



**UNIVERSIDAD DE TALCA  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Sistema integrado de gestión hídrica orientado a pequeños  
y medianos productores de cultivos y frutales de la Región  
del Maule**

**FABIÁN IGNACIO OYARCE VALENZUELA**

Profesor Guía: RICARDO ENRIQUE PÉREZ GUZMÁN

Profesor Co-guía: FERNANDO FUENTES PEÑAILILLO

Memoria para optar al título de  
Ingeniero Civil en Computación

Curicó – Chile  
Diciembre, 2020

## CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular official stamps and handwritten signatures in blue ink. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS UNIVERSIDAD DE TALCA' and the right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICO'.

Curicó, 2022

*Dedicado a mis padres, hermana y abuelos*

## AGRADECIMIENTOS

Agradezco a mi familia, en especial a mis padres Renato Aurelio Oyarce Mondaca y Angélica Millaray Valenzuela Villalobos, a mi hermana Catalina Andrea Oyarce Valenzuela, mis abuelos Elías Valenzuela Domínguez, Jovita Villalobos Reyes, Juan Oyarce Lorca, Ester Mondaca Salazar, mi tía Evelyn Valenzuela Villalobos y su esposo Williams Maldonado Paredes, por ser el pilar fundamental en la construcción de esta memoria y en especial en la construcción de mi vida personal. Por guiarme en mi formación para llegar a ser la persona que hoy soy. Por estar siempre atentos y preocupados que nada me falte y entregarme todo lo que en su momento precisé.

A los profesores, en especial a Fernando Fuentes Peñailillo y Ricardo Pérez Guzmán por su cercanía, disposición, preocupación y simpatía; que con mucha paciencia me han entregado todos los conocimientos necesarios para mi vida profesional y la construcción de esta memoria. Adicionalmente, quiero hacer una mención a la profesora Ruth Garrido con la cual tuve el agrado de trabajar desde mi segundo semestre en la Universidad hasta el penúltimo de este, lo que me permitió ir desarrollando mis habilidades sociales y me enseñó a ser una mejor persona capaz de desenvolverme sin ningún problema en el mundo universitario y laboral. A los profesores Rodrigo Bustamante y Luis Silvestre quienes desde un comienzo siempre mostraron su apoyo e interés ante mi trabajo.

A mis amigos Fernanda López, Máximo Hernández, Nicolás Delaunoy, Germán Arenas, Gabriel Bertoni, Andrés León, Emilio Arenas, y en especial a Cristóbal Mardones quien desde la básica siempre ha sido un gran amigo y apoyo fundamental en toda mi vida y carrera universitaria.

Por último, a todas las otras personas que no he mencionado y que de una u otra manera me han ayudado a completar esta etapa.

Mientras tenga amigos y familia como ustedes, estaré agradecido con la vida. Gracias por siempre estar cuando los necesito y por ser mis ángeles de la guarda.

Gracias.

## TABLA DE CONTENIDOS

	página
<b>Dedicatoria</b>	<b>I</b>
<b>Agradecimientos</b>	<b>II</b>
<b>Tabla de Contenidos</b>	<b>III</b>
<b>Índice de Figuras</b>	<b>VI</b>
<b>Índice de Tablas</b>	<b>VIII</b>
<b>Resumen</b>	<b>IX</b>
<b>1. Introducción</b>	<b>10</b>
1.1. Contexto . . . . .	10
1.2. Definición del problema . . . . .	11
1.3. Objetivos . . . . .	12
1.3.1. Objetivo general . . . . .	12
1.3.2. Objetivos específicos . . . . .	12
1.4. Alcances . . . . .	13
1.5. Propuesta de solución . . . . .	13
1.6. Trabajos relacionados . . . . .	14
1.7. Resumen del capítulo . . . . .	16
<b>2. Marco Teórico</b>	<b>17</b>
2.1. Conceptos y tecnologías . . . . .	17
2.1.1. Tecnologías utilizadas . . . . .	18
2.2. Metodología . . . . .	20
2.2.1. Análisis del estado del arte . . . . .	20
2.2.2. Análisis de las propuestas . . . . .	20
2.2.3. Desarrollo e implementación PXP . . . . .	20
2.3. Evaluación del proyecto . . . . .	21
2.3.1. Prueba de caja negra . . . . .	21

2.3.2. Prueba de usabilidad SUS . . . . .	22
2.4. Resumen del capítulo . . . . .	23
<b>3. Marco Metodológico</b>	<b>25</b>
3.1. Personal Extreme Programming (PXP) . . . . .	25
3.2. Requisitos . . . . .	26
3.3. Planificación . . . . .	27
3.4. Iteraciones . . . . .	28
3.4.1. Inicio de iteración . . . . .	30
3.4.2. Diseño . . . . .	30
3.4.3. Implementación . . . . .	30
3.4.4. Prueba del sistema . . . . .	30
3.4.5. Retrospectiva . . . . .	30
3.5. Justificación de la metodología . . . . .	31
3.6. Resumen del capítulo . . . . .	31
<b>4. Desarrollo de la solución</b>	<b>32</b>
4.1. Hardware . . . . .	32
4.1.1. Sensores y Componentes . . . . .	32
4.1.2. Diseño de placa de circuito impreso . . . . .	35
4.1.3. Programación de la placa . . . . .	37
4.2. Software . . . . .	38
4.2.1. Arquitectura Física . . . . .	38
4.2.2. Arquitectura Lógica . . . . .	39
4.2.3. Historias de usuario . . . . .	40
4.2.4. Desarrollo de la solución . . . . .	41
4.3. Resumen del Capítulo . . . . .	58
<b>5. Evaluación Experimental</b>	<b>59</b>
5.1. Fases de experimentación del Software . . . . .	59
5.1.1. Definición . . . . .	59
5.1.2. Diseño de la experimentación . . . . .	59
5.1.3. Ejecución de la experimentación . . . . .	60
5.1.4. Sesión de Experimentación . . . . .	61
5.1.5. Análisis de Experimentación . . . . .	61

5.2. Pruebas de caja negra hardware . . . . .	63
5.3. Resumen del Capítulo . . . . .	64
<b>6. Conclusiones</b>	<b>65</b>
6.1. Sobre los Resultados Alcanzados . . . . .	65
6.2. Sobre el Cumplimiento de los Objetivos . . . . .	66
6.2.1. Objetivos específicos . . . . .	66
6.2.2. Objetivo general . . . . .	67
6.3. Trabajo Futuro . . . . .	68
6.4. Resumen del Capítulo . . . . .	69
<b>Bibliografía</b>	<b>70</b>
<b>Anexos</b>	
<b>A: Versiones de las placas de circuito impreso</b>	<b>74</b>
<b>B: Prototipos de las estaciones construidas</b>	<b>78</b>
<b>C: Conversión analógica digital 5V a 3.3V</b>	<b>81</b>
<b>D: Pruebas de campo de radio frecuencia</b>	<b>83</b>
<b>E: Ensamblado de estación meteorológica final</b>	<b>87</b>
<b>F: Pruebas de caja negra</b>	<b>92</b>

## ÍNDICE DE FIGURAS

	página
1.1. Causas de la brecha y riesgo hídrico en los territorios en Chile . . . . .	10
2.1. Representación de una prueba de caja negra . . . . .	22
3.1. Flujo de trabajo utilizado en el proyecto . . . . .	27
4.1. Diseño de circuito impreso V7.0 . . . . .	36
4.2. Flujo de programación del microcontrolador . . . . .	38
4.3. Arquitectura Física . . . . .	39
4.4. Arquitectura Lógica . . . . .	40
4.5. Estructura de árbol de la base de datos . . . . .	43
4.6. Vista Login. . . . .	44
4.7. Vista Google Auth. . . . .	44
4.8. Vista Recuperar Contraseña. . . . .	45
4.9. Vista Registro. . . . .	45
4.10. Clima Soleado. . . . .	46
4.11. Clima Nublado. . . . .	46
4.12. Clima Lluvioso. . . . .	47
4.13. Clima Nocturno. . . . .	47
4.14. Gráfico Humedad. . . . .	48
4.15. Gráfico Temperatura. . . . .	48
4.16. Gráfico Presión Atmosférica. . . . .	49
4.17. Gráfico Lluvia. . . . .	49
4.18. Cambio de Estación . . . . .	50
4.19. Modelo de Penman Monteith . . . . .	52
4.20. Vista consulta consumo hídrico. . . . .	53
4.21. Vista consumo hídrico. . . . .	53
4.22. Circuito Impreso de la Placa. . . . .	54
4.23. Caja Contenedora Prototipo Uno. . . . .	55
4.24. Receptor RF. . . . .	56
4.25. Prototipo Funcional Uno. . . . .	57



A.1. Diseño de circuito impreso V1.0 . . . . .	74
A.2. Diseño de circuito impreso V2.0 . . . . .	75
A.3. Diseño de circuito impreso V3.0 . . . . .	75
A.4. Diseño de circuito impreso V4.0 . . . . .	75
A.5. Diseño de circuito impreso V5.0 . . . . .	76
A.6. Diseño de circuito impreso V6.0 . . . . .	76
A.7. Diseño de circuito impreso V7.0 . . . . .	76
A.8. Receptor de la estación meteorológica . . . . .	77
B.1. Primer prototipo de estación meteorológica . . . . .	78
B.2. Segundo prototipo de estación meteorológica . . . . .	79
B.3. Tercer prototipo de estación meteorológica . . . . .	79
B.4. Primer armado sensores de viento y lluvia . . . . .	80
C.1. Calibración de la dirección del viento a 3.3V . . . . .	81
C.2. Trasformación respuesta analógica a dirección del viento . . . . .	82
D.1. Campo San Clemente . . . . .	83
D.2. Prueba de distancia a 3 kilómetros de radio frecuencia en campo . . . . .	84
D.3. Prueba de distancia a 500 metros lateral de radio frecuencia en ciudad . . . . .	85
D.4. Prueba de distancia a 500 metros rectos de radio frecuencia en ciudad . . . . .	86
E.1. Construcción placa estación meteorológica . . . . .	88
E.2. Armado cableado estación meteorológica . . . . .	89
E.3. Armado empaquetado estación meteorológica . . . . .	90
E.4. Ensamblado caja estación meteorológica . . . . .	91

## ÍNDICE DE TABLAS

	página
1.1. Tabla comparativa de los sistemas existentes . . . . .	16
2.1. Preguntas SUS originales traducidas al español . . . . .	23
3.1. Ejemplo de historia de usuario . . . . .	26
3.2. Planificación de las iteraciones . . . . .	29
4.1. Descripción de las historias de usuario . . . . .	41
5.1. Prueba caja negra #4 - Calculo de Et0 . . . . .	64
F.1. Prueba caja negra #1 - Lectura de datos meteorológicos . . . . .	93
F.2. Prueba caja negra #2 - Visualización de datos en la aplicación móvil . . .	94
F.3. Prueba caja negra #3 - Visualización históricas de datos en la aplicación móvil . . . . .	95
F.4. Prueba caja negra #4 - Calculo de Et0 . . . . .	96
F.5. Prueba caja negra #5 - Visualización de Et0 . . . . .	97
F.6. Prueba caja negra #6 - Soporte múltiples de estaciones . . . . .	98
F.7. Prueba caja negra #7 - Visualización de estaciones . . . . .	98

## RESUMEN

Debido al cambio climático y a los nuevos objetivos productivos, hoy la agricultura requiere de herramientas objetivas para el monitoreo del clima, como lo son las estaciones meteorológicas automáticas (EMA). Sin embargo, los pequeños agricultores se ven excluidos en el acceso a estos avances, debido principalmente al alto costo de implementación de estas tecnologías.

En este proyecto se desarrolló un sistema integrado de bajo costo que combina software y hardware para determinar el consumo hídrico de los cultivos. Adicionalmente, el desarrollo consideró la integración de un modelo de estimación de la evapotranspiración, capaz de estimar con precisión el consumo hídrico de los cultivos. Con esta información y a través de una aplicación móvil, se obtuvieron las variables de interés agronómico, como grados día acumulados (GDD), horas frío (HF) y evapotranspiración del cultivo (ET<sub>0</sub>). Estos valores permiten tomar decisiones productivas en base a datos objetivos, aumentando la eficiencia del riego y mejorando la calidad de la producción.

Dentro de los principales resultados luego de realizar las pruebas de hardware y software, se logró determinar que es factible desarrollar una estación meteorológica de bajo costo (menor a 300 USD) utilizando plataformas de Cloud Computing mediante módulos de radio frecuencia y WiFi, capaz de medir todas las variables agroclimáticas para estimar el consumo hídrico de los cultivos. Además, mediante el uso de Firebase se generó una base de datos robusta capaz de ser monitorizada en tiempo real. Finalmente, la base de datos es consumida directamente por la aplicación móvil desarrollada junto al modelo matemático en R para la estimación del consumo hídrico.

# 1. Introducción

---

En este capítulo se describe la situación actual del recurso hídrico de nuestro país enfocado directamente en la zona central de Chile. Se realiza una descripción del contexto del problema de investigación, objetivos y alcances del proyecto.

## 1.1. Contexto

La correcta utilización de los recursos naturales y en particular del agua, es uno de los objetivos fundamentales de la mayoría de los países en la actualidad. A pesar de que este recurso cubre el 70 % de la superficie del planeta tierra solo un reducido porcentaje del 0.5 % corresponde a agua dulce. Adicionalmente, es importante mencionar que de este reducido porcentaje, el rubro agrícola concentra más del 70 % del consumo [24]. Razón por la cual, es una tarea fundamental utilizar apropiadamente el agua dulce destinada al riego agrícola. Según muestran algunos estudios de la Fundación Chile sobre las causas de la brecha y riesgo hídrico, se realizó un análisis de seis cuencas, seleccionadas por ser representativas de diversas realidades y problemáticas hídricas en Chile (Copiapó, Aconcagua, Maipo, Maule, Lebu y Baker). Escenarios Hídricos 2030 identificó las causas de la brecha y riesgo hídrico que se dan en los territorios en Chile. Estas causas y sus índices de impacto se pueden apreciar en la Figura 1.1 [11].

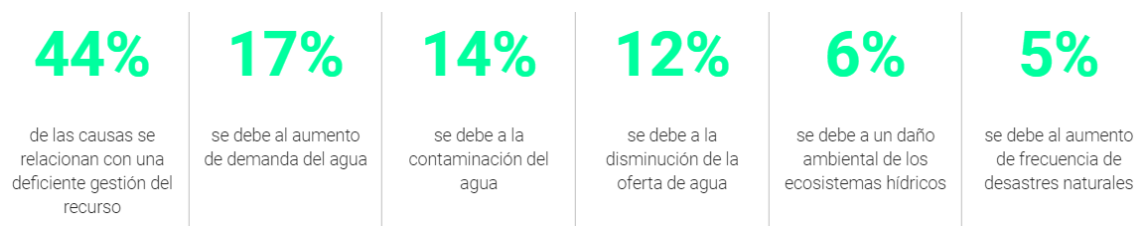


Figura 1.1: Causas de la brecha y riesgo hídrico en los territorios en Chile

Como se puede observar en la Figura 1.1, la mayor causa del riesgo hídrico (44%), se relaciona con una deficiente gestión del recurso. Razón por la cual, la producción hortofrutícola ve disminuido el potencial productivo de los huertos debido a la escasa información meteorológica localizada, lo que imposibilita la toma de decisiones en base a criterios objetivos. Por otra parte, el alto costo de las estaciones meteorológicas automáticas (EMA) que existen en el mercado, impide que pequeños agricultores puedan acceder a ellas. Esta herramienta es fundamental para la programación del riego, porque de ella se obtiene la evapotranspiración del cultivo (ET<sub>0</sub>). Este valor permite definir el tiempo y la frecuencia de riego, lo que se ve reflejado en mejor rendimiento, calidad y menores costos de productos [9].

Dotar al agricultor de una herramienta capaz de estimar los tiempos y la frecuencia de riego es un elemento vital para la agricultura, debido a la necesidad de herramientas objetivas y precisas de monitoreo de las condiciones meteorológicas. Si se pretende establecer sistemas de producción modernos, estas herramientas deben ser accesibles a un amplio espectro de usuarios, sin importar su tamaño. Si la información no está disponible, impacta de manera negativa en toda la cadena de producción, disminuyendo la rentabilidad agrícola.

## 1.2. Definición del problema

Una reciente investigación [14], indica que la zona central de Chile enfrentará una reducción del recurso hídrico de hasta un 40% en 2050, principalmente debido al cambio climático. En este escenario, la búsqueda de alternativas económicas y eficientes para el aprovechamiento de este recurso es una de las prioridades del país y la región. Por esta razón, es indispensable desarrollar e implementar nuevas técnicas y/o tecnologías que nos permitan desagrar las estimaciones propuestas y mejorar la utilización de este recurso [14].

Datos recientes señalan para nuestro país que la disponibilidad de agua ha disminuido en un 37% en los últimos años, lo cual ha generado cambios drásticos en los sistemas agrícolas [1]. Por tanto, para el agricultor se vuelve una tarea primordial la determinación objetiva del volumen de agua a aplicar. Para realizar esta tarea con éxito se requiere de una estación meteorológica automática (EMA). Sin embargo, estos dispositivos tienen un costo muy elevado lo que dificulta enormemente su implementación [25]. Adicionalmente, es importante indicar que una estación meteorológica automática (EMA) por sí sola no es

capaz de realizar “cálculos” sobre la correcta distribución del agua, para esto, se pueden utilizar algunos modelos matemáticos encontrados en la literatura [14] para determinar las necesidades hídricas de los cultivos.

Si bien, se han desarrollado algunas investigaciones que buscan acercar estas tecnologías de transferencia a los pequeños y medianos agricultores, todavía existen insuficiencias en el modelo utilizado, específicamente, respecto a la capacidad de adquirirla a un precio razonable y su despliegue de acuerdo a las características de la zona. De esta manera, se define como **problema** de nuestra investigación la inexistencia de un sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales de la región del Maule de bajo costo que permita al agricultor mitigar los efectos del cambio climático y, en consecuencia, la reducción de agua disponible.

Finalmente, es importante mencionar que, gracias al financiamiento otorgado íntegramente por la empresa Biovisión Agrícola en combinación con la información publicada y de libre acceso que han generado científicos locales, se espera resolver esta brecha mediante la realización del presente trabajo de investigación y desarrollo.

### 1.3. Objetivos

En este apartado se detalla el objetivo principal del desarrollo de la memoria junto a los objetivos específicos asociados.

#### 1.3.1. Objetivo general

- Determinar el consumo hídrico de frutales y vives, a través de una aplicación móvil integrando técnicas de Cloud Computing y protocolos de comunicación.

