



**ESPE**  
**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**Desarrollo de un sistema software para optimizar la medición y registro histórico de temperatura y humedad en el centro agropecuario “Pilonera Verito” mediante el uso del Internet de las Cosas y Tecnología Móvil.**

Jimbo Iza, Steve Isaac y Martínez Udeo, Luis David

Departamento de Ciencia de la Computación

Carrera de Ingeniería de Software

Trabajo de titulación, previo a la obtención del título de Ingeniero de Software

Ing. Velasco Sánchez, Diego Ismael Msc.

03 de agosto del 2022

Latacunga

## Reporte de verificación de contenido



SW-TESIS-JIMBO-MARTINEZ.pdf

Scanned on: 15:41 July 27, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	371
Words with Minor Changes	319
Paraphrased Words	81
Omitted Words	0



Website | Education | Businesses

A handwritten signature in blue ink, appearing to read "Diego Ismael Msc.", written over a dotted line.

Ing. Velasco Sánchez, Diego Ismael Msc.

C. C.: 0502603855

Director de Tesis



**Departamento de Ciencias de la Computación**

**Carrera de Ingeniería de Software**

**Certificación**

Certifico que el trabajo de titulación: **"Desarrollo de un sistema software para optimizar la medición y registro histórico de temperatura y humedad en el centro agropecuario "Pilonera Verito" mediante el uso del Internet de las Cosas y Tecnología Móvil"** fue realizado por los señores **Jimbo Iza, Steve Isaac y Martínez Udeo, Luis David**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

**Latacunga, 26 de julio del 2022**

**Ing. Velasco Sánchez, Diego Ismael Msc.**

**C. C.: 0502603855**



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**Departamento de Ciencias de la Computación  
Carrera de Ingeniería de Software**

**Responsabilidad de Autoría**

Nosotros, **Jimbo Iza, Steve Isaac**, con cédula de ciudadanía N°1726725318, y **Martinez Udeo, Luis David**, con cédula de ciudadanía N°0504066358, declaramos que el contenido, ideas y criterios del trabajo de titulación: **"Desarrollo de un sistema software para optimizar la medición y registro histórico de temperatura y humedad en el centro agropecuario "Pilonera Verito" mediante el uso del Internet de las Cosas y Tecnología Móvil"** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Latacunga, 26 de julio del 2022**

.....  
**Jimbo Iza, Steve Isaac**

C.C.: 1726725318

.....  
**Martinez Udeo, Luis David**

C.C.: 0504066358



**Departamento de Ciencias de la Computación**

**Carrera de Ingeniería de Software**

**Autorización de Publicación**

Nosotros **Jimbo Iza, Steve Isaac**, con cédula de ciudadanía N°1726725318, y **Martínez Udeo, Luis David**, con cédula de ciudadanía N°0504066358, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Desarrollo de un sistema software para optimizar la medición y registro histórico de temperatura y humedad en el centro agropecuario "Pilonera Verito" mediante el uso del Internet de las Cosas y Tecnología Móvil"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

**Latacunga, 26 de julio del 2022**

.....  
**Jimbo Iza, Steve Isaac**

C.C.: 1726725318

.....  
**Martínez Udeo, Luis David**

C.C.: 0504066358

## Dedicatoria

A mi madre Norma Iza quien desde mis 6 años se hizo cargo de mí, cuidándome y apoyándome en cada intento y momento de mi vida, quien se ha esforzado entregando todo de sí por darme todo a mí. Su dedicación y amor me han enseñado durante todos estos años las mejores lecciones de vida.

A mi padre Luis Jimbo quien me ha dado una importante guía y apoyo con sus constantes palabras y enseñanzas, se ha esforzado constantemente para brindarme su ayuda de diferentes formas.

A mi hermana Andrea Jimbo quien me ha impulsado a ser más independiente y fuerte desde pequeño, en quien confío y es mi confidente, siendo también el ejemplo constante de lo que significa el esfuerzo, la valentía y el no rendirse.

A mi hermano Henry Jimbo quien ha sido un maestro para mí, enseñándome sobre la vida y sobre el mundo que nos rodea con entrañables aventuras y miles anécdotas las cuales se me quedarán grabadas en el corazón.

A mi novia María Guallichico, a quien adoro y amo, que durante todo el desarrollo del proyecto me brindó palabras de aliento, ánimos para seguir y escapadas del estrés. Es mi luz en los momentos más oscuros.

A mi mejor amigo David Seis quien con más de 10 años de amistad me ha demostrado el valor de una verdadera amistad, me ha brindado su apoyo incondicional y hemos compartido tantas cosas.

A mi mejor amigo Luis Martínez quien conocí por azares del destino y se convertiría en un hermano, al cual considero es uno de los mejores programadores y digno de una gran admiración por su capacidad de atreverse a hacer cosas nuevas y enfrentar retos que muchas personas no lo harían. Destacando su gran y buen corazón.

**Steve I. Jimbo I.**

## Dedicatoria

A mi padre Luis Alberto Martínez Chuquitarco, quien con sus conocimientos sobre la vida me ha permitido estar centrado en mis objetivos, quien me ha enseñado que existe una gran satisfacción por el trabajo duro, que el ser fiel a uno mismo, a sus convicciones y deseos, es un gran motor para mantenerse en el camino de la rectitud, que no importa cuán difícil se vea el panorama lo importante es seguir trabajando para conseguir lo que soñamos, no solo para nosotros sino también para nuestra familia.

A mi madre Ana Mercedes Udeo Velasco, quien me enseñó la importancia de la empatía, el pensar también en los demás y en el bien común, que cuando algo nos importa mucho no importa el cansancio o las heridas siempre continuamos, que existen buenas personas en todas partes, que lo más importante es la sinceridad, el mantenerse en paz con uno mismo, y sobre todo a valorar a quienes nos aman, a nuestros amigos y familia.

A mi hermana Jhoana Marisol Martínez Udeo, quien me acompañó en las desventuras de esta etapa, quien compartía conmigo sus experiencias y me animaba a mejorar cada día, quien me enseñó que cuando los problemas nos hagan retroceder siempre es bueno tomarse un descanso reflexionar y seguir adelante.

A mi amiga Odalis Dayanna Valencia Valencia, quien me enseñó la importancia de la amabilidad y el optimismo, quien transmitía alegría a donde iba, y quien me enseñó el valor de formar parte de una sociedad.

Y finalmente a la persona quien me ha acompañado en esta travesía, un gran amigo, hermano, Steve Isaac Jimbo Iza, quien me mostró otros puntos de vista de la vida, otras formas de obrar y sobre todo que confió en mí enseñándome a atreverme a vivir, a equivocarme y mejorar, probar cosas nuevas y nunca dejar de creer en mí mismo.

**Luis D. Martínez. U**

## **Agradecimientos**

Agradezco a la “Fuerza” quien me permitió conocer a Luis Martínez quien se convirtió en mi mejor amigo acompañándome durante toda la Universidad siendo un gran compañero/amigo/hermano y con el cual realizo este trabajo de titulación.

Agradezco a Ing. Diego Velasco, Ing. Ximena López, y Ing. Fabian Montaluisa quienes han sido para mí, docentes que tienen una verdadera vocación por enseñar y me han transmitido grandes conocimientos.

Agradezco al Ingeniero William Panchi quien ha demostrado ser una persona humilde y quien nos ha abierto las puertas de su centro agropecuario para realizar el presente proyecto estando pendiente y cubriendo todas las necesidades que pudimos presentar.

Agradezco a Mónica y Orlando quienes a más de ser los dueños del restaurante donde hemos desayunado y almorzado durante tantos semestres, se han convertido en amigos.

**Steve I. Jimbo. I**

## **Agradecimientos**

Primero que nada, agradezco a todos mis compañeros de carrera, personas sin las que esta travesía habría sido muy aburrida, tuve la gran suerte de pertenecer a un grupo repleto de buenas personas.

A mis profesores por enseñarme todo aquello que debe y no debe hacerse en el mundo profesional, siendo un ejemplo de humildad, paciencia, objetividad y sobre todo dominio de los conocimientos impartidos en clases.

Al Ingeniero William Panchi por su apertura y compromiso con este trabajo, brindando su apoyo tanto de forma personal como profesional, por su amabilidad y paciencia a la hora de instruirnos.

Al Ingeniero Diego Velasco por su apoyo e iniciativa con el desarrollo de este trabajo, que en el rol de tutor nos supo encaminar y brindar su apoyo con conocimientos y experiencias.

**Luis D. Martínez U.**

**ÍNDICE DE CONTENIDO**

<b>Carátula .....</b>	<b>1</b>
<b>Reporte de verificación de contenido.....</b>	<b>2</b>
<b>Certificación .....</b>	<b>3</b>
<b>Responsabilidad de Autoría.....</b>	<b>4</b>
<b>Autorización de Publicación .....</b>	<b>5</b>
<b>Dedicatoria .....</b>	<b>6</b>
<b>Agradecimientos.....</b>	<b>8</b>
<b>Índice de contenido .....</b>	<b>10</b>
<b>Índice de Tablas .....</b>	<b>14</b>
<b>Índice de Figuras .....</b>	<b>16</b>
<b>Resumen.....</b>	<b>19</b>
<b>Abstract .....</b>	<b>20</b>
<b>Capítulo I: Aspectos Generales .....</b>	<b>21</b>
<b>Introducción.....</b>	<b>21</b>
<b>Antecedentes.....</b>	<b>23</b>
<b>Justificación e importancia .....</b>	<b>24</b>
<b>Alcance .....</b>	<b>25</b>
<b>Objetivos.....</b>	<b>27</b>
<b><i>Objetivo General .....</i></b>	<b>27</b>
<b><i>Objetivos Específicos.....</i></b>	<b>27</b>

VARIABLES DE LA INVESTIGACIÓN.....	27
<i>Variable Independiente</i> .....	27
<i>Conceptualización de la variable independiente</i> .....	28
<i>Variable Dependiente</i> .....	28
<i>Indicadores</i> .....	28
Capítulo II: Marco Teórico .....	29
Invernaderos.....	29
<i>Importancia</i> .....	29
<i>Medición Manual</i> .....	31
Temperatura.....	32
Humedad.....	33
IoT.....	34
Protocolo MQTT .....	35
Amazon Web Services (AWS).....	36
Flutter .....	37
Patrón Bloc .....	38
Scrum.....	39
Infraestructura IoT.....	39
<i>ESP32</i> .....	39
<i>Sensor DHT22</i> .....	41
<i>Sistemas de alimentación</i> .....	41

Capítulo III: Desarrollo del sistema software .....	43
<b>Metodología SCRUM .....</b>	<b>43</b>
<b>Fase 01 Inicio.....</b>	<b>43</b>
<i>Visión del proyecto.....</i>	<i>43</i>
<i>Formar el Scrum Team .....</i>	<i>44</i>
<i>Desarrollar épicas.....</i>	<i>44</i>
<b>Fase 02 Planificación y Estimación .....</b>	<b>47</b>
<i>Criterios de aceptación .....</i>	<i>47</i>
<i>Crear Historias de Usuario.....</i>	<i>47</i>
<i>Aprobar, estimar y comprometer historias de usuario .....</i>	<i>51</i>
<i>Generación del Product BackLog.....</i>	<i>52</i>
<i>Crear Tareas.....</i>	<i>53</i>
<i>Estimar Tareas .....</i>	<i>55</i>
<i>Crear sprint backlog .....</i>	<i>59</i>
<b>Fase 03 Implementación .....</b>	<b>63</b>
<i>Crear entregables .....</i>	<i>63</i>
<i>Sprint 1 .....</i>	<i>63</i>
<i>Sprint 2 .....</i>	<i>85</i>
<i>Sprint 3 .....</i>	<i>98</i>
<b>Fase 04 Revisión y Retrospectiva.....</b>	<b>107</b>
<i>Sprint 1 .....</i>	<i>107</i>

<i>Sprint 2</i> .....	108
<i>Sprint 3</i> .....	110
Fase 05 Liberación .....	112
<i>Despliegue de la aplicación en la Play Store</i> .....	112
Capítulo IV: Validación .....	113
Indicador Precisión de datos .....	113
<i>Temperatura</i> .....	114
<i>Humedad</i> .....	119
<i>Interpretación de los resultados</i> .....	123
Indicador Tiempo empleado .....	123
<i>Interpretación de los resultados</i> .....	129
Indicador Costos derivados .....	129
<i>Interpretación de los resultados</i> .....	136
Capítulo V: Conclusiones y Recomendaciones.....	137
Conclusiones.....	137
Recomendaciones.....	139
Bibliografía.....	140
Anexos .....	144

## ÍNDICE DE TABLAS

<b>Tabla 1</b>	<i>Componentes principales del protocolo MQTT</i>	36
<b>Tabla 2</b>	<i>Componentes de la arquitectura de Flutter</i>	37
<b>Tabla 3</b>	<i>Principales características del módulo ESP32</i>	40
<b>Tabla 4</b>	<i>Especificaciones técnicas del adaptador de corriente alterna</i>	42
<b>Tabla 5</b>	<i>Equipo Scrum</i>	44
<b>Tabla 6</b>	<i>Tipos de usuario</i>	46
<b>Tabla 7</b>	<i>Estimación de historias de usuario</i>	51
<b>Tabla 8</b>	<i>Product Backlog</i>	52
<b>Tabla 9</b>	<i>Creación de tareas</i>	53
<b>Tabla 10</b>	<i>Estimación de tareas</i>	55
<b>Tabla 11</b>	<i>Sprint backlog del sprint 1</i>	59
<b>Tabla 12</b>	<i>Sprint backlog del sprint 2</i>	61
<b>Tabla 13</b>	<i>Sprint backlog del sprint 3</i>	62
<b>Tabla 14</b>	<i>Lista de chequeo de los criterios de aceptación del Sprint 1</i>	107
<b>Tabla 15</b>	<i>Lista de chequeo de los criterios de aceptación del Sprint 2</i>	109
<b>Tabla 16</b>	<i>Lista de chequeo de los criterios de aceptación del Sprint 3</i>	110
<b>Tabla 17</b>	<i>Valores de Temperatura de monitorización manual y automatizado</i>	114
<b>Tabla 18</b>	<i>Resultados de la Prueba t obtenidos por Análisis de Datos en Excel</i>	117
<b>Tabla 19</b>	<i>Valores de Temperatura de monitorización manual y automatizado</i>	119
<b>Tabla 20</b>	<i>Resultados de la Prueba t obtenidos por Análisis de Datos en Excel</i>	122
<b>Tabla 21</b>	<i>Valores de tiempo empleado de monitorización manual y automatizado</i>	124
<b>Tabla 22</b>	<i>Valores de Temperatura de monitorización en minutos</i>	125
<b>Tabla 23</b>	<i>Resultados de “Excel” obtenidos para el análisis del tiempo</i>	128
<b>Tabla 24</b>	<i>Costos relacionados con el proceso de monitorización</i>	131
<b>Tabla 25</b>	<i>Costos totales derivados del proceso de monitorización</i>	132

<b>Tabla 26</b> <i>Resultados de “Excel” obtenidos para el análisis de los costos.....</i>	<b>135</b>
--	------------

## ÍNDICE DE FIGURAS

<b>Figura 1</b>	<i>Módulo ESP32</i>	40
<b>Figura 2</b>	<i>Sensor DHT 22</i>	41
<b>Figura 3</b>	<i>Épicas de IoTemp</i>	46
<b>Figura 4</b>	<i>Historia de usuario HU-001</i>	48
<b>Figura 5</b>	<i>Historia de usuario HU-002</i>	48
<b>Figura 6</b>	<i>Historia de usuario HU-003</i>	49
<b>Figura 7</b>	<i>Historia de usuario HU-004</i>	49
<b>Figura 8</b>	<i>Historia de usuario HU-005</i>	50
<b>Figura 9</b>	<i>Historia de usuario HU-006</i>	50
<b>Figura 10</b>	<i>Historia de usuario HU-007</i>	51
<b>Figura 11</b>	<i>Diagrama de contenedores del sistema</i>	63
<b>Figura 12</b>	<i>Instancia creada del servidor en AWS</i>	64
<b>Figura 13</b>	<i>Servicio EMQ funcionando en el servidor</i>	65
<b>Figura 14</b>	<i>Plugin de backend ejecutándose en EMQ</i>	66
<b>Figura 15</b>	<i>Plugin de autenticación ejecutándose en EMQ</i>	67
<b>Figura 16</b>	<i>Vista aérea del invernadero</i>	68
<b>Figura 17</b>	<i>Infraestructura IoT planificada para el invernadero</i>	69
<b>Figura 18</b>	<i>Cableado físico para la comunicación de los sensores</i>	70
<b>Figura 19</b>	<i>Soldadura realizada del cable UTP con los sensores DHT22</i>	71
<b>Figura 20</b>	<i>Instalación finalizada del dispositivo IoT</i>	72
<b>Figura 21</b>	<i>Código del módulo ESP 32 para realizar mediciones</i>	73
<b>Figura 22</b>	<i>Código del módulo ESP 32 para conectarse al servidor</i>	74
<b>Figura 23</b>	<i>Parámetros de configuración del broker EMQ</i>	75
<b>Figura 24</b>	<i>Envío de datos desde el dispositivo IoT hacia el servidor</i>	76
<b>Figura 25</b>	<i>Diseño de la pantalla resumen y el módulo de temperatura</i>	77

<b>Figura 26</b>	<i>Implementación del módulo de temperatura</i> .....	78
<b>Figura 27</b>	<i>Estructura inicial del proyecto según el patrón bloc</i> .....	79
<b>Figura 28</b>	<i>Creación de subcomponentes del patrón bloc</i> .....	80
<b>Figura 29</b>	<i>Recepción de datos de temperatura</i> .....	81
<b>Figura 30</b>	<i>Diseño del módulo de humedad</i> .....	82
<b>Figura 31</b>	<i>Implementación del módulo de humedad</i> .....	83
<b>Figura 32</b>	<i>Recepción de datos de humedad</i> .....	84
<b>Figura 33</b>	<i>Directorio de la API bajo Slim Framework</i> .....	85
<b>Figura 34</b>	<i>Código de la API para funcionalidad de Usuarios</i> .....	86
<b>Figura 35</b>	<i>Diseño de la pantalla de inicio de sesión</i> .....	87
<b>Figura 36</b>	<i>Implementación del inicio de sesión</i> .....	88
<b>Figura 37</b>	<i>Estructuración del proyecto para el inicio de sesión</i> .....	89
<b>Figura 38</b>	<i>Diagrama de casos de uso del sistema</i> .....	90
<b>Figura 39</b>	<i>Menú de la aplicación</i> .....	91
<b>Figura 40</b>	<i>Código de la API para manejo de los datos históricos</i> .....	92
<b>Figura 41</b>	<i>Diseño de la pantalla estadística</i> .....	93
<b>Figura 42</b>	<i>Implementación de la pantalla de gráfico</i> .....	94
<b>Figura 43</b>	<i>Implementación del selector de fechas</i> .....	95
<b>Figura 44</b>	<i>Recepción de datos históricos</i> .....	96
<b>Figura 45</b>	<i>Despliegue de la información del servidor en el gráfico de líneas</i> .....	97
<b>Figura 46</b>	<i>Código de la API para almacenamiento de la configuración de los rangos normales</i> .....	98
<b>Figura 47</b>	<i>Diseño de la pantalla de configuración</i> .....	99
<b>Figura 48</b>	<i>Implementación de la pantalla de configuración</i> .....	100
<b>Figura 49</b>	<i>Datos de rangos obtenidos del servidor</i> .....	101
<b>Figura 50</b>	<i>Almacenamiento de intervalos</i> .....	102

<b>Figura 51</b>	<i>Script de escucha y alerta para anomalías en rangos</i>	103
<b>Figura 52</b>	<i>Panel de cloud messaging en firebase console</i>	104
<b>Figura 53</b>	<i>Recepción de alerta con firebase</i>	105
<b>Figura 54</b>	<i>Configuración de una alerta en firebase</i>	106
<b>Figura 55</b>	<i>Burndown chart del sprint 1</i>	108
<b>Figura 56</b>	<i>Burndown chart del sprint 2</i>	109
<b>Figura 57</b>	<i>Burndown chart del sprint 3</i>	111
<b>Figura 58</b>	<i>Envío de la aplicación a la Play Store</i>	112
<b>Figura 59</b>	<i>Distribución T-Student para pruebas de dos colas</i>	116
<b>Figura 60</b>	<i>Gráfica T Student para la variable de temperatura</i>	118
<b>Figura 61</b>	<i>Distribución T-Student para pruebas de dos colas</i>	121
<b>Figura 62</b>	<i>Gráfica T Student para la variable de humedad</i>	122
<b>Figura 63</b>	<i>Distribución T Student para la medición del tiempo</i>	127
<b>Figura 64</b>	<i>Gráfica T Student para el análisis del tiempo empleado</i>	128
<b>Figura 65</b>	<i>Distribución T Student para el análisis de los costos</i>	134
<b>Figura 66</b>	<i>Gráfica T Student para el análisis de costos derivados</i>	135

## Resumen

Siendo el Ecuador un país eminentemente agrícola, la necesidad de invertir en el mejoramiento de la producción se ha convertido en un reto debido a la necesidad de soluciones eficientes y económicas por lo que el presente proyecto de titulación está enfocado en el desarrollo de un sistema software basado en dispositivos IoT destinado al proceso de monitorización de variables ambientales (temperatura y humedad relativa) del invernadero del Centro Agropecuario “Pilonera Verito” que permita optimizar los tiempos y costos relacionados a dicho proceso. Para ello se ha desarrollado un conjunto de soluciones en distintas áreas del desarrollo de software las cuales cumplen con las necesidades de recopilación, almacenamiento, visualización y análisis de datos correspondientes a las tareas realizadas de forma manual en el establecimiento. Además, mediante el desarrollo del proyecto bajo un marco de trabajo ágil se realizaron entregas de valor al establecimiento desde etapas tempranas del proyecto, pudiendo de esta manera ofrecer una aplicación completamente adaptada a las necesidades del responsable del cuidado de los cultivos, mismas que fueron abordadas mediante un sistema software que ofrece: monitorización a tiempo real, datos históricos con gráficos estadísticos que ayudan a la toma de decisiones, un amplio uso de la aplicación sin restricciones geográficas; haciendo de la solución una herramienta accesible por sus bajos costos y enfocada en la optimización de tiempos.

*Palabras Clave:* Dispositivo IoT, Amazon Web Services, Protocolo MQTT, Desarrollo Ágil, Aplicación Móvil

### **Abstract**

Being Ecuador an eminently agricultural country, the need to invest in the improvement of production has become a challenge because of the need for efficient and economical solutions so this degree project is focused on the development of a software system based on IoT devices for the monitoring process of environmental variables (temperature and relative humidity) of the greenhouse of the Agricultural Center "Pilonera Verito" to optimize the time and costs related to this process. For this purpose, a group of solutions has been developed in different areas of software development to satisfy the needs of collecting, storing, visualizing and analyzing data corresponding to the tasks performed manually in the establishment. In addition, through the development of the project under an agile framework, value was delivered to the establishment from early stages of the project, offering an application completely adapted to the needs of the person responsible for the care of the crops, which were addressed through a software system that offers: real-time monitoring, historical data with statistical graphs that help decision making, a wide use of the application without geographical restrictions; making the solution an accessible tool for its low costs and focused on time optimization.

*Key Words:* IoT Device, Amazon Web Services, MQTT Protocol, Agile Development.  
Mobile App.

## Capítulo I

### Aspectos Generales

#### Introducción

La automatización de procesos en la última década ha contribuido con el desarrollo de diferentes industrias, ya sea mediante la implementación de soluciones complementarias o soluciones que sustituyen tareas manuales y tediosas que solo complican y aumentan los costos de los diferentes procesos. (Chasiluisa, Basantes, 2020), afirma que “actualmente en el área de la agricultura los procesos automatizados han permitido una mejoría en el cultivo en diferentes estaciones del año, así como la optimización de recursos y la mejora del producto” (p.87).

(Quezada, 2017) afirma que “en países como Ecuador, el acceso a recursos tecnológicos aún es una limitante, debido a los altos costos y desconocimiento de soluciones adecuadas disponibles en el mercado” (p.1). En ese aspecto, pequeños y medianos negocios basados en la agricultura aún se basan en un flujo de trabajo manual, o han empezado a implantar sistemas automatizados los cuales tienden a ser costosos y de difícil mantenimiento.

(Meléndez, 2019) afirma que “El cultivo bajo invernaderos ha permitido tener un control sobre los parámetros físicos involucrados en el crecimiento de los productos, ampliando su variabilidad y su calidad de cara al consumo masivo” (p.2). Siendo el Ecuador un país eminentemente agrícola, la necesidad de invertir en el mejoramiento de la producción, tanto en proceso como en calidad, se vuelve vital para mantenerse en un mercado competitivo, ya no solo a nivel nacional sino internacional.

Esto deriva en la necesidad de intentar mantener un control sobre la producción de los productos agrícolas, el primer paso de este control son los invernaderos donde hasta cierto punto se controlan parámetros físicos para evitar la influencia del ambiente. El segundo paso es la monitorización de los parámetros físicos dentro del invernadero, esto es de suma importancia ya que hay que mantener parámetros aceptables dependiendo del producto que esté creciendo, si estos sobrepasan límites tanto de máximo y mínimo aceptable puede influir en la producción desde enfermedades en las plantas hasta perderlas por completo.

(Meléndez, 2019) afirma que “gran parte de las empresas o agricultores aún realizan estas tareas de forma manual, ya sea mediante sistemas de riego con mangueras, monitoreando variables ambientales en base a horarios, ampliando el esfuerzo de trabajo y el margen de error al momento de mantener un ambiente ideal para cada planta” (p. 2). Al ser pequeñas empresas no cuentan con el capital suficiente para implementar soluciones profesionales y especializadas para la monitorización, sin embargo, esto no impide que lo realicen, mediante un proceso manual donde involucra a trabajadores y medidores digitales de temperatura y humedad relativa.

(Ramírez, Ruales, 2021) afirman que “actualmente el sector agrícola en entornos andinos presenta impacto del cambio climático, variación ambiental y alteración de suelo, motivo por el cual el internet de las cosas, tecnologías móviles, sensores y procesamiento de datos se convierten en soluciones ideales para la monitorización, control y obtención de datos de cultivos” (p.3).

## Antecedentes

(Sanchez, 2020) afirma que “en el Ecuador el sector agrícola es el cuarto sector económico más importante para el país” (p.4), dándole a este sector y sus principales negocios como los invernaderos un énfasis en mantenerse como una principal fuente económica para el país y a su vez como el principal sector para cubrir la demanda de alimentos dentro del país.

“Los invernaderos son lugares cerrados que se destinan al cultivo de plantas de diferente índole, ya que al presentar un microclima interno es propicio para la producción fuera de época o un crecimiento acelerado” (INSST, 2019, p.3).

“El internet de las cosas o IoT se considera como la conexión dentro de una red de diversos objetos físicos los cuales tiene la capacidad de auto gestionarse, compartir información o datos con el fin de proporcionar servicios inteligentes” (Hedge, 2010, p.2).

“El internet de las cosas (IoT) dentro del contexto de una empresa o negocio que aporte a la gestión en la cadena de suministro propone como objetivo principal, la obtención de información sin ayuda o intervención humana para posterior uso de los encargados e interesados dentro del proceso para toma de decisiones o intervención para un correcto control del mismo” (Pérez, 2019, p.1).

Los dispositivos IoT destinados a invernaderos llegan a ser aquellos con el objetivo de recopilar información del ambiente a través de la interacción de sensores con el mundo real, esta interacción permite controlar de mejor manera el microclima presente en el invernadero lo cual aporta a la producción de las plantas dándoles las condiciones ambientales propicias.

“La temperatura y la humedad relativa son dos de los más importantes factores ambientales ya que son la base sobre la cual un invernadero forma su microclima por lo que es de suma importancia un constante monitoreo ya que anomalías en estos factores provocan alteraciones en la producción” (INSST, 2019, p.3).

### **Justificación e importancia**

(Sanchez, 2020) afirma que “en el Ecuador el sector agrícola aporta al PIB nacional entre un 9,5 a 9,8% del total, lo que significa un aproximado de \$9.626.014 miles de dólares, convirtiéndose así en el cuarto sector económico más importante para el país” (p.1). Dada esta importancia para el país se requieren soluciones tecnológicas que impacten positivamente a la producción.

Al ser el sector agrícola uno de los más importantes en el país ya no solo por su papel dentro del PIB, sino porque es uno de los sectores que mantiene el estilo de vida de muchas familias ecuatorianas al ser la productora de alimentos de primera necesidad, “es importante mantener un mercado competente tanto en producción de la cantidad y calidad de productos para mantener así la demanda de las familias ecuatorianas” (Sanchez, 2020, p.2).

“IoT permitirá la automatización de diferentes procesos de producción presentes en la cadena de suministro de una organización” (Hedge, 2010, p.1), en nuestro caso el enfoque hacia la agricultura será una herramienta para preservar y mantener el capital más importante de un invernadero que son las plantas, y por ende los productos de las mismas que serán sinónimo de exportación o consumo interno, pero siempre siendo la base de la economía o alimentación del Ecuador.

La implementación de IoT dentro de la solución software será una opción que apoye al proceso de producción de un invernadero siendo la monitorización uno de los procesos fundamentales, el cual al ser automatizado se optimizará el tiempo y los costos derivados del mismo, aportando también en que las personas encargadas del invernadero podrán tener un rápido accionar frente a alteraciones internas de temperatura y humedad relativa lo que preservará su capital.

