



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

SISTEMA DE VERMICOMPOSTAJE CON INTERNET DE LAS COSAS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

MATÍAS GABRIEL ZAMORA QUITRAL

PROFESOR GUÍA:
MATÍAS MATTAMALA ARAVENA

MIEMBROS DE LA COMISIÓN:
SANDRA CÉSPEDES UMAÑA
CÉSAR AZURDIA MEZA

Este trabajo ha sido parcialmente financiado por Beauchef Proyecta

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERÍA CIVIL ELÉCTRICA
POR: MATÍAS GABRIEL ZAMORA QUITRAL
FECHA: 2019
PROF. GUÍA: MATÍAS MATTAMALA ARAVENA

SISTEMA DE VERMICOMPOSTAJE CON INTERNET DE LAS COSAS

En la actualidad, la cantidad y el depósito de desechos constituyen un problema de gran importancia para la sociedad. Si un desecho no es tratado, este pasa a convertirse en un residuo, quedando inutilizable para cualquier actividad humana o proceso natural.

El vermicompostaje es el proceso mediante el cual desechos orgánicos son degradados por lombrices rojas, obteniéndose como resultado un fertilizante orgánico de alta calidad, el cual recibe el nombre de hummus. Este es capaz de mejorar el suelo por su alto contenido de nutrientes hidrosolubles de fácil absorción para las plantas y por su gran capacidad de conservación de la humedad.

Para que el tratamiento de los desechos orgánicos pueda ser llevado a cabo adecuadamente, se requiere mantener condiciones idóneas de temperatura y humedad del ecosistema en el cual habitan las lombrices, llamado vermicompost. Bajo estas condiciones idóneas, se logra una reducción veloz e inodora de los desechos y, en consecuencia, un hummus de buena calidad.

Se crea conjunto con la estudiante de Diseño Industrial, Camila Martín Canales, un sistema de vermicompostaje con internet de las cosas orientado a mejorar la experiencia del usuario y se lo valida a nivel de prototipo funcional. El sistema consiste en un electrodoméstico el cual tiene asociado una aplicación móvil. Mediante la aplicación, los usuarios del electrodoméstico pueden educarse interactivamente sobre el vermicompostaje y realizar un monitoreo periódico, cada 1 hora, de la temperatura y la humedad del vermicompost desde un *smartphone* con conexión a internet. Frente a condiciones de operación fuera de rangos adecuados, el electrodoméstico activa un led indicador y la aplicación móvil notifica al usuario indicándole qué parámetro físico se encuentra fuera de rango y qué acción de control ejercer para volver a estabilizar las condiciones idóneas del vermicompost. De este modo, el sistema de vermicompostaje desarrollado busca mejorar la experiencia del usuario. Esto sienta las bases para en un futuro poder crear una red de reciclaje orgánico urbano en torno al vermicompostaje, la cual puede ser articulada e interconectada en forma virtual.

En este trabajo se propone, diseña e implementa una arquitectura de internet de las cosas a nivel de software y hardware, la cual permite incorporar esta nueva tecnología al sistema de vermicompostaje. Se obtiene como resultado un electrodoméstico inteligente de carácter funcional, que cuenta con un sistema de monitoreo de parámetros físicos en tiempo cuasi-real, es decir, cercano a un monitoreo en tiempo real. El monitoreo se realiza a través de la aplicación móvil asociada al electrodoméstico.

Este trabajo busca apoyar el reciclaje y la reutilización de desechos mediante el empleo de las nuevas tecnologías, por lo cual se encuentra plenamente documentado en todos sus niveles, tal que se facilite el escalamiento y masificación de la solución.

A la gente de la FAU, por abrirme los ojos.

Agradecimientos

A Camila Martín, por sentar las bases de este proyecto y escogerme como el complemento que hacía falta para llevar a cabo su realización.

Agradezco a Miguel Patiño, Matías Mattamala y Andrés Caba por despertar mi interés por los proyectos prácticos y multidisciplinarios, por depositar su confianza en mí para realizar su ejecución y por su disposición a responder mis consultas, brindarme retroalimentación y guía en los diversos aspectos tanto administrativos como académicos que he tenido que cubrir a lo largo de este camino.

A la Profesora Sandra Céspedes y al Profesor César Azurdia por su interés en este proyecto y los consejos útiles que me brindaron a lo largo del desarrollo del proyecto.

A Beauchef Proyecta por el financiamiento brindado y su interés en la aplicación práctica de los conocimientos teóricos aprendidos durante la estadía de los estudiantes en la Facultad de Ciencias Físicas y Matemáticas.

A la gente del Spot, la cual me aportó con instancias de discusión de diversas ideas y momentos de relaxo a lo largo de toda la carrera. Saludos afectuosos a los que estuvieron y los que están.

A los compañeros, compañeras, amigos y amigas de la Facultad de Arquitectura y Urbanismo que he podido conocer en los intentos de querer acercar conocimientos teóricos y tecnología de punta a la gente común y corriente, mediante la creación de productos tecnológicos orientados a contribuir íntegra y positivamente a su diario vivir.

A Florencia Bruzzone y Carolina Belmar por contactarme con Camila Martín y darme la oportunidad de poder aplicar y transmitir mis conocimientos técnicos.

A mi familia, por ser mi motivación principal para seguir adelante con mis objetivos. Cuando las fuerzas eran pocas siempre estuvieron ahí con palabras de ánimo y buena comida para reponer las energías y poder continuar para llegar hasta el final.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Antecedentes	2
1.3. Definición del Problema	2
1.4. Enfoque Multidisciplinario	3
1.5. Objetivos	4
1.6. Estructura del Documento	4
2. Marco Teórico	5
2.1. Estado del Arte	5
2.1.1. Automatización del Proceso de Vermicompostaje	5
2.1.2. Uso de Módulos GSM para Transmisión de Datos y Notificaciones al Usuario	7
2.1.3. Uso de Zigbee para Transmisión de Datos	7
2.1.4. Uso de Raspberry Pi como servidor	8
2.1.5. Hogar Inteligente con ESP8266 y Firebase	9
2.1.6. Sistema de Seguridad para Hogar Inteligente basado en Android utilizando internet de las cosas y Firebase	10
2.2. Síntesis del Estado del Arte	10
2.2.1. Microcontroladores, Procesadores y Protocolos de Comunicación	10
2.2.2. Almacenamiento de Datos	11
2.2.3. Interfaz de Usuario y Sistema de Notificaciones	11
3. Metodología	12
3.1. Metodología <i>Agile-Scrum</i> para desarrollo de software y hardware	12
3.1.1. El equipo <i>Scrum</i>	13
3.1.2. El proceso <i>Scrum</i>	13
3.1.3. Beneficios de <i>Scrum</i> para desarrollo de software y hardware	14
3.2. Metodología de trabajo propia basada en <i>Scrum</i>	14
4. Desarrollo del Trabajo	16
4.1. Bases del Estado del Arte	16
4.1.1. Microcontroladores, Procesadores y Protocolos de Comunicación	17
4.1.2. Almacenamiento de Datos	17
4.1.3. Interfaz de Usuario y Sistema de Notificaciones	17
4.2. Arquitectura de Internet de las Cosas	18

4.3.	Comportamiento del Electrodoméstico	20
4.3.1.	Rangos Óptimos de Temperatura y Humedad	20
4.3.2.	Diagrama de Comportamiento	20
4.4.	Arquitectura de Software	23
4.4.1.	Thunkable	23
4.4.2.	Firebase	24
4.4.3.	Librerías de Arduino	24
4.5.	Arquitectura de Hardware	26
4.5.1.	Led indicador	26
4.5.2.	Sensor de Temperatura	27
4.5.3.	Sensores de Humedad	28
4.5.4.	Microcontrolador ESP8266	30
4.5.5.	Fuente de Alimentación y Consumo de Corriente	31
4.6.	Obtención del Porcentaje de Humedad Relativa del Vermicompost	32
4.7.	Diseño e Implementación del Electrodoméstico	33
4.7.1.	La Vermi 1.0 - Primer Prototipo	33
4.7.2.	La Vermi 2.0 - Segundo Prototipo	40
4.7.3.	Versión Final	42
4.8.	Aplicación Móvil	46
4.8.1.	Sistema de Notificaciones	48
4.8.2.	Pantallas	50
4.9.	Resultados obtenidos	57
5.	Proyecciones	59
5.1.	Proyecciones de Software	59
5.1.1.	Arquitectura del Proyecto	59
5.1.2.	Base de Datos	60
5.1.3.	Aplicación Móvil Nativa	60
5.1.4.	Predicción de Series de Tiempo	61
5.2.	Proyecciones de Hardware	62
5.2.1.	Control Inteligente de Humedad	62
5.2.2.	Placa de Circuito Impreso	62
5.2.3.	Etiqueta de Eficiencia Energética	64
5.3.	Otras Proyecciones	65
6.	Conclusión	66
7.	Glosario	68
8.	Bibliografía	70
9.	Anexos	74
9.1.	Conexión al Wifi	74
9.2.	Costo de Hardware y Materiales	75
9.3.	Repositorio de Github	75

Índice de Tablas

4.1. Librerías utilizadas en el desarrollo del trabajo.	25
4.2. Consumos de corriente de la Vermi 2.0	31
4.3. Tabla con el nombre de las 5 pantallas principales que componen la aplicación móvil junto con su descripción y la referencia a la Figura correspondiente. . .	47
9.1. Tabla de costos de hardware y materiales eléctricos.	75