#### 1.3.2. Objetivos específicos

- Mejorar la eficiencia de la utilización del agua, mediante un modelo matemático, la información de múltiples sensores y la utilización de Cloud Computing.
- Disminuir los costos de una estación meteorológica para que pueda ser accesible a cualquier tipo de productor agrícola.
- Disminuir la latencia de los protocolos de comunicación convencionales, a través de la utilización de protocolos de comunicación eficientes como MQTT.

## 1.4. Alcances

El proyecto se encuentra delimitado por los alcances que se detallan a continuación:

- Se analizó e implementó un modelo matemático (Modelo de Penman-Monteith) para determinar el consumo hídrico.
- Se desarrolló una aplicación móvil que permitió visualizar los datos obtenidos de una estación meteorológica de bajo costo.
- Los resultados obtenidos de la cantidad de agua a utilizar son válidos única y exclusivamente para el sitio experimental.
- Esta propuesta solo contempla el desarrollo de una aplicación móvil para dispositivos Android 8 o superior y en idioma español.
- Para que los cálculos obtenidos sean válidos, se requiere que la estación meteorológica se encuentre ubicada en una área de referencia cubierta por pasto.

## 1.5. Propuesta de solución

La solución propuesta consiste en un sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales del Maule. Esto incluye la creación de una estación meteorológica de bajo costo, orientada a la recopilación de información de interés agrícola, para ser utilizada por los agricultores en la toma de decisiones productivas. Su implementación pretende mejorar la eficiencia, rendimiento y calidad de sus cultivos.

El valor de esta investigación radica en que utiliza el modelo de Penman-Montieth para la obtención de la evapotranspiración de referencia (ET<sub>0</sub>) que se encuentra validado para la zona central de Chile, gracias a trabajos previos desarrollados por el equipo de investigación. De esta forma, mediante una aplicación móvil capaz de mostrar las variables de forma amigable para el agricultor, se obtendrá información de interés agrícola como temperatura, humedad relativa, luminosidad, cantidad de lluvia caída, velocidad y dirección del viento, grados día acumulados, horas frío, y evapotranspiración.

La aplicación tendrá menús desplegables donde el agricultor podrá seleccionar su superficie, el cultivo y las características de su sistema de riego, con el fin de entregar información personalizada. El producto final consistirá en un kit auto-administrable, que contendrá instrucciones sencillas para que sea el mismo agricultor quien lo ubique dentro de su predio y le realice el mantenimiento necesario (principalmente limpieza periódica de los sensores).

## 1.6. Trabajos relacionados

Si bien, actualmente existen numerosas aplicaciones para gestionar datos meteorológicos, aún hay insuficiencias en la gestión de la información, la aplicabilidad de un modelo matemático específico para la región, o la planificación del agua y el estado de su cultivo.

Dentro de las aplicaciones similares se destacan las siguientes:

- My AcuRite, es una aplicación para Android que utiliza sensores ambientales de AcuRite para ofrecer lecturas de mediciones climáticas. My AcuRite también es su aplicación de alerta meteorológica, que notifica cuando las condiciones requieren atención. Sin embargo, esta aplicación solo permite capturar datos medidos y posteriormente mostrarlos al usuario. Además, solo es compatible con unidades meteorológicas desarrolladas por la misma empresa (AcuRite), las cuales están desarrolladas principalmente para uso particular y no agrícola [22].
- Ambient Weather, es una plataforma web que permite recibir datos de diferentes estaciones meteorológicas desarrolladas por terceros mediante el protocolo GPRS, almacena los datos climáticos en la nube para luego utilizar el software WeatherLink y crear gráficos, generar resúmenes, entre otros [2].
- Weather Link, es una aplicación que permite al usuario poder ver sus datos meteorológicos personales haciendo uso de las estaciones “Davis Instruments” y pronósticos locales. Además, permite comparar los pronósticos tales como lluvia, barómetro, radiación solar y gráficos interactivos UV con terceros, ya que empresas, escuelas y aficionados comparten sus datos meteorológicos con el mundo mediante esta aplicación [29].



- Otras alternativas comerciales similares a este desarrollo:
  - Davis Instruments,
  - Spectrum Technologies (WhatchDog Weather Station),
  - Campbell scientific.

Estas compañías ofrecen estaciones meteorológicas automáticas completas que se basan en el cálculo de la evapotranspiración, a través de la ecuación de FAO Penman Monteith y que cuentan con sensores de radiación solar, humedad relativa, temperatura, precipitaciones y velocidad del viento. Las estaciones meteorológicas de estas compañías tienen un alto nivel de desarrollo, siendo precisas en sus mediciones, por lo que su utilidad puede ser a nivel comercial y de investigación. Debido a esto, su costo es mayor. Para el caso de una estación WatchDog (más software), el precio es de 3.441.842 CLP, mientras que una estación marca Metos modelo iMetos ag 280 tiene un valor de 2.720.474 CLP (no incluye software).

Además, dependiendo del software que contengan, entregan los datos climáticos brutos, o en su defecto el nivel de evapotranspiración de referencia, por lo que su principal debilidad es la necesidad de contar con personal altamente especializado para la interpretación de la información.

A continuación, en la Tabla 1.1 se entrega, a modo de resumen, una comparativa entre la propuesta de esta memoria respecto a trabajos relacionados. Se puede apreciar claramente que, si bien todas las plataformas tienen varias cosas en común, no hay ninguna que este enfocada directamente en la agricultura y, por lo tanto, que sea capaz de tomar datos recibidos por los sensores para lograr determinar el volumen de agua que un campo requiere.

	Propuesta	My AcuRite	Ambient Weather	Weather Link	Davis Instruments	Spectrum Technologies	Campbell scientific
DATOS CLIMÁTICOS	✓	✓	✓	✓	✓	✓	✓
CANTIDAD DE AGUA REQUERIDA	✓	✗	✗	✗	✓	✓	✓
COMPATIBLE CON ANDROID	✓	✓	✓	✓	✓	✓	✓
ENFOCADA A LA AGRICULTURA	✓	✗	✗	✗	✓	✓	✓
SINCRONIZACIÓN EN LA NUBE	✓	✓	✓	✓	✗	✗	✗
BAJO COSTO	✓	✓	✓	✓	✗	✗	✗

Tabla 1.1: Tabla comparativa de los sistemas existentes

## 1.7. Resumen del capítulo

Una vez finalizado este capítulo, podemos comprender a grandes rasgos el tema que trata la memoria pudiendo conocer el contexto dentro del cual está enfocado, siendo netamente la eficiencia del uso del agua utilizada para agricultura en la región del Maule.

De acuerdo con los objetivos planteados, esta propuesta pretende lograr la creación de una estación meteorológica de bajo costo capaz de comunicarse con una aplicación móvil y luego aplicar un modelo matemático capaz de determinar la cantidad de agua que un campo requiere día a día. Este elemento resalta la novedad de la memoria, siendo un punto relevante y diferenciador respecto a los trabajos relacionados.

## 2. Marco Teórico

---

El proyecto propuesto integra una serie de conceptos que van desde agrícolas hasta modelos matemáticos de programación. En este capítulo se describen estos y otros conceptos, brindando al lector los conocimientos necesarios sobre el contexto teórico de desarrollo del proyecto. Además, se incluyen aspectos relevantes como las tecnologías, metodologías y mecanismos de prueba.

### 2.1. Conceptos y tecnologías

Para poder comprender en su totalidad esta memoria se debe dejar en claro el significado de ciertos conceptos dentro de este documento, para lo cual, se detallan en la siguiente sección:

- Estación meteorológica: entiéndase como estación meteorológica a un conjunto de sensores (humedad, temperatura, viento, etcétera) interconectados mediante un protocolo de comunicación, destinados a medir y registrar diversas variables meteorológicas. Estos datos se utilizan como entrada para el modelo matemático que dará como resultado la estimación de agua requerida.
- Sistema semiautomático de agua intrapredial: consta de una serie de sensores de humedad del suelo, conectados de forma inalámbrica, que permiten definir parámetros objetivos para la apertura o cierre de una bomba de agua para regar la cosecha. Para esto, se define un valor crítico de humedad relativa en relación con el tipo de suelo y una vez que el promedio de los nodos de humedad alcanzan el valor establecido, la bomba queda activa hasta que el suelo alcanza su capacidad de campo [31].

- **Microcontrolador:** es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida. Estas partes están interconectadas dentro del microcontrolador, y en conjunto forman lo que se le conoce como micro computadora [30].
- **IoT:** el Internet of Things (IoT) describe la red de objetos físicos (cosas) que llevan sensores integrados, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas a través de Internet [15].
- **Cloud Computing:** es la prestación de servicios informáticos, incluidos servidores, almacenamiento, bases de datos, redes, software, análisis e inteligencia, a través de Internet ("la nube"), para ofrecer una innovación más rápida, recursos flexibles y economías de escala [7].
- **Protocolo de comunicación MQTT:** es un protocolo de red ligero y flexible que logra el equilibrio adecuado para los desarrolladores de IoT y permite que se implemente tanto en hardware de dispositivos muy restringidos como en redes de alta latencia y ancho de banda limitado. Su flexibilidad permite soportar diversos escenarios de aplicaciones para dispositivos y servicios IoT [18].

### 2.1.1. Tecnologías utilizadas

Para la creación de la aplicación móvil y su posterior visualización de las métricas capturadas por los sensores, se disponen a continuación las herramientas y lenguajes que se utilizan.

- **Google Cloud Platform:** es un espacio virtual a través del cual se puede realizar una serie de tareas que antes requerían de hardware o software y que ahora utilizan la nube de Google como única forma de acceso, almacenamiento y gestión de datos. Como protocolo de comunicación está diseñado para trabajar directamente con el protocolo MQTT, el que permite enviar datos desde los sensores conectados a la placa, directamente a la nube. Posteriormente, los datos son procesados en la plataforma de Cloud Computing, la cual publica los datos mediante MQTT a su motor de base de datos Firebase, el cual es definido a continuación [17].
- **Firebase:** es una plataforma ubicada en la nube, integrada con Google Cloud Platform, la cual usa un conjunto de herramientas para la creación y sincronización

de proyectos permitiendo así simplificar el trabajo, ya que además, actúa como un servidor de backend en la nube capaz de almacenar datos permitiendo conexiones simples con cualquier ambiente de desarrollo [12].

- IDE Arduino: es una aplicación multiplataforma (para Windows, macOS, Linux ) que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en placas compatibles con Arduino. En este caso, será utilizado para programar la placa base ESP32, la cual se programa en lenguaje C++ . El IDE de Arduino admite los lenguajes C y C ++ utilizando reglas especiales de estructuración de códigos [19].
- Android Studio: es el entorno de desarrollo integrado oficial para la plataforma Android ya que fue diseñado específicamente para este, permitiendo desarrollar aplicaciones móviles. Es compatible con lenguajes de programación tales como: Java, Kotlin, Flutter siendo éste último el que se utilizará en el desarrollo de este proyecto [3].
- Flutter: es un SDK de código fuente abierto de desarrollo de aplicaciones móviles creado por Google. Se caracteriza por tener una curva de aprendizaje bastante rápida y por ser multiplataforma, permitiendo realizar un solo desarrollo que sea compatible tanto para Android como para iOS [13].
- R: es un entorno y lenguaje de programación con un enfoque al análisis estadístico que se utiliza en este proyecto para el análisis e implementación de los modelos matemáticos [23].
- GitKraken: para el control de versiones de la aplicación se utiliza GitHub sincronizado con GitKraken, ya que es más completo en todos los sentidos, al ser una herramienta más potente, sencilla de utilizar y más prolija en cuanto a la interfaz de usuario. De esta manera, permite realizar todas las funcionalidades que pueda ofrecer cualquier controlador de versión en un solo lugar, de fácil acceso y con buena sincronización [16].
- Trello: es un software de administración de proyectos que permite mantenerlos ordenados y organizados, siendo utilizado en un formato Kanban; definiendo y planificando el proyecto desde su concepción hasta su finalización [27].

- EasyEDA: es un software que permite realizar de manera gratuita y sin ninguna restricción el circuito impreso PCB. Una de las ventajas de este software es que se utiliza desde una página web y no necesita de la instalación de algún plugin adicional, lo cual lo hace muy atractivo para realizar un circuito rápido sin poseer un equipo muy potente. En el contexto de este proyecto, esta herramienta se utilizará para el diseño del circuito de la estación meteorológica [6].

## 2.2. Metodología

Para lograr los objetivos previamente establecidos se optó por un modelo metodológico, el cual es habitualmente empleado en investigaciones científico-técnicas.

### 2.2.1. Análisis del estado del arte

Tras la búsqueda exhaustiva de información científica en diversas bases de datos disponibles, se obtuvieron resultados sobre los cuales se realiza un estudio para mostrar la situación actual de las investigaciones, en cuanto al desperdicio hídrico que se presenta hoy en Chile se refiere y en los distintos protocolos de comunicación utilizados en dispositivos IoT.

### 2.2.2. Análisis de las propuestas

Se discutieron las distintas propuestas planteadas por terceros, estableciendo la mejor alternativa para la determinación del uso adecuado del recurso hídrico y protocolos de comunicación eficientes en dispositivos IoT.

### 2.2.3. Desarrollo e implementación PXP

Para el desarrollo de la aplicación móvil se ha optado por hacer uso de la metodología de programación extrema personal, o Personal Extreme Programming (PXP), la cual es una de las metodologías de desarrollo ágil más exitosas en la actualidad. Su principal autor es Kent Beck, quien eligió algunas características de otras metodologías y las relacionó de tal forma que cada una se complementara una con la otra [26]. Esta metodología tiene como base la simplicidad y como objetivo principal la satisfacción del cliente; para lograrlo se deben tomar en cuenta cuatro valores fundamentales: comunicación, simplicidad, coraje y retroalimentación. Sin embargo, se realiza una modificación para ser utilizada por

un único desarrollador, adaptando la ceremonia de la metodología (*sprint plannig, srprint review, sprint retrospective, daily scrum meeting*), para ser realizadas de forma individual y consultando con terceros el avance del proyecto. Para controlar la ejecución y desarrollo del proyecto se utiliza un tablero Trello y para el control de versiones se utiliza GIT bajo el cliente GitKraken.

Para el desarrollo de esta propuesta se contemplan las siguientes etapas de la metodología :

- Planificación de la aplicación móvil: esta etapa comprende la definición de historias de usuario con el cliente.
- Diseño: considera la creación de mockups para la aplicación y los diseños de su arquitectura.
- Codificación: corresponde a la programación o codificación de las historias o requerimientos.
- Pruebas: realización de testeo del código para verificar su funcionalidad.

### **2.3. Evaluación del proyecto**

Para evaluar los resultados de de la aplicación móvil se realiza lo siguiente:

#### **2.3.1. Prueba de caja negra**

Una prueba de caja negra consiste en verificar una cierta funcionalidad dando un cierta entrada y esperando una cierta salida, pero el evaluador no sabe cómo ni por qué ocurre, sólo que ocurre. A continuación, en la Figura 2.1 se muestra una representación gráfica del funcionamiento de una prueba de caja negra.

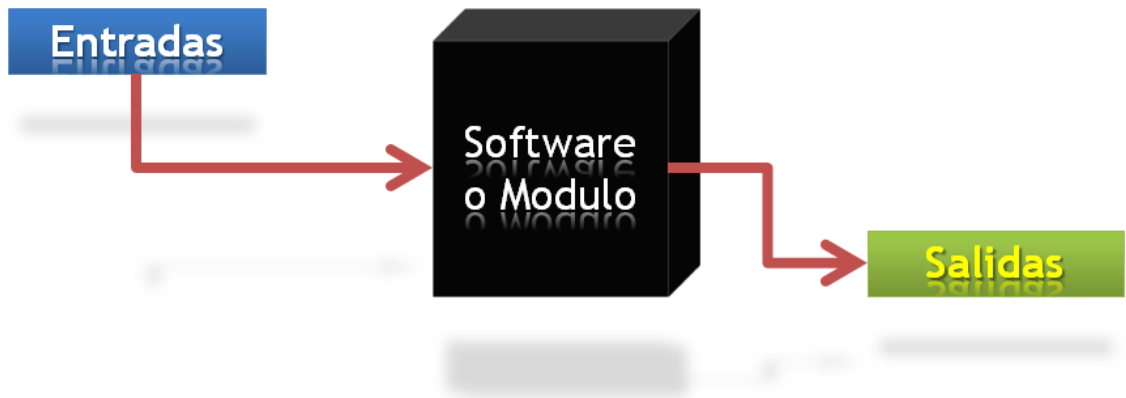


Figura 2.1: Representación de una prueba de caja negra

En este proyecto estas pruebas fueron utilizadas para comprobar el correcto funcionamiento de los sensores junto a la comunicación con la aplicación móvil y los resultados obtenidos luego de procesar los valores de los sensores con los modelos matemáticos.

### 2.3.2. Prueba de usabilidad SUS

Para validar que la aplicación sea fácil e intuitiva de utilizar es necesario utilizar encuestas de usabilidad, aplicadas directamente al agricultor, para comprobar si efectivamente es lo que él necesita y buscar la forma más cómoda para él de visualizar los datos. Para este caso se utilizaron pruebas SUS, sistema de escalas de usabilidad (por sus siglas en inglés, System Usability Scale). Esta es una herramienta metodológica muy similar a la Escala de Likert y que se usa para medir la usabilidad de un objeto, dispositivo o aplicación. La escala de medición consiste en 10 preguntas, cada una de las cuales puede ser puntuada de 1 a 5, donde 1 significa en total desacuerdo y 5 en total acuerdo [10].

En el documento de encuesta, en primer lugar, se presenta la aplicación que desea medir a los usuarios de prueba. Las preguntas realizadas siempre deben ser muy similares, lo cual es muy importante ya que para modificarlas hay que tener consideraciones que afectan los algoritmos de los cuales se obtienen los resultados.

A continuación, en la Tabla 2.1 se muestran las preguntas originales de usability.gov traducidas al español, levemente adaptadas para el uso de esta aplicación [28].



Tabla 2.1: Preguntas SUS originales traducidas al español

Número	Pregunta
1	Creo que usaría esta aplicación frecuentemente
2	Encuentro esta aplicación innecesariamente compleja
3	Creo que la aplicación fue fácil de usar
4	Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación
5	Las funciones de esta aplicación están bien integradas
6	Creo que la aplicación es muy inconsistente
7	Imagino que la mayoría de la gente aprendería a usar esta aplicación en forma muy rápida
8	Encuentro que la aplicación es muy difícil de usar
9	Me siento confiado al usar esta aplicación
10	Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación

Para obtener los resultados, se suman los resultados promediados obtenidos de los cuestionarios, considerando lo siguiente: las preguntas impares (1,3,5,7 y 9) toman el valor asignado por el usuario, y se les resta 1. Para las preguntas pares (2,4,6,8 y 10), será de 5 menos el valor asignado por los entrevistados. Una vez obtenido el número final, se multiplica por 2,5. Realizado este cálculo se obtiene un puntaje de máximo 100 puntos, donde entre más alto mejor usabilidad entrega la aplicación.