Al buscar ser una solución eficiente y económica, el desarrollo de nuestros propios componentes es fundamental para no depender de terceros que signifiquen un costo y por ende una mayor inversión para la implementación del sistema software, además, que al controlar los diferentes componentes lo podemos optimizar para centrarnos en ser eficientes buscando el objetivo de ser una solución ligera y económica para la agricultura.

Además, nuestra solución abarcará las principales áreas de desarrollo software ya que consta de 4 partes principales: la primera será el servidor, la segunda será una API, la tercera será una aplicación móvil y por último la cuarta es una infraestructura IoT. El desarrollo e implementación de estas partes será un esfuerzo conjunto por demostrar los conocimientos adquiridos a lo largo de la carrera y siendo un software ya en producción.

### **Alcance**

Se realizó la implementación de un sistema software basado en una infraestructura IoT dentro del Centro Agropecuario “Pilonera Verito” de la ciudad de Latacunga, este software está enfocado en la monitorización de parámetros físicos como lo son la humedad relativa y temperatura propia del invernadero.

La solución implementada consiste en 4 partes: la primera será el servidor, la segunda será una API, la tercera será una aplicación móvil y por último la cuarta es una infraestructura IoT.

La infraestructura IoT cuenta con 6 sensores extendidos y ubicados estratégicamente dentro de las diferentes áreas del invernadero donde cada punto de sensores es encargado de medir la temperatura y humedad relativa ubicadas a nivel del suelo y del aire.

El servidor será el punto de comunicación que bajo el protocolo MQTT podrá intercambiar la información de los distintos clientes IoT que conforman la estructura del sistema software. El cliente IoT más importante será el dispositivo que maneja la comunicación de los sensores, la transforma y la envía hacia el servidor.

Los otros clientes consisten en dispositivos móviles con una aplicación enfocada en la recepción de los datos, desplegando la información de manera resumida y también detallada mediante gráficos. La aplicación móvil permite, mediante la configuración de rangos mínimos y máximos para cada sensor, la recepción de alertas en caso de medidas anormales captadas.

De esta manera se obtuvo información acerca del diferencial del rendimiento de trabajo que se presentaba antes y después de la colocación de los sensores, ya que se lo realizaba de forma manual por los trabajadores del centro agropecuario.

## Objetivos

### ***Objetivo General***

- Desarrollar un sistema software de monitorización mediante el uso de IoT que optimice los tiempos y costos derivados de la lectura de las variables ambientales de temperatura y humedad en el centro agropecuario “Pilonera Verito”.

### ***Objetivos Específicos***

- Elaborar el marco teórico sobre las soluciones software que existen en la monitorización de variables ambientales en invernaderos y centros agropecuarios mediante el uso del internet de las cosas (IoT).
- Diseñar la infraestructura IoT necesaria para el correcto funcionamiento y comunicación del sistema software.
- Desarrollar el sistema software móvil mediante el uso de la tecnología Flutter y patrón block.
- Desarrollar el sistema software bajo la metodología de trabajo SCRUM
- Desplegar e implementar el sistema software en el centro agropecuario “Pilonera Verito”.
- Probar y validar el sistema software mediante pruebas comparativas del tiempo empleado en las lecturas.

## Variables de la Investigación

### ***Variable Independiente***

Sistema software de monitorización basada en IoT

Tiempo y costo

**Conceptualización de la variable independiente**

Tiempo: Cantidad de minutos utilizada en el monitoreo de las variables ambientales (temperatura y humedad relativa) dentro del invernadero

Costo: Cantidad de dinero en dólares que corresponde a la inversión del centro agropecuario en el proceso de monitoreo de las variables ambientales.

**Variable Dependiente**

Optimizar: relación de costo, beneficio y tiempo de las tareas de lectura de las variables ambientales de temperatura y humedad realizadas

Optimiza los tiempos derivados de la lectura de las variables ambientales de temperatura y humedad en el centro agropecuario "Pilonera Verito".

**Indicadores**

- Precisión de los datos monitorizados
- Tiempo empleado para la monitorización
- Costos asociados en el proceso de monitorización

## Capítulo II

### Marco Teórico

#### **Invernaderos**

Se entiende por invernadero un lugar cerrado, estático y accesible a pie, dotado habitualmente de una cubierta exterior translúcida de vidrio o de plástico, dentro del cual se puede obtener un microclima mediante el control de la temperatura, de la humedad y de otros factores ambientales, además, se pueden proporcionar sistemas automáticos de riego y ventilación, lo cual se utiliza para la producción y de cultivos de forma controlada (INSST, 2019).

Esto presenta una serie de beneficios tales como la precocidad en el crecimiento de los frutos, permitir la producción fuera de época, ahorro de agua y fertilizantes, facilitar el control de insectos y plagas, etc. (INSST, 2019).

El crecimiento de la población ha provocado un aumento en la cantidad de recursos necesarios para abastecer las necesidades en todas las partes del mundo. La industria madre de la producción de alimentos, la agricultura, se ha visto obligada a mejorar, no solo sus herramientas sino también sus procesos, en busca de un mayor rendimiento, a la par de un mayor volumen (Guerrero-Ibañez et al., 2017).

#### ***Importancia***

Las necesidades derivadas de un crecimiento poblacional, han provocado el aumento de la demanda de productos de gran calidad, es así como la actividad agrícola bajo invernadero ha tomado protagonismo.

Las actividades agrícolas bajo invernadero han permitido alargar el ciclo de cultivo, donde mediante entornos controlados los cultivos son capaces de soportar las épocas más difíciles del año, obteniendo así productos de mayor valor y además permitiendo al agricultor diseñar estrategias de inversión, así como para el cuidado del ambiente ideal, reflejado posteriormente en procesos de alto rendimiento y calidad (Pacheco Cueva & Herrera Albarracin, 2021).

Actualmente las actividades agrícolas en muchos invernaderos del sector rural ya sean pequeños o medianos se llevan a cabo de forma manual, los agricultores en este sector deben desplazarse largas distancias para poder monitorear el estado de sus cultivos (Eduardo, 2021), derivando en tareas que reducen la productividad del negocio y aumentan la probabilidad de que existan pérdidas de productos que dependan de la precisión o el cuidado continuo.

La importancia del invernadero radica en la obtención de productos de gran calidad sin importar la época del año, para que esto ocurra es necesario tener un control sobre la composición del invernadero, las estructuras que lo conforman, persianas, aspersores, donde los cultivos requieren condiciones mínimas y máximas para un correcto desarrollo y la prevención de plagas (Cisneros & Mosquera, 2019).

### ***Medición Manual***

En Ecuador gran parte de empresas pequeñas y medianas dedicadas a la agricultura se caracterizan por el uso de herramientas tradicionales, ya sea tanto para el cultivo como para el control de sus invernaderos, dígase trasladarse a cada una de las zonas del invernadero para conocer las variables ambientales de dicha zona, esto implica que en mayor medida las tareas están destinadas a ser completadas por trabajadores, estando su rendimiento en función de su condición física, algo que impide una optimización de recursos y tiempo (Vásquez & Eduardo, 2019).

El uso de herramientas tradicionales provoca que las actividades de monitoreo dentro del invernadero consuman más tiempo de lo que deberían, derivando además en una recolección de datos poco precisa, algo que influye en la toma de decisiones en cuanto a la estructura del invernadero, además el rendimiento de los responsables de realizar el monitoreo podría mermar en función del área total a cubrir, así como la organización de los cultivos, siendo también importante tomar en cuenta el clima y la mano de obra disponible para realizar dichas tareas (Vásquez & Eduardo, 2019).

El entorno controlado de un invernadero obedece a diferentes factores, entre ellos las variables ambientales, entre ellas la humedad y temperatura relativas. Dichas variables inciden directamente en el desarrollo de los cultivos y varían en función de la estructura del invernadero, persianas, aspersores.

Existen pérdidas en la producción de zonas agrícolas en el Ecuador debido a procesos de monitoreo donde interviene el ser humano, provocando que no se lleve a cabo un proceso de medición con valores precisos (Rivas & Benavides, 2020). El monitoreo en invernaderos pequeños se ve complicado por el tamaño del área, cuanto más grande sea el área un control humano pierde precisión, aumenta el esfuerzo necesario y los recursos para realizar la toma de datos.

El manejo de los invernaderos abarca la manipulación del entorno en el que se desenvuelven los cultivos, ya sea mediante la apertura o cierre de persianas, o el control de sistemas de riego, herramientas que afectan directamente el comportamiento de variables ambientales tales como temperatura y humedad relativa. La manipulación de estas herramientas surge de la toma de decisiones en base a datos recogidos manualmente, algo que cuando existan condiciones normales podría no verse reflejado en el rendimiento de la producción, pero que en mayor medida consume el tiempo de otras tareas y no se sustenta en datos precisos tomados en el tiempo indicado (Torres, 2017).

### **Temperatura**

La temperatura es una magnitud física que indica la energía interna de un cuerpo, dicha energía se expresa coloquialmente en términos de calor y frío, siendo el calor el término asociado con una temperatura más alta, mientras que el frío se asocia con una temperatura más baja. Sus unidades de medida de temperatura son los grados Celsius ( $^{\circ}\text{C}$ ), los grados Fahrenheit ( $^{\circ}\text{F}$ ) y los grados Kelvin (K) y son medidos por un termómetro.

La temperatura es una variable ambiental que incide en el crecimiento de la planta, así como el tiempo de crecimiento y tamaño de los productos. (Fischer, 2014 como se citó en ANAYA, 2019) afirma: “La temperatura de la zona radical tiene efectos marcados sobre el crecimiento y desarrollo de la planta. En general, el crecimiento y desarrollo de la planta aumenta al incrementar la temperatura hasta un máximo entre 25° y 36° C dependiendo la especie. En consecuencia, un excesivo registro de temperatura provocaría un proceso de crecimiento más rápido, pero en ocasiones desfavorable ya que implicaría un aumento de energía necesaria en etapas tempranas, restándole de etapas tardías, resultando en productos más pequeños (Canna Research, 2003 como se citó en ANAYA, 2019).

La temperatura puede variar conforme transcurre el día, existen mediciones tanto diurnas como nocturnas las cuales pueden llegar a variar bruscamente, ocasionando saltos térmicos los cuales pueden derivar en desequilibrios vegetativos (Cisneros & Mosquera, 2019). Es entonces que es prioritario el monitoreo permanente, con el fin de conocer los datos clave para el mantenimiento del invernadero.

### **Humedad**

La humedad relativa se expresa en porcentaje y representa la masa de vapor de agua que hay en un determinado volumen de aire, y la que este tendría si estuviera saturado. Su rango parte de 0% a 100% donde mientras más próximo esté al 100% es indicativo de que más húmedo está el ambiente.

La humedad es una variable ambiental que incide directamente en el proceso de fotosíntesis, variando en función del agua presente alrededor de la planta. (ANAYA, 2019) sostiene: “En el caso de los invernaderos, una buena humedad alrededor de la planta es todavía más importante que en otras condiciones, porque la planta únicamente puede absorber una pequeña cantidad de agua y, por tanto, evapora menos agua. Si la planta pierde demasiada agua, las estomas se cerrarán, lo cual provocará que la fotosíntesis se frene.”

Al igual que con la temperatura, las humedades relativas combinadas con altas temperaturas pueden incidir negativamente en el desarrollo de las plantas, (Cisneros & Mosquera, 2019) sostiene: “La coincidencia de altas temperaturas y baja humedad relativa puede ocasionar la caída de flores y de frutos recién cuajados”.

## **IoT**

“El término Internet de las Cosas o IoT por sus siglas en inglés fue empleado por primera vez por Kevin Ashton en 1999 teniendo como objetivo principal la promoción de información de sentido computacional sin ayuda o intervención humana, recopilando información del ambiente e interactuando con el mundo físico (actuando, mandando y controlando)” (Perez, 2019, p.1).

“IoT es considerado como la conexión en red de objetos físicos o dispositivos en una red abierta y llena de objetos inteligentes que tiene la capacidad de auto gestionarse, compartir información, datos y recursos, reaccionar y actuar frente a situaciones y cambios en el medio ambiente” (Hegde, 2010, p.2).

“Se compone de seis elementos: objetos o cosas, contexto, ruta de comunicación, servicio, relaciones y temporalidad; de esta forma, IoT busca conectar algún lugar, comunicar alguna red, recolectar algún dato, servir a algún objeto, procesar alguna información, e integrar cosas” (Kaur, 2016, p.3).

“Una definición centrada en la transmisión de datos muestra a IoT como una red que interconecta objetos físicos ordinarios con las direcciones identificables para proporcionar servicios inteligentes” (Ma, 2011, p.2).

### **Protocolo MQTT**

“MQTT es un protocolo de comunicación enfocado en la conectividad M2M (Machine to Machine), hace referencia a la tecnología que permite a los dispositivos la comunicación entre ellos de forma inalámbrica. Es así como la conectividad M2M se considera como una de las partes fundamentales del Internet de las cosas (IOT)” (Ditel, 2020, p.1.).

(Guash y Calleja, 2019) describe que “El MQTT es óptimo para redes de bajo ancho de banda, poco fiables, alta latencia y dispositivos restringidos. Por tanto, se basa en minimizar los requerimientos de recursos de los dispositivos conectados a la red, pero siempre manteniendo la máxima seguridad entre ellos, sin dejar que un dispositivo externo se pueda conectar con facilidad.” (p.12).

Al ser el protocolo MQTT de código abierto su uso y desarrollo está abierto a la comunidad por lo que son cada vez más los dispositivos IoT que utilizan este protocolo para su comunicación y funcionamiento. En la Tabla 1 se muestran los componentes principales dentro del protocolo MQTT.

**Tabla 1***Componentes principales del protocolo MQTT*

<b>Componente</b>	<b>Definición</b>
Broker	Es el servidor que comunicará a los diferentes clientes IoT y permitirá el intercambio de mensajes
Cliente	Son los diferentes dispositivos IoT que se comunican al broker
Publicar	Es la acción de un cliente al enviar un mensaje hacia el broker bajo un tópico específico
Subscripción	Es la acción que puede realizar un cliente para recibir los mensajes que existan dentro de un tópico en específico
Tópico	Es la etiqueta que se le da al publicar un mensaje para que exista una diferenciación para evitar que clientes reciban mensajes a los que no están suscritos.

*Nota.* La tabla muestra las definiciones de los diferentes componentes que interactúan en la comunicación mediante el protocolo MQTT. Tomado de Galvan 2021, p.1

### **Amazon Web Services (AWS)**

Amazon Web Services o por su abreviatura AWS, es una colección de servicios web que en conjunto forman una plataforma de computación en la nube. Es una de las más grandes ofertas a nivel internacional y una de las más importantes de entre las diversas plataformas de computación en la nube compitiendo así directamente con otras plataformas de empresas como Microsoft Azure. Google Cloud Platform e IBM Cloud. (Amazon, 2011).

Presente en 18 regiones geográficas distribuidas en los diferentes continentes del mundo, AWS otorga a sus usuarios una capa gratuita donde desarrolladores, programadores o personas afines a las TI, son capaces de ejecutar proyectos pequeños y medianos siempre respetando los límites de capa.

AWS se convierte la mejor opción para la implementación de servidores ya que otorga en su capa gratuita diferentes imágenes de sistemas operativos tanto libres como de pago además de ser las características de hardware lo necesario para llevar a cabo el presente proyecto.

## Flutter

“Flutter es un kit de desarrollo software de código fuente abierto creado por Google. Es utilizado actualmente en el desarrollo de aplicaciones para Android, iOS y Web, así como método primario para crear aplicaciones para Google Fuchsia. El proceso utilizado es la compilación del código en lenguaje Dart a código nativo de cada plataforma” (Menacho, 2020, p.2).

Este framework es de estilo reactivo y permite a los desarrolladores reducir tiempos de desarrollo, mejorar la experiencia nativa y mejorar la renderización de la interfaz. Como principal atractivo de este framework es la capacidad que tiene para ser multiplataforma ya que con un solo proyecto desarrollado en Dart se ajusta a las plataformas móviles existentes y web.

En la Tabla 2 se puede observar los componentes que son parte de la arquitectura de Flutter.

**Tabla 2**

*Componentes de la arquitectura de Flutter*

<b>Componente</b>	<b>Definición</b>
Plataforma Dart	Para dispositivos móviles se realiza compilación AOT que permite que Flutter tenga un rendimiento elevado. Además, permite hot reload durante el desarrollo, es decir, permite realizar cambios y ver su comportamiento sin necesidad de volver a compilar toda la aplicación.

Componente	Definición
Motor de Flutter	Escrito en C++ y con soporte de renderizado de bajo nivel gracias a las librerías de gráficos de Skia de Google. Interactúa con el SDK específico de cada plataforma para implementar gráficos, entrada y salida de ficheros y de red, así como el tiempo de ejecución de Dart.
Widgets	Específicos para cada plataforma, concretamente widgets de Material Design que implementan el lenguaje de diseño de Google y widgets Cupertino que implementan las guías de interfaz humana de iOS de Apple. Además, existe la posibilidad de crear tus propios widgets ya que en Flutter todo es un widget.

*Nota.* La tabla muestra las definiciones de los diferentes componentes que son parte de la arquitectura de Flutter. Tomado de Menacho 2020, p.2

### **Patrón Bloc**

Siglas de “Business Logic Component”, es un patrón de diseño que agrupa la lógica del negocio, estado de la aplicación y los adapta a la vista. Este patrón basa su funcionamiento en la emisión y recepción de eventos, los cuales inciden directamente en el estado de la aplicación. Puede ser utilizado para cada uno de los componentes que conforman el sistema siempre y cuando estos tengan la suficiente complejidad (Ferrer, 2022).

El objetivo del patrón Bloc es extraer completamente la lógica de negocio, separándola de las vistas y concentrándola en clases llamadas blocs, las cuales no deben tener dependencias de librería, o en caso de necesitarlas estas deberán ser inyectadas. Estas clases centralizan la actualización de los estados del aplicativo, procesando la información en función de las necesidades de la vista, aumentando la escalabilidad (Sánchez, 2020).

## **Scrum**

Scrum es un marco de trabajo utilizado para la gestión de proyectos, este permite a los equipos generar soluciones de valor en tiempos relativamente cortos, permitiendo crear productos adaptables de forma evolutiva (Schwaber & Sutherland, 2020). En el desarrollo de software puede utilizarse para un desarrollo incierto, en el cual los requerimientos iniciales no están completamente claros, y tan solo se tenga una idea generalizada de lo que se espera del producto final.

Este marco de trabajo está enfocado en los equipos, donde cada miembro posee las habilidades para realizar el trabajo, pudiendo compartir sus conocimientos con los demás integrantes. En Scrum los procesos y resultados son transparentes para todo el equipo, conformado además de los desarrolladores, por los interesados en que el proyecto salga adelante (Schwaber & Sutherland, 2020).

## **Infraestructura IoT**

### ***ESP32***

Los módulos ESP32 como se puede observar en la Figura 1 pertenecen a los microcontroladores que son manufacturados por la empresa Espressif Systems, su particularidad es la presencia de tecnología Wifi y Bluetooth integrados de forma dual. Estos módulos pertenecen a la serie de SoC (System on chip) que se caracterizan por ser módulos de bajo costo y bajo consumo energético (Guerra, 2021).

**Figura 1***Módulo ESP32*

*Nota.* Tomado de Guerra 2021.

En la Tabla 3 se puede observar las principales características del módulo ESP32

**Tabla 3***Principales características del módulo ESP32*

<b>Característica</b>	<b>ESP32</b>
Microprocesador	Xtensa Dual-Core 32-bit LX6 con 600 DMIPS
Wi-Fi (802.11 b / g / n)	HT40
Bluetooth	Bluetooth 4.2 y BLE
Frecuencia de operación	160 MHz
SRAM	448 KB
Flash	520 KB
GPIO	34
PWM (hardware)	No posee
PWM (software)	16 canales
Interfaz MAC Ethernet	Sí
Temperatura de trabajo	-40°C to 125°C

*Nota.* La tabla muestra las características del módulo ESP32. Tomado de Guerra 2021.

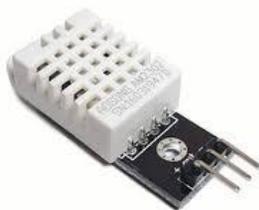
### **Sensor DHT22**

El sensor DHT22 es un componente electrónico que será utilizado para medir la temperatura y la humedad relativa presente en el invernadero. Este sensor se caracteriza por ser un sensor digital de buen rendimiento y bajo costo, integrando tanto un sensor capacitivo para la medición de la humedad y un termistor para la medición de temperatura.

El sensor DHT22 como se puede observar en la Figura 2 presenta tres pines de los cuales dos son de alimentación y el tercero es la salida digital de datos. A nivel de software este sensor dispone de librerías para analizar los datos previstos por el pin de salida y obtener el valor de humedad y temperatura.

### **Figura 2**

*Sensor DHT 22*



*Nota.* Tomado de Guerra 2021

### **Sistemas de alimentación**

La alimentación del módulo se dará por medio de un adaptador de corriente alterna que se conectará al puerto micro-usb otorgándole un voltaje de 5V, posteriormente se utilizará el pin de 3.3v del módulo ESP32 para alimentar a los 6 sensores que tendremos colocados en el invernadero.

La Tabla 4 muestra las especificaciones técnicas del adaptador de corriente alterna.

**Tabla 4**

*Especificaciones técnicas del adaptador de corriente alterna*

<b>Parámetro</b>	<b>Valor</b>
Modelo	XJS-0060
INPUT	100-240V AC 0.5A Max
OUTPUT	5V 2A

*Nota.* Tomado de las especificaciones presentes en la caja del adaptador de corriente alterna.

## Capítulo III

### Desarrollo del sistema software

#### Metodología SCRUM

Es una metodología ágil enfocada en la planificación y ejecución de proyectos de todo tipo, siendo muy utilizada en el desarrollo de software con un enfoque iterativo e incremental. Consta de ciclos de desarrollo (Sprint) así como de productos (entregables) al inicio y final de cada uno de ellos. El proceso de desarrollo bajo este marco de trabajo varía en función de cada proyecto, y es ampliamente usado cuando no se tienen bien definidos los requisitos del sistema y este tiende a evolucionar (The Scrum Primer, 2009 como se citó en Bahit, 2012).

El presente proyecto ha sido desarrollado bajo esta metodología debido al desconocimiento total del flujo de trabajo y requerimientos específicos que se tenían sobre los aspectos relacionados a la monitorización de variables ambientales en un invernadero.

#### Fase 01 Inicio

##### *Visión del proyecto*

Constituye el fundamento del proyecto, no requiere ser detallada o extensa y debe enfocarse en el problema, necesidad u oportunidad. Para IoTemp hemos diseñado la siguiente visión:

“**Para** negocios y personas que se dedican a la producción de productos agrícolas

**Quienes** requieren monitorear la temperatura y humedad relativa de sus invernaderos

IoTemp **es una** aplicación móvil que integra IoT (Internet of things)

**Que** es una solución ágil y económica para monitorear y preservar el capital de los invernaderos

**Diferente a** los controles tradicionales que no son automatizados

**Nuestro producto** es una opción de bajo costo para optimizar la monitorización y así mejorar el control de la temperatura al momento de la producción de cultivos.”

### ***Formar el Scrum Team***

En primera instancia se ha realizado un análisis para definir los miembros del equipo SCRUM lo cual puede observarse en la Tabla 5.

**Tabla 5**

*Equipo Scrum*

<b>Roles</b>	<b>Descripción</b>	<b>Persona</b>
<b>Product Owner</b>	Dueño del producto	Ing. William Panchi
<b>Scrum Master</b>	Director del proyecto	Ing. Diego Velasco
<b>Development Team</b>	Equipo de desarrollo	Steve Jimbo Luis Martínez

### ***Desarrollar épicas***

Para el levantamiento de requisitos se realizó una visita guiada por parte del Ingeniero Agrónomo en el invernadero, mediante la cual se definió el funcionamiento del mismo, quienes son las personas involucradas, con qué herramientas cuentan para realizar cada una de las actividades, así como el tipo de plan de acción u organización que siguen para el monitoreo de los cultivos.

De igual forma, el Ingeniero Agrónomo resaltó el proceso manual de monitorización, el cual requería que él o uno de los trabajadores se desplazará a cada uno de los sectores en los que contaban con termómetros para verificar el estado de las variables ambientales, significando además que se debía hacer un registro manual en caso de que ocurrieran incidencias.

Gracias a ello se dio a conocer la necesidad de un proceso automatizado para la monitorización de variables ambientales, tomando en cuenta que el establecimiento ya contaba con un sistema automatizado de riego, la infraestructura era la adecuada para la instalación de nuevos componentes, sentando las bases del sistema en la monitorización y el registro automático de información que permita mejorar la productividad en el establecimiento, bases abarcadas en un proceso principal:

Monitorización el sistema deberá permitir conocer el estado de las variables ambientales de todo el invernadero, proporcionando información clara e inmediata a la cual se tenga acceso desde un teléfono inteligente, mismo en el que se podrán ver registros anteriores de cada medición y se tengan notificaciones cuando existan problemas con las variables ambientales.

Además, el acceso al sistema estará reservado únicamente para el Ingeniero Agrónomo quien podrá configurar los parámetros de alertas de la aplicación y trabajadores específicos que determine, controlando su uso mediante credenciales y limitando la funcionalidad dependiendo del rol que cumpla dicho trabajador.

**Tabla 6***Tipos de usuario*

<b>Tipo de usuario</b>	<b>Descripción</b>
<b>Administrador</b>	Ingeniero Agrónomo. Es el personal especializado y capacitado para analizar, medir, tomar decisiones con respecto a los procesos de trabajo en el invernadero y control de los cultivos. Posee control directo de los procesos de mitigación, así como de la determinación de información vital para el mantenimiento del invernadero.
<b>Técnico</b>	Es el personal encargado de la monitorización, transporte y venta, efectuando actividades de mano de obra dentro del establecimiento.

*Nota.* Tabla de los usuarios identificados dentro del proceso de monitorización del centro agropecuario “Pilonera Verito”.

Una épica es considerada una historia de usuario que, por su gran tamaño, debe ser descompuesta en historias de usuarios con un tamaño más ajustado para ser propuesta y desarrollada en un sprint.

**Figura 3***Épicas de IoTemp*

<b>ID: EP-001</b>	<b>Épica</b>
	Descripción: <b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> Tener un sistema automatizado para la monitorización de las variables ambientales en el invernadero <b>Para</b> Controlar la evolución y estado de los cultivos dentro y fuera del establecimiento.

*Nota.* Épica elaborada en conjunto con el Ingeniero Agrónomo.

Como se puede observar en la Figura 1 al momento de hablar con el Ingeniero Agrónomo se obtuvo una gran épica que sería la base del desarrollo de este proyecto de titulación. La “EP-001” nos introduce en el objetivo principal del sistema que será la monitorización de las variables ambientales del invernadero para múltiples dispositivos sin restricciones geográficas.

## **Fase 02 Planificación y Estimación**

### ***Criterios de aceptación***

Los criterios de aceptación son parte de las buenas prácticas de la ingeniería de software ágil, se basan en definir las condiciones específicas sobre cada historia de usuario respecto a su comportamiento o calidad técnica, se lo realiza previo a su implementación por lo que suele realizarse en el refinamiento previo al Sprint.

### ***Crear Historias de Usuario***

A partir del análisis de la épica extraída de los requerimientos presentados por el usuario se obtuvieron 7 historias de usuario como se observa en la Figura 1, las cuales engloban el desarrollo de la infraestructura de sensores, API y aplicación móvil. A continuación, se detallan cada una de las historias de usuario con su respectivo criterio de aceptación.