Índice de Ilustraciones

1.1. De izquierda a derecha: Matías Zamora, Lorna Lares, Camila Martín y Matías Mattamala.	3
2.1. Idea básica del módulo vermicompostador automático. Fuente: Pruthvi et al. [4]	6
2.2. Aplicación en Android mostrando las medidas en tiempo real de humedad, temperatura y concentración de metano. Fuente: Pruthvi et al. [4]	6
2.3. Raspberry Pi 3 Modelo B+. Es un pequeño computador. Fuente https://www.newegg.com/raspberry-pi-2773729-1gb-1pddr2-sdram/p/N82E16813142011	8
2.4. Microcontrolador ESP8266, ESP-12E o NodeMCU1.0 Fuente: https://www.letstango.com/esp8266-serial-wifi-module-nodemcuawifi-v3	9
2.5. Lecturas de Temperatura y Humedad del aire desplegadas en la aplicación móvil para la cocina de una casa. Fuente: Rawat et al. [10]	9
4.1. Primer prototipo de la Vermi en su fase de construcción. Se observa un diseño inicial de carácter modular, pensado para realizar vermicompostaje utilizando uno o más módulos montables uno sobre otro. Fuente: Camila Martín.	19
4.2. Diagrama de arquitectura de internet de las cosas desarrollado en el trabajo. A la izquierda se encuentra el electrodoméstico denominado Vermi, compuesto por el microcontrolador ESP8266, el sensor de humedad, el sensor de temperatura y el led indicador. La Vermi se comunica por <i>wifi</i> con el <i>router</i> del usuario para enviar los datos a la base de datos <i>cloud</i> Firebase. A la derecha se ilustra la aplicación móvil desarrollada en Thunkable para el sistema operativo móvil <i>Android</i> , la cual funciona como interfaz gráfica de usuario y muestra alertas en base a la información sensada.	19
4.3. Diagrama de Comportamiento de Software y Hardware de la Vermi. El indicador hace referencia al led RGB.	22
4.4. De izquierda a derecha: en primer lugar se tiene al Core de Arduino para ESP8266, el cual permite programar en forma sencilla al microcontrolador utilizando el IDE de Arduino. Es posible utilizar librerías de conectividad para hacer llamados a la API de Firebase y así ir almacenando los datos de la Vermi. Mediante esta misma API, la base de datos puede ser consultada desde la aplicación móvil. La aplicación móvil consiste en un <i>mockup</i> funcional creado en la plataforma web Thunkable.	23

4.5.	Arquitectura de hardware de la Vermi. De izquierda a derecha se observa en primer lugar el hardware que compone a la Vermi, con los sensores de temperatura, humedad y el led indicador. Luego, se puede observar que el ESP8266 se conecta al <i>router</i> del usuario mediante <i>wifi</i> , pudiendo así registrar la información en la base de datos <i>cloud</i> . Finalmente, las lecturas de temperatura y humedad se despliegan en el <i>smartphone</i> del usuario.	26
4.6.	Circuito auxiliar que permite regular los 5[V] que se requieren para un adecuado brillo del led rojo utilizando una señal de control de baja potencia, en este caso de 3.3[V]. Fuente https://electronics.stackexchange.com/questions/87967/rgb-led-brightness-levels?rq=1	27
4.7.	Termómetro Digital DS28B20 con su cubierta impermeable. Posee 3 cables, el rojo para la alimentación, el negro para la tierra y el amarillo para la transmisión de datos.	27
4.8.	Sensor de humedad resistivo YL-69. Se observan los electrodos del sensor (izquierda) y el amplificador de señal (derecha). Fuente https://www.fasttech.com/product/4738201-yl-69-soil-humidity-moisture-detection-sensor	28
4.9.	Sensor de Humedad Capacitivo V1.0 fabricado por DFRobot. Fuente https://wecl-stem.com/shop/058-77-2031/	29
4.10.	ESP8266MOD, también conocido como ESP12-E. Se observa la antena Wifi embebida. Fuente https://es.aliexpress.com/item/32324005820.html	30
4.11.	Pines del NodeMCU 1.0. Se aprecia el ESP8266MOD con su antena <i>wifi</i> , las funciones de sus pines, el puerto micro USB y los botones de <i>flash</i> y <i>reset</i> . Fuente https://iotbytes.wordpress.com/nodemcu-pinout/	31
4.12.	Fuente de Voltaje de propósito general con entrada de 220[Vac] y salidas de voltajes DC de 5[V] y 12[V]. Esta fuente cuenta con un fusible de 2[A] como protección. Esta fuente es capaz de alimentar cargas inductivas sin problemas de inestabilidad en los voltajes de alimentación.	32
4.13.	Primer prototipo de la Vermi instalado en una cocina. Está construido en terciado de pino insignie, un material barato y fácil de trabajar en una CNC. Tiene un sellado interior hecho con aceite de linaza, protegiendo la madera de la humedad. Fuente: Camila Martín.	34
4.14.	Módulo de sensado de la Vermi 1.0. Se tiene el sensor de temperatura DS18B20 , el sensor de humedad YL-69 , el led indicador de encendido y el Switch On/Off . Dentro de la caja fabricada con cartón pluma se encuentra el ESP8266, por la parte trasera sale el cable del cargador de teléfono Android.	34
4.15.	Estructura del documento JSON en la base de datos Firebase que guarda los datos históricos. Posee 3 campos que se actualizan en tiempo cuasi-real llamados: <i>hum</i> , <i>temp</i> y <i>timestamp</i> y 1 campo secundario llamado <i>logs</i> , el cual almacena los 3 campos principales separados por un espacio a lo largo del tiempo.	35
4.16.	Gráfico de Humedad y Temperatura en el tiempo (horas) para el día 21 de Febrero (02-21) del 2019 construido con 10 muestras tomadas en la Vermi 1.0. Cada marca en el eje del tiempo corresponde a las horas del día a partir de las 3 de la tarde. El valor representativo de la Humedad está dado por el sensor YL-69. Los puntos graficados son datos reales unidos por una línea de tendencia.	36

4.17. Gráfico de Humedad y Temperatura en el tiempo (días) para el mes de Febrero de 2019 construido con 128 muestras tomadas en la Vermi 1.0. Cada marca en el eje del tiempo corresponde a las 00:00 horas de día correspondiente. El valor representativo de la Humedad está dado por el sensor YL-69. Los puntos graficados son datos reales unidos por una línea de tendencia.	36
4.18. Sensores de Humedad YL-69. Al centro se observa un sensor YL-69 nuevo y a ambos lados otros dos utilizados en la Vermi 1.0, los cuales presentan signos de corrosión.	37
4.19. Gráfico de Humedad y Temperatura en el tiempo (días) para el mes de Marzo de 2019 construido con 241 muestras tomadas en la Vermi 1.0. El valor representativo de la Humedad está dado por el sensor de humedad capacitivo de DFRobot. Los puntos graficados son datos reales unidos por una línea de tendencia.	38
4.20. Gráfico de Humedad y Temperatura en el tiempo (días) para el mes de Marzo de 2019 construido con 241 muestras tomadas en la Vermi 1.0. El valor representativo de la Humedad está dado por la medida del sensor de humedad capacitivo de DFRobot incluyendo el ajuste lineal para la obtención del porcentaje de humedad relativa del vermicompost. Los puntos graficados son datos reales unidos por una línea de tendencia.	39
4.21. Segundo prototipo de la Vermi (Vermi 2.0). Corresponde a una maqueta construida con couché forrado que cuenta con un sistema de 2 cajones removibles que permiten la cosecha del hummus sólido y líquido. Fuente: Camila Martín.	40
4.22. Conexión al servidor web del ESP8266 desde un iPhone para entregar credenciales de la red local del usuario final. Fuente: Camila Martín.	40
4.23. Sensores de Temperatura (izquierda) y Humedad (derecha) instalados en la Vermi 2.0. Se observa una malla diseñada para permitir un arneado del hummus. Fuente: Camila Martín.	41
4.24. Parte trasera de la Vermi 2.0, se muestra el circuito situado por dentro con sus componentes afirmados por tornillos para madera de 5mm. Este espacio es cubierto por una tapa. Se tiene el microcontrolador ESP8266 , un botón de reset accesible, la fuente de voltaje que alimenta a toda la circuitería y el circuito auxiliar del led RGB	41
4.25. Vermi 2.0 sin su carcasa externa, se muestra el cableado del led RGB y el propio led RGB. La carcasa externa tiene un logo transparente hecho con mica translúcida que queda situado justo por delante del led. Fuente: Camila Martín.	42
4.26. Prototipo funcional del electrodoméstico. Este prototipo contiene vermicompost en su interior, se conecta a la red local del usuario mediante wifi solicitando sus credenciales, envía datos de temperatura y humedad cada 1 hora a la base de datos y activa su led indicador frente a condiciones de operación fuera de rango. Fuente: Camila Martín.	43
4.27. Prototipo visual del electrodoméstico instalado en una cocina. Sólo para fines estéticos. Fuente: Camila Martín.	44
4.28. Modelos 3D de la versión final del electrodoméstico. Fuente: Camila Martín.	45

