

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

FACULTAD DE INGENIERÍA Y GESTIÓN

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“DESARROLLO DE UNA APLICACIÓN MÓVIL PARA CONTROLAR
LAS VARIABLES CLIMÁTICAS EN EL INVERNADERO DE LA
UNTELS”**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de

INGENIERO DE SISTEMAS

PRESENTADO POR EL BACHILLER

SUPO SAMANEZ, MACIEL DENI

Villa El Salvador

2019

DEDICATORIA

*A DIOS, A MI ABUELITA QUE ESTÁ
EN EL CIELO, A MIS HERMANOS, A
MI FAMILIA QUE SIEMPRE ME HA
APOYADO EN TODO MOMENTO.*

AGRADECIMIENTO

A MI FAMILIA, A LOS DOCENTES QUE HAN INTERVENIDO EN MI FORMACION ACADEMICA A LO LARGO DE MI CARRERA, A MI ASESOR EL PROFESOR LEYVA Y AL PROFESOR DAVID LICCAPA POR HABERME GUIADO EN TODO ESTE PROCESO Y APOYADO.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA.....	3
1.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA	3
1.2 JUSTIFICACIÓN DEL PROBLEMA.....	3
1.3 DELIMITACIÓN DEL PROYECTO	4
1.3.1 Teórica.....	4
1.3.2 Temporal.....	4
1.3.3 Espacial	4
1.4 FORMULACIÓN DEL PROBLEMA	5
1.4.1 Problema general.....	5
1.4.2 Problemas específicos	5
1.5 FORMULACIÓN DE OBJETIVOS	5
1.5.1 Objetivo general.....	5
1.5.2 Objetivos específicos	5
CAPÍTULO II: MARCO TEÓRICO	6
2.1 ANTECEDENTES DE LA INVESTIGACIÓN	6
2.1.1 Antecedentes internacionales	6
2.1.2 Antecedentes Nacionales	8
2.2. BASES TEÓRICAS.....	9
2.2.1 INVERNADEROS	9
2.2.1.1 Definición de invernadero	9
2.2.1.2 Importancia de un invernadero.....	9
2.2.1.3 Aspectos Físicos de un invernadero	9
2.2.1.4.1 Invernadero plano o tipo parral	10
2.2.1.4.2 Invernadero en raspa y amagado.....	11
2.2.1.4.3 Invernadero asimétrico o inacral	12

2.2.1.4.4 Invernadero túnel o semicilíndrico	13
2.2.1.2 CONTROL DE LAS VARIABLES CLIMÁTICAS	14
2.2.1.2.1 Definición	14
2.2.1.2.2 PARÁMETROS A CONSIDERAR EN EL CONTROL CLIMÁTICO DEL INVERNADERO	17
A.Temperatura.....	17
1. Control por calefacción	17
2. Control por ventilación	18
B. Humedad relativa del aire.....	18
C. Iluminación.....	19
D. Humedad del suelo	20
1. Control utilizando un interruptor horario	20
2.2.2 SISTEMA ARDUINO Y SENSORES.....	21
2.2.2.1 Tipos de microcontroladores	21
2.2.2.2 Tipos de sensores para medir las variables climáticas.....	22
2.2.2.2.1 Sensores de humedad y temperatura	22
2.2.2.2.2 Sensores de humedad del suelo	23
2.2.3 APLICACIONES MÓVILES.....	24
2.2.3.1 Definición.....	24
2.2.3.2 METODOLOGÍA BASADA EN SCRUM	24
2.2.3.3 GESTOR DE BASE DE DATOS	28
2.3 DEFINICIÓN DE TERMINOS BÁSICOS.....	28
CAPÍTULO III: DESARROLLO DEL TRABAJO DE SUFICIENCIA PROFESIONAL	30
3.1 DESARROLLO DEL HADWARE CON LA TECNOLOGIA ARDUINO	31
3.1.1 REQUERIMIENTOS DE EQUIPOS PARA CAPTURAR LAS VARIABLES CLIMÁTICAS.....	31
3.1.1.1 Sistema microcontrolador.....	31

3.1.1.2 Sensores.....	31
3.1.1.1.3 Lcd(Liquid Crystal Display).....	32
3.1.1.1.3 Nodemcu	33
3.1.1.1.4 Relay y motores de riego	34
3.1.2 DESARROLLO DEL SOFTWARE EN ARDUINO	35
3.1.2.1 ARDUINO	35
3.1.2.2 NODEMCU	45
3.2 DESARROLLO SOFTWARE DE LA APLICACIÓN MÓVIL.....	54
3.2.1 ANALISIS DEL PROCESO	54
3.2.2. ENTREVISTA	54
3.3 MARCO DE DESARROLLO BASADO EN LA METODOLOGÍA SCRUM ..	54
3.3.1 Roles	54
3.3.2 Product Backlog.....	55
3.3.2.1. Historia de Usuarios.....	55
3.3.3. Sprint Planning.....	62
3.3.3.1 Prototipos.....	64
3.3.3.2 mapa de navegación de como se verá la aplicación móvil	67
3.3.4. Sprint Backlog.....	68
3.3.5 TaskBoard Inicial.	68
3.3.6. Desarrollo y Sprint Review.	69
3.3.6.1. Sprint 1.	69
3.3.6.1.1. Desarrollo de Tareas.....	69
3.3.6.1.2. Taskboard	72
3.3.6.1.3. Sprint Review.....	73
3.3.6.2. Sprint 2.	74
3.3.6.2.1. Desarrollo de Tareas.....	74
3.3.6.2.2 Taskboard.....	76

3.3.6.2.3 Sprint Review	77
3.4. RECURSOS.	78
3.4.1. Recursos Humanos.....	78
3.4.2 Recursos software	78
3.4.3. Recursos de Hardware.	79
3.5 DESARROLLO DEL SOFTWARE EN ANDROID	80
3.6 RESULTADOS	108
CONCLUSIONES	121
RECOMENDACIONES	122
BIBLIOGRAFÍA.....	123
ANEXOS.....	126

LISTADO DE TABLAS

TABLA 1: CONTROL DE LA TEMPERATURA Y HUMEDAD	15
TABLA 2: CUADRO COMPARATIVO DE CONTROLADORES	22
TABLA 3: SPRING BACKLOG	62
TABLA 4: SPRING BACKLOG DE SPRINT 1	63
TABLA 5: SPRING BACKLOG DE SPRINT 2.	63
TABLA 6: SPRINT BACKLOG	68
TABLA 7: TASKBOARD INICIAL DEL PROYECTO.....	69
TABLA 8: TASKBOARD.....	72
TABLA 9: PLANTILLA DE REVISIÓN DEL SPRINT1	73
TABLA 10 : TASKBOARD DE SPRINT2.....	76
TABLA 11: SPRINT REVIEW DEL SPRINT2.....	77
TABLA 12: RECURSOS HUMANOS	78
TABLA 13: RECURSOS SOFTWARE	78
TABLA 14: RECURSOS HADWARE	79

LISTADO DE FIGURAS

FIGURA 1 : INVERNADERO TIPO PLANO O TIPO PARRAL	11
FIGURA 2: INVERNADERO EN RASPA Y AMAGADO	12
FIGURA 3: INVERNADERO TIPO TÚNEL.....	13
FIGURA 4: INVERNADERO DE CRISTAL.....	14
FIGURA 5: DIAGRAMA DE CONTROL DE VARIABLES CLIMÁTICAS.....	16
FIGURA 6: CONTROLADOR INTERRUPTOR HORARIO	21
FIGURA 7: SENSOR DHT11	23
FIGURA 8: SENSOR HIGRÓMETRO (FC-28)	24
FIGURA 9: ARQUITECTURA DE LA APLICACIÓN MÓVIL	30
FIGURA 10: MICROCONTROLADOR ARDUINO UNO	31
FIGURA 11: SENSORES DE TEMPERATURA Y HUMEDAD	32
FIGURA 12 :LCD CONECTADO A ARDUINO	33
FIGURA 13 : NODEMCU CONECTADO A ARDUINO	34
FIGURA 14: ESQUEMA GENERA ARDUINO.....	35
FIGURA 15.IMPORTACIÓN EN ARDUINO DE LIBRERÍA FIREBASE	45
FIGURA 16: IMPORTACIÓN PARA LA LIBRERÍA FIREBASE-ARDUINO-MASTER.....	46
FIGURA 17: LIBRERÍA ARDUINO JSON.....	46
FIGURA 18: IMPORTACIÓN DE LA LIBRERÍA ESP8266	47
FIGURA 19: SECRETO DE LA BASE DE DATOS.....	48
FIGURA 20: CASOS DE USO DEL SISTEMA	56
FIGURA 21: PROTOTIPO DE LOGIN.....	64
FIGURA 22: REGISTRO USUARIO	64
FIGURA 23: INTERFAZ PLANTA	65
FIGURA 24: INTERFAZ GRÁFICA	65
FIGURA 25: INTERFAZ PLANTA2	66
FIGURA 26: MAPA DE NAVEGACIÓN.....	67
FIGURA 27: VISTA DE LOGIN	70
FIGURA 28: VISTA REGISTRO USUARIO.....	71
FIGURA 29: VISUALIZACIÓN DE LAS VARIABLES CLIMÁTICAS	74
FIGURA 30: GRÁFICA DE LAS VARIABLES CLIMÁTICAS	75

FIGURA 31: ESQUEMA DE MI CIRCUITO	108
FIGURA 32: VISUALIZACIÓN DE LAS VARIABLES CLIMÁTICAS EN LA PANTALLA LCD	109
FIGURA 33: VISUALIZACIÓN DE MODULOS RELAY	110
FIGURA 34: BOMBAS DE AGUA CONECTADAS AL CIRCUITO.....	111
FIGURA 35: NODEMCU CONECTADO AL CIRCUITO	112
FIGURA 36: PLANTA BEGONIA.....	113
FIGURA 37: SENSOR DE HUMEDAD Y TEMPERATURA CONECTADA A LA PLANTA BEGONIA.....	113
FIGURA 38: NODEMCU CONECTADA AL CIRCUITO.....	114
FIGURA 39: VISULIZACIÓN EN TIEMPO REAL DE LAS VARIABLES CLIMÁTICAS	115
FIGURA 40: ACTIVACIÓN DE RIEGO PARA LA PLANTA	115
FIGURA 41: VARIACIÓN DE LAS VARIABLES CLIMÁTICAS.....	116
FIGURA 42: CONTROL DE RIEGO	116
FIGURA 43: MEDICIÓN DE LAS VARIABLES CLIMÁTICAS	117
FIGURA 44: SENSORES DE HUMEDAD Y TEMPERATURA EN LA PLANTA GERANIO	117
FIGURA 45: VISUALIZACIÓN DE LAS VARIABLES CLIMÁTICAS EN FIREBASE	118
FIGURA 46: CONTROL DE RIEGO EN LA PLANTA GERANIO.....	119
FIGURA 47: VARIACIÓN DE LAS VARIABLES CLIMÁTICAS.....	120

INTRODUCCIÓN

Los países en América Latina hay un número mayor de porcentaje de personas que padece de hambre, por eso se ha incrementado la aparición de invernaderos mayormente para uso personal de las familias que quieren cultivar productos en una temporada del año al que no pertenece y de buena calidad.

Los invernaderos permiten tener un mayor control del entorno donde crecen las plantas, estos son utilizados para aumentar la calidad y rendimiento de las plantaciones en ciertas ubicaciones que tienen estaciones cortas de crecimiento, o bien malas condiciones de iluminación debido a las localizaciones geográficas, por lo que permiten mejorar la producción de alimentos.

Las tecnologías de control climático en el mercado tienen un costo que no se ajusta para cualquier invernadero familiar construidos con bajo presupuesto estas tecnologías comerciales están destinadas al uso en invernaderos de producción a gran escala.

El presente proyecto se realizó en el invernadero el cual está ubicado en el interior de la Universidad Nacional Tecnológica de Lima Sur en el cual se dedican al cultivo de plantas entre hortalizas y frutas entre las cuales están margarita , begonia ,geranio , anturio rojo, medusa ,helecho ,vordihuela , mucura ,boa ,cardenal grande ,calanchoy ,cedrón , ala de ángel ,oreja de elefante ,sabila enana ,cristalina, limón ,entre otros.

Para analizar la problemática es necesario mencionar sus causas una de ellas es que actualmente para verificar si una planta necesita riego se guía mediante el tacto y no hay un control preciso de las variables climáticas lo cual implicaría un problema ya que estas plantas son de cuidado para su correcto desarrollo.

El presente proyecto del desarrollo de la aplicación móvil permitirá llevar el seguimiento y monitoreo en tiempo real de las variables climáticas de temperatura humedad relativa y humedad del suelo, logrando mejorar y optimizar los tiempos que la información esté almacenada en la base de datos y esta puede ser visualizada en el aplicativo móvil.

En la metodología se dividió en dos partes una para el desarrollo en arduino y la otra para el marco de desarrollo de la aplicación basada en scrum, para ello se realizó entrevista y reuniones con el encargado del invernadero para identificar los requerimientos.

El presente proyecto se divide en 3 capítulos; Capítulo I: Planteamiento del problema, en esta parte se definieron las problemáticas y los objetivos que se necesitan cumplir; Capítulo II: Marco teórico esencial para dar a conocer en base a que se va a desarrollar el proyecto y el Capítulo III: Desarrollo del objetivo de trabajo de suficiencia, este capítulo muestra cómo se realizó la implementación del proyecto en base a la metodología y herramientas utilizadas el cual se dividió en dos partes desarrollo del sistema en arduino y desarrollo de la aplicación móvil .

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA

El invernadero que se encuentra en el interior de la Universidad Nacional Tecnológica de Lima Sur se encarga del cultivo de hortalizas y frutas.

Actualmente en este invernadero se realiza el cultivo de plantas mediante la propagación de estas, el lugar se encuentra en un ambiente cerrado y el control de riego se realiza basada en la experiencia de la persona que se encarga del control del invernadero, es decir cuando el encargado verifica que en el suelo no está húmedo sabe que la planta necesita riego.

El encargado del invernadero de la UNTELS sabe que el procedimiento que realiza no es preciso, y que esto puede causar problemas en el desarrollo adecuado de la planta ya que cada planta se desarrolla en condiciones climáticas distintas.

El control del invernadero de la UNTELS se realiza de lunes a viernes por lo tanto los fines de semana no existe un control de riego en las plantas.

La verificación de las variables climáticas que es realizada vía manual puede ocasionar errores por la cual el propósito de esta investigación es monitorear desde cualquier lugar las variables climáticas de humedad del suelo, temperatura, humedad relativa del invernadero y controlar el riego a través de una aplicación móvil.

1.2 JUSTIFICACIÓN DEL PROBLEMA

Ante el problema de llevar un monitoreo y control adecuado y preciso de las variables climáticas del invernadero de la UNTELS resulta de especial interés desarrollar una aplicación móvil la cual permitirá al personal que trabaja en el invernadero monitorear en tiempo real y de forma remota la temperatura y humedad del suelo, humedad relativa ,ya que actualmente lo realizan de manera

manual, a partir de ahí adoptar medidas que permita controlar estas variables de forma fácil mediante la aplicación móvil .

El presente proyecto surge de la necesidad de llevar un control adecuado ya que cada variedad de planta se desarrolla en condiciones distintas y necesitan desarrollarse en un entorno adecuado y en los rangos climáticos establecidos ya que si no es así la planta no cumple su ciclo de vida y muere debido a ello el propósito de este proyecto es obtener estas variables de forma precisa sin ningún margen de error y en consecuencia emitir un control adecuado a cada variedad de planta.

Además el proyecto busca proporcionar información que será útil para las carreras de ingeniería electrónica, ingeniería de sistemas y en general para toda la comunidad universitaria en la parte académica de los estudiantes en la parte de investigación.

1.3 DELIMITACIÓN DEL PROYECTO

13.1 Teórica

El presente proyecto es una solución informática, desarrollada en Android.

1.3.2 Temporal

El proyecto tuvo una duración aproximadamente de dos meses, empezó el 10 de enero del 2019 y finalizó el 30 de marzo 2019.

1.3.3 Espacial

El proyecto se realizó en el Invernadero de Untels el cual está ubicado en Sector 3 Grupo 1A 03 Cercado (cruce de la Av. Central y Av. Simón Bolívar), Villa El Salvador Lima Metropolitana, Lima)

1.4 FORMULACIÓN DEL PROBLEMA

1.4.1 Problema general

¿Cómo controlar las variables climáticas en el invernadero de la UNTELS?

1.4.2 Problemas específicos

¿Cuáles son las variables climáticas a monitorear en el invernadero de la UNTELS?

¿Cómo desarrollar el hardware con tecnología arduino para capturar las variables climáticas en el invernadero de la UNTELS?

¿Cómo desarrollar el software que permita visualizar los datos en tiempo real en una aplicación móvil?

1.5 FORMULACIÓN DE OBJETIVOS

1.5.1 Objetivo general

Controlar las variables climáticas en el invernadero de la UNTELS

1.5.2 Objetivos específicos

- Identificar las principales variables climáticas para el monitoreo en el invernadero de la UNTELS.
- Desarrollar el hardware con tecnología arduino que permita para capturar las variables climáticas en el invernadero de la UNTELS.
- Desarrollar el software que permita visualizar los datos en tiempo real en una aplicación móvil en el invernadero de la UNTELS

CAPÍTULO II: MARCO TEÓRICO

2.1 ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1 Antecedentes internacionales

(Jhonatan Baltazar Aguilar&Enciso Hernandez Daniel, 2014) .Diseño e implementación de un sistema digital para el control de temperatura en un invernadero de tomate, concluye que : En la actualidad el sector agroalimentario se ha visto afectado por los cambios del medio ambiente, dichos cambios climáticos afectan directamente a cualquier cultivo, la variable que tiene mayor importancia para el desarrollo, crecimiento y maduración es la temperatura para que la planta tenga un buen desarrollo.

(Calderon Quintana & Torres Suarez, 2018). Diseño de un sistema de control y mantenimiento de un bio-huerto vertical en edificaciones comerciales de la zona céntrica de Guayaquil concluye que: Los factores que inciden en el cuidado y mantenimiento de los huertos verticales, pueden ser controlados mediante sensores, de tal manera que optimiza el tiempo y garantiza un mejor control en las plantas.

(Acosta, 2015) .Automatización de bajo costo concluye que: El proyecto se realizó plataforma basada en la tecnología arduino yun, una aplicación web y una aplicación móvil que llego como resultado del proyecto monitorear variables ambientales en el interior de un biohuerto e invernadero, además permite controlar en tiempo real mecanismos de riego instalados, además es de muy bajo costo y fácil de adquirir en el mercado, en este proyecto se logró monitorear las variables climáticas(humedad y temperatura) mediante ethernet por lo cual se propone que más adelante sea mediante wifi, la base de datos donde se irán guardando estos valores es MySql con las cuales se obtuvo un cuadro comparativo, también se logró optimizar los recursos de agua mediante el control del riego.

(Narvaez, 2006). Proyecto de automatización de un invernadero para el cultivo de diversos productos auténticos de la zona donde se encuentra ubicada la hacienda "AYALA" concluye que el control automático disminuye los costos de recursos humanos y permite optimizar el consumo de agua, para un correcto funcionamiento del sistema de control se debe dar un mantenimiento periódico dirigida principalmente a los sensores y a los cable que conducen las señales.

(Honores & Sanchez, 2016). Implementación de un módulo de agricultura de precisión para monitoreo y control de riego, integrado al sistema IOTMACH. La agricultura de precisión, hoy en día es una de las soluciones más avanzadas para incrementar eficiencia y productividad de los cultivos mediante el uso de las TIC, basado en la información y la tecnología impulsada, diseñado en mejorar los procesos agrícolas mediante el control de precisión asegurando la máxima producción agrícola con un mínimo impacto ambiental.

(Perea Palacios, 2016). Diseño de un sistema de monitoreo, registro, control de temperatura y humedad para un cultivo de invernadero concluye que: El desarrolló un sistema de bajo costo y fácil implementación que permite el monitoreo, registro y control de variables físicas (Temperatura, humedad del ambiente, luminosidad y humedad del suelo) de un cultivo de invernadero fue implementado ya que los consumidores demandan productos agrícolas de alta calidad y diversidad durante todo el año, esto no es posible debido a los cambios climáticos que se producen. Los invernaderos permiten obtener condiciones climáticas adecuadas para el desarrollo y crecimiento de las plantas dentro de un espacio cerrado. Por lo tanto, se crea una barrera física entre los cultivos dentro del invernadero y el clima exterior, esto ayuda a proteger a las plantas de los fenómenos atmosféricos como la caída de granizas y heladas que dañan los cultivos, así como también de los fuertes vientos, las lluvias y elevadas temperaturas. Por otro lado, facilita el control de plagas y enfermedades las cuales pueden ocasionar grandes pérdidas en la producción. En consecuencia, la implementación de invernaderos permite al agricultor obtener cosechas durante todo el año, incluso en épocas de invierno, ayuda a reducir el tiempo y obtener productos agrícolas de alta calidad, el registro de las variables climáticas se

realizó vía Ethernet las cuales son almacenadas en una base de datos previamente elaborada en mySql las cuales se emiten mediante un reporte.

(Perez Rojas & Cortez De Paul, 2001). Simulación y control de la temperatura dentro de un invernadero entre los resultados que se obtuvo fue imulación del modelo propuesto para la planta, este modelo se ajusta a las condiciones reales del invernadero sobre el cual se realizo el trabajo. Para esto fue necesario realizar un trabajo de investigación en cuanto a todos los valores de los parámetros que influyen en el modelo, se diseño un sistema de control del modelo propuesto en el invernadero.

2.1.2 Antecedentes Nacionales

(Aliaga Mendoza & Pedro, 2015) . Sistema de control de humedad relativa para un invernadero utilizando el controlador lógico programable concluye que: El regulador proporcional integral y derivativo implementado en el controlador lógico programable, es un algoritmo de control que regula satisfactoriamente, y se encuentran en la mayoría de controles de invernaderos con esta tecnología. El diseño del lazo de control, se basó el análisis unificado entre las necesidades de desempeño en frecuencia y en tiempo, ya que cada uno de éstos aporta diferentes criterios de estabilidad.

(Barrozo, 2015). Control y monitorización de un invernadero a través de una aplicación móvil: En este trabajo se realizado un control de un invernadero mediante la tecnología arduino y android.

(Mori, 2018). Implementación , control y monitoreo de un sistema de riego por goteo subterráneo con microcontroladores. En este proyecto se realizó el monitoreo de una sola variable climática que es la humedad del suelo y control de riego para ello se utilizo App Inventor donde se genero el código y para capturar las variables de la humedad del suelo se utilizó el microcontrolador arduino Mega 2560 , se logro eliminar el control de riego manual y ahorro de tiempo

2.2. BASES TEÓRICAS

2.2.1 INVERNADEROS

2.2.1.1 Definición de invernadero

Los primeros invernaderos fueron construidos con el fin de hacer posible la producción de cultivos en épocas del año donde normalmente la producción no es posible (Octavio, Sarmiento, Fabián, & Rojas, 2006) . En el caso de los invernaderos caseros, son instalaciones económicas construidas con materiales de bajo costo o materiales reciclables que brinden las condiciones climáticas adecuadas para el cultivo de variedades de plantas principalmente hortalizas (Acosta, 2015)

El aumento en la construcción de invernadero y huertos caseros se ha debido en los últimos años a las irregularidades climáticas en el cultivo en suelo abierto, el incremento del precio de los productos por los intermediarios, la necesidad de consumir alimentos frescos y libres de contaminación (Acosta, 2015)

2.2.1.2 Importancia de un invernadero

Los invernaderos nos dan un mayor control sobre el ambiente y el crecimiento de las plantas, se puede controlar la temperatura, luz y sombra, riego, aplicación de fertilizantes.

Los invernaderos pueden permitir que ciertos cultivos se cultiven durante todo el año, los invernaderos son cada vez más importante en el suministro de alimentos de los países. (Atom, 2013)

2.2.1.3 Aspectos Físicos de un invernadero

Los procesos que contribuyen al crecimiento y la producción de las plantas son la fotosíntesis, la respiración y la transpiración. Estos procesos son condicionados a corto y largo plazo por factores como la radiación local, la concentración de CO₂, la temperatura y la presión de vapor de agua del ambiente y a su vez en nuestro caso estas condiciones son afectadas por la existencia de la cubierta o el cerramiento del invernadero y por el propio cultivo. La cubierta del invernadero produce una atmósfera cerrada, reduciendo el intercambio de aire entre el

ambiente del cultivo y el ambiente exterior y una reducción muy marcada de la velocidad del aire en el interior. La energía captada y el vapor de agua transpirado son a su vez alterados por el cerramiento, limitando el intercambio de forma que son acumulados ambos en el ambiente interior. El calentamiento del invernadero se produce cuando el infrarrojo largo, procedente de la radiación que pasa a través del material de cubierta, se transforma en calor. Esta radiación es absorbida por las plantas, los materiales de la estructura y el suelo. Como consecuencia de esta absorción, éstos emiten radiación de longitud más larga que tras pasar por el obstáculo que representa la cubierta, se emite radiación hacia el exterior y hacia el interior, calentando el invernadero como ya se mencionó, la radiación y la concentración de CO₂ influyen en el proceso de fotosíntesis. Más concentración de CO₂, proporciona más fotosíntesis; sin embargo, concentraciones superiores pueden resultar tóxicas para los cultivos. Por otro lado, la tasa de absorción de CO₂ es proporcional a la cantidad de luz recibida. A mayor luminosidad en el interior del invernadero se debe aumentar la temperatura, la humedad y el CO₂, para que la fotosíntesis sea máxima. Para mejorar la luminosidad natural se usan materiales de cubierta con buena transparencia, y que permitan recibir la mayor radiación solar posible. La temperatura va estar en función de la radiación solar, que va a calentar el invernadero. Para el manejo de la temperatura es importante conocer las necesidades y limitaciones de la especie cultivada. Además, existe una relación inversa con la humedad; si aumenta la temperatura, disminuye la humedad relativa, y viceversa. La humedad es una variable importante para el desarrollo del cultivo, ya que si la humedad es muy baja, las plantas podrían deshidratarse; por el contrario, si es alta, favorece la proliferación de plagas ((Reyna, 2015)

2.2.1.4.1 Invernadero plano o tipo parral

Este tipo de invernadero se utiliza en zonas poco lluviosas, aunque no es aconsejable su construcción, una estructura vertical y otra horizontal La estructura de estos invernaderos se encuentra constituida por dos partes

La estructura vertical está constituida por soportes rígidos que se pueden diferenciar según sean perimetrales (soportes de cerco situados en las bandas y los esquineros) o interiores (pies derechos).

La estructura horizontal está constituida por dos mallas de alambre galvanizado superpuestas, implantadas manualmente de forma simultánea a la construcción del invernadero y que sirven para portar y sujetar la lámina de plástico.

En las ventajas esta la mayor resistencia al viento y en las desventajas su mala ventilación, poco aconsejable en lugares lluviosos, poco estanco al goteo del agua de lluvia y al aire ya que es preciso hacer orificios en el plástico para la unión de las dos mallas con alambre, lo que favorece la proliferación de enfermedades fúngicas (INFOAGRO, n.d.)



FIGURA 1 : INVERNADERO TIPO PLANO O TIPO PARRAL
(FUENTE: INFOAGRO)

2.2.1.4.2 Invernadero en raspa y amagado

Suelen ser invernaderos típicos de la zona de Almería, ya que son el resultado de una modificación de los invernaderos planos o de tipo parral. El objetivo principal con el que se construyó fue para poder evacuar el agua de la lluvia debido a que en los invernaderos planos al llover, crean grandes bolsas de agua en el techo de la estructura y las perjudica seriamente. Este tipo de invernadero consiste en que la parte alta denominada como “raspa”, esté sostenida mediante tubos galvanizados o de perfiles laminados y con alambres o trenzas de hilos de alambre.

Por el contrario, la parte baja se le conoce como “amagado”, se une a la estructura mediante horquillas de hierro sujetas a la base del invernadero. También necesita de una cimentación a base de tubos cilíndricos de hormigón para mantener la estructura bien sujeta al terreno y ocasionar futuros problemas, una de las ventajas es que tiene buen estanque de lluvia y la desventaja es que pierde calor dentro del invernadero debido a su altitud. (tu industria, 2016)



FIGURA 2: INVERNADERO EN RASPA Y AMAGADO
(FUENTE: TU INDUSTRIA)

2.2.1.4.3 Invernadero asimétrico o inacral

Difiere del tipo raspa y amagado en el aumento de la superficie en la cara expuesta al sur, con objeto de aumentar su capacidad de captación de la radiación solar. Para ello el invernadero se orienta en sentido este-oeste, paralelo al recorrido aparente del sol, permite el aprovechamiento de la luz en época de invierno, Tiene más pérdidas de calor a través de la cubierta debido a su mayor superficie desarrollada en comparación con el tipo plano (INFOAGRO, n.d.)

2.2.1.4.4 Invernadero túnel o semicilíndrico

Se caracteriza por la forma de su cubierta y por su estructura totalmente metálica. Los soportes son de tubos de hierro galvanizado y el ancho de estas naves está comprendido entre 6 y 9 m y permiten el adosamiento de varias naves en batería. La ventilación es mediante ventanas cenitales que se abren hacia el exterior del invernadero, su cubierta permite que entre mucha luz (Lopez B. , 2012)



**FIGURA 3: INVERNADERO TIPO TÚNEL
(FUENTE: INVERNADEROS CHILE)**

2.2.1.3.1 Invernadero de cristal

En la base de la estructura se utilizan pilares y canales de recogida de aguas de acero reforzados con cerchas que se instalan en el techo. Para el remate del invernadero se usan perfiles y piezas de aluminio; estos elementos nos aportan el acabado estético óptimo e incorporan los alojamientos para la colocación del material de cubierta. Suele ser habitual la creación de un pequeño muro en la base para mejorar el aspecto final del invernadero, El uso de este material para cubrir la estructura permite un incremento en el rendimiento del cultivo gracias a la transparencia del mismo y la desventaja sería que no entra mucha luminosidad.(FERTRI,n.d)



FIGURA 4: INVERNADERO DE CRISTAL
(FUENTE: FERTRI)

2.2.1.2 CONTROL DE LAS VARIABLES CLIMÁTICAS

2.2.1.2.1 Definición

El control del clima de un invernadero es muy importante dado que afecta a la respuesta del cultivo a corto y largo plazo (intercambios gaseosos, crecimiento y desarrollo), determinando las componentes del rendimiento (productividad y calidad) y, por lo tanto, la rentabilidad del sistema de cultivo.(Conference & Engineering, n.d.)

Una manera de emular cambios de temperatura dentro del invernadero es utilizando suelo radiante, el cual es un sistema de calefacción que se basa en la circulación de agua caliente o vapor a través de tuberías instaladas bajo el suelo, donde el calor se transmite por convección y conducción hacia el ambiente. Este tipo de calefacción tiene la ventaja de tener una distribución más uniforme en

comparación con los sistemas de calefacción convencionales(Fernández & Flaviano, 2013)

El sistema de control de la humedad relativa funciona de modo similar al descrito. El PLC interpreta la lectura de humedad relativa, y la rutina de control Difuso determina el porcentaje de voltaje que se aplicará a la bomba y al ventilador para que sólo circule la cantidad de agua necesaria perfeccionados (Reveles, Minjares, Varela, & Telles, 2008).

