en los pines Gnd y Vin del conector POWER.

La placa puede funcionar con un suministro externo de entre 6 a 20 voltios máximo. Sin embargo, se recomienda que el voltaje de entrada de la fuente externa sea entre los 7 a 12V, si la fuente externa ofrece menos de 7V el pin de 5 V podría proporcionar menos de cinco voltios y la placa puede ser inestable. Igualmente Si usa más de 12 V, el regulador de voltaje puede sufrir daños, sobrecalentarse o dañar el tablero. Por lo tanto el rango recomendado para alimentación del Arduino mega usando fuente externa es de 7 a 12 voltios (Max).

El Mega2560 difiere de todos los tableros anteriores en que no usa el chip de controlador USB a serie FTDI. En cambio, presenta el Atmega8U2 programado como un convertidor de USB a serie.

Los pines de alimentación son los siguientes:

- VIN. Es el pin de entrada para la placa Arduino cuando está usando una fuente de alimentación externa (a diferencia de los 5 voltios desde la conexión USB u otra fuente de alimentación regulada). Puede suministrar voltaje mediante este pin, o, si suministrando voltaje a través de la toma de corriente, acceda a través de este pin.
- 5V. La fuente de alimentación regulada utilizada para alimentar el microcontrolador y otros componentes en el tablero. Esta puede provenir de VIN a través de un regulador de a bordo, o puede ser suministrado por USB u otro suministro regulado de 5V.

- 3V3. Una fuente de 3.3 voltios generada por el regulador a bordo. El consumo máximo de corriente es de 50 mA.
- GND. Pines de tierra.

El Arduino Mega2560 tiene una serie de instalaciones para comunicarse con una computadora, a diferencia de otras placas de desarrollo Arduino, El Arduino mega AT Mega2560 está equipado con cuatro puertos UART de hardware para comunicación serie TTL (5V).

Un Arduino mega con el chip ATmega8U2 en la placa orienta uno de estos a través de USB y proporciona un puerto COM virtual para el software en la computadora (las computadoras con sistema operativo Windows necesitarán de un archivo .inf, pero las que funcionan con sistema operativo OS X y Linux reconocerán la placa como un puerto COM automáticamente. El software Arduino incluye un monitor serial que permite texto simple datos que se enviarán desde y hacia el tablero. Los LED RX y TX de la placa parpadearán cuando se estén recopilando datos transmitido a través del chip ATmega8U2 y conexión USB a la computadora (pero no para comunicación serie) en los pines 0 y 1). (Radionics, s.f.)

### ***Nodemcu***

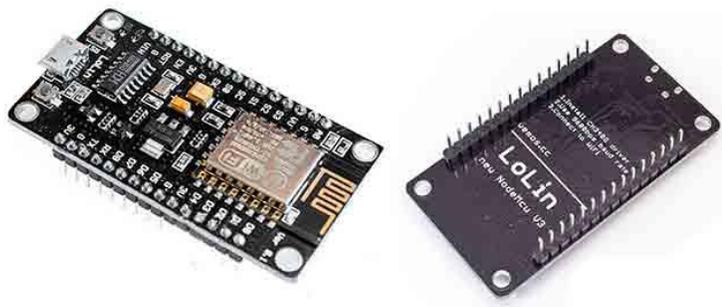
Nodemcu es una placa de desarrollo IOT basada en el microcontrolador ESP8266, incluye firmware que se ejecuta en el Soc Wifi, es una plataforma de desarrollo de código abierto (open source), que permite la comunicación inalámbrica entre un dispositivo e internet.

Nodemcu es un nombre que abarca tanto un firmware de código abierto (open source), como una placa de desarrollo basada en el chip ESP 8266. (Llamas, 2018)

La placa de desarrollo NodeMcu esp 8266, está basada en el ESP 12E, de las que existe una gran generación entre las cuales se encuentran versiones V0.9, V1.0 y V2, el Nodemcu basado en ESP 12E integra características y funcionalidades de todas las versiones anteriores y aumenta capacidades y facilidad de manejo y desarrollo. (Llamas, 2018)

Integra Características Como:

- Puerto micro USB y conversor Serie.
- Programación directa desde el IDE de Arduino.
- Alimentación mediante USB.
- Pines de conexión.
- Botón de reset y memoria flash.



*Figura 12. Placa de desarrollo Nodemcu ESP8266- 12E (Llamas, 2018)*

### **Protocolo Modbus protocolo serie**

Modbus es el protocolo de comunicación basado en la arquitectura maestro/esclavo, ampliamente usado en los sistemas SCADA por su facilidad de uso y poca necesidad de

desarrollo, este protocolo de comunicación hace uso de bloques de datos sin restricciones, en la actualidad propone ser el protocolo de mayor uso y disponibilidad en la conexión de dispositivos electrónicos industriales (logicbus, 2018).

El protocolo modbus, permite realizar comunicación serial mediante envío o recepción de bloques de datos, para esto debe programarse los dispositivos a implementar como maestro o como esclavo, puede usarse este protocolo para el control de una red de dispositivos, para la comunicación con una pc o para realizar la supervisión con unidad remota (RTU). (logicbus, 2018).



*Figura 13. Protocolo de comunicación modbus (logicbus, 2018)*

### **App inventor**

La plataforma de desarrollo de aplicaciones móviles para sistema operativo Android, que le permite a casi cualquier usuario crear aplicaciones móviles de forma simple, grafica e intuitiva.

App inventor es un entorno de programación creado para el desarrollo de aplicaciones móviles orientadas únicamente a equipos con sistema operativo Android, gracias a su programación basada en bloques permite crear desde aplicaciones simples hasta aplicaciones muy complejas. (Appinventor, 2018)

App inventor busca ofrecer a todos los usuarios, especialmente a los jóvenes, una plataforma de desarrollo de software que les permita introducirse en el mundo del desarrollo y pasar del consumo de tecnología a la creación de la tecnología. (Appinventor, 2018).



*Figura 14. MIT App inventor (Appinventor, 2018)*

### **Aplicación móvil**

El software o “programa” informático que permite a cualquier persona (usuario), con un teléfono inteligente acceder a juegos, noticias, eventos, ubicación geográfica, redes sociales, etc. A menudo se encuentran estas aplicaciones de forma gratuita y muy fácil de descargar, esto puede realizarse directamente en el teléfono móvil o desde otro dispositivo.

Una aplicación móvil es un programa que el usuario puede descargar y acceder directamente mediante su teléfono o desde otro dispositivo móvil (Tablet, reproductor mp3). (comercio, 2018)

Una aplicación móvil mejora la experiencia del usuario, permite ingresar, buscar, descubrir o simplemente leer o divertirse, todo con su dispositivo móvil, existen aplicaciones que funcionan solamente con internet y existen las que funcionan sin necesidad de internet. La mayoría de aplicaciones móviles permite al usuario navegar con una sencillez y usabilidad óptima (Multiplicalia, 2018).



*Figura 15. Aplicación móvil (Multiplicalia, 2018)*

## **Thingspeak**

Thingspeak es una plataforma open source (internet of things) y API para almacenar datos de cosas en tiempo real utilizando HTTP a través de internet o por medio de una red de área local.

Con ThingSpeak, se puede crear aplicaciones de registro de sensores, aplicaciones de

ThingSpeak es una aplicación de código abierto “Internet of Things” y API para almacenar y

recuperar datos de cosas usando HTTP a través de Internet o a través de una red de área local.

Con ThingSpeak, puede crear aplicaciones de registro de sensores, aplicaciones de seguimiento de ubicación y una red social de cosas con actualizaciones de estado

Mediante la llave o “API” de escritura o “API WRITE”, ThingSpeak permite almacenar datos enviados mediante internet a sus canales activos “Field”, en este caso para el monitoreo se activan 6 canales de almacenamiento de datos en ThingSpeak, para guardar y posteriormente leer esos datos con la aplicación móvil que se desarrolló. (Chwalisz, 2018)

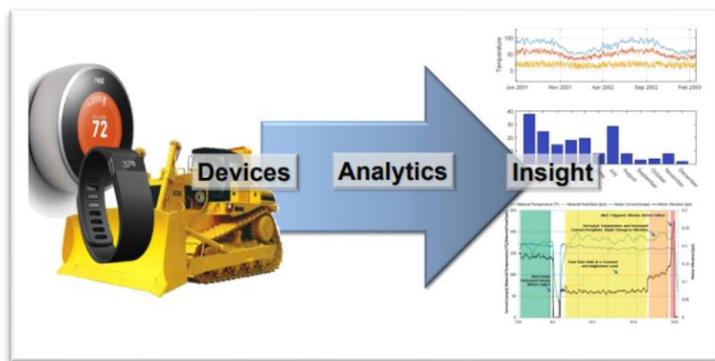


Figura 16. Esquema Thingspeak (Ning, 2017)

### Sensor DHT11

Este es un sensor de temperatura y humedad digital. Utiliza un sensor capacitivo de humedad y hace uso de un termistor para medir el aire circundante y transmite las señales por aparte mediante el pin de datos, en este caso no hay señales analógicas (dfrobot, 2017). Según el fabricante las características de este sensor se describen a continuación:

Tabla.2 Características sensor temperatura y humedad relativa DHT11.

Características	
Alimentación:	$3Vdc \leq Vcc \leq 5Vdc$
Rango de medición de temperatura	0 a 50 °C
Precisión de medición de temperatura	$\pm 2.0$ °C .

Resolución Temperatura	0.1°C
Rango de medición de humedad	20% a 90% RH.
Precisión de medición de humedad	4% RH
Resolución Humedad	1% RH
Tiempo de censado	1 seg.

---

**Nota.** Recuperado de: (Mouser.com, 2018), Características del sensor de temperatura y humedad DHT11.

### **Antecedentes**

A continuación, se presentan algunos ejemplos de sistemas similares desarrollados tanto en Colombia como en el extranjero.

#### **Antecedes internacionales**

##### ***Sistema de inyección de fertilizante con bombas de paletas y control por conductividad eléctrica en soluciones madre para fertirriego***

En 2010 el doctor ante Miguel Ángel Peña Peralta, el profesor Federico Hahn Schlam y el doctor ante José Antonio Yam Tzec realizaron el diseño de un sistema de inyección de fertilizante con bombas de paletas y control por conductividad eléctrica en soluciones madre para fertirriego con el que buscaban innovar la aplicación de fertilizantes en cultivos (Peña Peralta, 2010). Ya que según lo que expresan los autores del texto “Las técnicas actuales que usan tubos Venturi para la inyección de fertilizantes ocasionan las pérdidas de presión en el

sistema de riego presurizado por lo cual estos dispositivos no son los ideales” (Peña Peralta, 2010)

En búsqueda de una mejor opción los autores de este trabajo afirman, “este trabajo presenta un sistema de fertirriego innovador en cuanto a la forma de inyectar los fertilizantes al utilizar bombas de paletas rotatorias que ofrecen altas presiones y caudales continuos”. (Peña Peralta, 2010)

Haciendo uso de este sistema que desarrollaron buscaban agilizar y potenciar los sistemas de fertirriego o la aplicación de fertilizantes en los cultivos obteniendo grandes ventajas ante el sistema que se usaba ofreciendo una mayor potencia con el uso de bombas de paletas, a continuación, se muestra el plano del sistema de fertirriego echo por los citados autores:

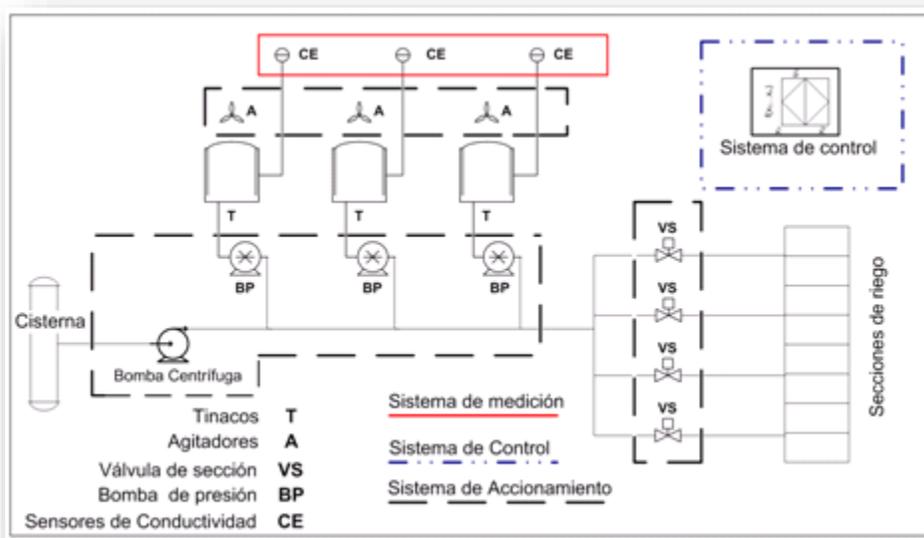


Figura 17. Sistema de inyección de fertilizante con bombas de paletas (Peña Peralta, 2010)

***Diseño e implementación de un sistema de riego automatizado y controlado de forma inalámbrica para una finca ubicada en el sector popular de Balerio Estacio.***

En mayo de 2013 los estudiantes de la facultad de ingeniería de la universidad politécnica Salesiana de Ecuador, Juan Carlos Vasconez Cuzco y Felipe de Jesús Chamba Tenemaza Realizan un sistema capaz de controlar y monitorear de forma inalámbrica En la banda de Frecuencia libre de 2.4GHz, haciendo uso del módulo Wi-Fi Xbee.

Este proyecto buscaba Reducir el tiempo y la necesidad que el agricultor este siempre en constante interacción con los cultivos, estar atento a si falta Riego a este, monitoreando de forma autónoma o automatizada el nivel de humedad en el suelo y enviando esa información hacia el Microcontrolador.

Este proyecto se creó por la necesidades presentes en la explotación del suelo y sobre todo por la constancia de permanecer en lugares donde se necesita riego constante y controlado, con el fin de que los productos en el agro no se dañen o peor aún que se pierdan en su totalidad, nace la necesidad de automatizar un sector a través de un sistema de riego, el mismo que controla las diferentes áreas de siembra el cual tienen varios sensores que interactúan con un microcontrolador que tomara las decisiones de riego en el caso de surgir la necesidad. (Vascones Cuzco & Chamba Tenemaza, 2013)

***Antecedentes locales***

***Planteamiento de un sistema de riego en el caso específico de la hacienda***

***Asturias, enfocado al desarrollo y tecnificación del campo en Colombia***

En 2011 el estudiante Lucas Mateo Tamayo Bastidas propone un proyecto con fin específico.

El proyecto de grado busca acoplar sistemas de tecnificación del campo para el caso concreto de Colombia. Buscando agregar por medio de los recursos hídricos disponibles, los nutrientes necesarios para dar la duración esperada al campo.

El país cuenta con grandes extensiones de tierra que actualmente están siendo subutilizadas por diferentes motivos, en el 2009 el aporte del campo al PIB Colombiano fue del 7%, por esto es importante optimizar los procesos para obtener su mayor productividad en aquellos espacios donde aún es posible trabajar el campo Mundial. (Bastidas, 2011)

### ***Conceptualización y necesidades de una agricultura climáticamente inteligente***

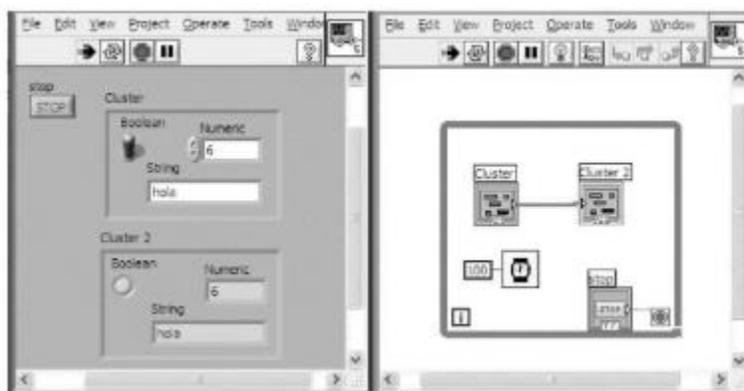
Una agricultura climática inteligente, pretende mejorar la capacidad de los sistemas agrícolas para prestar apoyo a la seguridad alimentaria, e incorporar la necesidad de adaptación y las posibilidades de mitigación en las estrategias de desarrollo agrícola sostenible

La eficiencia, desde un punto de vista reduccionista, se obtiene de la razón de las salidas (*outputs*) sobre las entradas (*inputs*) de un sistema. Para el caso de los sistemas agroalimentarios, estos son más eficientes si se aumenta la eficiencia de los recursos (generales, específicos de la producción de plantas o agrícola y de la producción animal o pecuaria), se establecen sistemas integrados y se reducen las pérdidas de alimentos y los residuos. (Soler-Tovar, 2017)

### ***Automatización y telecontrol de sistemas de riego***

Los señores Antonio Ruiz Canales, José Miguel Molina Martínez (2010), ponen a disposición de todos un escrito sobre los sistemas de Riego automatizado y telecontrol en cualquier ambiente, expresan en su trabajo un poco de historia sobre los sistemas de Riego que se han utilizado en la antigüedad y los que actualmente se implementan para los regadíos en España,

exponen la problemática y la necesidad de disminuir el desperdicio de agua en los momentos de Realizar el riego en los cultivos, también crean a partir de las consideraciones generales de ahorro de agua y tiempo, un sistema SCADA que pueda controlar y monitorear la necesidad de los cultivos y realizar el riego de forma controlada. (Ruiz & Molina., 2010).