4.29. Rangos adecuados de funcionamiento para humedad y temperatura almacenados en la base de datos bajo los nombres <i>maxSoilHumidity</i> , <i>minSoilHumidity</i> , <i>maxSoilTemp</i> y <i>minSoilTemp</i>	49
4.30. Alerta desplegada dentro de la aplicación para el caso en el cual la humedad se encuentra alta.	49
4.31. <i>Home</i> es la primera pantalla que se muestra cuando se abre la aplicación. Aquí se informa del estado general de la Vermi mediante el visor que se observa en la parte superior, en la parte inferior se despliega un dato curioso relativo al vermicompostaje o una notificación según corresponda.	50
4.32. Notificaciones de funcionamiento fuera de rango según cada caso. Se tienen 4 escenarios posibles definidos por umbrales inferiores y superiores para la humedad y temperatura respectivamente.	51
4.33. En la segunda pantalla se muestra el <i>Dashboard</i> de la aplicación, el cual funciona como un visor en tiempo cuasi-real de la humedad y temperatura del vermicompost. Las medidas se actualizan cada 1 hora desde el circuito y la aplicación responde frente a estos cambios periódicos en la base de datos. . .	52
4.34. En la pantalla <i>Problems</i> se muestran 6 botones con los problemas más típicos que se presentan al realizar el proceso de vermicompostaje, se pueden presionar para acceder a información más detallada sobre como abordar cada uno de ellos.	52
4.35. Información detallada sobre los problemas más típicos que pueden ocurrir. Al presionar <Volver, se vuelve a la pantalla <i>Problems</i> . Para una mejor visualización de la figura recurrir a la versión digital de este documento.	53
4.36. En la pantalla <i>Information</i> se muestran 3 botones deslizables que conducen a otras pantallas con información sobre las lombrices, los beneficios que el abono producido tiene para el suelo y los desechos permitidos y no permitidos que pueden agregarse a la Vermi.	54
4.37. Información general que muestran los 3 botones de la pantalla <i>Information</i> . Al presionar <Volver, se vuelve a la pantalla <i>Information</i>	55
4.38. En la pantalla <i>Community</i> se muestran dos botones. A la izquierda se accede a la página web del proyecto y a la derecha, como proyección futura, se encuentra el acceso a lo que será la comunidad, en la cual se podrá intercambiar lombrices, abono y experiencias.	56
4.39. Prueba del sistema de vermicompostaje con internet de las cosas.	57
5.1. Arquitectura de Software incluyendo una API, la cual vincula a la base de datos con las aplicaciones y maneja la lógica de negocio. Fuente https://cianbox.com/blog/novedades-de-agosto-y-septiembre-2018/14986416433514_api/	60
5.2. Ejemplos de series de tiempo estacionaria (verde) y no estacionarias (rojo). Fuente: Aishwarya singh - A Gentle Introduction to Handling a Non-Stationary Time Series in Python.	62
5.3. Esquemático del circuito a implementar en la Vermi hecho en Fritzing. . . .	63

- 5.4. La etiqueta de eficiencia energética tiene dos partes fundamentales. En la parte izquierda, la matriz, que contiene las categorías energéticas, unidades y conceptos sobre los que se va a dar información. En la parte derecha, la ficha, que contiene los datos específicos que corresponden al electrodoméstico que lleva la etiqueta. Por medio de la escala de clases o categorías, se ofrece información sobre el grado de eficiencia energética del electrodoméstico en comparación con otros aparatos similares. La escala utiliza un código de colores y letras. La gama va desde el color verde y la letra A para los equipos más eficientes, hasta el color rojo y la letra G para los equipos menos eficientes. 64

Capítulo 1

Introducción

En este primer capítulo se comienza explicando la motivación a la realización de este trabajo, para luego dar antecedentes sobre enfoque tecnológico. Posteriormente se procede a la definición del problema y luego se destaca el aspecto multidisciplinario del trabajo. Finalmente, se enuncian los objetivos tanto generales como específicos que se esperan alcanzar y se detalla la estructura del documento.

1.1. Motivación

El presente trabajo consiste en la creación y validación de un sistema de vermicompostaje con internet de las cosas. Se propone, diseña e implementa una arquitectura de internet de las cosas a nivel de software y hardware, que permite incorporar esta tecnología al sistema de vermicompostaje. El resultado obtenido es un electrodoméstico inteligente que permite realizar este proceso a nivel doméstico y en espacios reducidos que cuenta con una aplicación móvil asociada. Este sistema de vermicompostaje con internet de las cosas es desarrollado en conjunto con la Diseñadora Industrial de la Facultad de Arquitectura y Urbanismo de la Universidad de Chile, Camila Martín Canales¹.

Mediante el sistema de vermicompostaje, el usuario puede participar activamente de la formación, mantenimiento y cosecha del hummus. El electrodoméstico cuenta con una unidad de sensado inalámbrica capaz de medir la humedad y temperatura del vermicompost, los valores de estos parámetros físicos son proyectados en un *dashboard*² dentro de la aplicación móvil en tiempo cuasi real³. Si la humedad o la temperatura se encuentran fuera de rangos previamente determinados, se muestran alertas correspondientes tanto en la aplicación como en el electrodoméstico. Así, el usuario es instruido sobre el funcionamiento del sistema y de ser requerida alguna acción de mantención, esta le será notificada a través de la misma aplicación. Mantener las condiciones idóneas de temperatura y humedad es clave para la

¹camila.martin@ug.uchile.cl

²Tipo de interfaz gráfica de usuario destinada a informar sobre el estado parámetros relevantes de un proceso.

³i.e. cercano al monitoreo en tiempo real.

obtención de un hummus de buena calidad. Con esto se busca mejorar la experiencia entre el usuario y el proceso de vermicompostaje a través de su participación activa utilizando las nuevas tecnologías.

Los componentes de la arquitectura de internet de las cosas propuesta se escogen privilegiando su compatibilidad, escalabilidad, *performance* y también considerando los costos asociados al desarrollo, procurando minimizarlos utilizando servicios gratuitos, software *open source*⁴ y prefiriendo hardware accesible comprando a proveedores locales.

1.2. Antecedentes

El paradigma tecnológico del internet de las cosas cambiará prontamente la forma en la cual los humanos interactúan con objetos cotidianos en su diario vivir.

En 1999, Kevin Ashton utilizó por primera vez el término para referirse a la forma en la cual la red de computadores de ese entonces recolectaban la información, la cual posteriormente era almacenada en un servidor y consultada por un cliente [1]. En aquellos tiempos, toda la información proveniente del mundo real disponible en la red era creada por los mismos usuarios, los cuales tienen la capacidad de subir contenido de diversas formas: comunicándose a través de plataformas web escribiendo con su teclado, creando archivos de tipo multimedia tomando fotografías o grabando videos con una cámara.

Si la red recolecta datos que representen la información cuantificable del mundo real, sería posible monitorear y/o controlar diversas variables físicas de interés para cada proceso, siendo posible la reducción de desechos, ahorro de recursos y desarrollo de los procesos dentro de rangos óptimos dependiendo del caso de uso. Para concretizar esta idea, la red debe ser capaz de obtener datos del mundo real por sí misma, sin depender de la intervención de los usuarios.

Con el fin de medir parámetros físicos, se requiere de sensores y microcontroladores que realicen una lectura y digitalización del parámetro de interés. Luego, estos datos deben ser transportados hasta un centro de almacenamiento para luego ser desplegados ante el usuario final con el objetivo de que este sea capaz de realizar un control (manual o automático) y/o tareas de monitoreo a modo de lograr una supervisión inteligente del proceso. Para realizar este levantamiento de datos, es necesario escoger los componentes de una arquitectura de internet de las cosas tanto a nivel de software como a nivel de hardware.

1.3. Definición del Problema

Cualquier proceso en el cual existan variables físicas involucradas en forma importante es una potencial aplicación del internet de las cosas, a fin de medir y/o controlar estas variables.

⁴El propósito fundamental de las licencias *open source* es denegar a cualquiera el derecho de explotar en forma exclusiva un trabajo.

Particularmente, dentro del contexto de la sustentabilidad y agricultura urbana, el vermicompostaje surge como la temática central a orientar en este trabajo. El vermicompostaje es el proceso mediante el cual la materia orgánica es degradada en forma rápida y eficiente por lombrices rojas californianas y otros microorganismos, obteniéndose como productos del proceso biomasa de lombrices y hummus. Dentro del sistema de vermicompostaje, el vermicompost es una mezcla compuesta por tierra húmeda, lombrices rojas y desechos orgánicos en proceso de degradación. Mantener a esta mezcla dentro de rangos adecuados de humedad y temperatura es clave para una reproducción adecuada de las lombrices, una reducción veloz e inodora de los desechos orgánicos y también para la obtención de un hummus de buena calidad [2]. Así, la necesidad de medir periódicamente estas variables físicas hacen del vermicompostaje un caso de uso interesante en el cual aplicar una solución de internet de las cosas.

1.4. Enfoque Multidisciplinario

Un aspecto fundamental de esta memoria es su carácter multidisciplinario. Su ejecución se lleva a cabo bajo la tutela de Beauchef Proyecta. Beauchef Proyecta es la nueva área de proyectos multidisciplinarios de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, iniciativa inserta en el proyecto *Ingeniería y Ciencias 2030*. Este tiene por objetivo la participación de los estudiantes en el desarrollo de proyectos de ingeniería multidisciplinarios, desde su concepción hasta el producto final [3].

Esta memoria se enmarca dentro del proyecto *Sustenta Vermi*, el cual es encabezado por la diseñadora industrial de la Facultad de Arquitectura y Urbanismo (FAU) Camila Martín Canales, guiada por la académica y diseñadora industrial Lorna Lares. Como contraparte de ingeniería, el desarrollo tecnológico está liderado por el autor de este documento, Matías Zamora, con apoyo de Matías Mattamala, ingeniero eléctrico. El equipo se muestra en la Figura 1.1.



Figura 1.1: De izquierda a derecha: Matías Zamora, Lorna Lares, Camila Martín y Matías Mattamala.

1.5. Objetivos

El objetivo general de esta memoria es crear un sistema de vermicompostaje con internet de las cosas orientado a mejorar la experiencia de usuario y validarlo a nivel de prototipo funcional.

Los objetivos específicos de este trabajo son:

- Diseñar e implementar una arquitectura de software y hardware que permita incorporar el internet de las cosas al sistema de vermicompostaje y evaluar su funcionamiento en un despliegue real en campo.
- Diseñar e implementar, en forma multidisciplinaria, un electrodoméstico funcional que permita realizar el proceso de vermicompostaje a nivel doméstico y en espacios reducidos.
- Diseñar e implementar, en forma multidisciplinaria, una aplicación móvil asociada al electrodoméstico como sistema de notificación en base a las condiciones de temperatura y humedad del vermicompost.
- Documentar debidamente tanto el proceso de desarrollo y como el funcionamiento de hardware y software.