#### Figura 4

##### Historia de usuario HU-001

<b>HU-001: Temperatura</b>	<b>Épica: EP-001</b>
<b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> tener un proceso automatizado de medición de temperatura relativa en el invernadero <b>Para</b> mantener un control constante de dicha variable ambiental.	
<b>Criterios de aceptación</b>	
<p><b>Contexto:</b> Los sensores de temperatura están instalados en las zonas designadas</p> <p><b>Evento:</b> El dispositivo IoT cada 15 minutos realiza una recopilación de mediciones de todos los sensores</p> <p><b>Resultado:</b> Se envía información desde el dispositivo IoT hacia el servidor mediante protocolo MQTT con las mediciones de temperatura relativa</p>	

#### Figura 5

##### Historia de usuario HU-002

<b>HU-002: Humedad</b>	<b>Épica: EP-001</b>
<b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> tener un proceso automatizado de medición de humedad relativa en el invernadero <b>Para</b> mantener un control constante de dicha variable ambiental	
<b>Criterios de aceptación</b>	
<p><b>Contexto:</b> Los sensores de humedad están instalados en las zonas designadas</p> <p><b>Evento:</b> El dispositivo IoT cada 15 minutos realiza una recopilación de mediciones de todos los sensores</p> <p><b>Resultado:</b> Se envía información desde el dispositivo IoT hacia el servidor mediante protocolo MQTT con las mediciones de Humedad Relativa</p>	

## Figura 6

### Historia de usuario HU-003

<b>HU-003: Resumen</b>	<b>Épica:</b> EP-001
<b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> poder visualizar un resumen de monitorización en tiempo real de la temperatura y humedad relativa <b>Para</b> conocer el estado de cada zona en el invernadero	
<b>Criterios de aceptación</b>	
<p><b>Contexto:</b> El usuario tiene instalada la aplicación</p> <p><b>Evento:</b> Abre la aplicación móvil y se logea</p> <p><b>Resultado:</b> Se despliega en la pantalla principal los valores correspondientes a las mediciones de humedad realizadas en las 3 zonas por los sensores.</p>	

## Figura 7

### Historia de usuario HU-004

<b>HU-004: Gráfico estadístico</b>	<b>Épica:</b> EP-001
<b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> acceder a un gráfico estadístico histórico de cada variable ambiental en un tiempo determinado <b>Para</b> conocer la evolución de las mediciones.	
<b>Criterios de aceptación</b>	
<p><b>Contexto:</b> El usuario está logueado</p> <p><b>Evento:</b> El usuario selecciona un rango de fechas en la pantalla estadística</p> <p><b>Resultado:</b> Se despliega un gráfico de líneas representando las mediciones obtenidas en ese dicho período de tiempo</p>	

## Figura 8

*Historia de usuario HU-005*

<b>HU-005: Intervalos de normalidad</b>	<b>Épica:</b> EP-001
<b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> configurar intervalos de normalidad para las mediciones en el día y en la noche <b>Para</b> tener un control dinámico de las variables ambientales	
<b>Criterios de aceptación</b>	
<p><b>Contexto:</b> El usuario está logueado como administrador</p> <p><b>Evento:</b> El usuario accede a la pantalla de configuración de los intervalos y modifica los rangos establecidos</p> <p><b>Resultado:</b> Se configura en el servidor y en la aplicación los nuevos rangos de normalidad</p>	

## Figura 9

*Historia de usuario HU-006*

<b>HU-006: Alertas</b>	<b>Épica:</b> EP-001
<b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> recibir alertas cuando las mediciones no estén dentro de los rangos normales <b>Para</b> tomar decisiones inmediatas en consecuencia	
<b>Criterios de aceptación</b>	
<p><b>Contexto:</b> El usuario tiene instalada la aplicación</p> <p><b>Evento:</b> Las mediciones que llegan al servidor están fuera de los rangos de normalidad</p> <p><b>Resultado:</b> Se despliega una notificación en el celular para alertar al usuario.</p>	

## Figura 10

*Historia de usuario HU-007*

<b>HU-007: Credenciales</b>	<b>Épica: EP-001</b>
<b>Como</b> Ingeniero Agrónomo <b>Quisiera</b> acceder con un nombre y una clave a la información de la monitorización en el invernadero <b>Para</b> preservar la privacidad de la información.	
<b>Criterios de aceptación</b>	
<b>Contexto:</b> El usuario tiene instalada la aplicación <b>Evento:</b> El usuario accede a la aplicación e ingresa sus credenciales en el formulario de inicio de sesión <b>Resultado:</b> Se autoriza o niega el acceso a la aplicación	

### ***Aprobar, estimar y comprometer historias de usuario***

Las historias de usuario deben ser planteadas al equipo que posteriormente aprobará y estimará basándose en la aplicación de técnicas de planificación como el “Método Póker Scrum”. Al aprobar y estimar estas historias de usuario el Scrum Team se compromete con cada una de las historias.

### **Tabla 7**

*Estimación de historias de usuario*

<b>ID</b>	<b>Título</b>	<b>Puntos de historia</b>
<b>HU-001</b>	Temperatura	13
<b>HU-002</b>	Humedad	13
<b>HU-003</b>	Resumen	8
<b>HU-004</b>	Gráfico Estadístico	5
<b>HU-005</b>	Intervalos de normalidad	13
<b>HU-006</b>	Alertas	8
<b>HU-007</b>	Credenciales	5
<b>Total</b>		65

*Nota.* Historias de Usuario aprobadas y estimadas mediante la técnica de Poker Scrum.

### **Generación del Product BackLog**

En la aplicación del marco de trabajo SCRUM, existe la necesidad de generar y usar artefactos que permitan mantener una buena organización durante el desarrollo del proyecto. Uno de ellos es el product backlog, el cual es un listado establecido y mantenido por el dueño del producto en el cual puede observarse claramente los requerimientos funcionales del sistema, así como su prioridad, complejidad y esfuerzo necesario para ser completado. Este artefacto es el punto de partida del cual el equipo de desarrollo obtendrá los ítems a cumplir en los posteriores sprints (Bahit, 2012).

**Tabla 8**

#### *Product Backlog*

<b>ID</b>	<b>Título</b>	<b>Prioridad</b>	<b>Complejidad</b>	<b>N. Sprint</b>	<b>Duración (días)</b>
HU-001	Temperatura	Alta	Alta	1	5
HU-002	Humedad	Alta	Media	1	5
HU-003	Resumen	Alta	Media	1	4
HU-007	Credenciales	Media	Baja	2	7
HU-004	Gráfico Estadístico	Media	Baja	2	7
HU-005	Intervalos de normalidad	Media	Alta	3	7
HU-006	Alertas	Media	Alta	3	7
<b>Total</b>					<b>42</b>

*Nota.* Product backlog creado a partir de las historias de usuario.

}

Como se puede observar en la Tabla 8 hemos realizado una priorización de las diferentes historias de usuario a abarcar a lo largo del proyecto. Se determinó que cada Sprint tenga como duración el tiempo de 2 semanas que sería el mínimo según la documentación de la metodología. De tal manera se han priorizado las historias de usuario que aportan mayor valor al negocio, optando por aquellas que proporcionan la información clave al usuario para poder realizar las actividades dentro del establecimiento.

### **Crear Tareas**

**Tabla 9**

*Creación de tareas*

<b>HU</b>	<b>Título</b>	<b>Tareas</b>
HU-001	Temperatura	<ul style="list-style-type: none"> <li>● Diseño de la arquitectura del sistema</li> <li>● Despliegue de Servidor</li> <li>● Configuración de servicios MQTT</li> <li>● Configuración de backend para mensajes MQTT</li> <li>● Configuración de autenticación para conexión MQTT</li> <li>● Diseño de la infraestructura IoT</li> <li>● Cableado de la infraestructura IoT</li> <li>● Instalación de los sensores de temperatura</li> <li>● Instalación del módulo IoT</li> <li>● Configuración del módulo IoT para monitorizar temperatura</li> <li>● Conexión del módulo IoT con el servidor</li> <li>● Configuración del servidor para envío de datos de temperatura</li> </ul>
HU-002	Humedad	<ul style="list-style-type: none"> <li>● Instalación de los sensores de humedad</li> <li>● Configuración del módulo IoT para monitorizar humedad</li> <li>● Conexión del módulo IoT con el servidor</li> <li>● Configuración del servidor para envío de datos de humedad</li> <li>● Recepción de datos de temperatura desde el servidor en la aplicación móvil</li> <li>● Envío de datos de humedad desde el servidor hacia la aplicación móvil</li> <li>● Diseño del módulo de humedad en la pantalla de resumen de la aplicación móvil</li> </ul>

HU	Título	Tareas
HU-007	Credenciales	<ul style="list-style-type: none"> <li>● Implementación del módulo de humedad en la pantalla de resumen de la aplicación móvil</li> </ul> Recepción de datos de humedad desde el servidor en la aplicación móvil
HU-007	Credenciales	<ul style="list-style-type: none"> <li>● Creación de API bajo Slim Framework</li> <li>● Programación de API para funcionalidad de Usuarios</li> <li>● Diseño de la pantalla de inicio de sesión</li> <li>● Implementación de la pantalla de inicio de sesión</li> <li>● Conexión de la aplicación móvil con el servidor</li> <li>● Configuración de roles en la aplicación móvil</li> </ul>
HU-004	Gráfico estadístico	<ul style="list-style-type: none"> <li>● Programación API para manejo de la data histórica</li> <li>● Diseño de la pantalla estadística</li> <li>● Implementación de la pantalla estadística</li> <li>● Implementación del selector de fechas</li> <li>● Recepción de datos desde el servidor</li> <li>● Despliegue de la información del servidor en el gráfico de líneas</li> </ul>
HU-005	Intervalos de normalidad	<ul style="list-style-type: none"> <li>● Programación API para almacenamiento de la configuración de los rangos normales.</li> <li>● Diseño de la pantalla de configuración de intervalos</li> <li>● Implementación de la pantalla de configuración de intervalos</li> <li>● Recepción de los datos de intervalos del servidor</li> <li>● Almacenamiento de nuevos intervalos en el dispositivo</li> </ul>
HU-006	Alertas	<ul style="list-style-type: none"> <li>● Programación SCRIPT para escucha y manejo de los rangos normales para alertar anomalías</li> <li>● Configuración del servidor con el servicio de Firebase para uso de Alertas</li> <li>● Configuración de la aplicación para el envío de alertas</li> <li>● Configuración de la aplicación para la recepción de alertas</li> <li>● Configuración de la aplicación para funcionamiento en segundo plano</li> </ul>

*Nota.* Tareas planificadas para cada historia de usuario.

**Estimar Tareas****Tabla 10***Estimación de tareas*

<b>ID Historia De Usuario</b>	<b>Tarea</b>	<b>Dueño/Voluntario</b>	<b>Horas estimadas totales</b>
<b>HU-001</b>	Diseño de la arquitectura de sistema	Luis Martínez	6
	Despliegue de servidor	Steve Jimbo	2
	Configuración de servicios MQTT	Steve Jimbo	2
	Configuración de backend para mensajes MQTT	Steve Jimbo	1
	Configuración de autenticación para conexión MQTT	Steve Jimbo	1
	Diseño de la infraestructura IoT	Steve Jimbo	8
	Cableado de la infraestructura IoT	Steve Jimbo	4
	Instalación de los sensores de temperatura	Steve Jimbo	4
	Instalación del módulo IoT	Steve Jimbo	8
	Configuración del módulo IoT para monitorizar temperatura	Steve jimbo	2
	Conexión del módulo IoT con el servidor	Steve jimbo	1
	Configuración del servidor para envío de datos de temperatura	Steve Jimbo	1
<b>HU-002</b>	Instalación de los sensores de humedad	Steve Jimbo	4
	Configuración del módulo IoT para monitorizar humedad	Steve Jimbo	2

<b>ID Historia De Usuario</b>	<b>Tarea</b>	<b>Dueño/Voluntario</b>	<b>Horas estimadas totales</b>
	Conexión del módulo IoT con el servidor	Steve Jimbo	1
	Configuración del servidor para envío de datos de humedad	Steve Jimbo	1
<b>HU-003</b>	Envío de datos de temperatura desde el servidor hacia la aplicación móvil	Steve Jimbo	6
	Diseño del módulo de temperatura en la pantalla de resumen de la aplicación móvil	Luis Martínez	4
	Implementación del módulo de temperatura en la pantalla de resumen de la aplicación móvil	Luis Martínez	3
	Recepción de datos de temperatura desde el servidor en la aplicación móvil	Luis Martínez	2
	Envío de datos de humedad desde el servidor hacia la aplicación móvil	Steve Jimbo	6
	Diseño del módulo de humedad en la pantalla de resumen de la aplicación móvil	Luis Martínez	3
	Implementación del módulo de humedad en la pantalla de resumen de la aplicación móvil	Luis Martínez	3
	Recepción de datos de humedad desde el servidor en la aplicación móvil	Luis Martínez	3

<b>ID Historia De Usuario</b>	<b>Tarea</b>	<b>Dueño/Voluntario</b>	<b>Horas estimadas totales</b>
<b>HU-007</b>	Creación de API bajo Slim Framework	Steve Jimbo	8
	Programación de API para funcionalidad de Usuarios	Steve Jimbo	8
	Diseño de la pantalla de inicio de sesión	Luis Martínez	10
	Implementación de la pantalla de inicio de sesión	Luis Martínez	10
	Conexión de la aplicación móvil con el servidor	Luis Martínez	10
	Configuración de roles en la aplicación móvil	Luis Martínez	10
<b>HU-004</b>	Programación API para manejo de la data histórica	Steve Jimbo	16
	Diseño de la pantalla estadística	Luis Martínez	10
	Implementación de la pantalla estadística	Luis Martínez	10
	Implementación del selector de fechas	Luis Martínez	8
	Recepción de datos desde el servidor	Luis Martínez	10
	Despliegue de la información del servidor en el gráfico de líneas	Luis Martínez	2
<b>HU-005</b>	Programación API para almacenamiento de la configuración de los rangos normales.	Steve Jimbo	8
	Diseño de la pantalla de configuración de intervalos	Luis Martínez	10
	Implementación de la pantalla de configuración de intervalos	Luis Martínez	12

<b>ID Historia De Usuario</b>	<b>Tarea</b>	<b>Dueño/Voluntario</b>	<b>Horas estimadas totales</b>
	Recepción de los datos de intervalos del servidor	Luis Martínez	10
	Almacenamiento de nuevos intervalos en el dispositivo	Luis Martínez	16
<b>HU-006</b>	Programación SCRIPT para escucha y manejo de los rangos normales para alertar anomalías	Steve Jimbo	13
	Configuración del servidor con el servicio de Firebase para uso de Alertas	Steve Jimbo	13
	Configuración de la aplicación para el envío de alertas	Luis Martínez	10
	Configuración de la aplicación para la recepción de alertas	Luis Martínez	10
	Configuración de la aplicación para funcionamiento en segundo plano	Luis Martínez	10

**Crear sprint backlog****Sprint 1****Tabla 11***Sprint backlog del sprint 1*

<b>ID Historia De Usuario</b>	<b>Por hacer</b>	<b>En proceso</b>	<b>En pruebas</b>	<b>Finalizado</b>
<b>HU-001</b>	Diseño de la arquitectura del sistema  Despliegue de servidor  Configuración de servicios MQTT  Configuración de backend para mensajes MQTT  Configuración de autenticación para conexión MQTT  Diseño de la infraestructura IoT  Cableado de la infraestructura IoT  Instalación de los sensores de temperatura  Instalación del módulo IoT  Configuración del módulo IoT para monitorizar temperatura  Conexión del módulo IoT con el servidor  Configuración del servidor para envío de datos de temperatura			
<b>HU-002</b>	Instalación de los sensores de humedad  Configuración del módulo IoT para monitorizar humedad  Conexión del módulo IoT con el servidor			

ID Historia De Usuario	Por hacer	En proceso	En pruebas	Finalizado
	Configuración del servidor para envío de datos de humedad			
<b>HU-003</b>	<p data-bbox="391 396 792 491">Envío de datos de temperatura desde el servidor hacia la aplicación móvil</p> <p data-bbox="391 527 837 621">Diseño del módulo de temperatura en la pantalla de resumen de la aplicación móvil</p> <p data-bbox="391 657 792 751">Implementación del módulo de temperatura en la pantalla de resumen de la aplicación móvil</p> <p data-bbox="391 787 821 882">Recepción de datos de temperatura desde el servidor en la aplicación móvil</p> <p data-bbox="391 917 756 1012">Envío de datos de humedad desde el servidor hacia la aplicación móvil</p> <p data-bbox="391 1047 802 1142">Diseño del módulo de humedad en la pantalla de resumen de la aplicación móvil</p> <p data-bbox="391 1178 792 1272">Implementación del módulo de humedad en la pantalla de resumen de la aplicación móvil</p> <p data-bbox="391 1308 821 1381">Recepción de datos de humedad desde el servidor en la aplicación móvil</p>			

*Nota.* Tareas planificadas que fueron incluidas en el Sprint Backlog 1.

**Sprint 2****Tabla 12***Sprint backlog del sprint 2*

<b>ID Historia De Usuario</b>	<b>Por hacer</b>	<b>En proceso</b>	<b>En pruebas</b>	<b>Finalizado</b>
<b>HU-007</b>	Creación de API bajo Slim Framework  Programación de API para funcionalidad de Usuarios  Diseño de la pantalla de inicio de sesión  Implementación de la pantalla de inicio de sesión  Conexión de la aplicación móvil con el servidor  Configuración de roles en la aplicación móvil			
<b>HU-004</b>	Programación API para manejo de la data histórica  Diseño de la pantalla estadística  Implementación de la pantalla estadística  Implementación del selector de fechas  Recepción de datos desde el servidor  Despliegue de la información del servidor en el gráfico de líneas			

*Nota.* Tareas planificadas que fueron incluidas en el Sprint Backlog 2.

**Sprint 3****Tabla 13***Sprint backlog del sprint 3*

<b>ID Historia De Usuario</b>	<b>Por Hacer</b>	<b>En proceso</b>	<b>En pruebas</b>	<b>Finalizado</b>
<b>HU-005</b>	<p>Programación API para almacenamiento de la configuración de los rangos normales.</p> <p>Diseño de la pantalla de configuración de intervalos</p> <p>Implementación de la pantalla de configuración de intervalos</p> <p>Recepción de los datos de intervalos del servidor</p> <p>Almacenamiento de nuevos intervalos en el dispositivo</p>			
<b>HU-006</b>	<p>Programación SCRIPT para escucha y manejo de los rangos normales para alertar anomalías</p> <p>Configuración del servidor con el servicio de Firebase para uso de Alertas</p> <p>Configuración de la aplicación para el envío de alertas</p> <p>Configuración de la aplicación para la recepción de alertas</p> <p>Configuración de la aplicación para funcionamiento en segundo plano</p>			

*Nota.* Tareas planificadas que fueron incluidas en el Sprint Backlog 3

## Fase 03 Implementación

### Crear entregables

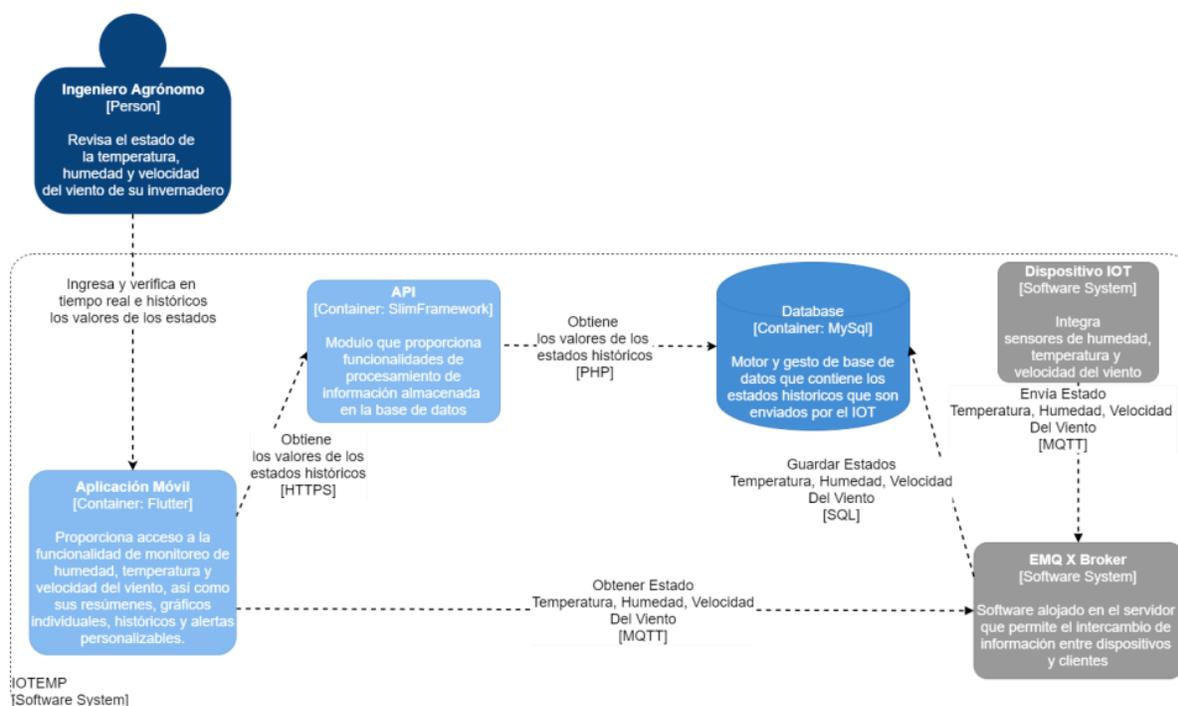
Cada miembro del Scrum Team se encarga de desarrollar las diferentes tareas que se asignan en cada Sprint, resultado de ese desarrollo es un entregable que cumpliría con la funcionalidad descrita en la tarea.

### Sprint 1

#### Tarea: Diseño de la arquitectura del sistema

Figura 11

Diagrama de contenedores del sistema

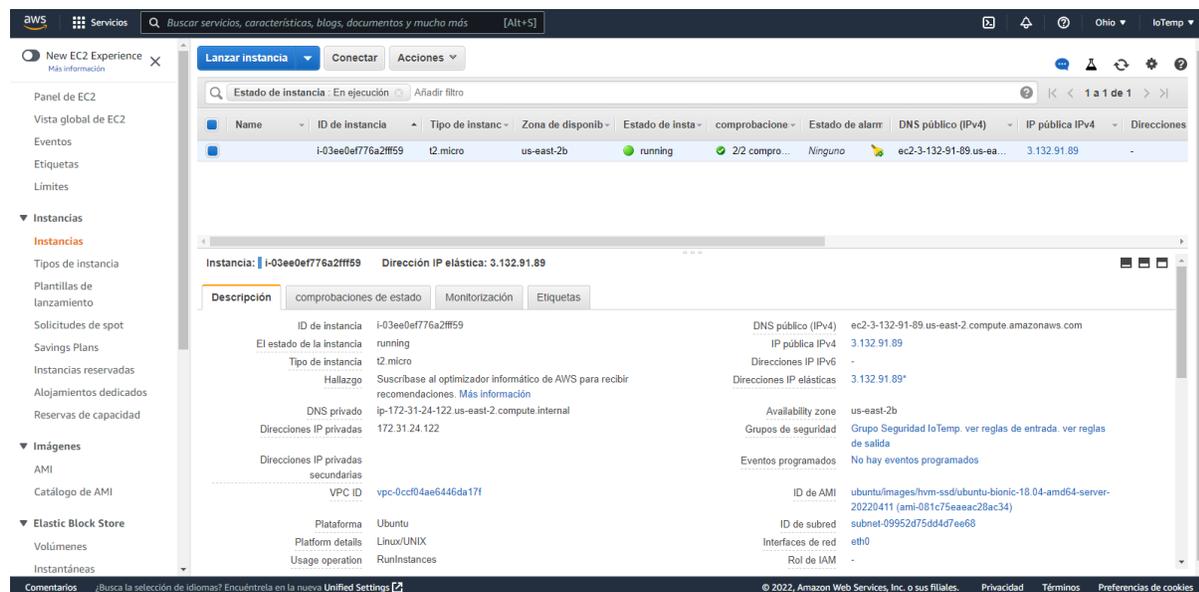


Como puede observarse en la Figura 11, el sistema consiste en una aplicación móvil la cual tiene acceso a un API conectado a una base de datos, mediante la cual se gestionan los registros de mediciones, permitiendo almacenar un historial. De igual forma la aplicación móvil debe conectarse a un broker el cual permite la conexión con un dispositivo mqtt el cual publica información en un intervalo de tiempo hacia un cliente, proveyendo información ya sea para desplegarla y almacenarla.

### Tarea: Despliegue de servidor

## Figura 12

### Instancia creada del servidor en AWS



The screenshot displays the AWS Management Console interface for an EC2 instance. The instance is named 'i-03ee0ef776a2ff59' and is in a 'running' state. The console shows various details for the instance, including its ID, type (t2.micro), availability zone (us-east-2b), and public IP address (3.132.91.89). The instance is running Ubuntu Linux on the 'ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20220411' AMI.

Nombre	ID de instancia	Tipo de instancia	Zona de disponibilidad	Estado de instancia	Comprobación	Estado de alarma	DNS público (IPv4)	IP pública IPv4	Direcciones
	i-03ee0ef776a2ff59	t2.micro	us-east-2b	running	2/2 comprobaciones	Ninguno	ec2-3-132-91-89-us-east-2.compute.amazonaws.com	3.132.91.89	-

Instancia: i-03ee0ef776a2ff59		Dirección IP elástica: 3.132.91.89	
<b>Descripción</b>	comprobaciones de estado	Monitorización	Etiquetas
ID de instancia	i-03ee0ef776a2ff59	DNS público (IPv4)	ec2-3-132-91-89-us-east-2.compute.amazonaws.com
El estado de la instancia	running	IP pública IPv4	3.132.91.89
Tipo de instancia	t2.micro	Direcciones IP IPv6	-
Hallazgo	Suscríbase al optimizador informático de AWS para recibir recomendaciones. <a href="#">Más información</a>	Direcciones IP elásticas	3.132.91.89*
DNS privado	ip-172-31-24-122.us-east-2.compute.internal	Availability zone	us-east-2b
Direcciones IP privadas	172.31.24.122	Grupos de seguridad	Grupo Seguridad loTemp. <a href="#">ver reglas de entrada</a> . <a href="#">ver reglas de salida</a>
Direcciones IP privadas secundarias	-	Eventos programados	No hay eventos programados
VPC ID	vpc-0ccf04ae646da17f	ID de AMI	ubuntu/images/hvm-ssd/ubuntu-bionic-18.04-amd64-server-20220411 (ami-081c75eaeac28ac34)
Plataforma	Ubuntu	ID de subred	subnet-09952d75d4d47ee68
Platform details	Linux/UNIX	Interfaz de red	eth0
Usage operation	RunInstances	Rol de IAM	-

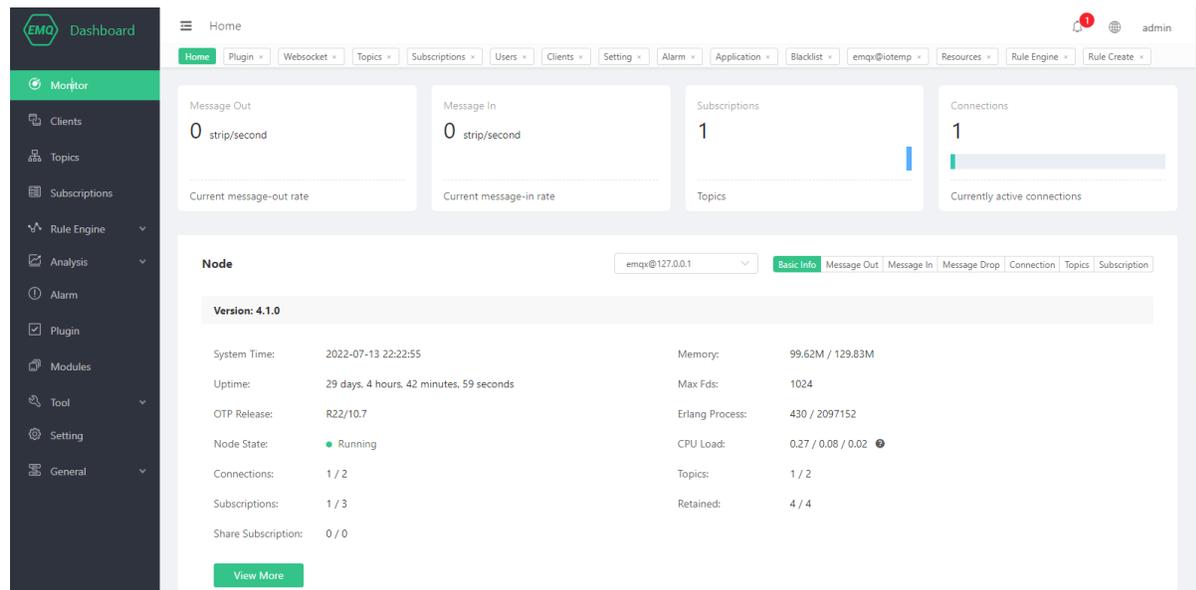
*Nota.* Dashboard de instancias EC2 de la cuenta loTemp. Tomado de la página web [aws.amazon.com](https://aws.amazon.com).

Como se puede observar en la Figura 12 se logra realizar el despliegue de una instancia de servidor en Amazon Web Services. Esta instancia EC2 es de tipo “t2.micro” que corresponde a un servidor de 1 núcleo y 1GB de RAM lo cual es suficiente para realizar el despliegue total del proyecto con los servicios MQTT y Web para el alojamiento de la API en PHP.

### **Tarea: Configuración de servicios MQTT**

#### **Figura 13**

*Servicio EMQ funcionando en el servidor*



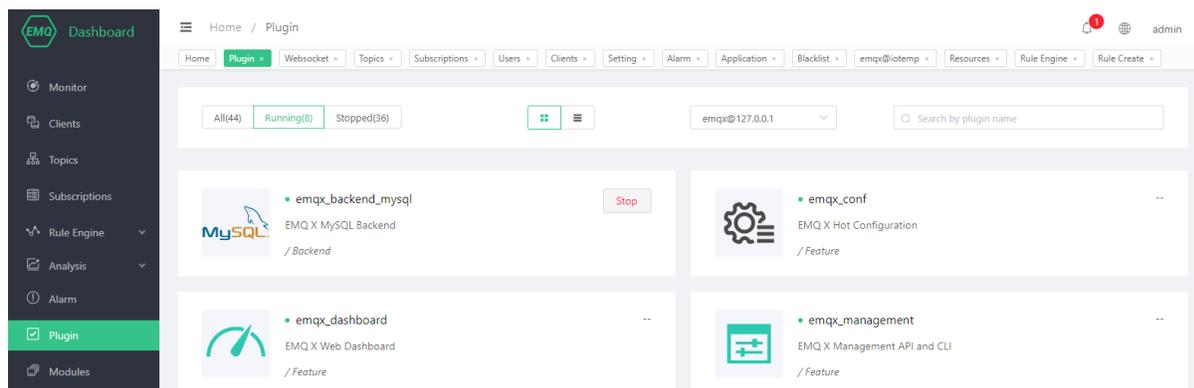
*Nota.* Dashboard de EMQ con información del servicio MQTT. Tomado del servidor de IoTemp.