Humedad Relativa en %	Temperatura en °C		
	0 – 5 °C Mínima letal	6 – 12 °C Mínima Biológica	13 – 20 °C Óptima
0 – 30 % Baja	Pared húmeda y calefactor al máximo	Pared húmeda al máximo y calefactor a media potencia	Pared húmeda al máximo
31 – 50 % Regular	Pared húmeda a media potencia y calefactor al máximo	Pared húmeda y calefactor a media potencia	Pared húmeda a media potencia
51 – 60 % Óptima	Calefactor al máximo	Calefactor a media potencia	Pared húmeda y calefactor desactivados

TABLA 1: CONTROL DE LA TEMPERATURA Y HUMEDAD
FUENTE (REVELÉS , MINJARES , VARELA, TELLES,RIVAS,ELIAS)

Red de Sensores Inalámbricos (WSN). Es una tecnología moderna que integra el conocimiento de sensores, control de automatización, transmisión de la red digital, almacenamiento de información y procesamiento de información. (Machala, 2016)

Sensores de Temperatura y Humedad Los sensores son un conjunto de captadores capaces de medir diferentes variables, en el caso que nos ocupa las variables de temperatura y humedad. En el proyecto se utiliza sensores DHT11 [26]. Este sensor se compone de un sensor capacitivo y de un termistor para medir la humedad. Funcionan con ciclos de operación de duración determinada. Está diseñado para medir temperaturas entre 0 y 50°C con una precisión de $\pm 2^{\circ}\text{C}$

y para medir humedad entre 20% y 80% con una precisión de 5% [Acosta, A., & Aguilar, 2015]

exceso de calor en el invernadero durante la floración puede impedir una correcta polinización, dando lugar a la aparición de malformaciones en la planta (Anglés, 2001)

Las aplicaciones de los dispositivos móviles son muy variadas desde el cálculo y diseño de instalaciones hasta la supervisión y control de los procesos en los invernaderos (Molina & Jiménez, 2010)

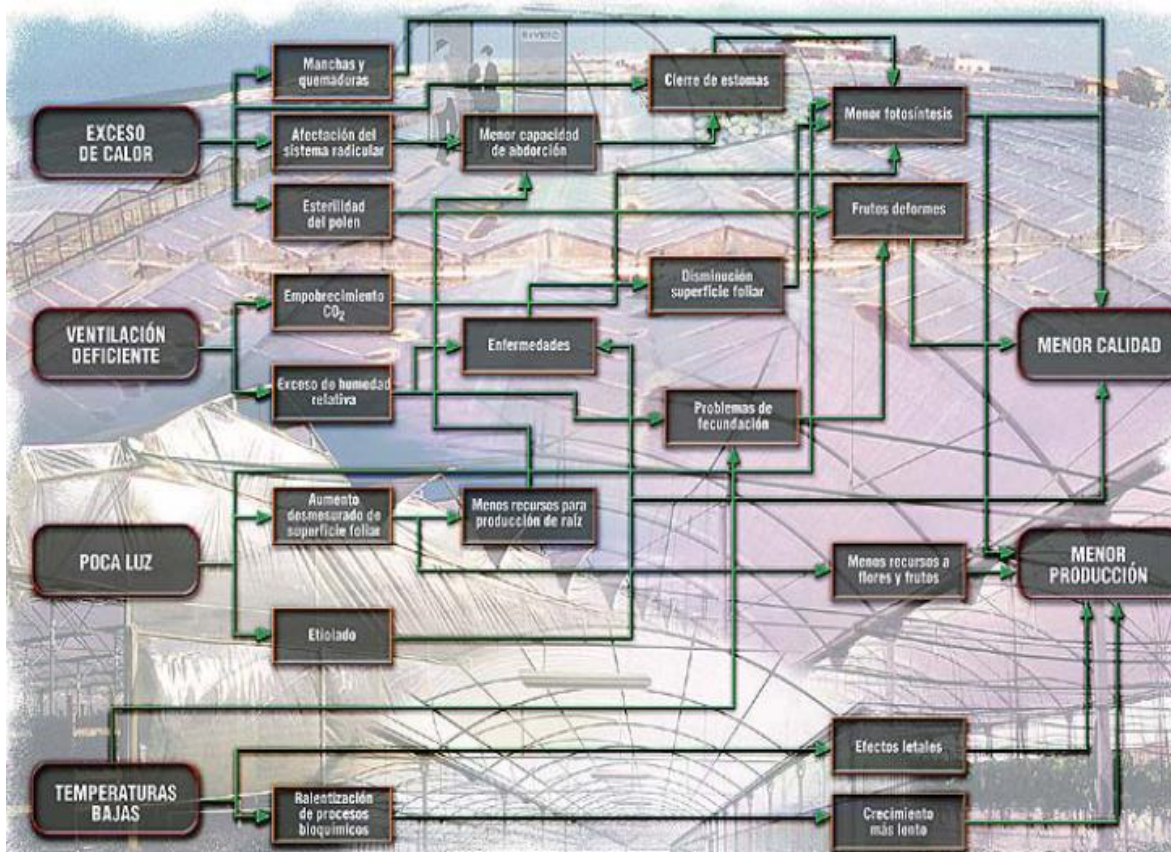


FIGURA 5: DIAGRAMA DE CONTROL DE VARIABLES CLIMÁTICAS (FUENTE: ÁNGLES)

2.2.1.2.2 PARÁMETROS A CONSIDERAR EN EL CONTROL CLIMÁTICO DEL INVERNADERO

A. Temperatura

La temperatura idónea en invernadero varía en función del cultivo y sus estadios, o etapas de desarrollo en las que se encuentre. Generalmente, la temperatura mínima requerida para las plantas de invernadero es de 10-15°C, mientras que 30°C podría ser la temperatura máxima.

Una variación o diferencia de temperatura de 5 – 7°C entre las temperaturas diurnas y nocturnas suele resultar beneficiosa para las plantas. La temperatura del suelo es incluso más importante que la temperatura del aire en un invernadero (Temperatura del suelo por debajo de 7°C, las raíces crecen más despacio y no absorben fácilmente el agua ni los nutrientes). Se debe conseguir un suelo templado, para que las semillas germinen y para se desarrollen los esquejes de raíces. La temperatura ideal para la germinación de la mayoría de las semillas es 18-25°C.

Resulta complicado regular las altas temperaturas en invernadero, sobre todo en verano. Por tanto, es conveniente disponer de un sistema de ventilación en la cubierta o contar con una malla de sombreado por fuera (Novagric, 2015)

1. Control por calefacción

Los generadores empleados, calderas de agua caliente normalmente (por combustión de biomasa, gas o gasóleo), normalmente esta calefacción es más estable

- Calefacción por agua a alta temperatura (hasta 90° C), la distribución del calor al invernadero se realiza mediante un sistema de tuberías de hierro situadas a nivel del suelo (raíles), en la canal (a modo opcional en lugares con altas precipitaciones de nieve) o a nivel de cultivo (mantiene la temperatura a nivel foliar de la planta).
- Calefacción por agua a baja temperatura ($\pm 40^{\circ}$ C), la distribución del agua caliente se realiza a través de un sistema de tuberías de polietileno colocadas a nivel de suelo próximas a la planta

Sistema de calefacción con generadores de aire caliente:

estos aparatos funcionan con gasoleo o gas y están dotados de un intercambiador de calor dentro del cual se separan los residuos de la combustión para llevarlos hacia el exterior, en la parte exterior del intercambiador un flujo de aire limpio es calentado e impulsado al interior del invernadero. Un adecuado número y disposición de los generadores, junto con un adecuado caudal de aire y dardo de alcance de los ventiladores, permite una distribución adecuada del aire caliente en todo el volumen del invernadero. (Ima invernaderos, s.f.)

2. Control por ventilación

El control se lleva a cabo con la instalación de ventilación, principalmente las ventanas instaladas cenitales La tasa de ventilación de un invernadero depende:

- del grado de apertura de la instalación de ventilación
- de la velocidad del viento
- y de la diferencia de temperatura entre el interior y el exterior. (Ima invernaderos, s.f.)

B. Humedad relativa del aire

Las humedades de la mayoría de las plantas prefieren una humedad relativa del aire entre el 45 y el 60%. Para su control se coloca sensores de humedad en el interior del invernadero (higrómetro). Humedades altas favorecen la transmisión de plagas y enfermedades, y las humedades bajas podrían secar la planta. La transpiración de las hojas aumenta la humedad de un invernadero. Si el grado de humedad es demasiado bajo, podemos elevarlo mojando el suelo o vaporizando las plantas de forma periódica.

Cuando la HR es excesiva las plantas reducen la transpiración y disminuyen su crecimiento, se producen abortos florales por apelmazamiento del polen y un mayor desarrollo de enfermedades criptogámicas. Por el contrario, si es muy baja, las plantas transpiran en exceso, pudiendo deshidratarse, además de los comunes problemas de mal cuaje.

El exceso puede reducirse mediante ventilado, aumento de la temperatura y evitando el exceso de humedad en el suelo. La falta puede corregirse con riegos, llenando canalillas o balsetas de agua, pulverizando agua en el ambiente, ventilado y sombreado. La ventilación cenital en invernaderos con anchura superior a 40 m es muy recomendable, tanto para el control de la temperatura como de la HR. (INFOAGRO, n.d.)

C. Iluminación

A mayor luminosidad en el interior del invernadero se debe aumentar la temperatura, la HR y el CO₂, para que la fotosíntesis sea máxima; por el contrario, si hay poca luz pueden descender las necesidades de otros factores. Para mejorar la luminosidad natural se usan los siguientes medios:

- Materiales de cubierta con buena transparencia.
- Orientación adecuada del invernadero.
- Materiales que reduzcan el mínimo las sombras interiores.
- Aumento del ángulo de incidencia de las radiaciones sobre las cubiertas.
- Acolchados del suelo con plástico blanco.

En verano para reducir la luminosidad se emplean:

- Blanqueo de cubiertas.
- Mallas de sombreado.
- Acolchados de plástico negro.

Es interesante destacar el uso del blanqueo ya que esta labor está en función del desarrollo del cultivo y de las temperaturas, y tiene efectos contradictorios que hay que conocer para hacer un correcto uso. Hay que saber que la planta sombreada se ahila y se producen abortos de flores en determinadas especies sensibles a la luz (especialmente tomate, pimiento y berenjena), por lo que el manejo del riego y de la solución nutritiva tiene que ir unida al efecto que produce el blanqueo. Los plásticos sucios o envejecidos provocan el mismo efecto que el blanqueo. (INFOAGRO, n.d.)

D. Humedad del suelo

El monitorear el contenido de agua en el suelo es esencial para ayudar a los agricultores a optimizar la producción, conservar agua, reducir los impactos ambientales y ahorrar dinero. El monitorear la humedad del suelo le puede ayudar a tomar mejores decisiones en la programación del riego, tales como el determinar la cantidad de agua a aplicar y cuándo aplicarla. También le puede ayudar a igualar los requerimientos de agua del cultivo con la cantidad aplicada con el riego; y así evitar pérdidas de agua excesivas por percolación profunda o por escurrimientos o bien evitar aplicar una cantidad insuficiente. El exceso de irrigación puede incrementar el consumo de energía y los costos de agua, aumentar el movimiento de fertilizantes por debajo de la zona radicular, producir erosión y transporte de suelo y partículas de químicos a los canales de drenaje. El riego insuficiente puede reducir la producción de las cosechas (Enciso & Porter, 2018)

1. Control utilizando un interruptor horario

A la hora fijada, dicho interruptor cierra su contacto de salida, dando tensión al relay., este a su vez, cierra el circuito del receptor, en este caso una electroválvula. Si el interruptor horario no da tensión directamente a la electroválvula es porque, una vez calculada, la intensidad eléctrica que consume resulta superior a la que puede suministrar el contacto de salida del interruptor horario. De ahí la necesidad de instalar un relé intermedio (Navas, Benavente, & Elena, 2000)

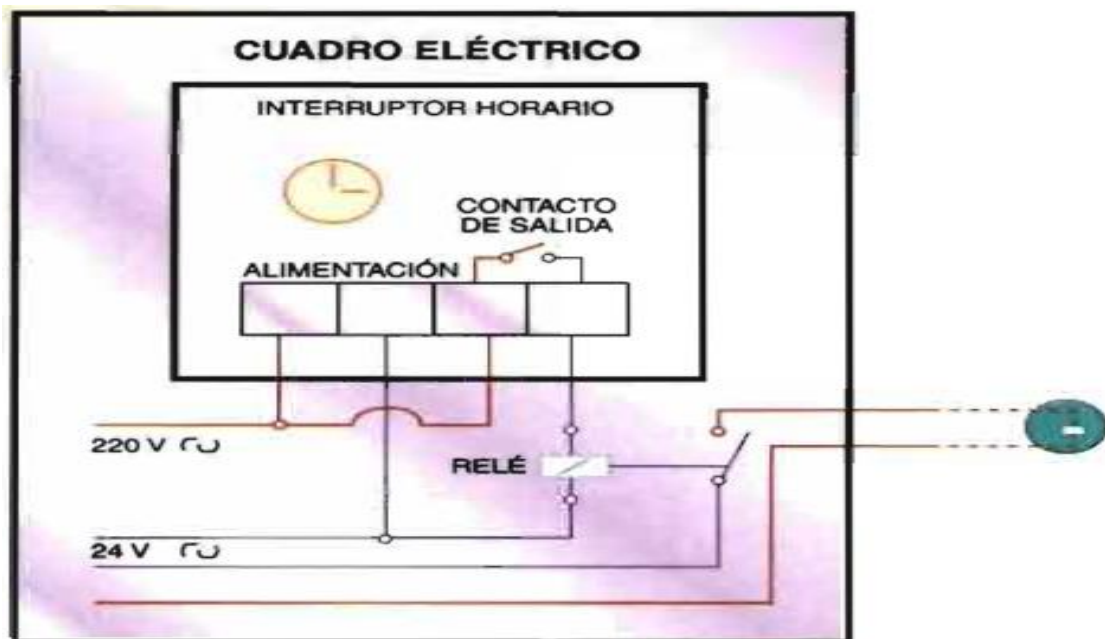


FIGURA 6: CONTROLADOR INTERRUPTOR HORARIO
(FUENTE: NAVAS)

2.2.2 SISTEMA ARDUINO Y SENSORES

TIPOS DE MICROCONTROLADORES

2.2.2.1 Tipos de microcontroladores

plc que permiten reprogramación, consumen poca potencia, realiza operaciones aritméticas, mayor capacidad de almacenamiento (Acosta, 2015)

Arduino hardware libre, micro controlador, reprogramable, compatibilidad con software libre, amplio conjunto de sensores de bajo costo. (Acosta, 2015)

PIC 16F628 un reducido número de instrucciones de longitud fija (Acosta, 2015)

Dispositivos temporizadores contador activo y desactiva una acción cada cierto tiempo, no programable, plataforma cerrada (Acosta, 2015)

Dispositivo	Características	Precio
PLC Controlador Lógico Programable	<ul style="list-style-type: none"> ○ Permiten reprogramación ○ Consumen poca potencia ○ Realiza Operaciones Aritméticas ○ Mayor Capacidad de Almacenamiento. 	1000\$-4000\$
Arduino	<ul style="list-style-type: none"> ○ Hardware libre ○ Microcontrolador reprogramable ○ Compatibilidad con software libre. ○ Amplio conjunto de sensores de bajo costo ○ Conexiones de forma sencilla 	50.00\$-90.00\$
PIC 16F628	<ul style="list-style-type: none"> ○ Un reducido número de instrucciones de longitud fija. ○ Instrucciones duran 1 tiempo de instrucción 	6.00\$-10.00\$
Dispositivos Temporizadores	<ul style="list-style-type: none"> ○ Contador activa y desactiva una acción cada cierto tiempo. ○ No programable ○ Plataforma cerrada. 	24.00\$ - 90.00 \$

TABLA 2: CUADRO COMPARATIVO DE CONTROLADORES
(FUENTE: ACOSTA)

2.2.2.2 Tipos de sensores para medir las variables climáticas

El control climático en invernaderos es posible gracias al uso de sensores capaces de medir las variables climáticas como temperatura, humedad ,radiación solar ,velocidad y dirección del viento (Perdigones & Pascual, 2004)

2.2.2.2.1 Sensores de humedad y temperatura

Mide la resistencia entre el cursor y un extremo (Perdigones & Pascual, 2004), también podríamos decir que es un dispositivo capaz medir la humedad relativa (Kouro, 2001)

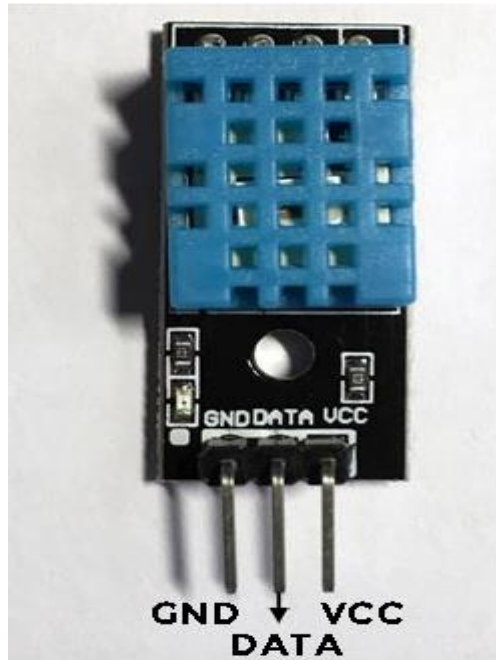


FIGURA 7: SENSOR DHT11
(FUENTE: PROGRAMAR FÁCIL)

Los pines del DHT11 son:

- GND: conexión con tierra
- DATA: transmisión de datos
- VCC: alimentación

2.2.2.2 Sensores de humedad del suelo

Consiste en un elemento sensitivo ,una fuente de tensión auxiliar y un instrumento de medida, su rango de medida suele ser de -40 grados a 60 grados con una precisión de 0.1 grados centígrados ,la ventaja en los invernaderos de las sondas a cuatro hilos ya que el sensor puede estar a 20 o 30 cm de distancia respecto al cuadro de medida (Perdigones & Pascual, 2004)

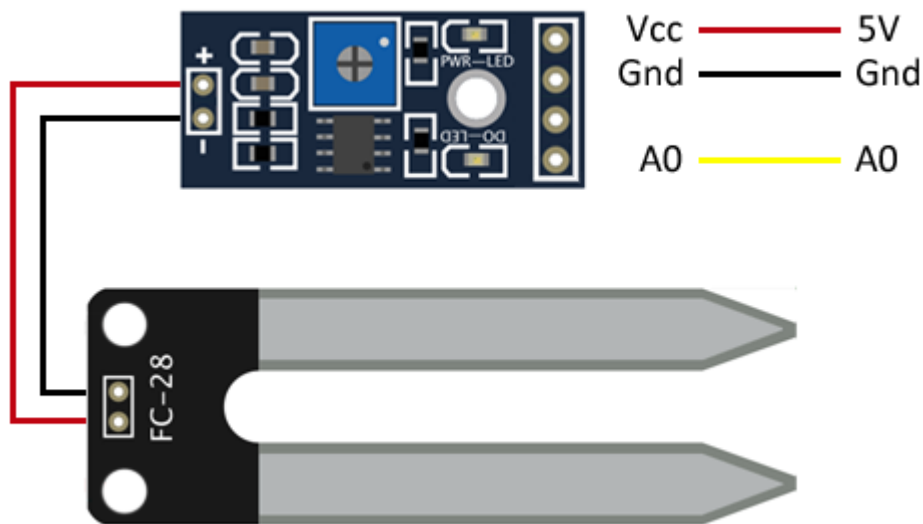


FIGURA 8: SENSOR HIGRÓMETRO (FC-28)
(FUENTE: INGENIERÍA, INFORMÁTICA, DISEÑO)

2.2.3 APLICACIONES MÓVILES

2.2.3.1 Definición

El mercado de aplicaciones móviles ha experimentado una rápida expansión durante los últimos 10 años, las plataformas móviles siguen mejorando su desempeño, y la necesidad de los usuarios de una amplia variedad de aplicaciones móviles va en aumento. El desarrollo de software para las plataformas móviles viene con características únicas y restricciones que se aplican a la mayoría de las etapas del ciclo de vida. Las características distintivas más importantes son: un alto nivel de competitividad, corto tiempo de entrega, movilidad, portabilidad, capacidades específicas y constantemente cambiantes de las terminales, sistemas operativos diferentes e incompatibles, entre otras (Amaya, 2013)

2.2.3.2 METODOLOGÍA BASADA EN SCRUM

Los procesos de Scrum abordan las actividades específicas y el flujo de un proyecto de Scrum. En total hay diecinueve procesos de Scrum

INICIO.

1 Crear la visión del proyecto: En este proceso se revisa el caso de negocio del proyecto (Project Business Case) a fin de crear una Declaración de la visión del proyecto, que servirá de inspiración y proporcionará un enfoque para todo el proyecto. En este proceso se identifica al Product Owner.

2. Identificar al Scrum Master y Stakeholder(s): En este proceso se identifica al Scrum Master y stakeholders utilizando criterios de selección específicos.

3. Formar Equipos Scrum: En este proceso se identifican a los miembros del Equipo Scrum. Normalmente, el Product Owner es el responsable principal de la selección de los miembros del equipo, pero con frecuencia lo hace en colaboración con el Scrum Master.

4. Desarrollar épica(s): En este proceso la Declaración de visión del proyecto sirve como base para el desarrollo de épicas. Se pueden llevar a cabo reuniones de grupos de usuarios para hablar sobre las épicas adecuadas.

5. Crear el Backlog Priorizado del Producto: En este proceso se refinan y se crean las épicas, y después se priorizan para crear un Backlog Priorizado del Producto para el proyecto. A este punto también se establecen los criterios de terminado.

6. Realizar la planificación del lanzamiento: En este proceso el equipo principal de Scrum revisa las historias de usuario en el Backlog Priorizado del Producto para desarrollar un cronograma de planificación del lanzamiento, que es esencialmente un programa de implementación por fases que se puede compartir con los stakeholders del proyecto. En este proceso también se determina la duración del sprint. (Study, 2017)

PLANIFICACIÓN Y ESTIMACIÓN

1. Crear historias de usuario: En este proceso se crean las historias de usuario y los criterios de aceptación de las historias de usuario. Las historias de usuario generalmente las escribe el Product Owner y están diseñadas para asegurar que

los requisitos del cliente estén claramente representados y puedan ser plenamente comprendidos por todos los stakeholders.

2. Comprometer historias de usuario: En este proceso, el Equipo Scrum se compromete a entregar al Product Owner las historias de usuario aprobadas para un sprint. El resultado de este proceso serían las historias de usuario comprometidas.

3. Estimar historias de usuario: En este proceso, el Product Owner aclara las historias de usuario para que el Scrum Master y el Equipo Scrum puedan estimar el esfuerzo necesario para desarrollar la funcionalidad descrita en cada historia de usuario. En este proceso, el Equipo Scrum se compromete a entregar al Product Owner las historias de usuario aprobadas para un sprint. El resultado de este proceso serían las historias de usuario comprometidas.

4. Identificar tareas: En este proceso, las historias de usuario comprometidas se desglosan en tareas específicas y se compilan en una lista de tareas.

5. Estimar tareas: En este proceso, el equipo principal de Scrum estima el esfuerzo necesario para cumplir con cada tarea en la lista de tareas. El resultado de este proceso es una: Effort Estimated Task List.

6. Crear el Sprint Backlog: En este proceso, el equipo principal de Scrum elabora un Sprint Backlog que contiene todas las tareas a ser completadas en un sprint como parte de la Reunión de Planificación del Sprint. 1.4.4.3 Implementación (Study, 2017)

IMPLEMENTACIÓN

1. Crear entregables: En este proceso, el Equipo Scrum trabaja en las tareas en el Sprint Backlog para crear los entregables del sprint. Generalmente se utiliza un Scrumboard para dar seguimiento a las actividades que se llevan a cabo. Los asuntos o problemas que enfrenta el equipo Scrum pudieran actualizarse en un Impediment Log (o registro de impedimentos).

2. Realizar Daily Standup : En este proceso, se lleva a cabo diariamente una reunión altamente focalizada con un time-box, conocida como Daily Standup. Es

aquí donde los miembros del Equipo Scrum se actualizan el uno al otro referente a sus progresos y sobre los impedimentos que pudieran enfrentar.

3. Refinamiento del Backlog Priorizado del Producto—En este proceso, el Backlog Priorizado del Producto se actualiza y se refina continuamente. Se puede considerar realizar una reunión de revisión del Backlog Priorizado del Producto, en la que se analiza cualquier cambio o actualización al backlog.

REVISIÓN Y RETROSPECTIVA

1. Demostrar y validar el sprint—En este proceso, el Equipo Scrum muestra los entregables del sprint al Product Owner y a los stakeholders relevantes en una Reunión de Revisión del Sprint. El propósito de esta reunión es asegurar que se obtenga la aprobación y aceptación del Product Owner respecto a los entregables elaborados en el sprint.

2. Retrospectiva del sprint: En este proceso, el Scrum Master y el Equipo Scrum se reúnen para analizar las lecciones aprendidas durante todo el Sprint. Esta información se documenta en forma de lecciones aprendidas que pueden aplicarse a futuros sprints. (Study, 2017)

LANZAMIENTO

1. Enviar entregables—En este proceso, los entregables aceptados se entregan o se envían a los stakeholders relevantes.

2. Retrospectiva del proyecto: En este proceso, mismo que concluye el proyecto, los stakeholders y miembros del equipo principal de Scrum se reúnen para hacer una retrospectiva del proyecto e identificar, documentar e internalizar las lecciones aprendidas. (Study, 2017)

2.2.3.3 GESTOR DE BASE DE DATOS

FIREBASE:

La principal ventaja de esta base de datos es que al estar alojada en la nube de Google no hay necesidad de mantener servidores, y el SDK de Android permite que la sincronización sea fácil de implementar, utilizando las funciones que Firebase proporciona. Al ser una base de datos noSQL, es importante estructurar bien los datos. Para ello se han seguido las directivas proporcionadas por Firebase en su documentación donde informan de que todos los datos de Firebase Realtime Database se guardan en formato JSON. Es decir, que no existen tablas ni registros para guardar los datos, y cada vez que se agregan datos al árbol JSON, estos se convierten en un nodo en la estructura JSON existente (García, 2017)

2.3 DEFINICIÓN DE TERMINOS BÁSICOS

1. Arduino:

Arduino es una plataforma de hardware y software de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación "Processing" (Amangandi, 2012)

2. Sensor de humedad:

El sensor DHT11 es un módulo que nos permite medir la humedad relativa y temperatura ambiental. Este sensor tiene una resolución de 1°C y un rango de trabajo de 0°C a 50°C para la temperatura, y resolución de 1% y un rango de trabajo desde el 20% hasta el 95% de humedad relativa. El sensor trabaja con una tensión de 3.3 V a 5 V (Dea, 2014)

3. Sensor de humedad del suelo

Este sensor es ideal para monitorear el nivel de humedad de tus plantas y así recordar cuando necesitan ser regadas o incluso para realizar un sistema totalmente automatizado de riego. Si el sistema se conecta a internet podríamos controlar/monitorear nuestro invernadero desde cualquier lugar del mundo. (Sensor de humedad del suelo, n.d.)

4. Invernadero

Un invernadero es lugar cerrado, estático y accesible a pie, que se destina a la producción de cultivos, dotado habitualmente de una cubierta exterior translúcida de vidrio o plástico, que permite el control de la temperatura, la humedad y otros factores ambientales para favorecer el desarrollo de las plantas. En la jardinería antigua española, el invernadero se llamaba estufa fría (Sanchez Castro, 2012)

5. Android:

Android es un sistema operativo móvil basado en el kernel de Linux, con una interfaz de programación Java, diseñado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros. Desarrollado por la Open Handset Alliance la cual es liderada por Google. Android permite programar aplicaciones en una variación de Java llamada Dalvik. El sistema operativo proporciona todas las interfaces necesarias para desarrollar Estudio y desarrollo de aplicaciones para dispositivos móviles Android, aplicaciones que accedan a las funciones del teléfono (GPS, llamadas, sms, agenda, entre otras.) de una forma muy fácil en un lenguaje de programación muy popular como es Java. (Monica, 2013)

CAPÍTULO III: DESARROLLO DEL TRABAJO DE SUFICIENCIA PROFESIONAL

El proyecto se dividió en 2 módulos uno para el hardware y el otro módulo para el software.

En esta imagen vemos como la arquitectura de la aplicación móvil: para poder visualizar las variables climáticas la placa arduino manda mediante el módulo ESP8266 mediante internet los datos de las variables de humedad del suelo, temperatura y humedad relativa a la base de datos Firebase y esta envía los datos a la aplicación y la aplicación envía datos firebase de estado 1 o 0 para que se active o desactive el riego.

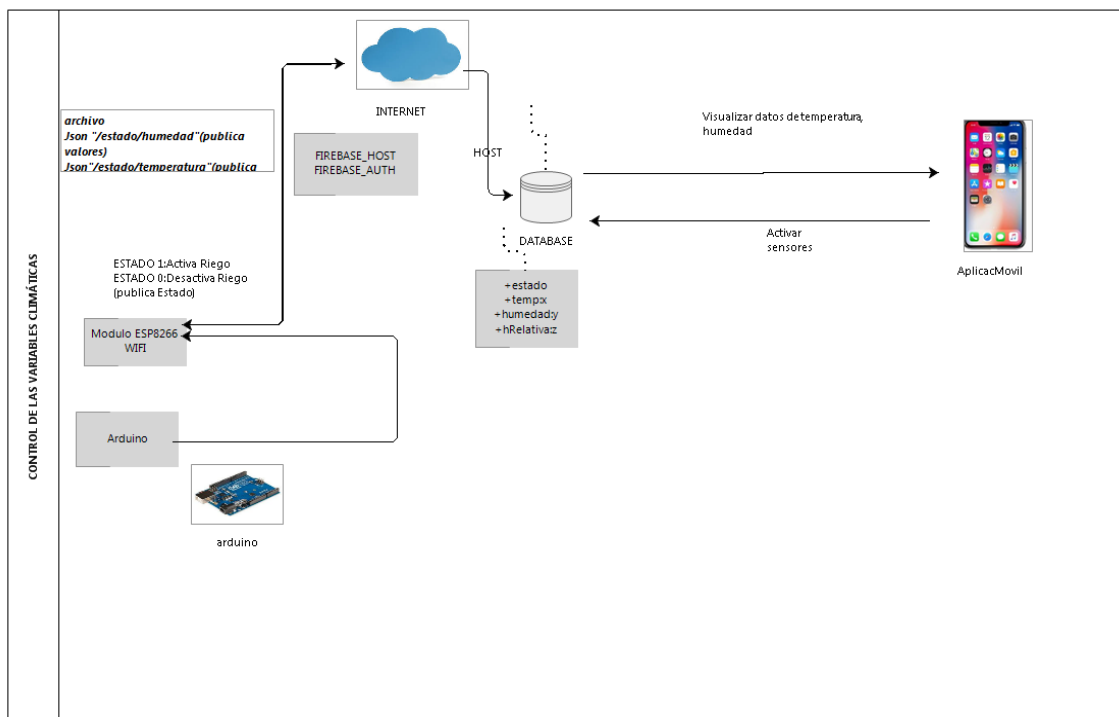


FIGURA 9: ARQUITECTURA DE LA APLICACIÓN MÓVIL (FUENTE: PROPIA)

3.1 DESARROLLO DEL HADWARE CON LA TECNOLOGIA ARDUINO

3.1.1 REQUERIMIENTOS DE EQUIPOS PARA CAPTURAR LAS VARIABLES CLIMÁTICAS

3.1.1.1 Sistema microcontrolador

Por todas las características que nos brinda sumado al bajo precio de mercado y a su versatilidad para realizar montajes electrónicos de control, registró y censado de variables físicas se optó por micro controlador Arduino UNO.

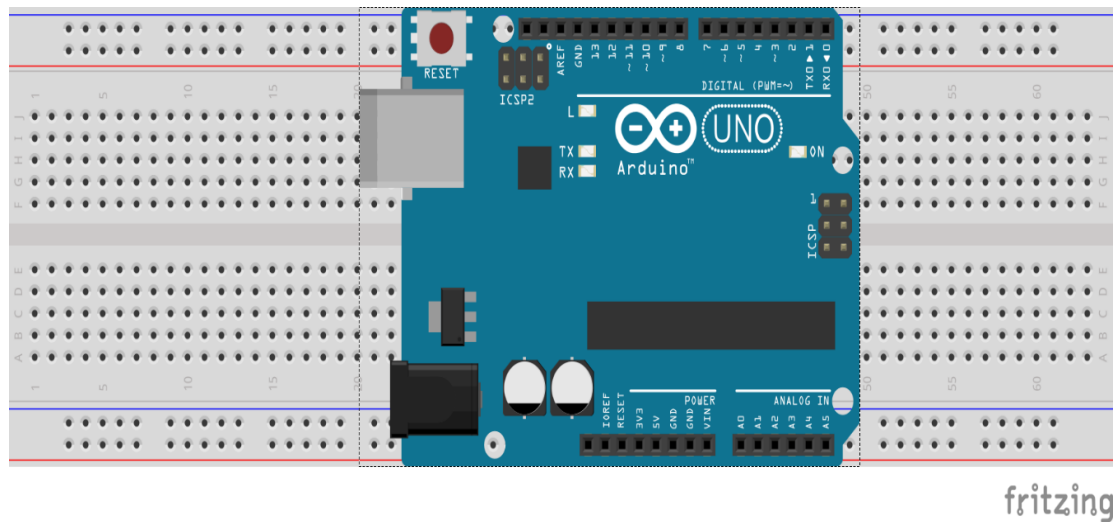


FIGURA 10: MICROCONTROLADOR ARDUINO UNO
(FUENTE: PROPIA)

3.1.1.2 Sensores

1.- Hardware

Todos los componentes del sistema juegan un papel fundamental en el funcionamiento. Sin embargo, el microcontrolador es el engranaje en nuestro proyecto, ya que si este dispositivo no funciona correctamente, no es relevante sensores y actuadores cumplan óptimamente sus funciones, de igual manera el sistema no cumplirá la finalidad de la obtención y visualización de los parámetros del invernadero. Esto debido a que, el microcontrolador, es la encargada de almacenar toda la información, y relacionarla con la obtenida desde los diferentes sensores de humedad y temperatura (DHT11,DHT22,higrómetro), y según el control desde la App se genere un riego en un instante.