1.6. Estructura del Documento

Este documento se estructura como sigue: en el Capítulo 2 se analiza el estado del arte y se resumen las ideas más relevantes a este trabajo. En el Capítulo 3 se presenta la metodología de desarrollo utilizada. El Capítulo 4 presenta el desarrollo del electrodoméstico propuesto para el sistema de vermicompostaje, especificando los aspectos de diseño e implementación de hardware y software, definidos junto a Camila Martín. El Capítulo 5 presenta las principales proyecciones futuras en caso de continuar con el proyecto y la Conclusión resume los desarrollos y aprendizajes adquiridos en este trabajo de memoria.

Capítulo 2

Marco Teórico

En este segundo capítulo se estudian iniciativas relacionadas a este trabajo a nivel del estado del arte, las cuales permitirán esclarecer como otros desarrolladores proponen e implementan arquitecturas de internet de las cosas, partiendo desde el sensado de los parámetros físicos hasta el despliegue de los datos ante usuario final. Finalmente, se realiza una síntesis del estado del arte. Esta síntesis sienta las bases para que el trabajo sea atractivo e innovador. Se detalla tanto el hardware como el software utilizado.

2.1. Estado del Arte

2.1.1. Automatización del Proceso de Vermicompostaje

El trabajo de Pruthvi et al. [4] aborda el tema de las condiciones óptimas para la vida de las lombrices y la operación dentro de rangos adecuados de parámetros físicos relativos al vermicompostaje. Esto con el fin de producir hummus de buena calidad para su uso en la agricultura. Particularmente, se destaca que los fertilizantes orgánicos como el hummus son preferidos por sobre los fertilizantes químicos para cuidar la calidad del producto comestible.

Los autores proponen un módulo vermicompostador automático, el cual cuenta con sensores de humedad del suelo, temperatura y metano, un microcontrolador, un módulo bluetooth para la transmisión de datos y un actuador orientado a humectar el vermicompost. El esquema general se muestra en la Figura 2.1.

Así, los autores implementan un sistema de control de la humedad utilizando una válvula solenoide conectada a la salida de un estanque de agua para regar en forma automática la mezcla si la humedad se encuentra fuera de los rangos adecuados. Como interfaz de usuario los autores implementaron una aplicación en Android, la cual recibe la información desde el vermicompostador mediante bluetooth y despliega las lecturas de los parámetros en tiempo real. La interfaz de esta aplicación se muestra en la Figura 2.2.

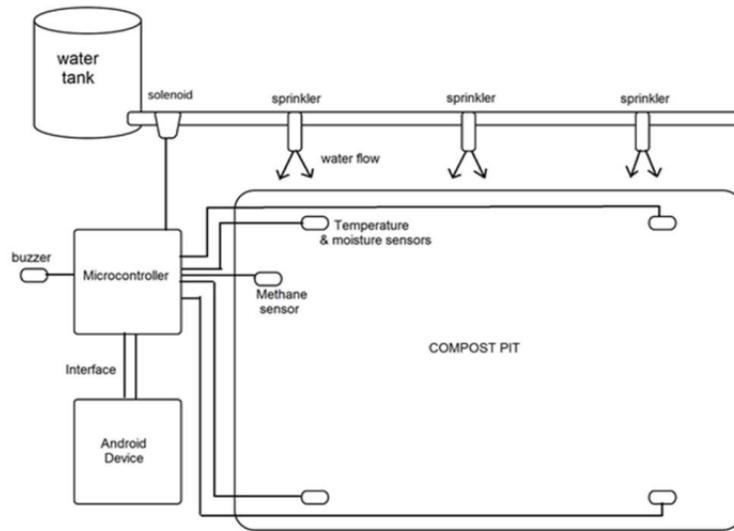


Figura 2.1: Idea básica del módulo vermicompostador automático. Fuente: Pruthvi et al. [4]

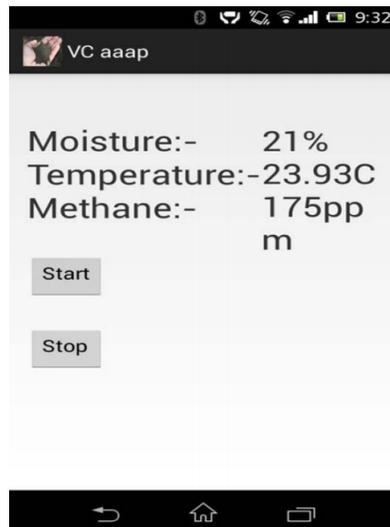


Figura 2.2: Aplicación en Android mostrando las medidas en tiempo real de humedad, temperatura y concentración de metano. Fuente: Pruthvi et al. [4]

2.1.2. Uso de Módulos GSM para Transmisión de Datos y Notificaciones al Usuario

En el trabajo de Vishal et al. [5] se aborda el cultivo de tomates dentro del contexto de la agricultura en la India. En particular, se abarca la problemática de la cosecha del tomate, buscando notificar al granjero con un mensaje a su teléfono cuando el tomate se encuentre listo para ser cosechado basándose en su coloración. Para ello se utilizan técnicas de procesamiento de imágenes.

Como sensor del mundo físico se utiliza una cámara, como unidad de procesamiento y almacenamiento un computador, un microcontrolador arduino programado con las alertas para el granjero y finalmente un módulo GSM¹ para la transmisión notificaciones vía SMS².

Los autores plantean como trabajo futuro el envío de un documento html³ con los resultados del procesamiento de imágenes, con el fin de entregar al granjero información en forma más gráfica y elaborada en comparación a un mensaje de texto.

2.1.3. Uso de Zigbee para Transmisión de Datos

En el trabajo de Mayuresh et al. [6] se propone el uso del protocolo de comunicación Zigbee para interconectar redes de sensores de parámetros físicos dentro del caso de uso de la agricultura.

Los parámetros físicos de interés en ese trabajo son principalmente la humedad del suelo para el diseño de un sistema de riego automático por zonas y en segundo plano el ph del suelo, para evitar su sobre-fertilización.

Para articular la red, se quiere de un nodo Zigbee que funcione como receptor del emisor para así realizar el levantamiento de los datos a internet. En este caso, el nodo Zigbee se encuentra conectado a un notebook dentro de la casa del agricultor.

Por otro lado, en el trabajo de Mazaruddin et al. [7] se utiliza el mismo protocolo de comunicación para diseñar estaciones de monitoreo ambiental de la temperatura y humedad del aire alimentadas con paneles solares para ser dispuestas en los campos de cultivo de la India. Estas estaciones tienen la capacidad de comunicarse entre sí, pudiendo servir al mismo tiempo tanto como una unidad de sensado como una unidad de extensión de la red. Las lecturas son enviadas a una estación de monitoreo central mediante el protocolo Zigbee, desde esta estación y utilizando un sistema GSM se envían las lecturas a un computador para su despliegue final en un servidor web con interfaz gráfica de usuario.

Este protocolo de comunicación posee fortalezas importantes a considerar dentro de las aplicaciones del internet de las cosas, como lo son un bajo consumo energético, buena penetración de paredes, fiabilidad en la transmisión y una buena extensibilidad de la red en sí

¹ *Global System for Mobile communications*

² *Short Message Service*

³ *Hypertext Markup Language*

misma, ya que es posible agregación o eliminación de nodos Zigbee para aumentar o disminuir el rango de cobertura inalámbrica.

Zigbee opera en las bandas de radio ISM (*Industrial - Scientific - Medical*):

- 915 Mhz en Estados Unidos y Australia
- 868 Mhz en Europa
- 2.4 Ghz en el resto del mundo

2.1.4. Uso de Raspberry Pi como servidor

En el trabajo de Qian et al. [8] se fabrica un sistema de detección de humedad y temperatura de un sistema de comunicaciones basado en la placa de desarrollo Raspberry Pi. Para realizar el levantamiento de datos a internet, se aprovecha el adaptador de red Ethernet que la Raspberry Pi tiene incorporado, como muestra la Figura 2.3.



Figura 2.3: Raspberry Pi 3 Modelo B+. Es un pequeño computador. Fuente <https://www.newegg.com/raspberry-pi-2773729-1gb-1pddr2-sdram/p/N82E16813142011>

El adaptador de red Ethernet permite una conexión directa, sencilla y compatible con cualquier *router*. Así, los datos recolectados por los sensores de humedad y temperatura son accesibles a través este medio.

En el servidor web se ejecuta el procesamiento y el despligue de los datos adquiridos. Los usuarios remotos pueden utilizar cualquier dispositivo con acceso a internet para acceder a la dirección IP⁴ del servidor web utilizando cualquier navegador web, pudiendo realizar monitoreo remoto en tiempo real de la humedad y la temperatura del sistema de comunicaciones.

El entorno de desarrollo web que se utiliza es WAMP, el cual permite crear aplicaciones web en un servidor web Apache, se programa en PHP en el lado del servidor y se implementa un sistema de base de datos MySQL [9], realizando así la integración completa del sistema de detección.

⁴*Internet Protocol*

2.1.5. Hogar Inteligente con ESP8266 y Firebase

En el trabajo de Rawat et al. [10] se presenta un prototipo de *Smart Home* que utiliza *wifi* para la comunicación entre 2 tipos de sensores y una aplicación móvil que despliega alertas de acuerdo a los valores de los parámetros sensados en tiempo real.

Los sensores utilizados corresponden al sensor de movimiento PIR (*Passive Infrared Sensor*) y los sensores de gases MQ135 y MQ6. En este trabajo se busca dar la posibilidad a usuarios con limitaciones en su movimiento, como puede ser gente de la tercera edad, a poder controlar la iluminación de su hogar a través de la aplicación móvil, así como también poder estimar la calidad de aire y detectar incendios.

El microcontrolador al cual van conectados los sensores es el ESP-12E (ver Figura 2.4), el cual a su vez registra los datos en la base de datos Firebase de Google [11]. La aplicación móvil (ver Figura 2.5) es programada en Java utilizando *Android Studio* y se conecta a la base de datos Firebase, obteniendo de esta forma una arquitectura de internet de las cosas con interfaz de usuario en un dispositivo móvil con sistema operativo Android.