*Figura 18. Creación de clúster de controles e indicadores (Ruiz & Molina., 2010)*

En este documento los señores Antonio Ruiz Canales, José Miguel Molina Martínez, ponen un gran énfasis en la importancia del agua para el mundo y la disminución en el desperdicio de esta con la aplicación de tecnologías actuales que permitan controlar el riego y niveles óptimos de los cultivos.

### **Sistema Automático Remoto Para El Cuidado De Cultivos Cereales (Frijol Y Arveja)**

En 2015, Nilson Gamba Castelblanco desarrolló el diseño y montaje de un sistema automático de riego por goteo, que a su vez permitiera también el riego de insecticidas o fungicidas de manera remota, para un cultivo cereal (frijol y arveja). También agrego al sistema que permitiera

gestionar tiempos de riego, visualizar y mantener un nivel de humedad en el suelo, controlar la cantidad de ciclos de riego tanto de insecticidas como fungicidas.

En este montaje Castelblanco hizo uso de sus conocimientos en bioingeniería, electrónica digital, instrumentación, etc. para realizar el montaje y control de su sistema mediante un computador en el cual se podían obtener lecturas de los sensores y accionar los actuadores en el momento necesario. (Castelblanco, 2015)

“Lo importante es lograr que los espacios de siembra sean mejor aprovechados, ya que a nivel nacional los suelos aptos para la agricultura son del 13% y solamente se utilizan el 4%”  
(Castelblanco, 2015)

## **Diseño metodológico**

Para el diseño e implementación de este proyecto se desglosa en cinco etapas, las cuales se describen a continuación:

### **Análisis y levantamiento de información:**

En esta fase se realizó la respectiva investigación exploratoria que permitió profundizar y analizar toda la información referente a los cultivos de habichuela, características y necesidades del ambiente para el óptimo crecimiento del cultivo.

También se investigó sobre el invernadero, las características y si el tiempo y lugar es propicio para el cultivo y para montaje del sistema automático de riego, fertilización y fumigado.

Posteriormente se realizó el análisis de la información obtenida con el objetivo de obtener los parámetros necesarios para la etapa de diseño.

De igual manera se realizó investigación minuciosa sobre los elementos que permitieron desarrollar la automatización de algunas de las tareas dentro del invernadero, consultando características importantes sobre los circuitos sensores y actuadores.

### **Diseño**

En esta etapa lo que se asimiló y diseño con el fin de llevar a cabo el objetivo principal del proyecto, en este caso la necesidad era automatizar un invernadero que contenía siembras de habichuela, el prototipo es a una escala asequible a pequeños agricultores, por ende las tecnologías que se utilizaran deben estar acordes a este requerimiento, específicamente se procedió a realizar la programación las tarjetas de desarrollo (Arduino), al igual que de la aplicación móvil en la que se ha realizado el monitoreo del invernadero en tiempo real, es esta

parte gracias a los CAD (diseño asistido por computadora) se logró realizar el diseño de hardware y software.

### **Desarrollo**

En esta etapa se desarrolló mediante el uso de software de programación los códigos e instrucciones lógicas necesarias para las lecturas de los sensores, tanto los de temperatura como los de humedad, también las instrucciones para los actuadores, la codificación y posterior envío de los datos recolectados hacia la aplicación móvil mediante internet y la realización de pruebas fuera del invernadero.

### **Implementación**

Posterior a la codificación o desarrollo del software de control para cada sistema, se procedió a realizar los montajes, implantación de los equipos, actuadores, controladores y sensores dentro del invernadero para realizar las pruebas, los ensayos del funcionamiento del sistema, posteriormente se realizó la ejecución de todo el sistema que funciona correctamente, aunque en algunos ensayos anteriores se necesitó realizar algunos cambios en los códigos.

### **Evaluación del sistema y resultados obtenidos**

En esta etapa ya con el invernadero acoplado, el sistema de control de riego, fertilización y fumigado funcionando correctamente, se procedió a realizar la siembra del cultivo de habichuela y se puso en funcionamiento todo el sistema para realizar el estudio y control en tiempo real del cultivo en el invernadero en forma física y vía internet observando todas las características y ventajas que el sistema ofrecer al agricultor.

Posteriormente se ha realizado la evaluación del producto obtenido dentro del invernadero automatizado verificando que se obtiene un producto de calidad y apto para el consumo.

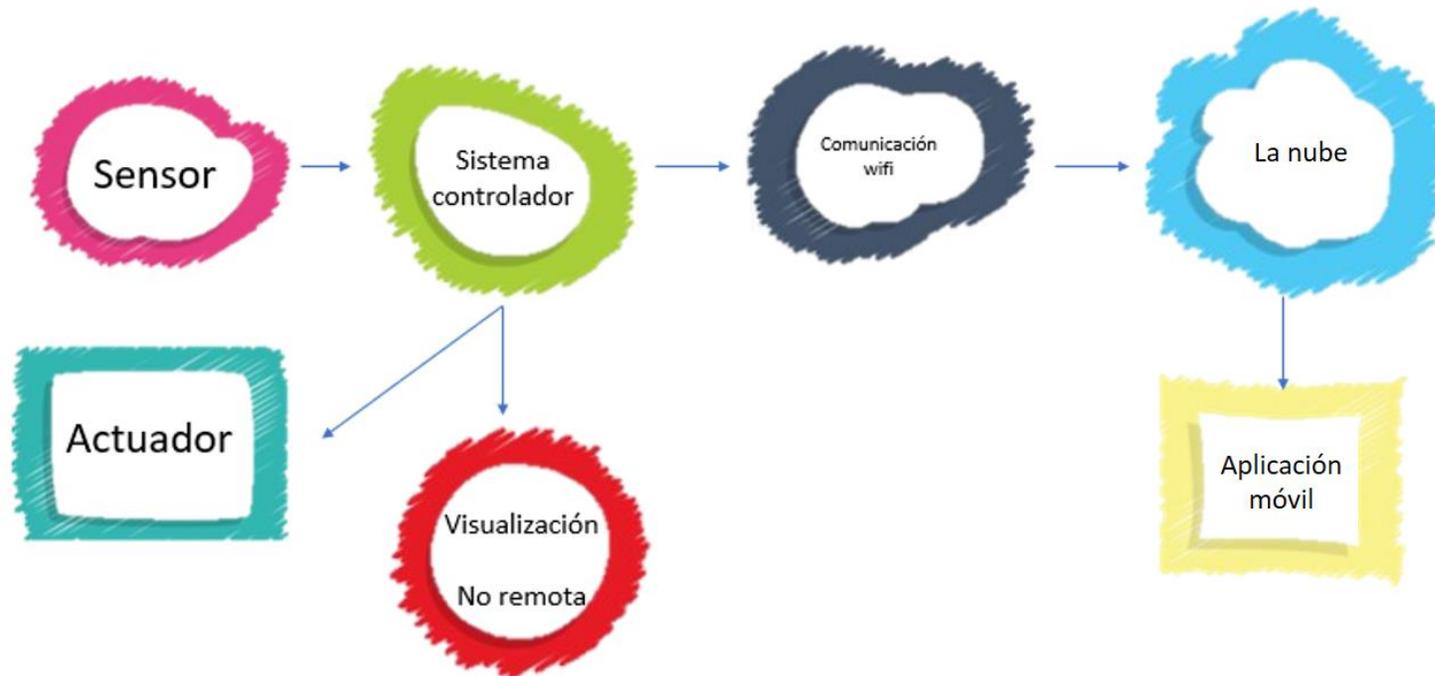
## **Ingeniería Del Proyecto**

### **Invernadero a automatizar**

Con el fin de realizar una cultivación de habichuelas y realizar la automatización de algunas de las tareas más importantes en el proceso de cultivación de judías, se realizó la estructura de un invernadero utilizando bambú o guaduas, plástico especial para invernaderos y poli sombra con el fin de proteger el cultivo de insectos y plagas, el invernadero está ubicado en la vereda Claros del municipio de Garzón Huila.



*Figura 19. Invernadero automatizado*

**Diagrama general del sistema automatizado.**

*Figura 20. Diagrama general. Se describen de forma general por partes los bloques, que se requieren para cumplir con lo propuesto en el proyecto. (autoría propia)*



## Diagrama de componentes

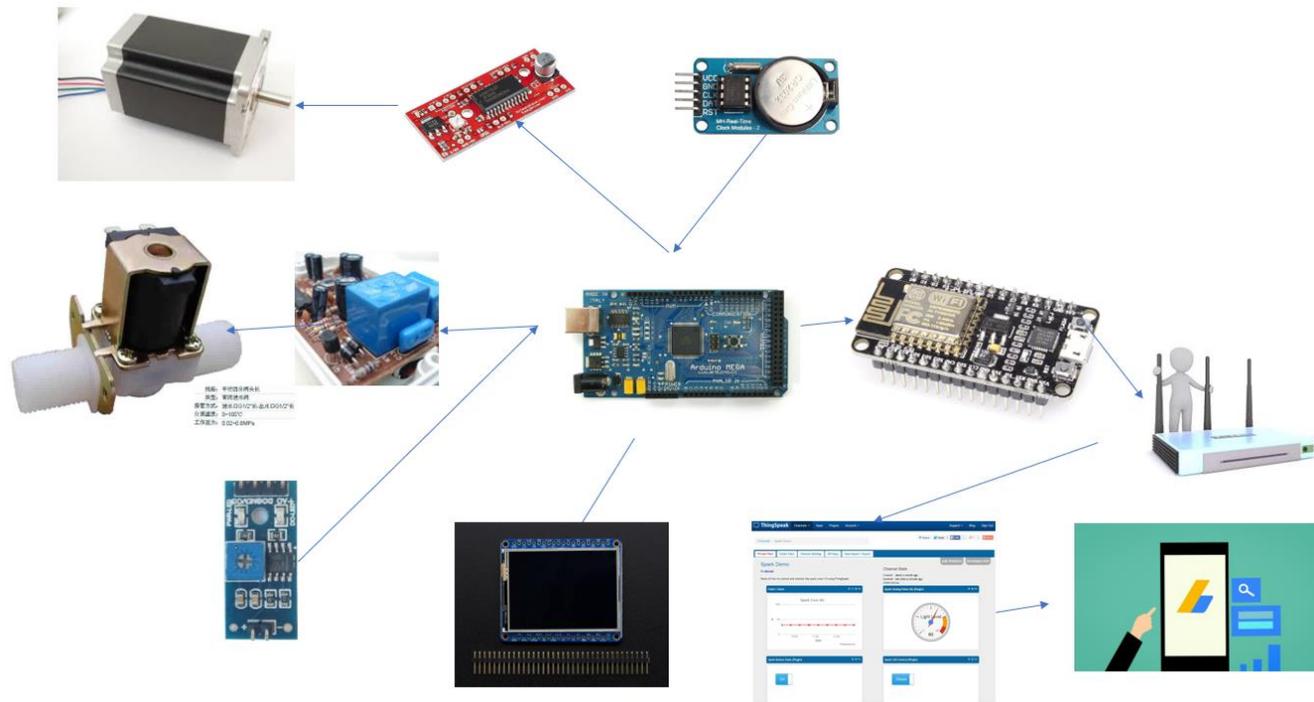
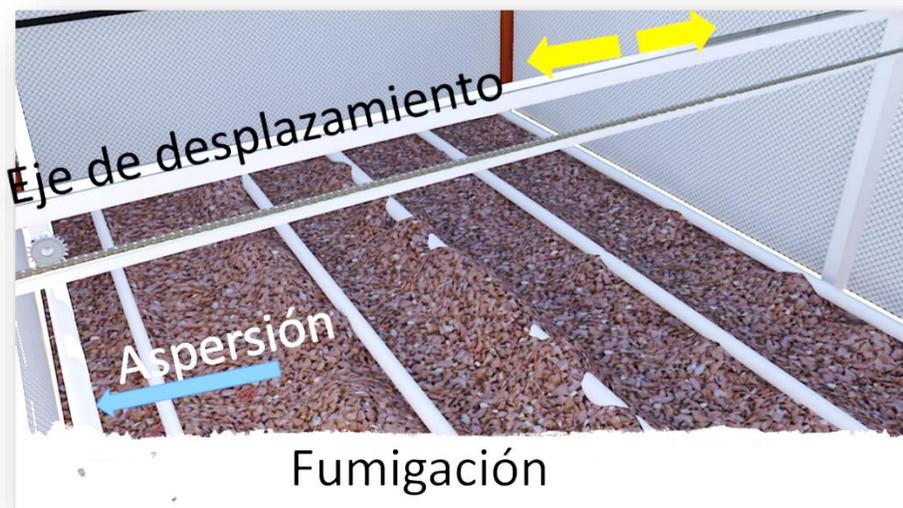


Figura 22. Diagrama de componentes (autoría propia)

### Sistema de fumigación

Para lograr la automatización de la fumigación al inicio del proyecto se implementan aspersores fijos en algunos puntos sobre la estructura del invernadero, más sin embargo de esa forma el proceso no se logró optimizar totalmente, por ende, se diseña un sistema de desplazamiento de forma lineal, que por medio de aspersores instalados sobre un tubo permite fumigar el cultivo dentro del invernadero.

Este sistema consiste en un eje de forma opuesta al invernadero, se desplazará libremente impulsado por un motor paso a paso, un piñón y una cadena ilustrado en la siguiente figura.

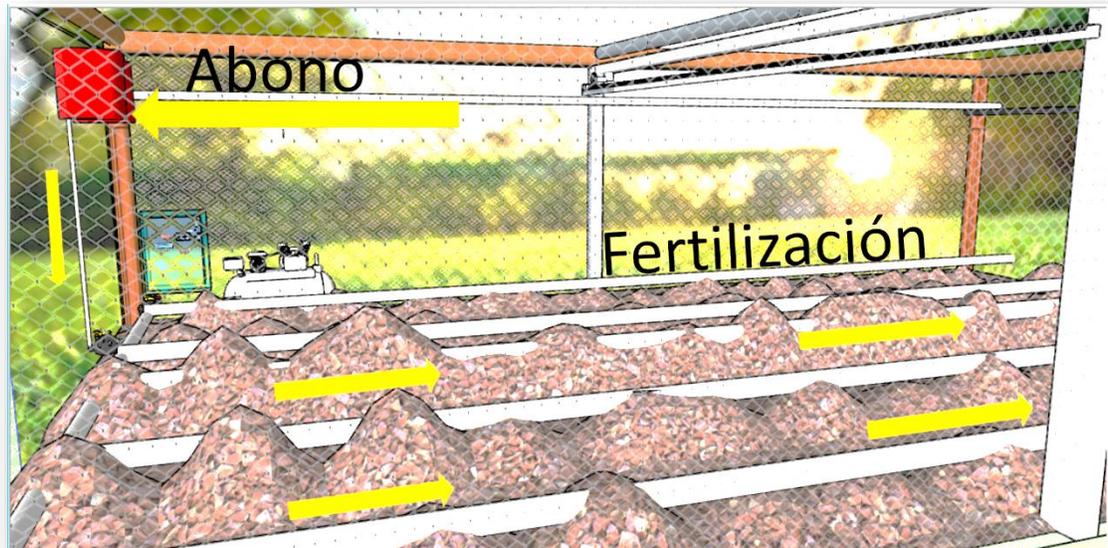


*Figura 23. Sistema de fumigación (autoría propia)*

### Fertilización

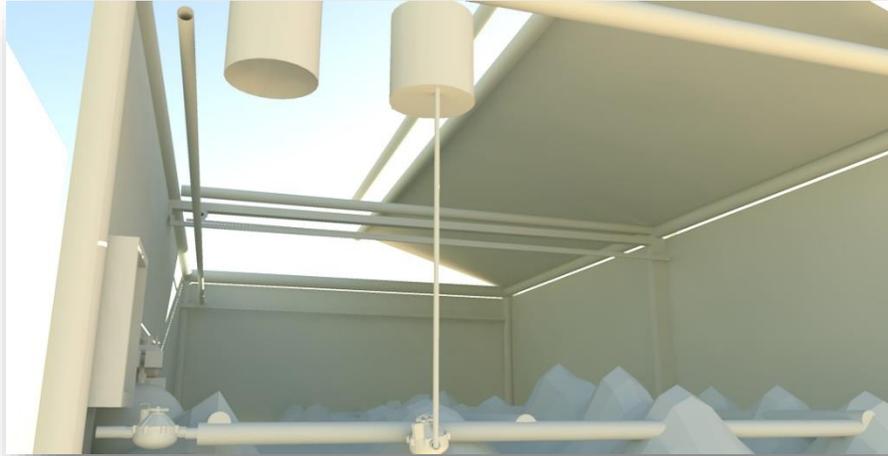
Para el proceso de fertilización se ubica un estante a dos metros de altura para situar los nutrientes, por medio de gravedad llegará a las mangueras que distribuirán el abono sobre

las plantas de habichuela figura 6, a través las mismas mangueras del riego, el proceso de fertilización por medio de goteo es un sistema que añade nutrientes al suelo de forma pausada, permitiendo tener este por un periodo de tiempo más largo sobre el suelo y que la planta los absorba de esta misma forma.