La Figura 13 nos muestra el dashboard del servicio EMQ ejecutándose en el servidor IoTemp. EMQ es el Broker que nos permitirá realizar la comunicación bajo el protocolo MQTT de nuestro proyecto por lo que es parte esencial dentro del mismo, la configuración que se realiza antes de su ejecución incluye el cambio de los puertos necesarios para escuchar las comunicaciones de los clientes.

## Tarea: Configuración de backend para mensajes MQTT

Figura 14

Plugin de backend ejecutándose en EMQ



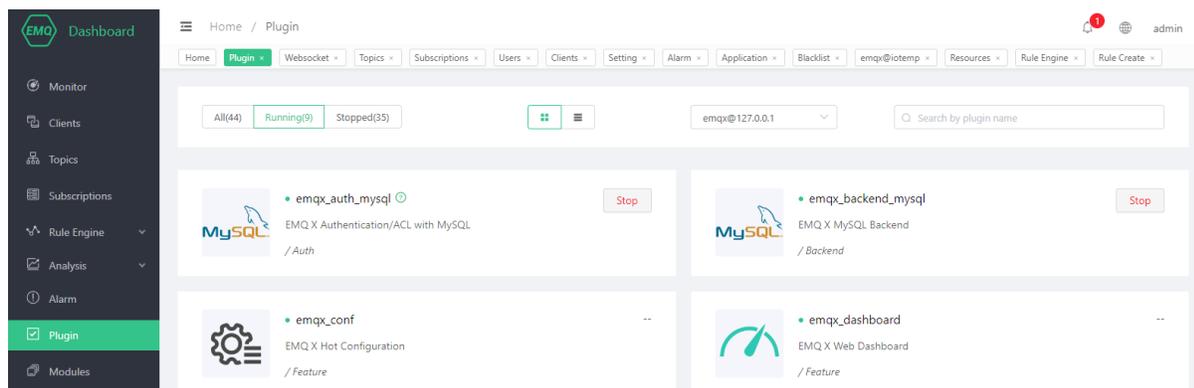
Nota. Apartado de plugins dentro de EMQ

El plugin de backend que tenemos en EMQ nos permite realizar una fácil gestión de los mensajes, además de brindarnos diferentes apartados como los mensajes retenidos, o la información de las conexiones establecidas en el servidor. Para su configuración hay que contar con un servicio de base de datos en MySQL donde almacenaremos todos los aspectos anteriormente comentados. Como se puede observar en la Figura 14 una vez configurada, testada y activa el plugin obtiene un estatus de Running el cual nos indica que se encuentra trabajando en el servidor.

## Tarea: Configuración de autenticación para conexión MQTT

**Figura 15**

*Plugin de autenticación ejecutándose en EMQ*



*Nota.* Apartado de plugins en EMQ

El plugin de autenticación permite mantener un control de los dispositivos que quieren conectarse al servidor, al exigir usuario y contraseña deben coincidir con las almacenadas en la base de datos para poder ingresar y corroborar la identidad del cliente. La contraseña posee una encriptación de tipo SHA256 que aumenta una capa de seguridad al momento de realizar las comprobaciones de identidad. Como se puede observar en la Figura 15 tenemos el plugin de autenticación correctamente configurado y en funcionamiento.

## Tarea: Diseño de la infraestructura IoT

El invernadero del centro agropecuario “Pilonera Verito” se encuentra en San Buenaventura/Latacunga, el invernadero cuenta con tres naves, dos de igual longitud y la tercera un poco más larga. Como se puede observar en la Figura 16, la cual es una vista aérea del invernadero de donde se tomará la referencia para la distribución de sensores y el diseño final de la infraestructura IoT.

## Figura 16

*Vista aérea del invernadero*



*Nota.* Vista aérea del invernadero en el centro agropecuario “Pilonera Verito” tomada de la página web [earth.google.com/web](http://earth.google.com/web).

Para determinar la ubicación de los sensores se basó en la petición del ingeniero agrónomo quien nos especificó cuáles sería la ubicación ideal de cada uno de los sensores, el dispositivo IoT por su necesidad de conexión a internet debe encontrarse en una ubicación con cobertura. Además, considerando futuros mantenimientos al dispositivo o futuras adiciones de control se busca que este en un lugar de fácil acceso. Todo esto lleva a estructura la siguiente infraestructura que se detalla en la Figura 16.

**Figura 17**

*Infraestructura IoT planificada para el invernadero*



*Nota.* Diseño de la infraestructura IoT del invernadero. Imagen base tomada de Google Earth y editada por Steve Jimbo.

Como se puede ver en la Figura 17 se especifica los lugares donde se encontrará los sensores y el dispositivo IoT, previamente el invernadero ya contaba con infraestructura de cables por temas de riego y fungicidas por lo que la conexión sería cableada desde los sensores hacia el dispositivo IoT.

**Tarea: Cableado de la infraestructura IoT**

**Figura 18**

*Cableado físico para la comunicación de los sensores*



*Nota.* Momento donde se realizó el cableado interno en el invernadero para la instalación de los diferentes sensores para las tres zonas.

Se extendió el aproximado de 130 metros de cable UTP categoría 6 desde la oficina del invernadero hacia su interior donde se distribuyeron a las diferentes zonas donde se instalarán los sensores. La zona 1 se utilizó un aproximado de 50 metros, la zona 2 se utilizó un aproximado de 30 y por último para la zona 3 se utilizó un aproximado de 50 metros. Se escogió el cable UTP por cumplir con las características necesarias para el proyecto por su bajo costo por metro y por la capacidad de poder conectar dos sensores en un mismo cable, reduciendo significativamente el costo y la cantidad de cable necesario.

**Tarea: Instalación de los sensores de temperatura y humedad**

**Figura 19**

*Soldadura realizada del cable UTP con los sensores DHT22*



*Nota.* Estación creada para la soldadura de los sensores DHT22 en su respectivo cable UTP.

La soldadura se realizó siguiendo el siguiente patrón: Verde: Energía 5V, Tomate: GND y Café: Datos. Al ser un cable UTP encontramos dos juegos de estos colores, uno de color sólido y otro bicolor con blanco, se repartió de forma que los colores sólidos sean para los sensores que se ubican a la altura de la cama del invernadero mientras que los bicolors se ubican en la parte alta.

**Tarea: Instalación del módulo IoT**

**Figura 20**

*Instalación finalizada del dispositivo IoT*



*Nota.* Dispositivo IoT instalado en la oficina del invernadero.

Una vez el cableado ya se encontraba tendido y los sensores instalados se procedió a la instalación y empaquetamiento del Módulo ESP32 con la conexión de los respectivos sensores. Para el empaquetamiento se utilizó una Caja Legrand Plexo 092044 la cual es utilizada para la entrada de cables que protegerá la integridad del mismo, es una excelente opción al momento de cuidar el módulo y sus conexiones además de ser muy fácil de realizar mantenimientos o similares.

## Tarea: Configuración del módulo IoT para monitorizar temperatura y humedad

Figura 21

Código del módulo ESP 32 para realizar mediciones

```

15 # DECLARACION DE VARIABLES
16
17 sensor1 = DHT22(Pin(27)) #27 - Zona 1 - Baja
18 sensor2 = DHT22(Pin(13)) #13 - Zona 1 - Alta
19 sensor3 = DHT22(Pin(32)) #32 - Zona 2 - Baja
20 sensor4 = DHT22(Pin(33)) #33 - Zona 2 - Alta
21 sensor5 = DHT22(Pin(25)) #25 - Zona 3 - Baja
22 sensor6 = DHT22(Pin(26)) #26 - Zona 3 - Alta
23
24 # Función para leer los datos desde los sensores
25 def MedirData():
26     try:
27         sensor1.measure()
28         sensor2.measure()
29         sensor3.measure()
30         sensor4.measure()
31         sensor5.measure()
32         sensor6.measure()
33         valores = {
34             "tempdz1": sensor1.temperature(),
35             "tempuz1": sensor2.temperature(),
36             "tempdz2": sensor3.temperature(),
37             "tempuz2": sensor4.temperature(),
38             "tempdz3": sensor5.temperature(),
39             "tempuz3": sensor6.temperature(),
40             "humdz1": sensor1.humidity(),
41             "humuz1": sensor2.humidity(),
42             "humdz2": sensor3.humidity(),
43             "humuz2": sensor4.humidity(),
44             "humdz3": sensor5.humidity(),
45             "humuz3": sensor6.humidity()
46         }
47         payload = ujson.dumps(valores)
48         client.publish('pillonera/data', payload, retain=True, qos=1)
49     except OSError as e:
50         Reiniciar_conexion()

```

*Nota.* Código desarrollado para la monitorización de temperatura y humedad escrita en Python.

La configuración del módulo IoT empieza por la declaración de las variables para cada uno de los sensores especificando su respectivo pin físico del módulo. Se utiliza la librería DHT22 escrita propiamente para estos sensores en concreto que facilita y automatiza el código. En la función “measure” que se utiliza para realizar la medición del sensor, la librería obtiene los datos en bruto y los transforma en sus versiones físicas que son reconocibles como serían los grados centígrados y porcentaje. Posteriormente armamos un json que será la información que enviemos hacia el servidor.

### Tarea: Conexión del módulo IoT con el servidor

Figura 22

Código del módulo ESP 32 para conectarse al servidor

```
55 # Funcion que conecta al servidor MQTT y se suscribe a los respectivos topicos
56 def Conexion_MQTT():
57     try:
58         client_id = 'PILONERA-IOT-1'
59         mqtt_server = '3.132.91.89'
60         port_mqtt = 1883
61         user_mqtt = 'PILONERA-IOT-1'
62         pswd_mqtt = 'PiloneraIoT1'
63         client = MQTTClient(client_id, mqtt_server, port_mqtt, user_mqtt, pswd_mqtt, 60)
64         client.set_callback(form_sub)
65         client.connect()
66         client.subscribe('pilonera/monitor')
67         return client
68     except OSError as e:
69         Reiniciar_conexion()
```

Nota. Código desarrollado para la conexión del módulo ESP32 con el servidor MQTT escrito en Python.

Para realizar la conexión del módulo ESP32 con el servidor MQTT es necesario utilizar la librería “umqtt” la cual se especializa en comunicaciones de este protocolo, para una conexión se precisa de datos importantes: dirección IPV4 del servidor, puerto de comunicación, usuario y contraseña del dispositivo y topic al cual será suscrito para el funcionamiento del dispositivo. Una vez realizada la configuración y pruebas de estabilidad se considera que el dispositivo está listo para conectarse al servidor MQTT.

## Tarea: Configuración del servidor para envío de datos de temperatura y humedad

**Figura 23**

### Parámetros de configuración del broker EMQ

mqtt_wildcard_subscription:	<input type="text" value="true"/>	Whether the Server supports MQTT Wildcard Subscriptions
mqtt_strict_mode:	<input type="text" value="false"/>	Whether to parse the MQTT frame in strict mode
mqtt_shared_subscription:	<input type="text" value="true"/>	Whether the Server supports MQTT Shared Subscriptions
mqtt_retain_available:	<input type="text" value="true"/>	Whether the Server supports MQTT retained messages
mqtt_max_topic_levels:	<input type="text" value="0"/>	Maximum topic levels allowed. 0 means no limit
mqtt_max_topic_alias:	<input type="text" value="65535"/>	Maximum Topic Alias. 0 means no topic alias supported
mqtt_max_qos_allowed:	<input type="text" value="2"/>	Maximum QoS allowed
mqtt_max_packet_size:	<input type="text" value="1MB"/>	Maximum MQTT packet size allowed
mqtt_max_clientid_len:	<input type="text" value="65535"/>	Maximum length of MQTT clientid allowed
mqtt_ignore_loop_deliver:	<input type="text" value="false"/>	Whether to ignore loop delivery of messages (for MQTT v3.1.1)
flapping_detect_policy:	<input type="text" value="30, 1m, 5m"/>	Specify the global flapping detect policy
enable_acl_cache:	<input type="text" value="on"/>	Whether to enable ACL cache
allow_anonymous:	<input type="text" value="false"/>	Allow anonymous authentication by default if no auth plugins loaded. Disable the option in production deployment
acl_nomatch:	<input type="text" value="allow"/>	Allow or deny if no ACL rules matched
acl_deny_action:	<input type="text" value="ignore"/>	The action when acl check reject current operation
acl_cache_ttl:	<input type="text" value="1m"/>	The time after which an ACL cache entry will be deleted
acl_cache_max_size:	<input type="text" value="32"/>	The maximum count of ACL entries can be cached for a client

**Nota.** Configuraciones de EMQ para el funcionamiento del servidor MQTT. Tomado de la sección de Settings en el servidor [iotemp.ml:18083/#/setting](http://iotemp.ml:18083/#/setting)

Se realizó la correcta configuración de los parámetros del broker para que el funcionamiento y estabilidad sean los adecuados, dentro de las configuraciones más relevantes tenemos el parámetro de “allow\_anonymus” el cual debe configurarse en “false” para que el servidor no permite conexiones no autenticadas hacia el servidor así se evita que dispositivos de origen desconocido puedan entrar al servidor MQTT.

***Tarea: Envío de datos de temperatura y humedad desde el servidor hacia la aplicación móvil***

**Figura 24**

*Envío de datos desde el dispositivo IoT hacia el servidor*

```
47     payload = ujson.dumps(valores)
48     client.publish('pilonera/data', payload, retain=True, qos=1)
```

*Nota.* Código desarrollado para el envío de datos desde el dispositivo hacia el servidor escrito en Python.

El dispositivo IoT envía los datos hacia el servidor a su respectivo “topic”, sin embargo, si el cliente no está conectado no recibiría esta información, para subsanar esta eventualidad es necesario enviar los datos de una forma especial, el parámetro “retain” que se puede observar en la Figura 24 permite que el servidor identifique estos mensajes para ser enviados inmediatamente después de que una persona se suscriba la mismo tópico por lo que si el cliente se conecta posteriormente al envío de datos, el servidor se encargará de enviarle esta información retenida y mientras dure la conexión actualizarla con nuevos datos que reciba el servidor.

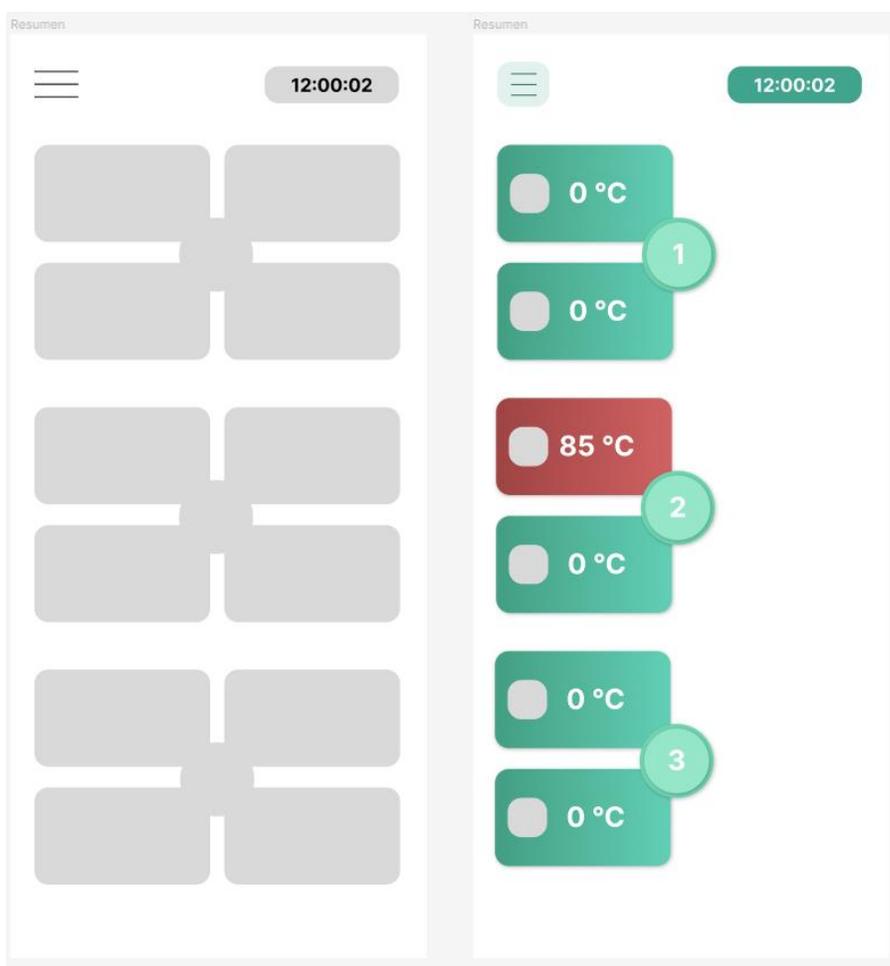
***Tarea: Diseño del módulo de temperatura en la pantalla de resumen de la aplicación móvil***

La aplicación ha sido desarrollada con el objetivo de ofrecer al encargado del cuidado del invernadero información precisa y ordenada, mediante interfaces sencillas y eficientes al momento de presentar el estado de las variables ambientales. Se ha priorizado el despliegue de pocos componentes en cada una de las pantallas para no saturar de información al usuario.

Esta aplicación ha sido desarrollada en Android Studio, con el Framework Flutter junto con el lenguaje de programación Dart, debido a que permite el desarrollo de aplicaciones de forma rápida y organizada, teniendo una curva de aprendizaje inicial baja, permitiendo seccionar el desarrollo en diferentes piezas o Widgets integrables a cada pantalla y maximizando su reutilización. Sumado a las posibilidades que provee Flutter se han añadido varias librerías para complementar la funcionalidad, entre ellas la más significativa `mqtt_client` la cual permite realizar conexiones a un servidor `mqtt`, mismo encargado de distribuir la información entre los distintos dispositivos cliente.

### Figura 25

*Diseño de la pantalla resumen y el módulo de temperatura*



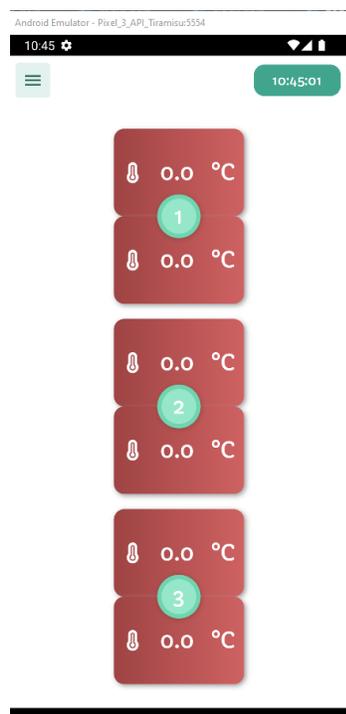
El diseño de la aplicación empezó con la pantalla de resumen, consistiendo este proceso en la proposición de un borrador el cual mostraba la distribución de elementos en la pantalla, para basado en este proceder a colocar los elementos visuales en un diseño simple que resuma el módulo de temperatura y permita al usuario conocer con inmediatez el estado actual de las mediciones.

Este módulo consiste en 6 componentes distribuidos en 3 zonas Figura 25, los cuales despliegan la información de la temperatura tanto a nivel de aire como a nivel de cama. Así mismo cada componente cambia de color cuando la medición está fuera de los rangos establecidos como normales por el usuario.

### ***Tarea: Implementación del módulo de temperatura en la pantalla de resumen de la aplicación móvil***

#### **Figura 26**

##### *Implementación del módulo de temperatura*

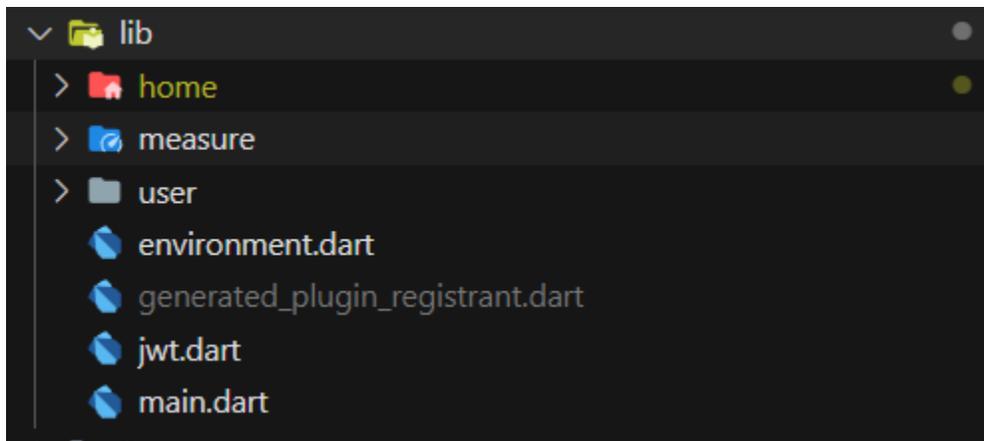


Como puede observarse en la Figura 26, el módulo de temperatura ha sido desarrollado basado en el diseño obtenido en tareas anteriores, adaptando su apariencia al marco de diseño disponible en flutter y pensado para actualizar su diseño en función de los datos publicados por el servidor mqtt.

Para la apertura del proyecto fue necesaria la implementación de un patrón para el manejo de cambio de estado y organización del proyecto, para ello se realizó una estructuración inicial del proyecto dividiéndolo en módulos representativos de la lógica del negocio, Figura 27.

### Figura 27

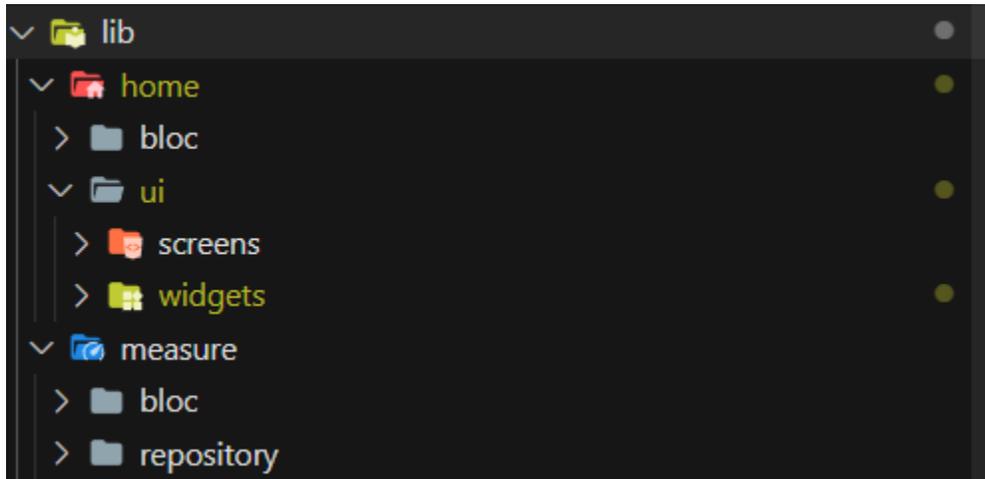
*Estructura inicial del proyecto según el patrón bloc*



La Figura 27, muestra como la lógica del negocio se ha dividido en tres secciones: usuario (user), home y measure (medida), mismas que cuentan con los subcomponentes sugeridos por bloc: bloc, repository, ui. Dichos módulos son encargados de gestionar el cambio de estado y la información pertinente al negocio, proveer conexiones con el servidor y presentar la información respectivamente.

**Figura 28**

*Creación de subcomponentes del patrón bloc*



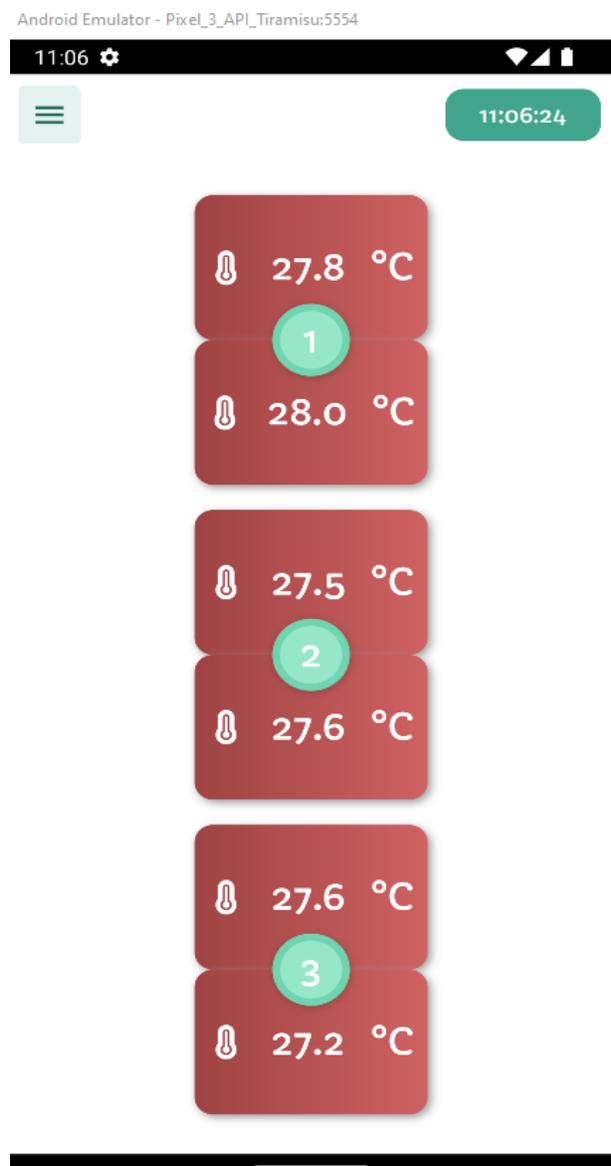
La Figura 28, presenta la división de responsabilidades de cada componente de flutter, separando la interfaz, representada en screens y widgets, de la lógica de negocio, representada en bloc y repository.

**Tarea: Recepción de datos de temperatura desde el servidor en la aplicación**

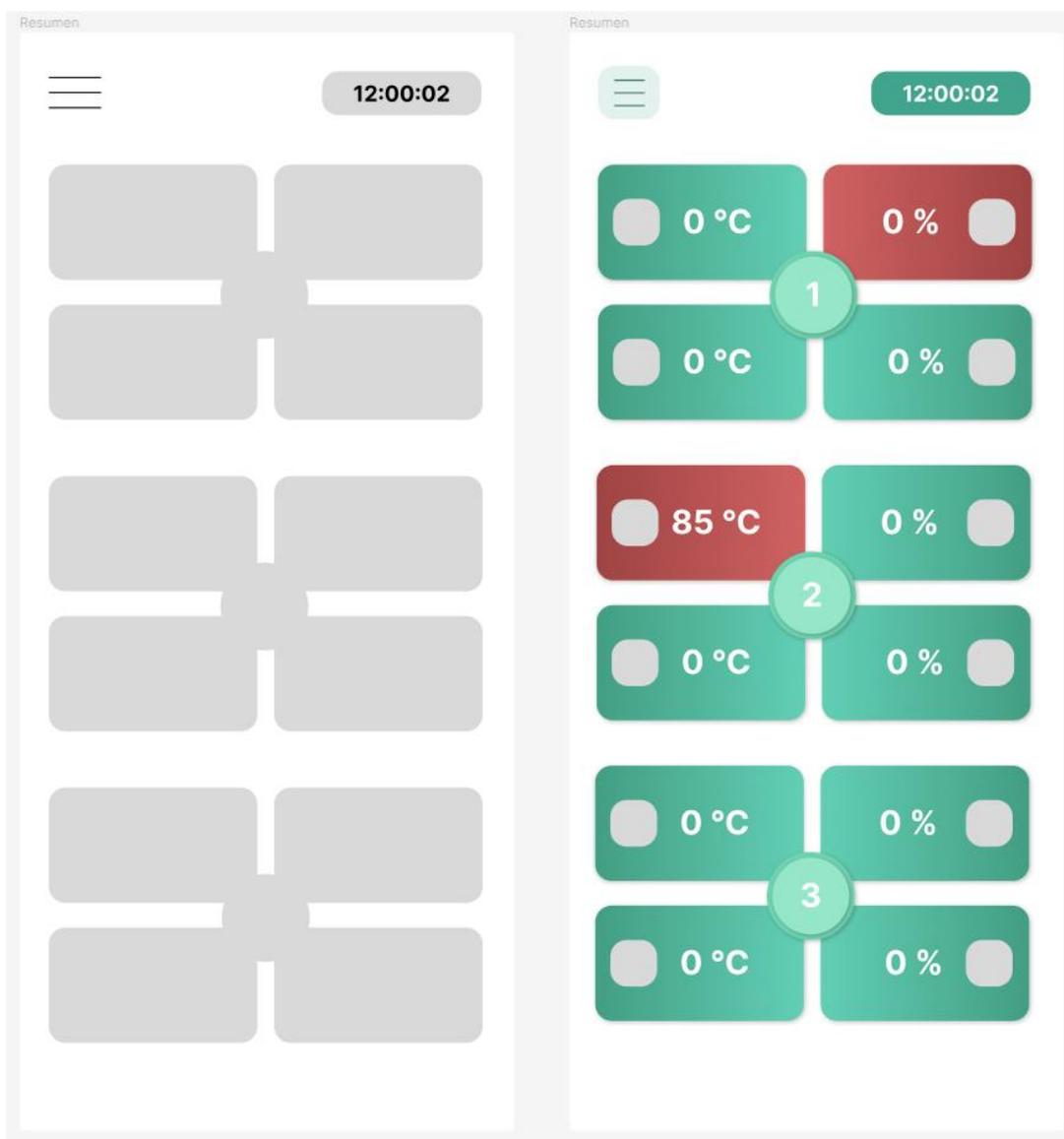
*móvil*

**Figura 29**

*Recepción de datos de temperatura*



La Figura 29, muestra la presentación de los datos de temperatura en la interfaz, siendo dicha información publicada por el servidor mqtt y actualizada por los componentes del bloc para ser mostrada por los diferentes widgets de flutter.