Figura 2.4: Microcontrolador ESP8266, ESP-12E o NodeMCU1.0 Fuente: <https://www.letstango.com/esp8266-serial-wifi-module-nodemculuawifi-v3>

A screenshot of a mobile application interface. At the top, there is a status bar with the time 22:34, data speed 0.01K/s, signal strength, VoLTE, and battery level at 54%. Below the status bar, there is a table with four rows of data. Each row represents a reading from the kitchen. The columns are labeled 'Kitchen', temperature in degrees Celsius, humidity in percent, and another temperature in degrees Celsius.

Kitchen	26.0 *C	40.0 %	25.7 *C
Kitchen	25.0 *C	38.0 %	24.55 *C
Kitchen	26.0 *C	39.0 %	25.67 *C
Kitchen	25.0 *C	40.0 %	24.6 *C

Figura 2.5: Lecturas de Temperatura y Humedad del aire desplegadas en la aplicación móvil para la cocina de una casa. Fuente: Rawat et al. [10]

2.1.6. Sistema de Seguridad para Hogar Inteligente basado en Android utilizando internet de las cosas y Firebase

En el trabajo de Gayen et al. [12] se lleva a cabo la implementación y diseño de un sistema de control inalámbrico para hogares inteligentes utilizando *wifi*. Se detalla como el internet de las cosas puede ser implementado en el ámbito de la seguridad y automatización de casas utilizando el microcontrolador ESP8266, Firebase como sistema de manejo de base de datos y una aplicación móvil en Android.

Se utiliza el sensor PIR para la detección de movimiento a fin de notificar al usuario si se presenta movimiento en cierta área del hogar y también un sensor de fuego para notificar al usuario de incendios. Para el sistema de notificaciones, se utiliza el sistema de manejo de base de datos Firebase. Cada vez que los sensores son disparados, el ESP8266 envía un JSON⁵ a la base de datos el cual contiene el *id*⁶ de la notificación correspondiente, el *id* del dispositivo móvil y el *id* del usuario a notificar. Luego, la notificación es enviada al dispositivo mediante las *Firebase Functions*, escritas en Javascript. Estas se encargan de procesar la información de los sensores y emitir la alerta en el dispositivo del usuario.

2.2. Síntesis del Estado del Arte

El presente trabajo intenta validar el diseño y la implementación una arquitectura de internet de las cosas a nivel de software y hardware tomando en consideración los trabajos anteriormente presentados. Para validar el despliegue real de la arquitectura propuesta, esta se implementa en un electrodoméstico inteligente de carácter funcional único en su tipo.

2.2.1. Microcontroladores, Procesadores y Protocolos de Comunicación

A continuación se detallan los protocolos de comunicación, procesadores y microcontroladores utilizados en los trabajos presentados. En el trabajo de Pruthvi et al. [4] comunican al microcontrolador con el *smartphone* inalámbricamente mediante bluetooth. Por otro lado, los trabajos de Vishal et al. [5], Mayuresh et al. [6] y Mazaruddin et al. [7], utilizan los protocolos de comunicación inalámbrica GPRS/GSM y Zigbee para el traspaso de datos.

En el trabajo de Qian et al. [8] se aprovecha el adaptador de red Ethernet embebido en la Raspberry Pi para establecer una conexión cableada en forma directa al *router*.

Alternativamente, los trabajos de Rawat et al. [10] y Gayen et al. [12] utilizan la interfaz *wifi* embebida en el microcontrolador ESP8266 para establecer una conexión inalámbrica con el *router* utilizando un enlace que opera en 2.4 [GHz].

⁵Un *Javascript Object Notation* es una estructura de datos abstracta en forma de texto utilizada para enviar y recibir información mediante llamadas http.

⁶identifier

2.2.2. Almacenamiento de Datos

Se resumen a continuación los sistemas de almacenamiento de los datos recolectados por los sistemas de sensado presentados. El trabajo de Pruthvi et al. [4] no realiza almacenamiento de datos de ningún tipo. Por otro lado, los trabajos de Vishal et al. [5], Mayuresh et al. [6] y Mazaruddin et al. [7] almacenan y procesan los datos en servidores locales, los cuales se encuentran ubicados en computadores que son propiedad de los mismos usuarios.

En el trabajo de Qian et al. [8] también se utiliza un servidor local para el almacenamiento de los datos, sin embargo este se encuentra implementado en la misma Raspberry Pi, es decir, la arquitectura de internet de las cosas desarrollada en este trabajo sensa, almacena, procesa y dispone los datos dentro de la red utilizando este pequeño computador.

Los trabajos de Rawat et al. [10] y Gayen et al. [12] utilizan la base de datos *cloud* Firebase para almacenar, procesar los datos y realizar una acción en base a esta información.

2.2.3. Interfaz de Usuario y Sistema de Notificaciones

La interfaz de usuario desarrollada en el trabajo de Pruthvi et al. [4] muestra medidas en tiempo real de parámetros de interés del vermicompost mediante una aplicación móvil en *Android*. Esto se ilustra en la Figura 2.2. La planta automática de vermicompost implementada no posee ningún sistema de alertas ni notificaciones en base a estas medidas.

El trabajo de Vishal et al. [5] implementa un sistema de notificaciones para el usuario mediante mensajes de texto. No se desarrolla una interfaz gráfica de usuario propiamente tal.

Por otro lado, los trabajos de Mayuresh et al. [6], Mazaruddin et al. [7] y Qian et al. [8] implementan a modo de interfaz gráfica de usuario una página web utilizando un servidor local, el cual realiza consultas a una base de datos, la cual también es de carácter local. En esta interfaz se despliegan alertas frente a operación fuera de rangos predeterminados por los autores.

Finalmente, Rawat et al. [10] y Gayen et al. [12] desarrollan aplicaciones móviles para *Android* a modo de interfaz gráfica de usuario, las cuales muestran notificaciones y lecturas de los parámetros de interés en tiempo real, como muestran la Figura 2.5. En ambos casos, la aplicación se encuentra vinculada a Firebase, un servicio de base de datos *cloud*.

Capítulo 3

Metodología

En este capítulo se presenta la metodología utilizada en este trabajo multidisciplinario, la cual tiene como eje central de funcionamiento la comunicación entre los miembros del equipo. Esta metodología se aplicó tanto para el desarrollo de prototipos funcionales a nivel de software y hardware.

3.1. Metodología *Agile-Scrum* para desarrollo de software y hardware

Las metodologías ágiles están pensadas para el desarrollo frecuente de prototipos funcionales, comunicación continua tanto con los clientes y como con los miembros del equipo y flexibilidad frente a cambios en los requerimientos iniciales. Las metodologías ágiles se utilizan para desarrollar productos en forma incremental mediante iteraciones del proceso de producción y suelen ser más efectivas cuando el proyecto a desarrollar requiere de creatividad, cuando puede ser dividido en partes más pequeñas y cuando debe ser diseñado y desarrollado para ajustarse a ciertas expectativas [13].

La forma de trabajo más típica dentro de las metodologías ágiles se conoce como *Scrum*. Esta metodología ha demostrado ser de utilidad para aumentar la productividad, eficiencia y la calidad del trabajo de los equipos de desarrollo de software [14]. A pesar de esto, muchos equipos *Scrum* han logrado aplicar exitosamente esta metodología de trabajo al desarrollo de hardware [15].

3.1.1. El equipo *Scrum*

Existen 3 roles en el equipo *Scrum*:

- *Product Owner*: Encargado de mantener las conversaciones con el cliente. Es el responsable de mantener actualizados los requerimientos del proyecto y comunicarse con los clientes, esto con el fin de asegurarse de que el equipo se encuentra desarrollando el producto correcto.
- *Scrum Master*: Facilita el éxito del equipo removiendo los impedimentos a la comunicación y al progreso técnico. Es el encargado de pavimentar el camino para que el progreso sea llevado a cabo como está planeado.
- *Development Team*: Responsables de ejecutar y completar las tareas a mano. Suelen tener un carácter multidisciplinario y auto-organizativo. Esto significa que cada miembro puede realizar un número de tareas, sin que estas sean delegadas específicamente por el *Scrum Manager*.

El autor de este documento toma el rol de *Scrum Manager*, manteniendo actualizado un *backlog*¹ de acuerdo a los avances logrados luego de cada jornada de trabajo.

Camila Martín es quién mejor entiende los requerimientos de los usuarios finales del producto, por lo cual asume el rol de *Product Owner*.

La figura del cliente dentro del esquema *Scrum* es el usuario final del producto.

Por último, tanto el autor del documento como Camila componen también el *Development Team*, aportando al trabajo desde la perspectiva de la ingeniería y del diseño de productos.

3.1.2. El proceso *Scrum*

El proceso comienza cuando los miembros del equipo se reúnen por primera vez y crean un *backlog*.

Una vez creado el *backlog*, los miembros del equipo se reúnen y lo optimizan agregando o quitando tareas, creando subdivisiones de las tareas complejas en partes más pequeñas, ordenando las tareas por orden de importancia, estimando el valor del negocio, clarificando los requerimientos de cada tarea y determinando que tareas ya se encuentran listas. Este proceso debe ejecutarse continuamente a lo largo del desarrollo del proyecto. Luego, las tareas son divididas en *sprints* de corta duración, con una duración entre 1 y 4 semanas máximo.

Para que todo el equipo se encuentre alineado en cuanto al desarrollo que los distintos integrantes del mismo están llevando a cabo, tener reuniones seguidas es clave para el éxito de esta metodología. Lo típico es tener reuniones diarias y breves de 15 minutos y una reunión más larga pero con límite de 2 horas máximo para cada *sprint* semanal. En la reunión corta

¹Un *backlog* es una lista de tareas que permiten desarrollar un proyecto basándose en las necesidades del cliente.

se informa del progreso y/o de impedimentos que puedan estar entorpeciendo el desarrollo, mientras que en la reunión larga se trabaja sobre los puntos más prioritarios del *backlog*.