*Figura 24. Diseño de sistema de fertilización (autoría propia).*

Según lo investigado es necesario omitir el proceso de riego para, añadir abono ya que, si se deposita agua y nutrientes al mismo tiempo, estos últimos se pueden dispersar en el terreno y no ser absorbidos por las plantas de habichuelas



*Figura 25. Diseño sistema de fertilización (autoría propia)*

### **Humedad relativa**

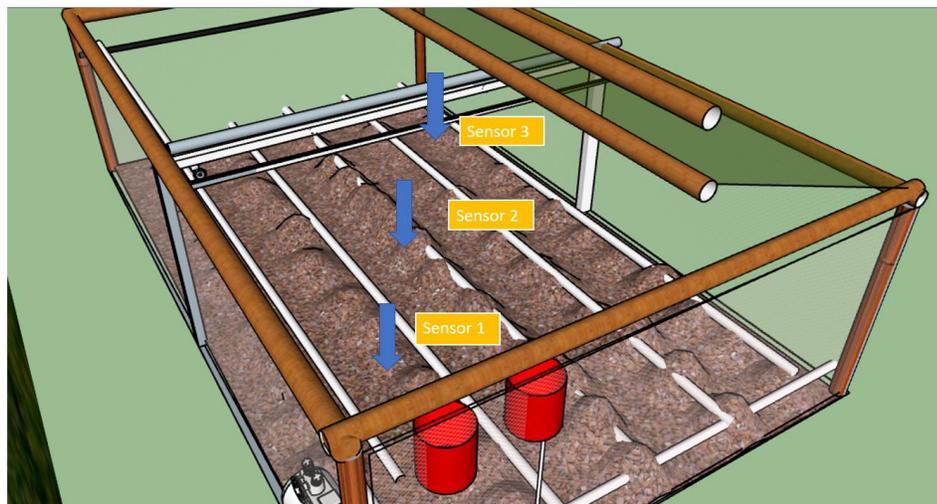
Para el diseño respecto la visualización de la humedad relativa, se mide esta variable física por medio del sensor DHT11, el cual puede medir tanto temperatura como la humedad relativa del ambiente, en un tiempo de aproximadamente 2 segundos, implementa un pin digital para el envío de datos, lo que permite evitar o disminuir ruido en las mediciones.

La humedad relativa dentro del invernadero depende de la humedad del suelo y la temperatura ambiente, por eso fue preciso mantener la humedad del suelo en un nivel óptimo, también gracias al diseño y ambiente en el que fue construido el invernadero, la humedad relativa no se hace extrema, estando siempre alrededor y dentro de los rangos óptimos necesarios para el cultivo.

## Humedad del suelo

Para el manejo de la humedad se adecuaron tres sensores que estarán repartidos sobre el invernadero con el fin de censar la humedad desde diferentes puntos como se muestra en la figura 8, posteriormente a través de las instrucciones de código desarrolladas en Arduino se promedian los valores de los tres sensores y a partir de esos valores se determina el estado del riego (apagado-encendido).

La eficiencia de este sistema está principalmente influenciada por el hecho de estar dentro de un invernadero y se puede sostener la humedad requerida para la germinación de las semillas, crecimiento de la planta y producción del producto.



*Figura 26. Ubicación de sensores de humedad sobre invernadero  
(autoría propia)*

## Sistema de riego

Los sensores de humedad van respectivamente conectados en los puertos analógicos del Arduino. La funcionalidad de este sensor es básica; el agua por sus componentes es un

material conductivo y hace que fluya la corriente a través de ella, permitiendo detectar el nivel de conductividad entre los terminales del sensor, para obtener un intervalo de la humedad y permitir automatizar el sistema de riego por medio de electroválvulas.

El sistema de riego se activa cuando el porcentaje de humedad está por debajo del 60% esto debido a que según una investigación realizada por DANE (2016) es la condición del suelo más adecuada para la cultivación de las habichuelas.



*Figura 27. Diseño de sistema de riego por goteo (autoría propia)*

## Etapa de potencia

### Sistema de apagado automático

Con el simple objetivo de mitigar el daño en los dispositivos que se encuentran en el invernadero, ya sea por recalentamiento o por tiempo de uso, se implementó un circuito que en horas de la noche desabastecerá de alimentación todos los elementos conectados.

#### *Sensor de luz*

Se implementa mediante una Fotorresistencia, que varía su resistencia de acuerdo a la cantidad de luz que absorba en su superficie, ya que la fotorresistencia al haber poca luz se polariza con una alta impedancia, mientras que con buena luz se polariza de forma inversa dejando pasar la corriente a través de ella, se implementó de esta forma aprovechando esto logrando que durante el día la fotorresistencia deje pasar la corriente y activar el sistema, mientras que al caer la noche la fotorresistencia se polariza e impide el flujo de la corriente, lo cual apaga todo el sistema electrónico del invernadero.

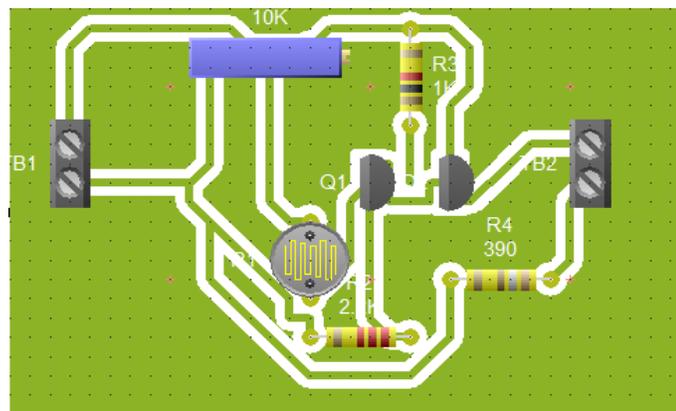


Figura 28. Esquemático Control de alimentación por fotorresistencia

(autoría propia)

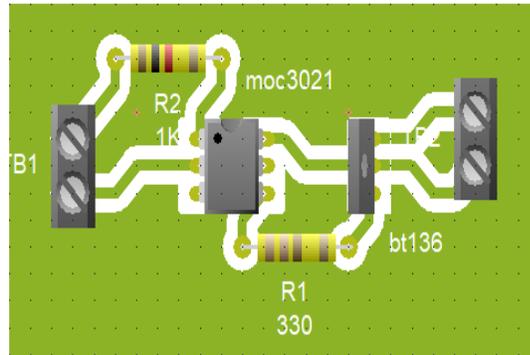
### ***Relé Digital***

El relé digital se conforma por un optoacoplador moc3021 y un triac bt136, este sistema permite aislar la etapa de potencia de la de baja potencia, en este montaje se utilizó para poder cortar la corriente del sistema, es decir cortar la corriente del sistema electrónico con el fin de apagar todo el sistema.

Según la impedancia de la fotorresistencia, se enciende o apaga el sistema, pero para poder cortar la corriente se hace necesario este relé digital.

El moc3021 es un optoacoplador que se alimenta de tan solo 5v, aislando también la etapa de potencia o alto voltaje con la de bajo voltaje, al ser alimentado enciende un pequeño led que excita un fototriac el cual permite conducir la corriente entre sus pines 4 y 6, posteriormente esta corriente alimenta la compuerta o “gate” del triac.

El triac bt136 es capaz de soportar 600v y 4Amp, lo cual es más que suficiente para soportar la corriente del sistema, el triac es capaz de conducir corriente hacia ambas direcciones por lo que es apropiado para conducir corriente alterna, esta función es la que se aprovecha del triac, al ser alimentado el moc3021, este enviara o conducirá corriente hacia la compuerta del triac el cual dejara conducir corriente alterna entre sus pines T1 y T2 (haciendo un switcheo o de interruptor), esta será la alimentación de todo el sistema electrónico de monitoreo del invernadero.



*Figura 29. Esquemático Circuito con relé digital. (autoría propia)*

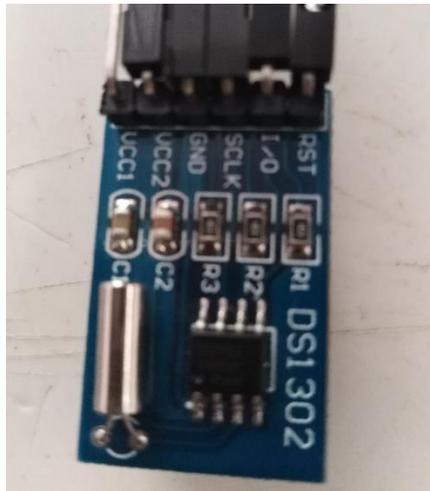
### **Elección de Arduino**

La elección del sistema embebido es sumamente importante para el desarrollo del proyecto, existen una serie de ventajas que ofrece Arduino, es una plataforma de prototipos de electrónica de código abierto, basada en hardware y software, es flexible fácil de usar, hay otros puntos a favor que favorecieron el hecho de escoger Arduino como plataforma principal, entre los que se destacan, barato, multiplataforma, entorno de programación simple y claro, código abierto y software y hardware extensible. (Arduino, Qué es arduino, 2018)

Para el desarrollo del proyecto existen una serie de requerimientos para el dispositivo o sistema embebido que será el encargado de controlar el invernadero, se considera Arduino por ser una plataforma asequible y de calidad, específicamente se decidió trabajar con la tarjeta Arduino mega 2560, tarjeta basada en el micro controlador Atmega2560, por su extendida capacidad de procesamiento, memoria instalada, además de la variedad de pines analógicos y digitales con los que cuenta hacen de la tarjeta más adecuada para satisfacer de forma unificada los requerimientos del proyecto.

### **Reloj de tiempo real (RTC)**

Para el control de las tareas del invernadero se hace necesario llevar un control en el tiempo con el fin de precisar las labores de fertilización (cada 8 días) y fumigado (cada 15 días), para ello existe una tarjeta electrónica que nos permite la conexión con Arduino; a nivel de hardware se necesita lo siguiente:

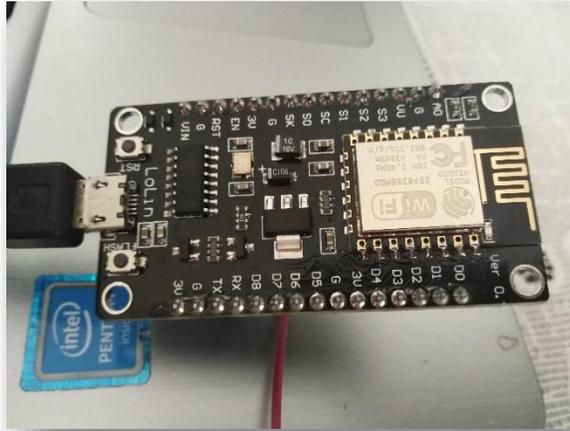


*Figura 30. Módulo RTC (autoría propia)*

### **Sistema de comunicación**

Para lograr monitorear las variables que se ven sujetas en el invernadero se debe implementar un sistema capaz de enviar datos a la internet, El módulo Esp8266 compatible con Arduino y programable con el IDE de este mismo es el medio más adecuado y fiable para realizar la conexión, por medio de su protocolo serie, se logra enviar los datos desde Arduino hacia el módulo Wi-Fi para posteriormente, a través de una red conectada a internet enviar datos a Thingspeak, plataforma de Matlab para desarrollo de proyectos relacionados con internet de las cosas.

## Node Mcu Esp8266



*Figura 31. Node Mcu Esp8266 (autoría propia)*

## Thingspeak

Para lograr comunicarse con Thingspeak hay que realizar una serie de instrucciones desde el Arduino mega que será el encargado de censar las condiciones del invernadero, posteriormente enviar datos al NODEMCU Esp 8266 y por último enviar datos a Thingspeak, que permite en tiempo real monitorear las magnitudes físicas comprometidas en el proceso de cultivación de tuberculosas bajo invernadero específicamente las habichuelas; a continuación se evidencia la simulación de envío de datos desde Arduino, Esp 8266 y Thingspeak.

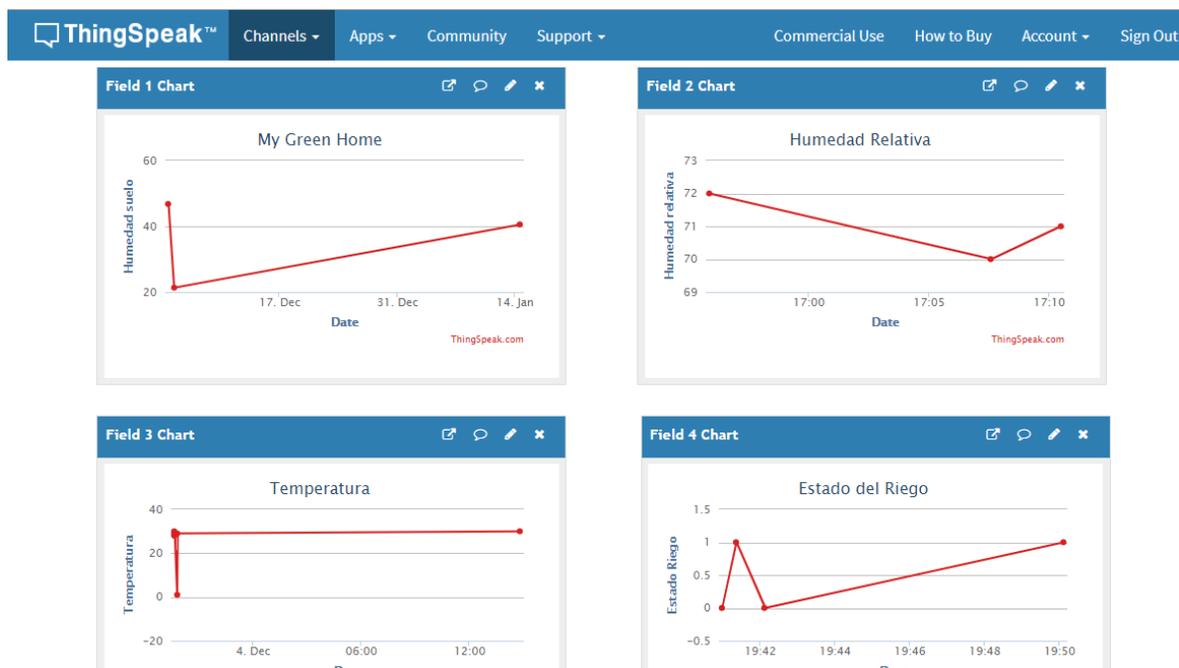


Figura 32. Envío de datos a Thingspeak (autoría propia)

### Aplicación para monitoreo

Se implementa una aplicación por medio de App inventor, en el que se muestra el estado de la humedad del suelo dentro el invernadero, la temperatura ambiente, humedad relativa y el estado de actuadores; esta aplicación estará conectada con la plataforma de las IOT - Thingspeak, siendo la encargada de almacenar datos en tiempo real.

### Entorno de desarrollo App Inventor

App inventor es una plataforma online gratuita de desarrollo de aplicaciones móviles para sistemas Android, su modo diseño y programación grafica permite que sea una de las plataformas más útiles y sencillas en el desarrollo de aplicaciones, App inventor permite realizar conexiones a internet mediante servidor web, haciendo uso de esta característica y mediante la llave de lectura de ThingSpeak, se logra leer los valores de cada uno de los seis

canales donde están almacenados los datos recogidos por el sistema automatizado y mostrarlos en tiempo real en la aplicación.

En el desarrollo de la aplicación se diseñan 4 botones, mediante los cuales al ser presionados enviarán la solicitud de “lectura de datos” a ThingSpeak en canales específicos donde se encuentran almacenados los datos requeridos.