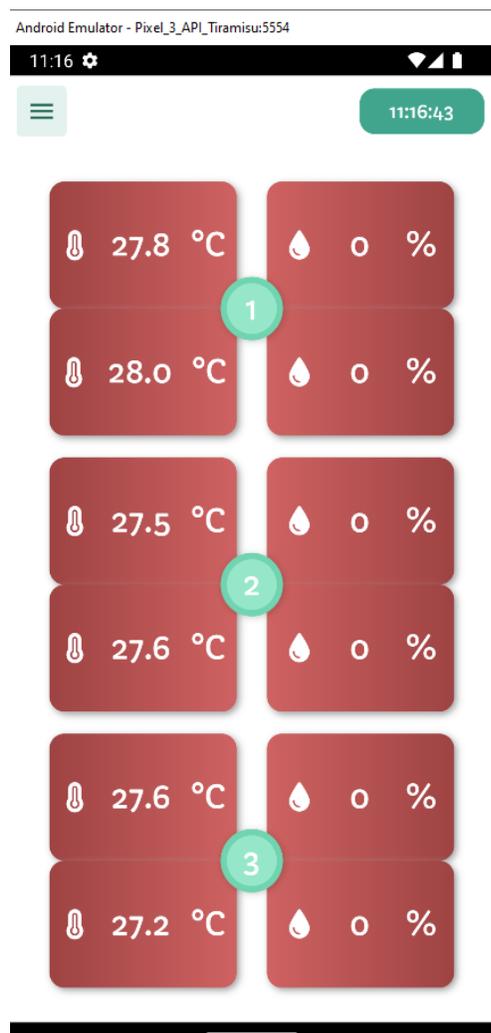
**Tarea: Diseño del módulo de humedad en la pantalla de resumen de la aplicación****móvil****Figura 30***Diseño del módulo de humedad*

Como se muestra en la Figura 30, se añadió al diseño previo el módulo de humedad distribuido en 3 zonas de 2 componentes cada uno, desplegando datos de mediciones a nivel de aire y a nivel de cama, cambiando de color cuando se presentan mediciones anormales en los sensores.

**Tarea: Implementación del módulo de humedad en la pantalla de resumen de la aplicación móvil**

**Figura 31**

*Implementación del módulo de humedad*

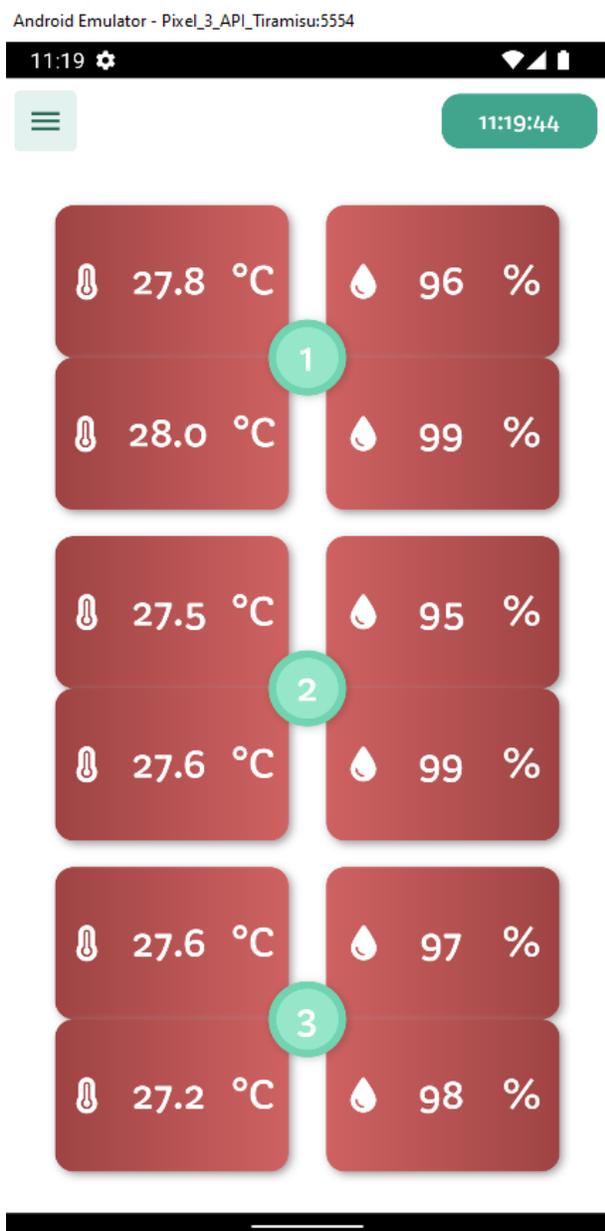


Como puede observarse en la Figura 31, se añadieron los módulos de humedad a la interfaz, diseñados para recibir información y actualizar su diseño en función a ella.

**Tarea: Recepción de datos de humedad desde el servidor en la aplicación móvil**

**Figura 32**

*Recepción de datos de humedad*



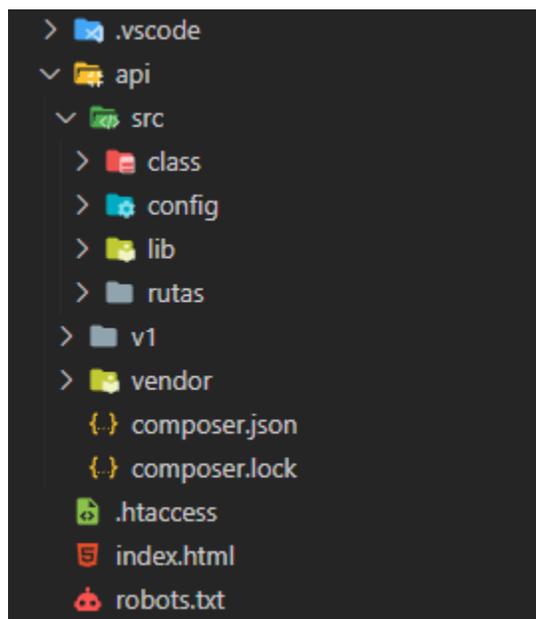
La Figura 32, muestra la recepción de datos de humedad adicionada a la aplicación móvil, al igual que la temperatura, las funciones fueron distribuidas en tres partes: repository para la recepción de datos, bloc para el tratamiento de datos y actualización de las variables o estado, y ui para la representación de los datos mediante pantallas y widgets.

## Sprint 2

### Tarea: Creación de API bajo Slim Framework

#### Figura 33

Directorio de la API bajo Slim Framework



*Nota.* Directorio de archivos del servidor que se genera bajo el Slim Framework.

Slim Framework es un micro framework de PHP que se utiliza para crear aplicaciones web basadas en MVC o APIs por su alto manejo de rutas y el soporte con mensajes HTTP PSR-7. Dentro del directorio los principales archivos que le dan vida a la aplicación se encuentran en “vendor”, las librerías complementarias como “jwt” o “mqtt” serán ubicadas en la carpeta “lib”, en la carpeta “class” podemos colocar diferentes clases que ayudan en la programación de las APIs, finalmente en la carpeta “rutas” se coloca las declaraciones de las diferentes rutas de la API y su correspondiente implementación.

## Tarea: Programación de API para funcionalidad de Usuarios

Figura 34

Código de la API para funcionalidad de Usuarios

```

api > src > rutas > users.php > ...
1  <?php
2
3  use \Psr\Http\Message\ServerRequestInterface as Request;
4  use \Psr\Http\Message\ResponseInterface as Response;
5  use Firebase\JWT\JWT;
6  use Servidor;
7
8  // Usuarios
9
10 //Inicio de Sesión mediante autenticación e credenciales
11 > $app->get('/users/login', function(Request $request, Response $response){ ...
43 });
44
45 //Registro de Usuarios en el sistema
46 > $app->post('/users/create', function(Request $request, Response $response){ ...
64 });

```

*Nota.* Declaración e implementación de las principales funcionalidades para el apartado de usuarios.

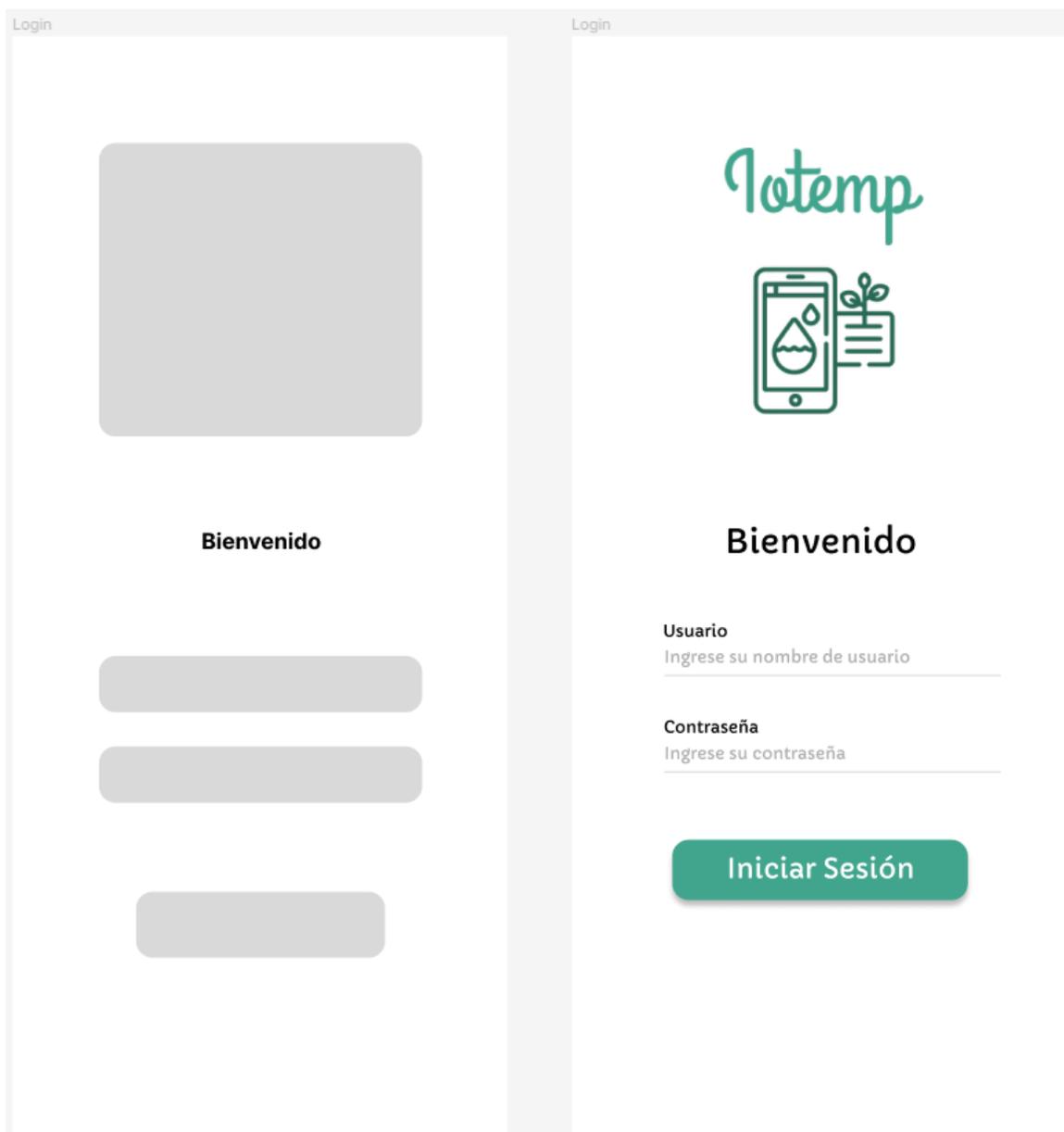
Cada petición a la API debe ser acompañada de sus respectivas cabeceras para comprobar el origen y así evitar que las peticiones sean realizadas fuera de la aplicación. Para el apartado de usuarios las dos principales funcionalidades que se implementan son:

- Login: Esta funcionalidad permite realizar una autenticación de las credenciales de los diferentes usuarios comparándola con la base de datos y su correspondiente encriptación, se busca mantener el acceso limitado para conservar la privacidad de la información.
- Create: Esta funcionalidad permite registrar nuevos usuarios dentro del sistema, para conservar la privacidad esta funcionalidad no está abierta al público ya que la app será distribuida públicamente en la Play Store para Android.

**Tarea: Diseño de la pantalla de inicio de sesión**

**Figura 35**

*Diseño de la pantalla de inicio de sesión*

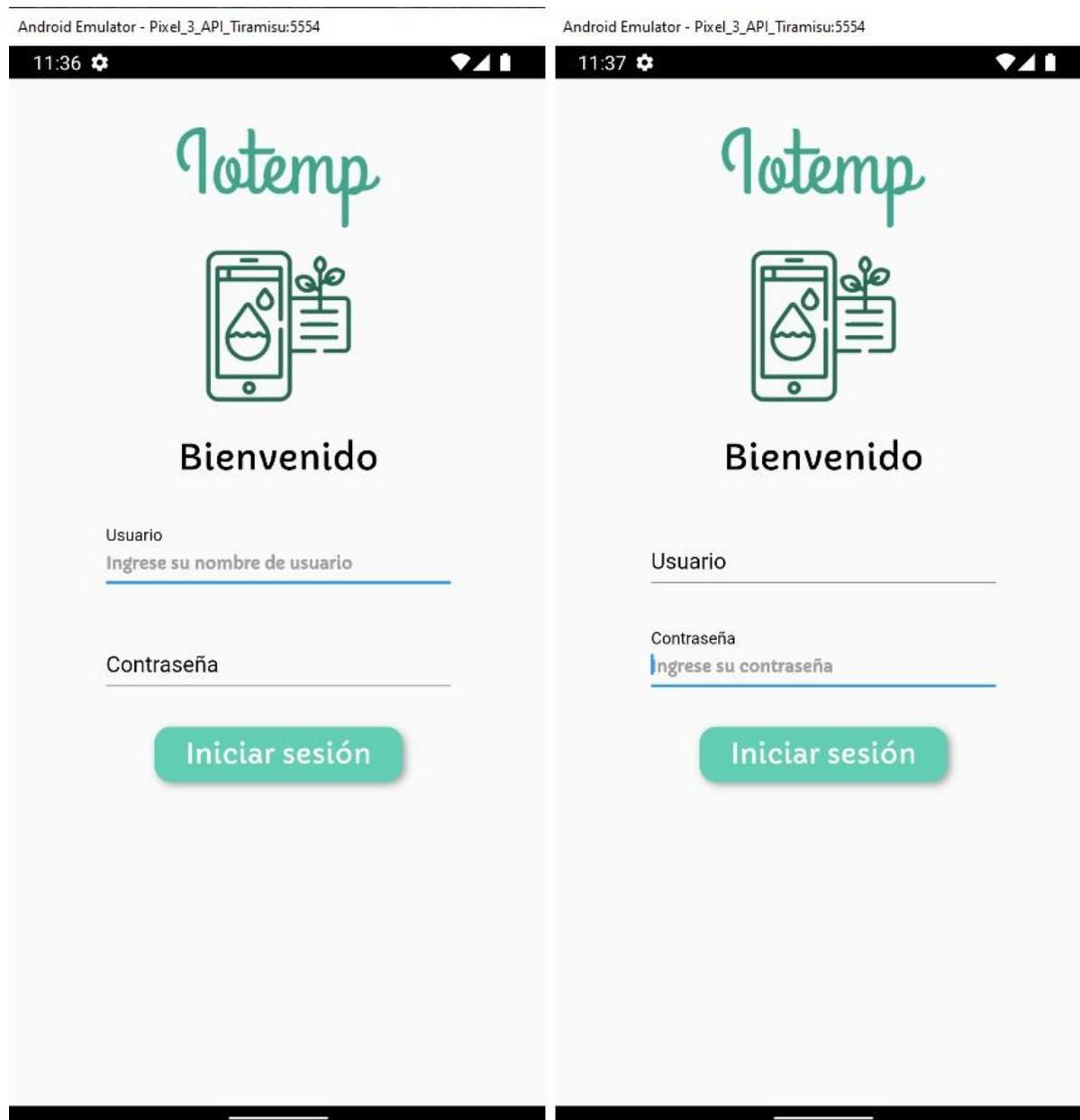


La Figura 35, muestra el proceso de diseño para la pantalla de inicio de sesión, pensada para ser simple, requiriendo que el usuario proporcione un nombre de usuario y contraseña.

**Tarea: Implementación de la pantalla de inicio de sesión**

**Figura 36**

*Implementación del inicio de sesión*



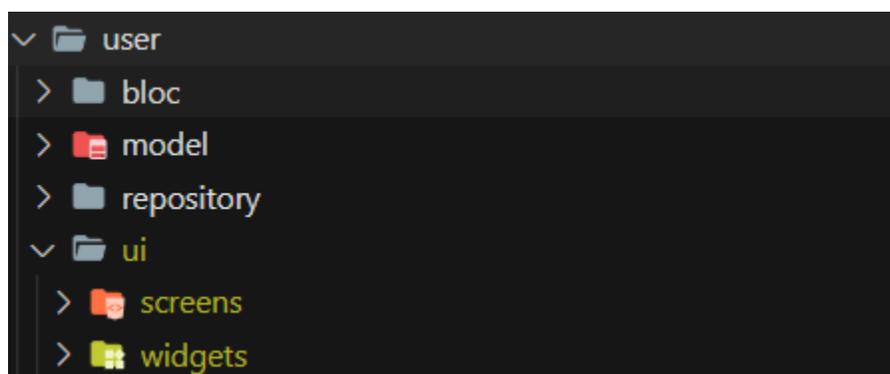
La Figura 36, muestra la implementación de la pantalla de inicio de sesión en base al diseño previamente elaborado.

### ***Tarea: Conexión de la aplicación móvil con el servidor***

La conexión con el servidor fue implementada bajo la estructura del patrón bloc, teniendo un repository encargado de la conexión, un bloc encargado de tratar la información y proveerla a la aplicación y los componentes de ui de la pantalla de login encargados de reflejar el estado de dicha información. De igual forma se añadió un componente model (modelo) Figura 37, en el cual se definió una estructura para la información del usuario, mediante la cual en un desarrollo tardío sería posible gestionar la aplicación en base al tipo de usuario que ingrese a la aplicación.

### **Figura 37**

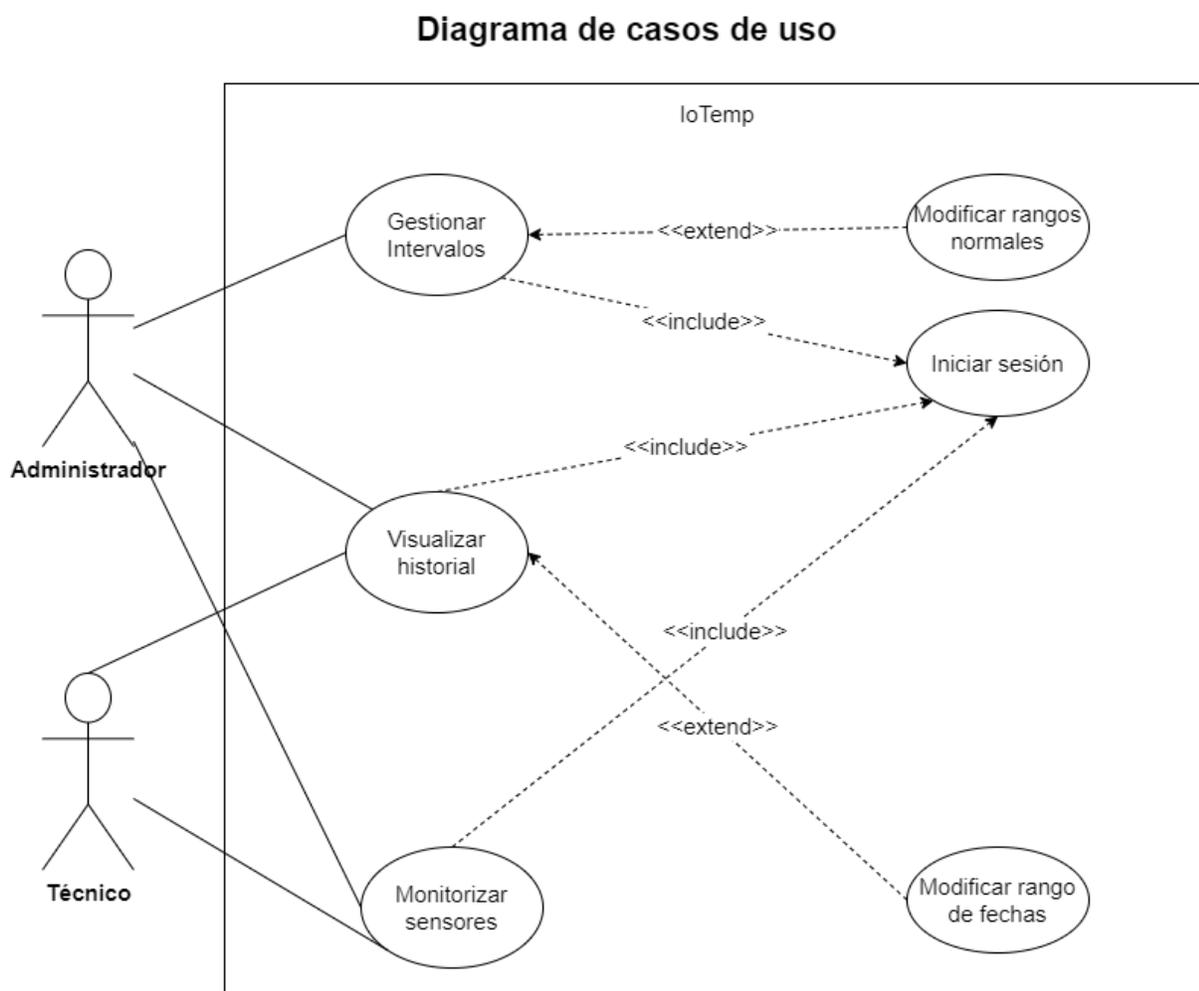
*Estructuración del proyecto para el inicio de sesión*



**Tarea: Configuración de roles en la aplicación móvil**

**Figura 38**

Diagrama de casos de uso del sistema

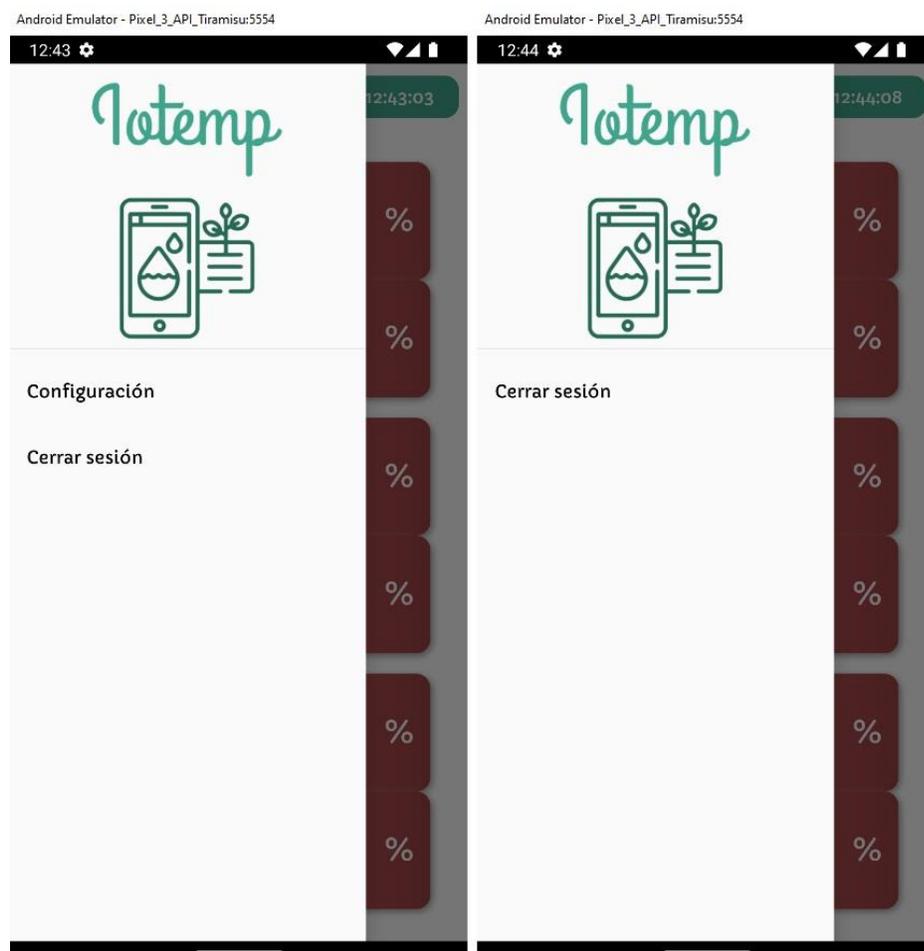


La Figura 38, muestra las funcionalidades a las que tiene acceso cada tipo de usuario en el sistema, dividido en tres módulos principales los cuales requieren de un inicio de sesión obligatorio. El componente de gestión de intervalos abarca la modificación de rangos normales, tarea que solo puede ser llevada a cabo por el identificado como administrador del sistema, mientras que las demás funcionalidades pueden ser utilizadas por ambos tipos de usuario.

Para esto se ha desarrollado en la aplicación un menú en el cual se tenga acceso a las funcionalidades de la aplicación Figura 39, el cual cambia dependiendo del tipo de usuario que ha iniciado sesión.

### Figura 39

#### *Menú de la aplicación*



La Figura 39, muestra cómo se diferencia la aplicación para cada tipo de usuario, limitando las funcionalidades para un rol técnico el cual tendrá acceso únicamente al despliegue de la información y conocimiento del historial de mediciones, dejando la tarea de configurar los rangos normales y por ende las condiciones para el envío de alertas al rol administrador.

### Tarea: Programación API para manejo de la data histórica

Figura 40

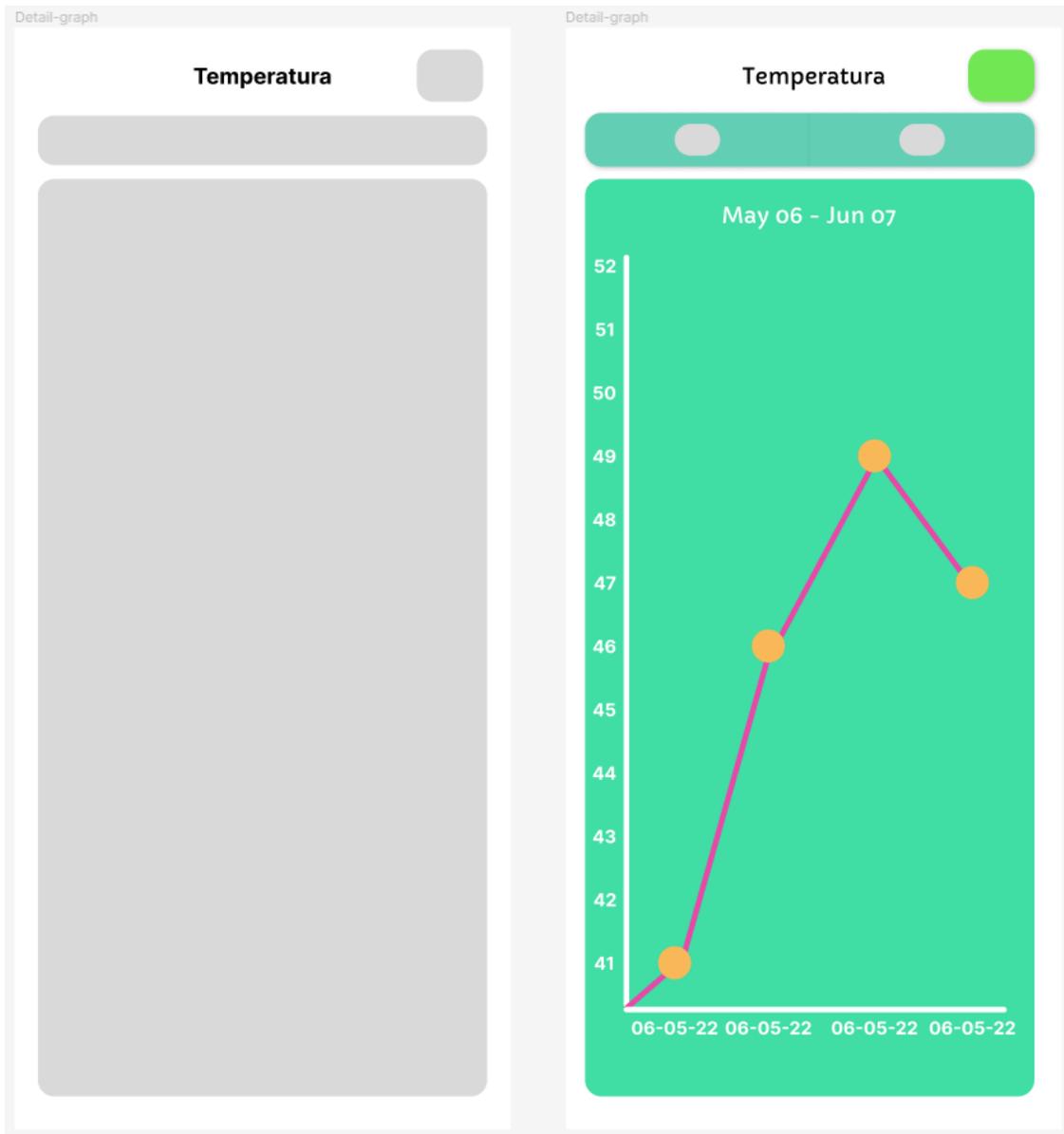
Código de la API para manejo de los datos históricos

```
api > src > rutas > data.php > ...
1  <?php
2
3  use \Psr\Http\Message\ServerRequestInterface as Request;
4  use \Psr\Http\Message\ResponseInterface as Response;
5  use Firebase\JWT\JWT;
6
7  // Valores Historicos
8
9  // Promedio diario entre fechas determinadas
10 > $app->get('/data/pd/zn', function(Request $request, Response $response){ ...
176 });
177
```

*Nota.* Declaración e implementación de la funcionalidad para el manejo de los datos históricos almacenados en un determinado periodo de tiempo.