## 2.4. Resumen del capítulo

En este capítulo se dio cuenta a todo el marco teórico que se requiere para poder comprender la memoria, dejando en claro los conceptos relacionados como también las tecnologías y herramientas utilizadas. Además, se describe la metodología utilizada y su forma de evaluación.

Para el caso particular de este proyecto es muy importante señalar que la aplicación sea simple de utilizar y sencilla de visualizar, ya que el usuario final es un agricultor, que

dará suma importancia a las pruebas de usabilidad de la aplicación para validar este punto. Del mismo modo, una vez conociendo los conceptos y tecnologías a utilizar se está en condiciones de continuar con el siguiente capítulo donde se da comienzo a la aplicación de la metodología.

## 3. Marco Metodológico

---

En este capítulo se da cuenta de la implementación de la metodología Personal Extreme Programming (PXP) al proyecto, explicando sus fases junto a su debida documentación. Además, se describe la planificación y los requisitos que el sistema requiere.

### 3.1. Personal Extreme Programming (PXP)

PXP, es una metodología muy útil para desarrollos unipersonales y rápidos ya que permite entregar prototipos funcionales en tiempos relativamente cortos. Esta metodología ajusta su fase de desarrollo a un solo programador y, por dicha razón, elimina todas las actividades grupales, simplificación que le permite ajustarse de mejor manera al presente proyecto. De manera similar ocurre con los roles, dado que PXP nos permite reducir a solo dos: cliente y desarrollador.

- **Cliente:** es quien describe lo que desea para su aplicación, es decir, nos entrega todos los requisitos que debemos cumplir. Además, determina la prioridad de una tarea sobre otra. Es muy común que durante el desarrollo de la aplicación ocurran cambios y se requieran ajustar los requisitos, por lo cual, es muy importante durante todo el proceso del desarrollo de la aplicación mantener una buena comunicación entre el cliente y el desarrollador.
- **Desarrollador:** es quien desarrolla la aplicación, por lo que debe estar en constante comunicación con el cliente, define la arquitectura de la aplicación y el mismo es el encargado de realizar pruebas y ajustar sus requerimientos. Es de suma importancia contar con una planificación de la etapa de desarrollo, con el objetivo de determinar los plazos a cumplir y llevar notas de las decisiones, cambios o problemas que surgen durante este proceso.

### 3.2. Requisitos

En esta etapa, se obtienen los requisitos que la aplicación debe cumplir, es decir, lo que la aplicación debe realizar. Como se menciona anteriormente, estos requisitos son dados por cliente e implementados por el desarrollador. Por lo cual, al inicio del proyecto, en las dos primera semanas, se organizó una reunión con el cliente enfocada en la recopilación de las funcionalidad que debe cumplir la aplicación. Para este proyecto los requisitos capturados se especifican formalmente en historias de usuarios y se deben escribir de la siguiente manera: como [rol] quiero [evento] para [funcionamiento].

Las historias de usuario se detallan en un documento llamado pila de producto donde se les asigna un identificador, enunciado, alias, estado, tiempo estimado, sprint a que pertenece y la prioridad que tiene.

A continuación, en la Tabla 3.1 se muestra un ejemplo de la definición para una historia de usuario con todos sus apartados.

Tabla 3.1: Ejemplo de historia de usuario

Historia de usuario	
Nombre: US001	Número: 001
Modificaciones: Sin modificaciones	
Usuario: Agricultor	Iteración asignada: 1
Prioridad: Alta	Puntos estimados: 160
Riesgo: Critico	Puntos reales: 150
Descripción: Como agricultor quiero conocer datos meteorológicos de mi campo tales como: humedad, temperatura, presión atmosférica, luminosidad, velocidad del viento, dirección del viento y cantidad de lluvia caída para saber las condiciones que mi campo se encuentra expuesto día a día.	
Observaciones: Esta tarea implica la construcción del hardware necesario para obtener dichos datos meteorológicos.	

### 3.3. Planificación

En esta etapa se describen, a modo general, las iteraciones que implica el desarrollo de esta aplicación. Para cada iteración se define un tiempo de duración de 3 semanas aproximadamente y, además, se realiza la estimación del esfuerzo que conlleva cada tarea.

Para llevar registro de la planificación se hizo uso de un tablero Trello, el cual permite hacer seguimiento visual al flujo de trabajo. Dentro de este tablero, se definen tarjetas que representan el estado actual del desarrollo de dicha tarea, la cual se irá moviendo de sección en sección según su progreso.

A continuación, en la figura 3.1 se muestra el flujo de trabajo utilizado en el proyecto.

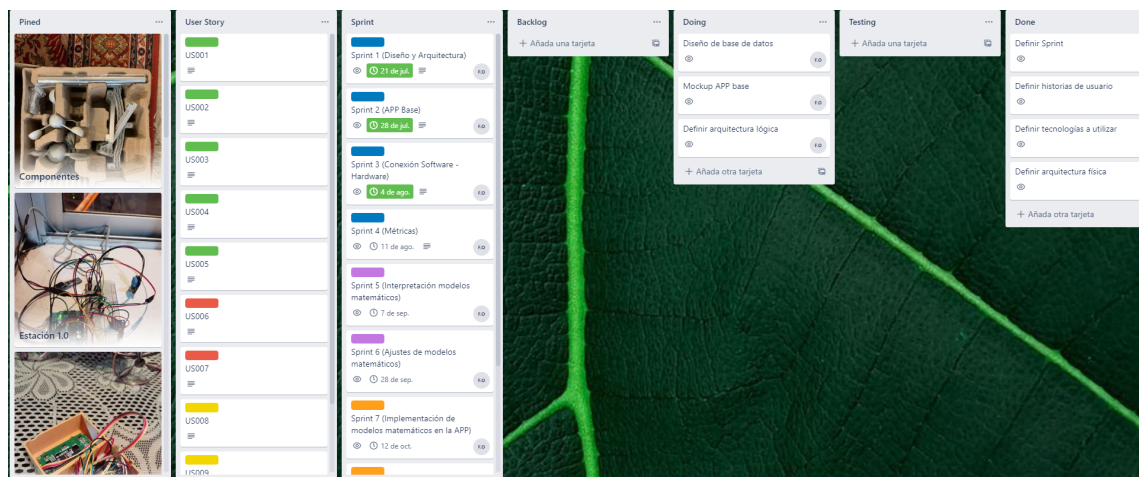


Figura 3.1: Flujo de trabajo utilizado en el proyecto

- **Pined:** en esta columna se define la información relevante para el proyecto y/o de acceso continuo, por lo cual está siempre disponible y de fácil acceso. En este apartado, además se dejan establecidas todas las tareas extras que conlleva el proyecto, por ejemplo, en este caso los prototipos de construcciones de las estaciones meteorológicas junto a su descripciones y códigos realizados, diseños de arquitectura, circuitos digitales, entre otros.
- **User Story:** en esta sección se definen las historias de usuario obtenidas en la reunión con el cliente junto a su respectiva definición, las cuales serán asociadas a tareas específicas. A cada historia de usuario se asocia un identificador al cual, posteriormente, se referencia desde una tarea o varias tareas designadas para completar en totalidad la historia.

- **Sprint:** en esta columna se definen todas las iteraciones del proyecto, asignando un encargado de la tarea, duración y estimación. Cada sprint tiene un nombre que generaliza la función principal de ese sprint, el que luego será desglosado en varias tareas que, a su vez, se deben desarrollar para completar el sprint en su totalidad y generar un prototipo funcional.
- **Backlog:** en esta columna se definen todas las tareas que estarán en la lista de tareas por hacer, las cuales van asociadas a un respectivo sprint asignando un encargado de la tarea, duración y estimación. Estas tareas, una vez comenzadas, se mueven a su siguiente sección.
- **Doing:** columna intermedia de transición donde se mueven las tareas del backlog una vez que se comiencen a realizar hasta que se encuentren en condiciones de pasar a testing.
- **Testing:** columna que mantiene las tareas que actualmente se encuentran en periodo de validación de prueba antes de pasar a done donde la tarea finaliza. En caso de encontrarse algún error en la fase de testing, esta tarjeta deberá ser devuelta al área de doing para reparar los errores encontrados.
- **Done:** columna final donde quedan las tareas en estado de término. Para que una tarea llegue a este estado debe pasar las pruebas y ser validada.

A continuación, en la Tabla 3.2 se especifica la planificación para cada iteración dejando detallada su fecha de inicio, fecha de término, número de iteración y objetivo general que se desea completar al finalizar dicha iteración. Cada iteración va relacionada a cumplir con la realización de las historias de usuario y genera tareas específicas para lograrlo.

### **3.4. Iteraciones**

Esta es la etapa iterativa más importante de la metodología PXP. Tiene una duración aproximada de 3 semanas por iteración y cada iteración tiene como objetivo tener un prototipo funcional al término de cada una.

Cada iteración cuenta de 5 fases; inicio de iteración, diseño, implementación, prueba del sistema y retrospectiva. A continuación se describe cómo se implementa cada una de estas fases.

Tabla 3.2: Planificación de las iteraciones

Fecha de inicio	Fecha de termino	Iteración	Objetivo general
2 de junio de 2020	16 de junio de 2020	1	Esta iteración contempla la obtención de requisitos y creación de historias de usuario, diagrama de base de datos, definición de tecnologías y alternativas para el desarrollo de la APP, diseños preliminares de la APP, arquitectura física y lógica.
16 de junio de 2020	7 de julio de 2020	2	Esta iteración considera la creación del proyecto base sobre el cual se va a desarrollar la APP y la construcción preliminar del hardware a utilizar.
7 de julio de 2020	21 de julio de 2020	3	Esta iteración contempla la integración de la lectura de los sensores desde la APP.
21 de julio de 2020	11 de agosto de 2020	4	Esta iteración contempla la incorporación de datos históricos en la APP.
11 de agosto de 2020	7 de septiembre de 2020	5	Esta iteración considera la interpretación de los modelos matemáticos.
7 de septiembre de 2020	28 de septiembre de 2020	6	Esta iteración contempla los ajustes necesarios de los modelos matemáticos para la aplicación.
28 de septiembre de 2020	12 de octubre de 2020	7	Esta iteración considera la implementación de los modelos matemáticos en la aplicación.
12 de octubre de 2020	9 de noviembre de 2020	8	Esta iteración contempla la visualización de los resultados de los modelos matemáticos en la aplicación.

### **3.4.1. Inicio de iteración**

Se comienza con la elección de las historias de usuarios a desarrollar por cada iteración, dicha elección, se registra en la pila del producto y se definen las tareas para dar cuenta a la/las historias de usuario involucradas en la iteración. Se define la duración de la tarea junto con su estimación de esfuerzo.

### **3.4.2. Diseño**

En esta etapa se generan diagramas relevantes para el desarrollo del proyecto, tales como, arquitectura física, lógica, diagrama de base de datos, etcétera, con la finalidad de orientar al cliente y al desarrollador a lo largo del proyecto. Generalmente, en las primeras iteraciones esta etapa consume bastante tiempo y es la etapa más relevante dado que es aquí donde se define la principal arquitectura que sostendrá todo el proyecto. Luego, en las siguientes iteraciones se utiliza esta fase para realizar cambios menores a los distintos diagramas en caso de requerir algún tipo de ajuste o perfeccionamiento.

### **3.4.3. Implementación**

En esta fase se realiza una adaptación de la metodología original PXP, ya que se altera la etapa de pruebas unitarias debido a que, en este caso del proyecto, las pruebas se basan en acción - respuesta de los sensores a la aplicación, por lo cual, en la fase de implementación se codifican las tareas seleccionadas para la iteración.

### **3.4.4. Prueba del sistema**

Etapa donde se valida el funcionamiento del sistema completo enfocándose en las nuevas características agregadas recientemente en la iteración respectiva. Para dar inicio a la siguiente iteración, es necesario cumplir con todas las pruebas y se debe efectuar lo solicitado por la historia del usuario. Para validar lo anterior, se aplican pruebas de caja negra. Si alguna tarea no se logra completar, no se agrega como funcionalidad del prototipo y se mueve a la siguiente iteración para terminar su desarrollo.

### **3.4.5. Retrospectiva**

Esta es la etapa de finalización de cada iteración donde interactúa el desarrollador y el cliente. En esta etapa, se revisa el prototipo funcional generado y la pila de producto



analizando los objetivos planteados para la iteración. Además, en esta etapa se realiza el ajuste de estimación haciendo un contraste con el tiempo estimado y el tiempo real demorado de la iteración. Generalmente, esta fase también es utilizada para detectar la forma de trabajo del desarrollador, para verificar el progreso del desarrollo y detectar algún tipo de contratiempo o mejoras en la planificación de la siguiente iteración.

### **3.5. Justificación de la metodología**

El desarrollo de software se basa en la selección adecuada de la metodología. Si bien, actualmente hay varias metodologías para el desarrollo de software, debemos analizar y justificar para determinar la más adecuada. La presente investigación considera un trabajo unipersonal y un modelo de desarrollo ágil, por lo que solo se tienen en cuenta estrategias que no requieran demasiado detalle de profundización en la documentación, que tengan un periodo corto de finalización y que facilite el trabajo.

En algunas investigaciones se utiliza con frecuencia la metodología Scrum para aplicaciones similares, sin embargo, esta metodología requiere modificaciones para ser adaptadas a un desarrollo unipersonal. Estas modificaciones al final tienden a forzar una metodología grupal y en ocasiones no se obtienen los resultados esperados. Por ende, teniendo todo esto en consideración, se opta por utilizar una metodología de desarrollo Personal Extreme Programming (XP), la cual, es una programación extrema de carácter personal.

### **3.6. Resumen del capítulo**

En este capítulo se presentó la forma de implementación de la metodología XP adaptada a este proyecto, especificando la forma de trabajo, planificación y el cómo se llevarán a cabo las iteraciones. Es muy importante seguir minuciosamente la metodología planteada en este apartado, ya que, una falta en ella podría ocasionar un mal producto y, en consecuencia, conllevar a futuros problemas.

En el siguiente capítulo, se da a conocer el desarrollo completo de esta memoria a nivel de hardware y software, detallando cada uno de los elementos desarrollados para lograr una correcta ejecución del proyecto.

## 4. Desarrollo de la solución

---

En este capítulo se presenta la construcción e implementación de la estación meteorológica de bajo costo. Además, se detalla el proceso seguido en cada iteración para lograr la construcción del software, con el cual interactúa la estación meteorológica. La estructura del capítulo se encuentra dividida en dos grandes secciones: hardware y software.

### 4.1. Hardware

Para la construcción de esta memoria se desarrolló un prototipo capaz de medir variables tales como: temperatura, humedad relativa, presión atmosférica, velocidad del viento, dirección del viento y luminosidad. Además, para que el dispositivo fuera autónomo se implementa un sistema de carga y descarga de batería en conjunto con la adición de un panel solar. A continuación se detallan los sensores y componentes utilizados para la construcción de la estación, los diseños de los circuitos impresos y el funcionamiento y programación de la placa.

#### 4.1.1. Sensores y Componentes

Se detalla la lista de sensores y componentes que son parte de la estación junto a una pequeña descripción y sus especificaciones técnicas.

##### **Sensores y Componentes:**

- **BME280:** es un sensor de temperatura, humedad relativa y presión atmosférica especialmente desarrollado para aplicaciones móviles y donde el tamaño y el bajo

consumo de energía son parámetros clave de diseño. Es un dispositivo con alta precisión y perfectamente factible para un bajo consumo de corriente, estabilidad a largo plazo y altamente robusto [5].

Datos técnicos:

- Rango operacional:

Presión atmosférica: 300 a 1100 [hPa]

Temperatura: -40 a 85 [°C]

Humedad relativa: 0 a 100 %

- Precisión absoluta:

Presión atmosférica: +/- 1 [hPa]

Temperatura: +/- 1 [°C]

Humedad relativa: +/- 3 %

- **BH1750:** es un sensor digital de intensidad de luz ambiente, tiene un conversor ADC de 16 bits interno. Esta es una versión mejorada del típico sensor de luz, basado en un LDR, el cual simplemente entrega un valor analógico. El BH1750, entrega automáticamente el valor en Lumenes (desde 1 lx hasta 65535 lx) [8].

Rangos de medición:

- Noche: 0.001—0.02 [lx]
  - Luz Lunar: 0.02—0.3 [lx]
  - Nublado Exterior: 50—500 [lx]
  - Soleado Exterior: >500 [lx]
- **Pluviómetro:** es un tipo de cubo basculante de vaciado automático. Cada 0.2794 [mm] de lluvia provoca un cierre de contacto momentáneo, el cual se registra con un contador digital y entrada de interrupción de microcontrolador en caso que este se encuentre "dormido"[4].
  - **Anemómetro:** mide la velocidad del viento cerrando un contacto como un imán que pasa por un interruptor. Una velocidad del viento de 2,4 [km/h] provoca que el interruptor se cierre una vez por segundo [4].

- **Veleta:** posee ocho interruptores, cada uno conectado a una resistencia diferente. La veleta a medida que gira cierra dos interruptores a la vez, permitiendo generar hasta 16 diferentes posiciones para indicar la dirección del viento. El cierre del contacto del interruptor produce una salida de voltaje que se puede medir con un convertidor, de analógico a digital [4].

Originalmente este sensor fue diseñado para trabajar con un voltaje de entrada de 5 [V], lo cual permite generar entre los diferentes cierres del contacto, salidas que van desde los 0.32 [V] hasta los 4.78 [V]. Sin embargo, el conversor analógico digital que viene incorporado con la placa central (ESP32), solo es capaz de medir entradas analógicas hasta 3.3 [V]. Por este motivo, se realiza una modificación a la arquitectura física de este circuito para que, en vez de trabajar con 5 [V], sea capaz de operar a 3.3 [V]. Para realizar dicho cambio se modifica la resistencia del circuito de 10 [k $\Omega$ ] a una de 4.7 [k $\Omega$ ].

Lamentablemente, este cambio funcional limita el rango de valores medidos de dirección del viento, pasando de 16 posibles mediciones a 8. Sin embargo, estas son suficientes para un correcto funcionamiento de la estación. Posteriormente se muestra la tabla de conversión original y la adaptación para este proyecto, donde se demuestra el planteamiento anterior.

- **TTGO LORA32 V2.1.1.6:** es una placa que tiene integrado el microcontrolador ESP32, el chip LoRa SX1276 y una pantalla OLED de 0,96 pulgadas. Además, tiene un casillero disponible para poder agregarle una tarjeta MicroSD. La placa permite comunicación a través de: WiFi, Bluetooth y LoRa (LoRaWan) [20].