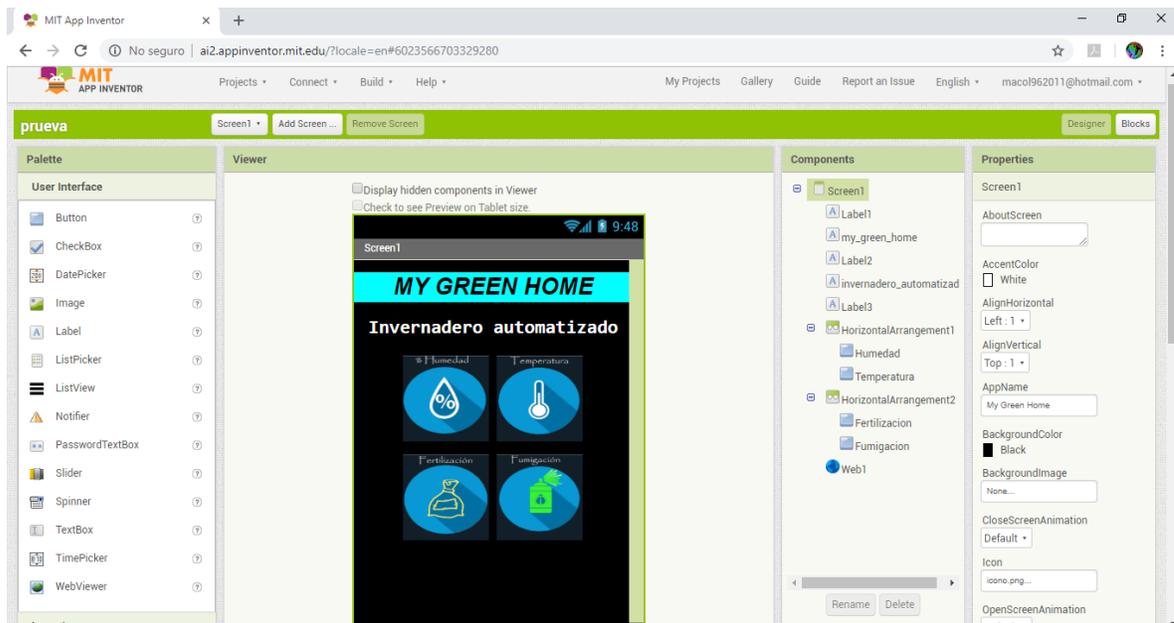


Figura 33. Diseño de aplicación móvil (Autoría propia)

## Diagrama de flujo funcionamiento de la aplicación:

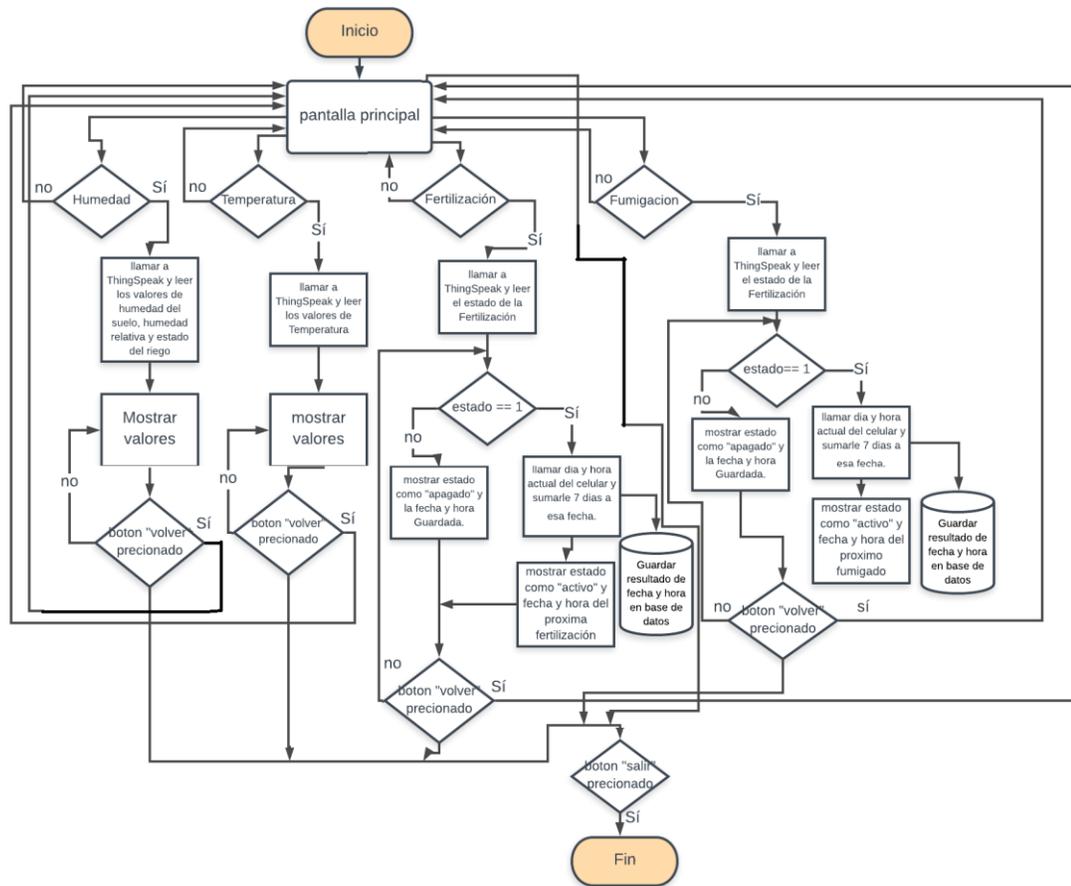


Figura 34. Diagrama de flujo MY GREEN HOME. (Autoría propia)

## Componentes y conexiones

Existen diversos tipos de conexiones que se necesitan para llevar a cabo el funcionamiento del sistema automatizado, desde señales digitales, analógicas y hasta voltajes de 110 voltios. La conexión entre Arduino mega y nodeMCU esp8266 se hace por medio del protocolo serial a 115200 baudios. Para la comunicación entre Arduino y algunos componentes se realiza la comunicación mediante protocolo I2C, entre los que se encuentran, easy driver v4 tarjeta encargada de velocidad y dirección de giro del motor

paso a paso, reloj RTC, encargado de alojar la fecha actual, sensor DHT11, que tiene el trabajo de censar humedad relativa y temperatura del ambiente.

Para el funcionamiento del easy driver v4 se utiliza una fuente de 12 voltios de corriente directa y aun amperaje de 1.5 esta fuente es con la que funciona el motor y la tarjeta que lo controla, ya que está trae circuitos de potencia que regulan la corriente, la velocidad y dirección se controla por medio puertos digitales, trabajando con el protocolo I2C; los tres sensores que se ubicaron en el suelo trabajan con señales analógicas que oscilan en los cinco voltios, el sensor DHT11 comunica por medio del protocolo I2C por medio de puertos digitales.

Las conexiones y los protocolos empleados se describen a continuación.

## Diagrama conexionado de Arduino

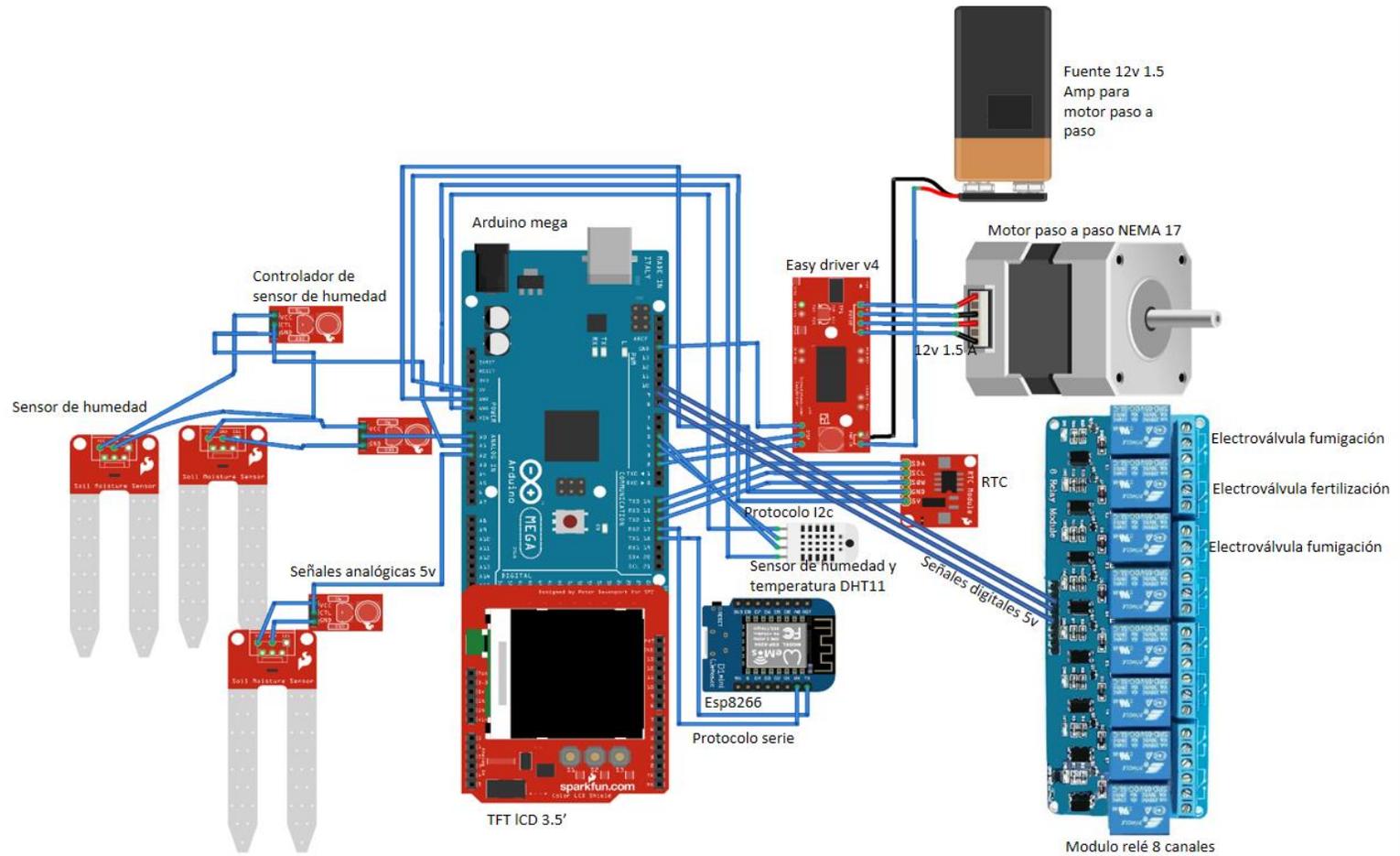


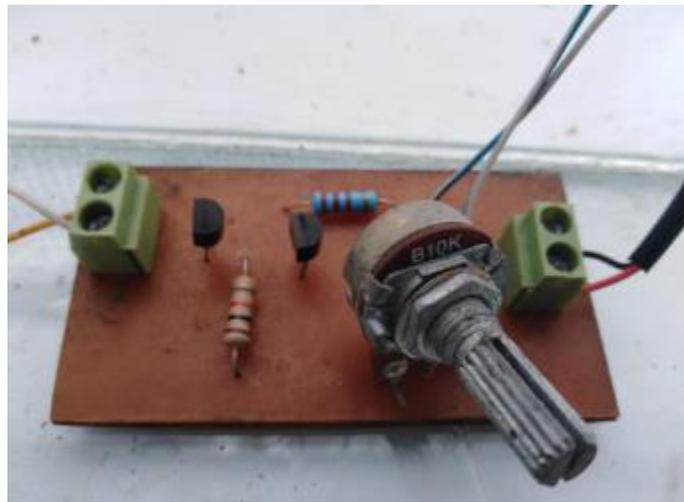
Figura 35. Diagrama conexionado de Arduino (autoría propia)

## Desarrollo del sistema automatizado

### Circuitos.

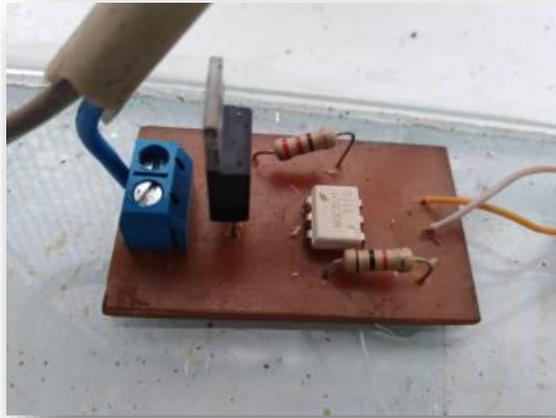
#### Sistema de apagado automático

Se implementó el sistema que permite en horas del día mantener todo el sistema electrónico funcionando y en horas de la noche el sistema se apaga, a continuación, se describe el circuito que circuito con relé digital. Este circuito es el encargado de cortar o dar continuidad al flujo de corriente para todos los circuitos dentro del invernadero.



*Figura 36. Montaje Real Control de alimentación por fotorresistencia*

*(autoría propia)*



*Figura 37. Montaje Real relé digital (autoría propia).*

## **Descripción de software y comunicación**

### **Comunicación Arduino – Nodemcu**

El protocolo de comunicación implementado en la comunicación entre el Arduino mega 2560 y el NODEMCU ESP 8266- 12E, es el protocolo Modbus RTU, que es un protocolo que permite la comunicación serial entre dos dispositivos, enviando buffer de datos de forma ordenada y continua entre los dos dispositivos, este protocolo de comunicación tiene la capacidad maestro / esclavo, lo cual permite enviar los datos de forma que el nodemcu pueda ejecutar las acciones correspondientes en forma ordenada, leer los datos conforme son enviados por el Arduino y hacer el envío de datos a ThingSpeak, antes de realizar una nueva lectura del buffer.

Arduino y nodemcu están sincronizados para hacer envío y lectura de datos, cada 15 segundos el Arduino genera un buffer de datos con todos los datos adquiridos por los sensores y los dispositivos dentro del invernadero, el nodemcu realiza la lectura de estos

datos en el orden que los envía el Arduino, los guarda y posterior mente realiza el envío de estos datos a ThingSpeak, lo cual tiene un tiempo de aproximadamente 15 segundos en el envío de datos desde nodemcu a ThingSpeak.

La programación de la aplicación se desarrolla en el entorno de app inventor, básicamente la programación se hace mediante una base gráfica, que se describe a continuación:

### Desarrollo de aplicación en app inventor

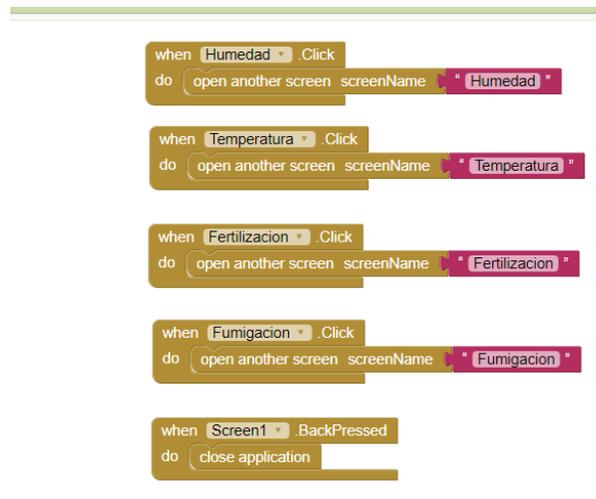


Figura 38. Código pantalla principal App inventor (autoría propia)

```

when Humedad.Initialize
do
  set Web1.Uri to "https://api.thingspeak.com/channels/620423/field..."
  set Web2.Uri to "https://api.thingspeak.com/channels/620423/field..."
  set Web3.Uri to "https://api.thingspeak.com/channels/620423/field..."
  call Web1.Get
  call Web2.Get
  call Web3.Get

when Web1.GotText
uri responseCode responseType responseContent
do set humedad_suelo.Text to join [get responseContent] [%]

when Web2.GotText
uri responseCode responseType responseContent
do set humedad_relativa.Text to join [get responseContent] [%]

when Web3.GotText
uri responseCode responseType responseContent
do if [get responseContent] = 1

```

Figura 39. Código pantalla humedad del suelo, relativa y estado del riego App inventor (autoría propia)

```

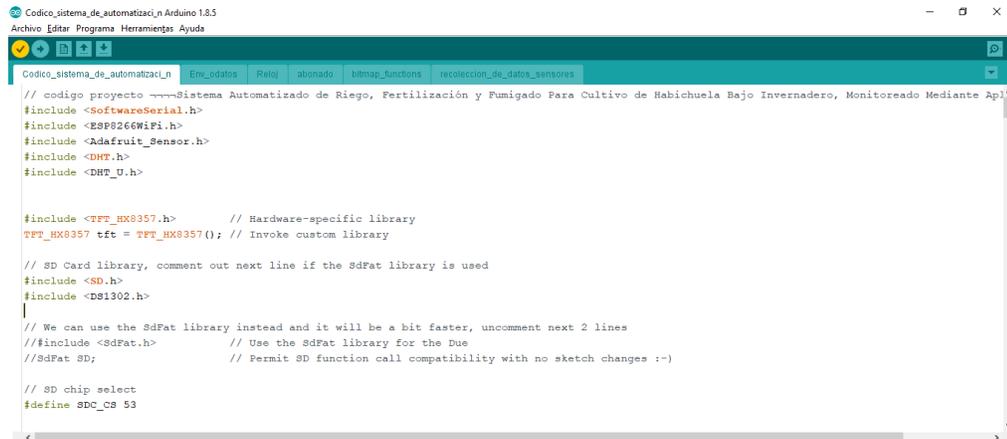
when Fertilizacion.Initialize
do
  set Web1.Uri to "https://api.thingspeak.com/channels/620423/field..."
  call Web1.Get

when Web1.GotText
uri responseCode responseType responseContent
do
  if [get responseContent] = 1
  then set estado_fertilizacion.Text to "ACTIVO"
  if [get responseContent] = 0
  then set estado_fertilizacion.Text to "APAGADO"
  if [get responseContent] = 1
  then
    set fecha_fertilizacion.Text to
      call Clock1.FormatDateTime
      instant call Clock1.Add Days
      instant call Clock1.Now
      quantity 7
      pattern "MM/dd/yyyy hh:mm:ss a"
    call TinyDB1.StoreValue

```

Figura 40. Programación de fertilización app inventor (autoría propia)

El código de programación para las tarjetas que estarán encargadas del control y comunicación dentro del invernadero se ha desarrollado mediante el entorno del Arduino IDE, toda la programación se ha dividido en diferentes archivos como se describe en la siguiente imagen.



```

Codigo_sistema_de_automatizaci_n_Arduino_1.0.5
Archivo Editar Programa Herramientas Ayuda
Codigo_sistema_de_automatizaci_n
Etu_datos Reloj abonado Irrigacion_funciones recepcion_datos_sensores
// codigo proyecto Sistema Automatizado de Riego, Fertilización y Fumigado Para Cultivo de Habichuela Bajo Invernadero, Monitoreado Mediante Ap
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#include <TFT_HX8357.h> // Hardware-specific library
TFT_HX8357 tft = TFT_HX8357(); // Invoke custom library

// SD Card library, comment out next line if the SdFat library is used
#include <SD.h>
#include <DS1302.h>
|
// We can use the SdFat library instead and it will be a bit faster, uncomment next 2 lines
// #include <SdFat.h> // Use the SdFat library for the Due
// SdFat SD; // Permit SD function call compatibility with no sketch changes :-|

// SD chip select
#define SDC_CS 53

```

*Figura 41. Programación de códigos en Arduino IDE(autoría propia)*

### **Código principal Arduino**

En el código principal para Arduino mega, desarrollado en el entorno de Arduino IDE en esta sección del código se añaden todas las librerías, las configuraciones para el resto de los códigos, además se hace la configuración de la LCD TFT que permite la visualización no remota del invernadero y también desde este código se llaman el resto de los códigos, véase en anexo 1.