Cuado un *sprint* llega a su fin, se invita al cliente a la reunión larga para mostrar los trabajos realizados, independiente de si este está completado o no. La idea es obtener retroalimentación directamente de la contraparte y considerarlo para los futuros *sprints*.

3.1.3. Beneficios de *Scrum* para desarrollo de software y hardware

A continuación se listan los beneficios más importantes y con influencia determinante en el desarrollo de hardware y software:

1. Se le entrega al cliente una visión mucho más clara en cuanto al progreso del proceso constructivo de su producto.
2. Como se considera un espacio para cambios en las expectativas y requerimientos, el proceso de desarrollo resulta en un mejor producto. Las metodologías tradicionales yacen sobre un conjunto de requerimientos establecidos al comienzo del proceso de desarrollo. Si estos requerimientos cambian, el resultado será un producto menos satisfactorio para el cliente.
3. Gracias a las reuniones frecuentes, los miembros del equipo tienen la posibilidad de evaluar continuamente ventajas y desventajas de la manera en la cual se lleva a cabo del desarrollo, teniendo la capacidad de cambiar elementos para facilitar un proceso más eficiente y efectivo.
4. Conservando la idea original de *Scrum* de entregar productos funcionales con frecuencia, construir prototipos y maquetas sencillas con materiales baratos que permitan validar los requerimientos del *backlog*, o preferir simulaciones de los productos previo a su desarrollo concreto, permite ahorrar esfuerzos en términos de tiempo y dinero. Es más fácil realizar ajustes sobre simulaciones que sobre entregas parciales del producto.
5. La modularidad permite paralelismo en el trabajo.
6. Las metodologías ágiles son adaptables a cada proyecto particular. Cada equipo es diferente, por lo cual se pueden proponer distintas adaptaciones a *Scrum*.

3.2. Metodología de trabajo propia basada en *Scrum*

En el proceso de desarrollo del presente trabajo se utilizó una adaptación de la metodología ágil *Scrum*. Se preservan los aspectos siguientes:

- Frecuencia de las reuniones: La comunicación es uno de los aspectos más importantes dentro de esta metodología.
 - Se acuerdan reuniones, en conjunto con Camila, con el profesor guía Matías Matamala y la Profesora Lorna Lares cada un lapso de 2 semanas.
 - Por su parte, Camila Martín realiza correcciones cada 2 semanas con su Profesora Guía Lorna Lares.

- Se agenda una reunión mensual con los profesores César y Sandra en conjunto con Camila para mostrar avances de prototipo funcional.
- Para asuntos que sean demasiado específicos de cada disciplina, Camila o el autor acuerdan consultar directamente al profesor más adecuado y luego sintetizar lo discutido, poniendo énfasis al aspecto que se encuentra en ese momento deteniendo el avance del proyecto.
- Uso de herramientas y materiales de prototipado:
 - Con el fin de ahorrar tiempo y dinero en el desarrollo del electrodoméstico, se construyeron prototipos de baja resolución fabricados con materiales de bajo costo, de fácil obtención y sencillos de manipular. Ejemplos de esto son el pino insigne terciado para el primer prototipo del electrodoméstico o el cartón pluma para las cajas que contienen la circuitería. Esto además permite obtener dimensiones aproximadas del tamaño del circuito final a utilizar.
 - Pasando al plano del software, hoy en día existen herramientas de prototipado rápido que permiten realizar pruebas de concepto fácilmente. Se destaca en primer lugar Node-RED [16], herramienta de programación basada en flujos para *IoT*. Otra herramienta a utilizar es Thunkable[17]. Thunkable sirve para crear aplicaciones móviles funcionales y es un software de tipo *drag and drop*², por lo cual es fácil de utilizar e iterar la estética y funcionalidad de la aplicación móvil.
- En este caso particular, en vez de la figura del cliente dentro de la metodología *Scrum*, el usuario final del electrodoméstico es quien toma este rol. Dentro del marco del desarrollo de productos destinados a un público masivo, realizar pruebas con los usuarios finales con el fin de iterar ideas y adaptarlas continuamente de acuerdo a su retroalimentación es esencial para lograr una correspondencia entre el usuario y el producto.
- La implementación del *backlog* se realizó utilizando la aplicación Google Keep [18]. Sus ventajas son su sencilla interfaz, posibilidad de incluir colaboradores y fiabilidad al ser un servicio *cloud*.

²Que permite el movimiento de objetos virtuales desplegados en pantalla utilizando el mouse u otro dispositivo similar.

Capítulo 4

Desarrollo del Trabajo

En este capítulo se detalla el desarrollo del trabajo en términos de la definición de las arquitecturas de software y hardware de acuerdo a ciertos criterios, los primeros prototipos de circuitos y el progreso hacia el diseño final, pruebas parciales de las distintas funcionalidades del electrodoméstico, el funcionamiento de la aplicación móvil y para finalizar se enuncian los principales resultados obtenidos.

4.1. Bases del Estado del Arte

Considerando el problema de crear un flujo de datos a través de internet, se requiere en primer lugar una conversión de los parámetros del mundo físico al mundo digital. Para esto, el electrodoméstico debe contar con una unidad de sensado, la cual se encarga de realizar una conversión analógica-digital de estos parámetros. En segundo lugar, se debe realizar un levantamiento de datos a la red, para esto la información debe quedar almacenada en una base de datos, la cual se encuentre siempre disponible para ser consultada por la aplicación móvil. La aplicación móvil debe disponer de esta información para poder interactuar con el usuario, haciéndolo partícipe activo del proceso de vermicompostaje.

Se comienza por definir la arquitectura de software y hardware a utilizar en base a la investigación realizada sobre el estado del arte. El hardware a utilizar se encuentra restringido por las opciones que ofrecen las tiendas comerciales en Chile, de modo tal que si se quisiera reproducir este trabajo, exista la oferta de electrónica necesaria, realizando importaciones de componentes solamente con el fin de rebajar los costos utilizando economías de escala. Con respecto al software, se utilizará de tipo *open-source*, cuyas características principal deben ser la compatibilidad, escalabilidad y facilidad de uso.

La programación escrita en todos los niveles que comprende el trabajo, se encuentra plenamente documentada en un repositorio de Github [19], con el fin de plasmar la extensibilidad, escalabilidad, difusión y robustez de la solución.

4.1.1. Microcontroladores, Procesadores y Protocolos de Comunicación

Con respecto al protocolo de comunicación, espacios compuestos por áreas pequeñas como lo son un departamento o una casa, pueden ser cubiertos utilizando una conexión *wifi* sin tener problemas con el rango de cobertura, inclusive considerando el efecto de atenuación de la señal producto de la penetración de paredes. Para implementar un sistema de comunicaciones en grandes extensiones de terreno deben usarse redes de sensores tipo Zigbee, ya que en estas áreas una cobertura utilizando *wifi* no sería factible por problemas de rango.

En cuanto a los microcontroladores y procesadores, la placa de desarrollo ESP8266 es de pequeño tamaño, cuenta con una interfaz inalámbrica dedicada a la comunicación por *wifi* y tiene un costo 10 veces menor en comparación a una Raspberry Pi en el mercado local, como muestra la Tabla 9.1. Si se utiliza un Arduino Uno como microcontrolador se requiere de hardware y conexiones adicionales para lograr la misma funcionalidad.

4.1.2. Almacenamiento de Datos

Desde el punto de vista del almacenamiento de datos, si se utiliza como servidor local el computador personal del usuario, este debe permanecer siempre encendido y no puede ser desconectado ni transportado si se requiere. Esta arquitectura presenta un punto crítico de falla. Si el almacenamiento de la información es de carácter local y el hardware que lo soporta se ve afectado por factores externos como por ejemplo un exceso de humedad que ocasione un deterioro de los componentes o alguna otra falla eléctrica, acceder los datos recolectados puede ser un problema. Esta es la razón por la cual se utiliza un servicio de base de datos *cloud* como Firebase.

4.1.3. Interfaz de Usuario y Sistema de Notificaciones

En general, dentro del contexto de la implementación de arquitecturas de internet de las cosas, los desarrollos se enfocan en la funcionalidad del sistema y no en la interfaz de usuario, lo cual es perjudicial a la hora de intentar masificar la solución. Por ejemplo, la recepción de información vía SMS es poco vinculante entre el agricultor y el cultivo de tomates, ya que únicamente se informa con una frase que el tomate está listo para ser cosechado.

Por otro lado, las aplicaciones móviles desarrolladas tienen un objetivo de visualización simple como interacción con el usuario final, sin generar entregar más información que las mismas lecturas en tiempo real. Ocurre algo similar con las aplicaciones web investigadas. En general, la interfaz gráfica es poco amigable para el usuario al cual los sistemas se encuentran destinados.

Expuesto lo anterior, se opta por desarrollar una aplicación móvil orientada a vincular al usuario con el producto que sirva no solamente como visualizador de la información si no que

también como un guía a lo largo del proceso de vermicompostaje mediante la implementación de un sistema de notificaciones utilizando alertas y otras pantallas informativas.

Una solución de internet de las cosas que busque generar un gran impacto en la experiencia entre el usuario y el proceso que lo implementa, debe contar con una aplicación móvil para *smartphone* que lo complemente. Con esto se busca involucrar al usuario en forma activa con el proceso, teniendo la oportunidad de aprender e interactuar en tiempo cuasi real con el sistema de vermicompostaje utilizando este objeto cotidiano.

4.2. Arquitectura de Internet de las Cosas

Para escoger las componentes de la arquitectura de internet de las cosas a nivel de software y hardware se privilegia por un lado el software de tipo *open-source*, con una comunidad que brinde soporte activo y comparta experiencias prácticas de implementación. Por otro lado, se prefiere el hardware de bajo costo, de rápida disponibilidad y buena *performance* de acuerdo al caso de uso.