Para la obtención de los valores de temperatura y humedad relativa se optó por los sensores DHT11 y DHT22 siendo los mismos seleccionados por sus características las cuales datan de mucho mayor estabilidad y mejor precisión en la entrega de los valores de sensores, al ser de formato digital se utilizara un pin digital por cada uno de ellos, los cuales serán respectivamente el pin 9 y pin 10. Para la obtención de la variable de humedad del suelo se seleccionó el higrómetro FC-28 el cual entrega sus valores en formato analógico para los cuales se utilizaran los pines analógicos A0 y A1 respectivamente. En si cada de cada planta se obtendrá tres variables de sensores temperatura, humedad relativa y humedad del suelo.

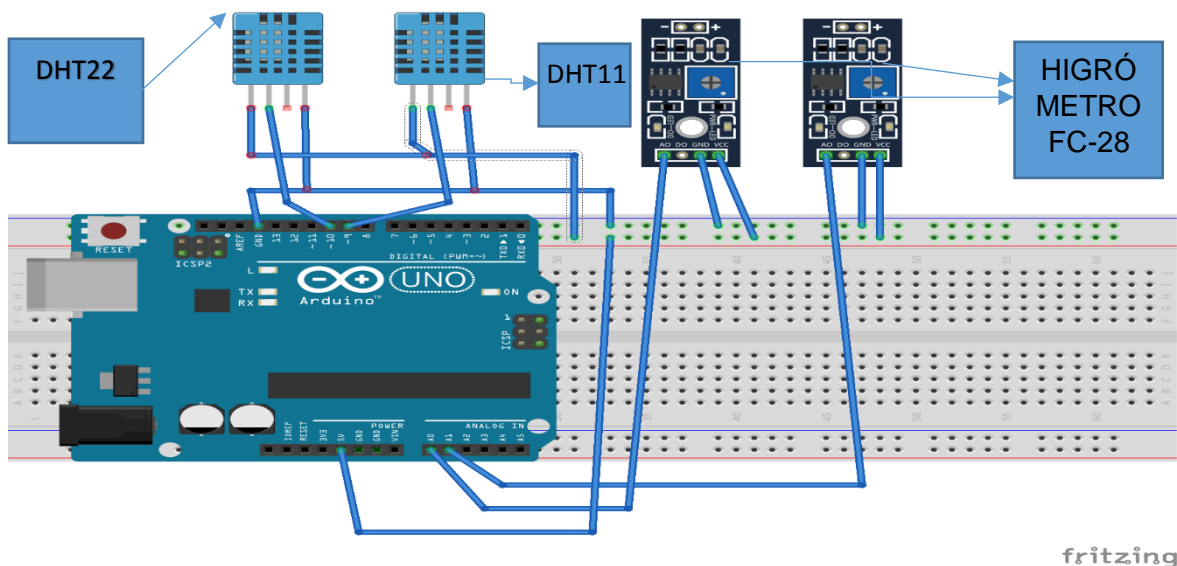


FIGURA 11: SENSORES DE TEMPERATURA Y HUMEDAD (FUENTE: PROPIA)

3.1.1.1.3 Lcd(Liquid Crystal Display)

Para la visualización de las variables en el proyecto de forma local se optó por un display lcd 2*16 , el cual utiliza 6 pines para comunicarse con el microcontrolador y para esto se dispuso de los siguientes pines : pin digital D7 → RW, pin digital D6→ E, pines digitales D5-D2 para el envío de datos desde el microntrolador hacia el lcd.

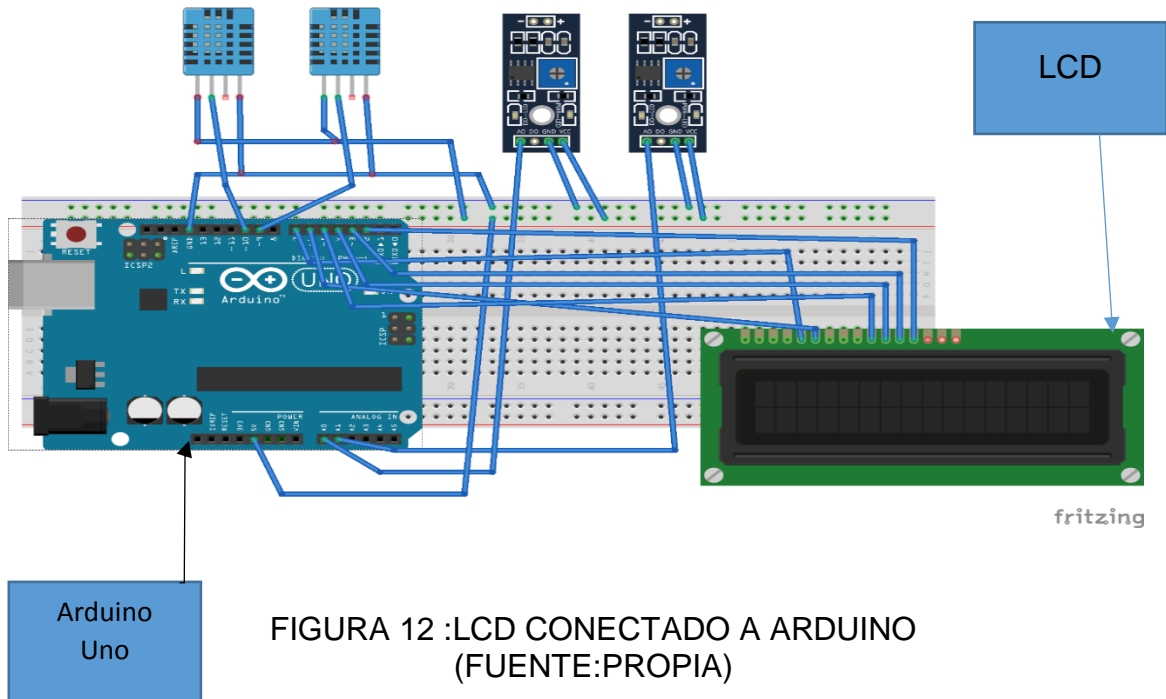
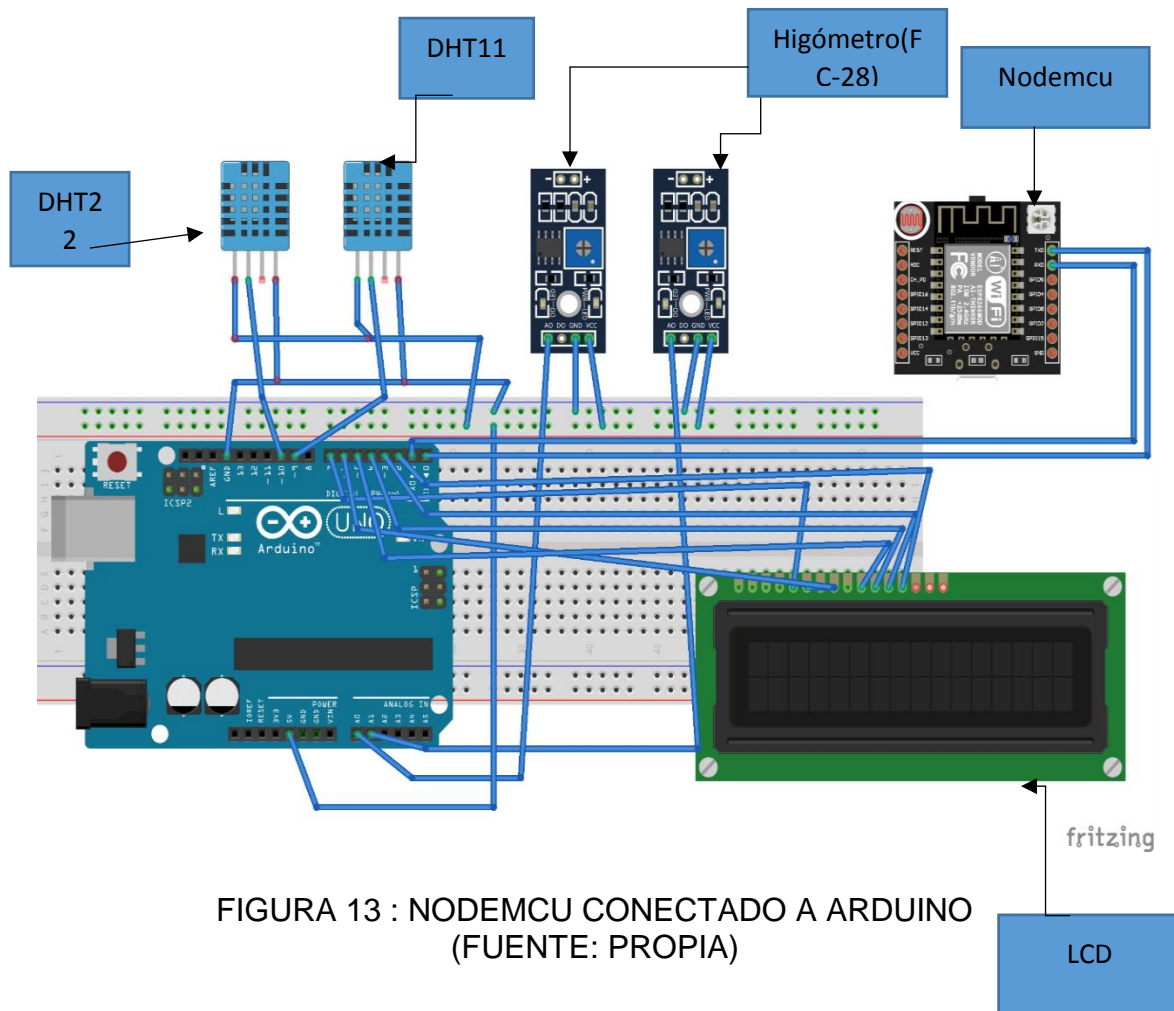


FIGURA 12 :LCD CONECTADO A ARDUINO
(FUENTE:PROPIA)

3.1.1.1.3 Nodemcu

Para la transmisión y recepción de los valores de los sensores y los estados de riego de cada planta respectivamente , su utilizo el protocolo de comunicación UART , en el microcontrolador para poder hacer uso de ello se debe seleccionar los pines D1 y D0 (TX y RX) los cuales irán conectados hacia el nodemcu el cual es un microcontrolador incluido con un periférico wifi el cual le permite tener conectividad hacia el internet , a través de ello se puede implementar soluciones de conectividad hacia una base de datos , un servidor , una de la solución que nosotros vayamos a implementar . En este caso se establece un comunicación hacia una base de datos FIREBASE de tiempo real, ya que el registro de datos ocurre instantáneamente se logre publica los valores de los sensores. Los pines de TX y RX del nodemcu irán conectados de forma inversa a lo pines del microcontrolador de arduino lo está , esto quiere decir que el pin TX de arduino ira conectado hacia el pin RX de nodemcu y respectivamente lo mismo para el pin RX del arduino conectado con el pin TX del nomdecu, además se tendrá que compartir el pin de GND para los dos dispositivos ya que esto es muy importante para poder referenciar los diferentes niveles de voltaje TTL que son enviados a través de los cables de transmisión .



3.1.1.1.4 Relay y motores de riego

Luego de haber agregado el nodemcu hacia el diseño , se agregara dos relay para el accionar de los motores de cada riego para cada planta, esto se realiza ya que mientras se realizaba las pruebas no se llegaba accionar los motores con los pines digitales del microcontrolador de Arduino y esto se debía al poco amperaje que entregaba los pines . Para dar una solución a esto se optó por una fuente externa de 5v que alimentara a los relay y así poder generar un voltaje adecuado para el control ON/OFF de los motores.

Se optó por este método ya que al aislar las alimentaciones de voltaje de los relay y motores de la alimentación principal de los sensores no se genere corrientes parasitas que induzcan valores extraños de los sensores.

Además se agregó un capacitor de 1000uF para estabilizar el voltaje que se obtenía del microcontrolador el cual se repartía hacia los demás sensores.

Por último se agregó un potenciómetro al lcd para poder regular el contraste del mismo.

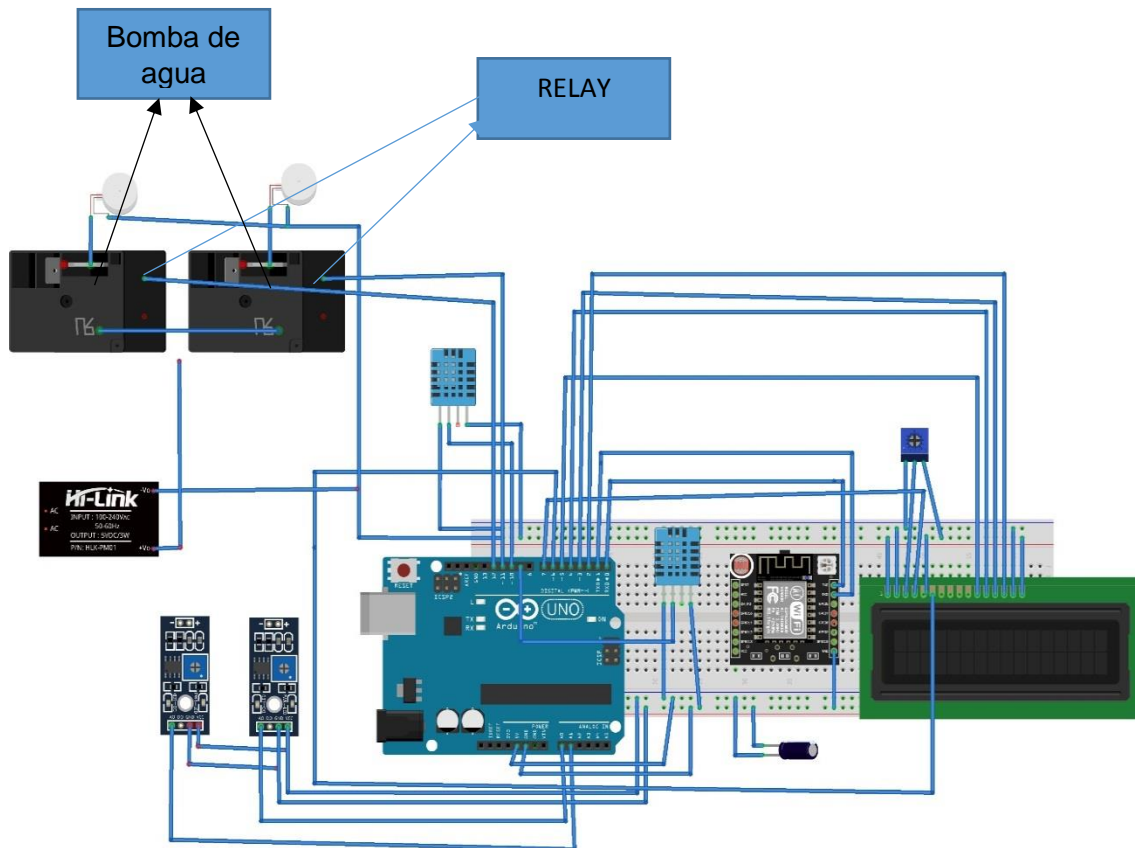


FIGURA 14: ESQUEMA GENERA ARDUINO (FUENTE: PROPIA)

3.1.2 DESARROLLO DEL SOFTWARE EN ARDUINO

Se desarrollaron dos programas uno para el arduino y el otro para el Nodemcu , los cuales se detallaran a continuación,la versión que tenemos instalado es la arduino 1.8.8.

3.1.2.1 ARDUINO

Primeramente incluimos las librerías que vamos a ocupar, en este caso del sensor DHT11 y del lcd 2*16

```
#include "DHT.h"  
  
#include <LiquidCrystal.h>
```

Luego configuramos los pines que utilizaremos para cada dispositivo los cuales son el lcd y los sensores digitales, los sensores analógicos no necesitan configurarse ya que por defecto siempre serán entradas del microcontrolador. vemos que para los sensores DHT los pines a utilizar son para el DHT11 se define el pin 9 y para el DHT22 se define el pin 10.

```
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); //(RS,E,D4,D5,D6,D7)

//CONFIGURACION DEL SENSOR DHT11 Y DHT22

#define DHTPIN 9

#define DHTPIN2 10

#define DHTTYPE DHT11

#define DHTTYPE2 DHT22

DHT dht(DHTPIN,DHTTYPE);

DHT dht2(DHTPIN2,DHTTYPE2);
```

Definimos nuestras variables globales que serán utilizadas en nuestra programación

```
int riego1; //ESTADO DEL RIEGO1

int riego2; //ESTADO DEL RIEGO2

int pinriego1 = 11; //PIN DE SALIDA DEL RIEGO1

int pinriego2 = 12; //PIN DE SALIDA DEL RIEGO2

int valor; //VARIABLE GLOBAL QUE ALMACENA EL VALOR DE LECTURA DEL SENSOR DE HUMEDAD

int valorMaximo = 500;//VARIABLE PARA COLOCAR VALOR MAXIMO

int val;

int val1;
```

Definimos además una variable de tipo byte el cual tendrá como función generar el simbolo de grado en el lcd , esto es asi ya que caracteres especiales se deben de configurar e implementar de una manera distinta porque sera hará uso de la SDRAM del lcd.

```
byte grado[8] =  
{  
0b00001100,  
0b00010010,  
0b00010010,  
0b00001100,  
0b00000000,  
0b00000000,  
0b00000000,  
0b00000000
```

Luego dentro del void setup se inicializa todos los dispositivos, comunicación serial y definición de Entrada y Salida para los pines digitales, se agregó también una visualización inicial del lcd donde se vea el siguiente mensaje “Invernadero Maciel”.

```

void setup() {

//CONFIGURACION INICIAL

dht.begin();//INICIALIZACION DEL SENSOR dht11

dht2.begin();//INICIALIZACION DEL SENSOR dht22

Serial.begin(115200); //COMUNICACION Y VELOCIDAD DEL MONITOR
SERIAL Y ARDUINO

lcd.begin(16, 2);

pinMode(pinriego1,OUTPUT); //DEFINIMOS COMO PIN DE SALIDA
pinMode(pinriego2,OUTPUT); //DEFINIMOS COMO PIN DE SALIDA

lcd.setCursor(5,0); //

lcd.print("invernadero");

lcd.setCursor(5,1);

lcd.print("Maciel");

delay(2000);

lcd.clear();

digitalWrite(pinriego1,HIGH);

digitalWrite(pinriego2,HIGH);

}

```

Lo siguiente es definir las funciones que se realizaran repetitivamente dentro del bucle principal, estos son las funciones de obtención de cada sensor temperatura, humedad relativa y humedad del suelo respectivamente.


```

int sensorT() {

    //int temp= thermocouple.readCelsius();// lectura del sensor con la libreria

    int t = dht.readTemperature();

    delay(500);

    lcd.createChar(1,grado);

    lcd.setCursor(0,0);

    lcd.print("T:");

    lcd.print(t);

    lcd.write(1);

    lcd.print("C");

    return t;

}

int sensorH(){

    int lectura = analogRead(0); //realizamos la lectura del sensor A0

    lectura = map(lectura,400,1023,100,0);//mapeamos O CAMBIAMOS LOS VALORES
DE 0 A 100%

    lcd.setCursor(6,0);

    lcd.print("H:");

    lcd.print(lectura);

    lcd.print("%");

    return lectura;
}

```

```

}

int sensorH2(){
    int h2 = dht.readHumidity();//PARA OBTENER LA HUMEDAD RELATIVA DEL
    SENSOR
    delay(500);
    lcd.setCursor(11,0);
    lcd.print("h:");
    lcd.print(h2);
    lcd.print("%");
    return h2;
}

int sensor2T() {
    int t2 = dht2.readTemperature(); //PARA OBTENER LA TEMERATURA DEL
    SENSOR
    lcd.createChar(1,grado);
    lcd.setCursor(0,1);
    lcd.print("T:");
    lcd.print(t2);
    lcd.write(1);
    lcd.print("C");
    return t2;
}

```

```

int sensor2H(){
    int lectura2 = analogRead(1); //realizamos la lectura del sensor A1

    lectura2 = map(lectura2,400,1023,100,0);//mapeamos O CAMBIAMOS LOS
    VALORES DE 0 A 100%

    lcd.setCursor(6,1);

    lcd.print("H:");

    lcd.print(lectura2);

    lcd.print("%");

    return lectura2;

}

int sensor2H2(){
    int h22 = dht2.readHumidity();

    delay(500);

    lcd.setCursor(11,1);

    lcd.print("h:");

    lcd.print(h22);

    lcd.print("%");

    return h22;

}

```

Además se implementó una función llamada “activarriego()” que obtenga los valores del estado de riego , dependiendo del estado realizara un estado distinto del pin digital al cual ha sido asignado , para el estado de riego de la planta 1 y la planta 2 son los pines D11 y D12 respectivamente.

```
void activarriego(){
  if(riego1 == 1){
    digitalWrite(pinriego1,LOW);
    delay(500);
  }
  if(riego1 == 0){
    digitalWrite(pinriego1,HIGH);
    delay(500);
  }
  if(riego2 == 1){
    digitalWrite(pinriego2,LOW);
    delay(500);
  }
  if(riego2 == 0){
    digitalWrite(pinriego2,HIGH);
    delay(500);
  }
}
```

En el bucle principal ira una función llamada “recibirdata ()” el cual alberga las demás funciones y las ejecuta de una manera secuencial y del cual se estableció un orden de envío de datos para que al recepcionarlos en el dispositivo nodemcu se obtengan de manera correcta.

```
void loop()

{//BUCLE INFINITO REPETITIVO

  recibirdata();

}

void recibirdata(){

  int sensor1 = sensorT();

  int sensor2 = sensorH();

  int sensor3 = sensorH2();

  int sensor4 = sensor2T();

  int sensor5 = sensor2H();

  int sensor6 = sensor2H2();

  Serial.print("I"); //inicio de trama

  Serial.print(sensor1);

  Serial.print(":");

  Serial.print(sensor2);

  Serial.print(":");

  Serial.print(sensor3);

  Serial.print(":");

  Serial.print(sensor4);

  Serial.print(":");
```

```
Serial.print(sensor5);

Serial.print(":");

Serial.print(sensor6);

Serial.print("F"); //fin de trama

Serial.flush(); //Esperamos hasta que se envíen los datos

delay(100);

//---Leemos la respuesta del Esclavo-----

if(Serial.find("i"))//esperamos el inicio de trama
{
    riego1=Serial.parseInt(); //recibimos valor numérico
    if(Serial.read()==':') //Si el fin de trama es el correcto
    {
        riego2 = Serial.parseInt();
        if(Serial.read()=='f')
            activarriego();
    }
}

}
```

3.1.2.2 NODEMCU

Primeramente incluimos las librerías que vamos a ocupar, en este caso la utilización del enlace Wifi y la conexión hacia la base de datos Firebase.

Lo que vamos a hacer es importar la librería firebase-arduino-master

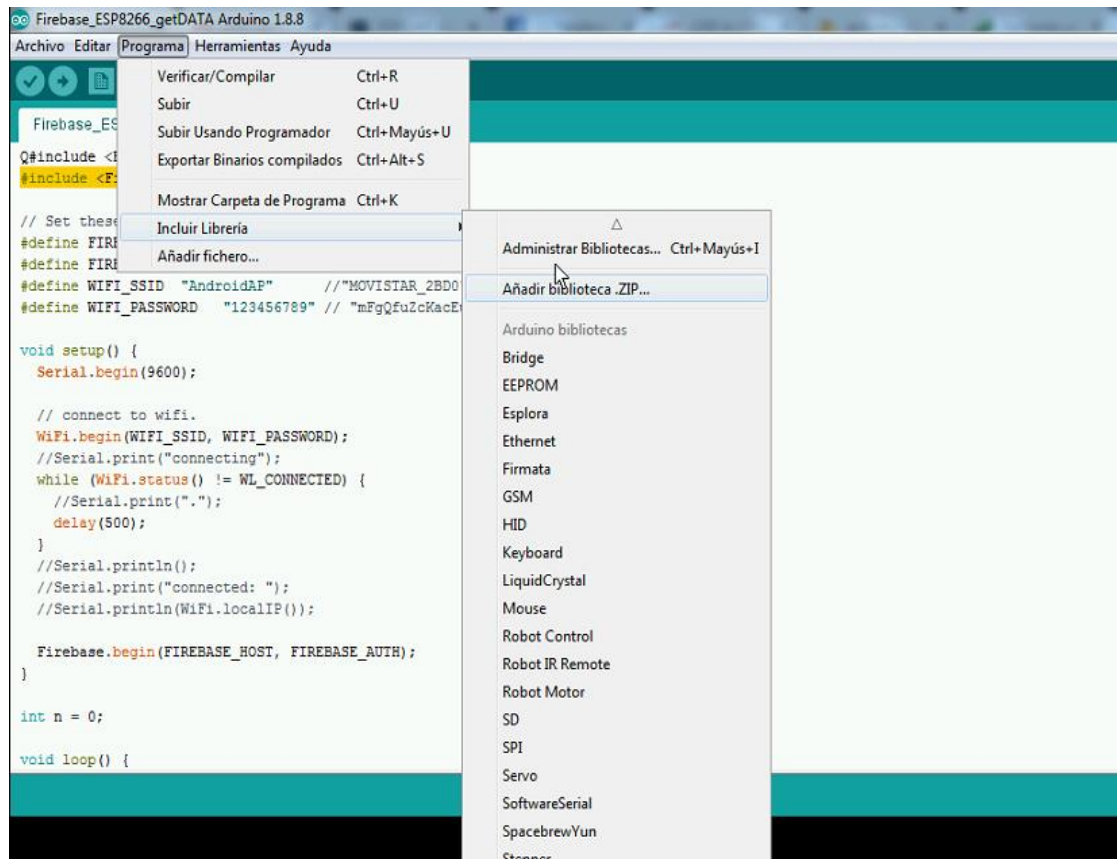


FIGURA 15. IMPORTACIÓN EN ARDUINO DE LIBRERÍA FIREBASE (FUENTE: PROPIA)

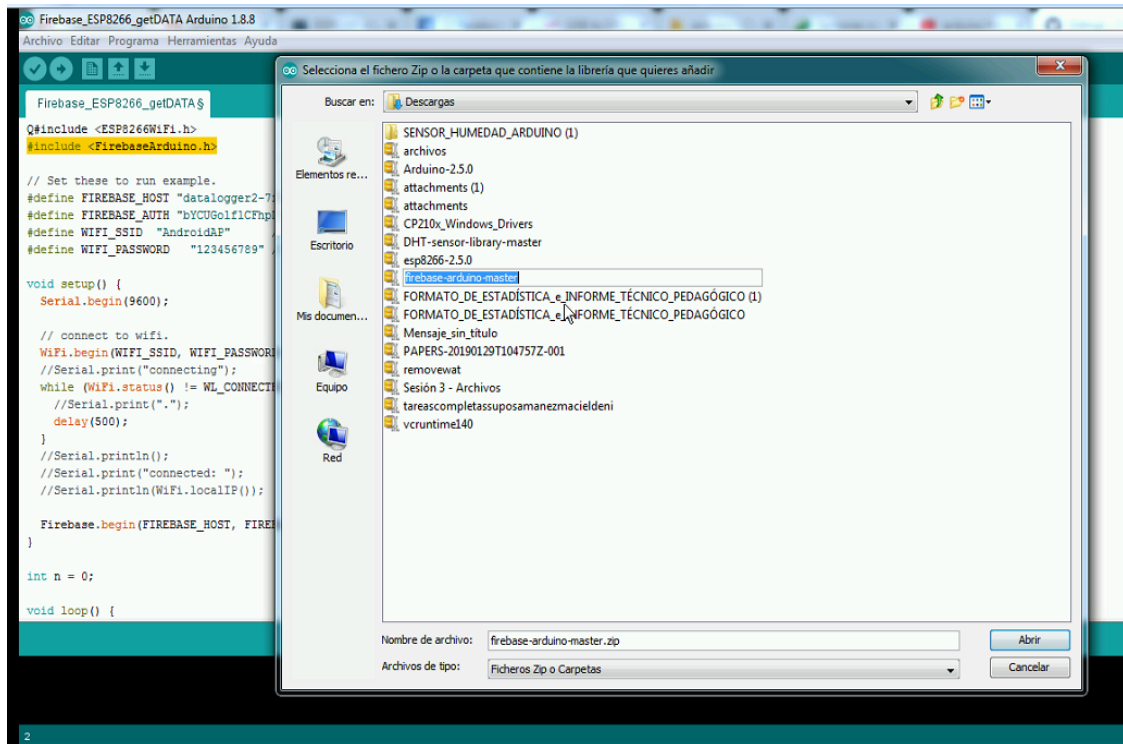


FIGURA 16: IMPORTACIÓN PARA LA LIBRERÍA FIREBASE-ARDUINO-MASTER (FUENTE:PROPIA)

Luego importamos la librería arduino-Json en la versión estable 5.13.1

Però tiene dependencia con <https://github.com/bblanchon/ArduinoJson>

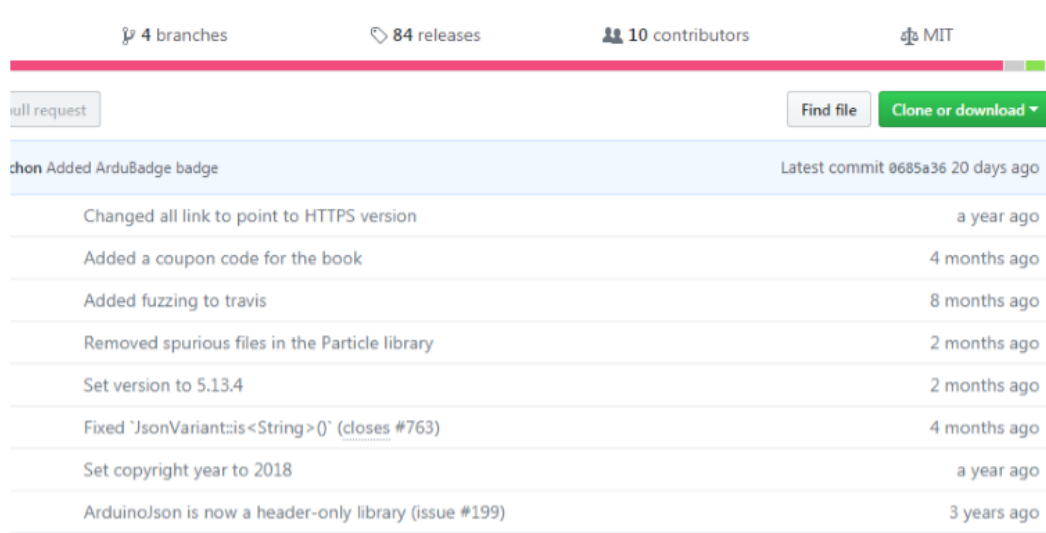


FIGURA 17: LIBRERÍA ARDUINO JSON (FUENTE: PROPIA)

Importamos la librería ESP8266

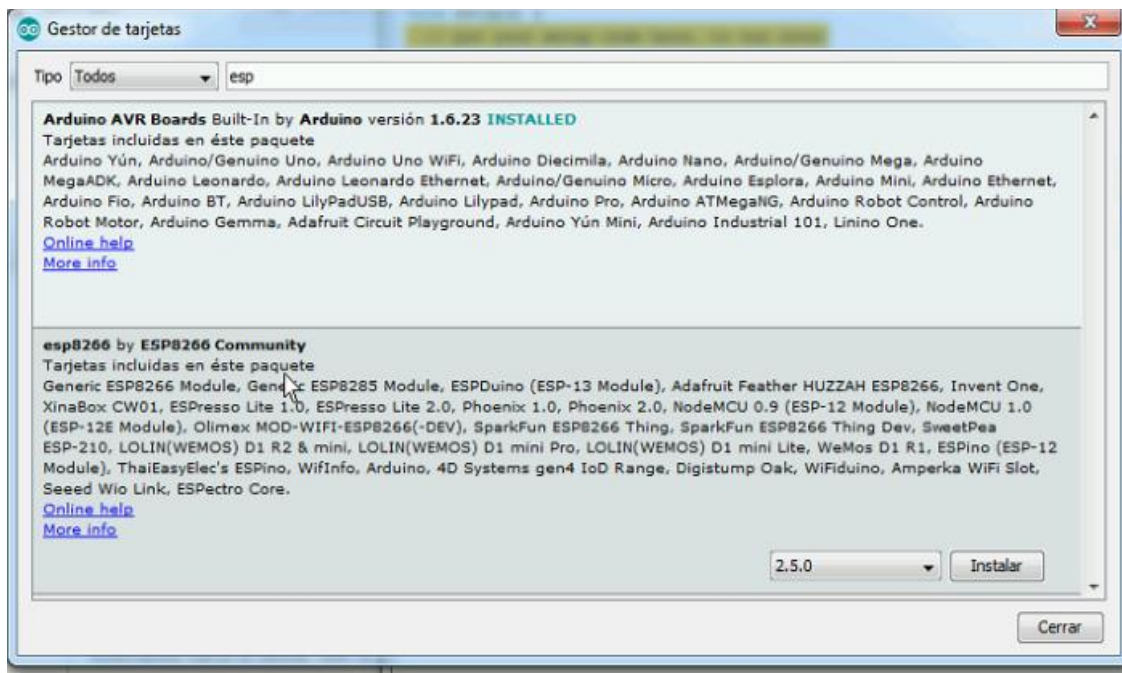


FIGURA 18: IMPORTACIÓN DE LA LIBRERÍA ESP8266
(FUENTE: PROPIA)

```
#include <ESP8266WiFi.h>  
  
#include <FirebaseArduino.h>
```