### **Código envío de datos a NODEMCU ESP8266**

En el envío de datos hacia el módulo wifi, nodemcu esp8266, se implementó la comunicación serial entre el Arduino mega y el nodemcu, para esto se hizo uso del protocolo de comunicación Modbus, que se describió anteriormente, este permite enviar o recibir un bus de datos sin restricciones, para ello se necesita programar los dispositivos a usar como maestro o esclavo, en este caso el Arduino mega como maestro y el nodemcu como esclavo. Ver anexo 2.

### **Código de programación para sistema de fumigación**

En este caso se utiliza la librería del RTC, el cual provee la fecha actual y por medio de la fijación de una fecha a futuro se realiza el proceso de fumigación, además se encienden el puerto digital encargados de activar los relés, que alimentara las electroválvulas y se envía pulsos para la tarjeta easy driver versión 4 que es la encargada de alimentar y controlar el motor paso a paso para la fumigación, esto se describe en el anexo 3.

### **Código de programación para sistema de fertilización**

Este código es similar a la fumigación se inserta una fecha futura para realizar la fumigación y en este caso se enciende el pin digital para la activación del relé encargado de alimentar la electroválvula que da paso al fertilizante, como se puede observar en el anexo 4.

### **Código funciones para Bitmap**

Este código está estrictamente relacionado con el primero ya que es el complemento de las funciones para mostrar las imágenes en la pantalla, en esta se especifica el tamaño de la LCD, la conversión del bmp al tipo de formato aceptado para la visualización y posteriormente se envían los colores por líneas para cada pixel, véase en el anexo 5.

### **Código de programación para recolección de datos de sensores**

En este código básicamente lo que se hace es la recolección de los datos de los sensores (DHT11, HY-06) para posteriormente procesar esa información y de acuerdo a los datos obtenidos proceder a enviar los datos al Node Mcu esp8266 por medio del protocolo serial, como se describe en el anexo 6.

### **Código de programación para recepción de datos y envío a thingspeak**

En esta parte del código de programación se realiza la parte del censado de las variables dentro del invernadero, que permiten a partir de los valores activar los actuadores. En esta misma etapa se envían los datos a Thingspeak, que luego son enviados a la aplicación móvil, el código se describe en el anexo 7.

### **Visualización no remota mediante TFT LCD**



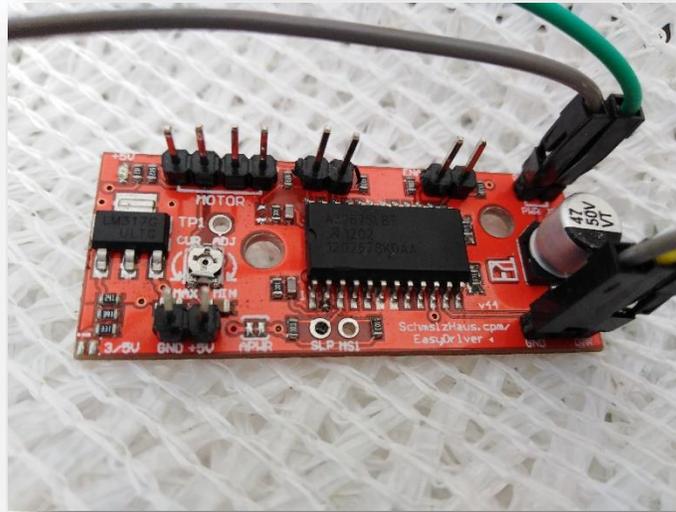
*Figura 42. Pantalla TFT para visualización de datos (autoría propia)*

### Sistema de fumigación

Para la adecuación del sistema de fumigación se realizaron una serie de configuraciones para tratar de llevar a cabo lo que se prepuso en la etapa de diseño, realizando los cambios necesarios en el recorrido y los tiempos de encendido, pausas en los recorridos y el apagado.



*Figura 43. Sistema de desplazamiento para fumigación (autoría propia)*



*Figura 44. Tarjeta encargada de controlar el motor paso a paso (autoría propia)*



*Figura 45. Boquilla para fumigación (autoría propia)*



*Figura 46. Electroválvula para fumigación (autoría propia)*

### **Sistema de fertilización**



*Figura 47. Estante de almacenamiento de fertilizante (autoría propia)*



*Figura 48. Electroválvula para fertilización (autoría propia)*

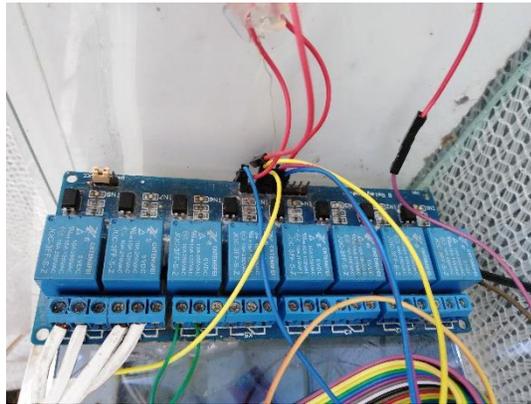
## **Control de humedad del suelo**

### **Sensor de humedad del suelo HL-69**



*Figura 49. Sensor de humedad instalado (autoría propia)*

En este caso se requiere del uso de corrientes y voltajes que están por encima de lo soportado por las tarjetas de control, para ello se adecuó un módulo relé que por medio de optoacopladores y relés nos permite hacer un sistema de onn-off para activar actuadores cuando se requiera.



*Figura 50. Implementación de modulo relé en invernadero (autoría propia)*

### **Adecuación de electroválvula**



*Figura 51. Electroválvula para riego (autoría propia)*

### **Implementación de riego por goteo**

El riego por goteo es un sistema muy eficiente para cualquier tipo de cultivo, según una investigación realizada por La universidad nacional abierta y a distancia, el riego por gravedad es el más aconsejable para las habichuelas en terrenos planos o de poca pendiente, siempre que se disponga de agua suficiente y pueda ser conducida por los surcos dentro de los cultivos, tanto de tipo arbustivo como voluble. Por su parte, el riego por goteo es el más recomendado en condiciones de baja disponibilidad de agua y debido a las ventajas que ofrece. (DANE, 2016)

Se implementa el riego por goteo con la ventaja de poder controlar está variable ya que solo se agrega agua de esta manera y debido a que es cosechado dentro de un invernadero se cohibe de fuentes de agua externas, permitiendo ser eficiente el sistema de riego por goteo y mantener constante la humedad según las especificaciones que se comparten por (DANE, 2016)



*Figura 52. Pruebas de funcionamiento del sistema de riego por goteo (autoría propia). Desarrollo y funcionamiento del sistema automatizado para invernadero*

### **Adecuación de sensor de humedad relativa y temperatura (DHT11)**

Se instala el sensor DHT11 en la parte central del invernadero con el objetivo de censar la humedad relativa y temperatura ambiente.



*Figura 53. Adecuación de sensor DHT11 (autoría propia)*

### **Funcionamiento del sistema**

#### **Visualización no remota**

En la visualización no remota que consistía en la LCD que permitía monitorear datos desde el invernadero, se logró obtener datos de humedad del suelo, humedad relativa, temperatura, estado de la fumigación, estado del riego y estado de la fertilización.

El funcionamiento del monitoreo de forma no remota se puede observar en el siguiente video.

<https://youtu.be/7xXdU16fqw0>

## **Fumigación**

Se implementa un sistema capaz de realizar el proceso de fumigación de forma rápida y eficiente, este sistema consta de un motor paso a paso y guías de rodamiento que permite al motor desplazar las boquillas de fumigación por todo el invernadero, permitiendo que todo el cultivo sea fumigado completamente, para el control de este proceso se han programado fechas específicas en las que se considera conveniente para realizar la fumigación.

Gracias al ambiente en el que se implementó el Sistema Automatizado de Riego, Fertilización y Fumigado, no se tuvo la necesidad de realizar un plan de fumigación constante, los nutrientes del suelo y niveles de temperatura y humedad mantuvieron el cultivo en crecimiento siempre muy sano, no hubo presencia ni indicios de enfermedades en el cultivo, por lo cual el sistema de fumigado se mantuvo en estado preventivo y monitoreado por si en algún momento se hacía necesario encenderlo.

Aunque no fue necesaria la fumigación del cultivo durante su crecimiento las pruebas que se realizaron para este montaje se describen en los siguientes videos.

<https://youtu.be/SKQ8sE1bqE0>

[https://www.youtube.com/watch?v=hbXx5\\_uEVaQ](https://www.youtube.com/watch?v=hbXx5_uEVaQ)

## **Riego**

Se hace uso del sensor de humedad de suelo YL – 69 para mediciones en la humedad del cultivo, en este caso se han dispuesto tres sensores los cuales son promediados sus medidas para obtener un valor global en la humedad del suelo en el cultivo, según los estudios

hemos programado el nivel óptimo de humedad en 60%, si se detecta un nivel menos a este el sistema de riego será activado automáticamente.

En el siguiente video se puede ver el proceso del riego por goteo.

<https://www.youtube.com/watch?v=vnA6Lojl0wk>

### **Fertilización**

El sistema de fertilización está programado al igual que con el de fumigación por fechas en las que es necesario realizar la fertilización en el cultivo de habichuelas, para este caso se realizó cada ocho días, por medio de la electroválvula el fertilizante que fue previamente preparado y almacenado en el tanque, de forma que mediante gravedad se pueda dirigir el abono a las plantas a través de las mangueras del riego.

En el siguiente video se puede apreciar el proceso de fertilización del cultivo.

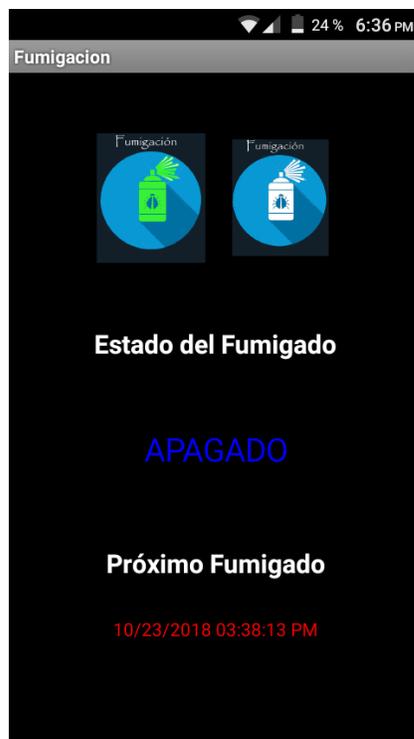
<https://www.youtube.com/watch?v=N2hSKqEp41U>

### **Monitoreo**

El sistema de monitoreo está basado en las tecnologías de las IOT, haciendo uso de módulos wifi para la transmisión de datos hacia el almacenamiento basado la nube ThingSpeak, del cual posteriormente podrán ser leídos los datos mediante la aplicación móvil que se ha desarrollado.

El funcionamiento de la aplicación se demuestra en el siguiente video:

<https://www.youtube.com/watch?v=tJmqw0DLENo>



*Figura 54. Aplicación puesta en marcha (autoría propia)*

## **Etapas de crecimiento y producción**

### **Germinación**

Durante este proceso se añadió solamente agua y se pudo evidenciar que germinaron todas y cada una de las semillas que se plantaron.



*Figura 55. Proceso de germinación planta de habichuela (autoría propia)*



*Figura 56. Germinación de planta de habichuelas (autoría propia)*



*Figura 57. Planta de habichuela recién germinada (autoría propia)*

Ya desde este momento se inicia el proceso de fertilización, añadiendo potasio y calcio, al suelo por medio del sistema de riego por goteo.



*Figura 58. Planta de habichuela recién germinada (autoría propia)*

### **Crecimiento**



*Figura 59. Planta en crecimiento (autoría propia)*



*Figura 60. Planta en crecimiento (autoría propia)*

### **Desarrollo**

Se evidencia las plantas a punto de producir, se desarrollan muy sanas.



*Figura 61. Plantas de habichuela desarrolladas (autoría propia)*

**Producción**

*Figura 62. Plantas florecidas (autoría propia)*



*Figura 63. Inicio de producción de habichuelas (autoría propia)*

### **Producto final**

Las habichuelas que se obtuvieron demuestra la funcionalidad del sistema, ya que las vainas fueron sanas y su tamaño comparado con las habichuelas que se obtienen en los cultivos convencionales es mayor.



*Figura 64. Producto final (habichuelas) (autoría propia)*

### **Análisis e Interpretación de Resultados**

El producto final de la implementación del Sistema Automatizado de Riego, Fertilización y Fumigado en el cultivo de Habichuela Bajo Invernadero, dieron excelentes

resultados, un cultivo de habichuela sano y apto para el consumo que no tuvo problemas, enfermedades ni ninguna afección en su crecimiento.

El producto final es una habichuela sana y que sin ningún problema podría llevarse a la venta y al consumo humano, lo que significa que la automatización de todos los procesos, los niveles y códigos programados, funcionaron correctamente, las investigaciones realizadas y los datos científicos de temperatura, humedad, humedad relativa, etc. En fin todas las características y necesidades del cultivo de habichuela en los cuales se centraron los códigos, son correctos y el cultivo puede crecer en este entorno, sin la constante necesidad de la mano humana o del agricultor.

El resultado obtenido es exactamente el producto esperado en el planteamiento de este proyecto, se buscaba comprobar que con las tecnologías actuales se puede automatizar un cultivo, en este caso de habichuela y que el resultado final, el producto es igual al cultivado de forma tradicional, pero a diferencia de los cultivos tradicionales en los que se hace necesaria la presencia del agricultor muy constantemente, en el cultivo automatizado esa necesidad se disminuye en hasta un 70%, siendo el otro 30% la necesidad de limpieza ocasional, preparación de los productos de fertilización y fumigado, pero no es necesario que el agricultor los aplique a la planta como en el cultivo tradicional, solamente se necesita de cambiar periódicamente estos productos en los tanques de almacenamiento del sistema ya que al paso del tiempo estos productos suelen vencerse y podría perjudicar el cultivo si se les agregan estos productos vencidos.

El sistema de Monitoreado Mediante Aplicación Móvil, aumenta la fiabilidad del sistema y brinda una mayor tranquilidad y facilidad al agricultor, ya que todos los datos que

necesita para verificar el estado de su cultivo, se encuentran siempre al alcance de sus manos, directamente en su teléfono móvil, estando toda esta información directamente almacenada en una base de datos “ThingSpeak”, el usuario puede acceder en cualquier momento y desde cualquier lugar desde su teléfono móvil y verificar el estado de su cultivo, lo cual hace todo mucho más fácil que en un cultivo tradicional.

## **Impactos**

La implementación de este proyecto abarca casi todas las necesidades del cultivo, dejando fuera solo algunas tareas que necesariamente debe hacer el agricultor, la implementación de este sistema automatizado brinda una fiabilidad y facilidad de control sobre el cultivo a los agricultores.

Aunque el montaje del sistema parece un poco costoso a corto plazo, realmente brinda grandes beneficios ya que reduce los tiempos necesarios del agricultor dentro del cultivo, brinda al usuario toda la información necesaria en su teléfono móvil y al mismo tiempo mantiene el cultivo en los niveles óptimos para su crecimiento, por lo cual son grandes ventajas en su aplicación.

Con las tecnologías actuales el montaje de este proyecto se hace algo posible de implementar y aporta a los avances tecnológicos en los cultivos automatizados, a los conocimientos en las IOT y a los mismos agricultores que salen beneficiados de este sistema, como ya se dijo, en un corto plazo el sistema se hace un poco costoso, pero en un mediano o largo plazo aparte de brindar los beneficios que se enumeraron anteriormente, brinda una beneficios en ahorro de agua y fertilizantes, al igual que podría reducir el costo de producción en este plazo.