Las arquitecturas de software y de hardware a utilizar están basadas en los trabajos de Rawat et al. [10] y Gayen et al. [12] investigados en el estado del arte con la salvedad de que no se desarrolla una aplicación nativa en Android propiamente tal, si no que se opta por elaborar un *mockup*¹ utilizando la plataforma web Thunkable a fin de acotar los tiempos de diseño e implementación.

Las arquitecturas utilizadas en ambos trabajos no dependen de un servidor local para el almacenamiento de datos, utilizando en su lugar la base de datos *cloud* Firebase. En cuanto al hardware, ambas utilizan el microcontrolador ESP8266, el cual es programable en C++ con el IDE² de Arduino y cuenta con una interfaz dedicada exclusivamente a la conexión *wifi* en 2.4[Ghz].

El electrodoméstico de ahora en adelante se referencia como Vermi. En la Figura 4.1 se muestra uno de los primeros prototipos de la Vermi.

El diagrama general de arquitectura de internet de las cosas diseñada e implementada en el presente trabajo se muestra en la Figura 4.2.

¹Modelo (opcionalmente funcional) creado para realizar evaluaciones de diseño. Estos modelos también son ampliamente utilizados con fines demostrativos y de difusión.

²*Integrated Development Environment*



Figura 4.1: Primer prototipo de la Vermi en su fase de construcción. Se observa un diseño inicial de carácter modular, pensado para realizar vermicompostaje utilizando uno o más módulos montables uno sobre otro. Fuente: Camila Martín.

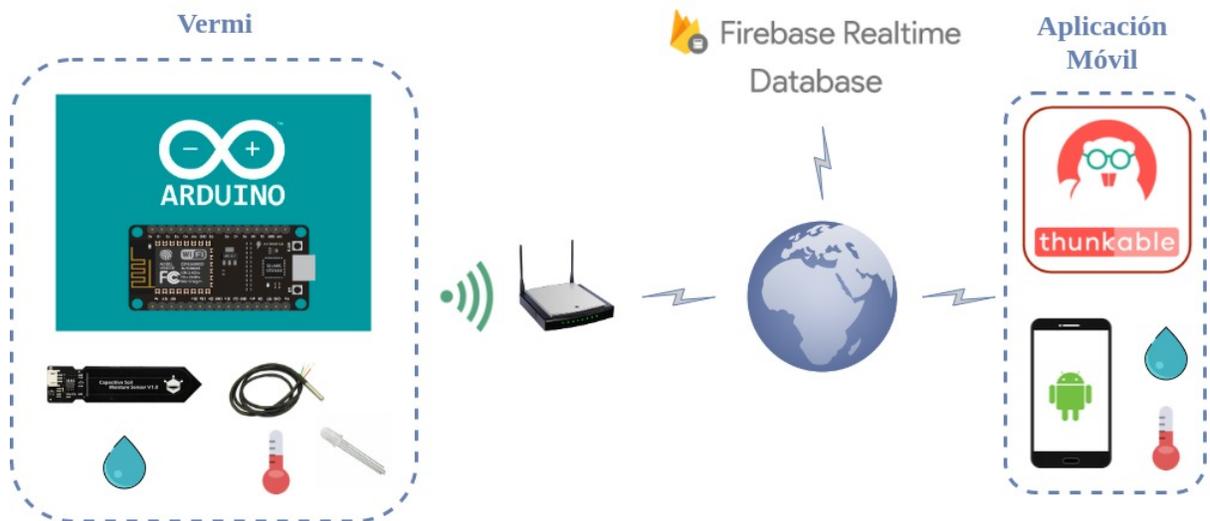


Figura 4.2: Diagrama de arquitectura de internet de las cosas desarrollado en el trabajo. A la izquierda se encuentra el electrodoméstico denominado Vermi, compuesto por el microcontrolador ESP8266, el sensor de humedad, el sensor de temperatura y el led indicador. La Vermi se comunica por *wifi* con el *router* del usuario para enviar los datos a la base de datos *cloud* Firebase. A la derecha se ilustra la aplicación móvil desarrollada en Thunkable para el sistema operativo móvil *Android*, la cual funciona como interfaz gráfica de usuario y muestra alertas en base a la información sensada.

4.3. Comportamiento del Electrodoméstico

4.3.1. Rangos Óptimos de Temperatura y Humedad

Los rangos óptimos de funcionamiento de acuerdo a la bibliografía consultada [20] y a observaciones experimentales del funcionamiento del primer prototipo de este electrodoméstico son los siguientes:

- 70 % <hum <85 %
- 7°C <temp <28,5°C

Bajo estas condiciones la reducción de los desechos orgánicos se logra en un tiempo que varía de 1 semana a 1 semana y media, dependiendo del tipo de desecho y de la cantidad de lombrices que habiten dentro del electrodoméstico. Las lombrices pueden vivir adecuadamente dentro del electrodoméstico, no escapan y son capaces de crecer y reproducirse exitosamente.

4.3.2. Diagrama de Comportamiento

A continuación se explica con detalle cada bloque del diagrama de comportamiento de la Figura 4.3, el cual ilustra el algoritmo que el electrodoméstico ejecuta internamente para funcionar.

1. **Encendido de la Vermi:** El cable de 220[V] que sale de la fuente de voltaje posee un switch de On/Off. Para encender la Vermi basta conectarla a cualquier enchufe y activar el switch.
2. **El Indicador comienza a parpadear en verde:** Primera respuesta de la interfaz del hardware al comenzar a funcionar. El parpadeo ocurre cada 0.4 segundos.
3. **La Vermi comienza a funcionar en modo *Station*:** Cualquier dispositivo conectado a una red *wifi* es considerado una *Station*, la cual establece un enlace inalámbrico a un *Access Point*, el cual provee la conexión a Internet. Un *Access Point* es identificado por su SSID (*Service Set Identifier*. Nombre visible de una red wifi.) y típicamente se requiere de una contraseña para acceder. Al par SSID y contraseña se le llama credenciales *wifi*.
4. **Hay conexión *wifi* utilizando las últimas credenciales conocidas?:** Gracias a la librería *WifiManager*, el ESP8266 es capaz de almacenar credenciales *wifi* en su memoria flash. Esta memoria es no volátil, lo cual significa que el ESP8266 recuerda las credenciales ingresadas. En caso de un corte de energía, este intenta conectarse nuevamente a la red conocida utilizando estas credenciales. Si la conexión es exitosa, sigue el paso 7.
5. **La Vermi funciona en modo *Access Point* a la espera de nuevas credenciales:** El ESP8266 crea una red propia a la cual es posible conectarse utilizando cualquier dispositivo con *wifi* en 2.4[Ghz]. Además, el ESP8266 crea un servidor web con una interfaz que permite al usuario seleccionar una red *wifi* detectada por el ESP8266 e

ingresar sus credenciales. Instrucciones más detalladas de este proceso se encuentran en Anexos - Conexión al *wifi*.

6. **Ingreso de nuevas credenciales de *wifi*:** Una vez que las nuevas credenciales de *wifi* son ingresadas por el usuario, el ESP8266 se reinicia, volviendo al punto 4.
7. **Apagar el Indicador:** Si la conexión *wifi* es exitosa, el indicador deja de parpadear.
8. **Reconectar al *wifi* y fijar el indicador en verde:** El indicador en verde fijo representa una conexión exitosa a la red *wifi*.
9. **Autenticación de acceso a la base de datos:** La API de Firebase requiere de credenciales de autenticación para permitir las operaciones relativas al almacenamiento de los datos. Estas credenciales se encuentran en el código fuente del ESP8266.
10. **Toma de medidas de temperatura y humedad:** El ESP8266 realiza la lectura de los sensores de temperatura y humedad, almacenándolas en dos variables llamadas *hum* y *temp*.
11. **Llamado al servidor de hora y fecha:** Para poder saber en qué momento el ESP8266 ha tomado las lecturas, se requiere de una marca de tiempo que contenga información de hora y fecha de acuerdo al horario de Chile. Para esto, el Servicio Hidrográfico y Oceanográfico de la Armada de Chile posee un servidor que funciona bajo el *Network Time Protocol* (NTP) [21], el cual incorpora modificaciones particulares de la hora y la fecha (como por ejemplo el cambio al horario de invierno). Esta información es guardada en una variable llamada *timestamp*.
12. **Envío de datos a la base de datos:** Las variables *hum*, *temp* y *timestamp* son enviadas a la base de datos. Si por algún motivo esta operación falla, el indicador se fija en color azul y se vuelve al paso 7.
13. **La humedad o temperatura se encuentra fuera de rango?:** En el código fuente del ESP8266 se configuran 4 umbrales, 2 para la temperatura y 2 para humedad, cuyos valores corresponden a los límites inferiores y superiores que definen los intervalos de buen funcionamiento para cada parámetro. Las lecturas se comparan con estos umbrales.
14. **La luz roja queda fija:** Si hay operación fuera de los rangos, el indicador queda fijo en rojo. Si no, el indicador sigue fijo en verde y sigue el paso 15.
15. **Esperar 1 hora:** El vermicompost es un sistema de respuesta lento, lo cual queda en evidencia en la Figura 4.16, por lo tanto no es necesario tener intervalos de muestreo frecuentes. Finalmente, se vuelve al paso 7.