Se define los parámetros del host; el cual es el nombre la base de datos de Firebase que hemos creado, el auth o key ; el cual se dispone como clave certificadora de autenticación con las base de datos, el SSID o nombre de la red a la cual se conectara el módulo nodemcu y la contraseña.

```
#define FIREBASE_HOST "arduino-4485f.firebaseio.com"

#define FIREBASE_AUTH "57oXRgoksfZyUn0SeIpApwy7UE87Rr2j9uslJav"

#define WIFI_SSID "AndroidAP" //"MOVISTAR_2BD0"

#define WIFI_PASSWORD "123456789" //"mFgQfuZcKacEuq8gNqfe"

Lo que definimos en FIREBASE_HOST lo obtenemos de aquí:
```

Lo que va en FIREBASE_AUTH es la clave que se encuentra en la base de datos en firebase que creamos

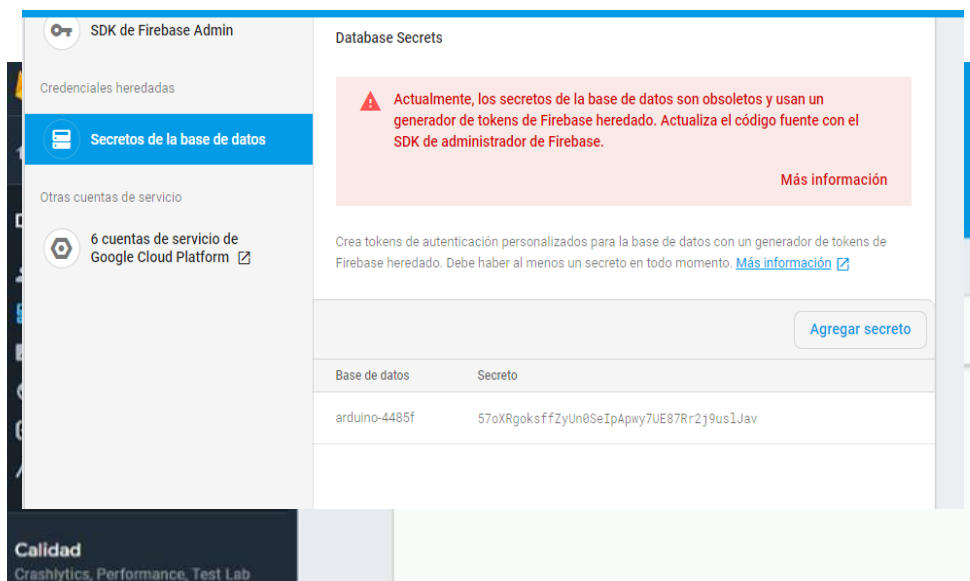


FIGURA 19: SECRETO DE LA BASE DE DATOS (FUENTE: PROPIA)

Lo que va en WIFI_SSID va la red WIFI que se va a conectar en NodemCu y en WIFI_PASSWORD ira la contraseña de la red wifi

Definimos las variables globales, ya que estas seran solicitadas en multiples partes del programa

```
int sensor1; //humedad
int sensor2; //temperatura
int sensor3; //humedadrelativa
int sensor4; //temperatura2
int sensor5; //humedad2
int sensor6; //humedadrelativa2
```

Inicializamos las comunicaciones por el puerto serie del arduino y el módulo WIFI, además de mostrar si se realizó con éxito o no las comunicaciones, como un medio de detección de errores. Por último luego de haber establecido una conexión Wifi se inicializa la conexión a la base de datos con los parámetros ya establecidos anteriormente.

```
void setup() {
  Serial.begin(115200);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  //Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    //Serial.print(".");
    delay(500);
  }
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); }
```

Las siguientes funciones se definieron para poder publicar los valores de cada sensor hacia la base de datos, para cada uno de ello se creó un child diferente con el propósito de tener una vista de los diferentes sensores.

```
void funciontemp(int dato)
{
  Firebase.setInt("/planta1/temperatura", dato);
  Serial.println("Temperaturaf");
  Serial.println(dato);
  delay(100);
}

void funcionhumedad(int dato)
{
  Firebase.setInt("/planta1/humedad", dato);
  Serial.println("humedadf");
  Serial.println(dato);
  delay(100);
}

void funciontemp2(int dato)
{
  Firebase.setInt("/planta2/temperatura2", dato);
  Serial.println("Temperatura2f");
  Serial.println(dato);
  delay(100);
}
```

```
void funcionhumedad2(int dato)
{
    Firebase.setInt("/planta2/humedad2", dato);
    delay(100);
    Serial.println("humedad2f");
    Serial.println(dato);
}

void funcionhumedadrelativa(int dato)
{
    Firebase.setInt("/planta1/HumedadRelativa", dato);
    Serial.println("HumedadRelativaf");
    Serial.println(dato);
    delay(100);
}

void funcionhumedadrelativa2(int dato)
{
    Firebase.setInt("/planta2/HumedadRelativa2", dato);
    Serial.println("HumedadRelativa2f");
    Serial.println(dato);
    delay(100);
}
```

Al igual que con el Arduino , en el bucle principal se estableció una función llamada “estadoriego()” que albergaba todas las demás funciones y las ejecutaba de manera secuencial y ordenada para la correcta recepción de los datos del arduino , seguidamente de su envió hacia la base de datos.

```
void loop() {  
  
    estadoriego();  
  
}
```

```
void estadoriego(){  
    if(Serial.available()>0){ //ESTA DISPONIBLE EL PUERTO SERIAL  
  
        if(Serial.read()=='I'){//inicio de la trama  
  
            sensor1 = Serial.parseInt();  
  
            if(Serial.read() == ':'){  
  
                sensor2 = Serial.parseInt();  
  
                if(Serial.read() == ':'){  
  
                    sensor3 = Serial.parseInt();  
  
                    if(Serial.read() == ':'){  
  
                        sensor4 = Serial.parseInt();  
  
                        if(Serial.read() == ':'){  
  
                            sensor5 = Serial.parseInt();  
  
                            if(Serial.read() == ':){
```

```

sensor6 = Serial.parseInt();

funciontemp(sensor1);

    funcionhumedad(sensor2);

    funciontemp2(sensor4);

    funcionhumedad2(sensor5);

    funcionhumedadrelativa(sensor3);

    funcionhumedadrelativa2(sensor6);

if(Serial.read()=='F') //indica el final de la trama
{
    int riego1 = Firebase.getInt("estados/estadoplanta1");
    int riego2 = Firebase.getInt("estados/estadoplanta2");

    Serial.print("i");//iniio de envio de la trama

    Serial.print(riego1);

    Serial.print(":");

    Serial.print(riego2);

    Serial.print('f');//fin de la trama
}
}

    }

        }

            }

                }

                    }

}    }

```

3.2 DESARROLLO SOFTWARE DE LA APLICACIÓN MÓVIL

3.2.1 ANALISIS DEL PROCESO

Para identificar las variables climáticas requeridas para el desarrollo de las plantas en el invernadero de la UNTELS se entrevistó a la persona encargada de la administración del invernadero

3.2.2. ENTREVISTA

En el invernadero de la Untels los procesos que presenta son:

➤ **Proceso de Monitoreo**

El proceso de monitoreo, inicia cuando la persona encargada del invernadero monitorea las variables climáticas de acuerdo a las condiciones climáticas y a su experiencia si el suelo esta húmedo, si la temperatura esta elevada, media o baja, etc.

➤ **Proceso de Control de riego en el invernadero**

El proceso de control de riego inicia cuando la persona encargada del invernadero de acuerdo a su experiencia verifica mediante el tacto que tan húmedo esta la tierra y por consiguiente inicia el proceso de riego.

3.3 MARCO DE DESARROLLO BASADO EN LA METODOLOGÍA SCRUM

3.3.1 Roles

Los roles dentro de este proyecto serán asignados de acuerdo a la función que cumpla cada persona en la realización del proyecto la cual debe llevar un orden asi como el control de tareas.

- Product Owner: David Licappa
- Product Master :Teodoro Diaz Leyva

- Equipo Scrum: Supo Samanez Maciel (programador)

3.3.2 Product Backlog.

Se presentará la lista de historias de usuario priorizada. El cual será elaborada por el product owner.

Esta Lista es como representar los casos de uso de sistema, el marco de trabajo ágil nos permite obviar la documentación y empezar a trabajar sobre poca documentación.

Se presentará la lista de historias de usuario priorizada. El cual será elaborada por el product owner.

Esta Lista es como representar los casos de uso de sistema, el marco de trabajo ágil nos permite obviar la documentación y empezar a trabajar sobre poca documentación.

3.3.2.1. Historia de Usuarios.

Las historias de usuario representan los casos de uso de sistema dentro del proyecto, así mismas están presentan una estructura fácil y sencilla de entender, también nos detalla las tareas que se debe realizar o desarrollar en cada historia.

Prioridad.

Esta característica será establecida por el product owner, por ser la persona que conoce el flujo de negocio de la encuesta.

Las prioridades que se establecen son:

- Alta.
- Media.
- Baja.

Valor.

El valor de cada historia de usuario será establecido por el equipo de desarrollo el cual se pondrá de acuerdo para darle este valor.

Se usará el planning póker para establecer valor a las historias de usuario cuando el equipo de desarrollo no este acuerdo en el valor de una historia.

Planning póker.

En este caso se establecer un juego de cartas parecido a la serie Fibonacci, lo cuales será:

0,1/2, 1, 2, 3, 5, 8, 13, 21.

CASOS DE USO DEL SISTEMA

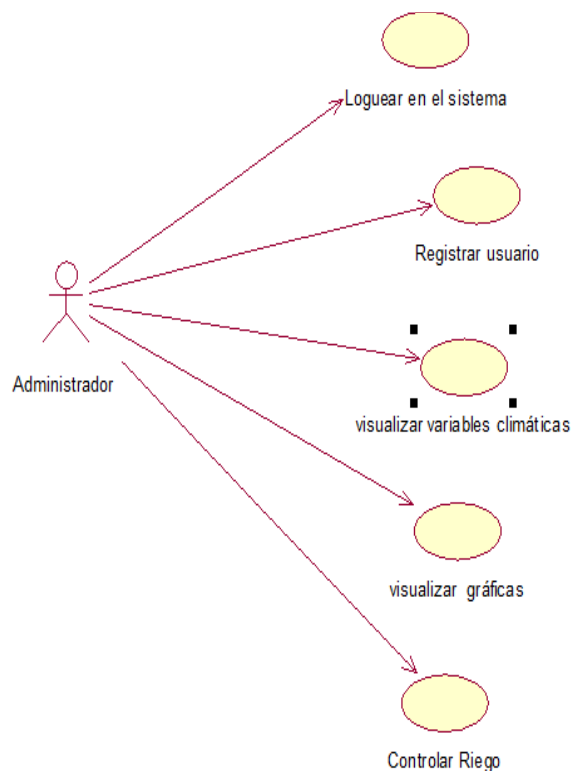


FIGURA 20: CASOS DE USO DEL SISTEMA
(FUENTE: PROPIA)

HISTORIAS DE USUARIO DEL PROCESO DE CONTROL DE LAS VARIABLES CLIMÁTICAS

Historia de Usuario		
Número: 001	Usuario: Encargado del invernadero	
Nombre historia: Login		
valor: 13	prioridad: Alta	print: 1
Descripción: Como usuario (Encargado) necesito ingresar a la aplicación mediante un usuario y contraseña por seguridad para poder ingresar a la interfaz de visualización de las variables climáticas de humedad del suelo, temperatura ambiente, humedad relativa.		
Precondición: Registro de Usuario		
Requerimientos: <ol style="list-style-type: none"> 1. Debe mostrar la pantalla donde visualizó una imagen de un invernadero. 2. Debe mostrar un botón de login 3. Debe mostrar un botón de registrar 4. Si no existe el usuario debería mostrar un mensaje "usuario no registrado" 		

Historia de Usuario		
Número: 002	Usuario: Encargado	
Nombre historia: Registro de usuario		
valor: 21	prioridad: Alta	print: 1
Descripción: Como usuario (Encargado) o alumno necesito registrarme para poder ingresar a la aplicación mediante un usuario y contraseña por seguridad para así poder visualizar las variables climáticas en tiempo real		
Precondición: Ninguna		
Requerimientos: <ol style="list-style-type: none"> 1. Debe mostrar una ventana donde me permita ingresar mi email y contraseña. 2. La caja de usuario debe aceptar letras y números. 3. Debe mostrar una caja de texto donde se ingresa el nombre del usuario. 4. Debe mostrar una caja de texto donde se ingresara el número de celular 5. Debe mostrar un botón que diga registrar 6. Debe mostrar un botón que diga cancelar 		

Historia de Usuario		
Número: 003	Usuario: Encargado	
Nombre historia: Visualización de las variables climáticas(temperatura y humedad)		
valor: 21	prioridad: Alta	print: 2
Descripción: Como usuario (Encargado) o alumno necesito visualizar las variables climáticas de humedad y temperatura en tiempo real		
Precondición: Registro de usuario,login		
Requerimientos: <ol style="list-style-type: none"> 1. Debe mostrar una ventana donde me permita visualizar la humedad del suelo, temperatura, humedad relativa. 2. Debe mostrar la información de los rangos climáticos adecuado de la planta 3. Debe mostrar un botón que me indique salir 		

Historia de Usuario		
Número: 004	Usuario: Encargado	
Nombre historia: Visualización de una gráfica de variación de (temperatura , humedad del suelo,humedad relativa del aire)		
valor: 13	prioridad: Media	print: 2
Descripción: Como usuario (Encargado) o alumno necesito visualizar una gráfica que me indique como está variando la humedad y temperatura		
Precondición: Registro de usuario,Logueo		
Requerimientos: <ol style="list-style-type: none"> 1. Debe mostrar una ventana donde me permita visualizar un primer gráfico donde me permita visualizar el último valor de la temperatura que captura en tiempo real. 2. Debe mostrar una ventana donde me permita visualizar un segundo gráfico donde me permita visualizar el último valor de la humedad del suelo que captura en tiempo real. 3. Debe mostrar una ventana donde me permita visualizar un tercer gráfico donde me permita visualizar el último valor de la humedad relativa que captura en tiempo real. 		

Historia de Usuario		
Número: 005	Usuario: Encargado	
Nombre historia: Control de riego		
valor: 21	prioridad: Alta	print: 2
Descripción: Como usuario (Encargado) o alumno necesito emitir un control de riego en las plantas en el invernadero de la UNTELS.		
Precondición: Registro de usuario, visualización de las variables climáticas, logueo		
Requerimientos: <ol style="list-style-type: none"> 1. La aplicación debe mostrar un botón que me indique activar riego 2. La aplicación debe mostrar un botón que me permita desactivar el riego. 3. La aplicación debe mostrar un botón de atrás . 		

N°	Historia de Usuario	Prioridad	Valor
1	Login	Alta	13
2	Registro Usuario	Alta	21
3	Visualización de las variables climáticas	Alta	21
4	Visualización de gráficas	Media	13
5	Control de Riego	Alta	21

TABLA 3: SPRING BACKLOG
(FUENTE: PROPIA)

3.3.3. Sprint Planning.

En esta etapa se planeará y definirá los Sprint que el equipo debe entregar al asesor en los cuales deben entregar software funcionando.

En esta ocasión definiremos dos fases. Primero, comprometernos con el trabajo, es decir, comprometernos con una cantidad de trabajo para cada sprint. Por el otro, detallar qué tareas deberemos realizar para terminar con todo el trabajo que se comprometió a presentar después de cada sprint.

Definiremos dos fases importantes dentro del sprint planning.

- **Estratégica:** se definirá el qué se va a trabajar cada sprint
- **Táctica:** se definirá el cómo. ¿Cómo se afrontará el desarrollador para completar cada sprint

Después de lo anterior se establecerá el detalle de a sprint

Sprint 1				
Item	Historia de Usuario	Prioridad	Valor	Tiempo(días)
1	Login	Alta	13	8
2	Registro de Usuario	Alta	21	9
Total			42	17

TABLA 4: SPRING BACKLOG DE SPRINT 1
(FUENTE: PROPIA)

Sprint 2				
Item	Historia de Usuario	Prioridad	Valor	Tiempo(días)
3	Visualización variables climáticas	Alta	21	15
4	Visualización de graficas	Media	13	7
5	Control de Riego	Alta	21	8
Total			55	30

TABLA 5: SPRING BACKLOG DE SPRINT 2.
(FUENTE: PROPIA)

3.3.3.1 Prototipos

PROTOTIPO DE LOGIN

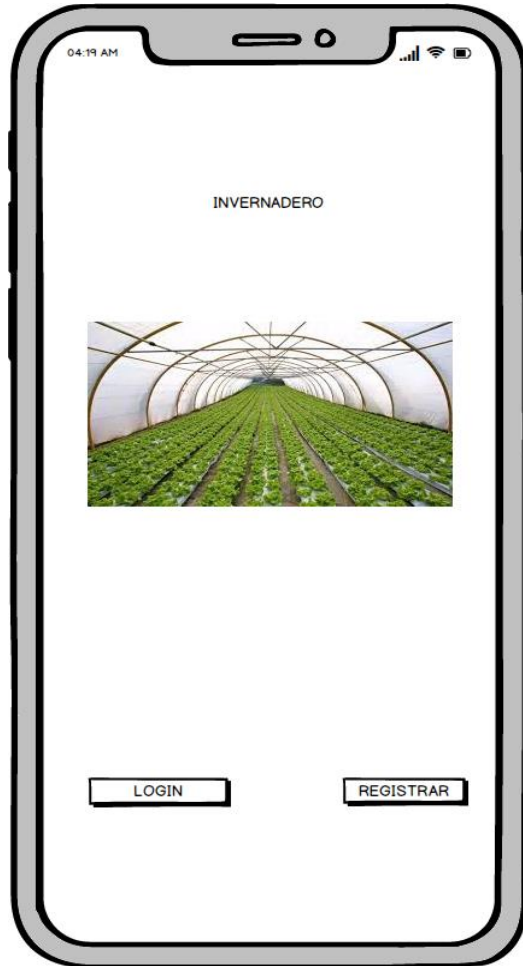


FIGURA 21: PROTOTIPO DE LOGIN
(FUENTE: PROPIA)

PROTOTIPO REGISTRO USUARIO

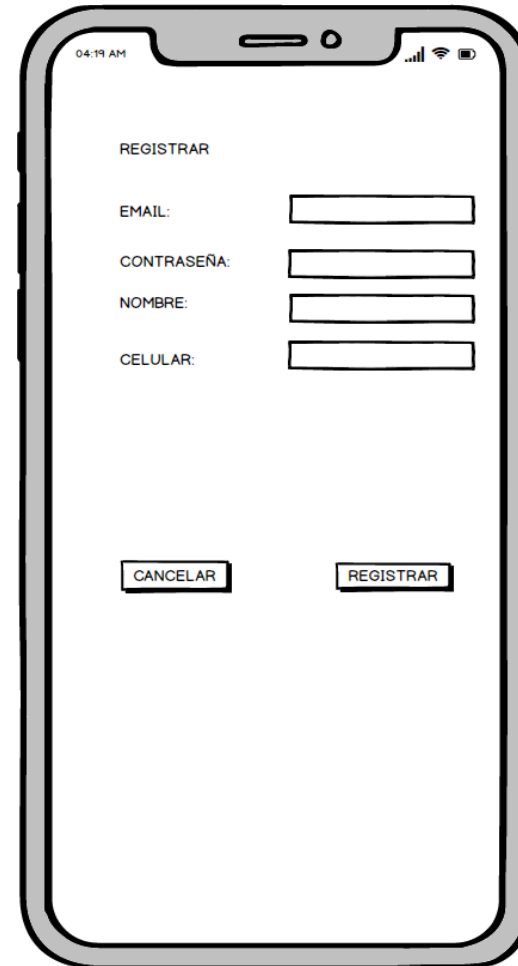


FIGURA 22: REGISTRO USUARIO
(FUENTE: PROPIA)

PROTOTIPO INTERFAZ PLANTAS



FIGURA 24: INTERFAZ PLANTA
(FUENTE: PROPIA)

PROTOTIPO INTERFAZ PLANTA1

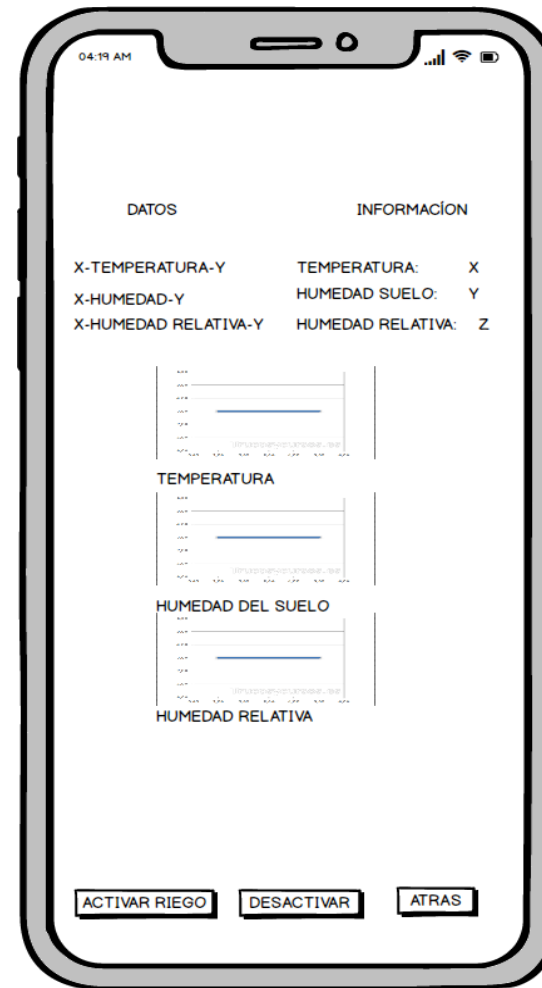


FIGURA 23: INTERFAZ GRÁFICA
(FUENTE: PROPIA)

PROTOTIPO INTERFAZ PLANTA 2

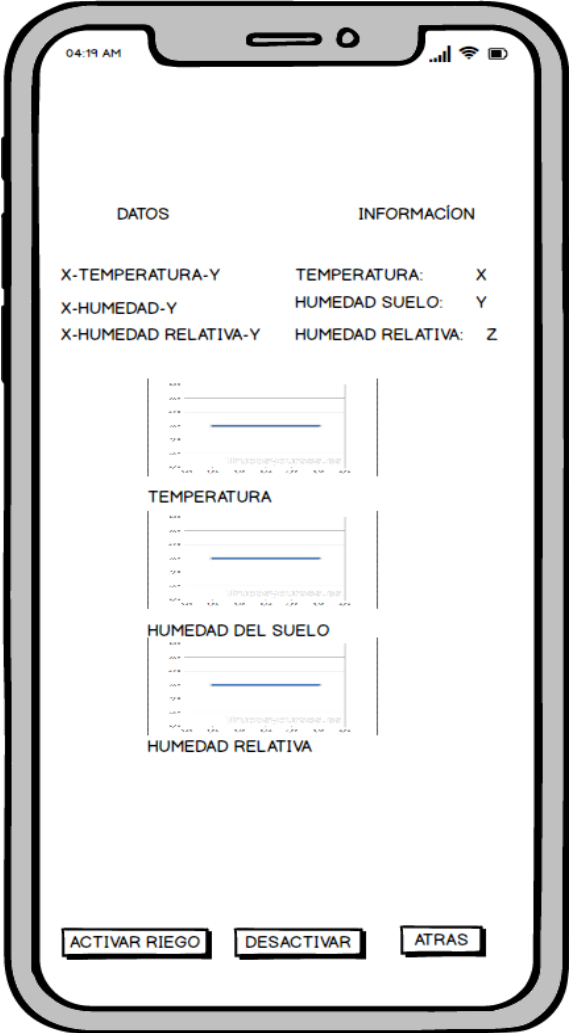


FIGURA 25: INTERFAZ PLANTA2
(FUENTE: PROPIA)

3.3.3.2 mapa de navegación de como se verá la aplicación móvil

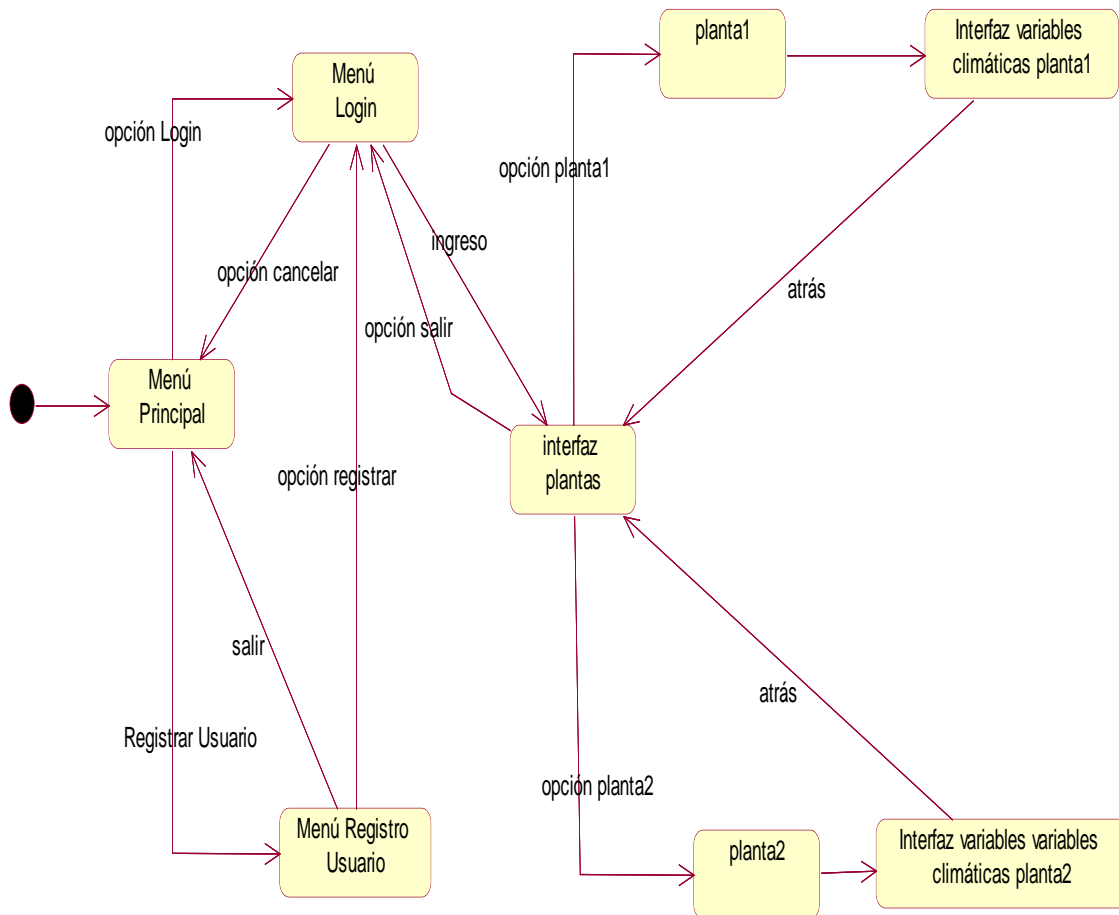


FIGURA 26: MAPA DE NAVEGACIÓN
(FUENTE: PROPIA)

3.3.4. Sprint Backlog.

Item	Tarea	N° Sprint	Días	Fecha Inicio	Fecha Final
1	<i>Login</i>	1	8	10/01/2019	18/01/2019
2	<i>Registro de Usuario</i>		9	19/01/2019	28/01/2019
3	<i>Visualización de variables climáticas</i>		15	29/01/2019	13/02/2019
4	<i>Visualización de gráficas</i>	2	7	14/03/2019	21/03/2019
5	<i>Control de Riego</i>		8	22/03/2019	30/03/2019

TABLA 6: SPRINT BACKLOG
(FUENTE: PROPIA)

3.3.5 TaskBoard Inicial.

El taskboard inicial presentara la estructura que tendrá el sprint backlog al inicio del proyecto, esta herramienta se realizara cuando se crea conveniente, pero en este caso se realizara cada al finalizar cada sprint para ver el avance del equipo de desarrollo.

Proyecto:	Aplicación móvil para encuesta			
Inicio	10/01/19	Fin	30/03/19	Semana: 0
N°	Tarea	Estado		
		Pendiente	En curso	Terminado
Sprint 1	Login	<input checked="" type="checkbox"/>		
	Registro Usuario	<input checked="" type="checkbox"/>		
Sprint 2	Visualización variables climáticas	<input checked="" type="checkbox"/>		
	Visualización de gráficas	<input checked="" type="checkbox"/>		
	Control de Riego	<input checked="" type="checkbox"/>		

TABLA 7: TASKBOARD INICIAL DEL PROYECTO.
(FUENTE: PROPIA)

3.3.6. Desarrollo y Sprint Review.

3.3.6.1. Sprint 1.

3.3.6.1.1. Desarrollo de Tareas.

En este primer sprint se concluirá con los módulos ya terminados, los cuales ya presentaran funcionalidad.

Se presentará las vistas de la aplicación de las tareas o asignadas a este sprint

Login.

Este módulo presentará el ingreso de los usuarios a la encuesta, tendrá que tener un usuario y contraseña para poder ingresar.

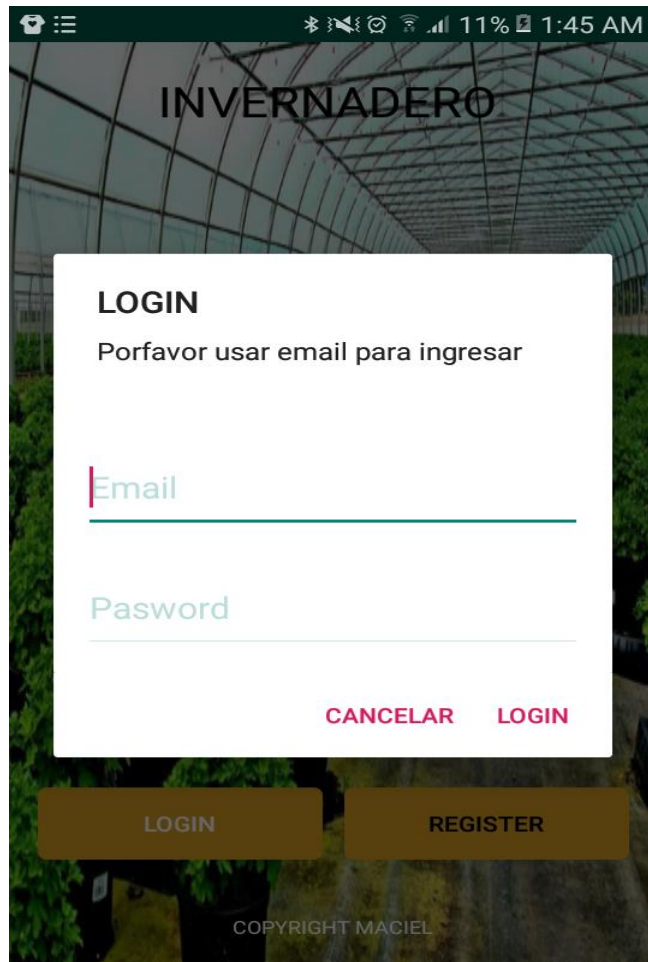


FIGURA 27: VISTA DE LOGIN
(FUENTE: PROPIA)

Registro Usuario

REGISTER
Porfavor usar email para registrarse

Email

Contraseña

Nombre

Celular

CANCELAR **REGISTER**

COPYRIGHT MACIEL

FIGURA 28: VISTA REGISTRO USUARIO
(FUENTE: PROPIA)

3.3.6.1.2. Taskboard

Al finalizar el sprint 1, se mostrará el taskboard en donde se observa el resultado de las tareas realizadas. En este caso de desarrollo satisfactoriamente las tareas asignadas con algunas observaciones.

Proyecto:	Aplicación móvil para encuesta			
Inicio	10/01/19	Fin	30/03/19	Semana: 0
N°	Tarea	Estado		
		Pendiente	En curso	Terminado
Sprint 1	Login			<input checked="" type="checkbox"/>
	Registro Usuario			<input checked="" type="checkbox"/>
Sprint 2	Visualización variables climáticas	<input checked="" type="checkbox"/>		
	Visualización de gráficas	<input checked="" type="checkbox"/>		
	Control de Riego	<input checked="" type="checkbox"/>		

TABLA 8: TASKBOARD
(FUENTE:PROPIA)

3.3.6.1.3. Sprint Review.

En esta reunión se entrega el primer incremento de software funcionando al cliente.