El sistema empleado no crea dificultades en ningún sentido por lo que puede ser implementado para diferentes tipos de cultivo y en cualquier lugar que se desee, solo con realizarle algunas modificaciones a los códigos fuente para que se adapte al cultivo deseado y talvez algunas modificaciones al hardware (sensores, actuadores, etc.) si se quiere estructurar a mayor escala o si se quiere utilizar componentes más precisos o agregarle otra función al sistema.

## **Verificación de Objetivos**

Dentro de los objetivos planteados para el desarrollo de este proyecto, se encuentra la investigación, desarrollo, implementación y ejecución del sistema, con el fin de comprobar si el sistema funcionaba correctamente y si el cultivo se gestaría y daría su cosecha tal como en un cultivo tradicional, para esto al principio del planteamiento de este proyecto se expusieron los siguientes objetivos los cuales se buscaba cumplir mediante la implementación del sistema:

### **Investigar parámetros y condiciones para cultivo de habichuelas bajo invernadero**

En las investigaciones documentadas en el marco teórico y conceptual podemos determinar el cumplimiento de este objetivo, además de la adquisición de los datos necesarios que han sido parte de estudios científicos y que muestran las necesidades del cultivo, también se recolectó la información necesaria sobre invernaderos, sensores a implementar, actuadores y todas las características de cada uno de estos, lo que facilita su diseño y posterior implementación.

### **Diseñar sistema de control que incluye circuitos electrónicos y actuadores**

Dentro de los diseños mediante computadora y las pruebas fotográficas que se encuentran en el desarrollo de este documento se puede comprobar el correcto cumplimiento de este objetivo necesario para la implementación y funcionamiento del proyecto, ya con las investigaciones necesarias y los circuitos y componentes listos dentro de la fase de diseño, se procedió a la implementación del sistema.

### **Implementar el sistema de control de humedad, fertilización y fumigación.**

En las evidencias fotográficas que se encuentran en la etapa de implementación se puede observar todos los componentes necesarios implementados directamente en el invernadero,

cada uno de los cuales cumple su función basados en el diseño que se hizo anteriormente y concatenándolo con las investigaciones realizadas, el sistema quedo completo y se procedió a la ejecución del sistema.

### **Verificar el correcto funcionamiento del Sistema**

Posterior a la implementación de los circuitos de control, sensores, actuadores y demás, se procedió a “encender” o ejecutar el sistema, realizando los cambios y adecuaciones necesarias, las pruebas y ensayos necesarios para que el sistema funcionara correctamente y lo cual también se encuentra documentado y evidenciado en las diferentes fotografías y videos que se encuentran en ítems anteriores y que comprueba el cumplimiento del objetivo planteado, con lo cual completamos, abarcamos y cumplimos también el objetivo general.

Posterior al cumplimiento de los objetivos se implementa en el invernadero el proceso de siembra, monitoreo y cosecha del producto final, lo cual cumple con lo esperado en el planteamiento del proyecto.

### **Aportes realizados en el proceso de producción**

El sistema automatizado de riego, fertilización y fumigado para cultivo de habichuela bajo invernadero, monitoreado mediante aplicación móvil, permite a los agricultores controlar el proceso de producción del cultivo en tiempo real y en cualquier momento que deseen, le permite visualizar valores enteros digitales ya sea en la visualización remota (aplicación móvil) o en la visualización no remota, lo que le permite saber de forma fácil y fiable los niveles de humedad, temperatura, etc. Al igual que le brinda la tranquilidad de saber que su cultivo se encuentra en óptimas condiciones.

En base a esto, el sistema ayuda al agricultor a disminuir sus tareas diarias en el invernadero, eliminando la tarea de riego y disminuyendo el trabajo de los procesos de fertilización y fumigado del cultivo, estas tres tareas son realizadas por el mismo sistema en el momento necesario, esto reduce el esfuerzo y tiempo necesario que un agricultor debe permanecer en su cultivo, por lo que va a tener un poco más de tiempo para dedicar a sus demás labores.

Permite al usuario o agricultor estar constantemente informado del estado de su cultivo, mientras el sistema se encarga de activar y ejecutar todas las tareas necesarias para que el cultivo crezca sano y de calidad. La disminución de las tareas dentro del invernadero automatizado en comparación con un cultivo tradicional es muy grande, ya que en el invernadero automatizado solo quedan tres tareas que el sistema no puede ejecutar automáticamente (limpieza del cultivo, cambio del producto de fertilización, cambio del producto de fumigación.), las demás tareas están siendo ejecutadas por el sistema automáticamente.

### **Conclusiones**

El proyecto del sistema automatizado de riego, fertilización y fumigado para cultivo de habichuela bajo invernadero, monitoreado mediante aplicación móvil, contribuye a disminuir las tareas presentes dentro de un cultivo, es decir que ayuda a disminuir las tareas del agricultor, al mismo tiempo que controla las necesidades del cultivo, al implementarlo se concluye que es un sistema muy fiable y con grandes capacidades si se es llevado a una mayor escala, la implementación demuestra que es posible cultivar bajo un invernadero automatizado y obtener productos de calidad y apto para el consumo.

## Recomendaciones

Las mejoras que se pueden establecer para el proyecto “sistema automatizado de riego fertilización para un cultivo de habichuela” pueden desencadenarse desde la comunicación y el control de forma remota, y la posibilidad del control de la humedad relativa, ya que en la mayoría de los casos es la variable que más influye en la producción de enfermedades en las plantas, esta es la variable más difícil de controlar dentro de invernaderos y fuera de ellos ya que esta depende del nivel de evaporación del líquido que se encuentra en el suelo y esta depende de la temperatura ambiente.

Para este proyecto se posibilita el monitoreo desde una aplicación móvil, del estado de la humedad relativa, humedad del suelo, temperatura ambiente, estado de la fertilización, sistema de riego y el estado de la fumigación. Más sin embargo para que el sistema sea completo se posibilita la comunicación de forma bidireccional, para lograr el control desde la aplicación móvil para así facilitar la tarea del agricultor.

Otro medio es la posibilidad de controlar la humedad relativa por medio de extractores de calor y del aire y de sensores de humedad relativa, para así establecer condiciones óptimas para el proceso cultivación de las habichuelas.

Y otra posible mejora es la implementación de sensores para la medición de los nutrientes en el suelo que son más importantes para al crecimiento y desarrollo de las habichuelas, permitiendo desarrollar un fertirriego a partir de estas variables.

## Bibliografía

- acondicionado, A. (2017). *Tipos de Compresores de aire acondicionado*. Obtenido de [http://www.aire-acondicionado.com.es/tipos-de-compresores\\_de\\_aire\\_acondicionado/](http://www.aire-acondicionado.com.es/tipos-de-compresores_de_aire_acondicionado/)
- Altec. (2018). *¿Que son las electrovalvulas?* Obtenido de <https://www.altecdust.com/soporte-tecnico/que-son-las-electrovalvulas>
- Appinventor, M. (2018). *Appinventor*. Obtenido de MIT App inventor.
- Arduino. (2018). *Arduino*. Obtenido de ARDUINO MEGA 2560 R3: <http://arduino.cl/arduino-mega-2560/>
- Arduino. (2018). *Qué es arduino*. Obtenido de Arduino: <http://arduino.cl/que-es-arduino/>
- Bastidas, L. M. (2011). *Planteamiento de un sistema de tiego para el caso específico de la hacienda asturias enfocado al desarrollo y tecnificación del campo en Colombia*. Obtenido de repository.eafit.edu.co: [https://repository.eafit.edu.co/bitstream/handle/10784/4399/LucasMateo\\_TamayoBastidas\\_2011.pdf?seque](https://repository.eafit.edu.co/bitstream/handle/10784/4399/LucasMateo_TamayoBastidas_2011.pdf?seque)
- Castelblanco, N. G. (2015). *http://repository.udistrital.edu.co*. Obtenido de SISTEMA AUTOMÁTICO REMOTO PARA EL CUIDADO DE CULTIVOS: <http://repository.udistrital.edu.co/bitstream/11349/2743/1/GambaCastelblancoNilsonEduardo2016.pdf>

- Castro, N. &. (25 de Junio de 2015). *Una red de sensores inalámbricos para la automatización y control del riego localizado*. Obtenido de scielo.org.:  
<http://www.scielo.org.co/pdf/rcia/v33n2/v33n2a10.pdf>
- Chwalisz, M. (17 de Mayo de 2018). *thingspeak Documentation*. Obtenido de readthedocs.:  
<https://media.readthedocs.org/pdf/thingspeak/latest/thingspeak.pdf>
- comercio, c. F. (2018). *Consumidor FTC*. Obtenido de  
<https://www.consumidor.ftc.gov/articulos/s0018-aplicaciones-moviles-que-son-y-como-funcionan>
- DANE. (Enero de 2016). *Boletín mensual Insumos y factores asociados a la producción agropecuaria*. Obtenido de Cultivo de la habichuela (*Phaseolus vulgaris* L.):  
[https://www.dane.gov.co/files/investigaciones/agropecuario/sipsa/Bol\\_Insumos\\_ene\\_2016.pdf](https://www.dane.gov.co/files/investigaciones/agropecuario/sipsa/Bol_Insumos_ene_2016.pdf)
- DEAQUIZ, Y. (2014). *Revista Colombiana de Ciencias Hortícolas. Efecto de diferentes láminas de riego sobre la producción y calidad de fresa*, 192-205.
- dfrobot. (14 de septiembre de 2017). *DHT11 Temperature and Humidity Sensor* . Obtenido de  
[https://www.dfrobot.com/wiki/index.php/DHT11\\_Temperature\\_and\\_Humidity\\_Sensor\\_\(SKU:\\_DFR0067\)](https://www.dfrobot.com/wiki/index.php/DHT11_Temperature_and_Humidity_Sensor_(SKU:_DFR0067))
- Groups, Q. R. (23 de 11 de 2018). *Openpr*. Obtenido de Energy & Environment:  
<https://www.openpr.com/news/1395303/Connected-Agriculture-Market-2018-Productivity-Data-Analysis-by-Key-Vendors-Trimble-Navigation-Link-Labs-LLC-Vodafone-PLC-Accenture-PLC-IBM-Corporation-PTC-SAP-SE.html>

- Hanke, F. (2008). La nutrición de la planta y su problemática en la agricultura. En F. Hanke, *Talleres Gráficos-Fundación Universitaria Juan De Castellanos*. .
- Hernández, L., Hernández, N., & Soto, F. &. (Mar de 2010). *Estudio fenológico preliminar de seis cultivares de habichuela de la especie Phaseolus vulgaris L. Cultivos tropicales*. Obtenido de <http://scielo.sld.cu/pdf/ctr/v31n1/ctr08110.pdf>
- Instruments, H. (2018). *Humedad-relativa*. Obtenido de Hanna Instruments: <http://www.hannainst.es/blog/humedad-relativa/>
- itlalaguna.edu. (2018). Octoacoplador. *OCTOACOPLADORES*, 6.
- Jasso, J. (1999). *Los sistemas de innovación como mecanismos de innovación y de transferencia tecnológica*. CIDE. México.
- Lapuente, G. (2015). *Automatización 8 beneficios que obtendrás de automatizar un proceso*. Obtenido de <http://axontec.es/automatizacion-8-beneficios-que-obtendras-de-automatizar-un-proceso/>
- Llamas, L. (2018). *NodeMcu, la popular placa de desarrollo con ESP8266*. Obtenido de <https://www.luisllamas.es/esp8266-nodemcu/>
- Lobato, M. B. (2018). Relé. *Implementación práctica del protocolo IEC 61850 en subestaciones eléctricas. Problemas y soluciones*, 2.
- logicbus. (2018). *logicbus*. Obtenido de <http://www.logicbus.com.mx/Modbus.php>
- LÓPEZ, G. R. (2012). Base de datos sobre necesidades hídricas. *Revista Ciencias Técnicas Agropecuarias*, 42-47.

- Mattarollo, Y. ( 09 de Junio de 2014). *altec dust.com*. Obtenido de ¿Cómo funcionan las electroválvulas o válvulas solenoides de uso general?:  
<https://www.altecdust.com/blog/item/32-como-funcionan-las-electrovalvulas-o-valvulas-solenoides-de-uso-general>
- Medina, P., de Obschatko, E. S., & Preciado Patiño, J. (11 de 09 de 2013). *repositorio.iica.in*. Obtenido de Agricultura inteligente:  
[https://cgspace.cgiar.org/bitstream/handle/10568/51367/CSA-en-Colombia\\_Spanish.pdf](https://cgspace.cgiar.org/bitstream/handle/10568/51367/CSA-en-Colombia_Spanish.pdf)
- Miserendino, I. E. (Febrero de 2014). *Invernaderos: aspectos básicos sobre estructura, construcción y condiciones ambientales*. Obtenido de  
[https://inta.gob.ar/sites/default/files/script-tmp-inta\\_agricultura23\\_invernadero.pdf](https://inta.gob.ar/sites/default/files/script-tmp-inta_agricultura23_invernadero.pdf)
- Miyara, F. (2004). Señales analógicas y digitales. *Convertidores D/A Y A/D*, 3.
- Mouser.com. (2018). *DHT11 humidity & temperature sensor*. Obtenido de  
<https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- Multiplicalia. (2018). *Aplicacion Movil para tienda online*.
- Ning, D. (2017). *Developing and Deploying Analytics for IoT Systems*. Obtenido de mathworks.com: [https://it.mathworks.com/content/dam/mathworks/mathworks-dot-com/company/events/conferences/matlab-conference-australia/2017/proceedings/Developing\\_Deploying\\_Analytics\\_for\\_IoT.pdf](https://it.mathworks.com/content/dam/mathworks/mathworks-dot-com/company/events/conferences/matlab-conference-australia/2017/proceedings/Developing_Deploying_Analytics_for_IoT.pdf)
- Norton, R. (1 de Febrero de 2013). *Focus on calcium - Its role in crop production*. Obtenido de <http://anz.ipni.net>: <http://anz.ipni.net/article/ANZ-3134>

- Peery, J. (12 de Septiembre de 2018). *¿Cómo influye la humedad en la calidad de los cultivos?* Obtenido de PROMIX: <https://www.pthorticulture.com/es/centro-de-formacion/como-influye-la-humedad-en-la-calidad-de-los-cultivos/>
- Peña Peralta, M. Á. (2010). *Revista Ciencias Técnicas Agropecuarias*. Obtenido de Inyección de fertilizante con bombas de paletas y control por conductividad eléctrica en soluciones madre para fertirriego:  
[http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S2071-00542010000300007&lng=es&tlng=es](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2071-00542010000300007&lng=es&tlng=es)
- Radionics, R. (s.f.). *mantech*. Obtenido de Arduino Mega 2560:  
<http://www.mantech.co.za/datasheets/products/A000047.pdf>
- Ruiz, c., & Molina., m. (2010). *automatizacion y telecontrol de sistemas de Riego. 1st ed. Barcelona: Marcombo.*
- SAC, S. d. (2010). *Una locomotora que impulsa el desarrollo del país.* 6 p.
- Sierra, J. B. (08 de 11 de 2016). *Seminis*. Obtenido de Que es el suelo:  
<http://www.seminis.mx/blog-que-es-el-suelo/>
- Soler-Tovar, & D. (2017). *Revista de Medicina Veterinaria*,. Obtenido de Conceptualización y necesidades de una agricultura climáticamente inteligente.:  
[http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0122-93542017000100007](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0122-93542017000100007)
- Technology, A. G. (2018). *Modelos de invernadero.*
- tecnol2-18. (2013). El relé. *Electrónica analoga versión 1.0*, 9.

teslabem. (1 de Junio de 2018). *Módulo de 8 Relevadores (8 Canales) 5VDC*. Obtenido de

<http://teslabem.com/modulo-de-8-relevadores-8-canales-5vdc.html>

Tienda8. (2018). *Módulo Relé 8 Canales Para Arduino, Raspberry, Pic*. Obtenido de

[http://www.tienda8.cl/503-large\\_default/modulo-rele-8-canales-para-arduino-raspberry-pic.jpg](http://www.tienda8.cl/503-large_default/modulo-rele-8-canales-para-arduino-raspberry-pic.jpg)

Vascones Cuzco, J. C., & Chamba Tenemaza, F. D. (Mayo de 2013).

<https://dspace.ups.edu.ec>. Obtenido de Diseño e implementación de un sistema de riego automatizado y controlado de forma inalámbrica para una finca ubicada en el sector popular de Balerio Estacio:

<https://dspace.ups.edu.ec/bitstream/123456789/5304/1/UPS-GT000434.pdf>

VEGA, J. A. (2013). *Agricultura de precisión hacia la integración de datos espaciales en la producción agraria*. Obtenido de

[http://www.mapama.gob.es/ministerio/pags/Biblioteca/Revistas/pdf\\_A/Ambienta\\_2013\\_105\\_16\\_27.pdf](http://www.mapama.gob.es/ministerio/pags/Biblioteca/Revistas/pdf_A/Ambienta_2013_105_16_27.pdf); consulta: junio, 2015.

Vélez, J. P. (2014). *Uso de los fertilizantes y su impacto en la producción agrícola*.