Para el desarrollo de este proyecto se utilizan dos placas: una emisora y otra receptora. La emisora es la encargada de recolectar todas las mediciones de los sensores conectados, para luego hacer el envío de estos mediante el protocolo de comunicación LoRaWan. La placa receptora mediante WiFi se encarga de subir las mediciones recolectadas directamente a la nube.

Con el fin de ahorrar energía, gracias a los distintos parámetros de configuración de la placa, se puede programar de tal forma de que solo realice mediciones de los datos cuando sea requerido y mientras tanto, se pone en modo deep sleep, enviando la placa a un estado de hibernación, disminuyendo considerablemente su consumo (tan solo 10  $\mu A$ ).

- **RTC DS3231 Reloj de tiempo real:** permite obtener mediciones de tiempo en cada uno de los módulos de la estación meteorológica (segundos, minutos, horas, días, semanas, meses y años). Posee un oscilador interno compensado por temperatura, lo que hace que su precisión sea muy alta. El DS3231 incorpora medición y compensación de la temperatura garantizando una precisión de al menos 2ppm, lo que equivale a un desfase máximo 172 ms/día o un segundo cada 6 días [21].

Si bien es cierto, la placa controladora tiene incorporado un reloj interno, requiere de conexión WiFi para poder sincronizarlo con el horario local. Por estas razones, se opta por incorporar un reloj externo ya que en la mayoría de los campos de la Región del Maule la conexión a internet es poco factible.

- **Cargador:** es un módulo que permite la carga y descarga en simultáneo de la estación en conjunto con el uso de un panel solar.
- **Regulador:** permite regular el voltaje entregado por el panel, transformando su voltaje variable en uno constante. Para este caso, el voltaje se fija en 5 [V].

#### 4.1.2. Diseño de placa de circuito impreso

Para el desarrollo de esta memoria se diseñaron 7 versiones de placas hasta llegar a una que fuera estable, funcional y completa. En estas secciones se describen cada una de ellas, haciendo énfasis en sus ventajas e inconvenientes. Se decide incluir esta información para que el prototipo pueda ser replicado.

- **Micromet V1.0:** esta versión de la placa está compuesta por un ESP32, una pantalla Oled, BME280, BH1715, micro SD, anemómetro, veleta, pluviómetro. Sin embargo, esta versión no es capaz de enviar datos a larga distancia ya que el controlador solo permite tener conexión directa con la nube mediante WiFi. De acuerdo a la realidad de la región esta alternativa no es viable, ya que la conectividad en medio de un campo es casi nula.
- **Micromet V2.0:** en esta versión de la placa se cambia el microcontrolador, pasando de estar compuesta por un ESP32 a un TTGO ESP32 V2.16. Esto permite tener un módulo de radio frecuencia y además tiene incorporada una pantalla oled y el lector de tarjetas micro SD. El resto de los componentes (BME280, BH1715, anemómetro, veleta y pluviómetro) se mantienen.

- **Micromet V3.0:** en esta versión del prototipo se realiza una mejora de diseño, a nivel de estructura de los componentes dentro de la placa.
- **Micromet V4.0:** en esta versión se comienza a trabajar de forma inversa, ya que en las anteriores versiones los componentes se soldaban sobre las pistas. Además, se incorporan dos agujeros en la placa para sostener los conectores RJ11 de los sensores de viento y lluvia respectivamente.
- **Micromet V5.0:** en esta versión de la placa se incorpora un reloj al circuito para así poder almacenar los datos recolectados con una fecha y hora de la toma de muestra. Esta información es muy importante, pues permite generar una marca (*timestamp*) a cada medición, para su posterior procesamiento.
- **Micromet V6.0:** en esta versión de la placa se incorporan dos nuevos componentes: cargador de batería y convertidor de voltaje para la adición de un panel solar.
- **Micromet V7.0 (prototipo 1):** esta versión de la placa se considera como el primer prototipo, el cual, a diferencia de la versión anterior (6.0) se rempazan los conectores de los sensores BME280 y BH1750 por puertos con pines RJ11, para una mayor facilidad y seguridad de conexión. A continuación, se presenta el diagrama de diseño impreso para la actual versión de la estación meteorológica.

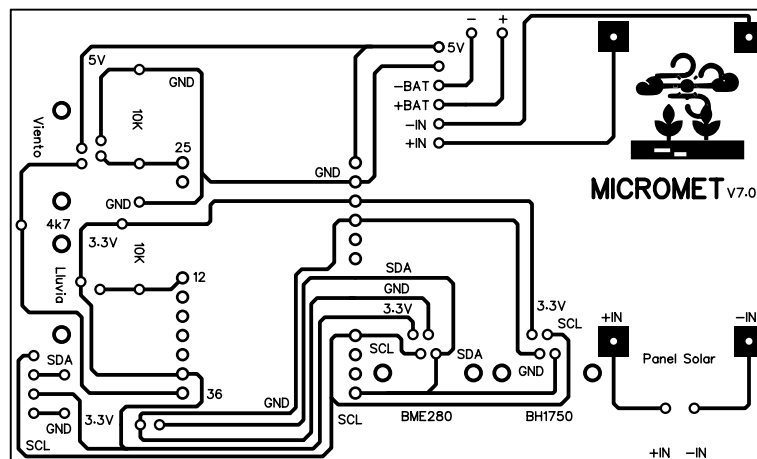


Figura 4.1: Diseño de circuito impreso V7.0

Para más detalles sobre las versiones de las placas consultar el primer anexo, donde se encuentra el diseño PCB de cada una de ellas.

### 4.1.3. Programación de la placa

Para la construcción de esta estación se realiza una programación en Arduino, la cual toma el muestreo de los datos cada 15 minutos, enviando el controlador a estado de hibernación el resto del tiempo. Esto, permite reducir el consumo y brindar una mayor autonomía. Además, este es el tiempo de medición estándar de cualquier estación meteorológica. Cabe destacar, que se programan dos placas, una emisora y otra receptora. las cuales tienen el siguiente flujo de control:

- La placa emisora comienza verificando si se ha realizado alguna medición de precipitación, que deba ser registrada. Luego, recolecta datos de humedad, temperatura, presión atmosférica y reserva 5 segundos para analizar la velocidad del viento y entregar un valor en metros por segundos. Una vez obtenidas las mediciones, se analiza la dirección del viento para determinar la dirección de la ráfaga.
- Adicionalmente, la placa almacena los datos recibidos en una memoria micro SD y hace el envío de ellos, a través del módulo de radiofrecuencia, con destino a la estación receptora. Una vez que los datos son enviados a la placa receptora, la placa emisora entra en modo deep sleep, el cual es interrumpida a los 15 minutos, o con alguna interrupción producida por alguna precipitación. En caso de detectar una interrupción, se aumenta el contador de lluvia y vuelve quedar en estado de hibernación, guardando el valor en la memoria interna para ser contabilizado en la próxima medición.
- Por otro lado, en la placa receptora, una vez que recibe alguna cadena de datos, es procesada para dividirla y almacenar cada valor en su variable correspondiente. Finalmente, una vez procesados los datos, son enviados a la nube mediante protocolo MQTT, almacenándolos en una base de datos en tiempo real de Firebase.

La figura 4.2 presenta un diagrama que muestra la secuencia de programación del microcontrolador.

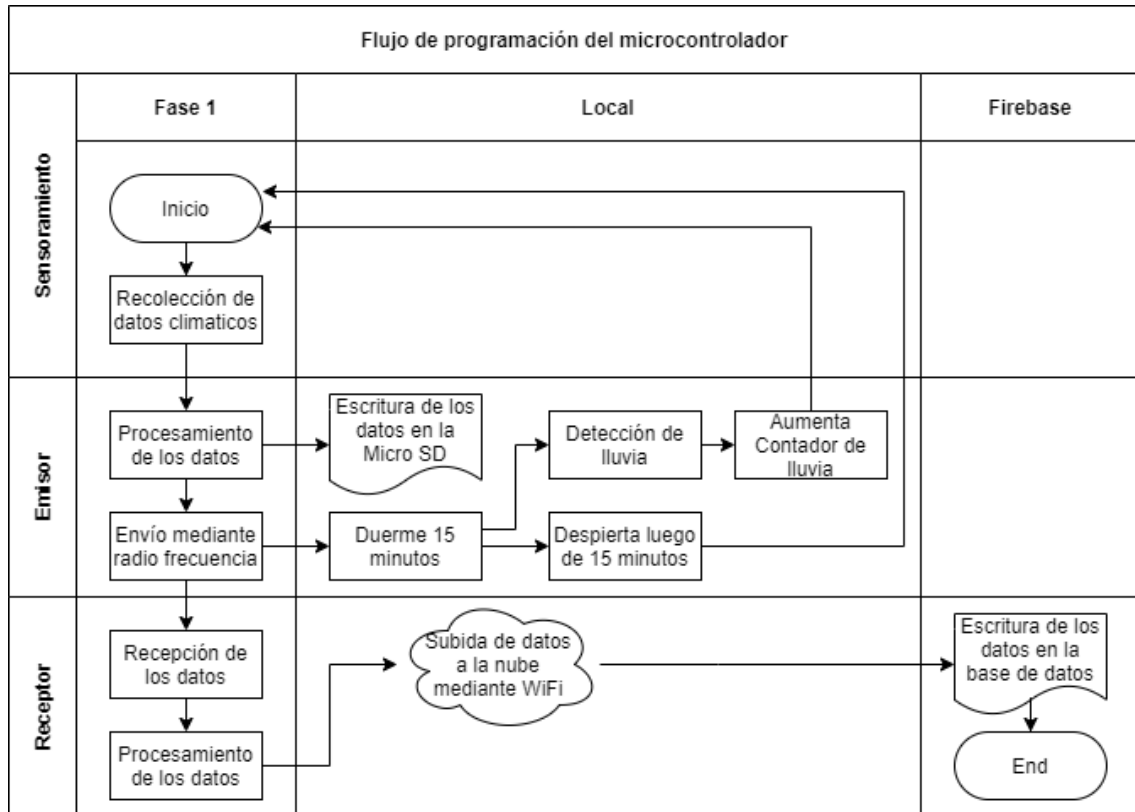


Figura 4.2: Flujo de programación del microcontrolador

## 4.2. Software

Una vez implementado el hardware del prototipo, se necesita una plataforma capaz de visualizar los datos generados por la estación. Para ello, se desarrolla una aplicación móvil en Flutter, capaz de mostrar estos datos y el consumo hídrico estimado mediante el modelo matemático. En esta sección se detallan las iteraciones generadas para dar cuenta del desarrollo de la aplicación móvil.

### 4.2.1. Arquitectura Física

El modelo de control de la arquitectura física se basa en un modelo de 4 capas, las cuales se pueden identificar como: recolección, transporte, subida y consumo de la información. Cada una de ellas se puede traducir del procedimiento antes explicado en la sección 4.1.3 con respecto a la programación de la placa. A continuación, en la figura 4.3 se presenta un diagrama que representa de mejor forma el flujo de los datos de la



arquitectura física.

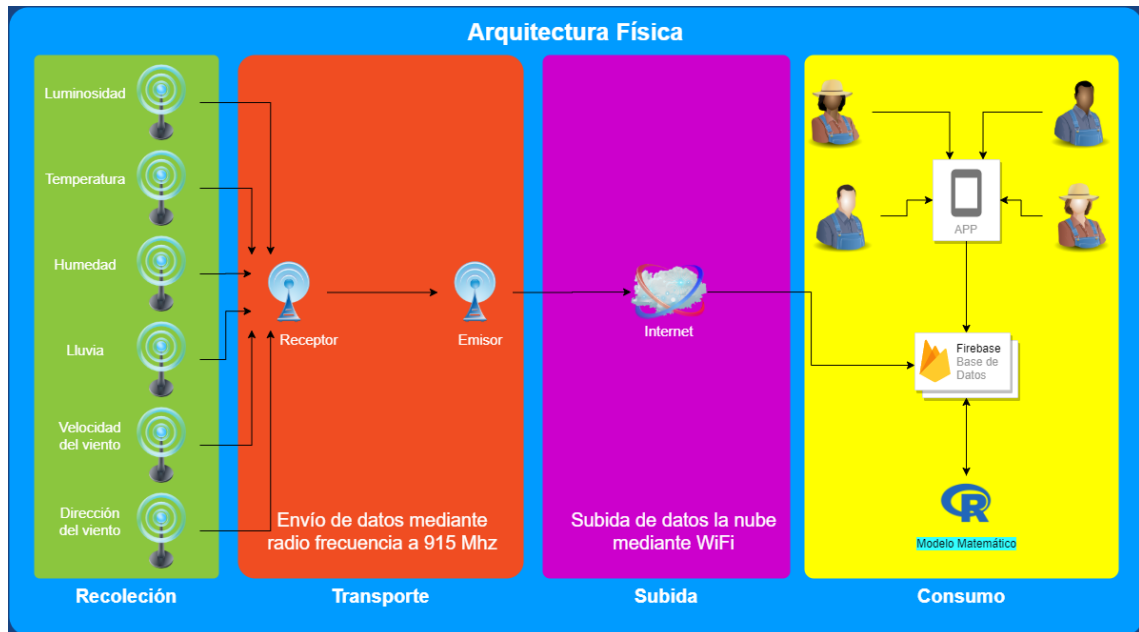


Figura 4.3: Arquitectura Física

#### 4.2.2. Arquitectura Lógica

El funcionamiento lógico de la aplicación móvil se puede representar en una arquitectura de 3 capas (Modelo - Vista - Controlador), con una capa extra de seguridad protegida por un token y reglas de lectura y escritura proporcionados directamente por Firebase. Por otro lado, el modelo matemático procesa los datos para generar un consumo de Et0 y los guarda en la base de datos, para que pueda ser consultada por la aplicación móvil.

La figura 4.4 presenta el diagrama de la arquitectura lógica, modelo, vista controlador (MVC) presentando la interacción de cada controlador con su vista correspondiente.

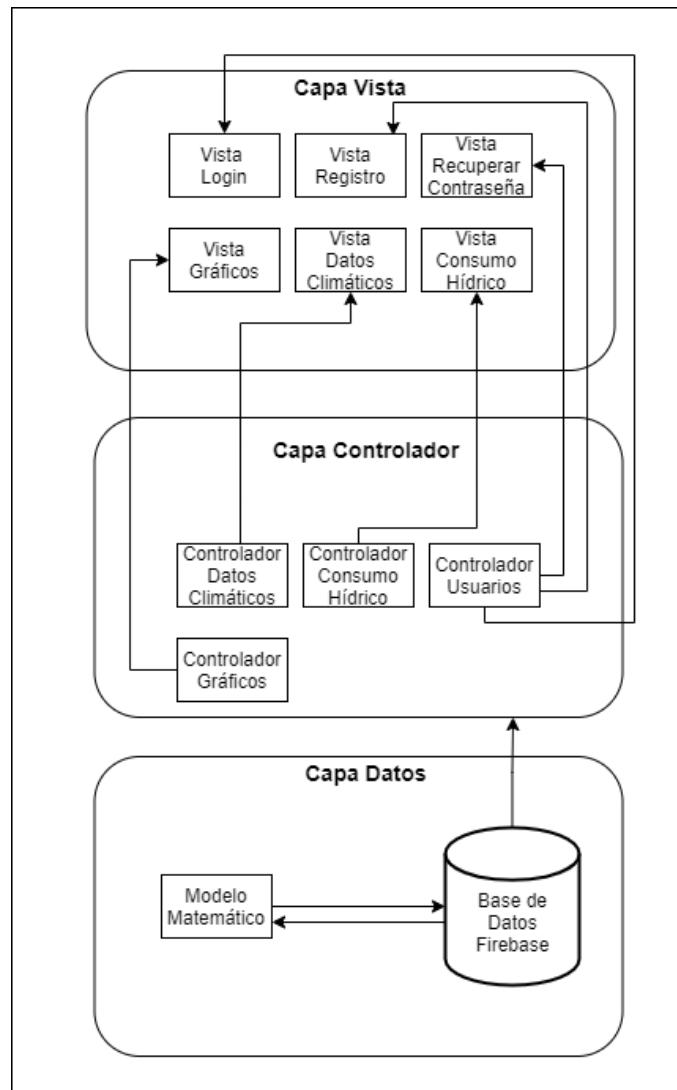


Figura 4.4: Arquitectura Lógica

### 4.2.3. Historias de usuario

Las historias de usuario se determinan basándose en los requerimientos obtenidos al inicio de este proyecto de una reunión con especialistas de la empresa Biovision Agrícola. Con la finalidad de dejar la menor carga posible de trabajo al usuario final, se realiza un cambio en la historia de usuario US006 Y US007. Inicialmente se considera que el usuario final es quien configura la estación, lo que fue modificado para que la aplicación fuese capaz de soportar dicha funcionalidad pero siendo el administrador del sistema, quien realiza la asignación de las estaciones. Del mismo modo, se modifica la programación del

microcontrolador para interactuar con estas modificaciones.

Tabla 4.1: Descripción de las historias de usuario

ID	Enunciado de la historia
US001	Como agricultor quiero conocer datos meteorológicos de mi campo tales como: humedad, temperatura, presión atmosférica, luminosidad, velocidad del viento, dirección del viento y cantidad de lluvia caída para saber las condiciones a la cuales mi campo se encuentra expuesto día a día.
US002	Como agricultor quiero poder visualizar los datos meteorológicos de mi campo en una APP móvil compatible con Android para poder monitorizar mis cultivos.
US003	Como agricultor quiero ver datos históricos diarios de los datos meteorológicos de mi campo con una antigüedad máxima de 1 semana para datos como: humedad, temperatura, presión atmosférica, velocidad del viento, lluvia caída para poder utilizarlos en la toma de decisiones.
US004	Como agricultor quiero saber cuánta es el agua que debo aplicar a mi campo día a día para reducir el consumo hídrico
US005	Como agricultor quiero recibir notificaciones día a día con la cantidad de agua que mi campo requiere para llevar un control del agua que este pierde.
US006	Como agricultor quiero poder asociar mi estación meteorológica con una aplicación móvil para poder visualizar los datos desde mi dispositivo.
US007	Como agricultor quiero poder agregar mas estaciones a mi perfil para poder monitorear todos mis campos.
US008	Como agricultor quiero poder visualizar todas mis estaciones meteorológicas en una misma aplicación móvil para mantener todos mis campos desde una sola cuenta.

#### 4.2.4. Desarrollo de la solución

En primer lugar, se describe el desarrollo comprendido del sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales de la Región del Maule. A continuación, se describen las etapas dispuestas en el apartado de iteraciones, que dicta la metodología de desarrollo PXP, dando a conocer cada una de las iteraciones, con la finalidad de exponer el cumplimiento de la metodología.