<i>Proyecto</i>	<i>Aplicación móvil para encuesta</i>		
<i>Lugar</i>	<i>INVERNADERO UNTELS</i>		
<i>Fecha</i>	<i>22/03/19</i>	<i>N° Sprint</i>	<i>1</i>
<i>Participantes:</i>	<i>David Licappa :Encargado del invernadero(Product owner)</i> <i>Teodoro Neri Diaz Leya(Scrum Master)</i> <i>Maciel Supo Samanez(Desarrollador)</i> <i>Usuario :David Licappa</i>		
	<i>ESTADO</i>		
<i>Item</i>	<i>Tarea</i>	<i>Aprobado</i>	<i>Observación</i>
<i>1</i>	<i>Login</i>	<input checked="" type="checkbox"/>	
<i>2</i>	<i>Registro de Usuario</i>	<input checked="" type="checkbox"/>	
	<i>OBSERVACIONES</i>		
<i>Aciertos</i>	<i>✓ La entrega del proyecto se realizo en la fecha establecida .</i>		
<i>Recomendaciones</i>	<i>+ Debe agregarse un sensor más que es humedad relativa</i>		

TABLA 9: PLANTILLA DE REVISIÓN DEL SPRINT1
(FUENTE: PROPIA)

3.3.6.2. Sprint 2.

3.3.6.2.1. Desarrollo de Tareas.

En este segundo sprint se presentará al cliente, los módulos ya terminados, los cuales ya presentaran funcionalidad

Se presentará las vistas de la aplicación de las tareas o asignadas a este sprint.

Vista

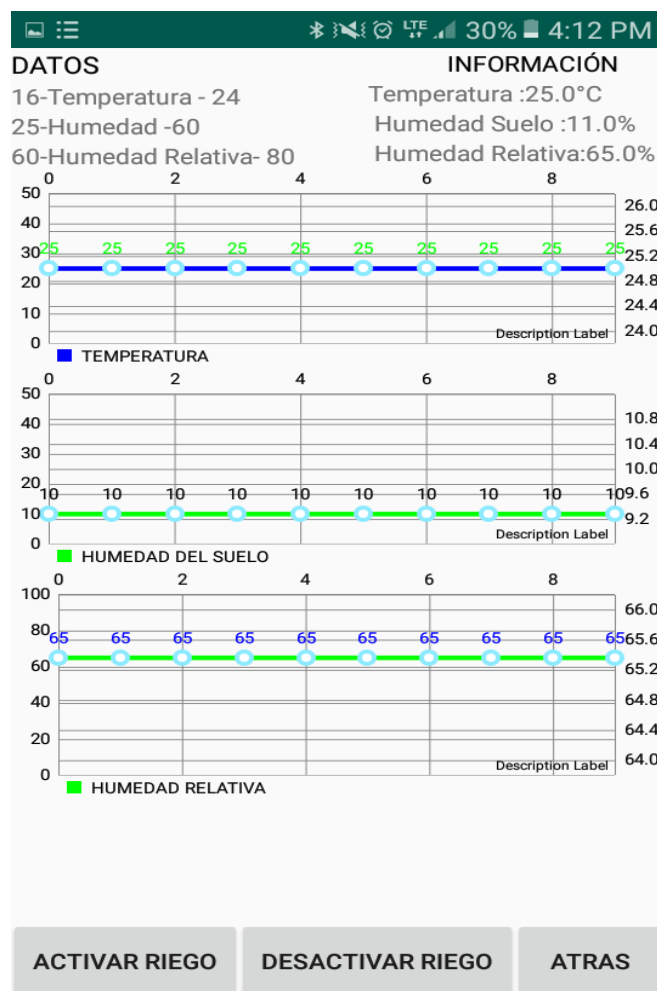


FIGURA 29: VISUALIZACIÓN DE LAS VARIABLES CLIMÁTICAS

(FUENTE: PROPIA)

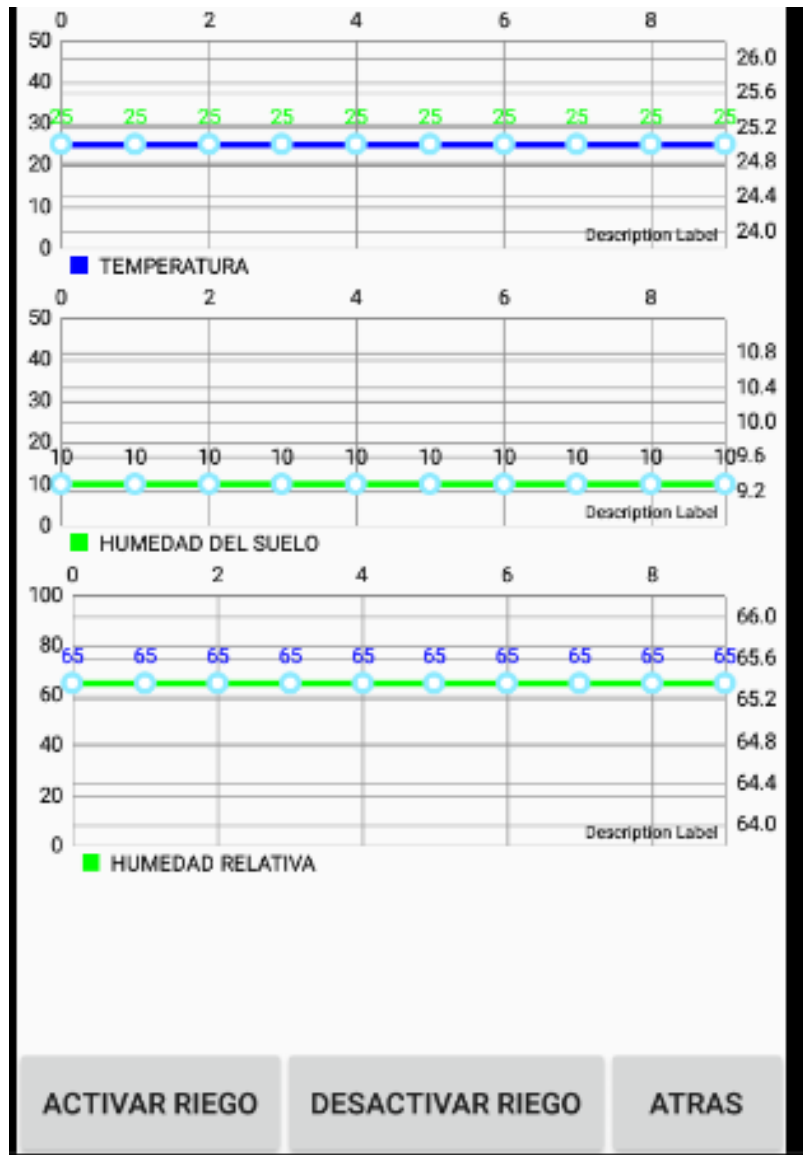


FIGURA 30: GRÁFICA DE LAS VARIABLES CLIMÁTICAS

(FUENTE: PROPIA)

3.3.6.2.2 Taskboard.

<i>Proyecto:</i>	<i>Aplicación móvil para encuesta</i>			
<i>Inicio</i>	<i>10/01/19</i>	<i>Fin</i>	<i>30/03/19</i>	<i>Semana: 0</i>
<i>N°</i>	<i>Tarea</i>	<i>Estado</i>		
		<i>Pendiente</i>	<i>En curso</i>	<i>Terminado</i>
<i>Sprint 1</i>	<i>Login</i>			<input checked="" type="checkbox"/>
	<i>Registro Usuario</i>			<input checked="" type="checkbox"/>
<i>Sprint 2</i>	<i>Visualización variables climáticas</i>			<input checked="" type="checkbox"/>
	<i>Visualización de gráficas</i>			<input checked="" type="checkbox"/>
	<i>Control de Riego</i>			<input checked="" type="checkbox"/>

TABLA 10 : TASKBOARD DE SPRINT2
(FUENTE : PROPIA)

3.3.6.2.3 Sprint Review

<i>Proyecto</i>	<i>Aplicación móvil para controlar las variables climáticas en el invernadero de la UNTELS</i>		
<i>Lugar</i>	<i>INVERNADERO DE LA UNTELS</i>		
<i>Fecha</i>	<i>30/03/19</i>	<i>N° Sprint</i>	<i>2</i>
<i>Participantes:</i>	<i>David Licappa (Product Owner)</i> <i>Teodoro Neri Diaz Leyva(Scrum Master)</i> <i>Maciel Deni Supo Samanez.(Desarrollador)</i> <i>Usuario(David Licappa)</i>		
ESTADO			
<i>Item</i>	<i>Tarea</i>	<i>Aprobado</i>	<i>Observación</i>
<i>3</i>	<i>Visualización de las variables climáticas</i>	<input checked="" type="checkbox"/>	
<i>4</i>	<i>Visualización de gráficas</i>	<input checked="" type="checkbox"/>	
<i>5</i>	<i>Control de Riego</i>	<input checked="" type="checkbox"/>	
OBSERVACIONES			
<i>Aciertos</i>	✓ <i>La entrega de módulos se realizó en la fecha establecida</i>		
<i>Errores</i>	✗ <i>El equipo logro entregar en la hora que no se acordó con el usuario.</i>		
<i>Recomendaciones</i>	+ <i>Se recomienda venir 1 día más para ir viendo las pruebas.</i>		

TABLA 11: SPRINT REVIEW DEL SPRINT2
(FUENTE : PROPIA)

3.4. RECURSOS.

El presente proyecto necesito de recursos para ser realizado, a continuación se detalla los recursos.

3.4.1. Recursos Humanos.

Pago de personal por mes

Cargo	Contrato	Monto
Programador		0

TABLA 12: RECURSOS HUMANOS
(FUENTE: PROPIA)

3.4.2 Recursos software

Tabla 4.RECURSOS DE SOFTWARE

Software	Versión	Licencia
Android Studio	6.0.1	Libre
Fritzing.	3.0.1	Libre
Firebase		Libre

TABLA 13: RECURSOS SOFTWARE
(FUENTE : PROPIA)

3.4.3. Recursos de Hardware.

Hardware para construcción de APK

ELEMENTO	CANTIDAD	PRECIO	TOTAL
Arduino Uno	1	S/28	S/28
Bomba de agua	2	S/15	S/30
Modulo Releí	2	S/5	S/10
Pantalla LCD	1	S/10	S/10
Fuente de Alimentación	1	S/20	S/20
DHT11	1	S/11	S/11
DHT22	1	S/18	S/18
potenciómetro	1	S/1	S/1
Sensor de humedad	2	S/8	S/16
Cable macho	40	S/0.10	S/5
cable	45	S/0.20	S/9
NodeMCU	1	S/30	S/30
condensador	1	S/1	S/1
protoboard	1	S/15	S/15
			S/204

TABLA 14: RECURSOS HADWARE
(FUENTE: PROPIA)

3.5 DESARROLLO DEL SOFTWARE EN ANDROID

1. CONEXIÓN APLICACIÓN MOVIL CON FIREBASE

El archivo **Google-services.json** contiene datos de configuración, como identificadores y claves, de los servicios de firebase la cual permite la conexión de la aplicación móvil con la plataforma firebase.

```
{
  "project_info": {
    "project_number": "790581908715",
    "firebase_url": "https://arduino-4485f.firebaseio.com",
    "project_id": "arduino-4485f",
    "storage_bucket": "arduino-4485f.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:790581908715:android:199f6970c7dbcb53",
        "android_client_info": {
          "package_name": "com.example.pc.biohuertoapp"
        }
      },
      "oauth_client": [
        {
          "client_id": "790581908715-
c7521giss0d8ek30e2f6bmb1rhuiln9s.apps.googleusercontent.com",
          "client_type": 1,
          "android_info": {
            "package_name": "com.example.pc.biohuertoapp",
            "certificate_hash":
"d0c51c87886be768043bfc22001717aca49f5cb0"
          }
        },
        {
          "client_id": "790581908715-
n39iisvmkp7n9edceqr8idhvjcvtlqoom.apps.googleusercontent.com",
          "client_type": 3
        }
      ],
      "api_key": [
        {
          "current_key": "AIzaSyB-5C_qbxmm0kRqSvTqNzeX5Kig5eoLmyo"
        }
      ],
      "services": {
        "analytics_service": {
          "status": 1
        },
        "appinvite_service": {
          "status": 2,
          "other_platform_oauth_client": [
            {
              "client_id": "790581908715-
n39iisvmkp7n9edceqr8idhvjcvtlqoom.apps.googleusercontent.com",
              "client_type": 3
            }
          ]
        },
        "ads_service": {
```

```

        "status": 2
    }
}
},
"configuration_version": "1"
}

```

2CREACIÓN DE VISTAS DE LA APLICACIÓN.

Se creará las interfaces usando archivos .XML

3.1 Interfaz de Logeo y Registro.

El archivo *activity_main.xml* presentará la interfaz que nos permitirá logearnos para poder acceder al menú de la aplicación y también registrar un usuario nuevo para poder logearnos.

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:gravity="center_horizontal"
    android:layout_marginTop="20dp">

    <TextView
        android:text="INVERNADERO "
        android:textSize="28sp"
        android:textAlignment="center"
        android:textColor="@android:color/black"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>

<TextView
    android:id="@+id/txt_w"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="20dp"
    android:layout_marginTop="20dp"
    android:text="COPYRIGHT MACIEL"
    android:textSize="12sp"
    android:textAlignment="center"
    android:textColor="@android:color/white"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<LinearLayout
    android:orientation="horizontal"
    android:weightSum="2"
    android:layout_above="@+id/txt_w"
    android:layout_margin="16dp"
    android:layout_centerHorizontal="true"

```

```

android:layout_width="match_parent"
android:layout_height="wrap_content">

<Button
    android:id="@+id/btn_sign"
    android:background="@drawable/btn_login_background"
    android:text="LOGIN"
    android:textColor="@android:color/white"
    android:layout_marginRight="6dp"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content" />

<Button
    android:id="@+id/btn_register"
    android:background="@drawable/btn_register_background"
    android:text="REGISTER"
    android:textColor="@android:color/black"
    android:layout_marginLeft="6dp"
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content" />

</LinearLayout>

```

3.1.1 Interfaz de login.

El archivo `layout_login.xml` presentará la interfaz es flotante en la cual se ingresará el usuario y la contraseña para poder logearse.

```

<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:cardElevation="10dp">

<LinearLayout
    android:orientation="horizontal"
    android:layout_margin="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

<com.rengwuxian.materialedittext.MaterialEditText
    android:id="@+id/getEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:text=""
    android:textColor="@color/colorPrimary"
    android:textColorHint="@color/colorPrimary"
    android:textSize="20sp"

    app:met_baseColor="@color/colorPrimary"
    app:met_floatingLabel="highlight"
    app:met_primaryColor="@color/colorPrimary"
    app:met_singleLineEllipsis="true" />

```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:orientation="horizontal"
    android:layout_marginTop="100dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
    <com.rengwuxian.materialEditText.MaterialEditText
```

```
        android:id="@+id/getpswd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Pasword"
        android:inputType="textEmailAddress"
        android:text=""
        android:textColor="@color/colorPrimary"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"
```

```
        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"
        app:met_singleLineEllipsis="true" />
```

```
</LinearLayout>
```

```
</android.support.v7.widget.CardView>
```

3.1.2 Interfaz de Registro.

El archivo `layout_register.xml` presentará la interfaz es flotante en la cual se ingresará el usuario, la contraseña, nombre, celular para poder registrar un usuario.

```
<android.support.v7.widget.CardView
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
    <LinearLayout
```

```
        android:orientation="vertical"
        android:layout_margin="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```
        <com.rengwuxian.materialEditText.MaterialEditText
```

```
            android:id="@+id/edtEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email"
            android:inputType="textEmailAddress"
            android:text=""
            android:textColor="@color/colorPrimary"
            android:textColorHint="@color/colorPrimary"
```

```

        android:textSize="20sp"

        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"
        app:met_singleLineEllipsis="true" /> </LinearLayout>
<LinearLayout
    android:layout_marginTop="110dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.rengwuxian.materialEditText.MaterialEditText
        android:id="@+id/edtpswd"
        android:hint="@string/contras"
        android:inputType="textPassword"
        android:text=""
        android:textColor="@color/colorPrimary"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        app:met_singleLineEllipsis="true"
        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"/>

</LinearLayout>

<LinearLayout
    android:layout_marginTop="200dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.rengwuxian.materialEditText.MaterialEditText
        android:id="@+id/edtname"
        android:hint="Nombre"
        android:inputType="text"
        android:text=""
        android:textColor="@color/colorPrimary"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        app:met_singleLineEllipsis="true"
        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"/>

</LinearLayout>

<LinearLayout
    android:layout_marginTop="300dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content">
        <com.rengwuxian.materialEditText.MaterialEditText
            android:id="@+id/edtphone"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Celular"
            android:inputType="phone"
            android:text=""
            android:textColor="@color/colorPrimary"
            android:textColorHint="@color/colorPrimary"
            android:textSize="20sp"

            app:met_baseColor="@color/colorPrimary"
            app:met_floatingLabel="highlight"
            app:met_primaryColor="@color/colorPrimary"
            app:met_singleLineEllipsis="true" />

    </LinearLayout>

</android.support.v7.widget.CardView>

```

1.1. Interfaz principal de plantas.

El archivo **activity_main2.xml** presentará la interfaz que nos permitirá ingresar a visualizar los datos de los sensores de la planta 1 y planta 2.

```

<LinearLayout
    android:id="@+id/linear111"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="INVERNADERO "
        android:textSize="34dp"
        android:textColor="@android:color/black" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" UNTELS"
        android:textSize="34dp"
        android:textColor="@android:color/black" />

</LinearLayout>

<LinearLayout
    android:id="@+id/linear222"
    android:layout_marginTop="150dp"
    android:layout_below="@+id/linear111"
    android:layout_width="match_parent"
    android:orientation="horizontal"
    android:layout_height="wrap_content">
    <Button
        android:id="@+id/planta1"

```

```

        android:text="PLANTA 1"
        android:textSize="30dp"
        android:layout_marginRight="30dp"
        android:layout_width="wrap_content"
        android:layout_height="match_parent" />
    <Button
        android:id="@+id/planta2"
        android:text="PLANTA 2"
        android:textSize="30dp"
        android:layout_width="wrap_content"
        android:layout_height="match_parent" />
</LinearLayout>
<LinearLayout
    android:id="@+id/lienar333"
    android:layout_below="@id/linear222"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <ImageView
        android:background="@drawable/imagen1"
        android:layout_width="170dp"
        android:layout_height="200dp" />
    <ImageView
        android:layout_marginLeft="30dp"

        android:background="@drawable/imagen2"
        android:layout_width="180dp"
        android:layout_height="200dp" />
</LinearLayout>

    <Button
        android:id="@+id/salir"
        android:text="SALIR"
        android:layout_alignParentBottom="true"
        android:gravity="center_horizontal"
        android:layout_centerHorizontal="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>

```


1.1. Interfaz de planta 1.

El archivo **activity_planta1.xml** presentará la interfaz que nos permitirá visualizar los datos del sensor de la planta1 también nos permitirá activar y desactivar el riego.

```
<LinearLayout
    android:id="@+id/linearuno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:layout_width="49dp"
        android:layout_height="wrap_content"
        android:textColor="@android:color/black"
        android:textSize="15sp"
        android:text="DATOS" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toEndOf="@+id/linearuno"
        android:text="16-Temperatura - 24" />
    <TextView
        android:text="25-Humedad -60"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:text="60-Humedad Relativa- 80"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
<LinearLayout
    android:layout_alignParentRight="true"
    android:layout_toRightOf="@id/linearuno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:text="
                                INFORMACIÓN"
        android:textColor="@android:color/black"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:text="
                                Temperatura :"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:id="@+id/lbltemp"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:orientation="horizontal">
    <TextView
        android:text="                Humedad Suelo : "
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/lblhum"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:weightSum="2"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="                Humedad Relativa: " />
    <TextView
        android:id="@+id/lbl3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
</LinearLayout>
</LinearLayout>
<LinearLayout

    android:id="@+id/lineardos"
    android:layout_below="@+id/linearuno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart"
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:layout_below="@+id/li2" />
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart2"
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:layout_below="@+id/li2" />
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart3"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:layout_below="@+id/li2" />
</LinearLayout>

```

```

<LinearLayout
    android:layout_alignParentBottom="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:gravity="center_horizontal"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btnriego"
        android:text="ACTIVAR RIEGO"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnnoriego"
        android:text="DESACTIVAR RIEGO"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnatras"
        android:text="ATRAS"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

</RelativeLayout>

```

1.1. Interfaz de planta 2.

El archivo *activity_planta2.xml* presentará la interfaz que nos permitirá visualizar los datos del sensor de la planta2 también nos permitirá activar y desactivar el riego.

```

<LinearLayout
    android:id="@+id/linearuno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:layout_width="49dp"
        android:layout_height="wrap_content"
        android:textColor="@android:color/black"
        android:textSize="15sp"
        android:text="DATOS" />
    <TextView
        android:text="18-Temperatura - 26"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:text="40 - Humedad - 70"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:text="60 - Humedad Relativa - 80"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
<LinearLayout
    android:layout_alignParentRight="true"
    android:layout_toRightOf="@+id/linearuno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:text="                INFORMACIÓN"
        android:textColor="@android:color/black"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:text="                Temperatura : "
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
            <TextView
                android:id="@+id/lbltemp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        </LinearLayout>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:text="                Humedad Suelo : "
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
            <TextView
                android:id="@+id/lblhum"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        </LinearLayout>
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:weightSum="2"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:text="                Humedad Relativa: " />
        <TextView
            android:id="@+id/lblhumerelativa"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>

```

```

        </LinearLayout>
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:id="@+id/lineardos"
    android:layout_below="@+id/linearuno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart11"
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:layout_below="@+id/li2" />
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart22"
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:layout_below="@+id/li2" />
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart33"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:layout_below="@+id/li2" />
</LinearLayout>

<LinearLayout
    android:layout_alignParentBottom="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:gravity="center_horizontal"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btnriego2"
        android:text="ACTIVAR RIEGO"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnnoriego2"
        android:text="DESACTIVAR RIEGO"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnatras2"
        android:text="ATRAS"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

</RelativeLayout>

```

1. CREACIÓN DE ACTIVIDADES.

1.1. Actividad de logeo y Registro.

En el archivo **MainActivity.java** se realizarán las acciones que se realizan en las interfaces.

Se importará las librerías de Firebase, la cuales nos brindaran las variables de conexión con la base de datos **Realtime** de **Firestore**.

```
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

En la clase **MainActivity.java** se declara las variables de botones y firebase.

```
Button btnlogin, btnregister;
FirebaseAuth auth;
FirebaseDatabase db;
DatabaseReference users;
```

El método reservado **OnCreate()** nos permitirá llamar a la interfaz **activity_main.xml**, enlazar las variables de los botones antes declarados con los id's de la **activity_main.xml** y usar los métodos predeterminados de Firebase.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btnlogin = findViewById(R.id.btn_sign);
    btnregister = findViewById(R.id.btn_register);
    auth = FirebaseAuth.getInstance();
    db = FirebaseDatabase.getInstance();
    users = db.getReference("Users");

    btnregister.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            registerdialog();
        }
    });

    btnlogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            logindialog();
        }
    });
}
```

```

    });
}
}

```

Se crearán los métodos **registerdialog()** y **logindialog()** los cuales interacción con los botones login y registro.

```

private void registerdialog() {

    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("REGISTER");
    dialog.setMessage("Porfavor usar email para registrarse");

    LayoutInflater inflater = LayoutInflater.from(this);
    View register_layout =
inflater.inflate(R.layout.layout_register, null);

    final MaterialEditText edtEmail =
register_layout.findViewById(R.id.edtEmail);
    final MaterialEditText edtpswd =
register_layout.findViewById(R.id.edtpswd);
    final MaterialEditText edtnamee =
register_layout.findViewById(R.id.edtname);
    final MaterialEditText edtphone =
register_layout.findViewById(R.id.edtphone);

    dialog.setView(register_layout);

    dialog.setPositiveButton("REGISTER", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();

            //check validate
            if(TextUtils.isEmpty(edtEmail.getText().toString())){
                Snackbar.make(rootLayout, "porfavor ingresar un email
valido", Snackbar.LENGTH_SHORT).show();
                return;
            }
            if(TextUtils.isEmpty(edtphone.getText().toString())){
                Snackbar.make(rootLayout, "porfavor ingresar una
contraseña valida", Snackbar.LENGTH_SHORT).show();
                return;
            }
            if(TextUtils.isEmpty(edtpswd.getText().toString())){
                Snackbar.make(rootLayout, "porfavor ingresar una
contraseña valida", Snackbar.LENGTH_SHORT).show();
                return;
            }
            if(edtpswd.getText().toString().length() < 6){

```

```

        Snackbar.make(rootLayout, "contraseña muy
corta", Snackbar.LENGTH_SHORT).show();
        return;
    }

    if (TextUtils.isEmpty(edtnamee.getText().toString())) {
        Snackbar.make(rootLayout, "porfavor ingresar una nombre
valido", Snackbar.LENGTH_SHORT).show();
        return;
    }

    auth.createUserWithEmailAndPassword(edtEmail.getText().toString(), edtpswd
.getText().toString()).addOnSuccessListener(new
OnSuccessListener<AuthResult>() {
        @Override
        public void onSuccess(AuthResult authResult) {

            Users user = new Users();
            user.setEmail(edtEmail.getText().toString());
            user.setPassword(edtpswd.getText().toString());
            user.setPhone(edtphone.getText().toString());
            user.setName(edtnamee.getText().toString());
            users.child(FirebaseAuth.getInstance()

.getCurrentUser().getUid()).setValue(user).addOnSuccessListener(new
OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    //Snackbar.make(rootLayout, "Register
successful", Snackbar.LENGTH_SHORT).show();
                    showSnackBar(MainActivity.this, "Registro
satisfactorio");
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    //Snackbar.make(rootLayout, "Failed"+e.getMessage(), Snackbar.LENGTH_SHORT)
                    .show();
                    showSnackBar(MainActivity.this, "Fallo!!");
                }
            });
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {

            //Snackbar.make(rootLayout, "Failed"+e.getMessage(), Snackbar.LENGTH_SHORT)
            .show();
            showSnackBar(MainActivity.this, "Fallo!!");
        }
    });
}

dialog.setNegativeButton("CANCELAR", new
DialogInterface.OnClickListener() {

```



```

        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });

    dialog.show();
}

```

El método login dialog ()

```

private void logindialog() {
    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("LOGIN");
    dialog.setMessage("Porfavor usar email para ingresar");

    LayoutInflater inflater = LayoutInflater.from(this);
    View login_layout = inflater.inflate(R.layout.lasyout_login,null);

    final MaterialEditText edtEmail =
login_layout.findViewById(R.id.getEmail);
    final MaterialEditText edtpswd =
login_layout.findViewById(R.id.getpswd);

    dialog.setView(login_layout);

    dialog.setPositiveButton("LOGIN", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();

            //waitingDialog = new SpotsDialog(MainActivity.this.,
"write_your_string_here. for ex: Loading..");
            //waitingDialog.show();

            btnlogin.setEnabled(false);

            //check validate
            if(TextUtils.isEmpty(edtEmail.getText().toString())){
                //Snackbar.make(rootLayout,"porfavor ingresar un email
",Snackbar.LENGTH_SHORT).show();
                showSnackBar(MainActivity.this,"porfavor ingresar un
email");
                return;
            }

            if(TextUtils.isEmpty(edtpswd.getText().toString())){
                //Snackbar.make(rootLayout,"porfavor ingresar una
contraseña ",Snackbar.LENGTH_SHORT).show();
                showSnackBar(MainActivity.this,"porfavor ingresar una
contraseña ");
                return;
            }

            if(edtpswd.getText().toString().length() < 6){
                //Snackbar.make(rootLayout,"contraseña muy

```

```

corta", Snackbar.LENGTH_SHORT).show();
        showSnackBar(MainActivity.this, "contraseña muy corta");
        return;
    }

    //login

    auth.signInWithEmailAndPassword(edtEmail.getText().toString(), edtPswd.getText().toString()).addOnSuccessListener(new
    OnSuccessListener<AuthResult>() {
        @Override
        public void onSuccess(AuthResult authResult) {
            startActivity(new
            Intent(MainActivity.this, Main2Activity.class));
            finish();
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {

            //Snackbar.make(rootLayout, "Failed", Snackbar.LENGTH_SHORT).show();
            showSnackBar(MainActivity.this, "Usuario no
            registrado!");

            btnlogin.setEnabled(true);
        }
    });
}

dialog.setNegativeButton("CANCELAR", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
    }
});

dialog.show();
}

```

Se crearán los eventos que van enlazados a los botones cuando estos se presionan en la aplicación. Estos eventos llaman a los métodos antes declarados.

```

btnregister.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        registerdialog();
    }
});

btnlogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        logindialog();
    }
});

```

```
    }  
});
```

1.1. Actividad llamada a menú principal.

En el archivo **Main2Activity.java** se realizarán las acciones que se realizan en las interfaces.

En la clase MainActivity.java se declara las variables de botones y firebase.

```
DatabaseReference myRef;  
ValueEventListener myvalueListener;  
Button planta1 , planta2 , salir;
```

El método reservado **OnCreate ()** nos permitirá llamar a la interfaz **activity_main2.xml** ,enlazar las variables de los botones antes declarados con los id's de la **activity_main.xml**

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main2);  
  
    planta1 = findViewById(R.id.planta1);  
    planta2 = findViewById(R.id.planta2);  
    salir = findViewById(R.id.salir);
```

Se crearán los eventos que van enlazados a los botones cuando estos se presionan en la aplicación. Estos eventos llaman a las actividades PLANTA1, PLANTA 2 Y MainActivity

```
planta1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        startActivity(new Intent(Main2Activity.this, PLANTA1.class));
```

```

    }
});

planta2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(Main2Activity.this, PLANTA2.class));
    }
});

salir.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(Main2Activity.this, MainActivity.class));
    }
});

```

1.1. Actividad de Visualización de Datos del Sensor de Planta 1.

En el archivo **PLANTA1.java** se realizarán las acciones que se realizan en las interfaces.

Se importará las librerías de Firebase, la cuales nos brindaran las variables de conexión con la base de datos **Realtime** de **Firebase**. También las librerías de los gráficos de visualización.

```

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.interfaces.datasets.ILineDataSet;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

```

En la clase **PLANTA1.java** se declara las variables de botones, gráficos y firebase.