Obtenido de Universidad Nacional de Colombia:

<http://www.bdigital.unal.edu.co/39459/1/71782231.2014.pdf>

Wild, A. (1992). *Condiciones del suelo y desarrollo de las plantas según Russell*. Ediciones Mundi-Prensa; Edición: 1 (1 de enero de 1992).

Zeballos, I. J. (2005). *PreDES*. Obtenido de Manual de operación y mantenimiento de un sistema por goteo : [http://www.preDES.org.pe/preDES/cartilla\\_riegoteo.pdf](http://www.preDES.org.pe/preDES/cartilla_riegoteo.pdf)

Zito, M. (2018). Internet de las cosas. *La sustentabilidad de internet de las cosas*, 8.

## Anexos

### Códigos de programación.

#### Anexo.1 Código principal Arduino Mega

```

// codigo proyecto
---Sistema Automatizado de
Riego, Fertilización y
Fumigado Para Cultivo de
Habichuela Bajo Invernadero,
Monitoreado Mediante
Aplicación Móvil.
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#include
<TFT_HX8357.h> //
Hardware-specific library
TFT_HX8357 tft =
TFT_HX8357(); // Invoke
custom library

// SD Card library, comment
out next line if the SdFat
library is used
#include <SD.h>
#include <DS1302.h>

// We can use the SdFat
library instead and it will
be a bit faster, uncomment
next 2 lines
//#include
<SdFat.h> // Use
the SdFat library for the
Due
//SdFat
SD; //
Permit SD function call
compatibility with no sketch
changes :-)
```

```

// SD chip select
#define SDC_CS 53

// Estos se utilizan al
llamar a la función drawBMP
()

#define BU_BMP 1 // Invierta
temporalmente las colas TFT
para los mapas de bits de
abajo hacia arriba estándar

#define TD_BMP 0 // Draw
inverted Top-Down bitmaps in
standard coord frame

/*****
*****
*****
****
** Function
name:          setup
**
Descriptions:          To
set things up
*****
*****
*****
****/
const int Riego = 5;
int H = 0;
int x= 0;
int pare = 6;
int rie = 0;
int smDirectionPin = 2;
//Dirección pin
int smStepPin = 3; //Stepper
pin
#include <Wire.h>
```

```

#include "RTCLib.h"
#include <TFT_HX8357.h> //
Biblioteca específica de
hardware para lcd
#define CENTRE 240
const int outputPin =
LED_BUILTIN;
bool state = false;
#define SDC_CS 53
//Estos se utilizan al
llamar a la función drawBMP
()
void setup()
{
  Serial.begin (115200);

  delay(4000);

  tft.init(); // Inicializar
la pantalla (varios
parámetros configurados)

  Serial.print(F("Initialisi
ng SD card..."));
  if (!SD.begin(SDC_CS)) {
    Serial.println(F("failed
!"));
    //return;
  }

  Serial.println(F("Here we
go..."));

  pinMode(pare, INPUT);

  pinMode(smDirectionPin,
OUTPUT);
  pinMode(smStepPin,
OUTPUT);

  tft.init();
  tft.setRotation(2);

  // Si se ha perdido la
corriente, fijar fecha y
hora

  tft.setRotation(2);
}

/*****
*****
*****
****
** Function
name:          loop
**
Descriptions:          Inf
inite loop
*****
*****
*****
****/
void printDate(DateTime
date)
{
  tft.setTextColor(TFT_BLACK
,TFT_YELLOW);  tft.setTextSi
ze(1);
  tft.setCursor(250, 0, 2);
  tft.println( date.hour(),
DEC);
  tft.setCursor(285, 0, 2);
  tft.println( date.minute(),
DEC);
  tft.setCursor(275, 0, 2);
  tft.println(": " );
  delay(500);
  tft.fillRoundRect(250, 0,
97-15, 20-1, 3,TFT_YELLOW);
}

void loop()
{
  // Los archivos BMP
estándar generalmente se
almacenan con un escaneo

```

```

ráster de valores de píxeles
desde abajo hacia arriba
  // Los nombres de los
archivos DEBEN utilizar el
formato de nombre de archivo
8.3 (8 caracteres +. +
Cualquiera de las 3 letras
de extensión)
  // Por convención, los
archivos de mapa de bits
terminan en .bmp
  // Los mapas de bits DEBEN
estar en formato de 24 bits
(no en escala de grises de
16 u 8 bits)
  // Landscape mode
  // descripción de código
para lcd
  tft.fillRect((TFT_BLACK);
  tft.fillRect(0, 0, 480,
320, TFT_BLACK);

  tft.fillRoundRect(30, 40,
280-17, 300-40,
3, TFT_BLACK);
  tft.fillRoundRect(28, 40,
282-15, 30-1, 3, TFT_BLACK);
  tft.fillRoundRect(28,
271, 282-15, 30-1,
3, TFT_BLACK);
  tft.setCursor(32, 100, 2);

  tft.setTextFont(4);
  tft.setTextSize(2);

tft.setTextColor(TFT_WHITE, T
FT_BLACK);
  tft.println("Bienvenido");

  tft.setTextSize(1);
  tft.setTextFont(4);
  tft.setCursor(90, 420,
2);
  tft.setTextSize(1);
  tft.setTextFont(4);

  tft.setTextColor(TFT_WH
ITE, TFT_BLACK);
  tft.println("UNAD 2018");

  delay(300);

  delay(2000);
  tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setTextFont(4);
  tft.setTextColor(TFT_WHIT
E, TFT_BLACK);
  tft.println("  Iniciando.
..");
  tft.fillRoundRect(28,
230, 51-15, 30-1,
3, TFT_YELLOW);
  delay(300);

  tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setTextFont(4);
  tft.setTextColor(TFT_WHITE
, TFT_BLACK);
  tft.println("
");
  delay(300);

  tft.setCursor(90, 200,
2);
  tft.setTextSize(1);
  tft.setTextFont(4);
  tft.println("  Iniciando.
..");
  tft.fillRoundRect(28,
230, 74-15, 30-1,
3, TFT_YELLOW);
  delay(300);
  tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setTextFont(4);
  tft.setTextColor(TFT_WHITE, T
FT_BLACK);
  tft.println("
");
  delay(300);

```

```

tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setFont(4);
  tft.println("  Iniciando.
..");
  tft.fillRect(28,
230, 97-15, 30-1,
3,TFT_YELLOW);
  delay(300);
  tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setFont(4);
tft.setTextColor(TFT_WHITE,T
FT_BLACK);
tft.println("
");
  delay(300);
tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setFont(4);
  tft.println("  Iniciando.
..");
  tft.fillRect(28,
230, 150-15, 30-1,
3,TFT_YELLOW);
  delay(300);
  tft.setCursor(90, 200,
2);
tft.setTextSize(1);
tft.setTextColor(TFT_WHITE,T
FT_BLACK);
  tft.setFont(4);
tft.println("
");
  delay(300);
  delay(300);
  tft.setCursor(90, 200,
2);
  tft.setTextSize(1);
  tft.setFont(4);
  tft.println("  Iniciando.
..");
  tft.fillRect(28,
230, 143-15, 30-1,
3,TFT_YELLOW);
  delay(300);
tft.setCursor(90, 200, 2);

tft.setTextSize(1);
  tft.setFont(4);
tft.println("
");
tft.setTextColor(TFT_WHITE,T
FT_BLACK);
  delay(300);
  tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setFont(4);
  tft.println("  Iniciando.
..");
  tft.fillRect(28,
230, 166-15, 30-1,
3,TFT_YELLOW);
  delay(300);
tft.setCursor(90, 200, 2);
tft.setTextSize(1);
  tft.setFont(4);
tft.setTextColor(TFT_WHITE,T
FT_BLACK);
tft.println("
");
  delay(300);
  tft.setCursor(90, 200, 2);
tft.setTextSize(1);
  tft.setFont(4);
  tft.println("  Iniciando.
..");
  tft.fillRect(28,
230, 189-15, 30-1,
3,TFT_YELLOW);
  delay(300);

  tft.setCursor(90, 200, 2);
tft.setTextSize(1);
  tft.setFont(4);
tft.setTextColor(TFT_WHITE,T
FT_BLACK);
tft.println("
");
  delay(300);
  tft.setCursor(90, 200, 2);
  tft.setTextSize(1);
  tft.setFont(4);
  tft.println("  Iniciando.
..");

```

```

    tft.fillRect(28,
230, 212-15, 30-1,
3,TFT_YELLOW);
    delay(300);
    tft.setCursor(90, 200, 2);
tft.setTextSize(1);
    tft.setFont(4);
    tft.setTextColor(TFT_WHITE
,TFT_BLACK);
tft.println("
");
    delay(300);
    tft.setCursor(90, 200, 2);
    tft.setTextSize(1);
    tft.setFont(4);

    tft.println(" Iniciando.
..");
    tft.fillRect(28,
230, 235-15, 30-1,
3,TFT_YELLOW);

    delay(300);
    tft.setCursor(90, 200, 2);
    tft.setFont(4);
tft.setTextSize(1);
tft.setCursor(90, 200, 2);
tft.setTextSize(1);

tft.setTextColor(TFT_WHITE,T
FT_BLACK);
    tft.setFont(4);
tft.println("
");
    delay(300);
    tft.setCursor(90, 200,
2);
    tft.setTextSize(1);
    tft.setFont(4);
    tft.println(" Iniciando.
..");
    tft.fillRect(28,
230, 258-15, 30-1,
3,TFT_YELLOW);
    delay(300);
    tft.setCursor(90, 200, 2);
tft.setTextSize(1);

    tft.setFont(4);
tft.println("
");
    delay(300);
    tft.setCursor(90, 200, 2);
    tft.setFont(4);
    tft.println(" Iniciando.
..");
    tft.fillRect(28,
230, 282-15, 30-1,
3,TFT_RED);
    delay(300);
    tft.setCursor(90, 200, 2);
tft.setTextSize(1);
    tft.setFont(4);

tft.setTextColor(TFT_WHITE,T
FT_BLACK);
tft.println("
");
    delay(300);
    tft.fillRect((TFT_BLAC
K));

drawBMP("desing.bmp", 0, 0,
BU_BMP); // se agrega la
imagen de fondo
    inicio:

    delay(500);

//tft.fillRect((TFT_BLACK)
);

    tft.setTextColor
(TFT_WHITE,TFT_BLACK); tft.
setTextSize(1);
    tft.setCursor(224, 215,
2);
    tft.println(H);
    if (H >= 60)
    {

```

```

        digitalWrite(Riego,
HIGH);

        Serial.println("ON ");

tft.setTextColor(TFT_WHITE,T
FT_BLACK); tft.setTextSize(
1);
        tft.fillRoundRect(225,
365, 40-15, 20-1,
3,TFT_BLACK);
        tft.setCursor(225, 365,
2);
        tft.println("ON");
        rie=1;
    }
    if (H <= 60){
        digitalWrite(Riego,
LOW);
        tft.setCursor(0, 55, 2);

        // Establezca el color de
fuente en blanco con un
fondo negro, configure el
multiplicador de tamaño de
texto en 1
        tft.setTextColor(TFT_WHITE
,TFT_BLACK); tft.setSiz
e(1);
        // We can now plot text on
screen using the "print"
class
        tft.setCursor(225, 365,
2);
        tft.println("OFF");
        rie=1;
    }

envio();
        goto inicio;
}
void rotate(int steps, float
speed){

```

```

        // en esta sección se
describe
        /*
Esta sección analiza el
argumento 'pasos' y almacena
'ALTO' en la variable
'dirección' si */
        /*'steps' contiene un
número positivo y 'LOW' si
contiene un negativo.*/
        int direction;

        if (steps > 0){
            direction = HIGH;
        }else{
            direction = LOW;
        }

        speed = 1/speed * 70; //se
calcula la velocidad
        steps = abs(steps);
//Almacena el valor absoluto
del contenido en 'pasos' de
nuevo en la variable 'pasos'

        digitalWrite(smDirectionPi
n, direction); //Escribe la
dirección (de nuestra
declaración if anterior), al
pin DIR de EasyDriver

        /*Steppin'*/
        for (int i = 0; i < steps;
i++){
            digitalWrite(smStepPin,
HIGH);
            delayMicroseconds(speed)
;
            digitalWrite(smStepPin,
LOW);
            delayMicroseconds(speed)
;
        }
    }
}

```

## Anexo.2 Código envío de datos a NodeMcu (ESP8266)

```

SoftwareSerial Serial1(3,
2); //declaración de pines que estaran a cargo de la comunicación
String str;
String str1;
String str2;
String str3;
String str4;
String str5;

int val;

void lo() {

    // aqui se deben poner las variables teniendo en cuenta los valores que les di a los String
    //
    // tienen que ir en este orden por que asi esta en ThingSpeak y en la aplicación

    // tambien se puede igualar por ejemplo la humedad de los sensores digamos H == humi y enviar el dato

    delay (500);

    //Humedad del suelo
    str1 =String(humi);
    Serial1.println(str);
    millis();

    // Humedad Relativa
    str1 =String(humil);
    Serial1.println(str1);
    millis();

    //Temperatura
    str =String(grados);
    Serial1.println(str2);
    millis();

    // Estado del Riego
    str1 =String(estado);
    Serial1.println(str3);
    millis();

    // Estado de la fertilizacion
    str1 =String(estado1);
    Serial1.println(str4);
    millis();

    // Estado de la fumigación
    str1 =String(estado2);
    Serial1.println(str5);
    millis();

    delay (15000);
}

```

## Anexo.3 Código de programación para sistema de fumigación

```

int hours;
int days;
int buttonState =
0; // variable for

reading the pushbutton
status
int years;
int months;
int mins;

```

```

int hora=10;
int daya=2;
int yeara=2018;
int montha=8;
int mina=50;
int houraa = 0;
int dayaa = 0;
int yearaa =
0;
int monthaa =
0;
int minaa =
0;
int value1;
int val2;
int
alue3;
int
vaue4;
int
vale5;
const
int kCePin = 2; // Chip
Enable
const int kIoPin = 3; //
Input/Output
const int kSclkPin = 4; //
Serial Clock
// Create a DS1302 object.
DS1302 rtc(kCePin, kIoPin,
kSclkPin);
String dayAsString(const
Time::Day day) {
switch (day) {
case Time::kSunday:
return "Sabado";
case Time::kMonday:
return "Lunes";
case Time::kTuesday:
return "Martes";
case Time::kWednesday:
return "Miercoles";
case Time::kThursday:
return "Jueves";
case Time::kFriday:
return "Viernes";
case Time::kSaturday:
return "Domingo";
}
return "(unknown day)";
}
void printTime() {
// Get the current time
and date from the chip.
Time t = rtc.time();
// Name the day of the
week.
const String day =
dayAsString(t.day);
// Format the time and date
and insert into the
temporary buffer.
char buf[50];
sprintf(buf, sizeof(buf),
"%s %04d-%02d-%02d
%02d:%02d:%02d",
day.c_str(),
t.yr, t.mon,
t.date,
t.hr, t.min,
t.sec);
hours = t.hr;
days = t.date;
months = t.mon;
years = t.yr;
mins = t.min;
// Print the formatted
string to serial so we can
see the time.
Serial.println(buf);
int hours;
}
void set() {
// put your setup code
here, to run once:
Time t(2018, 8, 02, 21, 11,
45, Time::kThursday);
}
void loo() {
//
if ((mina==
mins) && (daya==days) && (hora==

```

```

hours) && (montha==months) && (y
eara== years)){
    // turn LED on:

```

#### Anexo.4 Código de programación proceso de fertilización

```

const int abono = 43;
int estado1 = 0;
int retardo = 60*5;
void set() {
    // put your setup code
    here, to run once:
    pinMode(abono, OUTPUT);
}

void repetir() {
    //

    // llamar la funcion de los
    dias para saber si esta el
    dia indicado
    // si es el dia entonces
    ponemos el puerto del abono
    para que abone... y enviamos
    un 1 para thingspeak
    // en la pestaña de "Reloj"
    podemos poner la funcion
    "Dia" y "Hora" para que
    abone en la tarde y solo una
    vez ese dia.

    // if (dia == dia1 &&
    hora== h1)// Ejemplo:
    pongamos dia 19/11/ 2018 y
    hora 03:00 pm{
        // d = 1;
        //}
        // entonces llamamos la
        funcion "d" aqui para hacer
        el if.
        if (d == 1){

            digitalWrite (abono, LOW);
            // Se enciende pin digital
            encargado del control del
            abonado por medio de moulo
            relé.
            estado1 = 1;
            // este estado se envia a
            thignspeak
            delay (1000*retardo);
        }
    }

```

#### Anexo.5 Código funciones para bitmap

```

/*****
*****
*****
****
** nombre de
función:          drawRAW
**
Descripciones:   RA
W en formato de 565 archivo

de imagen en bruto de 16
bits.
*****
*****
*****
****/

// Esta función abre un
archivo de imagen ".raw" y