La funcionalidad para el manejo de los datos históricos es esencial ya que conforma una de las partes importantes de la aplicación, la complejidad de la misma radica en el manejo de los mensajes recibidos, de forma nativa el plugin de backend de EMQ guarda la carga del mensaje mqtt en formato BLOB. El primer tratamiento se realiza mediante SQL donde la sentencia ejecutada debe filtrar el topic y la fecha de recepción.

El segundo paso es la transformación del archivo BLOB a una variable de tipo json para posteriormente mediante técnicas de estructuras de datos ir almacenando e ir realizando los cálculos pertinentes hasta obtener los promedios diarios.

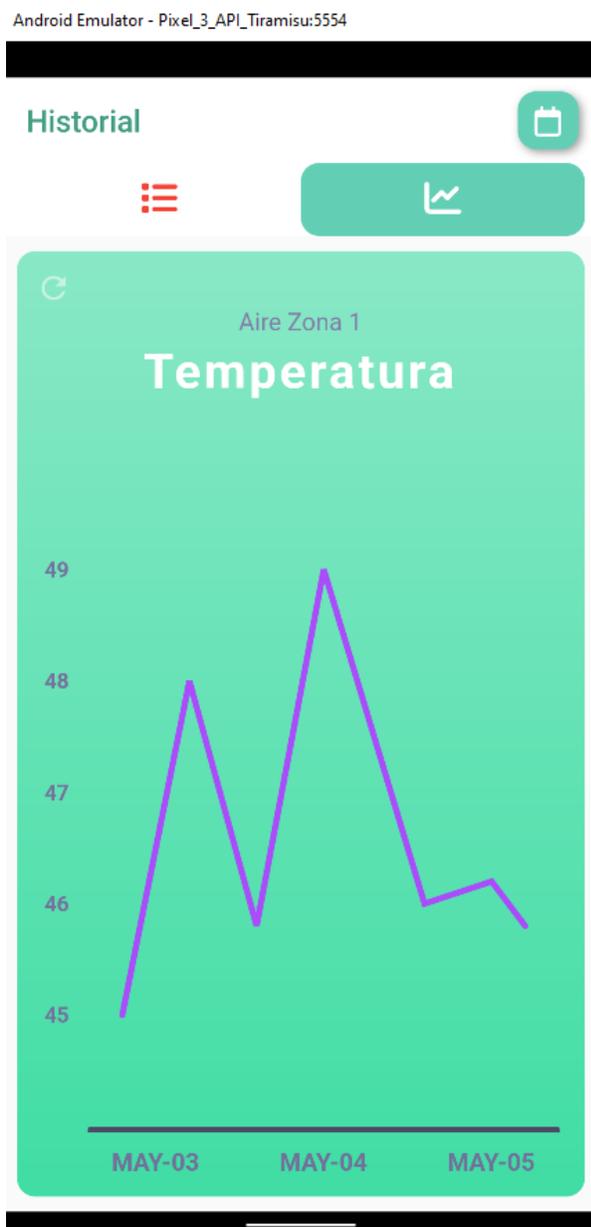
**Tarea: Diseño de la pantalla estadística****Figura 41***Diseño de la pantalla estadística*

Como puede observarse en la Figura 41, la pantalla estadística ha sido diseñada para desplegar un gráfico de líneas el cual sea capaz de desplegar las mediciones obtenidas en un período de tiempo seleccionable.

### Tarea: Implementación de la pantalla estadística

Figura 42

Implementación de la pantalla de gráfico

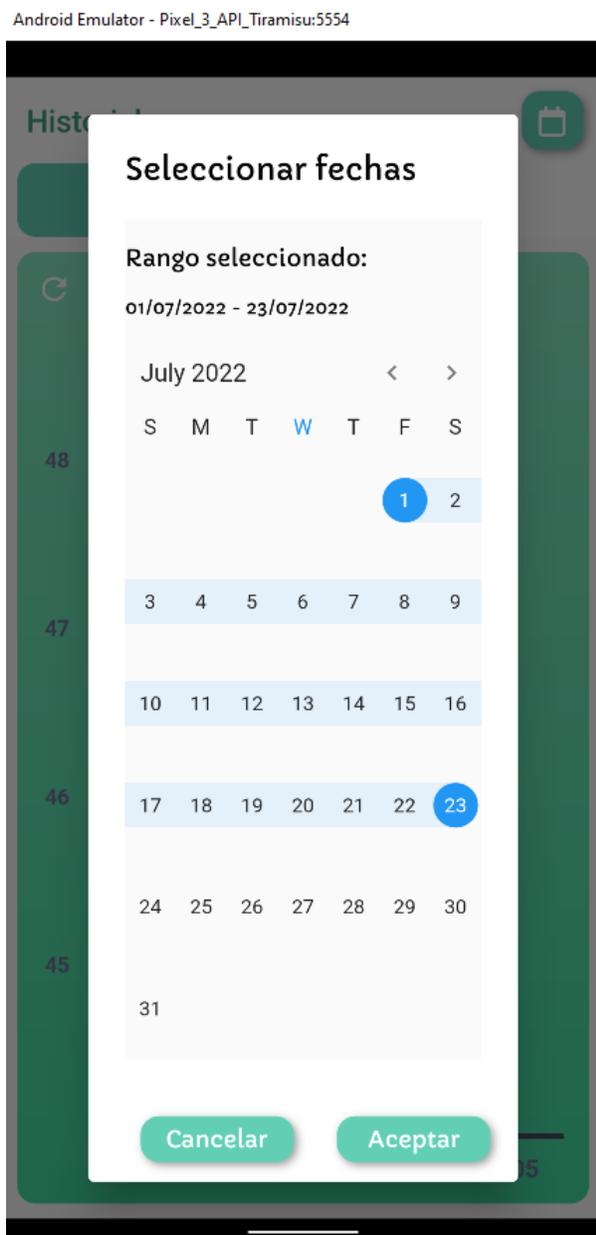


Como puede observarse en la Figura 42, se implementó un gráfico de líneas para la representación histórica de las mediciones de cada sensor, esta se presenta en función de un rango de tiempo modificable

### Tarea: Implementación del selector de fechas

Figura 43

Implementación del selector de fechas



Como se muestra en la Figura 43, se implementó un selector de fechas en formato de calendario para la pantalla de reporte estadístico, dichos valores son utilizados para obtener mediciones almacenadas en una base de datos y mostrarlas en el gráfico de líneas.

### Tarea: Recepción de datos desde el servidor

Figura 44

Recepción de datos históricos

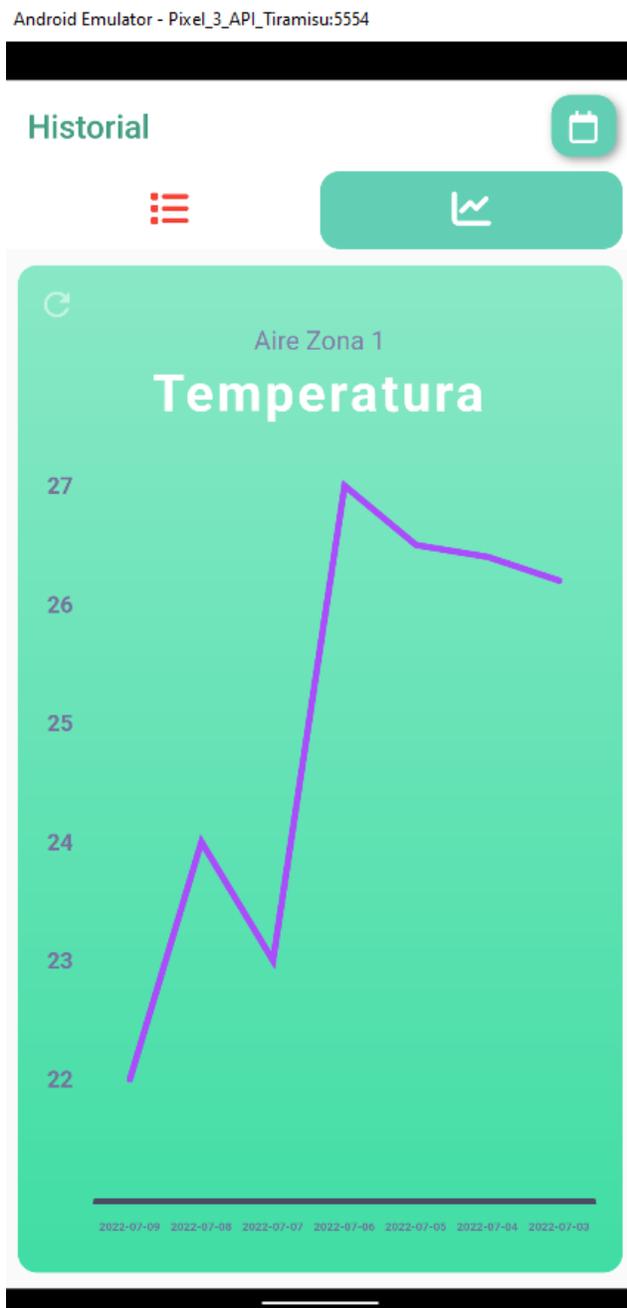
```
[
  {
    Fecha: 2022-07-09,
    TempUZ1: 15.75, TempDZ1: 22.58,
    TempUZ2: 16.44, TempDZ2: 14.33,
    TempUZ3: 24.35, TempDZ3: 20.54,
    HumUZ1: 67.85, HumDZ1: 96.34,
    HumUZ2: 60.54, HumDZ2: 59.4,
    HumUZ3: 82.36, HumDZ3: 80.95
  },
  {
    Fecha: 2022-07-10,
    TempUZ1: 11.5, TempDZ1: 11.9,
    TempUZ2: 11.5, TempDZ2: 11.9,
    TempUZ3: 11.5, TempDZ3: 12.1,
    HumUZ1: 99.9, HumDZ1: 99.9,
    HumUZ2: 99.9, HumDZ2: 99.9,
    HumUZ3: 99.9, HumDZ3: 99.9
  },
]
```

La Figura 44, muestra los resultados obtenidos al realizar una petición de datos históricos los cuales presentan la fecha y las medias de las medidas obtenidas en dicha fecha, estos datos son procesados y desplegados en el gráfico de líneas.

### Tarea: Despliegue de la información del servidor en el gráfico de líneas

Figura 45

Despliegue de la información del servidor en el gráfico de líneas



Como se muestra en la Figura 45, la aplicación obtiene datos desde la base de datos, estos son desplegados en el gráfico de líneas distribuyendo la información de cada zona y tipo de medición.

### Sprint 3

**Tarea: Programación API para almacenamiento de la configuración de los rangos normales.**

#### Figura 46

Código de la API para almacenamiento de la configuración de los rangos normales

```
api > src > rutas > ranges.php > ...
1  <?php
2
3  use \Psr\Http\Message\ServerRequestInterface as Request;
4  use \Psr\Http\Message\ResponseInterface as Response;
5  use Firebase\JWT\JWT;
6
7  // Rangos
8
9  // Actualizacion de Rangos
10 > $app->post('/ranges/up', function(Request $request, Response $response){ ...
100  });
101
102 // Lectura de rangos actuales
103 > $app->get('/ranges/rd', function(Request $request, Response $response){ ...
154  });
155
```

*Nota.* Declaración e implementación de las principales funcionalidades para el apartado de rangos normales.

Para este apartado de la API es necesario incluir dentro de las cabeceras el rol de la persona que realiza la petición ya que existe la limitación de que solamente el administrador del invernadero, es decir, el ingeniero agrónomo es el único que puede modificarlos. Como se puede observar en la Figura 46 existen dos funcionalidades:

- UP: Esta funcionalidad permite realizar un cambio a los rangos normales que existen actualmente en la base de datos, realiza un update a la base de datos con los nuevos valores recibidos en el body de la petición.

- RD: Esta funcionalidad permite obtener los rangos normales registrados en la base de datos, esta ruta es utilizada tanto por la aplicación como por el script que realiza las alertas mediante Firebase.

**Tarea: Diseño de la pantalla de configuración de intervalos**

**Figura 47**

*Diseño de la pantalla de configuración*

The figure displays two wireframe designs of a configuration screen. The left wireframe is a placeholder with a title 'Configuración', a toggle switch, a large grey rectangular area, and a bottom button. The right wireframe is the final design, titled 'Configuración' with a green toggle switch. It features a green header bar with two sliders. Below this, there are three sections for temperature ranges, labeled '1', '2', and '3'. Each section has two columns: 'Temperatura °C min' and 'Temperatura °C max'. Each column contains two input fields with a range slider between them. A green 'Guardar' button is located at the bottom right.

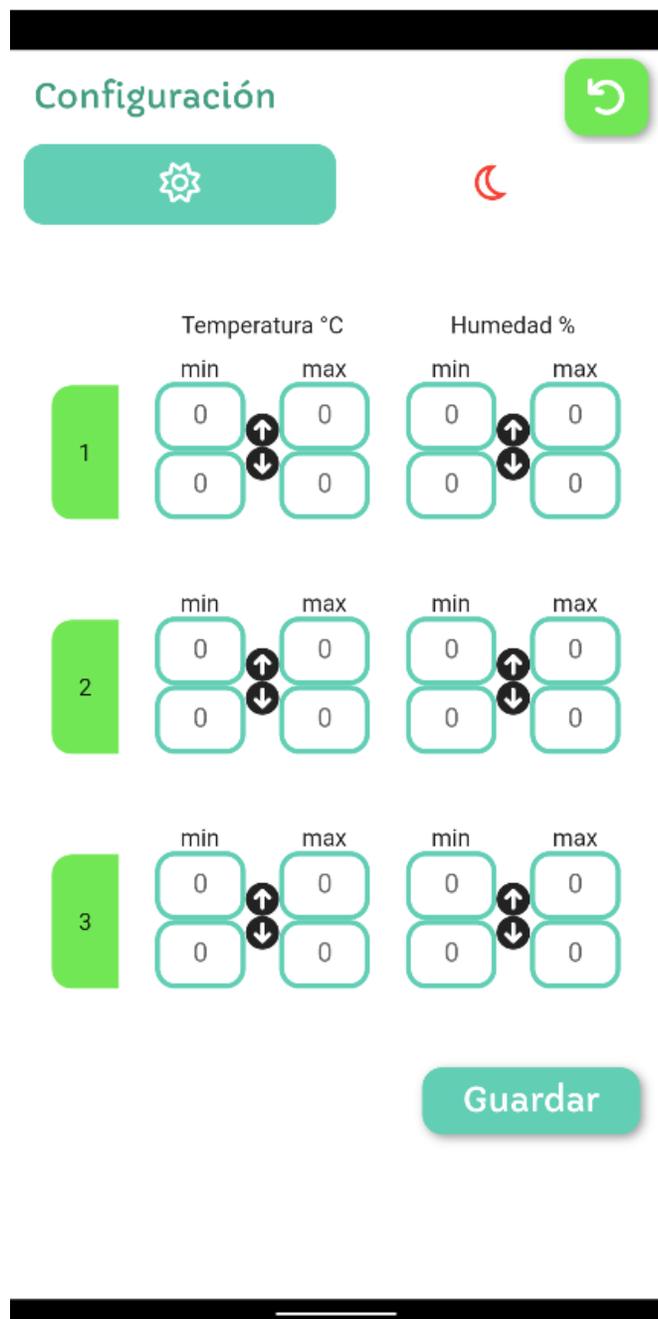
La Figura 47, muestra el diseño para la pantalla de configuración de intervalos, esta consiste en formulario con campos para cada uno de los valores máximos y mínimos de temperatura y humedad de cada sensor, tomando además en cuenta que los valores entre el día y la noche pueden diferir.

### Tarea: Implementación de la pantalla de configuración de intervalos

Figura 48

Implementación de la pantalla de configuración

Android Emulator - Pixel\_3\_API\_Tiramisu:5554



Nota. Pantalla de configuración de rangos para los distintos tipos de variables.

La aplicación permite al usuario definir los rangos considerados como normales, tanto para temperatura como humedad relativa, para ello se ha tomado en cuenta la variación de intervalos de normalidad en el día como en la noche, es por ello que el usuario deberá asignar valores para esos períodos, pudiendo especificar valores independientes para cada una de las zonas establecidas en el diseño de la distribución de sensores. Estos valores afectan directamente en cómo se presenta la información en la pantalla general de la aplicación Figura 48, así como el comportamiento de las alertas enviadas.

***Tarea: Recepción de los datos de intervalos del servidor***

**Figura 49**

*Datos de rangos obtenidos del servidor*

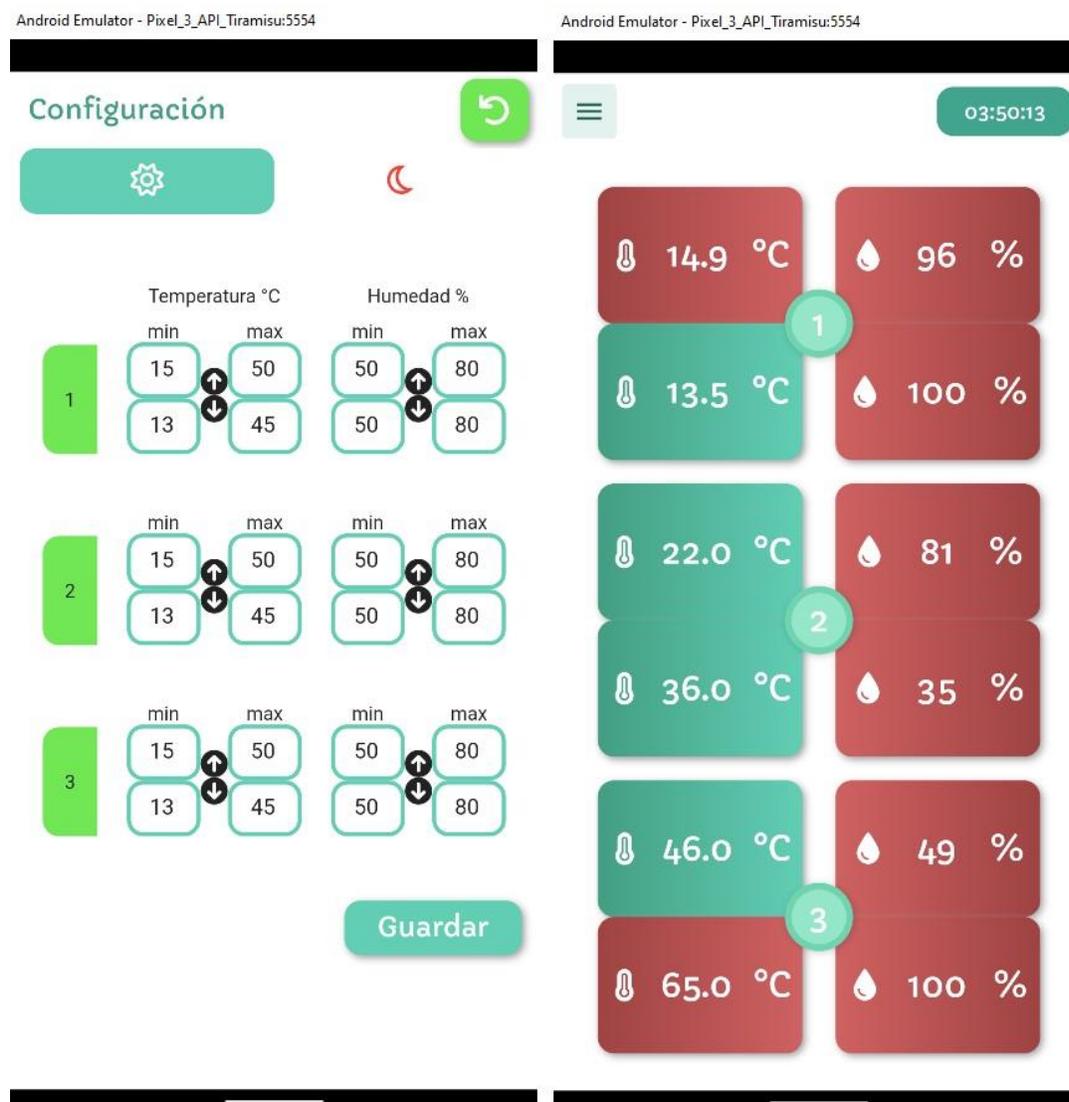
```
{  
  "dtem_min_up": "18",  
  "dtem_min_do": "18",  
  "dtem_max_up": "24",  
  "dtem_max_do": "24",  
  "dhum_min_up": "50",  
  "dhum_min_do": "50",  
  "dhum_max_up": "70",  
  "dhum_max_do": "70",  
  "ntem_min_up": "18",  
  "ntem_min_do": "18",  
  "ntem_max_up": "24",  
  "ntem_max_do": "24",  
  "nhum_min_up": "50",  
  "nhum_min_do": "50",  
  "nhum_max_up": "70",  
  "nhum_max_do": "70"  
}
```

Como se muestra en la Figura 49, los rangos son almacenados en una base de datos los cuales son obtenidos por la aplicación móvil, estos datos son obtenidos tanto para la configuración de día y de noche.

### **Tarea: Almacenamiento de nuevos intervalos en el dispositivo**

**Figura 50**

#### *Almacenamiento de intervalos*



*Nota.* Almacenamiento de los intervalos y validación en el resumen.

La Figura 50 muestra la configuración de los intervalos de aplicación y su incidencia en la presentación de los datos, estando el diseño de los módulos de medición en función de las mediciones obtenidas por los sensores y los intervalos de normalidad establecidos por el administrador del sistema.

**Tarea: Programación SCRIPT para escucha y manejo de los rangos normales para alertar anomalías**

**Figura 51**

*Script de escucha y alerta para anomalías en rangos*

```
E: > Programación > FTP-Servidor > Simuladores > ListenerAlerts.py > ...
1  import json
2  import time
3  import requests
4  import paho.mqtt.client as mqtt
5
6  > class Valores: ...
73
74  > def conectar_mqtt(): ...
84
85  > def on_connect(client, userdata, flags, rc): ...
92
93  > def on_message(client, userdata, message): ...
100
101 > def comprobacion(res): ...
195
196 > def enviar_alerta(men): ...
200
201 #Principal
202 val = Valores()
203 val.inicializar_rangos()
204 client = conectar_mqtt()
205 val.imprimir_valores()
206
207 while True:
208     client.loop_start()
```

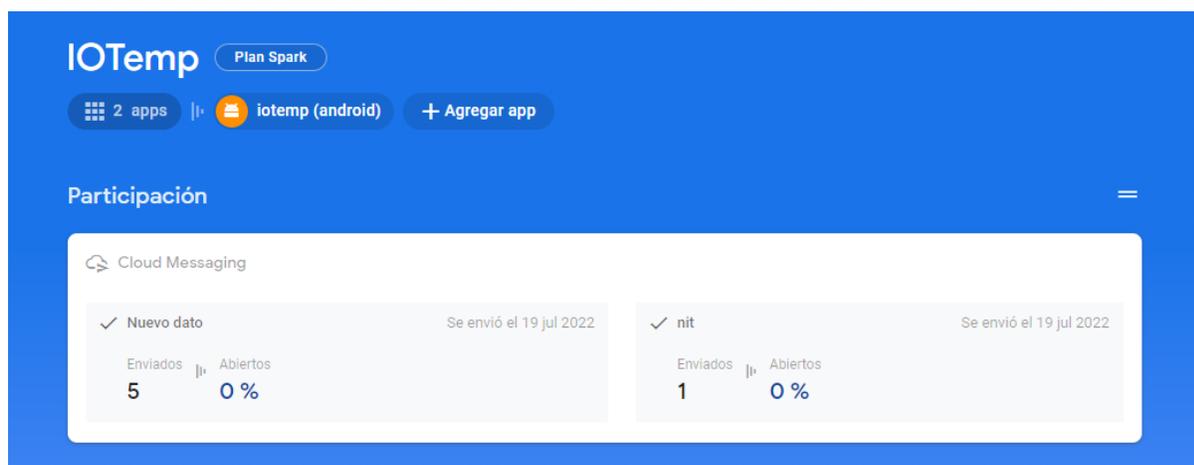
*Nota.* Código desarrollado para la alerta de anomalías durante la recepción de mensajes desde el dispositivo IoT escrito en Python.

La aplicación puede manejar las alertas de forma local cuando está en primer o segundo plano, sin embargo, cuando está cerrada la forma de manejar las alertas se debe hacer mediante firebase por lo que se desarrolló este script. Lo que realiza primero es inicializar los diferentes valores de los rangos normales que se encuentran registrados en la base de datos, para posteriormente conectarse y suscribirse al tópico donde se reciben los mensajes desde el dispositivo IoT, por cada mensaje compara el contenido con los respectivos rangos para en caso de existir una anomalía realice una petición a firebase para que pueda notificar a los clientes con las aplicaciones instaladas.

### ***Tarea: Configuración del servidor con el servicio de Firebase para uso de Alertas***

**Figura 52**

*Panel de cloud messaging en firebase console*



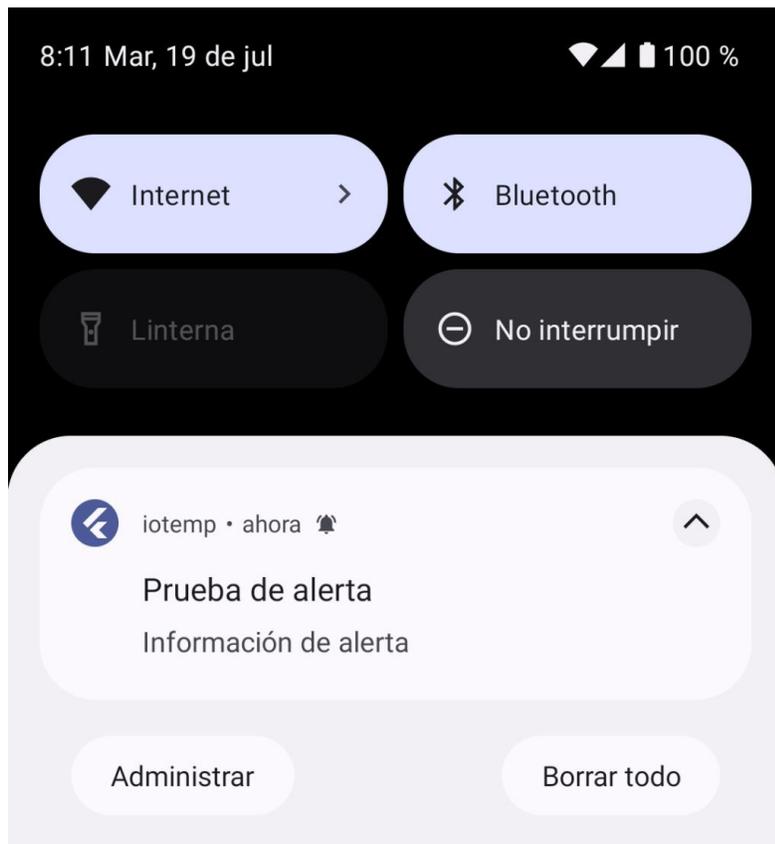
Para el envío de notificaciones se hizo uso cloud messaging de firebase Figura 52, mediante este servicio gratuito se ha creado un tema o tópico al cual se suscriben todos los usuarios con la aplicación en su dispositivo lo cual permite publicar notificaciones cuando se ha presentado una medición anormal captada por los sensores.