```

public class PLANTA1 extends AppCompatActivity {

    private static final String TAGLOG = "dbmaciel" ;
    DatabaseReference myRef;
    DatabaseReference myref2;
    ValueEventListener myvalueListener;
    TextView lbltemperatura, lblhumedad , lblhumedadrelativa;
    LineChart mChart ,mChart2, mChart3;
    Button btnriego, btnnoriego, btnatras;
    float x,hum ,z;

```

El método reservado **OnCreate()** nos permitirá llamar a la interfaz **activity_planta1.xml**, enlazar las variables de los botones antes declarados con los id's de la **activity_planta1.xml**.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_planta1);

    lbltemperatura = findViewById(R.id.lbltemp);
    lblhumedad = findViewById(R.id.lblhum);
    lblhumedadrelativa = findViewById(R.id.lbl3);
    btnnoriego = findViewById(R.id.btnnoriego);
    btnriego = findViewById(R.id.btnriego);
    btnatras = findViewById(R.id.btnatras);

    mChart = findViewById(R.id.linechart);
    mChart2 = findViewById(R.id.linechart2);
    mChart3 = findViewById(R.id.linechart3);
}
```

Se crearán los métodos **plot()**, **plot2()** y **plot3** los permitirán visualizar los datos de la base de datos **realtime** de **firebase**

```
public void plot() {
    mChart.setDragEnabled(true);
    mChart.setScaleEnabled(false);

    YAxis leftAxis = mChart.getAxisLeft();
    leftAxis.removeAllLimitLines();
    leftAxis.setAxisMaximum(50f);
    leftAxis.setAxisMinimum(0f);

    ArrayList<Entry> yValues = new ArrayList<>();

    for(int i= 0;10>i;i++){
        if(x != 0){
            yValues.add(new Entry(i,Float.valueOf(x)));}
    }

    LineDataSet set1 = new LineDataSet(yValues, "TEMPERATURA" );
    set1.setFillAlpha(110);
    set1.setColor(Color.BLUE);
    set1.setLineWidth(3f);
    set1.setValueTextSize(10f);
    set1.setValueTextColor(Color.GREEN);

    ArrayList<ILineDataSet> dataSets = new ArrayList<>();
    dataSets.add(set1);

    LineData data = new LineData(dataSets);
}
```

```

    mChart.setData(data);
    mChart.invalidate();

    // enable scaling and dragging
    /*mChart.setDragEnabled(true);
    mChart.setScaleEnabled(true);
    mChart.setDrawGridBackground(false);
    mChart.setHighlightPerDragEnabled(true);
    mChart.setBackgroundColor(Color.WHITE);

    String[] valuesX = new String[] {"Tiempo (ms)"};
    String[] valuesY = new String[] {"MiliVolts (mV)"};*/
}
public void plot2() {
    mChart2.setDragEnabled(true);
    mChart2.setScaleEnabled(false);

    YAxis leftAxis = mChart2.getAxisLeft();
    leftAxis.removeAllLimitLines();
    leftAxis.setAxisMaximum(50f);
    leftAxis.setAxisMinimum(0f);

    ArrayList<Entry> yValues2 = new ArrayList<>();

    for(int i= 0;10>i;i++){
        if(hum != 0){
            yValues2.add(new Entry(i,Float.valueOf(hum)));}
    }

    LineDataSet set1 = new LineDataSet(yValues2,"HUMEDAD DEL SUELO"
);
    set1.setFillAlpha(110);
    set1.setColor(Color.GREEN);
    set1.setLineWidth(3f);
    set1.setValueTextSize(10f);
    set1.setValueTextColor(Color.BLACK);

    ArrayList<ILineDataSet> dataSets2 = new ArrayList<>();
    dataSets2.add(set1);

    LineData data = new LineData(dataSets2);
    mChart2.setData(data);
    mChart2.invalidate();

    // enable scaling and dragging
    /*mChart.setDragEnabled(true);
    mChart.setScaleEnabled(true);
    mChart.setDrawGridBackground(false);
    mChart.setHighlightPerDragEnabled(true);
    mChart.setBackgroundColor(Color.WHITE);

    String[] valuesX = new String[] {"Tiempo (ms)"};
    String[] valuesY = new String[] {"MiliVolts (mV)"};*/
}
public void plot3() {
    mChart3.setDragEnabled(true);

```

```

mChart3.setScaleEnabled(false);

YAxis leftAxis = mChart3.getAxisLeft();
leftAxis.removeAllLimitLines();
leftAxis.setAxisMaximum(100f);
leftAxis.setAxisMinimum(0f);

ArrayList<Entry> yValues3 = new ArrayList<>();

for(int i= 0;10>i;i++){
    if(z != 0){
        yValues3.add(new Entry(i,Float.valueOf(z)));}
}

LineDataSet set3 = new LineDataSet(yValues3,"HUMEDAD RELATIVA" );
set3.setFillAlpha(60);
set3.setColor(Color.GREEN);
set3.setLineWidth(3f);
set3.setValueTextSize(10f);
set3.setValueTextColor(Color.BLUE);

ArrayList<ILineDataSet> dataSets3 = new ArrayList<>();
dataSets3.add(set3);

LineData data3 = new LineData(dataSets3);
mChart3.setData(data3);
mChart3.invalidate();

```

El método reservado **OnStart()** nos permitirá iniciar con los métodos de visualización de datos antes declarados.

```

protected void onStart() {
    super.onStart();
    plot();
    plot2();
    plot3();
}

```

Se crearán los eventos que van enlazados a los botones cuando estos se presionan en la aplicación. Estos eventos llaman a los métodos de firebase.

```

btnriego.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        myref2 =
        FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta
1");

        int x = 1;

        myref2.setValue(x);
    }
}

```

```

});

btnnoriego.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        myref2 =
        FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta
1");

        int x = 0;

        myref2.setValue(x);

    }
});

btnatras.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(PLANTA1.this, Main2Activity.class));
    }
});

```

con este método sacamos del firebase las variables temperatura, humedad del suelo y humedad relativa

```

myRef = FirebaseDatabase.getInstance().getReference().child("planta1");

myvalueListener = new ValueEventListener() {

    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

        biohuerto bio = dataSnapshot.getValue(biohuerto.class);
        x = bio.getTemperatura();
        hum = bio.getHumedad();
        z = bio.getHumedadrelativa();
        lbltemperatura.setText(x + "°C");
        lblhumedad.setText(hum + "%");
        lblhumedadrelativa.setText(z + "%");
        Log.e(TAGLOG, "ondatachange" +
dataSnapshot.getValue().toString());

    }
}

```


1.1. Actividad de Visualizacion de Datos del Sensor de la Planta 2.

En el archivo **PLANTA2.java** se realizarán las acciones que se realizan en las interfaces.

Se importará las librerías de Firebase, la cuales nos brindaran las variables de conexión con la base de datos **Realtime** de **Firestore**. También las librerías de los gráficos de visualización.

```
import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.interfaces.datasets.ILineDataSet;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
```

En la clase **PLANTA2.java** se declara las variables de botones, gráficos y firebase.

```
private static final String TAGLOG = "dbmaciel" ;
DatabaseReference myRef;
DatabaseReference myef2, my3;
ValueEventListener myvalueListener, myvalueListener2;
TextView lbltemperatura, lblhumedad , lblhumedadrelativa;
Button btnriego2, btnnoriego2, btnatras2;
LineChart mChart,mChart2,mChart3;
float tempe, hume, hure;
```

El método reservado **OnCreate()** nos permitirá llamar a la interfaz **activity_planta2.xml** ,enlazar las variables de los botones antes declarados con los id's de la **activity_planta2.xml** .

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_planta2);
    lbltemperatura = findViewById(R.id.lbltemp);
    lblhumedad = findViewById(R.id.lblhum);
    lblhumedadrelativa = findViewById(R.id.lblhumerrelativa);
    btnnoriego2 = findViewById(R.id.btnnoriego2);
    btnriego2 = findViewById(R.id.btnriego2);
    btnatras2 = findViewById(R.id.btnatras2);

    mChart = findViewById(R.id.linechart11);
    mChart2 = findViewById(R.id.linechart22);
    mChart3 = findViewById(R.id.linechart33);
}
```

Se crearán los métodos **plot()** , **plot2()** y **plot3** los permitirán visualizar los datos de la base de datos **realtime** de **firebase**

```
private void plot() {
    mChart.setDragEnabled(true);
    mChart.setScaleEnabled(false);

    YAxis leftAxis = mChart.getAxisLeft();
    leftAxis.removeAllLimitLines();
    leftAxis.setAxisMaximum(50f);
    leftAxis.setAxisMinimum(0f);

    ArrayList<Entry> yValues = new ArrayList<>();

    for(int i= 0;10>i;i++){
        if(tempe != 0){
            yValues.add(new Entry(i, tempe));}

    }

    LineDataSet set1 = new LineDataSet(yValues, "TEMPERATURA" );
    set1.setFillAlpha(110);
    set1.setColor(Color.BLUE);
    set1.setLineWidth(3f);
    set1.setValueTextSize(10f);
    set1.setValueTextColor(Color.GREEN);

    ArrayList<ILineDataSet> dataSets = new ArrayList<>();
    dataSets.add(set1);

    LineData data = new LineData(dataSets);
    mChart.setData(data);

    // enable scaling and dragging
    mChart.setDragEnabled(true);
    mChart.setScaleEnabled(true);
    mChart.setDrawGridBackground(false);
    mChart.setHighlightPerDragEnabled(true);
    mChart.setBackgroundColor(Color.WHITE);

    String[] valuesX = new String[] {"Tiempo (ms) "};
    String[] valuesY = new String[] {"MiliVolts (mV) "};
}
private void plot2() {
    mChart2.setDragEnabled(true);
    mChart2.setScaleEnabled(false);

    YAxis leftAxis = mChart.getAxisLeft();
    leftAxis.removeAllLimitLines();
    leftAxis.setAxisMaximum(50f);
    leftAxis.setAxisMinimum(0f);

    ArrayList<Entry> yValues = new ArrayList<>();

    for(int i= 0;10>i;i++){
```

```

        if(hume != 0){
            yValues.add(new Entry(i,hume));}
    }

    LineDataSet set1 = new LineDataSet(yValues,"HUMEDAD DEL SUELO" );
    set1.setFillAlpha(110);
    set1.setColor(Color.BLUE);
    set1.setLineWidth(3f);
    set1.setValueTextSize(10f);
    set1.setValueTextColor(Color.GREEN);

    ArrayList<ILineDataSet> dataSets = new ArrayList<>();
    dataSets.add(set1);

    LineData data = new LineData(dataSets);
    mChart2.setData(data);

    // enable scaling and dragging
    mChart2.setDragEnabled(true);
    mChart2.setScaleEnabled(true);
    mChart2.setDrawGridBackground(false);
    mChart2.setHighlightPerDragEnabled(true);
    mChart2.setBackgroundColor(Color.WHITE);

    String[] valuesX = new String[] {"Tiempo(ms)"};
    String[] valuesY = new String[] {"MiliVolts(mV)"};
}
private void plot3() {
    mChart3.setDragEnabled(true);
    mChart3.setScaleEnabled(false);

    YAxis leftAxis = mChart3.getAxisLeft();
    leftAxis.removeAllLimitLines();
    leftAxis.setAxisMaximum(100f);
    leftAxis.setAxisMinimum(0f);

    ArrayList<Entry> yValues = new ArrayList<>();

    for(int i= 0;10>i;i++){
        if(hure != 0){
            yValues.add(new Entry(i,hure));}
    }

    LineDataSet set1 = new LineDataSet(yValues,"HUMEDAD RELATIVA" );
    set1.setFillAlpha(110);
    set1.setColor(Color.BLUE);
    set1.setLineWidth(3f);
    set1.setValueTextSize(10f);
    set1.setValueTextColor(Color.GREEN);

    ArrayList<ILineDataSet> dataSets = new ArrayList<>();
    dataSets.add(set1);

    LineData data = new LineData(dataSets);
    mChart3.setData(data);
}

```

```

// enable scaling and dragging
mChart3.setDragEnabled(true);
mChart3.setScaleEnabled(true);
mChart3.setDrawGridBackground(false);
mChart3.setHighlightPerDragEnabled(true);
mChart3.setBackgroundColor(Color.WHITE);

String[] valuesX = new String[] {"Tiempo (ms)"};
String[] valuesY = new String[] {"MiliVolts (mV)"};
}
}

```

El método reservado **OnStart()** nos permitirá iniciar con los métodos de visualización de datos antes declarados.

```

protected void onStart() {
    super.onStart();
    plot();
    plot2();
    plot3();
}

```

Se crearán los eventos que van enlazados a los botones cuando estos se presionan en la aplicación. Estos eventos llaman a los métodos de firebase.

```

btnriego2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        myef2 =
        FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta
2");

        int x = 1;

        myef2.setValue(x);
    }
});

btnnoriego2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        myef2 =
        FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta
2");

        int x = 0;

        myef2.setValue(x);
    }
});

```

```

btnatras2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(PLANTA2.this,Main2Activity.class));
    }
});

```

Con este método sacamos del firebase las variables temperatura,humedad del suelo y humedad relativa

```

myRef = FirebaseDatabase.getInstance().getReference().child("planta2");
my3 = FirebaseDatabase.getInstance().getReference().child("dht11");
myvaluelistener = new ValueEventListener() {

    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        biohuerto bio = dataSnapshot.getValue(biohuerto.class);
        tempe = bio.getTemperatura2();
        hume = bio.getHumedad2();
        hure = bio.getHumedadRelativa2();
        lbltemperatura.setText(tempe+"°C");
        lblhumedad.setText(hume+"%");
        lblhumedadrelativa.setText(hure+"%");

        Log.e(TAGLOG , "ondatachange" +
dataSnapshot.getValue().toString());
    }

    @Override

    public void onCancelled(@NonNull DatabaseError databaseError) {
        Log.e(TAGLOG , "Error" + databaseError.toException());
    }
};

myRef.addValueEventListener(myvaluelistener);

```

3.6 RESULTADOS

El proyecto se realizó en 2 variedades de plantas: Geranio y Begonia

En este diagrama vemos cómo funciona el circuito, el arduino conectado a los sensores de humedad del aire, humedad del suelo y temperatura del ambiente

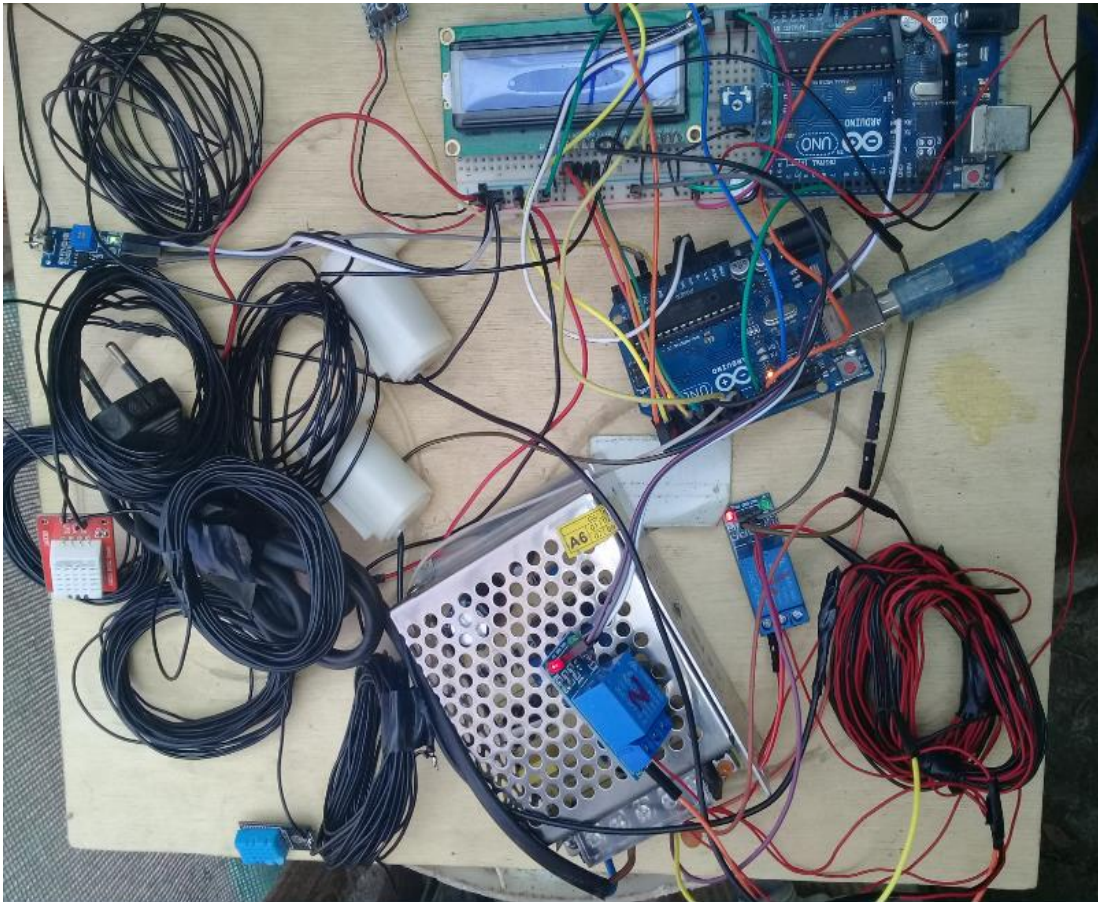


FIGURA 31: ESQUEMA DE MI CIRCUITO
(FUENTE: PROPIA)

EL arduino utilizado es el arduino uno que es mucho más comercial en el mercado. Dentro del proyecto propuesto, a través de la placa Arduino Uno se va a realizar el monitoreo de las variables climáticas y a través de la aplicación una vez realizado el monitoreo se va a activar el control de riego.

PANTALLA LED (LCD)

Dentro del Invernadero, en un lugar estratégico, se utilizará una pantalla de LED para mostrar la información en tiempo real de las variables climáticas. Esta

interface de salida es la única obligatoria dentro de la plataforma para la presentación de información, en esta pantalla LCD cuya dimensión es de 16 x 2. En esta pantalla Led vamos observar en dos filas la primera fila para la planta 1 y la segunda fila para la planta 2.



FIGURA 32: VISUALIZACIÓN DE LAS VARIABLES CLIMÁTICAS EN LA PANTALLA LCD

(FUENTE: PROPIA)

En esta medición vemos que la

- temperatura de la planta 1 es de 25 grados ,de la humedad del suelo 12% y de la humedad relativa del aire es de 65
- temperatura de la planta 2 es de 25 grados, de la humedad del suelo 77% y de la humedad relativa del aire es del 79%

MODULO RELAY (5 v)

El Relay es para controlar otros circuitos que operan a tensiones superiores a las del circuito. En el sistema que he desarrollado, el módulo de un relé será el encargado de encender o apagar la bomba de agua que es la encargada de permitir el paso del agua.

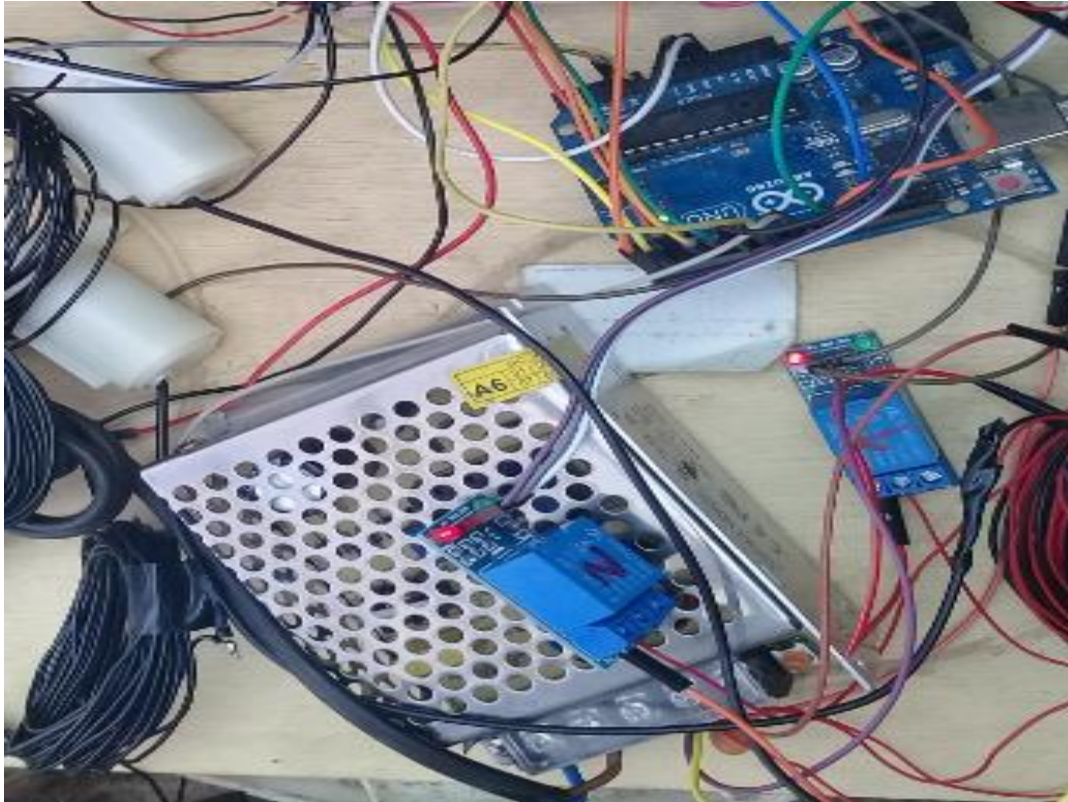


FIGURA 33: VISUALIZACIÓN DE MODULOS RELAY
(FUENTE: PROPIA)

En el proyecto ya que son dos plantas se utilizara dos módulos relay que iran conectados a las dos bombas que emitirán el riego tanto para la planta 1(Geranio) y el otro para la planta 2(Begonia)

BOMBA DE AGUA

Esta bomba de agua permitirá que el agua pase cuando la persona encargada de monitorear las variables climáticas tanto para la planta 1 como para la planta 2 mediante la aplicación mediante un botón de apagado y encendido del riego pueda activar o apagar el riego en las plantas.



FIGURA 34: BOMBAS DE AGUA CONECTADAS AL CIRCUITO
(FUENTE: PROPIA)

NodeMCU:

es una placa de desarrollo totalmente abierta, a nivel de software y de hardware. Al igual que ocurre con Arduino, en NodeMCU todo está dispuesto para facilitar la programación de un microcontrolador o MCU, es una placa o kits de desarrollo que llevan incorporados un chip que se suele llamar SoC (System on a Chip) que dentro tiene un microcontrolador o MCU.

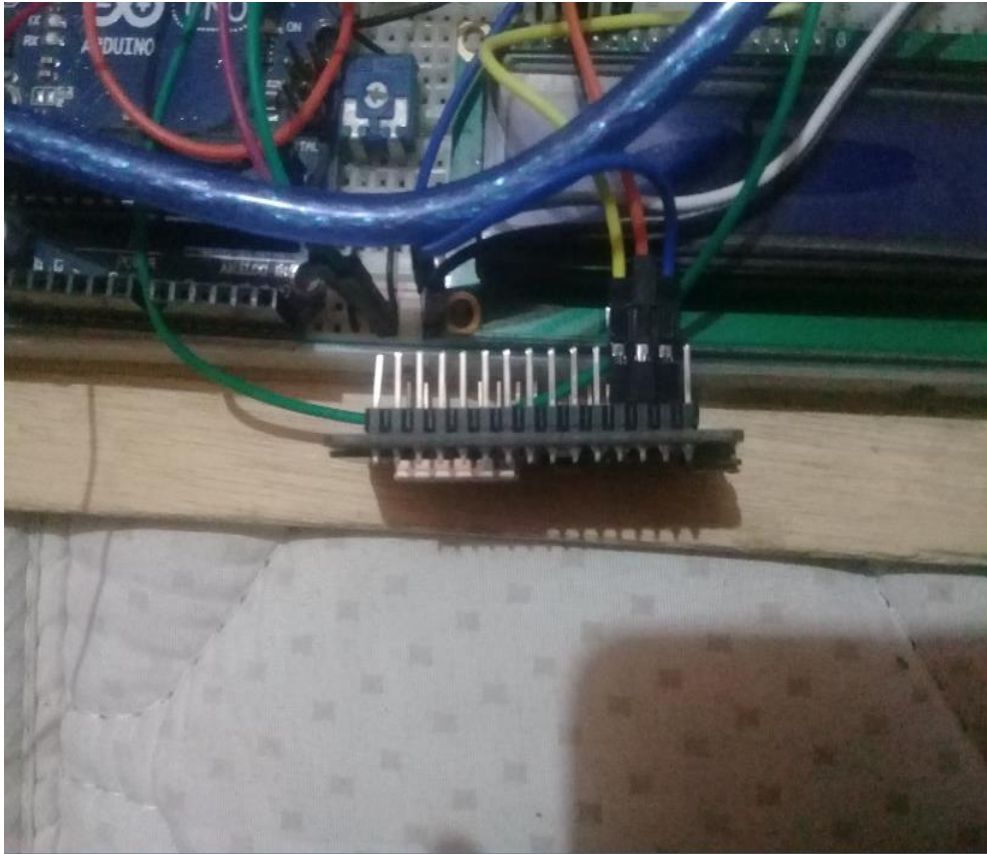


FIGURA 35: NODEMCU CONECTADO AL CIRCUITO
(FUENTE: PROPIA)

Para medir las variables climáticas de humedad del suelo, temperatura y humedad relativa de aire se eligió dos plantas que están dentro del invernadero de la UNTELS

BEGONIA

En la planta Begonia la temperatura adecuada varía entre 16°C a 24°C, necesitan un buen riego en verano y en época de floración, que la tierra nunca se seque pero que tampoco se encharque.

PASO 1 :

Colocamos el sensor de humedad en la tierra lo introducimos hasta el fondo y el sensor DHT que mide la temperatura y la humedad del aire



**FIGURA 36: PLANTA BEGONIA
(FUENTE: PROPIA)**



**FIGURA 37: SENSOR DE HUMEDAD Y TEMPERATURA CONECTADA A LA
PLANTA BEGONIA
(FUENTE: PROPIA)**

Paso 3:

Verificamos que en la pantalla LCD vemos los valores en tiempo real de la planta begonia:

- Humedad relativa del aire es de 65%
- Temperatura del aire es de 26
- Humedad del suelo es de 12%

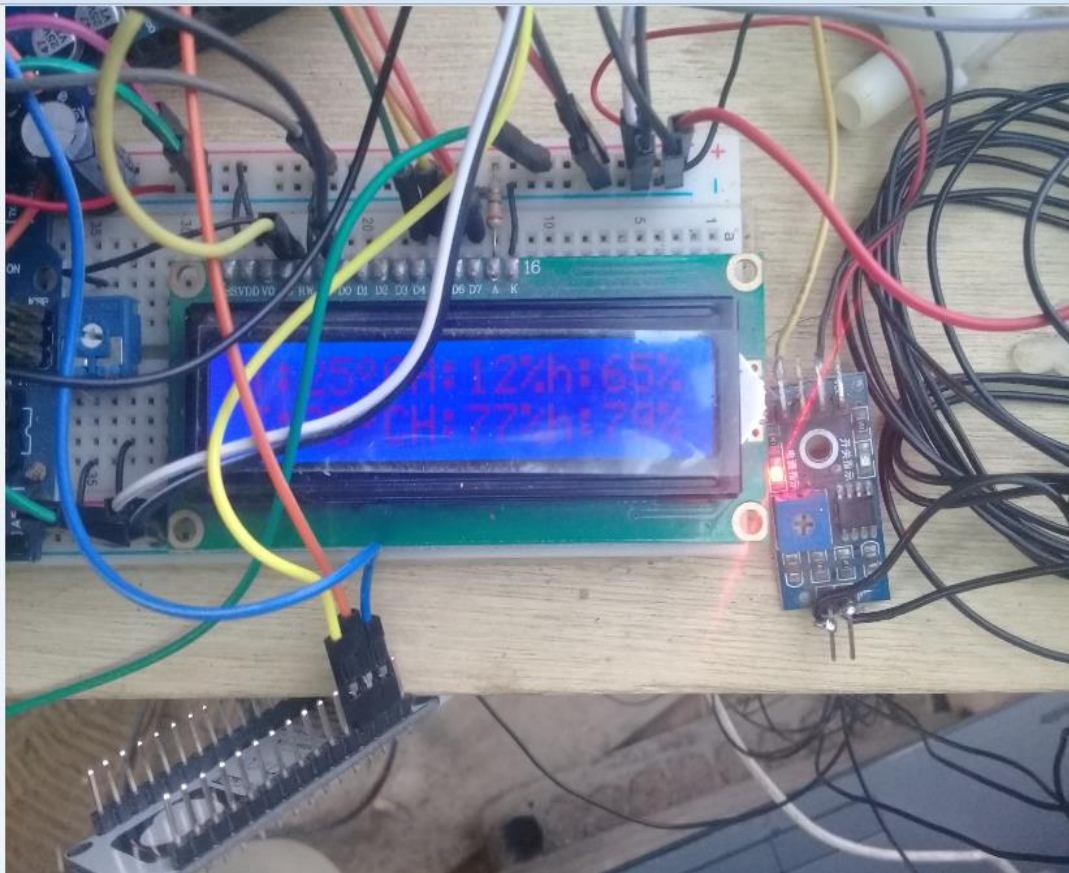


FIGURA 38: NODEMCU CONECTADA AL CIRCUITO
(FUENTE: PROPIA)

Paso 4:

Verificamos que en la base de datos ha cambiado en tiempo real: vemos que la humedad relativa de la planta 1(Begonia),la humedad del suelo es de 80% y la temperatura es del 25 C y la temperatura del ambiente es del 26

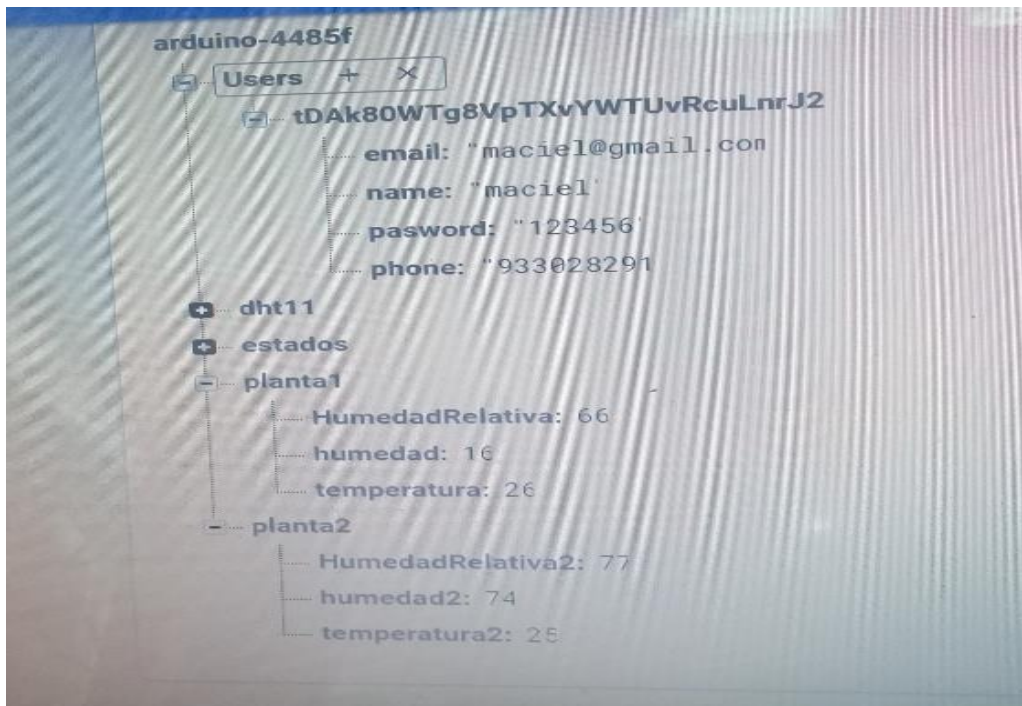


FIGURA 39: VISUALIZACIÓN EN TIEMPO REAL DE LAS VARIABLES CLIMÁTICAS (FUENTE: PROPIA)

Paso5: Como verificamos que la humedad esta baja activamos el riego mediante la aplicación



FIGURA 40: ACTIVACIÓN DE RIEGO PARA LA PLANTA (FUENTE: PROPIA)

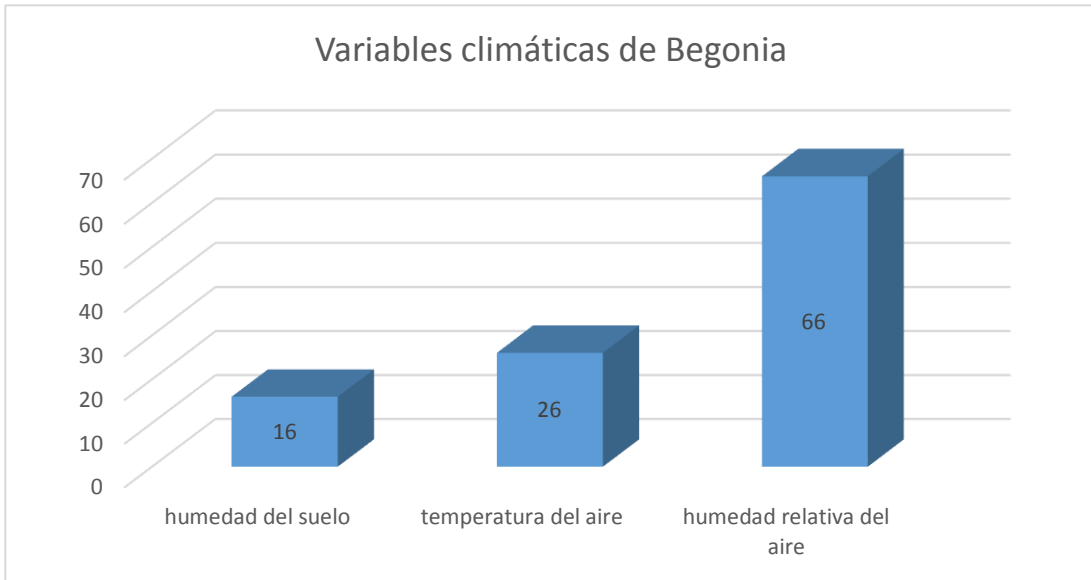


FIGURA 41: VARIACIÓN DE LAS VARIABLES CLIMÁTICAS (FUENTE: PROPIA)

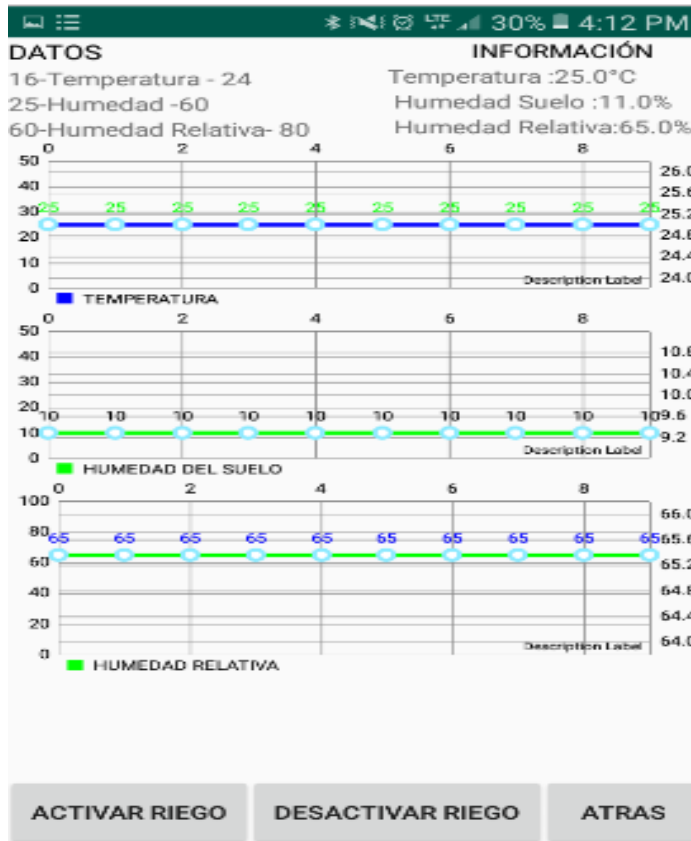


FIGURA 42: CONTROL DE RIEGO (FUENTE: PROPIA)

GERANIO

El geranio es una planta que puede soportar temperaturas hasta 30°C pero no puede soportar el frío, pero no puede soportar temperaturas menos de 10°C



FIGURA 43: MEDICIÓN DE LAS VARIABLES CLIMÁTICAS
(FUENTE: PROPIA)

PASO 1:

Colocamos el sensor de humedad en la tierra lo introducimos hasta el fondo y el sensor DHT que mide la temperatura y la humedad del aire



FIGURA 44: SENSORES DE HUMEDAD Y TEMPERATURA EN LA PLANTA GERANIO

FUENTE: PROPIA

Paso 2:

Verificamos que en la base de datos ha cambiado en tiempo real :vemos que la humedad relativa de la planta 2(Geranio),la humedad del suelo es de 80% y la temperatura es del 25 C

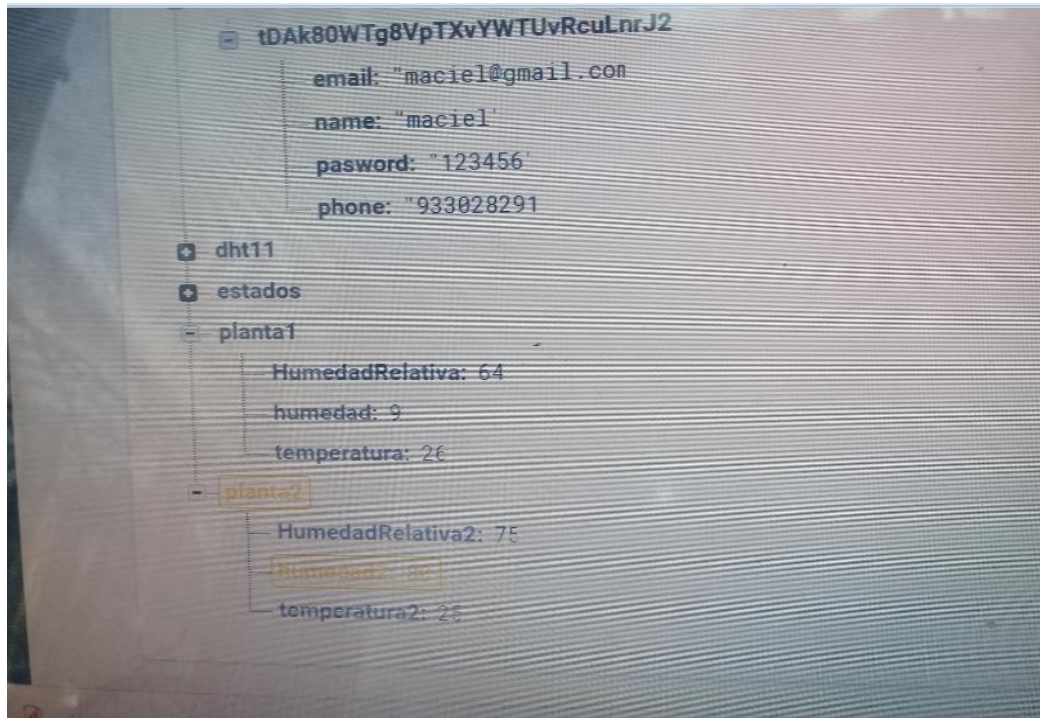


FIGURA 45: VISUALIZACIÓN DE LAS VARIABLES CLIMÁTICAS EN FIREBASE (FUENTE: PROPIA)

PASO 3:

Verificamos en la aplicación que los valores de las variables climáticas en tiempo real , en la parte izquierda verificamos la información de la planta 2(Geranio), en la parte derecha verificamos los valores en tiempo real y en la parte inferior los gráficos del ultimo valor captado en tiempo real

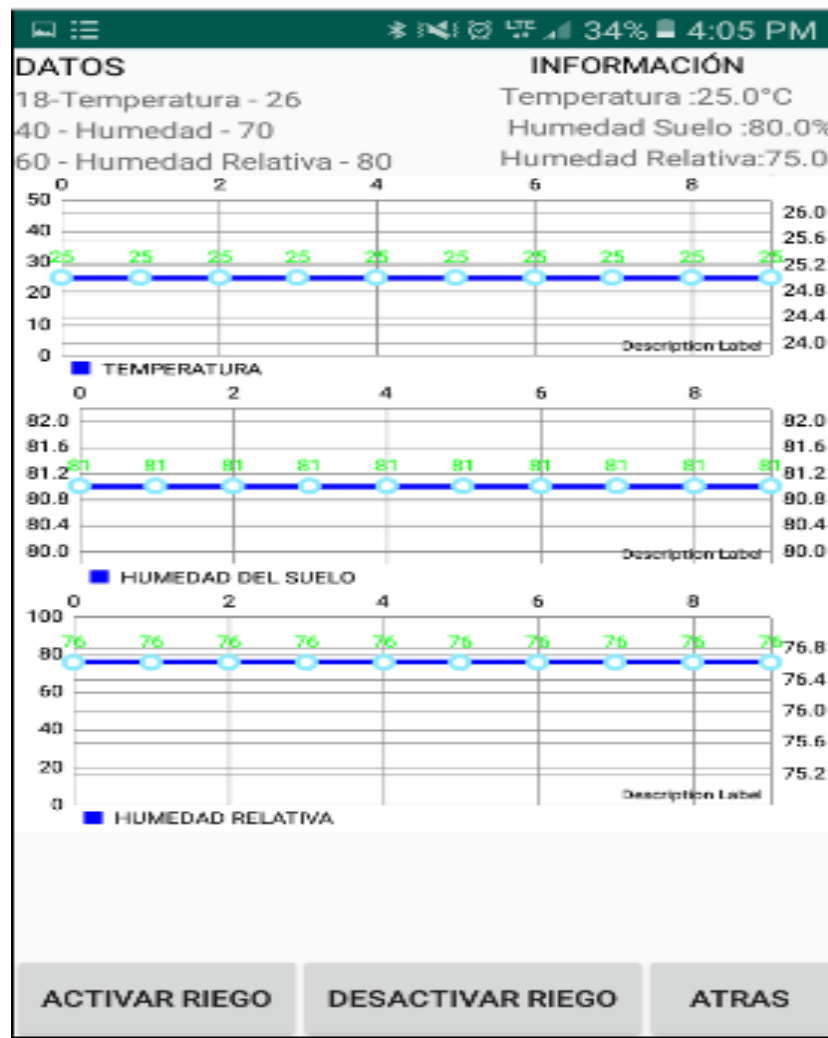


FIGURA 46: CONTROL DE RIEGO EN LA PLANTA GERANIO (FUENTE: PROPIA)

PASO4:

Verificamos que la humedad del suelo esta en 80% es decir esta muy húmedo por lo cual no es necesario activar el riego.

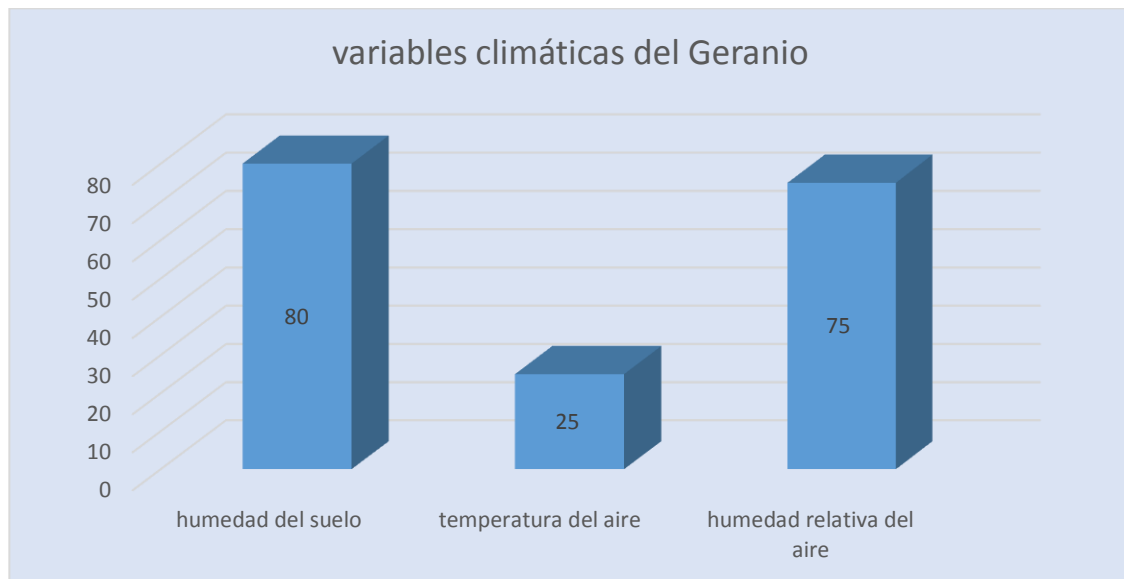


FIGURA 47: VARIACIÓN DE LAS VARIABLES CLIMÁTICAS (FUENTE: PROPIA)

CONCLUSIONES

La tecnología arduino permitio capturar las variables climáticas: humedad del suelo, humedad relativa del aire y temperatura del aire en el invernadero de la UNTELS.

La tecnología arduino permitió enviar las variables climáticas :humedad del suelo, humedad relativa del aire y temperatura del aire hacia firebase en el invernadero UNTELS

A través de la aplicación móvil se logró monitorear en tiempo real las variables climáticas en el invernadero de la UNTELS.

A través de la aplicación móvil se logró realizar el control de riego en las plantas en el invernadero de la UNTELS.

Se produjo un ahorro en el consumo de agua en las plantas ya que el riego se producirá en un determinado tiempo hasta que se desactive mediante la aplicación.

Se determino que es viable mejorar la eficiencia de los sistemas productivos con la tecnología arduino con precisión y de manera localizada lo cual implica ahorro en mano de obra y tiempo.

RECOMENDACIONES

Se recomienda implementar un balanceo de carga dinámico cuando la aplicación se use en grandes escalas.

Se debe incentivar a los agricultores para que hagan uso de la tecnología arduino por su bajo costo y efectividad en la captura de las variables climáticas

Se recomienda implementar un control de la temperatura con el uso de ventiladores o un sistema de calefacción que sea automatizado en un invernadero.

Se recomienda implementar más sensores para que el control del invernadero sea completo y sea generalizado para cualquier invernadero o biohuerto.

BIBLIOGRAFÍA

- Acosta, A. (2015). Automatización de bajo costo utilizada en la producción agrícola en invernaderos y huertos caseros.
- Acuña, J., Bojacá, C., & Villagra, E. (2016). Diseño y evaluación climática de un invernadero para condiciones de clima intertropical de montaña.
- Alaimo, D. (2013). Proyectos ágiles con Scrum. Buenos Aires.
- Aliaga Mendoza, G., & Pedro, Q. B. (2015). Sistema de control de humedad relativa para un invernadero utilizando el controlador lógico programable. Huancavelica.
- Amangandi. (2012). Arduino.
- Amaya, Y. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles.
- Atom. (14 de 3 de 2013). importancia de los invernaderos. Obtenido de Invernaderos: <http://rochinb4.blogspot.com/2013/03/importancia-de-los-invernaderos.html>
- Barrozo, A. (2015). CONTROL Y MONITORIZACIÓN DE UN INVERNADERO. Madrid.
- Calderon Quintana, M. F., & Torres Suarez, V. (2018). Diseño de un sistema de control y mantenimiento de un bio-huerto vertical en edificaciones comerciales de la zona céntrica de Guayaquil.
- Calvo, D. (2018). Metodología XP Programación Extrema (Metodología ágil).
- Dea. (9 de 9 de 2014). DHT11 Sensor de humedad y temperatura. Obtenido de Geekock Electronics: <http://www.geekbotelectronics.com/dht11-sensor-de-humedad-y-temperatura/>
- Enciso, J., & Porter, 2. (2018). Uso de sensores de humedad de suelo para eficientizar el riego.
- Gallego, M. T. (n.d.).
- García, M. C. (2017). Desarrollo de una aplicación Android de apuestas.

Honores, J., & Sanchez, A. (2016). Implementación de un módulo de agricultura de precisión para monitoreo y control de riego, integrado al sistema IOTMACH.

iberica. (n.d.). PCE-55. Obtenido de Medidores de humedad: <https://www.pce-iberica.es/Catalogo/catalogo-humedad.pdf>

Ima invernaderos. (s.f.). Obtenido de Sistemas de calefacción: <https://www.invernaderosima.com/es/sistemas-de-calefacci%C3%B3n.html>

INFOAGRO. (n.d.). Principales tipos de invernaderos (Parte I).

Jhonatan Baltazar Aguilar & Enciso Hernandez Daniel, V. D. (2014). Diseño e implementación de un dispositivo Digital para el control de la temperatura en un invernadero de tomate.

Jiménez, K. (2012). Propuesta de metodología y estándares para la administración de proyectos en las pequeñas y medianas empresas de software con base a los estándares del PMI. Costa Rica.

Kruchten, P. (2003). The Rational Unified Process and Introduction.

LLamas, L. (3 de 4 de 2019). ¿QUÉ ES UN HIGRÓMETRO FC-28? Obtenido de MEDIR LA HUMEDAD DEL SUELO CON ARDUINO E HIGRÓMETRO FC-28: <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

Lopez, B. (2012). Tipos de invernaderos.

Lopez, S. (2018). TRABAJO DE INVESTIGACIÓN PARA OBTENER EL GRADO.

Molina, & Jiménez. (2010). Aplicativos móviles.

Monica, T. M. (2013). ESTUDIO Y DESARROLLO DE APLICACIONES PARA.

Mori, G. (2018). IMPLEMENTACIÓN, CONTROL Y MONITOREO DE UN SISTEMA DE RIEGO POR GOTEO SUBTERRÁNEO CON MICROCONTROLADORES. Lima.

Narvaez, R. (2006). Proyecto de automatización de un invernadero para el cultivo de diversos productos auténticos de la zona donde se encuentra ubicada la hacienda "AYALA".

- Navarro, A., Fernandez, J., & Jonathan, M. (2017). Revisión de metodologías Ágiles para el desarrollo del software. España.
- Navas, Benavente, & Elena. (2000). Automatismos de control de clima en invernaderos.
- Novagric. (20 de 10 de 2015). Obtenido de Clima de un Invernadero. ¿Cómo conseguir la Temperatura Ideal?:
<http://www.novagric.com/es/blog/articulos/clima-invernadero-como-conseguir-temperatura-ideal>
- Perea Palacios, J. (2016). Diseño de un sistema de monitoreo, registro, control de temperatura y humedad para un cultivo de invernadero. Pereira.
- Perez Rojas, H., & Cortez De Paul, M. (2001). SIMULACION Y CONTROL DE LA TEMPERATURA DENTRO DE UN INVERNADERO. Bogota.
- Portillo, G. (14 de 8 de 2018). El psicrometro. Obtenido de Qué es el psicrómetro: <https://www.meteorologiaenred.com/psicrometro.html>
- Reveles, Minjares, Varela, & Telles. (2008). Control de Temperatura de un Invernadero a Escala .
- Reyna, C. (2015). Sistema automatizado para monitoreo y control de humedad en un invernadero.
- Sanchez Castro, R. (30 de 1 de 2012). Definición de invernadero. Obtenido de invernaderos: http://invernaderosramon.blogspot.com/2012/01/definicion-de-invernadero_30.html
- Sensor de humedad del suelo. (n.d.). Obtenido de Naylamp :
<https://naylampmechatronics.com/sensores-temperatura-y-humedad/47-sensor-de-humedad-del-suelo.html>
- Study, S. (2017). Una guía para el cuerpo de conocimiento de scrum(tercera edición).
- Valverde, D. (2010). Introducción a la programación Extrema.

ANEXOS

PROGRAMA EN ARDUINO PARA QUE ENVIE LOS DATOS MEDIANTE LOS SENSORES