## Iteración 1

El objetivo de esta iteración es definir el diagrama para la arquitectura física y lógica de la aplicación móvil. En cada etapa de la metodología PXP se realiza lo siguiente:

- Inicio de la iteración: las tareas seleccionadas corresponden a:
  - Obtención de requisitos y creación de historias de usuario.
  - Selección de base de datos.
  - Definición de tecnologías y alternativas para el desarrollo de la aplicación móvil.
- **Diseño:** se definen los diagramas de arquitectura física y lógica para la aplicación móvil. La arquitectura lógica está compuesta por un modelo de tres capas, vista, controlador y datos.
- **Código:** se comienza con la formación de una base de datos capaz de soportar la inserción de datos en tiempo real desde una plataforma IoT, por lo cual, se opta por utilizar Firebase Real Time.
- **Pruebas Unitarias:** se prueba el motor de bases de datos, enviando información simulada desde un computador a la nube.
- **Refactorización:** se organiza la base de datos para que siga una estructura de árbol, capaz de soportar el modelamiento de las estaciones. Dado que un agricultor puede tener muchas estaciones, se opta por dejar como identificador principal el correo del usuario. Dentro de cada cliente se pueden encontrar varias estaciones, a la cuales se les asigna un nombre como identificador. Luego, un nivel mas abajo se encuentran los datos de las estaciones, los que corresponden a cada una de las mediciones de temperatura, humedad, presión atmosférica, velocidad y dirección del viento y lluvia caída. Por otro lado, se tiene el análisis de ET0, el cual es almacenado en la base de datos para ser consultado por la aplicación móvil.

A continuación, se presenta la estructura de árbol que representa el almacenamiento de la información en la base de datos.

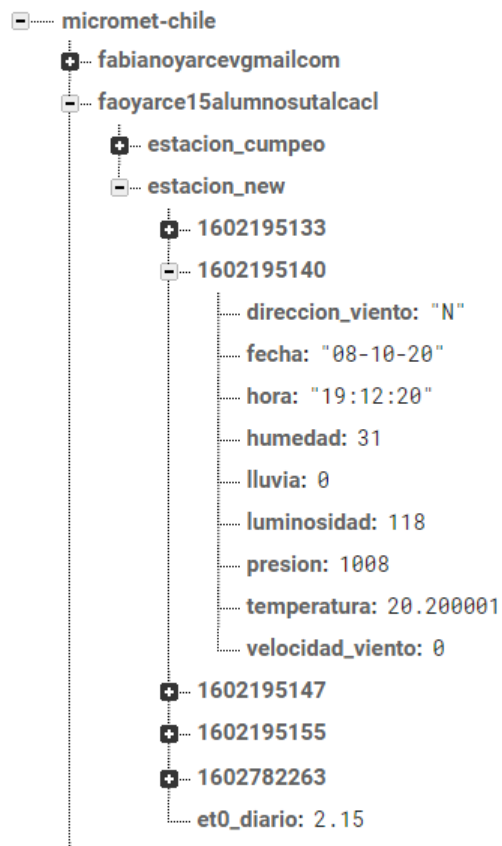


Figura 4.5: Estructura de árbol de la base de datos

- **Pruebas del sistema:** se hace uso de un sensor de temperatura conectado a un ESP32, el cual a modo de prueba por simplicidad y mediante WiFi escribe en la base de datos la estructura de árbol que soporta la inserción de los datos en tiempo real.
- **Retrospectiva:** en esta primera iteración se disponen de algunos diagramas que dan a conocer el flujo de control del sistema y se logra generar una estructura de datos funcional que es capaz de almacenar la información de las estaciones meteorológicas

## Iteración 2

El objetivo de esta iteración es generar un proyecto base, con las tecnologías seleccionadas, para comprobar el correcto funcionamiento de ellas y verificar la compatibilidad de las tecnologías tales como Flutter y Firebase.

**Estado del prototipo:** Al término de esta iteración se logra tener un proyecto básico funcional, que permite iniciar sesión mediante Google, o correo y contraseña, haciendo uso de Firebase Authentication. Del mismo modo, se crea la vista para restablecer la contraseña, haciéndolo a través de la API de Firebase. Adicionalmente, se crea y da funcionalidad a la vista de registro, la cual por simpleza solo trabaja con datos, tales como correo y contraseña.

Para facilitar el inicio de sesión al usuario se opta por dejar la sesión guardada en la memoria caché y así mantener la sesión iniciada hasta que sea cerrada manualmente por el usuario. Para ello, fue necesario agregar un botón para cerrar la sesión.

A continuación, se presentan imágenes de las vistas implementadas:



Figura 4.6: Vista Login.

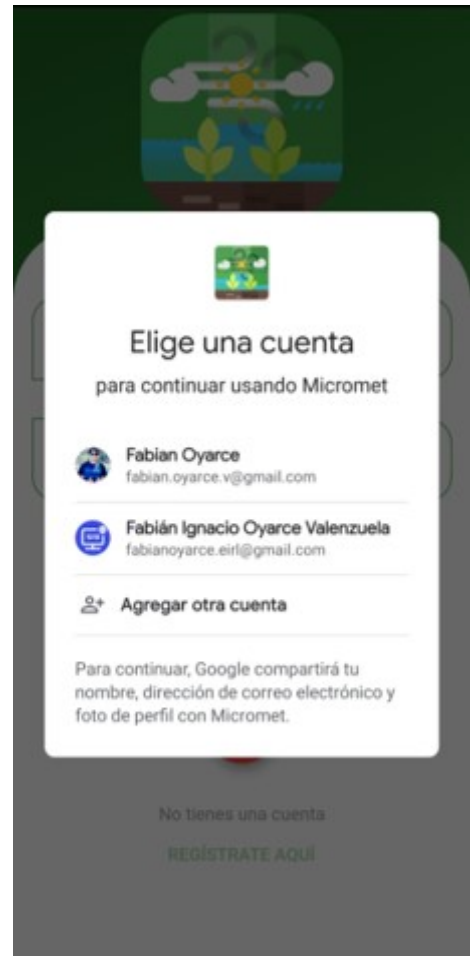


Figura 4.7: Vista Google Auth.



Figura 4.8: Vista Recuperar Contraseña.

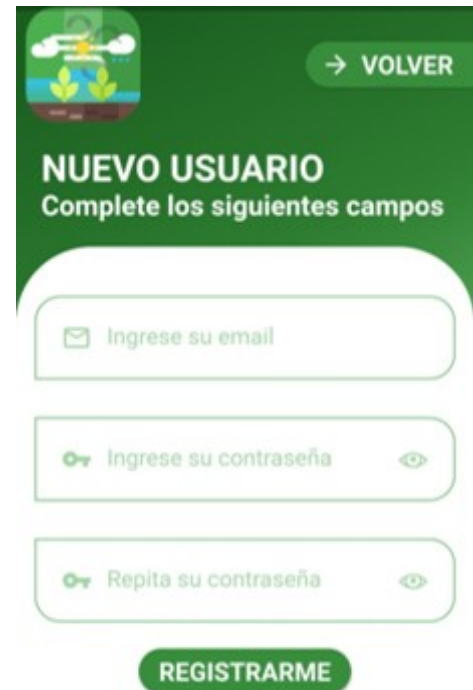


Figura 4.9: Vista Registro.

### Iteración 3

El objetivo de esta iteración es realizar la conexión Hardware-Software, permitiendo poder visualizar datos recolectados por la estación meteorológica en tiempo real desde la aplicación móvil.

**Estado del prototipo:** Al término de esta iteración se logra visualizar la humedad relativa, temperatura, presión atmosférica, cantidad de lluvia caída, velocidad y dirección del viento. Para ello, es necesario diseñar una vista amigable, que pueda mostrar todos estos datos de forma ordenada y entendible para el usuario.

Los datos mostrados en esta sección no son actualizados en forma automática, por lo cual, se agrega un botón que nos permita actualizarlos. Este botón, mediante una consulta

a la base de datos, trae el penúltimo valor recolectado y entrega la fecha de este dato, para saber cuando fue la ultima actualización.

Se opta por utilizar el penúltimo dato y no el último, dado que es una base de datos en tiempo real, y se pudiera dar el caso que justo al momento de hacer una petición, se estuvieran cargando datos en ese momento a la base, lo que eventualmente pudiera producir un error al estar leyendo y escribiendo al mismo tiempo. Lo que se traduce en recolección de datos nulos.

Adicionalmente, con el fin de hacer más interactiva la aplicación, se agrega una funcionalidad que permite cambiar el fondo de la aplicación en función del clima registrado en la actualización de los datos.

A continuación, se presentan imágenes de las vistas implementadas:



Figura 4.10: Clima Soleado.



Figura 4.11: Clima Nublado.





Figura 4.12: Clima Lluvioso.



Figura 4.13: Clima Nocturno.

#### Iteración 4

El objetivo de esta iteración es poder visualizar datos históricos de forma gráfica en la aplicación de variables tales como; humedad, temperatura, cantidad de lluvia caída, presión atmosférica y velocidad del viento.

**Estado del prototipo:** Al término de esta iteración se logran visualizar los datos esperados con una antigüedad de un día. La forma en que se muestran los datos se realiza mediante gráficos de línea.

A continuación se presentan imágenes de las vistas implementadas:



Figura 4.14: Gráfico Humedad.



Figura 4.15: Gráfico Temperatura.



Figura 4.16: Gráfico Presión Atmosférica.



Figura 4.17: Gráfico Lluvia.

### Iteración 5

El objetivo de esta iteración es poder visualizar todas las estaciones que un usuario pueda tener en una misma aplicación.

**Estado del prototipo:** Al término de esta iteración se logran visualizar todas las estaciones de un usuario en su mismo perfil. Para ello, se agrega un icono con forma de satélite, el cual permite cambiar de localización. Finalmente, esto permite tener todas las estaciones que se desean, asignando un nombre a cada una de ellas, el cual actúa como identificador para cada estación.

A continuación se presenta una imagen de la vista implementada:

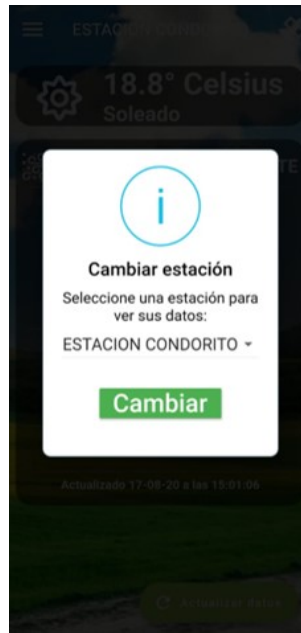


Figura 4.18: Cambio de Estación

### Iteración 6

El objetivo de esta iteración es lograr la interpretación del modelo matemático. Acá es necesario estudiar el modelo matemático de Penman Monteith, el cual se basa en el cálculo de la evapotranspiración; la que se define como la pérdida de humedad de una superficie por evaporación directa, junto con la pérdida de agua por transpiración de la vegetación.

El modelo utilizado es un trabajo previo de la tesis del Doctor Fernando Fuentes de más de 7 años de estudios y programación del modelo de Penman Monteith.

**Estado del prototipo:** Luego del estudio del modelo se sacan varias conclusiones entre las cuales se pueden destacar las siguientes:

- El dato más crítico a tener en consideración para el cálculo de la evapotranspiración es la radiación. Sin embargo, la estación no es capaz de medir dicha variable. La razón principal se basa en que un piranómetro (sensor de radiación) tiene un valor aproximado de \$1.000.000 versus uno de luminosidad que cuesta alrededor de \$2.000, por lo cual, en vez de contar con un piranómetro se ha optado por utilizar un sensor de luminosidad. El problema principal de estos dispositivos es que el modelo trabaja con radiación y no con lúmenes, razón por la cual se deben realizar trabajos futuros de pruebas experimentales de los estudios del Doctor Fernando

Fuentes que se basan en trazar una regresión lineal que sea proporcional de la radiación con la luminosidad para obtener un factor de calibración de luminosidad en base a la radiación. Para ello, se diseña un hardware con 2 sensores de luminosidad y en conjunto con un piranómetro, el cual debe ser puesto en el campo para realizar la comparación y escalar los resultados en la regresión. Estos resultados permiten generar el factor de calibración en trabajos futuros.

- Otro punto interesante a destacar de los análisis es el uso de la precipitación. A simple vista uno se podría imaginar que la cantidad de lluvia caída es muy importante para poder determinar cuánta agua aplicar en el campo. Sin embargo, al momento de realizar el análisis, se determinó que esta variable no afecta en ningún cálculo para la obtención de la evapotranspiración, ya que se basa en la pérdida de agua, y no en el ingreso de ella a la planta.
- Finalmente, se tiene la interpretación del modelo aplicado a diferentes cultivos. Esto es posible gracias a un coeficiente de cultivo estándar que está definido para cada plantación. Dicho coeficiente, para ser calculado de forma correcta, requiere que la estación debe estar ubicada en una zona de referencia, cubierta por pasto y adicionalmente, en una posición establecida a una altura determinada.

Siguiendo las instrucciones establecidas se consigue lo que se conoce como una zona de referencia la que permite tener una evapotranspiración multiplicable por cualquier factor de cultivo correspondiente para cada caso.

### **Iteración 7**

El objetivo de esta iteración es lograr la sincronización del modelo matemático con la base de datos.

**Estado del prototipo:** Al finalizar esta iteración se logra realizar la conexión del modelo matemático en R con la base de datos de la estación meteorológica. Para ello, mediante uso de las API de Firebase, la autenticación con token mas unas modificaciones de las reglas de escritura y lectura se conecta el modelo directamente a la base de datos, permitiendo así consumir las variables.

El modelo matemático toma como referencia para calcular la evapotranspiración diaria 12 valores, uno por cada hora. Para ello, se debe crear una estación de prueba a nivel de base de datos la cual sea capaz de almacenar 12 valores diarios, uno por cada hora y modificar el modelo para que lea directamente estos datos y los almacenara en un frame de 12 valores por cada variable.

Otras de las modificaciones que se deben hacer al modelo matemático para que pueda interactuar con la base de datos es modificar la forma en que los datos se almacenan en el frame, ya que originalmente se encuentran organizados por filas pero se debieron cambiar por columnas para que siguiera la estructura de orientación de la base de datos.

A continuación, se presenta una imagen de la adaptación del modelo:

```

91 #DAILYET (ETO)
92 MEANT<- mean(as.numeric(Ta[2:25]))
93 meanu<- mean(as.numeric(ws[2:25]))
94 TMEDFAO <- ((MAXT+MINT)/2)
95 e0 <- 0.6108*exp((17.27* TMEDFAO)/( TMEDFAO+237.3))
96 e0max <- 0.6108*exp((17.27*(MAXT))/(MAXT+237.3))
97 e0min <- 0.6108*exp((17.27*(MINT))/(MINT+237.3))
98 es <- ((e0max+e0min)/2)
99 Pen <- (4098*(0.6108*exp((17.27*TMEDFAO)/(TMEDFAO+237.3)))/((TMEDFAO+237.3)^2)
100 ea <- (((e0min*(max(RH[2:25])/100))+e0max*(min(RH[2:25])/100))/2)
101 DPV <- (e0min+e0max)/2-ea
102 mu= mean(as.numeric(ws[2:25])) #VELOCIDAD DEL VIENTO PARA CALCULAR Ra
103 ra <- 4.72*(log10(((z-d)/zo))^2)
104 Pa <-101.3*(((293-0.0065*z)/293))^5.26 #presion atmosferica (kPa) (ECUACION REVIS
105 ps <- ((cp*Pa)/(eps*v))# psicrometric constant (kPa/C)(ECUACION REVISADA)
106 wRADIARIO <- (pi/12)*((15+0.06667*(121-120)+Sc1)-12)#ángulos de radiación solar al i
107 wRAHORARIO <- (pi/12)*(((Hor[2:25])+0.06667*(71-70)+Sc1)-12)
108 N1 <- (24/pi)*ws #Insolacion Maxima
109 Radiario <- ((24*60)/pi)* 0.082*dr*(ws*sin(lat1)*sin(ds)+cos(lat1)*cos(ds)*sin(ws))
110 Rs<- mean(as.numeric(Rs[2:25]))*0.0864
111 Rso <- ((0.75+(2*10^-5)*z))^Radiario
112 Rns <- ((1-0.23)*Rs)
113 a<- (4.903*10^-9)*(max(Ta[2:25])+273)^4)
114 b<- (4.903*10^-9)*(min(Ta[2:25])+273)^4)
115 c <- (0.34-0.14*(sqrt(ea)))
116 d <- Rs/Rso
117 e <- 1.35 *d- 0.35
118 Rn1 <- (a+b/2)*c*e
119 RnPM <- Rs-Rn1
120 Gdiaro <- RnPM*0.1
121 etopaso1 <- (0.408*Pen*(RnPM-Gdiaro))/(Pen+ps*(1+0.34*meanu))
122 etopaso2 <- 900*meanu/(TMEDFAO+273)*ps*(es-ea)/(Pen+ps*(1+0.34*meanu))
123 ETo<- etopaso1+etopaso2

```

ETO	1.98065654897076
-----	------------------

Figura 4.19: Modelo de Penman Monteith

## Iteración 8

El objetivo de esta iteración es visualizar el resultado del modelo matemático en la aplicación móvil.

**Estado del prototipo:** Al finalizar esta iteración se logra visualizar el consumo hídrico de un cultivo. Para ello, se debe modificar el modelo matemático para que cuando se ejecute y realice el calculo de ETO, escriba el resultado directamente en la base de datos, para ser consultado desde la aplicación móvil.

A continuación se presentan fotografías de las vistas relacionadas con la visualización del consumo hídrico.



Figura 4.20: Vista consulta consumo hídrico.



Figura 4.21: Vista consumo hídrico.

Una vez realizadas las 8 iteraciones que dan cuenta del proyecto de memoria se logra obtener el primer prototipo funcional que cumple con el objetivo de esta memoria. No obstante, aún queda mucho trabajo por delante y mejoras que realizar, lo cual será detallado más adelante en la sección de trabajo futuro del capítulo de conclusiones.