```

```

lo muestra en las
coordenadas dadas.
// Es más rápido que trazar
imágenes BMP ya que el
archivo ya está en el lugar
correcto
// Formato para canalizar
directamente a la pantalla.
// El ancho y el alto de la
imagen en píxeles se deben
pasar a la función
// ya que estos parámetros
no están en el archivo

// Establecer el tamaño del
búfer, 2 veces este tamaño
de bytes RAM se usará en la
matriz.
// Reducir si la memoria RAM
es corta con alguna
penalización de rendimiento
// Poderes de 2 funcionan
bien

#define RBUFF_SIZE 256

void drawRAW(char *filename,
int16_t x, int16_t y,
int16_t rawWidth, int16_t
rawHeight) {
    File    rawFile;
    uint8_t sdbuffer[2 *
RBUFF_SIZE]; // SD read
pixel buffer (16 bits per
pixel)
    Serial.println(filename);

    // Check file exists and
open it
    if ((rawFile =
SD.open(filename)) == NULL)
    {
        Serial.println(F(" File
not found"));
        return;
    }

```

```

// Prepare the TFT screen
area to receive the data
    tft.setWindow(x, y, x +
rawWidth - 1, y + rawHeight
- 1);

// Work out how many whole
buffers to send
    uint16_t nr =
((long)rawHeight *
rawWidth)/RBUFF_SIZE;
    while(nr--) {
        rawFile.read(sdbuffer,
sizeof(sdbuffer));
        tft.pushColors(sdbuffer,
RBUFF_SIZE);
    }

// Send any partial buffer
nr = ((long)rawHeight *
rawWidth)%RBUFF_SIZE;
    if (nr) {
        rawFile.read(sdbuffer,
nr<<1); // We load 2 x nr
bytes
        tft.pushColors(sdbuffer,
nr); // We send nr 16 bit
pixels
    }

// Close the file
rawFile.close();
}
/*****
*****
*****
****
** Function
name:          drawBMP
**
Descriptions:          dra
w a BMP format bitmap to the
screen
*****
*****
*****
****/

```

```

// Esta función abre un
archivo de mapa de bits de
Windows (BMP) y
// lo muestra en las
coordenadas dadas. Se
aceleró
// leyendo muchos píxeles
por valor de datos a la vez
// (en lugar de píxel por
píxel). Incrementando el
buffer
// el tamaño hace que cargar
sea un poco más rápido pero
la ley de
// Se aplican mejoras de
velocidad que disminuyen
rápidamente.
// Sugerir 8 mínimo y 85
máximo (3 x este valor es
// almacenado en un byte =
255/3 max!)
// ;Un valor de 8 es solo un
20% más lento que 24 o 48!
// Tenga en cuenta que 5
veces se necesitará este
valor de RAM bytes
// Incrementar más allá de
48 da poco beneficio.
// Usar división integral de
TFT (o imagen típica de uso
frecuente)
// ancho para una velocidad
ligeramente mejor para
evitar la purga corta del
tampón

// Maximum of 85
#define BUFF_SIZE 80

void drawBMP(char *filename,
int x, int y, boolean flip)
{
    if ((x >= tft.width()) ||
(y >= tft.height())) return;
    File      bmpFile;

```

```

    int16_t  bmpWidth,
bmpHeight; // Imagen W + H
en píxeles
    //uint8_t  bmpDepth;
        // Profundidad de bits
(debe ser 24) pero no usamos
esto
    uint32_t
bmpImageoffset; //
Iniciar la imagen según la
dirección de archivo
    uint32_t
rowSize; // No
siempre = bmpWidth; puede
tener relleno
    uint8_t  sdbuffer[3 *
BUFF_SIZE]; // Memoria
intermedia de píxeles de
lectura SD (8 bits cada R +
G + B por píxel)
    uint16_t
tftbuffer[BUFF_SIZE];
// TFT pixel out buffer (16
bits por píxel)
    uint8_t  sd_ptr =
sizeof(sdbuffer); // puntero
sdbuffer (por lo tanto,
BUFF_SIZE debe ser menor que
86)
    boolean  goodBmp =
false; //
Marcador establecido en
verdadero en el encabezado
válido de análisis
    int16_t  w, h, row,
col; // para
almacenar ancho, alto, fila
y columna
    //uint8_t  r, g, b; //
Línea de codificación brg
concatenada para la
velocidad por lo que no se
utiliza
    uint8_t  rotation; //
para restaurar la rotación
    uint8_t  tft_ptr = 0; //
puntero del búfer

```

```

    // Check file exists and
    open it
    Serial.println(filename);
    if ((bmpFile =
SD.open(filename)) == NULL)
{
    Serial.println(F(" File
not found")); // Can comment
out if not needed
    return;
}

    // Parse BMP header para
    obtener la información que
    necesitamos
    if (read16(bmpFile) ==
0x4D42) { // Comprobación de
    firma de inicio de archivo
    BMP
        read32(bmpFile); //
        / Dummy lee para tirar y
        seguir adelante.
        read32(bmpFile); //
        / Leer e ignorar los bytes
        del creador
        bmpImageoffset =
        read32(bmpFile); // Inicio
        de los datos de la imagen.
        read32(bmpFile); //
        / Dummy lee para tirar y
        seguir adelante.
        bmpWidth =
        read32(bmpFile); // Ancho
        de la imagen
        bmpHeight =
        read32(bmpFile); // Altura
        de imagen

        //if (read16(bmpFile) ==
        1) {
            // Solo proceder si
            pasamos un chequeo de
            archivo de bitmap.
            // Número de planos de
            imagen: debe ser '1',

```

```

    profundidad 24 y 0 (formato
    sin comprimir)
        if ((read16(bmpFile) ==
1) && (read16(bmpFile) ==
24) && (read32(bmpFile) ==
0)) {
            //goodBmp = true; //
            Formato BMP soportado -
            ¡proceda!
            // Las filas BMP se
            rellenan (si es necesario)
            al límite de 4 bytes
            rowSize = (bmpWidth *
3 + 3) & ~3;
            // Área de cultivo a
            cargar.
            w = bmpWidth;
            h = bmpHeight;

            // Podríamos necesitar
            alterar la rotación para
            evitar la tediosa
            manipulación del puntero.
            // Guarda el valor
            actual para que podamos
            restaurarlo más tarde.
            rotation =
            tft.getRotation();
            // Use la rotación de
            coordenadas TFT SGRAM si la
            inversión está configurada
            para una representación un
            25% más rápida
            if (flip)
            tft.setRotation((rotation +
            (flip<<2)) % 8); // Value 0-
            3 mapped to 4-7

            // Es posible que
            tengamos que voltear y
            calcular la nueva coordenada
            de la gráfica relativo a la
            esquina superior izquierda
            también ...

            switch (rotation) {
                case 0:

```

```

        if (flip) y =
tft.height() - y - h; break;
        case 1:
            y = tft.height() -
y - h; break;
            break;
        case 2:
            if (flip) y =
tft.height() - y - h; break;
            break;
        case 3:
            y = tft.height() -
y - h; break;
            break;
    }

    // Establecer la
ventana de dirección TFT a
los límites de la imagen
    // Actualmente, la
imagen no se dibujará o se
corromperá si no encaja
    tft.setWindow(x, y, x
+ w - 1, y + h - 1);

    // Finalmente, estamos
listos para enviar filas de
píxeles y escribir así evita
la multiplicación lenta de
32 bits
    for (uint32_t pos =
bmpImageoffset; pos <
bmpImageoffset + h * rowSize
; pos += rowSize) {
        // Busque si es
necesario en los límites y
arregle para volcar el búfer
y comience de nuevo
        if
(bmpFile.position() != pos)
{
            bmpFile.seek(pos);
            sd_ptr =
sizeof(sdbuffer);
        }
        // Rellenar el
buffer de píxeles y trazar
        for (col = 0; col <
w; col++) { // Para cada
columna
            // ¿Tiempo para
leer más datos de píxeles?
            if (sd_ptr >=
sizeof(sdbuffer)) {
                // Empuje tft
buffer a la pantalla
                if (tft_ptr) {
                    // Aquí
estamos enviando una matriz
uint16_t a la función
                    tft.pushColors
(tftbuffer, tft_ptr);
                    tft_ptr = 0;
                // tft_ptr y sd_ptr no
siempre están sincronizados
                ...
            }
            // Finally
reading bytes from SD Card
            bmpFile.read(sdb
uffer, sizeof(sdbuffer));
            sd_ptr = 0; //
Set buffer index to start
        }
        // Convert pixel
from BMP 8+8+8 format to TFT
compatible 16 bit word
        // Blue 5 bits,
green 6 bits and red 5 bits
(16 bits total)
        // Is is a long
line but it is faster than
calling a library fn for
this
            tftbuffer[tft_ptr]
= (sdbuffer[sd_ptr++] >> 3)
;
            tftbuffer[tft_ptr]
|= ((sdbuffer[sd_ptr++] &
0xFC) << 3);

```

```

        tftbuffer[tft_ptr]
|= ((sdbuffer[sd_ptr++] &
0xF8) << 8);
        tft_ptr++;
    } // Next row
} // All rows done

// Escribe cualquier
buffer parcialmente lleno a
TFT
    if (tft_ptr)
tft.pushColors(tftbuffer,
tft_ptr);

    } // Fin del acceso al
mapa de bits.
    } // Fin del control de
archivo de mapa de bits
    //} // Podemos cerrar
el archivo ahora

    bmpFile.close();

    tft.setRotation(rotation);
// Ponga de nuevo la
rotación original.
}

/*****
*****
*****
****
** Nombre de
funcion: Funciones de
soporte para drawBMP ()

**
descripciones: Le

```

```

e tipos de 16 y 32 bits del
archivo de la tarjeta SD
*****
*****
*****
****/

```

```

// Los datos BMP se
almacenan en little-endian,
Arduino también es little-
endian.
// Es posible que deba
revertir el orden de los
subíndices si se traslada a
otro lugar.

```

```

uint16_t read16(File& f) {
    uint16_t result;
    ((uint8_t *)&result)[0] =
f.read(); // LSB
    ((uint8_t *)&result)[1] =
f.read(); // MSB
    return result;
}

```

```

uint32_t read32(File& f) {
    uint32_t result;
    ((uint8_t *)&result)[0] =
f.read(); // LSB
    ((uint8_t *)&result)[1] =
f.read();
    ((uint8_t *)&result)[2] =
f.read();
    ((uint8_t *)&result)[3] =
f.read(); // MSB
    return result;
}

```

## Anexo.6 Código de programación para recolección de datos de sensores.

```

//#define DHTPIN 13 //
Indicamos el pin donde

```

```

conectaremos la patilla data
de nuestro sensor

```

```

#define HT1 40

// El sensor que vas a
// emplear. En este caso usamos
// el DHT11
#define DHTTYPE DHT11 //
DHT 11

DHT dht1 (HT1,
DHTTYPE); //Indica el pin
con el que trabajamos y el
tipo de sensor

const int Riego = 41;
int estado = 0;
int humi = 0;
int x = 0;
int pare = 42;

dht1.begin(); //Iniciamos
el sensor

pinMode(Riego, OUTPUT);
pinMode(pare, INPUT);

void repeat()
{

float humil, grados = 0;
// La lectura de la
temperatura o de la humedad
lleva sobre 250 milisegundos
// La lectura del sensor
tambien puede estar sobre
los 2 segundos (es un sensor
muy lento)
humil =
dht1.readHumidity(); //Guar
da la lectura de la humedad
en la variable float h

grados =
dht1.readTemperature(); //G
uarda la lectura de la
temperatura en la variable
float t

// read the input on analog
pin 0:
// Convert the analog
reading (which goes from 0 -
1023) to a voltage (0 - 5V):
int
voltage1=map(analogRead(A0),
0,1023,100,0);
int
voltage2=map(analogRead(A1),
0,1023,100,0);
int
voltage3=map(analogRead(A2),
0,1023,100,0);

x = (voltage1 + voltage2 +
voltage3 );
humi = (x/3);
digitalRead (pare);
// lcd.noDisplay();
// lcd.begin(16, 2);
//lcd.setCursor(0,0);
// lcd.print("Niv Humedad:
");
// lcd.print(voltage);
//lcd.print("%");

if (humi <= 60 && pare ==
LOW) {
digitalWrite(Riego,
LOW);
}

if (humi >= 60 && pare
== LOW) {
digitalWrite(Riego,
HIGH);
}

```

```

    }

    if (humi <= 60 && pare ==
HIGH )
    {
        digitalWrite(Riego,
HIGH);
    }

    if (humi >= 60 && pare ==
HIGH )
    {
        digitalWrite(Riego,
HIGH);
    }
}

if (Riego == LOW){
    estado = 1;
}

if (Riego == HIGH){
    estado = 0;
}
}
}

```

### Anexo.7 Código programación para NodeMcu (ESP8266)

```

String apiWritekey =
"974Y0S65J6KOXMV7"; // Clave
de thigspeak
const char* ssid =
"JOLiMAR.S.A.P"; // nombre
wifi SSID
const char* password = "
";// Clave de wifi

const char* server =
"api.thingspeak.com";
float resolution=3.3/1023;//
3.3 is the supply volt &
1023 is max analog read
value
WiFiClient client;

void setu () {

    WiFi.disconnect();
    delay(100);
    WiFi.begin(ssid,
password);

    Serial.println();

    Serial.println();
    Serial.print("Connecting
to ");
    Serial.println(ssid);

    WiFi.begin(ssid,
password);

    while (WiFi.status() !=
WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("NodeMcu
connected to wifi...");
    Serial.println(ssid);
    Serial.println();
}

void looper() {

    Serial.flush();
    //desde aqui empieza a
recibir el string:

```

```

String F = "str" ;
if (Serial.available() )
{ char d = Serial.read();
  while ( d != '\n' )
//Hasta que el caracter sea
intro != significa diferente
a...
  { F = F + d ;
    //delay(25) ;
    d = Serial.read();
  }
} else {
  return;
}
delay(500);
Serial.println("Humedad
del suelo: ");
Serial.println(String
(F));
Serial.println('\n');
Serial.flush();
delay(1000);

String S = "str1" ;
if (Serial.available() )
{ char c = Serial.read();
  while ( c != '\n' )
//Hasta que el caracter sea
intro != significa diferente
a...
  { S = S + c ;
    //delay(25) ;
    c = Serial.read();
  }
} else {
  return;
}
delay(500);
Serial.print("Humedad
Relativa: ");
Serial.println(String
(S));
Serial.println('\n');
Serial.flush();
delay(1000);

```

```

String D = "str2" ;
if (Serial.available() )
{ char e = Serial.read();
  while ( e != '\n' )
//Hasta que el caracter sea
intro != significa diferente
a...
  { D = D + e ;
    //delay(25) ;
    e = Serial.read();
  }
} else {
  return;
}
delay(500);
Serial.println("Temperatur
a: ");
Serial.println(String
(D));
Serial.println('\n');
Serial.flush();
delay(1000);

String R = "str3" ;
if (Serial.available() )
{ char u = Serial.read();
  while ( u != '\n' )
//Hasta que el caracter sea
intro != significa diferente
a...
  { R = R + u ;
    //delay(25) ;
    u = Serial.read();
  }
} else {
  return;
}
delay(500);
Serial.println("Estado del
Riego: ");
Serial.println(String
(R));
Serial.println('\n');
Serial.flush();
delay(1000);

```

```

    String P = "str4" ;
    if (Serial.available() )
    { char i = Serial.read();
      while ( i != '\n' )
//Hasta que el caracter sea
intro != significa diferente
a...
      { P = P + i ;
        //delay(25) ;
        i = Serial.read();
      }
    } else {
      return;
    }
    delay(500);
    Serial.println("Estado de
la fertilización: ");
    Serial.println(String
(P));
    Serial.println('\n');
    Serial.flush();
    delay(1000);

    String C = "str5" ;
    if (Serial.available() )
    { char m = Serial.read();
      while ( m != '\n' )
//Hasta que el caracter sea
intro != significa diferente
a...
      { C = C + m ;
        //delay(25) ;
        m = Serial.read();
      }
    } else {
      return;
    }
    delay(500);
    Serial.println("Estado del
fumigado: ");
    Serial.println(String
(C));
    Serial.println('\n');
    Serial.flush();
    delay(1000);

```

```

    if
(client.connect(server,80))
    {

        String postStr =
apiWritekey;
        postStr
+ "&field1=";
        postStr +=
String(F);
        postStr
+ "&field2=";
        postStr +=
String(S);
        postStr
+ "&field3=";
        postStr +=
String(D);
        postStr
+ "&field4=";
        postStr +=
String(R);
        postStr
+ "&field5=";
        postStr +=
String(P);
        postStr
+ "&field6=";
        postStr +=
String(C);
        postStr +=
"\r\n\r\n\r\n\r\n\r\n\r\n\r\n";

        client.print("POS
T /update HTTP/1.1\n");
        client.print("Hos
t: api.thingspeak.com\n");
        client.print("Con
nection: close\n");
        client.print("X-
THINGSPEAKAPIKEY:
"+apiWritekey+"\n");

```

```

        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(postStr.length());
        client.print("\n\n");
        client.print(postStr);

        Serial.print("Temperatura: ");
        Serial.print(S);

        Serial.println("Humedad del suelo: ");
        Serial.print(F);
        Serial.println("uploaded to Thingspeak server....");
    }
    client.stop();

    Serial.println("Waiting to upload next reading...");
    Serial.println();
    // thingspeak needs minimum 15 sec delay between updates
    delay(15000);
}

```