```
//INCLUIR LIBRERIA PARA LA LECTURA DE LA TEMPERATURA
```

```
#include "DHT.h"
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); //(RS,E,D4,D5,D6,D7)
```

```
//CONFIGURACION DEL SENSOR DHT11 Y DHT22
```

```
#define DHTPIN 9
```

```
#define DHTPIN2 10
```

```
#define DHTTYPE DHT11
```

```
#define DHTTYPE2 DHT22
```

```
DHT dht(DHTPIN,DHTTYPE);
```

```
DHT dht2(DHTPIN2,DHTTYPE2);
```

```
int riego1; //ESTADO DEL RIEGO1
```

```
int riego2; //ESTADO DEL RIEGO2
```

```
int pinriego1 = 11; //PIN DE SALIDA DEL RIEGO1
```

```
int pinriego2 = 12; //PIN DE SALIDA DEL RIEGO2
```

```
//DEFINAMOS EL SIMBOLO DE °
```

```
byte grado[8] =
```

```
{
```

```
0b00001100,
```

```
0b00010010,
```

```
0b00010010,
```



```
0b00001100,  
0b00000000,  
0b00000000,  
0b00000000,  
0b00000000  
};
```

```
int valor; //VARIBLE GLOBAL QE ALMACENA EL VALOR DE LECTURA DEL  
SENSOR DE HUMEDAD
```

```
int valorMaximo = 500;//VARIBLE PARA COLOCAR VALOR MAXIMO
```

```
int val;
```

```
int val1;
```

```
void setup() {
```

```
//CONFIGURACION INICIAL
```

```
dht.begin();//INICIALIZACION DEL SENSOR dht11
```

```
dht2.begin();//INICIALIZACION DEL SENSOR dht22
```

```
Serial.begin(115200); //COMUNICACION Y VELOCIDAD DEL MONITOR  
SERIAL Y ARDUINO
```

```
lcd.begin(16, 2);
```

```
pinMode(pinriego1,OUTPUT); //DEFINIMOS COMO PIN DE SALIDA
```

```
pinMode(pinriego2,OUTPUT); //DEFINIMOS COMO PIN DE SALIDA
```

```
lcd.setCursor(5,0); //
```

```
lcd.print("invernadero");
```

```
lcd.setCursor(5,1);
```

```
lcd.print("Maciel");
```

```
delay(2000);
```

```
lcd.clear();
```

```
digitalWrite(pinriego1,HIGH);
```

```
digitalWrite(pinriego2,HIGH);
```

```
}
```

```
void loop()
```

```
{//BUCLE INFINITO REPETITIVO
```

```
recibirdata();
```

```
}
```

```
int sensorT() {
```

```
    //int temp= thermocouple.readCelsius();// lectura del sensor con la libreria
```

```
    int t = dht.readTemperature();
```

```
    delay(500);
```

```
lcd.createChar(1,grado);  
lcd.setCursor(0,0);  
lcd.print("T:");  
lcd.print(t);  
lcd.write(1);  
lcd.print("C");  
return t;  
  
}
```

```
int sensorH(){  
    int lectura = analogRead(0); //realizamos la lectura del sensor A0  
    lectura = map(lectura,400,1023,100,0);//mapeamos O CAMBIAMOS LOS  
    VALORES DE 0 A 100%  
    lcd.setCursor(6,0);  
    lcd.print("H:");  
    lcd.print(lectura);  
    lcd.print("%");  
    return lectura;  
  
}
```

```
int sensorH2(){
```

```
int h2 = dht.readHumidity();//PARA OBTENER LA HUMEDAD RELATIVA  
DEL SENSOR
```

```
delay(500);
```

```
lcd.setCursor(11,0);
```

```
lcd.print("h:");
```

```
lcd.print(h2);
```

```
lcd.print("%");
```

```
return h2;
```

```
}
```

```
int sensor2T() {
```

```
//int temp= thermocouple.readCelsius();// lectura del sensor con la libreria
```

```
int t2 = dht2.readTemperature(); //PARA OBTENER LA TEMERATURA DEL  
SENSOR
```

```
lcd.createChar(1,grado);
```

```
lcd.setCursor(0,1);
```

```
lcd.print("T:");
```

```
lcd.print(t2);
```

```
lcd.write(1);
```

```
lcd.print("C");
```

```
return t2;
```

```

}

int sensor2H(){

    int lectura2 = analogRead(1); //realizamos la lectura del sensor A1

    lectura2 = map(lectura2,400,1023,100,0);//mapeamos O CAMBIAMOS LOS
VALORES DE 0 A 100%

    lcd.setCursor(6,1);

    lcd.print("H:");

    lcd.print(lectura2);

    lcd.print("%");

    return lectura2;

}

```

```

int sensor2H2(){

    int h22 = dht2.readHumidity();

    delay(500);

    lcd.setCursor(11,1);

    lcd.print("h:");

    lcd.print(h22);

    lcd.print("%");

    return h22;

}

```

```
void activarriego(){  
    if(riego1 == 1){  
        digitalWrite(pinriego1,LOW);  
        delay(500);  
    }  
    if(riego1 == 0){  
        digitalWrite(pinriego1,HIGH);  
        delay(500);  
    }  
    if(riego2 == 1){  
        digitalWrite(pinriego2,LOW);  
        delay(500);  
    }  
  
    if(riego2 == 0){  
        digitalWrite(pinriego2,HIGH);  
        delay(500);  
    }  
}
```

```
void recibirdata(){  
    int sensor1 = sensorT();  
    int sensor2 = sensorH();
```

```

int sensor3 = sensorH2();

int sensor4 = sensor2T();

int sensor5 = sensor2H();

int sensor6 = sensor2H2();

Serial.print("I"); //inicio de trama

Serial.print(sensor1);

Serial.print(":");

Serial.print(sensor2);

Serial.print(":");

Serial.print(sensor3);

Serial.print(":");

Serial.print(sensor4);

Serial.print(":");

Serial.print(sensor5);

Serial.print(":");

Serial.print(sensor6);

Serial.print("F"); //fin de trama

Serial.flush(); //Esperamos hasta que se envíen los datos

delay(100);

//----Leemos la respuesta del Esclavo-----

if(Serial.find("i"))//esperamos el inicio de trama

{

    riego1=Serial.parseInt(); //recibimos valor numérico

    if(Serial.read()=='\n') //Si el fin de trama es el correcto

```

```

{
  riego2 = Serial.parseInt();

  if(Serial.read()=='f')
    activarriego();

}

}

}

```

PROGRAMA EN EL ARDUINO PARA CAPTURAR MEDIANTE EL NODEMCU LAS VARIABLES CLIMÁTICAS

```

#include <ESP8266WiFi.h>

#include <FirebaseArduino.h>

// Set these to run example.

#define FIREBASE_HOST "arduino-4485f.firebaseio.com"

#define FIREBASE_AUTH "57oXRgoksffZyUn0SeIpApwy7UE87Rr2j9uslJav"

#define WIFI_SSID "AndroidAP" //"MOVISTAR_2BD0"

#define WIFI_PASSWORD "123456789" //"mFgQfuZcKacEuq8gNqfe"

int sensor1; //humedad

int sensor2; //temperatura

int sensor3; //humedadrelativa

```



```
int sensor4; //temperatura2

int sensor5; //humedad2

int sensor6; //humedadrelativa2

void setup() {

  Serial.begin(115200);

  // connect to wifi.

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  //Serial.print("connecting");

  while (WiFi.status() != WL_CONNECTED) {

    //Serial.print(".");

    delay(500);

  }

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

}

void loop() {

  estadoriego();

}
```

```
}
```

```
void funciontemp(int dato)
```

```
{
```

```
  Firebase.setInt("/planta1/temperatura", dato);
```

```
  Serial.println("Temperaturaf");
```

```
  Serial.println(dato);
```

```
  delay(100);
```

```
}
```

```
void funcionhumedad(int dato)
```

```
{
```

```
  Firebase.setInt("/planta1/humedad", dato);
```

```
  Serial.println("humedadf");
```

```
  Serial.println(dato);
```

```
  delay(100);
```

```
}
```

```
void funciontemp2(int dato)
```

```
{
```

```
  Firebase.setInt("/planta2/temperatura2", dato);
```

```
Serial.println("Temperatura2f");  
Serial.println(dato);  
delay(100);  
}
```

```
void funcionhumedad2(int dato)  
{  
  Firebase.setInt("/planta2/humedad2", dato);  
  delay(100);  
  Serial.println("humedad2f");  
  Serial.println(dato);  
}
```

```
void funcionhumedadrelativa(int dato)  
{  
  Firebase.setInt("/planta1/HumedadRelativa", dato);  
  Serial.println("HumedadRelativaf");  
  Serial.println(dato);  
  delay(100);  
}
```

```
void funcionhumedadrelativa2(int dato)  
{  
  Firebase.setInt("/planta2/HumedadRelativa2", dato);
```

```

Serial.println("HumedadRelativa2f");

Serial.println(dato);

delay(100);

}

void estadoriego(){

if(Serial.available()>0){ //ESTA DISPONIBLE EL PUERTO SERIAL

    if(Serial.read()=='I'){//inicio de la trama

        sensor1 = Serial.parseInt();

        if(Serial.read() == ':'){

            sensor2 = Serial.parseInt();

            if(Serial.read() == ':'){

                sensor3 = Serial.parseInt();

                if(Serial.read() == ':'){

                    sensor4 = Serial.parseInt();

                    if(Serial.read() == ':'){

                        sensor5 = Serial.parseInt();

                        if(Serial.read() == ':'){

                            sensor6 = Serial.parseInt();

                            funciontemp(sensor1);

                                funcionhumedad(sensor2);

                                    funciontemp2(sensor4);

```

```

        funcionhumedad2(sensor5);

        funcionhumedadrelativa(sensor3);

        funcionhumedadrelativa2(sensor6);

if(Serial.read()=='F') //indica el final de la trama
{
    int riego1 = Firebase.getInt("estados/estadoplanta1");
    int riego2 = Firebase.getInt("estados/estadoplanta2");

    Serial.print("i");//iniio de envio de la trama

    Serial.print(riego1);

    Serial.print(":");

    Serial.print(riego2);

    Serial.print('f');//fin de la trama

}

}

}

}

}

```

```

    }

}

}

void firebaseSendValues(){
    int t = sensor1;
    int h = sensor2;
    if ((t >= -15 && t <= 80) && (h >= 0 && h <= 100)) {
        // === Push temperature value to Firebase ===
        String tempValueID = Firebase.pushInt("dht11/temperature", t);
        if (Firebase.failed()) {
            Serial.print("[ERROR] pushing /dht11/temperature failed:");
            Serial.println(Firebase.error());
            return;
        }
        Serial.print("[INFO] pushed: /dht11/temperature \tkey: ");
        Serial.println(tempValueID);

        // === Push humidity value to Firebase ===
        String humValueID = Firebase.pushInt("dht11/humidity", h);

```

```

if (Firebase.failed()) {
    Serial.print("[ERROR] pushing /dht11/humidity failed:");
    Serial.println(Firebase.error());
    return;
}

Serial.print("[INFO] pushed: /dht11/humidity \tkey: ");
Serial.println(humValueID);
Serial.println();
} else {
    Serial.println("Wrong values!");
}
}

```

CODIGO PARA ANDROID

CLASE BIOHUERTO

```

package com.example.pc.biohuertoapp;

public class biohuerto {

    private int temperatura;
    private int humedad;
    private int HumedadRelativa;
    private int temperatura2;
    private int humedad2;
    private int HumedadRelativa2;

    public biohuerto() {
        //por defecto se incluye el constructor
    }

    public biohuerto(int temperatura, int humedad , int

```

```

HumedadRelativa, int temperatura2, int humedad2 , int HumedadRelativa2) {
    this.temperatura = temperatura;
    this.humedad = humedad;
    this.HumedadRelativa = HumedadRelativa;
    this.temperatura2 = temperatura2;
    this.humedad2 = humedad2;
    this.HumedadRelativa2 = HumedadRelativa2;
}

public int getTemperatura() {
    return temperatura;
}

public void setTemperatura(int temperatura) {
    this.temperatura = temperatura;
}

public int getHumedad() {
    return humedad;
}

public void setHumedad(int humedad) {
    this.humedad = humedad;
}

public int getHumedadrelativa() {
    return HumedadRelativa;
}

public void setHumedadrelativa(int HumedadRelativa) {
    this.HumedadRelativa = HumedadRelativa;
}

public int getTemperatura2() {
    return temperatura2;
}

public void setTemperatura2(int temperatura2) {
    this.temperatura2 = temperatura2;
}

public int getHumedad2() {
    return humedad2;
}

public void setHumedad2(int humedad2) {
    this.humedad2 = humedad2;
}

public int getHumedadRelativa2() {
    return HumedadRelativa2;
}

public void setHumedadRelativa2(int humedadRelativa2) {
    HumedadRelativa2 = humedadRelativa2;
}
}

```


MAIN ACTIVITY BIOHUERTO

```
package com.example.pc.biohuertoapp;

import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;

import android.support.design.widget.Snackbar;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.RelativeLayout;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.rengwuxian.materialedittext.MaterialEditText;

public class MainActivity extends AppCompatActivity {

    Button btnlogin, btnregister;
    FirebaseAuth auth;
    FirebaseDatabase db;
    DatabaseReference users;
    RelativeLayout rootLayout;
    int duration = 45;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnlogin = findViewById(R.id.btn_sign);
        btnregister = findViewById(R.id.btn_register);
        auth = FirebaseAuth.getInstance();
        db = FirebaseDatabase.getInstance();
        users = db.getReference("Users");

        btnregister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                registerdialog();
            }
        });

        btnlogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                logindialog();
            }
        });
    }
}
```

```

    });
}

private void registerdialog() {

    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("REGISTER");
    dialog.setMessage("Porfavor usar email para registrarse");

    LayoutInflater inflater = LayoutInflater.from(this);
    View register_layout =
inflater.inflate(R.layout.layout_register, null);

    final MaterialEditText edtEmail =
register_layout.findViewById(R.id.edtEmail);
    final MaterialEditText edtpswd =
register_layout.findViewById(R.id.edtpswd);
    final MaterialEditText edtnamee =
register_layout.findViewById(R.id.edtname);
    final MaterialEditText edtphone =
register_layout.findViewById(R.id.edtphone);

    dialog.setView(register_layout);

    dialog.setPositiveButton("REGISTER", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();

            //check validate
            if(TextUtils.isEmpty(edtEmail.getText().toString())){
                Snackbar.make(rootLayout, "porfavor ingresar un email
valido", Snackbar.LENGTH_SHORT).show();
                return;
            }
            if(TextUtils.isEmpty(edtphone.getText().toString())){
                Snackbar.make(rootLayout, "porfavor ingresar una
contraseña valida", Snackbar.LENGTH_SHORT).show();
                return;
            }
            if(TextUtils.isEmpty(edtpswd.getText().toString())){
                Snackbar.make(rootLayout, "porfavor ingresar una
contraseña valida", Snackbar.LENGTH_SHORT).show();
                return;
            }

            if(edtpswd.getText().toString().length() < 6){
                Snackbar.make(rootLayout, "contraseña muy
corta", Snackbar.LENGTH_SHORT).show();
                return;
            }

            if(TextUtils.isEmpty(edtnamee.getText().toString())){
                Snackbar.make(rootLayout, "porfavor ingresar una
nombre valido", Snackbar.LENGTH_SHORT).show();
                return;
            }

```

```

    }

    auth.createUserWithEmailAndPassword(edtEmail.getText().toString(),edtpswd
    .getText().toString()).addOnSuccessListener(new
    OnSuccessListener<AuthResult>() {
        @Override
        public void onSuccess(AuthResult authResult) {

            Users user = new Users();
            user.setEmail(edtEmail.getText().toString());
            user.setPassword(edtpswd.getText().toString());
            user.setPhone(edtphone.getText().toString());
            user.setName(edtnamee.getText().toString());
            users.child(FirebaseAuth.getInstance()

            .getCurrentUser().getUid()).setValue(user).addOnSuccessListener(new
            OnSuccessListener<Void>() {
                @Override
                public void onSuccess(Void aVoid) {
                    //Snackbar.make(rootLayout,"Register
                    successful",Snackbar.LENGTH_SHORT).show();
                    showSnackBar(MainActivity.this,"Registro
                    satisfactorio");
                }
            }).addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {

                    //Snackbar.make(rootLayout,"Failed"+e.getMessage(),Snackbar.LENGTH_SHORT)
                    .show();

                    showSnackBar(MainActivity.this,"Fallo!!");

                }
            });
        }
    });

    dialog.setNegativeButton("CANCELAR", new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });

    dialog.show();
}

```

```

private void logindialog() {
    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setTitle("LOGIN");
    dialog.setMessage("Porfavor usar email para ingresar");

    LayoutInflater inflater = LayoutInflater.from(this);
    View login_layout =
inflater.inflate(R.layout.lasyout_login,null);

    final MaterialEditText edtEmail =
login_layout.findViewById(R.id.getEmail);
    final MaterialEditText edtpswd =
login_layout.findViewById(R.id.getpswd);

    dialog.setView(login_layout);

    dialog.setPositiveButton("LOGIN", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();