A continuación se presentan fotografías de la estación funcionando en el campo

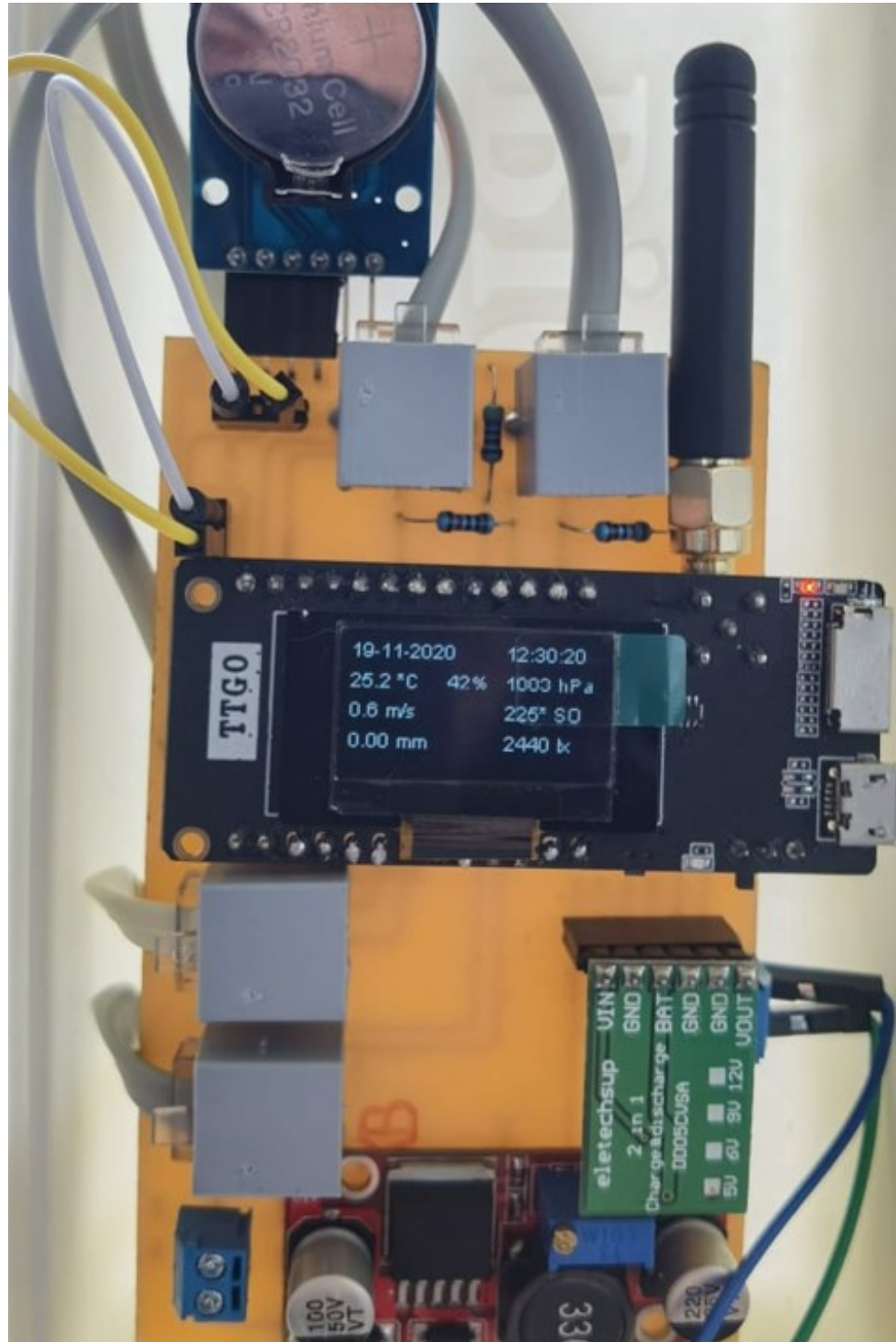


Figura 4.22: Circuito Impreso de la Placa.





Figura 4.23: Caja Contenedora Prototipo Uno.



Figura 4.24: Receptor RF.



Figura 4.25: Prototipo Funcional Uno.

### **4.3. Resumen del Capítulo**

En este capítulo, se detalló el proceso de construcción del sistema desde su hardware hasta su aplicación móvil y su conexión con el modelo matemático, el cual permitió obtener el consumo hídrico de un campo.

En el siguiente capítulo se presenta la evaluación realizada que validan la confección del sistema. Además se mostrarán las pruebas realizadas para demostrar el correcto funcionamiento.

## **5. Evaluación Experimental**

---

En el presente capítulo se describe la evaluación de la memoria propuesta. Para esto, se detalla la forma en que se aplica la metodología de evaluación implementada considerando cada una de sus etapas tanto para hardware como para el software. Finalmente, se dan a conocer los resultados obtenidos.

### **5.1. Fases de experimentación del Software**

A continuación, se detalla lo realizado en cada una de las fases de la experimentación informadas en el marco metodológico.

#### **5.1.1. Definición**

El objetivo del estudio es comprobar la usabilidad de la aplicación. Para validar la aplicación se hace uso de encuestas de usabilidad SUS (System Usability Scale) directamente al agricultor. El propósito es determinar si efectivamente es lo que él requiere y comprobar si la visualización de los datos es la adecuada. Además, se realiza la encuesta a funcionarios Biovision Agrícola quienes, del mismo modo, son usuarios encargados de validar la aplicación desde el punto de vista de especialistas en el ámbito de la agricultura.

#### **5.1.2. Diseño de la experimentación**

En esta etapa se definen los objetivos y características a evaluar junto con las pruebas de usabilidad. Los sujetos evaluados son agricultores de Río Claro e ingenieros agrónomos pertenecientes a Biovision Agrícola. Cabe mencionar que previamente a la utilización se realiza una pequeña capacitación.

### **Objetivo de evaluación**

Los puntos a evaluar con estas pruebas hacen referencia al cumplimiento del objetivo general. Para esto, con las respuestas obtenidas de los especialistas se pueden validar los resultados y con las respuestas de los agricultores se podrá determinar si este proyecto les ayuda realmente a mejorar la utilización del agua.

### **Características de la evaluación**

Las características a evaluar se pueden clasificar en dos aspectos:

- **Funcionalidad:** Con ello se determina si la aplicación entrega datos confiables y si cumple con los requerimientos para proporcionar funciones que logren satisfacer las necesidades de los agricultores. Dichas funcionalidades son validadas por el grupo de especialistas de Biovision Agrícola.
- **Usabilidad:** Se verifica si la aplicación es fácil de usar y resulta atractiva para el usuario. Esta característica es validada por los usuarios finales quienes son los agricultores de la Región del Maule.

### **Instrumentos de Medición**

Para realizar dichas mediciones se hace uso de las encuestas de usabilidad SUS a distintos tipos de agricultores, para determinar la efectividad de la aplicación. Por otro lado, se realiza una reunión con especialistas de Biovision Agrícola para determinar las funcionalidades de la aplicación.

### **5.1.3. Ejecución de la experimentación**

Para que la evaluación sea provechosa se define un protocolo que nos permite llevar con éxito dichas pruebas enfocadas en los agricultores.

#### **Protocolo de experimentación**

El protocolo efectuado para la evaluación se desglosa de la siguiente manera:

1. Presentación de la estación meteorológica y su método de funcionamiento.
2. Presentación de la aplicación móvil.
3. Explicación de los datos obtenidos y su forma de procesamiento.
4. Entrega de celular con la aplicación para que el usuario pueda interactuar.
5. Aplicación de la encuesta de usabilidad SUS.

#### 5.1.4. Sesión de Experimentación

Para la experimentación se realizan visitas en terreno a los campos de los agricultores, donde se cuenta con la participación de 8 agricultores de la comuna de Río Claro de un rango etario de 48-80 años.

#### 5.1.5. Análisis de Experimentación

En esta sección se detallan los resultados obtenidos luego de la realización de las pruebas de usabilidad aplicadas directamente a los agricultores en el campo. Para comenzar, se realiza el cálculo del puntaje según el algoritmo de las pruebas de usabilidad. Para ello, se suman los resultados promediados obtenidos de los cuestionarios, considerando lo siguiente: las preguntas impares (1,3,5,7 y 9) tomarán el valor asignado por el usuario, y se le restará 1. Para las preguntas pares (2,4,6,8,10), será de 5 menos el valor asignado por nuestros entrevistados. Una vez obtenido el número final, se multiplica por 2.5. Realizado este cálculo obtendremos un puntaje de máximo 100 puntos, donde entre más alto mejor usabilidad entrega la aplicación.

Para este caso, el puntaje promediado para cada pregunta fue de: 4,2,3,4,5,1,3,3,3 y 3 respectivamente, lo que llevado a la fórmula antes mencionada quedaría de la siguiente manera:  $((4 - 1) + (5 - 2) + (3 - 1) + (5 - 4) + (5 - 1) + (5 - 1) + (3 - 1) + (5 - 3) + (3 - 1) + (5 - 3)) \cdot 2,5 = 62,5$

Como se puede apreciar, el resultado obtenido, a pesar de estar sobre un 50, se sigue considerando como un valor bajo. Sin embargo, este valor no refleja las razones específicas de dicha puntuación. Para ello, se analizan con más detalle las preguntas realizadas donde se puede comprender de mejor forma este resultado.

Si dividimos las preguntas en dos secciones diferentes; funcionalidad y muestreo, podemos obtener una mejor conclusión sobre todo el trabajo realizado. Para la sección de funcionalidad se consideran las preguntas 1,5,6 y 9 y para las preguntas de muestreo se evaluarán las preguntas 2,3,4,7,8,10. Aplicando la fórmula mencionada anteriormente se encuentra el siguiente resultado:

$$((4 - 1) + (5 - 1) + (5 - 1) + (3 - 1)) \cdot 2,5 = 32,5 \text{ de } 40$$

$$((5 - 2) + (3 - 1) + (5 - 4) + (3 - 1) + (5 - 3) + (5 - 3)) \cdot 2,5 = 30 \text{ de } 60$$

Respecto a la sección de funcionalidad se obtuvo un valor de 32.5 de un total de 40

puntos, lo que llevado a una escala de 0-100 es equivalente a un 81.25, resultado que se considera bastante bueno. Cabe destacar que este valor está relacionado a las preguntas sobre la información que se muestra en la aplicación y sobre la confianza que el usuario tendrá al momento de visualizar estos datos.

Por otro lado, en la sección de muestreo se obtuvo un valor de 30 de un total de 60 puntos, este puntaje llevado a una escala de 0-100 sería equivalente a un 50. Dicho valor hace referencia a las preguntas relacionadas sobre la forma en que se muestra la información mediante la aplicación móvil.

Luego de analizar las respuestas obtenidas y los puntajes relacionados, a modo general, se concluye lo siguiente:

Si bien, para el agricultor es provechoso conocer datos meteorológicos de su campo en tiempo real ya que le puedan ayudar a la toma de decisiones (por ejemplo: saber cuánta agua aplicar, o en que momentos poder aplicar cierto producto), el pequeño agricultor no está dispuesto a intentar utilizar una aplicación móvil debido a la idea de que utilizará algo demasiado complejo que entorpecerá su rutina habitual. Esto se debe probablemente a que, en general, son personas mayores con bajo nivel de escolaridad. No obstante, hay otros agricultores más jóvenes que estarían dispuestos a utilizar este sistema pero con la ayuda de un tercero.

De esta manera se plantean dos posibles soluciones:

Opción 1: Para el agricultor es conveniente poder llamar a una línea telefónica en el momento en que quisiera saber el estado de su campo y consultar una determinada acción. De esta forma, si él quisiera regar su campo, podría llamar y consultar cuánta es el agua que debe aplicar a su cultivo desde la última vez que tuvo que regar. Otro ejemplo es que para poder aplicar cualquier tipo de producto fitosanitario a las plantas, la velocidad del viento no puede estar a más de 3 m/s. Por lo cual, antes de aplicar un producto podría llamar y consultar si según las mediciones de la última hora, u otro rango establecido por el usuario, es factible o no la aplicación de algún producto fitosanitario. De esta forma, la estación se transforma en un elemento clave ya que cumple las necesidades del agricultor de una rápida y simple. Sin embargo, la forma en que se entregan los datos es la que se debe mejorar.

Opción 2: Otra alternativa para la entrega de datos al agricultor según la información recopilada en las encuestas, es programar una API, que todos los días a una hora determinada haga envío de los datos por WhatsApp de cada campo a cada agricultor y, además, le permita a él enviar una palabra clave al mismo número y que éste responda con las



últimas mediciones realizadas por la estación en el campo. Esta alternativa se considera como la más factible y conveniente, por lo cual se valora positivamente este tipo de recomendaciones que nos ayudan a mejorar la propuesta.

## 5.2. Pruebas de caja negra hardware

Para la validación del funcionamiento del hardware se realizan pruebas de caja negra a los sensores en general, esto permite verificar si el valor enviado desde la estación emisora es el mismo dato que se recibe en la estación. Para ello, se realizan pruebas unitarias sensor por sensor, enviando datos de forma constate en tiempo real, los cuales son recepcionados de forma correcta en la base de datos. Adicionalmente, esta prueba nos sirven para probar el protocolo de comunicación mediante radio frecuencia en la estación.

Las pruebas son realizadas en 3 tipos de ambientes diferentes:

- Ambiente interior controlado: se realiza envío de datos de la estación meteorológica al receptor dentro de una misma casa. En estas pruebas, cada cierto tiempo se detectan anomalías con respecto a la recepción de los datos, ya que en algunos casos llegan paquetes no identificados debido a la alta interferencia dentro de la ciudad y de una casa, en donde de manera simultánea se hace uso de redes WiFi, Bluetooth, GPRS, entre otras.
- Ambiente exterior ciudad: en este caso se obtienen los mismos resultados que en un ambiente interior pero con mayor frecuencia de interferencia, esto debido a que las pruebas son realizadas en pleno centro de la ciudad donde se encuentran rodeados por edificios, autos y mucha afluencia de personas.
- Pruebas en campo: finalmente, estas pruebas si son satisfactorias, dado que al realizar las pruebas al aire libre y sin interferencias, se logra obtener la recepción de los datos sin mayores inconvenientes.

Adicionalmente, se realizan pruebas de caja negra para la validación del modelo matemático y las funcionalidades del sistema.

A continuación, se presenta un ejemplo de caja negra. El resto de la totalidad de las pruebas de caja negra se encuentran en la sección de anexos de este documento.

<b>Código</b>	P4	
<b>Historia/s de usuario</b>	US004	
<b>Descripción</b>	Calculo de Et0 diario mediante el modelo matemático	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Luminosidad</li> <li>■ Humedad</li> <li>■ Temperatura</li> <li>■ Velocidad del viento</li> <li>■ Dirección del viento</li> <li>■ Lluvia caída</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera un valor entre 0 y 10</li> </ul>	<ul style="list-style-type: none"> <li>■ 1.98</li> </ul>
<b>Observaciones</b>	Se obtiene un valor esperado dentro del rango establecido	

Tabla 5.1: Prueba caja negra #4 - Calculo de Et0

### 5.3. Resumen del Capítulo

Al finalizar este capítulo, se pudo identificar que la forma elegida para mostrar la información captada por la estación meteorológica mediante la aplicación móvil no es la más adecuada considerando el grupo de pequeños agricultores, ya que para ellos se torna muy complejo el hecho de interactuar frecuentemente con un dispositivo móvil. Sin embargo, los agricultores afirman que si bien ellos no usarían la aplicación, a sus hijos, que pertenecen a una generación más relacionada con la tecnología, les resulta muy simple e intuitivo la forma de interactuar con la aplicación. Por lo tanto, se podrían considerar dos grupos de personas: el grupo de adultos quienes por facilidad no interactuarían con la aplicación y, por esta razón, prefieren obtener la información mediante un mensaje por WhatsApp y el grupo más joven, que podría hacer un perfecto uso de la aplicación.

## 6. Conclusiones

---

En este capítulo se presentan las conclusiones obtenidas durante el proceso de desarrollo del sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales del Maule. A continuación, se expone el cumplimiento de los objetivos y el análisis de los resultados del funcionamiento del sistema construido. Además, se presenta el trabajo futuro que se desprende de este proyecto.

### 6.1. Sobre los Resultados Alcanzados

En este trabajo se aborda una problemática clave para el sector de la agricultura "Gestión Hídrica", en esta sección se presentan los resultados alcanzados para dar solución a la problemática.

- Se logra construir un prototipo de estación meteorológica de bajo costo que es capaz de medir datos tales como: humedad relativa, temperatura, presión atmosférica, luminosidad, cantidad de lluvia caída, velocidad y dirección del viento, capaz de enviar datos mediante radio frecuencia a un receptor, el cual envía los datos recibidos a la nube.
- Se diseña un circuito para realizar la calibración del sensor de luminosidad en función de la radiación. Sin embargo, dicha calibración solo se logra implementar a nivel de concepto, ya que no se alcanza a conseguir una cantidad de pruebas significativas para realizar la curva de calibración. Los datos utilizados son solo a nivel conceptual.
- Se logra modificar el modelo matemático de Penman Monteith programado en R por el Doctor Fernando Fuentes, para que sea capaz de interactuar directamente con la base de datos en tiempo de real alojada en Firebase.

- Se logra desarrollar una aplicación móvil capaz de mostrar todos los datos obtenidos tanto de la estación meteorológica como los del modelo matemático. Esto permite tomar decisiones productivas en base a datos objetivos, aumentando la eficiencia del riego, mejorando la calidad y los rendimientos de la producción de pequeños agricultores.

A modo de conclusión general se puede decir que sumados los cuatro puntos antes mencionados, en conjunto forman el sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales del Maule.

## **6.2. Sobre el Cumplimiento de los Objetivos**

En esta sección se analiza cada uno de los objetivos presentados al comienzo del documento para dar a conocer la forma en que cada uno de ellos fue cumplido. Se hace un recuento de los objetivos específicos que dan cuenta a del objetivo general final.

### **6.2.1. Objetivos específicos**

- Mejorar la eficiencia de la utilización del agua, a través de un modelo matemático, la información de múltiples sensores y la utilización de Cloud Computing.

Gracias al sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales del Maule que se conforma por una estación meteorológica de bajo costo, una aplicación móvil y un modelo matemático capaz de estimar la evapotranspiración en base a los datos recopilados de la estación meteorológica, el usuario puede saber cuánta es el agua que su cultivo pierde día a día, por lo tanto, se puede estimar la cantidad de agua a aplicar en el cultivo. Además, gracias al factor de cultivo, si la estación se encuentra ubicada en una zona de referencia la cual se basa en que la estación se encuentre en una superficie a la redonda cubierta por pasto a unos 10 centímetros de altura y la estación con su sensor de luminosidad a unos 2 metros aproximadamente de la superficie, estos resultados pueden ser aplicados en cualquier cultivo de la región del Maule.

- Disminuir los costos de una estación meteorológica para que pueda ser accesible por cualquier tipo de productor agrícola.