**Tarea: Configuración de la aplicación para el envío de alertas**

Para el despliegue de aletas se ha permitido a la aplicación desplegarlas mientras está en funcionamiento, esto implica que cuando se haya registrado una medición anormal, se visualizará una notificación indicando el problema. Este comportamiento es posible gracias a la librería flutter cloud messaging Figura 52.

**Tarea: Configuración de la aplicación para la recepción de alertas****Figura 53**

*Recepción de alerta con firebase*



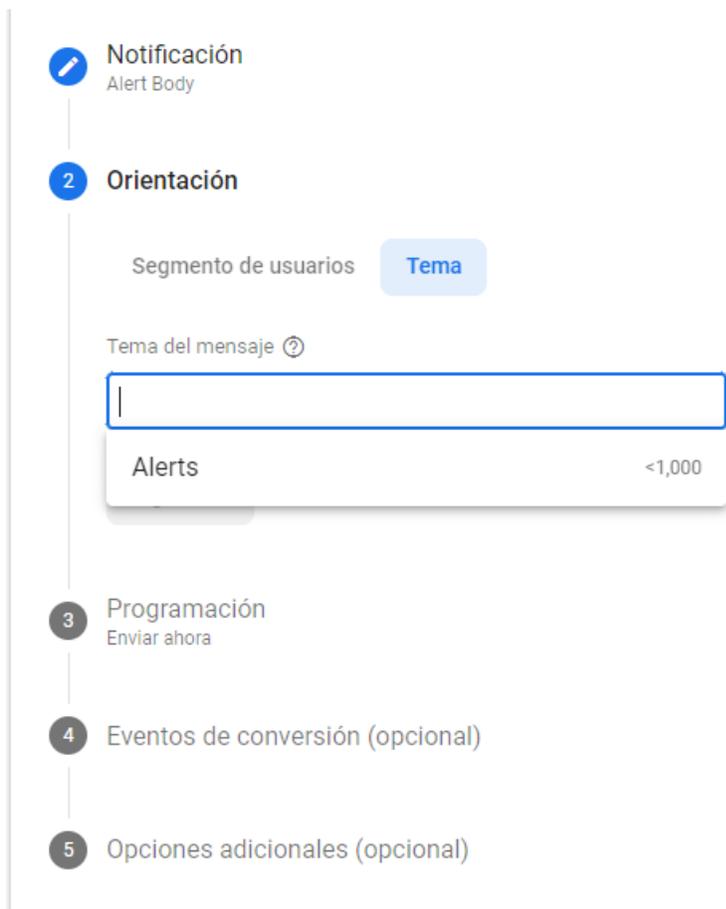
Mediante el uso de la librería flutter cloud messaging, la aplicación recibe alertas cuando está funcionando de fondo o cuando está cerrada, estas notificaciones informan de mediciones anormales y al momento de ser presionadas inician la aplicación en caso de que esta haya sido retirada de los procesos en segundo plano.

### **Tarea: Configuración de la aplicación para funcionamiento en segundo plano**

La configuración del servidor ha sido destinada para disparar un evento en el servicio de cloud messaging encargado de enviar las alertas a los dispositivos suscritos al mismo. Para ello se ha destinado el tópic "Alerts", al cual se suscriben y envían todas las alertas.

**Figura 54**

*Configuración de una alerta en firebase*



The image shows a vertical sequence of five steps in the Firebase console for configuring a notification:

- 1** Notificación  
Alert Body
- 2** Orientación  
Segmento de usuarios **Tema**  
Tema del mensaje ⓘ  
Alerts <1,000
- 3** Programación  
Enviar ahora
- 4** Eventos de conversión (opcional)
- 5** Opciones adicionales (opcional)

## Fase 04 Revisión y Retrospectiva

Como lo detalla la aplicación del marco de trabajo scrum al final de cada sprint se llevó a cabo una revisión del estado del proyecto, como este se llevó a cabo y cómo evolucionaron las tareas, representando en un gráfico dicha evolución.

### *Sprint 1*

Al finalizar el Sprint 1 procedemos a realizar una lista de chequeo sobre los criterios de aceptación descritos en las historias de usuario en las que se trabajó en dicho Sprint, al realizar esta validación corroboramos el cumplimiento de la funcionalidad propuesta en sistema software.

### Tabla 14

*Lista de chequeo de los criterios de aceptación del Sprint 1*

Lista de Chequeo				
Historia de Usuario	Criterio de Aceptación	Sí	No	Observación
HU-001	Temperatura	X		
HU-002	Humedad	X		
HU-003	Resumen	X		

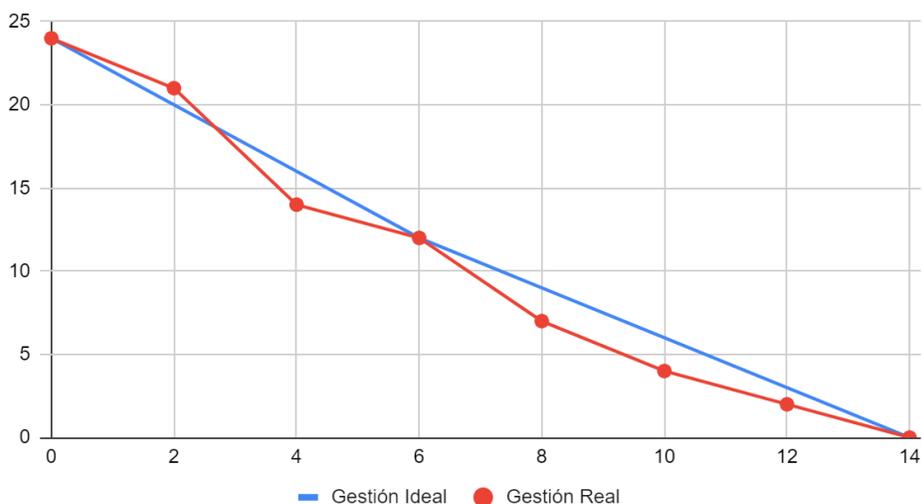
*Nota.* Lista de chequeo correspondiente a la revisión de los criterios de aceptación del sprint 1

Como se puede observar en la Tabla 14, se cumplió con los criterios de aceptación de las tres historias de usuario, por lo tanto, a nivel de funcionalidad se desarrolló lo que se planificó al momento de realizar el Sprint.

## Figura 55

### Burndown chart del sprint 1

Burndown Chart Sprint 1



*Nota.* Evolución del cumplimiento de tareas a lo largo del sprint 1,

Como puede observarse en la Figura 55, el burndown chart presenta una trazada ideal del cumplimiento de las tareas, así como el proceso real el cual fue actualizado conforme se cumplían los objetivos y se presentaban inconvenientes, llegando al final donde se logró cumplir con la planificación respectiva de las tareas a cumplir en el tiempo especificado.

### **Sprint 2**

Al finalizar el Sprint 2 procedemos a realizar una lista de chequeo sobre los criterios de aceptación descritos en las historias de usuario en las que se trabajó en dicho Sprint, al realizar esta validación corroboramos el cumplimiento de la funcionalidad propuesta en sistema software.

**Tabla 15**

*Lista de chequeo de los criterios de aceptación del Sprint 2*

Lista de Chequeo				
Historia de Usuario	Criterio de Aceptación	Sí	No	Observación
HU-007	Credenciales	X		
HU-004	Gráfico	X		

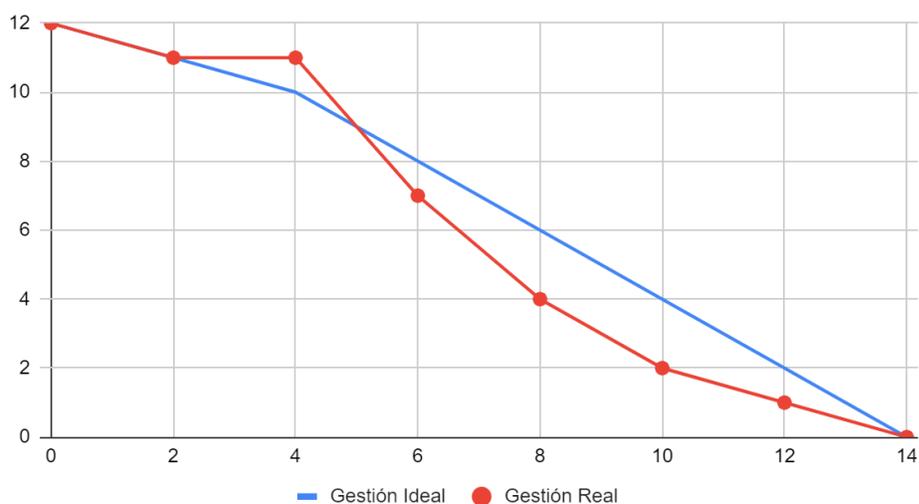
*Nota.* Lista de chequeo correspondiente a la revisión de los criterios de aceptación del sprint 2

Como se puede observar en la Tabla 15, se cumplió con los criterios de aceptación de las dos historias de usuario, por lo tanto, a nivel de funcionalidad se desarrolló lo que se planificó al momento de realizar el Sprint.

**Figura 56**

*Burndown chart del sprint 2*

Burndown Chart Sprint 2



*Nota.* Evolución del cumplimiento de tareas a lo largo del sprint 2.

### **Sprint 3**

Al finalizar el Sprint 3 procedemos a realizar una lista de chequeo sobre los criterios de aceptación descritos en las historias de usuario en las que se trabajó en dicho Sprint, al realizar esta validación corroboramos el cumplimiento de la funcionalidad propuesta en sistema software.

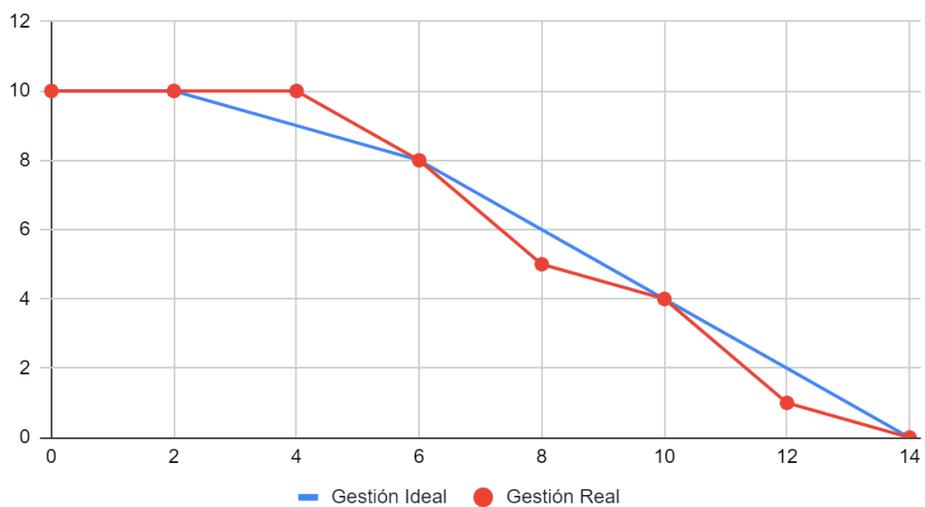
**Tabla 16**

*Lista de chequeo de los criterios de aceptación del Sprint 3*

<b>Lista de Chequeo</b>				
<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Sí</b>	<b>No</b>	<b>Observación</b>
HU-005	Intervalos de normalidad	X		
HU-006	Alertas	X		

*Nota.* Lista de chequeo correspondiente a la revisión de los criterios de aceptación del sprint 3.

Como se puede observar en la Tabla 16, se cumplió con los criterios de aceptación de las dos historias de usuario, por lo tanto, a nivel de funcionalidad se desarrolló lo que se planificó al momento de realizar el Sprint.

**Figura 57***Burndown chart del sprint 3***Burndown Chart Sprint 3**

*Nota.* Evolución del cumplimiento de tareas a lo largo del sprint 3,

## Fase 05 Liberación

### Despliegue de la aplicación en la Play Store

#### Figura 58

Envío de la aplicación a la Play Store



Nota. Captura de pantalla tomada desde la Play Store de Android

Para el despliegue se hizo uso de la Play Store de Android, mediante la cual se puso a disposición la aplicación a los usuarios de una manera de fácil acceso y a través de la cual podrán recibir actualizaciones futuras, pudiendo además controlar el estado de la aplicación, su uso, tasa de fallas y otras métricas que proporciona el servicio de aplicaciones de Play Store.

## Capítulo IV

### Validación

Se realizará las respectivas validaciones al sistema software que al final de su desarrollo fueron liberadas e implementadas de la siguiente manera: el servidor web se implementó en las instancias EC2 de AWS, la aplicación móvil se distribuye gratuitamente en la tienda de Play Store para Android y la infraestructura IoT se instaló en el centro agropecuario “Pilonera Verito” donde el dispositivo genera un total de 4 lecturas de variables ambientales por cada hora.

Se comenzará realizando una validación del sistema mediante una lista de chequeo de los criterios de aceptación expuestos en las historias de usuario del capítulo III, se proseguirá con tres análisis estadísticos con el objetivo de responder a la hipótesis planteada en el proyecto mediante el cumplimiento de los indicadores.

#### **Indicador Precisión de datos**

El primer indicador planteado evalúa la precisión de los datos monitorizados, este indicador nos permite comparar la precisión de los datos obtenidos mediante el proceso manual con el uso del sistema software.

Para ejecutar esta prueba se analizó los datos de dos horas obtenidos mediante el dispositivo IoT, que a la par se midió con el dispositivo HTC-2 que era utilizado en el proceso manual del invernadero.

Se aplicó la prueba T-Student para muestras independientes con  $(n_1+n_2-2)$  grados de libertad con un nivel de confianza del 95%, obteniendo como resultado las siguientes hipótesis.

Hipótesis nula (H0): La media de los datos son iguales

Hipótesis alternativa (H1): La media de los datos no son iguales

Los datos utilizados para esta prueba se presentan en la Tabla 15 y Tabla 16, donde se realizan dos pruebas, una por cada variable ambiental medida.

### **Temperatura**

**Tabla 17**

*Valores de Temperatura de monitorización manual y automatizado*

<b>Temperatura</b>		
<b>N° Medición</b>	<b>Manual</b>	<b>Automatizado</b>
<b>1</b>	9,9	10,3
<b>2</b>	9,7	10,2
<b>3</b>	10,9	11,4
<b>4</b>	9,9	10,4
<b>5</b>	10,7	11,4
<b>6</b>	8,4	8,8
<b>7</b>	9,3	10,6
<b>8</b>	10,3	9,2
<b>9</b>	8,9	10,1
<b>10</b>	10,6	9,6

Para el cálculo del estadístico de las mediciones manuales y automatizadas se realizó el siguiente proceso:

$$t = \frac{\underline{X}_1 - \underline{X}_2}{\sqrt{\frac{S_c^2}{n_1} + \frac{S_c^2}{n_2}}}$$

$t$ : Estadístico  $t$  calculado.

$\underline{X}_1; \underline{X}_2$ : medias muestrales

$$S_c^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \text{ varianza común}$$

$S_n^2$  varianza muestral

$n_1$ : número de muestras de mediciones manuales

$n_2$ : número de muestras de mediciones automatizadas

El proceso requiere tomar en cuenta las mediciones obtenidas tanto de forma manual, mediante un termómetro especializado, y de forma automatizada, mediante sensores.

$$n_1 = 10$$

$$n_2 = 10$$

$$\underline{X}_1 = 9,85$$

$$\underline{X}_2 = 10,2$$

$$S_1^2 = 0,658$$

$$S_2^2 = 0,713$$

$$S_c^2 = \frac{(10 - 1)0,658 + (10 - 1)0,713}{10 + 10 - 2} = 0,686$$

Al reemplazar los datos en la prueba de hipótesis T-Student se obtiene:

$$t = \frac{9,85 - 10,2}{\sqrt{\frac{0,686}{10} + \frac{0,686}{10}}} = -0,945$$

Luego se procede a obtener los grados de libertad y el valor crítico:

$$gl = (n_1 + n_2 - 2) \text{ grados de libertad}$$

$$gl = (10 + 10 - 2) = 18$$

$$\alpha = 0,05 \text{ nivel de significancia}$$

$$\alpha/2 = 0,025$$

Para la obtención de los puntos críticos se hizo uso de la tabla t-student, con grados de libertad = 10 y 0.025 resultado del nivel de significancia dividido para dos, siendo este caso en específico una prueba de dos colas

**Figura 59**

*Distribución T-Student para pruebas de dos colas*

		$\alpha/2$						
		0,0005	0,001	0,005	0,01	0,025	0,05	0,1
v grados de libertad	1	636,619	318,309	63,657	31,821	12,706	6,314	3,078
	2	31,599	22,327	9,925	6,965	4,303	2,920	1,886
	3	12,924	10,215	5,841	4,541	3,182	2,353	1,638
	4	8,610	7,173	4,604	3,747	2,776	2,132	1,533
	5	6,869	5,893	4,032	3,365	2,571	2,015	1,476
	6	5,959	5,208	3,707	3,143	2,447	1,943	1,440
	7	5,408	4,785	3,499	2,998	2,365	1,895	1,415
	8	5,041	4,501	3,355	2,896	2,306	1,860	1,397
	9	4,781	4,297	3,250	2,821	2,262	1,833	1,383
	10	4,587	4,144	3,169	2,764	2,228	1,812	1,372
	11	4,437	4,025	3,106	2,718	2,201	1,796	1,363
	12	4,318	3,930	3,055	2,681	2,179	1,782	1,356
	13	4,221	3,852	3,012	2,650	2,160	1,771	1,350
	14	4,140	3,787	2,977	2,624	2,145	1,761	1,345
	15	4,073	3,733	2,947	2,602	2,131	1,753	1,341
	16	4,015	3,686	2,921	2,583	2,120	1,746	1,337
	17	3,965	3,646	2,898	2,567	2,110	1,740	1,333
	18	3,922	3,610	2,878	2,552	2,101	1,734	1,330
	19	3,883	3,579	2,861	2,539	2,093	1,729	1,328
	20	3,850	3,552	2,845	2,528	2,086	1,725	1,325
	21	3,819	3,527	2,831	2,518	2,080	1,721	1,323
	22	3,792	3,505	2,819	2,508	2,074	1,717	1,321
	23	3,768	3,485	2,807	2,500	2,069	1,714	1,319

Nota. Tabla de distribución T-Student para pruebas de dos colas. Tomado de (Sánchez, 2015)

La comprobación de los cálculos se realizó mediante la herramienta de “Análisis de Datos” en el programa informático “Excel” perteneciente a Microsoft, los resultados obtenidos se pueden observar en la siguiente Tabla

**Tabla 18**

*Resultados de la Prueba t obtenidos por Análisis de Datos en Excel*

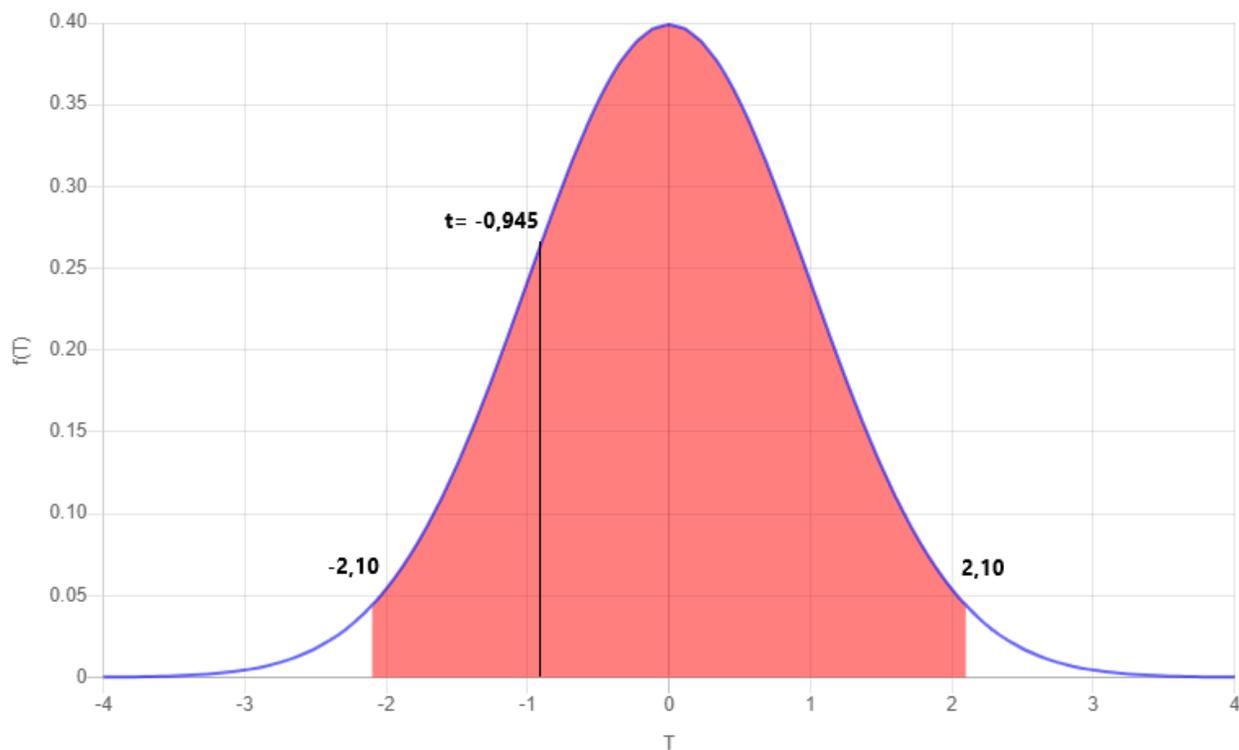
	<b>Manual</b>	<b>Automatizado</b>
Media	9,85	10,2
Varianza	0,658333333	0,713333333
Observaciones	10	10
Varianza agrupada	0,685833333	
Diferencia hipotética de las medias	0	
Grados de libertad	18	
Estadístico t	-0,945025989	
P(T<=t) dos colas	0,357159391	
Valor crítico de t (dos colas)	2,10092204	

*Nota.* Resultados generados mediante la herramienta Análisis de Datos en Excel.

A continuación, se realizó la gráfica correspondiente a la Prueba T- Student

**Figura 60**

*Gráfica T Student para la variable de temperatura*



Basándonos en los cálculos realizados y comprobados por la herramienta estadística llegamos a obtener la gráfica presentada en la Figura 60, al estar el valor  $t$  dentro de los valores críticos de las colas damos por aceptada la hipótesis nula la cual indicaba: las medias de los valores son iguales.

**Humedad****Tabla 19***Valores de Temperatura de monitorización manual y automatizado*

<b>Humedad</b>		
<b>N° Medición</b>	<b>Manual</b>	<b>Automatizado</b>
1	96	99
2	97	98
3	97	99
4	98	97
5	96	98
6	98	97
7	99	96
8	96	97
9	97	99
10	96	98

Para el cálculo del estadístico de las mediciones manuales y automatizadas se realizó el siguiente proceso:

$$t = \frac{\underline{X}_1 - \underline{X}_2}{\sqrt{\frac{S_c^2}{n_1} + \frac{S_c^2}{n_2}}}$$

*t*: Estadístico *t* calculado.

$\underline{X}_1$ ;  $\underline{X}_2$ : medias muestrales

$$S_c^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \text{ varianza común}$$

$S_n^2$  varianza muestral

$n_1$ : número de muestras de mediciones manuales

$n_2$ : número de muestras de mediciones automatizadas

El proceso requiere tomar en cuenta las mediciones obtenidas tanto de forma manual, mediante una herramienta especializada, y de forma automatizada, mediante sensores.

$$n_1 = 10$$

$$n_2 = 10$$

$$\underline{X}_1 = 97$$

$$\underline{X}_2 = 97,8$$

$$S_1^2 = 1.11$$

$$S_2^2 = 1.07$$

$$S_c^2 = \frac{(10 - 1)1.11 + (10 - 1)1.07}{10 + 10 - 2} = 1,09$$

Al reemplazar los datos en la prueba de hipótesis T-Student se obtiene:

$$t = \frac{97 - 97,8}{\sqrt{\frac{1,09}{10} + \frac{1,09}{10}}} = -1,71$$

Luego se procede a obtener los grados de libertad y el valor crítico:

$$gl = (n_1 + n_2 - 2) \text{ grados de libertad}$$

$$gl = (10 + 10 - 2) = 18$$

$$\alpha = 0,05 \text{ nivel de significancia}$$

$$\alpha/2 = 0,025$$

Para la obtención de los puntos críticos se hizo uso de la tabla t-student, con grados de libertad = 10 y 0.025 resultado del nivel de significancia dividido para dos, siendo este caso en específico una prueba de dos colas.

**Figura 61**

*Distribución T-Student para pruebas de dos colas*

		$\alpha/2$						
		0,0005	0,001	0,005	0,01	0,025	0,05	0,1
v grados de libertad	1	636,619	318,309	63,657	31,821	12,706	6,314	3,078
	2	31,599	22,327	9,925	6,965	4,303	2,920	1,886
	3	12,924	10,215	5,841	4,541	3,182	2,353	1,638
	4	8,610	7,173	4,604	3,747	2,776	2,132	1,533
	5	6,869	5,893	4,032	3,365	2,571	2,015	1,476
	6	5,959	5,208	3,707	3,143	2,447	1,943	1,440
	7	5,408	4,785	3,499	2,998	2,365	1,895	1,415
	8	5,041	4,501	3,355	2,896	2,306	1,860	1,397
	9	4,781	4,297	3,250	2,821	2,262	1,833	1,383
	10	4,587	4,144	3,169	2,764	2,228	1,812	1,372
	11	4,437	4,025	3,106	2,718	2,201	1,796	1,363
	12	4,318	3,930	3,055	2,681	2,179	1,782	1,356
	13	4,221	3,852	3,012	2,650	2,160	1,771	1,350
	14	4,140	3,787	2,977	2,624	2,145	1,761	1,345
	15	4,073	3,733	2,947	2,602	2,131	1,753	1,341
	16	4,015	3,686	2,921	2,583	2,120	1,746	1,337
	17	3,965	3,646	2,898	2,567	2,110	1,740	1,333
	18	3,922	3,610	2,878	2,552	2,101	1,734	1,330
	19	3,883	3,579	2,861	2,539	2,093	1,729	1,328
	20	3,850	3,552	2,845	2,528	2,086	1,725	1,325
	21	3,819	3,527	2,831	2,518	2,080	1,721	1,323
	22	3,792	3,505	2,819	2,508	2,074	1,717	1,321
	23	3,768	3,485	2,807	2,500	2,069	1,714	1,319

Nota. Tomado de (Sánchez, 2015)

La comprobación de los cálculos se realizó mediante la herramienta de “Análisis de Datos” en el programa informático “Excel” perteneciente a Microsoft, los resultados obtenidos se pueden observar en la siguiente Tabla

**Tabla 20**

*Resultados de la Prueba t obtenidos por Análisis de Datos en Excel*

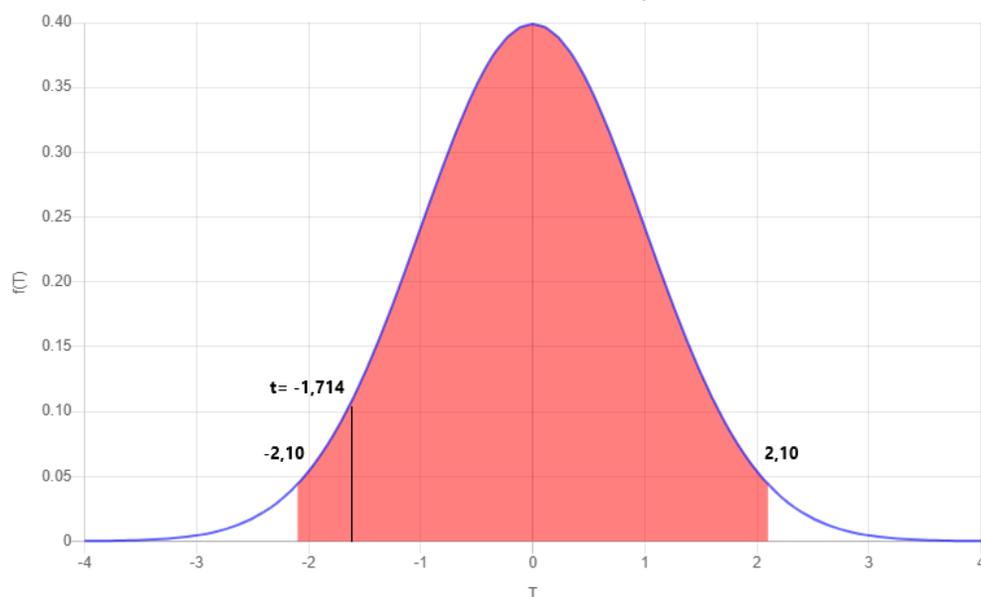
	<b>Manual</b>	<b>Automatizado</b>
Media	97	97,8
Varianza	1,111111111	1,066666667
Observaciones	10	10
Varianza agrupada	1,088888889	
Diferencia hipotética de las medias	0	
Grados de libertad	18	
Estadístico t	-1,714285714	
P(T<=t) dos colas	0,103643016	
Valor crítico de t (dos colas)	2,10092204	

*Nota.* Resultados generados mediante la herramienta Análisis de Datos en Excel.