            //waitingDialog = new SpotsDialog(MainActivity.this.,
"write_your_string_here. for ex: Loading..");
            //waitingDialog.show();

            btnlogin.setEnabled(false);

            //check validate
            if(TextUtils.isEmpty(edtEmail.getText().toString())){
                //Snackbar.make(rootLayout,"porfavor ingresar un
email ",Snackbar.LENGTH_SHORT).show();
                showSnackBar(MainActivity.this,"porfavor ingresar un
email");
            }
            return;

            if(TextUtils.isEmpty(edtpswd.getText().toString())){
                //Snackbar.make(rootLayout,"porfavor ingresar una
contraseña ",Snackbar.LENGTH_SHORT).show();
                showSnackBar(MainActivity.this,"porfavor ingresar una
contraseña ");
            }
            return;

            if(edtpswd.getText().toString().length() < 6){
                //Snackbar.make(rootLayout,"contraseña muy
corta",Snackbar.LENGTH_SHORT).show();
                showSnackBar(MainActivity.this,"contraseña muy
corta");
            }
            return;
        }
    });

    //login

```

```

auth.signInWithEmailAndPassword(edtEmail.getText().toString(), edtpwsd.getText().toString()).addOnSuccessListener(new
OnSuccessListener<AuthResult>() {
    @Override
    public void onSuccess(AuthResult authResult) {
        startActivity(new
Intent(MainActivity.this, Main2Activity.class));
        finish();
    }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {

//Snackbar.make(rootLayout, "Failed", Snackbar.LENGTH_SHORT).show();
        showSnackBar(MainActivity.this, "Usuario no
registrado!");

        btnlogin.setEnabled(true);
    }
});
}
});

dialog.setNegativeButton("CANCELAR", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
    }
});

dialog.show();
}

public void showSnackBar(Activity activity, String message){
    View rootView =
activity.getWindow().getDecorView().findViewById(android.R.id.content);
    Snackbar.make(rootView, message, duration).show();
}

}

```

CLASE DE LA PLANTA 1

```

package com.example.pc.biohuertoapp;

import android.content.Intent;
import android.graphics.Color;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

```

```

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.interfaces.datasets.ILineDataSet;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;
import java.util.List;

public class PLANTA1 extends AppCompatActivity {

    private static final String TAGLOG = "dbmaciel" ;
    DatabaseReference myRef;
    DatabaseReference myref2;
    ValueEventListener myvaluelistener;
    TextView lbltemperatura, lblhumedad , lblhumedadrelativa;
    LineChart mChart ,mChart2, mChart3;
    Button btnriego, btnnoriego, btnatras;
    float x,hum ,z;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_planta1);

        lbltemperatura = findViewById(R.id.lbltemp);
        lblhumedad = findViewById(R.id.lblhum);
        lblhumedadrelativa = findViewById(R.id.lbl3);
        btnnoriego = findViewById(R.id.btnnoriego);
        btnriego = findViewById(R.id.btnriego);
        btnatras = findViewById(R.id.btnatras);

        mChart = findViewById(R.id.linechart);
        mChart2 = findViewById(R.id.linechart2);
        mChart3 = findViewById(R.id.linechart3);
        btnriego.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                myref2 =
                FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta
1");

                int x = 1;

                myref2.setValue(x);
            }
        });

        btnnoriego.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                myref2 =
                FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta

```

```

1");

        int x = 0;

        myref2.setValue(x);

    }
});

btnatras.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new
Intent(PLANTA1.this, Main2Activity.class));
    }
});

myRef =
FirebaseDatabase.getInstance().getReference().child("planta1");

myvaluelistener = new ValueEventListener() {

    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot)
{

        biohuerto bio =
dataSnapshot.getValue(biohuerto.class);
        x = bio.getTemperatura();
        hum = bio.getHumedad();
        z = bio.getHumedadrelativa();
        lbltemperatura.setText(x + "°C");
        lblhumedad.setText(hum + "%");
        lblhumedadrelativa.setText(z + "%");
        Log.e(TAGLOG, "ondatachange" +
dataSnapshot.getValue().toString());

    }

    @Override

    public void onCancelled(@NonNull DatabaseError databaseError)
{

        Log.e(TAGLOG, "Error" + databaseError.toException());

    }
};

myRef.addValueEventListener(myvaluelistener);

}

@Override
protected void onStart() {
    super.onStart();
    plot();
}

```

```

        plot2();
        plot3();
    }

    public void plot() {
        mChart.setDragEnabled(true);
        mChart.setScaleEnabled(false);

        YAxis leftAxis = mChart.getAxisLeft();
        leftAxis.removeAllLimitLines();
        leftAxis.setAxisMaximum(50f);
        leftAxis.setAxisMinimum(0f);

        ArrayList<Entry> yValues = new ArrayList<>();

        for(int i= 0;10>i;i++){
            if(x != 0){
                yValues.add(new Entry(i,Float.valueOf(x)));}
        }

        LineDataSet set1 = new LineDataSet(yValues, "TEMPERATURA" );
        set1.setFillAlpha(110);
        set1.setColor(Color.BLUE);
        set1.setLineWidth(3f);
        set1.setValueTextSize(10f);
        set1.setValueTextColor(Color.GREEN);

        ArrayList<ILineDataSet> dataSets = new ArrayList<>();
        dataSets.add(set1);

        LineData data = new LineData(dataSets);
        mChart.setData(data);
        mChart.invalidate();

        // enable scaling and dragging
        /*mChart.setDragEnabled(true);
        mChart.setScaleEnabled(true);
        mChart.setDrawGridBackground(false);
        mChart.setHighlightPerDragEnabled(true);
        mChart.setBackgroundColor(Color.WHITE);

        String[] valuesX = new String[] {"Tiempo (ms)"};
        String[] valuesY = new String[] {"MiliVolts (mV)"};*/
    }
    public void plot2() {
        mChart2.setDragEnabled(true);
        mChart2.setScaleEnabled(false);

        YAxis leftAxis = mChart2.getAxisLeft();
        leftAxis.removeAllLimitLines();
        leftAxis.setAxisMaximum(50f);
        leftAxis.setAxisMinimum(0f);

        ArrayList<Entry> yValues2 = new ArrayList<>();

```



```

        for(int i= 0;10>i;i++){
            if(hum != 0){
                yValues2.add(new Entry(i,Float.valueOf(hum)));}
        }

        LineDataSet set1 = new LineDataSet(yValues2,"HUMEDAD DEL SUELO"
);
        set1.setFillAlpha(110);
        set1.setColor(Color.GREEN);
        set1.setLineWidth(3f);
        set1.setValueTextSize(10f);
        set1.setValueTextColor(Color.BLACK);

        ArrayList<ILineDataSet> dataSets2 = new ArrayList<>();
        dataSets2.add(set1);

        LineData data = new LineData(dataSets2);
        mChart2.setData(data);
        mChart2.invalidate();

        // enable scaling and dragging
        /*mChart.setDragEnabled(true);
        mChart.setScaleEnabled(true);
        mChart.setDrawGridBackground(false);
        mChart.setHighlightPerDragEnabled(true);
        mChart.setBackgroundColor(Color.WHITE);

        String[] valuesX = new String[] {"Tiempo (ms)"};
        String[] valuesY = new String[] {"MiliVolts (mV)"};*/
    }
    public void plot3() {
        mChart3.setDragEnabled(true);
        mChart3.setScaleEnabled(false);

        YAxis leftAxis = mChart3.getAxisLeft();
        leftAxis.removeAllLimitLines();
        leftAxis.setAxisMaximum(100f);
        leftAxis.setAxisMinimum(0f);

        ArrayList<Entry> yValues3 = new ArrayList<>();

        for(int i= 0;10>i;i++){
            if(z != 0){
                yValues3.add(new Entry(i,Float.valueOf(z)));}
        }

        LineDataSet set3 = new LineDataSet(yValues3,"HUMEDAD RELATIVA" );
        set3.setFillAlpha(60);
        set3.setColor(Color.GREEN);
        set3.setLineWidth(3f);
        set3.setValueTextSize(10f);
        set3.setValueTextColor(Color.BLUE);
    }

```

```

        ArrayList<ILineDataSet> dataSets3 = new ArrayList<>();
        dataSets3.add(set3);

        LineData data3 = new LineData(dataSets3);
        mChart3.setData(data3);
        mChart3.invalidate();

// enable scaling and dragging
        /*mChart.setDragEnabled(true);
        mChart.setScaleEnabled(true);
        mChart.setDrawGridBackground(false);
        mChart.setHighlightPerDragEnabled(true);
        mChart.setBackgroundColor(Color.WHITE);

        String[] valuesX = new String[] {"Tiempo (ms)"};
        String[] valuesY = new String[] {"MiliVolts (mV)"};*/
    }
}

```

CLASE DE LA PLANTA 2

```

package com.example.pc.biohuertoapp;

import android.content.Intent;
import android.graphics.Color;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.interfaces.datasets.ILineDataSet;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class PLANTA2 extends AppCompatActivity {

    private static final String TAGLOG = "dbmaciel" ;
    DatabaseReference myRef;
    DatabaseReference myef2, my3;
    ValueEventListener myvalueListener, myvalueListener2;
    TextView lbltemperatura, lblhumedad , lblhumedadrelativa;

```

```

Button btnriego2, btnnoriego2, btnatras2;
LineChart mChart,mChart2,mChart3;
float tempe, hume, hure;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_planta2);
    lbltemperatura = findViewById(R.id.lbltemp);
    lblhumedad = findViewById(R.id.lblhum);
    lblhumedadrelativa = findViewById(R.id.lblhumerelativa);
    btnnoriego2 = findViewById(R.id.btnnoriego2);
    btnriego2 = findViewById(R.id.btnriego2);
    btnatras2 = findViewById(R.id.btnatras2);

    mChart = findViewById(R.id.linechart11);
    mChart2 = findViewById(R.id.linechart22);
    mChart3 = findViewById(R.id.linechart33);

    btnriego2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            myef2 =
FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta
2");

            int x = 1;

            myef2.setValue(x);
        }
    });

    btnnoriego2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            myef2 =
FirebaseDatabase.getInstance().getReference().child("estados/estadoplanta
2");

            int x = 0;

            myef2.setValue(x);
        }
    });

    btnatras2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new
Intent(PLANTA2.this,Main2Activity.class));
        }
    });

    myRef =
FirebaseDatabase.getInstance().getReference().child("planta2");
    my3 =
FirebaseDatabase.getInstance().getReference().child("dht11");
    myvaluelistener = new ValueEventListener() {

```

```

        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot)
        {
            biohuerto bio = dataSnapshot.getValue(biohuerto.class);
            tempe = bio.getTemperatura2();
            hume = bio.getHumedad2();
            hure = bio.getHumedadRelativa2();
            lbltemperatura.setText(tempe+"°C");
            lblhumedad.setText(hume+"%");
            lblhumedadrelativa.setText(hure+"%");

            Log.e(TAGLOG , "ondatachange" +
dataSnapshot.getValue().toString());
        }

        @Override

        public void onCancelled(@NonNull DatabaseError databaseError)
        {
            Log.e(TAGLOG , "Error" + databaseError.toException());
        }
    };

    myRef.addValueEventListener(myvaluelistener);

    plot();
}

@Override
protected void onStart() {
    super.onStart();
    plot();
    plot2();
    plot3();
}

private void plot() {
    mChart.setDragEnabled(true);
    mChart.setScaleEnabled(false);

    YAxis leftAxis = mChart.getAxisLeft();
    leftAxis.removeAllLimitLines();
    leftAxis.setAxisMaximum(50f);
    leftAxis.setAxisMinimum(0f);

    ArrayList<Entry> yValues = new ArrayList<>();

    for(int i= 0;10>i;i++){
        if(tempe != 0){
            yValues.add(new Entry(i, tempe));}
    }

    LineDataSet set1 = new LineDataSet(yValues, "TEMPERATURA" );
    set1.setFillAlpha(110);
    set1.setColor(Color.BLUE);

```

```

set1.setLineWidth(3f);
set1.setValueTextSize(10f);
set1.setValueTextColor(Color.GREEN);

ArrayList<ILineDataSet> dataSets = new ArrayList<>();
dataSets.add(set1);

LineData data = new LineData(dataSets);
mChart.setData(data);

// enable scaling and dragging
mChart.setDragEnabled(true);
mChart.setScaleEnabled(true);
mChart.setDrawGridBackground(false);
mChart.setHighlightPerDragEnabled(true);
mChart.setBackgroundColor(Color.WHITE);

String[] valuesX = new String[] {"Tiempo (ms)"};
String[] valuesY = new String[] {"MiliVolts (mV)"};
}
private void plot2() {
    mChart2.setDragEnabled(true);
    mChart2.setScaleEnabled(false);

    YAxis leftAxis = mChart2.getAxisLeft();
    leftAxis.removeAllLimitLines();
    leftAxis.setAxisMaximum(50f);
    leftAxis.setAxisMinimum(0f);

    ArrayList<Entry> yValues = new ArrayList<>();

    for(int i= 0;10>i;i++){
        if(hume != 0){
            yValues.add(new Entry(i,hume));}
    }

    LineDataSet set1 = new LineDataSet(yValues,"HUMEDAD DEL SUELO" );
    set1.setFillAlpha(110);
    set1.setColor(Color.BLUE);
    set1.setLineWidth(3f);
    set1.setValueTextSize(10f);
    set1.setValueTextColor(Color.GREEN);

    ArrayList<ILineDataSet> dataSets = new ArrayList<>();
    dataSets.add(set1);

    LineData data = new LineData(dataSets);
    mChart2.setData(data);

// enable scaling and dragging
mChart2.setDragEnabled(true);
mChart2.setScaleEnabled(true);
mChart2.setDrawGridBackground(false);
mChart2.setHighlightPerDragEnabled(true);
mChart2.setBackgroundColor(Color.WHITE);

String[] valuesX = new String[] {"Tiempo (ms)"};

```

```

        String[] valuesY = new String[] { "MiliVolts (mV) " };
    }
    private void plot3() {
        mChart3.setDragEnabled(true);
        mChart3.setScaleEnabled(false);

        YAxis leftAxis = mChart3.getAxisLeft();
        leftAxis.removeAllLimitLines();
        leftAxis.setAxisMaximum(100f);
        leftAxis.setAxisMinimum(0f);

        ArrayList<Entry> yValues = new ArrayList<>();

        for(int i= 0;10>i;i++){
            if(hure != 0){
                yValues.add(new Entry(i,hure));}
        }

        LineDataSet set1 = new LineDataSet(yValues, "HUMEDAD RELATIVA" );
        set1.setFillAlpha(110);
        set1.setColor(Color.BLUE);
        set1.setLineWidth(3f);
        set1.setValueTextSize(10f);
        set1.setValueTextColor(Color.GREEN);

        ArrayList<ILineDataSet> dataSets = new ArrayList<>();
        dataSets.add(set1);

        LineData data = new LineData(dataSets);
        mChart3.setData(data);

        // enable scaling and dragging
        mChart3.setDragEnabled(true);
        mChart3.setScaleEnabled(true);
        mChart3.setDrawGridBackground(false);
        mChart3.setHighlightPerDragEnabled(true);
        mChart3.setBackgroundColor(Color.WHITE);

        String[] valuesX = new String[] { "Tiempo (ms) " };
        String[] valuesY = new String[] { "MiliVolts (mV) " };
    }
}

```

Main2 Activity

```

package com.example.pc.biohuertoapp;

import android.content.Intent;
import android.graphics.Color;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;

```

```

import android.widget.Button;
import android.widget.TextView;

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.interfaces.datasets.ILineDataSet;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class Main2Activity extends AppCompatActivity {

    private static final String TAGLOG = "dbmaciel" ;
    DatabaseReference myRef;
    ValueEventListener myvalueListener;
    Button planta1 , planta2 , salir;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        planta1 = findViewById(R.id.planta1);
        planta2 = findViewById(R.id.planta2);
        salir = findViewById(R.id.salir);

        planta1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(Main2Activity.this, PLANTA1.class));
            }
        });

        planta2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(Main2Activity.this, PLANTA2.class));
            }
        });

        salir.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new
Intent(Main2Activity.this, MainActivity.class));
            }
        });
    }
}

```

```
}  
  
}
```

CLASE USUARIO

```
package com.example.pc.biohuertoapp;  
  
public class Users {  
    private String email, password, phone, name;  
  
    public Users() {  
    }  
  
    public Users(String email, String password, String phone, String name)  
    {  
        this.email = email;  
        this.password = password;  
        this.phone = phone;  
        this.name = name;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
  
    public void setEmail(String email) {  
        this.email = email;  
    }  
  
    public String getPassword() {  
        return password;  
    }  
  
    public void setPassword(String password) {  
        this.password = password;  
    }  
  
    public String getPhone() {  
        return phone;  
    }  
  
    public void setPhone(String phone) {  
        this.phone = phone;  
    }  
}
```



```

    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

LAYOUT LOGIN

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:cardElevation="10dp">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_margin="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <com.rengwuxian.materialedittext.MaterialEditText
            android:id="@+id/getEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email"
            android:inputType="textEmailAddress"
            android:text=""
            android:textColor="@color/colorPrimary"
            android:textColorHint="@color/colorPrimary"
            android:textSize="20sp"

            app:met_baseColor="@color/colorPrimary"
            app:met_floatingLabel="highlight"
            app:met_primaryColor="@color/colorPrimary"
            app:met_singleLineEllipsis="true" />

    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_marginTop="100dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <com.rengwuxian.materialedittext.MaterialEditText
            android:id="@+id/getpswd"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

```

```

        android:hint="Pasword"
        android:inputType="textEmailAddress"
        android:text=""
        android:textColor="@color/colorPrimary"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"

        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"
        app:met_singleLineEllipsis="true" />
    </LinearLayout>
</android.support.v7.widget.CardView>

```

LAYOUT REGISTRER

```

<?xml version="1.0" encoding="utf-8" ?>
<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_margin="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <com.rengwuxian.materialedittext.MaterialEditText
            android:id="@+id/edtEmail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Email"
            android:inputType="textEmailAddress"
            android:text=""
            android:textColor="@color/colorPrimary"
            android:textColorHint="@color/colorPrimary"
            android:textSize="20sp"

            app:met_baseColor="@color/colorPrimary"
            app:met_floatingLabel="highlight"
            app:met_primaryColor="@color/colorPrimary"
            app:met_singleLineEllipsis="true" /> </LinearLayout>

        <LinearLayout
            android:layout_marginTop="110dp"
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <com.rengwuxian.materialedittext.MaterialEditText
                android:id="@+id/edtpswd"
                android:hint="@string/contra"

```

```

        android:inputType="textPassword"
        android:text=""
        android:textColor="@color/colorPrimary"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        app:met_singleLineEllipsis="true"
        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"/>
</LinearLayout>

<LinearLayout
    android:layout_marginTop="200dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.rengwuxian.materialedittext.MaterialEditText
        android:id="@+id/edtname"
        android:hint="Nombre"
        android:inputType="text"
        android:text=""
        android:textColor="@color/colorPrimary"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        app:met_singleLineEllipsis="true"
        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"/>
</LinearLayout>

<LinearLayout
    android:layout_marginTop="300dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <com.rengwuxian.materialedittext.MaterialEditText
        android:id="@+id/edtphone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Celular"
        android:inputType="phone"
        android:text=""
        android:textColor="@color/colorPrimary"
        android:textColorHint="@color/colorPrimary"
        android:textSize="20sp"

        app:met_baseColor="@color/colorPrimary"
        app:met_floatingLabel="highlight"
        app:met_primaryColor="@color/colorPrimary"
        app:met_singleLineEllipsis="true" />

```

```
</LinearLayout>

</android.support.v7.widget.CardView>
```

LAYOUT ACTIVITY MAIN

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/biohuerto"
android:id="@+id/rootLayout"
tools:context=".MainActivity">

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:gravity="center_horizontal"
    android:layout_marginTop="20dp">

    <TextView
        android:text="INVERNADERO "
        android:textSize="28sp"
        android:textAlignment="center"
        android:textColor="@android:color/black"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>

<TextView
    android:id="@+id/txt_w"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="20dp"
    android:layout_marginTop="20dp"
    android:text="COPYRIGHT MACIEL"
    android:textSize="12sp"
    android:textAlignment="center"
    android:textColor="@android:color/white"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```

<LinearLayout
    android:orientation="horizontal"
    android:weightSum="2"
    android:layout_above="@+id/txt_w"
    android:layout_margin="16dp"
    android:layout_centerHorizontal="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/btn_sign"
        android:background="@drawable/btn_login_background"
        android:text="LOGIN"
        android:textColor="@android:color/white"
        android:layout_marginRight="6dp"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btn_register"
        android:background="@drawable/btn_register_background"
        android:text="REGISTER"
        android:textColor="@android:color/black"
        android:layout_marginLeft="6dp"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content" />

</LinearLayout>

</RelativeLayout>

```

LAYOUT ACTIVITY _MAIN2

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/fondo"
    tools:context=".MainActivity">

    <LinearLayout
        android:id="@+id/linear111"
        android:layout_width="match_parent"
        android:orientation="vertical"
        android:gravity="center_horizontal"

```

```

android:layout_height="wrap_content">
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="INVERNADERO "
    android:textSize="34dp"
    android:textColor="@android:color/black" />
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" UNTELS"
    android:textSize="34dp"
    android:textColor="@android:color/black" />
</LinearLayout>

<LinearLayout
  android:id="@+id/linear222"
  android:layout_marginTop="150dp"
  android:layout_below="@+id/linear111"
  android:layout_width="match_parent"
  android:orientation="horizontal"
  android:layout_height="wrap_content">
  <Button
    android:id="@+id/planta1"
    android:text="PLANTA 1"
    android:textSize="30dp"
    android:layout_marginRight="30dp"
    android:layout_width="wrap_content"
    android:layout_height="match_parent" />
  <Button
    android:id="@+id/planta2"
    android:text="PLANTA 2"
    android:textSize="30dp"
    android:layout_width="wrap_content"
    android:layout_height="match_parent" />
</LinearLayout>
<LinearLayout
  android:id="@+id/linear333"
  android:layout_below="@id/linear222"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:orientation="horizontal">
  <ImageView
    android:background="@drawable/imagen1"
    android:layout_width="170dp"
    android:layout_height="200dp" />
  <ImageView
    android:layout_marginLeft="30dp"

    android:background="@drawable/imagen2"
    android:layout_width="180dp"
    android:layout_height="200dp" />
</LinearLayout>

<Button
  android:id="@+id/salir"
  android:text="SALIR"
  android:layout_alignParentBottom="true"
  android:gravity="center_horizontal"
  android:layout_centerHorizontal="true"

```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
</RelativeLayout>
```

LAYOUT ACTIVITY PLANTA1

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".PLANTA1">

    <LinearLayout
        android:id="@+id/linearuno"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="49dp"
            android:layout_height="wrap_content"
            android:textColor="@android:color/black"
            android:textSize="15sp"
            android:text="DATOS" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_toEndOf="@+id/linearuno"
            android:text="16-Temperatura - 24" />
        <TextView
            android:text="25-Humedad -60"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:text="60-Humedad Relativa- 80"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
<LinearLayout
    android:layout_alignParentRight="true"
    android:layout_toRightOf="@id/linearuno"
    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:text="
                                INFORMACIÓN"
            android:textColor="@android:color/black"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:text="
                                Temperatura :"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
            <TextView
                android:id="@+id/lbltemp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        </LinearLayout>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="center_horizontal"
            android:orientation="horizontal">
            <TextView
                android:text="
                                Humedad Suelo :"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
            <TextView
                android:id="@+id/lblhum"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />
        </LinearLayout>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:weightSum="2"
            android:orientation="horizontal">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:text="
                                Humedad Relativa:" />
            <TextView
                android:id="@+id/lbl3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

        </LinearLayout>
    </LinearLayout>
</LinearLayout>
</LinearLayout>
</LinearLayout>
<LinearLayout

```



```

        android:id="@+id/lineardos"
        android:layout_below="@+id/linearuno"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <com.github.mikephil.charting.charts.LineChart
            android:id="@+id/linechart"
            android:layout_width="match_parent"
            android:layout_height="130dp"
            android:layout_below="@+id/li2" />
        <com.github.mikephil.charting.charts.LineChart
            android:id="@+id/linechart2"
            android:layout_width="match_parent"
            android:layout_height="130dp"
            android:layout_below="@+id/li2" />
        <com.github.mikephil.charting.charts.LineChart
            android:id="@+id/linechart3"
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:layout_below="@+id/li2" />
    </LinearLayout>

    <LinearLayout
        android:layout_alignParentBottom="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:gravity="center_horizontal"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btnriego"
            android:text="ACTIVAR RIEGO"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <Button
            android:id="@+id/btnnoriego"
            android:text="DESACTIVAR RIEGO"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <Button
            android:id="@+id/btnatras"
            android:text="ATRAS"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>

</RelativeLayout>

```

LAYOUT ACTIVITY PLANTA2

```

<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

```

```

xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".PLANTA1">

```

```

<LinearLayout
    android:id="@+id/linearuno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:layout_width="49dp"
        android:layout_height="wrap_content"
        android:textColor="@android:color/black"
        android:textSize="15sp"
        android:text="DATOS" />

    <TextView
        android:text="18-Temperatura - 26"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:text="40 - Humedad - 70"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:text="60 - Humedad Relativa - 80"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
<LinearLayout
    android:layout_alignParentRight="true"
    android:layout_toRightOf="@+id/linearuno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:text="
                                INFORMACIÓN"
        android:textColor="@android:color/black"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <TextView
                android:text="
                                Temperatura :"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

            <TextView
                android:id="@+id/lbltemp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content" />

        </LinearLayout>

    </LinearLayout>

```

```

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:text="                Humedad Suelo : "
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/lblhum"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:weightSum="2"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:text="                Humedad Relativa: " />
    <TextView
        android:id="@+id/lblhumerelativa"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:id="@+id/lineardos"
    android:layout_below="@+id/linearuno"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart11"
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:layout_below="@+id/li2" />
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart22"
        android:layout_width="match_parent"
        android:layout_height="130dp"
        android:layout_below="@+id/li2" />
    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/linechart33"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:layout_below="@+id/li2" />
</LinearLayout>

<LinearLayout
    android:layout_alignParentBottom="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"

```

```
    android:gravity="center_horizontal"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btnriego2"
        android:text="ACTIVAR RIEGO"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnnoriego2"
        android:text="DESACTIVAR RIEGO"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnatras2"
        android:text="ATRAS"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

</RelativeLayout>
```

RIEGO AUTOMATICO

//-----PROGRAMA EN EL ARDUINO-----

//INCLUIR LIBRERIA PARA LA LECTURA DE LA TEMPERATURA

#include "DHT.h"

#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2); //(RS,E,D4,D5,D6,D7)

//CONFIGURACION DEL SENSOR DHT11 Y DHT22

#define DHTPIN 9

#define DHTPIN2 10

#define DHTTYPE DHT11

#define DHTTYPE2 DHT22

DHT dht(DHTPIN,DHTTYPE);

DHT dht2(DHTPIN2,DHTTYPE2);

int riego1; //ESTADO DEL RIEGO1

int riego2; //ESTADO DEL RIEGO2

int pinriego1 = 11; //PIN DE SALIDA DEL RIEGO1

int pinriego2 = 12; //PIN DE SALIDA DEL RIEGO2

int contH = 0;

int cont2H = 0;

int sensor2, sensor5;

//DEFINMOS EL SIMBOLO DE °

byte grado[8] =

{

0b00001100,

0b00010010,

0b00010010,

0b00001100,

0b00000000,

```

0b00000000,
0b00000000,
0b00000000
};

int valor; //VARIABLE GLOBAL QE ALMACENA EL VALOR DE LECTURA DEL SENSOR DE HUMEDAD
int valorMaximo = 500;//VARIABLE PARA COLOCAR VALOR MAXIMO
int val;
int val1;

void setup() {
//CONFIGURACION INICIAL
dht.begin();//INICIALIZACION DEL SENSOR dht11
dht2.begin();//INICIALIZACION DEL SENSOR dht22
Serial.begin(115200); //COMUNICACION Y VELOCIDAD DEL MONITOR SERIAL Y ARDUINO

lcd.begin(16, 2);
pinMode(pinriego1,OUTPUT); //DEFINIMOS COMO PIN DE SALIDA
pinMode(pinriego2,OUTPUT); //DEFINIMOS COMO PIN DE SALIDA
lcd.setCursor(5,0); //
lcd.print("invernadero");
lcd.setCursor(5,1);
lcd.print("Maciel");
delay(2000);
lcd.clear();

digitalWrite(pinriego1,HIGH);
digitalWrite(pinriego2,HIGH);
}

```

```

void loop()
{//BUCLE INFINITO REPETITIVO

recibirdata();
funcioncritica();
activarriego();

}

int sensorT() {

//int temp= thermocouple.readCelsius();// lectura del sensor con la libreria

int t = dht.readTemperature();
delay(500);
lcd.createChar(1,grado);
lcd.setCursor(0,0);
lcd.print("T:");
lcd.print(t);
lcd.write(1);
lcd.print("C");
return t;

}

int sensorH(){
int lectura = analogRead(0); //realizamos la lectura del sensor A0
lectura = map(lectura,400,1023,100,0);//mapeamos O CAMBIAMOS LOS VALORES DE 0 A 100%
lcd.setCursor(6,0);

```

```

    lcd.print("H:");
    lcd.print(lectura);
    lcd.print("%");
    return lectura;
}

```

```

int sensorH2(){
    int h2 = dht.readHumidity();//PARA OBTENER LA HUMEDAD RELATIVA DEL SENSOR
    delay(500);
    lcd.setCursor(11,0);
    lcd.print("h:");
    lcd.print(h2);
    lcd.print("%");
    return h2;
}

```

```

int sensor2T() {

    //int temp= thermocouple.readCelsius();// lectura del sensor con la libreria

    int t2 = dht2.readTemperature(); //PARA OBTENER LA TEMERATURA DEL SENSOR
    lcd.createChar(1,grado);
    lcd.setCursor(0,1);
    lcd.print("T:");
    lcd.print(t2);
    lcd.write(1);
    lcd.print("C");
    return t2;
}

```



```

}

int sensor2H(){
    int lectura2 = analogRead(1); //realizamos la lectura del sensor A1

    lectura2 = map(lectura2,400,1023,100,0);//mapeamos O CAMBIAMOS LOS VALORES DE 0 A
100%

    lcd.setCursor(6,1);
    lcd.print("H:");
    lcd.print(lectura2);
    lcd.print("%");
    return lectura2;

}

```

```

int sensor2H2(){
    int h22 = dht2.readHumidity();
    delay(500);
    lcd.setCursor(11,1);
    lcd.print("h:");
    lcd.print(h22);
    lcd.print("%");
    return h22;

}

```

```

void funcioncritica(){
    if(sensor2 <= 10){
        contH++;
        Serial.print(contH);
    }
    if(sensor5 <= 10){
        cont2H++;
    }
}

```

```

        Serial.print(cont2H);
    }
}

void activarriego(){

    if(riego1 == 1){
        digitalWrite(pinriego1,LOW);
        delay(500);
        contH = 0;
    }
    if(riego1 == 0){
        if(contH >= 300){// modo automatico
            digitalWrite(pinriego1,LOW);
            delay(60000);
            contH = 0;
            digitalWrite(pinriego1,HIGH);
            delay(100);
        }
        digitalWrite(pinriego1,HIGH);
        delay(500);
    }
    if(riego2 == 1){
        digitalWrite(pinriego2,LOW);
        delay(500);
        cont2H = 0;
    }

    if(riego2 == 0){
        if(cont2H >= 300){// modo automatico
            digitalWrite(pinriego2,LOW);

```

```
    delay(60000);  
    cont2H = 0;  
    digitalWrite(pinriego2,HIGH);  
    delay(100);  
  }  
  digitalWrite(pinriego2,HIGH);  
  delay(500);  
}  
}
```

```
void recibirdata(){  
  int sensor1 = sensorT();  
  sensor2 = sensorH();  
  int sensor3 = sensorH2();  
  int sensor4 = sensor2T();  
  sensor5 = sensor2H();  
  int sensor6 = sensor2H2();  
  Serial.print("I"); //inicio de trama  
  Serial.print(sensor1);  
  Serial.print(":");  
  Serial.print(sensor2);  
  Serial.print(":");  
  Serial.print(sensor3);  
  Serial.print(":");  
  Serial.print(sensor4);  
  Serial.print(":");  
  Serial.print(sensor5);  
  Serial.print(":");  
  Serial.print(sensor6);  
  Serial.print("F"); //fin de trama
```

```
Serial.flush(); //Esperamos hasta que se envíen los datos
delay(100);
//----Leemos la respuesta del Esclavo-----
if(Serial.find("i"))//esperamos el inicio de trama
{
    riego1=Serial.parseInt();//recibimos valor numérico
    if(Serial.read()==':') //Si el fin de trama es el correcto
    {
        riego2 = Serial.parseInt();
        if(Serial.read()=='f'){

        }

    }

}

}
```