Gracias al uso de sensores de bajo costo y las modificaciones necesarias para poder reemplazar los sensores de alto costo como lo es un piranómetro, se logra generar una estación de ultra bajo costo que en su etapa de construcción no supera los 300 dólares, comparado con las estaciones de hoy en día que para el caso de una estación WatchDog (más software), el precio es de 3.441.842 pesos, mientras que una estación marca Metos modelo iMetos ag 280 tiene un valor de 2.720.474 pesos (no incluye software). Adicionalmente, ya que la estación fue pensada para pequeños y medianos agricultores, eventualmente se puede conseguir algún tipo de financiamiento público como lo puede ser la entidad INDAP permitiendo así que cualquier agricultor pueda acceder de forma sencilla a la obtención del sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales del Maule.

- Disminuir la latencia de los protocolos de comunicación convencionales, a través de la utilización de protocolos de comunicación eficientes como MQTT.

Mediante el uso de tecnología de radio frecuencia y al protocolo de comunicación MQTT se logra realizar una comunicación efectiva que hace posible la instalación de la estación meteorológica en el campo, sin tener que preocuparse por tener una conexión a internet, lo cual en medio del campo no es factible, ya que sin el envío de datos por radio frecuencia esto no sería posible.

Conforme a las pruebas realizadas se logra obtener un radio de comunicación de 3 kilómetros a la redonda lo que forma una área suficiente al estar diseñado para pequeños y medianos agricultores, ya que INDAP está definido como un agricultor de no más de 12 hectáreas. Este tamaño permite estar dentro del alcance con radio frecuencia. Además, una vez que el receptor obtiene los datos de la estación hace uso del protocolo MQTT para subir los datos directamente a la base de datos de Firebase. Este protocolo permite tener una comunicación eficiente, confiable y en tiempo real.

### **6.2.2. Objetivo general**

- Determinar el consumo hídrico de frutales y vives, a través de una aplicación móvil integrando técnicas de Cloud Computing y protocolos de comunicación.

Luego de cumplir con cada uno de los objetivos específicos, en su conjunto, se

tiene la satisfacción de cumplir con el objetivo general, ya que gracias al hecho de tener una estación meteorológica de bajo costo accesible para cualquier agricultor, un modelo matemático capaz de estimar la evapotranspiración (permitiendo estimar la cantidad de agua requerida por un cultivo) y los protocolos de comunicación de radio frecuencia y MQTT permiten poder obtener todos estos datos en tiempo real en la nube, el agricultor, puede utilizar estos datos que se le entregan a través de una aplicación móvil para ser utilizados en la toma de decisiones productivas, con el fin de mejorar la eficiencia, rendimiento, calidad de sus cultivos y a su vez, determinar el consumo hídrico.

### 6.3. Trabajo Futuro

Si bien es cierto que en este trabajo se logra la obtención de un prototipo, esto es solo una parte de un proyecto mayor que involucra la colaboración de otras investigaciones. En la siguiente sección se detallan los trabajos futuros que derivan de este informe.

- Una investigación está destinada al desarrollo de una plataforma que procesa imágenes satélites, la cual requiere de estaciones meteorológicas que perciben las condiciones climáticas de ciertas zonas. Por convención, una estación cubre un área de 30 kilómetros en terrenos planos, sin embargo, esto no pasa en Chile debido a su geografía montañosa. Para tener un mejor modelo con datos reales asociados directamente a un campo se requiere tener una estación meteorológica dentro del mismo campo. Lamentablemente, hoy en día esto no es posible para un pequeño o mediano agricultor debido al alto costo que implica su instalación. Como agricultor se desea optimizar y mejorar el consumo hídrico del cultivo. Se considera este trabajo como una alternativa para la construcción de una estación de bajo costo, si resulta factible la idea de instalar una estación meteorológica en cualquier campo. Por ende, se debe integrar el almacenamiento de estos datos al estándar de ADCOM para que los datos sean compatibles con el de la plataforma satelital y se pueda obtener datos tanto del satélite, como de estaciones de campo de alto y bajo costo.
- Hacer compatible con el trabajo de otros investigadores, quienes deben programar un flujo que va a permitir al agricultor regar de forma eficiente y autónoma. Esto se conoce como programación del riego. Para ello, se requiere de la evapotranspiración de referencia que se puede obtener de este trabajo y el coeficiente de cultivo que se

desprende de la plataforma de otro colega, por lo cual, si se realiza una integración apropiada de ambos sistemas se pueden utilizar estos datos como input para realizar la programación del riego, la cual permite identificar tiempo y frecuencia de riego. Lo anterior resulta muy importante, ya que hoy en día los agricultores riegan al "ojo", siendo este proceso muy ineficiente en lo que respecta a la cantidad de agua aplicada. Si se logra hacer una programación basada en datos objetivos se puede incrementar la eficiencia del uso del agua y reducir la cantidad de agua utilizada sobre un cultivo determinado. Por ejemplo, si se tiene un terreno de 10 hectáreas y se tienen 10.000 litros de agua disponible y se usan todos para un pequeño cultivo, al tener una programación del riego en vez de usar esos 10.000 litros para un pequeño cultivo, se usarían solo 5.000, por ende, el agua sobrante permite incrementar la superficie de cultivo pudiendo regar un área mucho más grande, generando así mayores ingresos en el mismo campo.

#### **6.4. Resumen del Capítulo**

Para concluir con este capítulo y a su vez, con este trabajo, se presentan los resultados alcanzados de los cuáles se está muy conforme, ya que hoy en día lo que se encuentra desarrollado en el ámbito de la agricultura es tan escuálido que cualquier aporte resulta fundamental para lograr avances en el área. Además, se espera que el desarrollo de futuras investigaciones busquen seguir mejorando el uso de los recursos hídricos, y los beneficios que trae consigo, tanto económicos como medioambientales.

Con respecto al cumplimiento de los objetivos se tiene la satisfacción de cumplir con lo esperado y la sensación que queda al saber que el trabajo que se realiza puede llegar a tener un gran impacto resulta muy comfortable.

Finalmente, si bien el trabajo futuro que queda por delante es bastante amplio se tiene todo lo necesario para poder continuar sin ninguna limitante y cumplir con la meta final de tener un gran sistema integrado de gestión hídrica orientado a pequeños y medianos productores de cultivos y frutales del Maule y ¿Por qué no pensar también en ampliar este rango a todo el país y en un futuro no muy lejano a nivel mundial?

# Bibliografía

- [1] Agenda País. Chile se está secando: descenso de aguas superficiales causa preocupación a nivel nacional. <https://www.elmostrador.cl/agenda-pais/2019/08/27/chile-se-esta-secando-descenso-de-aguas-superficiales-causa-preocupacion-a-nivel-nacional/>, 2019.
- [2] Ambient Weathe. Ambient Weather. <https://www.ambientweather.com/amwevwamweac.html>.
- [3] Android Studio. Android Studio. <https://developer.android.com/>.
- [4] Argent Data System. Weather Sensor Assembly p/n 80422. *Imported by Argent Data Systems*, page 2.
- [5] Bosh. Datasheet BME280. <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>.
- [6] EasyEDA. EasyEDA. <https://easyeda.com/es>.
- [7] Hanan Elazhary. Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions. *Journal of Network and Computer Applications*, 128, 2019.
- [8] Electronica plug and play. Datasheet BH1750. <https://www.electronicaplugandplay.com/sensores-y-transductores/product/545-modulo-sensor-de-luminosidad-bh1750/>.
- [9] EMAs Chile. Agromet. <https://www.agromet.cl/>.
- [10] Fabio Devin. Sistema de Escalas de Usabilidad: ¿qué es y para qué sirve? <https://uxpanol.com/teoria/sistema-de-escalas-de-usabilidad-que-es-y-para-que-sirve/>.



- [11] Consuelo Fernandez. *Escenarios hídricos 2030. Transición Hídrica: El Futuro del Agua en Chile*. Journal of Chemical Information and Modeling, 2019.
- [12] Firebase. Firebase. <https://firebase.google.com/>.
- [13] Flutter. Flutter. <https://flutter.dev/docs>.
- [14] Fernando Fuentes-Peñailillo, Samuel Ortega-Farías, César Acevedo-Opazo, and David Fonseca-Luengo. Implementation of a Two-Source Model for Estimating the Spatial Variability of Olive Evapotranspiration Using Satellite Images and Ground-Based Climate Data. *Water*, 10(3):339, Mar 2018.
- [15] Emily Gertz and Patrick Di Justo. *Environmental Monitoring with Arduino*. O'Reilly Media, 2012.
- [16] GitKraken. GitKraken. <https://www.gitkraken.com/>.
- [17] Google Cloud Plataform. Google Cloud Plataform. <https://cloud.google.com/>.
- [18] V. G. Guimaraes, R. M. de Moraes, K. Obraczka, and A. Bauchspiess. A novel iot protocol architecture: Efficiency through data and functionality sharing across layers. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, 2019.
- [19] IDE Arduino. IDE Arduino. <https://www.arduino.cc/>.
- [20] LilyGo. Ttgo ESP32. <http://www.lilygo.cn/>.
- [21] Luis Llamas. Datasheet rtc. <https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/>.
- [22] My AcuRite. My AcuRite. <https://www.acurite.com/learn/my-acurite/website>.
- [23] R. R. <https://www.r-project.org/>.
- [24] Felipe Ramírez. Escasez hídrica en Chile, ¿cómo la enfrentamos? <https://www.uchile.cl/noticias/160634/escasez-hidrica-en-chile-como-la-enfrentamos>, Consultado el 5 de mayo de 2020 2020.

- [25] Roxana Alvarado. El 80% de los problemas de agua en Chile se resolverían con una buena gestión. <http://www.forestal.uchile.cl/noticias/146035/el-80-de-los-problemas-de-agua-se-resolverian-con-una-buena-gestion>, 2018.
- [26] Neldin Noel Pérez Reyes Sintya Milena Meléndez Valladarez, Maria Elizabeth Gaitan. Metodología Ágil De Desarrollo De Software Programacion Extrema. *Universidad Católica de Colombia*, page 146, 2016.
- [27] Trello. Trello. <https://trello.com/es>.
- [28] U.S. Department of Health y Human Services. System Usability Scale (SUS). <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>.
- [29] WeatherLink App. WeatherLink App. <https://www.davisinstruments.com/weatherlink/e>.
- [30] Don Wilcher. *Arduino Electronics Blueprints*. PACKT, 2015.
- [31] José Zalaquett. *Big Chile*. Una Pasión Predominante, 2019.

# **ANEXOS**

# A. Versiones de las placas de circuito impreso

---

A continuación, se presentan las diferentes versiones de los diagramas de diseño impreso para la estación meteorológica. En total se tienen 7 versiones más una versión que corresponde al receptor.

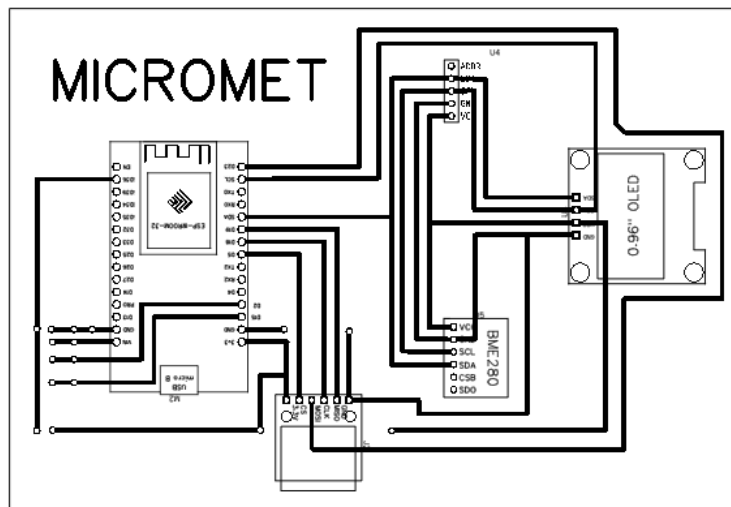


Figura A.1: Diseño de circuito impreso V1.0

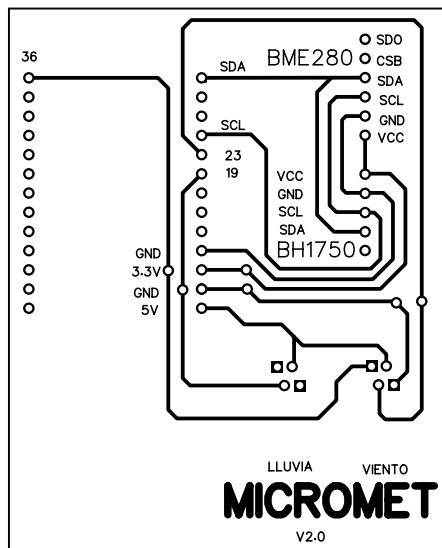


Figura A.2: Diseño de circuito impreso V2.0

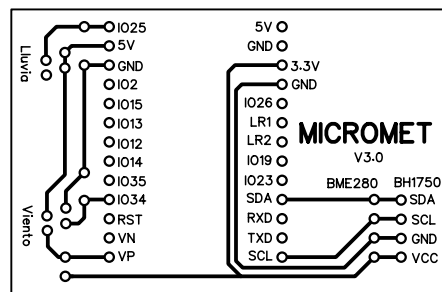


Figura A.3: Diseño de circuito impreso V3.0

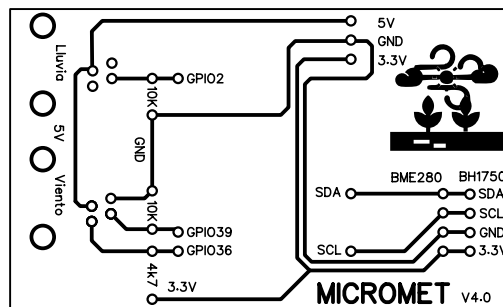


Figura A.4: Diseño de circuito impreso V4.0

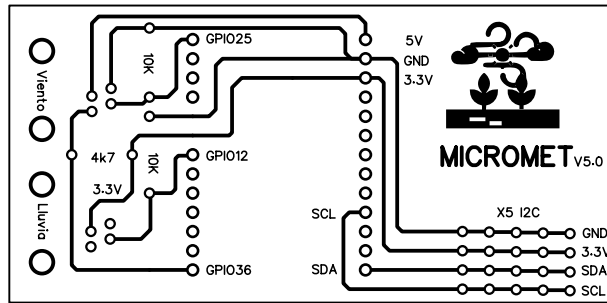


Figura A.5: Diseño de circuito impreso V5.0

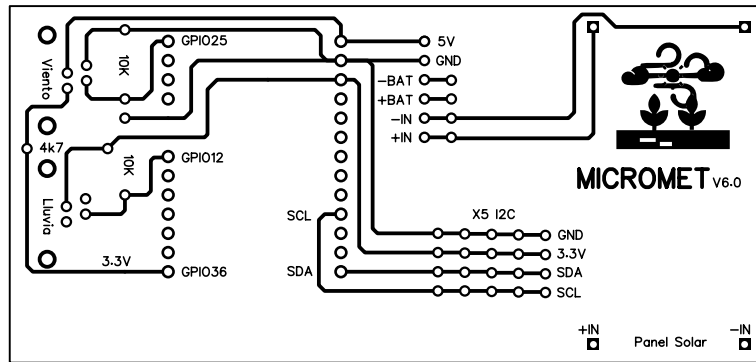


Figura A.6: Diseño de circuito impreso V6.0

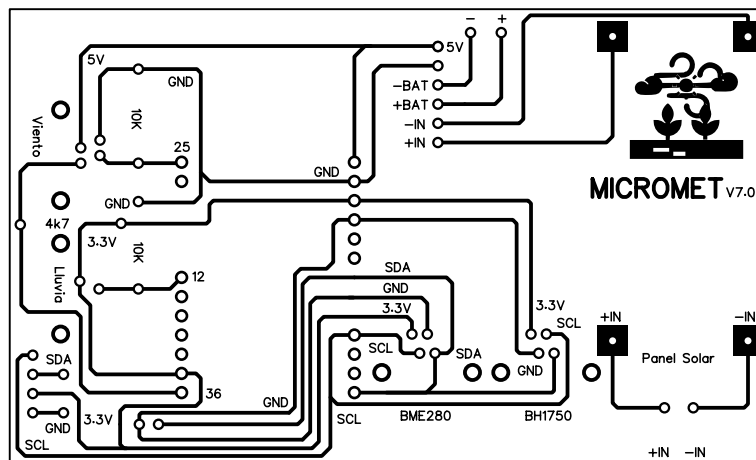


Figura A.7: Diseño de circuito impreso V7.0



## B. Prototipos de las estaciones construidas

---

En este anexo se presentan algunos de los prototipos de las estaciones construidas mostrando el paso a paso de versiones que dan cuenta para llegar al prototipo actual.