A continuación, se realizó la gráfica correspondiente a la Prueba T- Student

**Figura 62**

*Gráfica T Student para la variable de humedad*



Basándonos en los cálculos realizados y comprobados por la herramienta estadística llegamos a obtener la gráfica presentada en la Figura 62, al estar el valor  $t$  dentro de los valores críticos de las colas damos por aceptada la hipótesis nula la cual indicaba: las medias de los valores son iguales.

### ***Interpretación de los resultados***

El resultado de las pruebas estadísticas T-Student llevadas a cabo sobre cada variable ambiental fue que se aceptaba la hipótesis nula que indica que las medias de los datos son iguales, por lo tanto, llegamos a verificar que los valores de las variables ambientales medidos tanto por los sensores y por el dispositivo físico utilizado en la medición manual tienen una precisión similar. Esto es positivo con respecto al proyecto ya que significa una alta fidelidad de los valores medidos y siendo de utilidad para conocer el estado del invernadero con el aporte de mediciones fiables para el Ingeniero Agrónomo.

### **Indicador Tiempo empleado**

El tercer indicador planteado evalúa el tiempo empleado en el proceso de la monitorización, este indicador nos permite comparar la cantidad del tiempo empleado durante el proceso tanto manual como con el uso del sistema software.

Para ejecutar esta prueba se cronometró el tiempo empleado en realizar la monitorización de las 3 zonas del invernadero utilizando el método manual y el sistema software.

Se aplicó la prueba T-Student para muestras relacionadas utilizando un valor de significancia del 5%, obteniendo como resultado las siguientes hipótesis.

Hipótesis nula (H0):  $\mu_d \leq 0$

Hipótesis alternativa (H1):  $\mu_d > 0$

Los datos utilizados para esta prueba se presentan en la Tabla 19.

**Tabla 21**

*Valores de tiempo empleado de monitorización manual y automatizado*

N° Medición	Tiempo Empleado	
	Manual	Automatizado
1	10 min 12 seg	20 seg
2	12 min 0 seg	31 seg
3	11 min 8 seg	25 seg
4	10 min 8 seg	38 seg
5	8 min 2 seg	23 seg
6	9 min 1 seg	33 seg
7	8 min 1 seg	34 seg
8	12 min 0 seg	26 seg

Para la realización del análisis se igualaron las unidades a la misma, en este caso minutos

**Tabla 22**

*Valores de Temperatura de monitorización en minutos*

N° Medición	Tiempo Empleado			$(d - \underline{d})^2$
	Manual	Automatizado	Diferencia	
1	10,2 min	0,33 min	9,87 min	0,08
2	12 min	0,52 min	11,48 min	3,58
3	11,13 min	0,42 min	10,71 min	1,26
4	10,13 min	0,63 min	9,5 min	0,01
5	8,03 min	0,38 min	7,65 min	3,74
6	9,02 min	0,55 min	8,47 min	1,26
7	8,02 min	0,57 min	7,45 min	4,58
8	12 min	0,43 min	11,57 min	3,93

Para el cálculo estadístico se realizó el siguiente proceso:

$$t = \frac{\underline{d}}{S_d/\sqrt{n}}$$

$\underline{d}$ : media de las diferencias

$S_d$ : desviación estándar

$n$ : número de muestras

$\underline{d}$ : 9,59

$$S_d = \sqrt{\frac{\sum (d_i - \underline{d})^2}{n - 1}}$$

$$S_d = \sqrt{\frac{18,45}{8 - 1}}$$

$$S_d = 1,62$$

Se procede a calcular el valor T

$$t = \frac{9,59}{\frac{1,62}{\sqrt{8}}}$$

$$t = 16,70$$

Luego de ello se procede a calcular los grados de libertad y valores críticos a ser presentados en la gráfica T Student, para lo cual se ha tomado en cuenta un grado de significancia de 0,05.

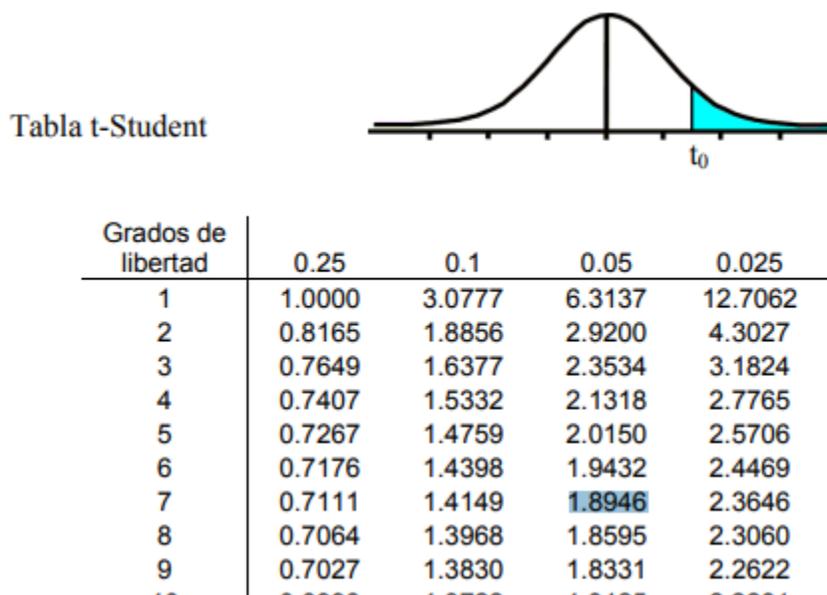
$$gl = (n - 1) = 8 - 1 = 7$$

$$\alpha = 0,05$$

Mediante la tabla de distribución T Student se obtuvo el valor crítico para una prueba de una cola.

**Figura 63**

*Distribución T Student para la medición del tiempo*



*Nota.* Tomado de (Tabla Distribución T, n.d.)

De esta manera se realizó el gráfico de campana con los datos obtenidos. Dicha información fue verificada en el programa informático "Excel", en el cual se obtuvieron los siguientes resultados:

**Tabla 23**

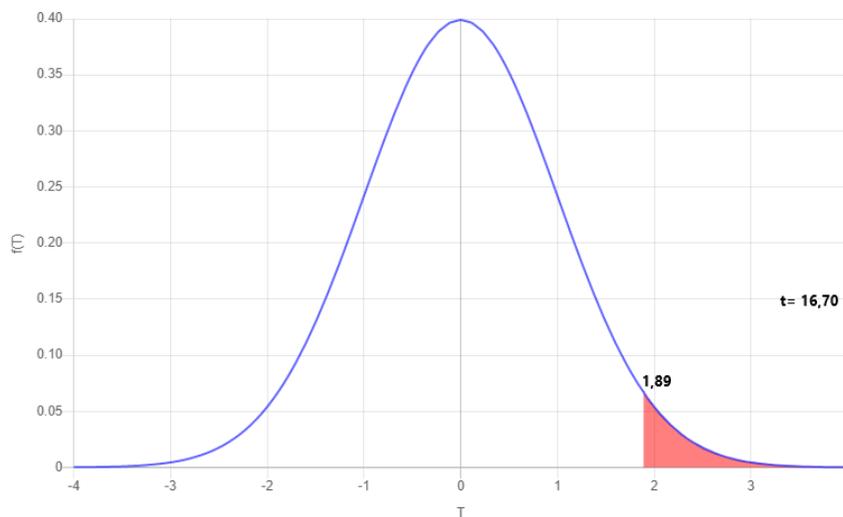
*Resultados de "Excel" obtenidos para el análisis del tiempo*

	<b>Manual</b>	<b>Automatizado</b>
Media	10,065625	0,47875
Varianza	2,581849696	0,0108125
Observaciones	8	8
Coefficiente de correlación de Pearson	-0,126870241	
Diferencia hipotética de las medias	0	
Grados de libertad	7	
Estadístico t	16,70425094	
P(T<=t) una cola	3,36876E-07	
Valor crítico de t (una cola)	1,894578605	

A continuación, se realizó la gráfica correspondiente a la Prueba T- Student de una sola cola hacia la derecha.

**Figura 64**

*Gráfica T Student para el análisis del tiempo empleado*



Basándonos en los cálculos realizados y comprobados por la herramienta estadística llegamos a obtener la gráfica presentada en la Figura 64, al estar el valor  $t$  hacia la derecha se ubica en la zona donde se rechaza la hipótesis nula, por lo tanto, se acepta la hipótesis alternativa la cual indica que el promedio de las diferencias es mayor a 0.

### ***Interpretación de los resultados***

El resultado de la prueba estadística es la aceptación de la hipótesis alternativa que indica que el promedio de las diferencias es mayor a 0, es decir, existe una diferencia entre el tiempo inicial (Proceso Manual) y el tiempo final (Sistema Software) lo que indica que la aplicación del proceso automatizado produce una reducción en el tiempo empleado y por lo tanto se produce una optimización en los tiempos derivados del proceso.

### **Indicador Costos derivados**

El cuarto indicador planteado evalúa los costos derivados del proceso de la monitorización, este indicador nos permite comparar los costos en dólares que se derivan del proceso de monitorización tomando en cuenta los diferentes valores que pueden influir.

Para ejecutar esta prueba se sumaron los diferentes valores que influyen o se derivan de todo el proceso de monitorización y se realizó una proyección durante un año calendario

Uno de los valores más importantes a ser tomado en cuenta es el que se le paga al personal del invernadero, ya que el sueldo que reciben no es solamente por realizar las mediciones manuales, si no que incluye muchas otras responsabilidades.

Es importante identificar la fracción del sueldo el cual estaría relacionado al proceso de monitorización manual, por lo que se realizó el siguiente calculo:

- Cantidad de mediciones diarias: para determinar la cantidad de veces que realizan mediciones nos basamos en la experiencia in situ donde se realiza un total de 4 mediciones de las variables ambientales a lo largo del día
- Cantidad de mediciones mensuales: la cantidad de días es proporcional a los días del mes, ya que al ser un centro agropecuario grande debe mantenerse funcionando los 7 días de la semana, por lo tanto, manejamos un valor de 30 días al mes, siendo el resultado de 120 veces se realiza el proceso manual de monitorización al mes.
- Tiempo empleado: el tiempo promedio empleado en cada proceso de medición manual es de 10 minutos, por lo que al multiplicar la cantidad mensual por el tiempo obtenemos un total de 1200 minutos o su equivalente de 20 horas en el mes.
- Valor por hora de trabajo: el salario que se le ofrece a los trabajadores del invernadero es el salario básico en el Ecuador (\$425 dólares). Por la alta carga de trabajo la cantidad de horas de trabajo al mes por parte del trabajador es de 160 horas repartidas en turnos durante el mes. Para calcular la hora de trabajo es necesario dividir el sueldo por la cantidad de días, que en este caso son 20 días al mes, siendo el resultado \$21,25 el valor percibido diariamente, este valor lo volveremos a dividir entre 8 que son las horas de la jornada de trabajo y obteniendo así \$2,66.
- Fracción del sueldo: al tener ya los principales valores que son, la cantidad de horas empleadas por un trabajador al mes realizando las mediciones y el valor por hora de trabajo, procedemos a multiplicar ambos valores nos da  $20 \times 2,656 = 53,125$ . Por lo tanto, el valor mensual que recibe un trabajador por su labor relacionada a la monitorización de las variables ambientales es \$53,13.

**Tabla 24***Costos relacionados con el proceso de monitorización*

Manual			Automatización		
Descripción	Costo	Tipo	Descripción	Costo	Tipo
Pago al Personal	53,13	Mensual	Cable UTP	78	Una sola vez
			Sensores DHT22	42	Una sola vez
			Módulo ESP32	7	Una sola vez
			Servidor	2,99	Mensual

*Nota.* Costos en dólares relacionados con el proceso de monitorización.

Se aplicó la prueba T-Student para muestras relacionadas utilizando un valor de significancia del 5%, obteniendo como resultado las siguientes hipótesis.

Hipótesis nula (H0):  $\mu_d \leq 0$

Hipótesis alternativa (H1):  $\mu_d > 0$

Los datos utilizados para esta prueba se presentan en la Tabla 23, donde se hace una pequeña proyección de lo que serían los costos en un año.

**Tabla 25***Costos totales derivados del proceso de monitorización*

<b>Costos derivados</b>				
<b>Mes</b>	<b>Manual</b>	<b>Automatizado</b>	<b>Diferencia</b>	$(d - \underline{d})^2$
1	53,13	129,99	-76,86	13552,84028
2	53,13	2,99	50,14	112,0069444
3	53,13	2,99	50,14	112,0069444
4	53,13	2,99	50,14	112,0069444
5	53,13	2,99	50,14	112,0069444
6	53,13	2,99	50,14	112,0069444
7	53,13	2,99	50,14	112,0069444
8	53,13	2,99	50,14	112,0069444
9	53,13	2,99	50,14	112,0069444
10	53,13	2,99	50,14	112,0069444
11	53,13	2,99	50,14	112,0069444
12	53,13	2,99	50,14	112,0069444

*Nota.* Costos en dólares durante un año relacionados al proceso de monitorización.

Para el cálculo estadístico se realizó el siguiente proceso:

$$t = \frac{\underline{d}}{S_d/\sqrt{n}}$$

$\underline{d}$ : media de las diferencias

$S_d$ : desviación estándar

$n$ : número de muestras

$\underline{d}$ : 39,55

$$S_d = \sqrt{\frac{\sum (d_i - \underline{d})^2}{n - 1}}$$

$$S_d = \sqrt{\frac{14784,9167}{12 - 1}}$$

$$S_d = 36,66$$

Se procede a calcular el valor T

$$t = \frac{39,55}{\frac{36,66}{\sqrt{12}}}$$

$$t = 3,737$$

Luego de ello se procede a calcular los grados de libertad y valores críticos a ser presentados en la gráfica T Student, para lo cual se ha tomado en cuenta un grado de significancia de 0,05.

$$gl = (n - 1) = 12 - 1 = 11$$

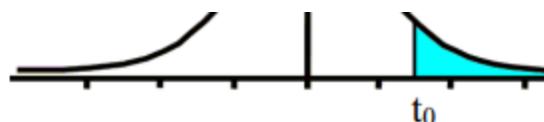
$$\alpha = 0,05$$

Mediante la tabla de distribución T Student para pruebas de una cola se determinó el valor crítico:

### Figura 65

*Distribución T Student para el análisis de los costos*

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025
1	1.0000	3.0777	6.3137	12.7062
2	0.8165	1.8856	2.9200	4.3027
3	0.7649	1.6377	2.3534	3.1824
4	0.7407	1.5332	2.1318	2.7765
5	0.7267	1.4759	2.0150	2.5706
6	0.7176	1.4398	1.9432	2.4469
7	0.7111	1.4149	1.8946	2.3646
8	0.7064	1.3968	1.8595	2.3060
9	0.7027	1.3830	1.8331	2.2622
10	0.6998	1.3722	1.8125	2.2281
11	0.6974	1.3634	1.7959	2.2010
12	0.6955	1.3562	1.7823	2.1788

*Nota.* Tomado de (Tabla Distribución T, n.d.)

De esta manera se realizó el gráfico de campana con los datos obtenidos. Dicha información fue verificada en el programa informático "Excel", en el cual se obtuvieron los siguientes resultados:

**Tabla 26**

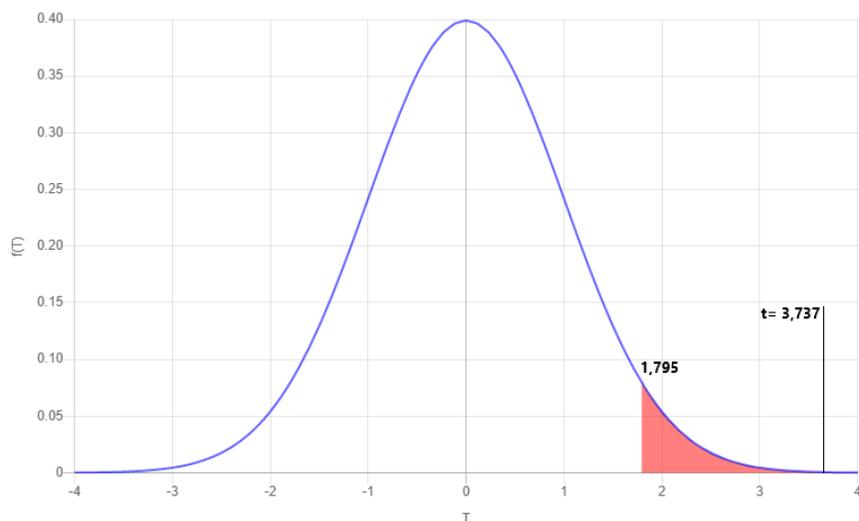
*Resultados de "Excel" obtenidos para el análisis de los costos*

	<b>Manual</b>	<b>Automatizado</b>
Media	53,13	13,57333333
Varianza	0	1344,083333
Observaciones	12	12
Coefficiente de correlación de Pearson	N/A	
Diferencia hipotética de las medias	0	
Grados de libertad	11	
Estadístico t	3,737637795	
P(T<=t) una cola	0,00163981	
Valor crítico de t (una cola)	1,795884819	

A continuación, se realizó la gráfica correspondiente a la Prueba T- Student de una sola cola hacia la derecha.

**Figura 66**

*Gráfica T Student para el análisis de costos derivados*



Basándonos en los cálculos realizados y comprobados por la herramienta estadística llegamos a obtener la gráfica presentada en la Figura 66, al estar el valor  $t$  hacia la derecha se ubica en la zona donde se rechaza la hipótesis nula, por lo tanto, se acepta la hipótesis alternativa la cual indica que el promedio de las diferencias es mayor a 0.

### ***Interpretación de los resultados***

El resultado de la prueba estadística es la aceptación de la hipótesis alternativa que indica que el promedio de las diferencias es mayor a 0, es decir, existe una diferencia entre los costos del proceso manual y los costos del sistema software lo que indica que la aplicación del sistema software produce una reducción y optimización en los costos derivados del proceso de monitorización.

## Capítulo V

### Conclusiones y Recomendaciones

#### Conclusiones

- Se cumplió con el objetivo general que fue desarrollar un Sistema Software de monitorización mediante el uso de IoT que optimice los tiempos y costos derivados de la lectura de las variables ambientales de temperatura y humedad en el centro agropecuario “Pilonera Verito”.
- La elaboración del marco teórico permitió recolectar información para comprender las diferentes opciones disponibles sobre el proceso de automatización, conociendo a fondo las temáticas de: dispositivos IoT, servidores y sus servicios, además de diferentes Frameworks para Frontend y BackEnd; logrando así planificar un sistema software único y óptimo.
- El diseño de la infraestructura IoT al ser mediante una conexión física instalada sobre un cableado previo permite darle una seguridad sobre incidentes físicos, al usar este tipo de conexión se logró una comunicación rápida e instantánea lo que permite al sistema software funcionar sin retrasos y logrando además un fácil mantenimiento en cualquiera de sus partes.
- El desarrollo de la aplicación móvil mediante el uso del patrón bloc enfocado en la organización y manejo de estado, logró centralizar la lógica de la misma en fases tempranas del proyecto, consistentes en puntos proveedores de información los cuales fueron aprovechados por nuevos módulos en fases tardías del desarrollo de la aplicación móvil, los cuales al ser desarrollados como widgets de Flutter fueron fácilmente adaptados, como pequeñas funcionalidades, a la interfaz, haciendo a la aplicación más escalable.

- El desarrollo del sistema software se realizó a través del marco de trabajo Scrum, cumpliendo con éxito cada fase que especifica el marco con su respectivo artefacto, se logró entregar valor al usuario desde etapas tempranas del proyecto, esta forma de desarrollo ágil permitió la realización de pruebas y retroalimentación de cómo debería comportarse la completitud del sistema llegada su finalización, dotándolo de fiabilidad, precisión y alta disponibilidad.
- Las diferentes partes del sistema software se desplegaron e implementaron con éxito. El servidor se encuentra en una instancia EC2 de AWS donde se encuentran activos los servicios de MQTT y la API desarrollada en PHP, la aplicación se distribuye públicamente a través de Play Store en Android, mientras que la infraestructura IoT se instaló correctamente dentro del invernadero, logrando así tener el sistema software completo en uso y disponible.
- La validación del sistema software se basó en los diferentes indicadores planteados en el presente proyecto, mediante el uso de la prueba estadística T-Student en dos de sus formatos (Muestras independientes y Muestras Relacionadas) que permitió obtener unos resultados positivos para el proyecto demostrando precisión en los datos y optimización tanto en tiempos como en costos relacionados al proceso de monitorización.
- El desarrollo de la aplicación bajo el patrón bloc permitirá añadir nuevas funcionalidades en base aquellas que se han construido, como por ejemplo el historial de mediciones, al tener acceso a dicha información se puede ampliar una nueva funcionalidad donde se transformarán los datos históricos bajo las reglas de negocio pertinentes y se vuelva una herramienta en la planificación y toma de decisiones para el invernadero.

## Recomendaciones

- Se recomienda el uso de Scrum para el desarrollo de proyectos de automatización de procesos manuales, debido a que se pueden obtener avances a corto plazo, permitiendo realizar pruebas reales y así adaptando el sistema no solo a las necesidades específicas del proceso sino también a los posibles cambios que se requieran al momento de convertirlo en un proceso automatizado.
- Se recomienda verificar la resistencia de los dispositivos a utilizar, debido a que en ambientes comunes como habitaciones no suponen un problema en cuanto a duración del material no siendo así en entornos especializados como un invernadero, ya que este además de presentar condiciones más extremas para los dispositivos, su área total puede presentar un problema al momento de realizar mantenimiento a la infraestructura del sistema.
- Se recomienda, para este tipo de proyectos, que el lugar en el cual va a ser implantado disponga de una conexión estable a internet, ya que este tipo de propuestas tecnológicas requiere de una conexión continua a la red para el envío de información, aspecto que influye en la fiabilidad y disponibilidad del sistema, algo que puede ser mitigado en la programación del módulo IoT, mediante un control de errores que permitan al dispositivo mantener un correcto funcionamiento en caso de presentarse algún inconveniente de cualquier índole.
- Se recomienda validar las mediciones obtenidas por sensores con la ayuda de una o más variedades de herramientas especializadas con el fin de asegurar la precisión de las medidas que se le presentan al usuario, debido a que esto influye directamente en las decisiones a tomar con respecto al funcionamiento del invernadero.

## Bibliografía

- Amazon. AWS - About Us. Amazon.com. septiembre de 2011. Consultado el 8 de septiembre de 2021.
- ANAYA, D. (2019). Elaboración del prototipo de un sistema de control de variables atmosféricas automatizado para el cultivo de plantas bajo invernadero en ambiente indoor en la Región Caribe. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699. [http://repositorio.cuc.edu.co/bitstream/handle/11323/6072/Elaboración del prototipo de un sistema de control de variables atmosféricas automatizado para el cultivo de plantas bajo invernadero en ambiente indoor en la Región Caribe.pdf?sequence=1&isAllowed](http://repositorio.cuc.edu.co/bitstream/handle/11323/6072/Elaboración%20del%20prototipo%20de%20un%20sistema%20de%20control%20de%20variables%20atmosféricas%20automatizado%20para%20el%20cultivo%20de%20plantas%20bajo%20invernadero%20en%20ambiente%20indoor%20en%20la%20Región%20Caribe.pdf?sequence=1&isAllowed)
- Ashton. K. (2002). That “Internet of Things” Things. *RFID Journal*, 4986
- Bahit, E. (2012). *Scrum & extreme Programming para programadores*. 162. <http://www.cursosdeprogramacionadistancia.com/static/pdf/material-sin-personalizar-agile.pdf>
- Cisneros, Y., & Mosquera, M. (2019). *DISEÑO DE UN PROTOTIPO DE INVERNADERO AUTOMATIZADO CON CULTIVO DE PIMIENTO Y APLICACIÓN EN AMBIENTE WEB PARA EL CONTROL Y MONITOREO DE LA PRODUCCIÓN*. Universidad De Guayaquil.
- Eduardo, O. (2021). Universidad Politécnica Salesiana Sede Quito. In *Tesis*. <http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf>

- Ferrer, R. (2022). Diseñar la estructura de una aplicación Flutter: el patrón BLoC.  
<https://www.raulferrergarcia.com/disenar-la-estructura-de-una-aplicacion-flutter-el-patron-bloc/>
- Galvan P. (2021). Hola Mundo IoT. Recuperado 8 de septiembre de 2021, de  
<https://sg.com.mx/revista/51/hola-mundo-iot>
- Guasch Llobera, J., & Calleja Collado, M. (2019). Monitorización de sensores con arduino utilizando el protocolo MQTT (Bachelor's thesis, Universitat Politècnica de Catalunya).
- Guerra Jose. (2021). ESP32 Wifi y Bluetooth en un solo chip. Recuperado 8 de septiembre de 2021, de <https://programarfacil.com/esp8266/esp32/>
- Guerrero-Ibañez, J. A., Estrada-Gonzalez, F. P., Medina-Tejeda, M. A., Rivera-Gutierrez, M. G., Alcaraz-Aguirre, J. M., Maldonado-Mendoza, C. A., ToledoZuñiga, D., & Lopez-Gonzalez, V. I. (2017). SGreenH-IoT: Plataforma IoT para Agricultura de Precisión. *CISCI 2017 - Decima Sexta Conferencia Iberoamericana En Sistemas, Cibernética e Informática, Decimo Cuarto Simposium Iberoamericano En Educacion, Cibernética e Informática, SIECI 2017 - Memorias*, 315–320.
- Hegde, S. G. (2010). Study of IoT: Understanding IoT Architecture, Applications, Issues and Challenges, 477–482
- Kaur, K. (2016). The idea of Smart villages based on Internet of Things ( IoT ), 165–168.
- León Merchán, K. L., & Sánchez Calderón, O. A. (2020). Sistema de monitoreo de variables ambientales en cultivos de papa mediante IOT y energía solar fotovoltaica.
- Ma, H. (2011). Internet of Things: Objectives and Scientific Challenges, 26, 919–924.  
<https://doi.org/10.1007/s11390-011-1189-5>

- Menacho Rodríguez, Á. (2020). EventYou: Aplicación móvil con Flutter y FlutterFire.
- Montoya, E. A. Q., Colorado, S. F. J., Muñoz, W. Y. C., & Golondrino, G. E. C. (2017). Propuesta de una arquitectura para agricultura de precisión soportada en IoT. *Revista Ibérica de Sistemas e Tecnologías de Informação*, (24), 39-56.
- Mosquera, R., Orlando, A., Ruales Galarza, C. A., & Rivas Lalaleo, D. R. Análisis comparativo de las tecnologías de comunicación para IoT en la monitorización de zonas agrícolas de las provincias de Cotopaxi y Tungurahua.
- Necpas Lechon, W. E., & Quishpe Tutillo, J. F. (2021). Desarrollo de un dispositivo IoT para el invernadero de la Asociación de Mujeres Productoras Agroecológicas de Cangahua (Bachelor's thesis).
- Pacheco Cueva, C. A., & Herrera Albarracin, R. C. (2021). Universidad técnica de cotopaxi. *Universidad Técnica de Cotopaxi*, 1, 101.  
<http://repositorio.utc.edu.ec/bitstream/27000/4501/1/PI-000727.pdf>
- Paredes Cruz, G. D. (2021). Sistema de monitoreo de humedad de suelo, de largo alcance y georeferenciación usando internet de las cosas (IOT) (Bachelor's thesis).
- PÉREZ, M. R., MENDOZA, M. A., & SUAREZ, M. J. (2019). Paradigma IoT: desde su conceptualización hacia su aplicación en la agricultura. *Paradigma*, 40(18).
- Rivas, J., & Benavides, G. (2020). *DESARROLLO DE SOFTWARE PARA EL CONTROL Y SUPERVISIÓN DE UN SISTEMA HIDROPÓNICO* [Universidad De Guayaquil].  
<http://repositorio.ug.edu.ec/bitstream/redug/49638/1/B-CISC-PTG-1839-2020> Rivas Chávez Joffre Joel - Benavides Castillo Gerald Samuel.pdf

Sánchez, J. (2020). INTRODUCCIÓN AL PATRÓN BLOC.

<http://xurxodev.com/introduccion-al-patron-bloc/>

Sánchez, R. (2015). Tabla t-Student. *Revista Mexicana de Ciencia*, 26(1), 56–61.

<http://www.scielo.org.mx/pdf/rmc/v26n1/v26n1a9.pdf>

Schwaber, K., & Sutherland, J. (2020). La Guía Scrum. La Guía Definitiva de Scrum: Las Reglas del Juego. 17. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

Torres, T. (2017). *IMPLEMENTACIÓN DE UN SISTEMA DE ALMACENAMIENTO DE LA INFORMACIÓN, MONITOREO Y CONTROL APLICANDO EL INTERNET DE LAS COSAS, PARA LA AUTOMATIZACIÓN DE UN INVERNADERO*. 111.

*Tabla distribución t.* (n.d.). 0–1. Retrieved July 21, 2022, from

[https://cms.dm.uba.ar/academico/materias/verano2022/probabilidades\\_y\\_estadistica\\_C](https://cms.dm.uba.ar/academico/materias/verano2022/probabilidades_y_estadistica_C)

Vásquez, C., & Eduardo, D. (2019). Análisis y desarrollo de prototipo automatizado para el monitoreo de un invernadero orgánico en la Finca “La Piedra” Recinto De Cerecita, en la localidad de bajada de Chanduy, por medio de alternativas tecnológicas de bajo costo [Universidad De Guayaquil]. <http://repositorio.ug.edu.ec/handle/redug/39249>

**Anexos**