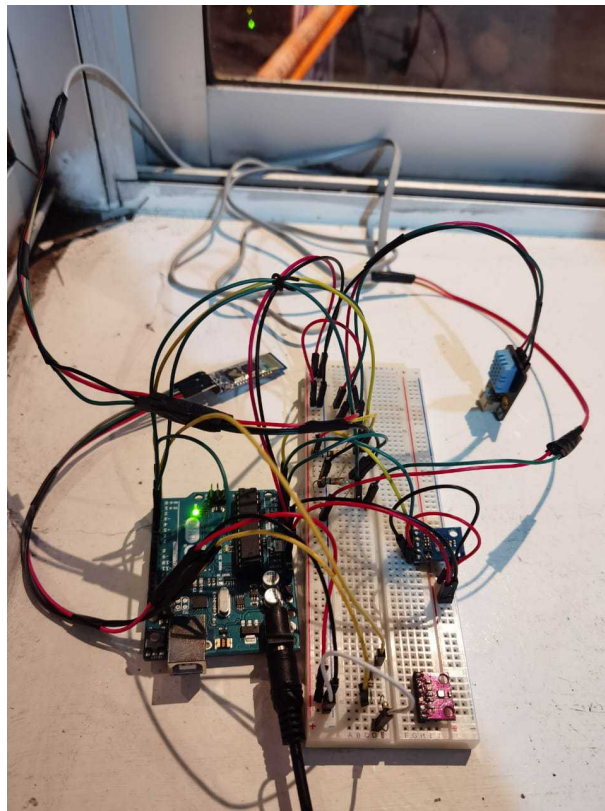


Figura B.1: Primer prototipo de estación meteorológica



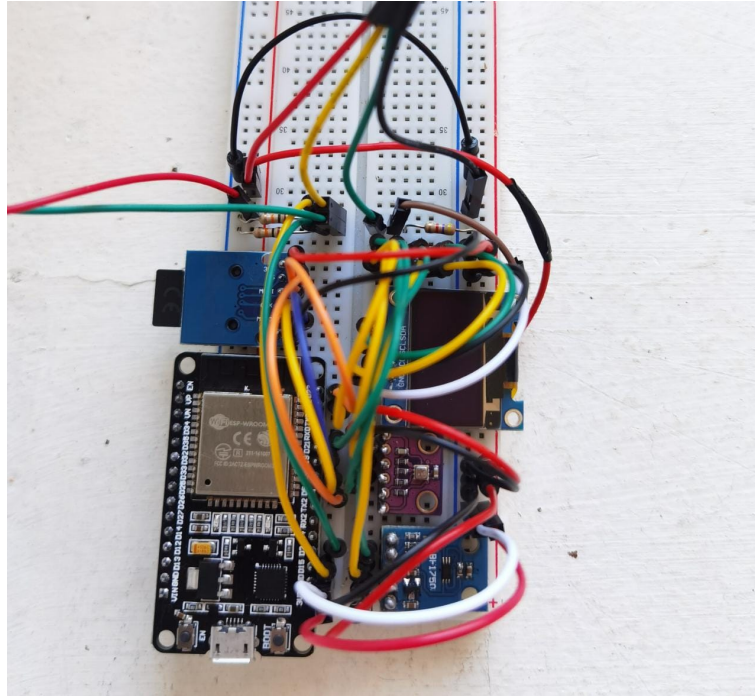


Figura B.2: Segundo prototipo de estación meteorológica

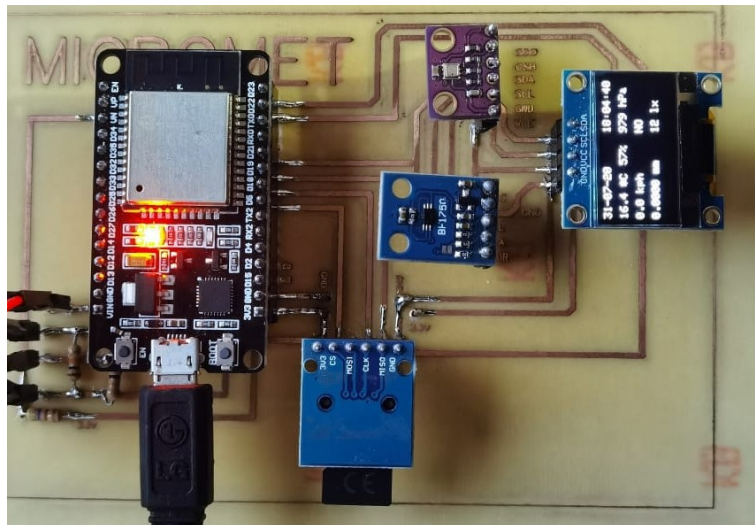


Figura B.3: Tercer prototipo de estación meteorológica



Figura B.4: Primer armado sensores de viento y lluvia

## C. Conversión analógica digital 5V a 3.3V

---

En este anexo se muestra la forma en la cual se calibró el sensor de dirección del viento para poder operar a un voltaje de 3.3V y no 5V debido a que nuestro microcontrolador no es capaz de transformar valores analógicos a digitales a un voltaje superior de 3.3V. Para ello, fue necesario ir girando ángulo por ángulo el sensor de dirección del viento y a su vez ir analizando el valor analógico de transformación y su voltaje entregado para poder realizar la transformación correspondiente.



Figura C.1: Calibración de la dirección del viento a 3.3V

```
float Calculate_WindDirection() {
  int val = analogRead(36); // VP
  //Serial.println(val);

  if (val < 700){
    windDir = "E";
    windDirAng = 90;
  }
  else if(val<1400){
    windDir = "SE";
    windDirAng = 135;
  }
  else if(val<2000){
    windDir = "S";
    windDirAng = 180;
  }
  else if(val<2600){
    windDir = "NE";
    windDirAng = 45;
  }
  else if(val<3200){
    windDir = "SO";
    windDirAng = 225;
  }
  else if(val<3600){
    windDir = "N";
    windDirAng = 0;
  }
  else if(val<3700){
    windDir = "NO";
    windDirAng = 315;
  }
  else{
    windDir = "O";
    windDirAng = 270;
  }
  return windDirAng;
}
```

Figura C.2: Trasformación respuesta analógica a dirección del viento

## D. Pruebas de campo de radio frecuencia

---

En este anexo se muestran algunas de las pruebas realizadas para la medición de la distancia de radio frecuencia. para ello se realizaron pruebas en dos campos. Un campo de 6 hectáreas ubicado en Cumpeo y otro campo de 400 hectáreas ubicado en San Clemente. Adicionalmente se realizó la prueba en el ciudad de Talca bajo efectos de interferencia.



Figura D.1: Campo San Clemente



Figura D.2: Prueba de distancia a 3 kilómetros de radio frecuencia en campo



Figura D.3: Prueba de distancia a 500 metros lateral de radio frecuencia en ciudad



Figura D.4: Prueba de distancia a 500 metros rectos de radio frecuencia en ciudad



## **E. Ensamblado de estación meteorológica final**

---

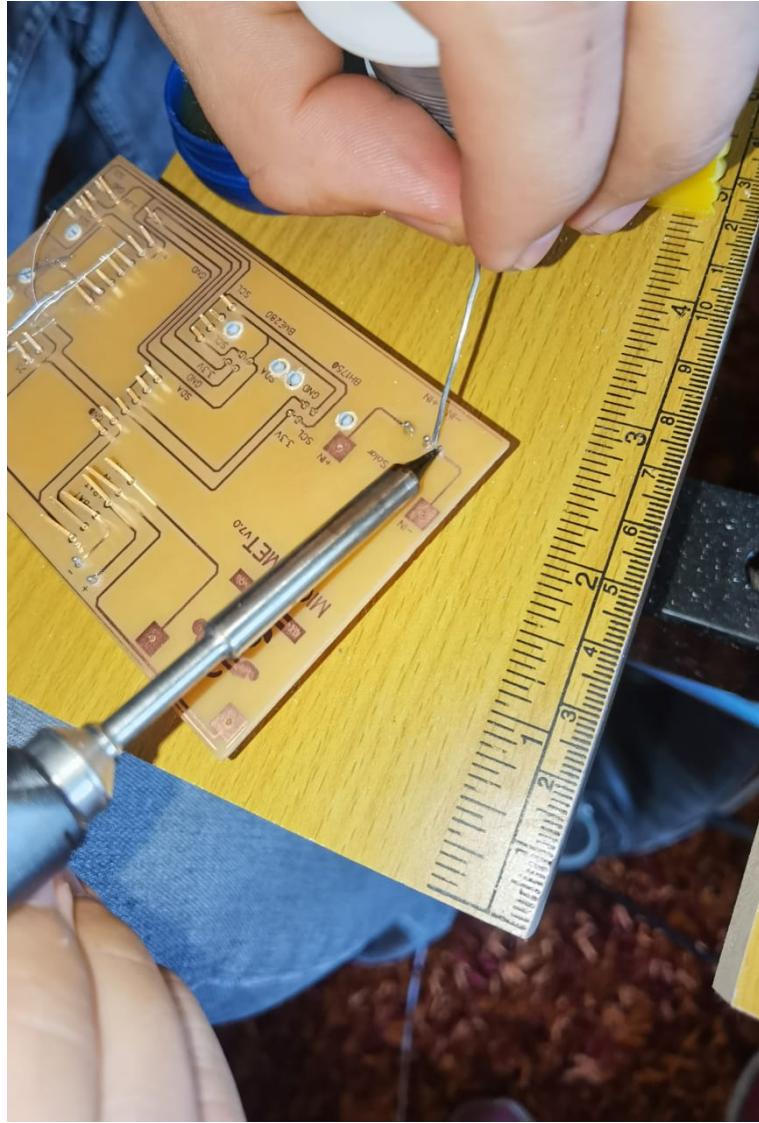


Figura E.1: Construcción placa estación meteorológica



Figura E.2: Armado cableado estación meteorológica

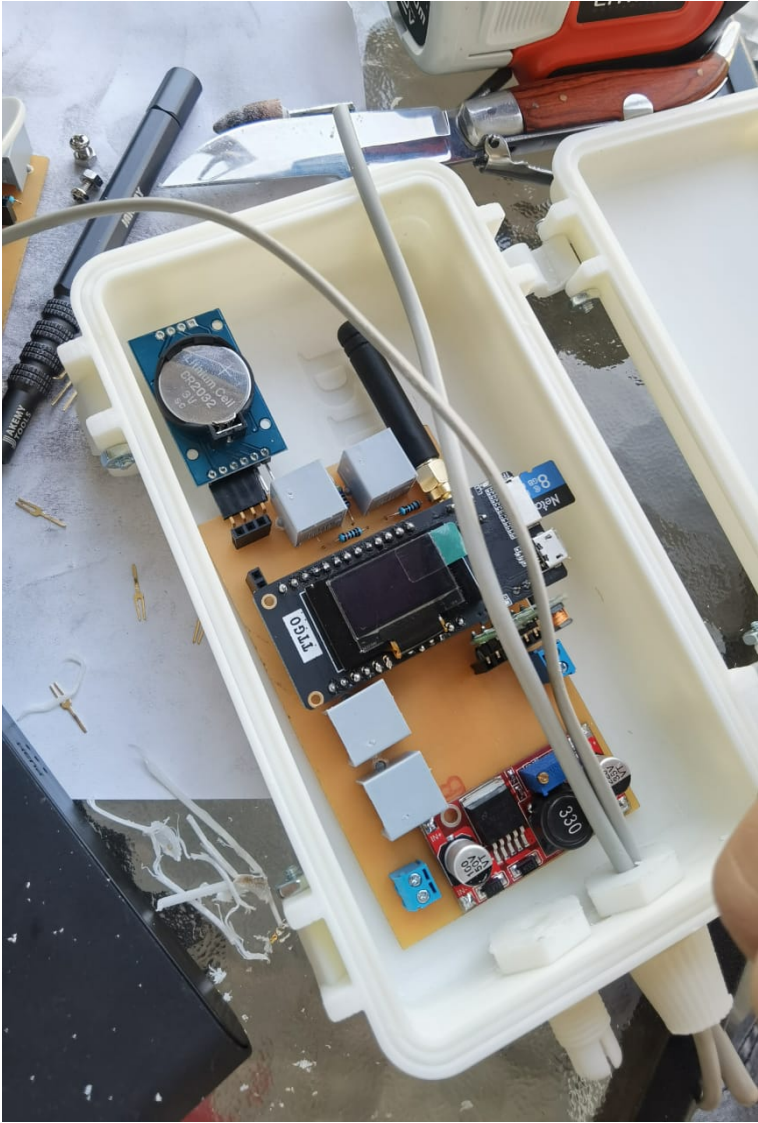


Figura E.3: Armado empaquetado estación meteorológica

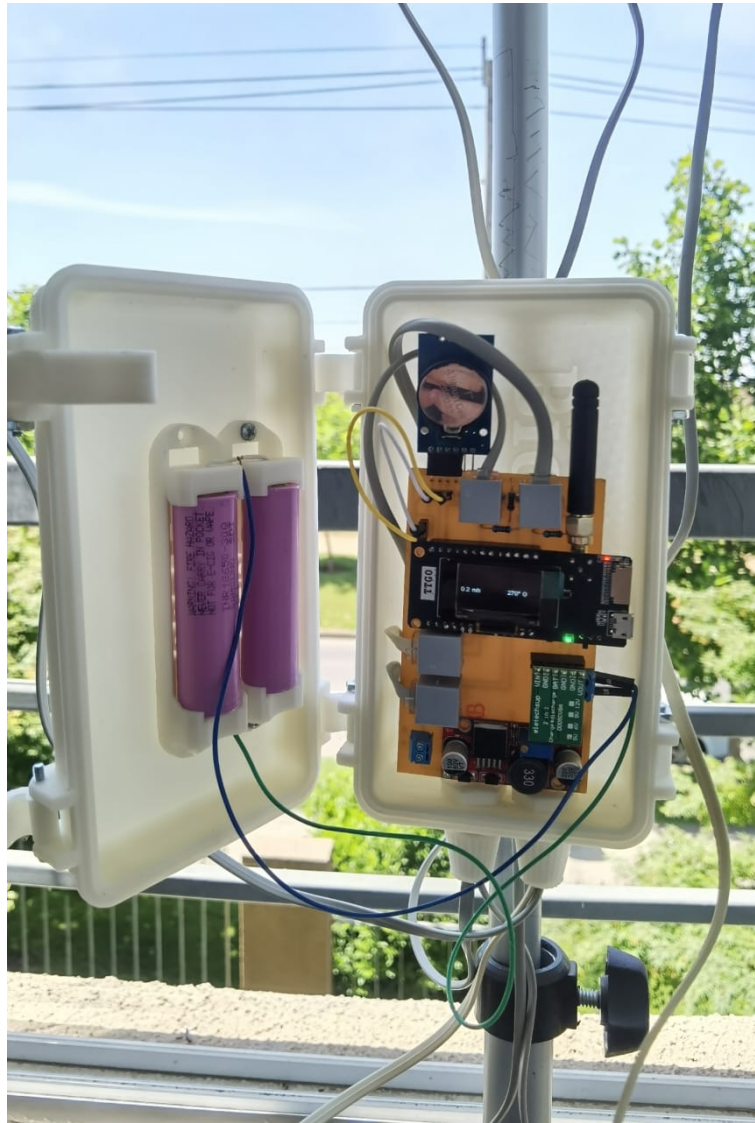


Figura E.4: Ensamblado caja estación meteorológica

## **F. Pruebas de caja negra**

---

En esta sección se dan a conocer las pruebas de caja negra realizadas en este proyecto asociados a cada historia de usuario con la finalidad de probar cada módulo de la aplicación móvil y la estación meteorológica.

<b>Código</b>	P1	
<b>Historia/s de usuario</b>	US001	
<b>Descripción</b>	Lectura de datos desde la estación meteorológica	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Sensor de luminosidad BH1750</li> <li>■ Sensor de humedad BME280</li> <li>■ Sensor de temperatura BME280</li> <li>■ Sensor de presión atmosférica BME280</li> <li>■ Sensor de Velocidad del viento anemómetro</li> <li>■ Sensor de dirección del viento veleta</li> <li>■ Sensor de lluvia caída pluviómetro</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera que cada sensor entregue un valor valido para cada caso. dichos valores son especificados en el apartado de hardware de esta memoria.</li> </ul>	<ul style="list-style-type: none"> <li>■ Cada sensor entregó un valor dentro de sus rangos establecidos.</li> </ul>
<b>Observaciones</b>	Se obtiene un valor esperado dentro del rango establecido	

Tabla F.1: Prueba caja negra #1 - Lectura de datos meteorológicos

<b>Código</b>	P2	
<b>Historia/s de usuario</b>	US002	
<b>Descripción</b>	Visualización de datos en la APP	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Luminosidad</li> <li>■ Humedad</li> <li>■ Temperatura</li> <li>■ Presión atmosférica</li> <li>■ Velocidad del viento</li> <li>■ Dirección del viento</li> <li>■ Lluvia caída</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera obtener el mismo valor registrado en la base de datos para cada sensor en la visualización desde la aplicación móvil.</li> </ul>	<ul style="list-style-type: none"> <li>■ Se logran visualizar los mismo datos registrado en la base de datos en la aplicación móvil.</li> </ul>
<b>Observaciones</b>	El valor registrado en la aplicación es el mismo que en la base de datos	

Tabla F.2: Prueba caja negra #2 - Visualización de datos en la aplicación móvil



<b>Código</b>	P3	
<b>Historia/s de usuario</b>	US003	
<b>Descripción</b>	Visualización de datos históricos en la APP	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Luminosidad</li> <li>■ Humedad</li> <li>■ Temperatura</li> <li>■ Presión atmosférica</li> <li>■ Velocidad del viento</li> <li>■ Lluvia caída</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera obtener la lista de valores registrados en la base de datos para cada sensor en la visualización desde la aplicación móvil del último día.</li> </ul>	<ul style="list-style-type: none"> <li>■ Se logran visualizar los datos registrado en la base de datos en la aplicación móvil correspondientes al día actual.</li> </ul>
<b>Observaciones</b>	Los valores registrado en la aplicación son los mismos que en la base de datos	

Tabla F.3: Prueba caja negra #3 - Visualización históricas de datos en la aplicación móvil

<b>Código</b>	P4	
<b>Historia/s de usuario</b>	US004	
<b>Descripción</b>	Calculo de Et0 diario mediante el modelo matemático	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Luminosidad</li> <li>■ Humedad</li> <li>■ Temperatura</li> <li>■ Velocidad del viento</li> <li>■ Dirección del viento</li> <li>■ Lluvia caída</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera un valor entre 0 y 10</li> </ul>	<ul style="list-style-type: none"> <li>■ 1.98</li> </ul>
<b>Observaciones</b>	Se obtiene un valor esperado dentro del rango establecido	

Tabla F.4: Prueba caja negra #4 - Calculo de Et0

<b>Código</b>	P5	
<b>Historia/s de usuario</b>	US005	
<b>Descripción</b>	Visualización de Et0 diario en la APP	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Luminosidad</li> <li>■ Humedad</li> <li>■ Temperatura</li> <li>■ Velocidad del viento</li> <li>■ Dirección del viento</li> <li>■ Lluvia caída</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera un valor entre 0 y 10 visualizado en la APP</li> </ul>	<ul style="list-style-type: none"> <li>■ 1.98</li> </ul>
<b>Observaciones</b>	Se obtiene un valor esperado dentro del rango establecido y se muestra de forma correcta en la app	

Tabla F.5: Prueba caja negra #5 - Visualización de Et0

<b>Código</b>	P6	
<b>Historia/s de usuario</b>	US006 y US007	
<b>Descripción</b>	Adición de estaciones	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Nombre de la estación</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera que la base de datos reconozca la estación asociada en el hardware y genere el documento para cada una.</li> </ul>	<ul style="list-style-type: none"> <li>■ El hardware escribe correctamente las estaciones con su respectiva estructura en la base de datos,</li> </ul>
<b>Observaciones</b>	Se genera la estructura de árbol en la base de datos.	

Tabla F.6: Prueba caja negra #6 - Soporte múltiples de estaciones

<b>Código</b>	P7	
<b>Historia/s de usuario</b>	US008	
<b>Descripción</b>	Visualización de estaciones en la APP	
<b>Entrada</b>	<b>Salida esperada</b>	<b>Salida obtenida</b>
<ul style="list-style-type: none"> <li>■ Estación Cumpeo</li> <li>■ Estación Talca</li> <li>■ Estación Condorito</li> </ul>	<ul style="list-style-type: none"> <li>■ Se espera poder visualizar todas las estaciones de un agricultor en una misma cuenta de la aplicación</li> </ul>	<ul style="list-style-type: none"> <li>■ Estación Cumpeo</li> <li>■ Estación Talca</li> <li>■ Estación Condorito</li> </ul>
<b>Observaciones</b>	Se logran visualizar todas las estaciones	

Tabla F.7: Prueba caja negra #7 - Visualización de estaciones