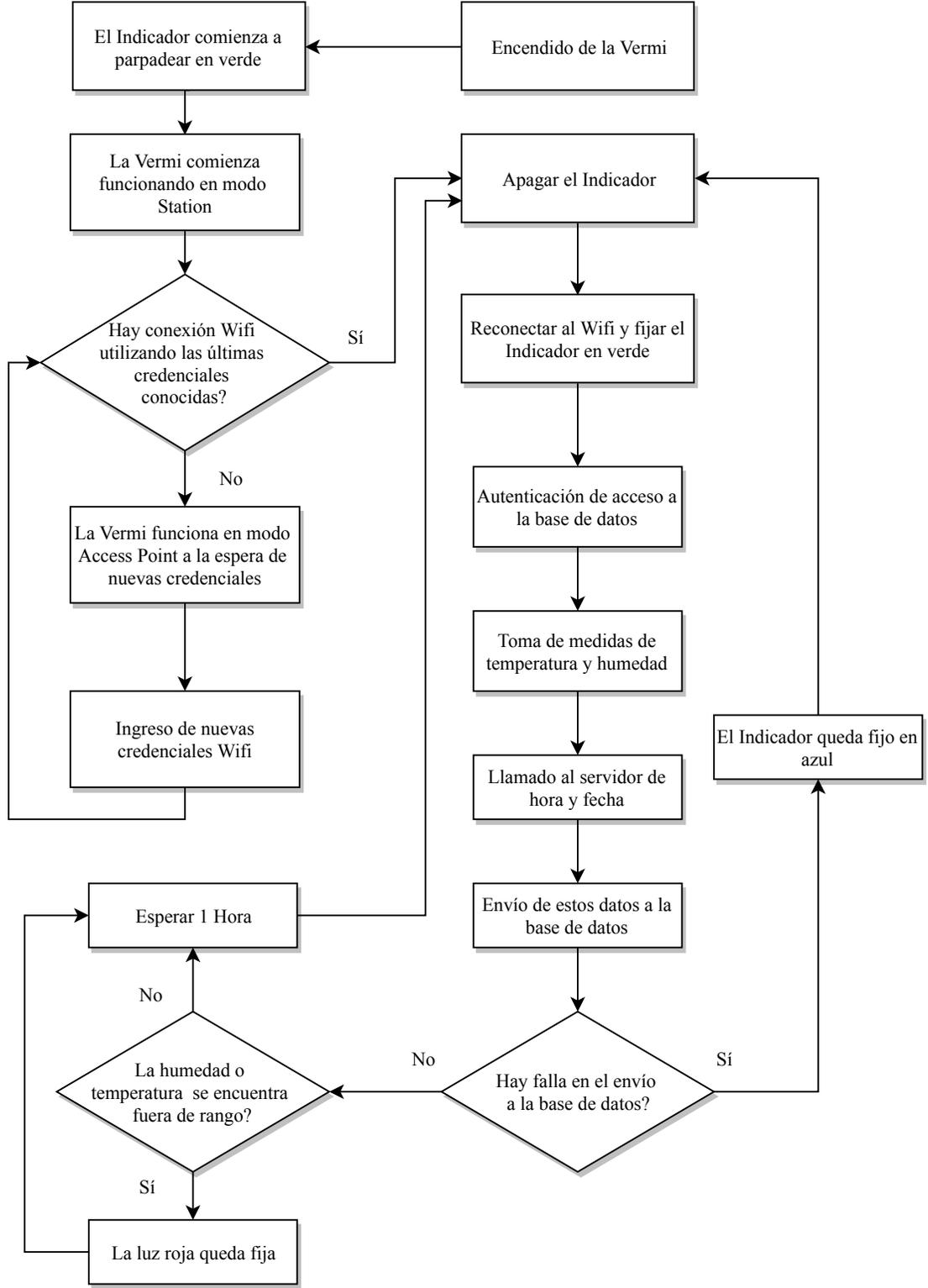


Figura 4.3: Diagrama de Comportamiento de Software y Hardware de la Vermí. El indicador hace referencia al led RGB.

4.4. Arquitectura de Software

La Figura 4.4 resume la arquitectura de software implementada.

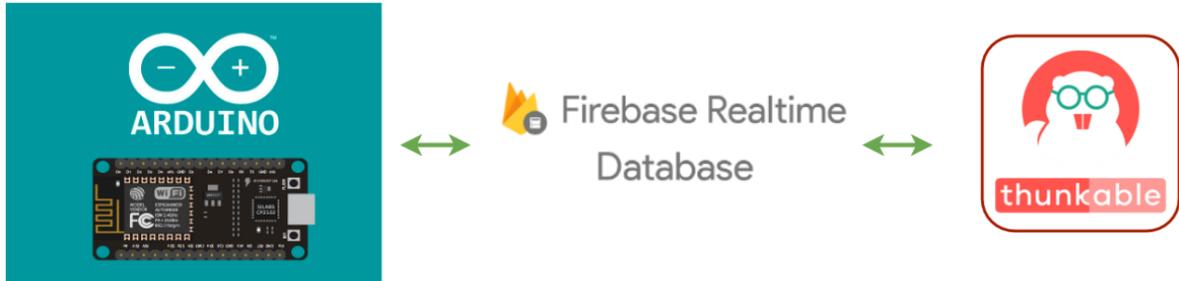


Figura 4.4: De izquierda a derecha: en primer lugar se tiene al Core de Arduino para ESP8266, el cual permite programar en forma sencilla al microcontrolador utilizando el IDE de Arduino. Es posible utilizar librerías de conectividad para hacer llamados a la API de Firebase y así ir almacenando los datos de la Vermi. Mediante esta misma API, la base de datos puede ser consultada desde la aplicación móvil. La aplicación móvil consiste en un *mockup* funcional creado en la plataforma web Thinkable.

4.4.1. Thinkable

El aspecto clave en cualquier objeto con internet de las cosas es realizar levantamiento de datos a internet y su posterior despliegue. Existe variado software para el diseño de aplicaciones móviles, a fin de acotar los tiempos de diseño e implementación, se escoge la herramienta Thinkable.

Thinkable [17] es una plataforma web que permite crear mockups funcionales de aplicaciones móviles con un buen control sobre su estética, permitiendo implementar en forma sencilla las funcionalidades básicas de una aplicación móvil, también permite programar el comportamiento de la aplicación en pseudolenguaje y utilizando el método de programación por bloques *drag and drop*. Para testear la aplicación en desarrollo, Thinkable cuenta con una aplicación móvil que permite esta visualización en tiempo real, esto es muy útil para la edición de la aplicación.

Thinkable posee variadas interfaces para el manejo de datos, en particular, es posible conectar una base de datos Firebase a la aplicación móvil mediante las *cloud variables* [22]. A nivel de la aplicación, una *cloud variable* es una referencia directa a una variable de la base de datos, cambiando su valor en la aplicación al momento de ocurrir alguna actualización en la base de datos.

4.4.2. Firebase

La elección del tipo de base de datos depende fundamentalmente del tipo de consultas que se realizan sobre los datos almacenados. En este caso, el propósito de los datos almacenados es ser desplegados y comparados con umbrales para la activación de notificaciones, entonces, para facilitar la comunicación en tiempo cuasi-real entre el hardware de la Vermi y el *smartphone* es idóneo que ambos se encuentren conectados a una base de datos centralizada [23]. Teniendo en consideración el contexto del desarrollo de productos, una base de datos no estructurada que almacena la información en formato JSON [24] provee flexibilidad en comparación a una base de datos estructurada, lo cual permite agregar nuevos campos en cualquier fase del desarrollo si es requerido. Al utilizar una base de datos *cloud* es posible acceder a la misma información desde cualquier lugar con conexión a internet y además es posible prescindir del hardware destinado a esta función.

Firebase [11] es un servicio *cloud* de Google destinado al desarrollo de aplicaciones multi-plataforma, ofreciendo soporte para los sistemas operativos móviles Android y iOS, clientes web implementados en Javascript y otros tipos de clientes implementados en C++ mediante APIs específicas para cada caso [25]. Su escalabilidad depende fundamentalmente del pago que se realice, en el desarrollo de este trabajo no hay costos asociados a cuotas de pago ya que se utiliza la cuota gratuita que provee la plataforma (*Starts for free, then pay as you go*) [26].

4.4.3. Librerías de Arduino

Los microcontroladores de Arduino son ampliamente utilizados en todo el mundo para realizar prototipos de proyectos de electrónica de bajo costo. Hoy en día existe una amplia gama de sensores compatibles con placas Arduino, tanto analógicos como digitales, y una comunidad dedicada al desarrollo de librerías que permiten un manejo más simple de los sensores y/o de diversas interfaces de conexión de estas placas de desarrollo.

Las placas Arduino se programan mediante el IDE para C++ de Arduino[27], el cual permite un manejo eficiente de las librerías, actualizaciones y posee una interfaz sencilla que permite configurar las capacidades del hardware y también permite cargar los *sketchs* (el código a ejecutar) mediante los puertos USB del computador. En la Tabla 4.1 se listan las librerías utilizadas en el desarrollo del trabajo.

Librería	Utilidad	Documentación Oficial
Arduino Core for ESP8266	Permite utilizar el IDE de Arduino para cargar sketches al microcontrolador ESP8266 mediante USB.	[28]
ESP8266Wifi	Su función principal es permitir la comunicación del ESP8266 mediante <i>wifi</i> .	[29]
FirebaseArduino	Sirve para simplificar la conexión al servicio de base de datos Firebase desde clientes Arduino. Corresponde a una abstracción completa de la API REST de Firebase implementada en C++.	[30]
ArduinoJSON	Permite el manejo de objetos JSON. Es dependencia de FirebaseArduino.h	[31]
TimeLib	Provee funciones asociadas a la sincronización temporal de las placas Arduino. En particular, permite utilizar servidores NTP.	[32]
OneWire	Permite manejar dispositivos creados por la compañía Dallas Semiconductor que se comunican por un único cable. Un ejemplo de esto es el sensor de temperatura DS18B20.	[33]
DallasTemperature	Librería orientada al manejo del sensor de temperatura DS18B20.	[34]
DNSServer	Permite la redirección de solicitudes HTTP a un servidor local a nivel del ESP8266.	[29]
ESP8266WebServer	Permite montar un servidor local a nivel del ESP8266 en el cual se despliega el portal web para configurar la conexión <i>wifi</i> .	[29]
WifiManager	Permite el despliegue del portal web y la memorización de credenciales de <i>wifi</i> en el ESP8266.	[29]
Ticker	Permite el llamado a funciones en forma periódica.	[35]

Tabla 4.1: Librerías utilizadas en el desarrollo del trabajo.

4.5. Arquitectura de Hardware

La arquitectura de hardware implementada se muestra en la Figura 4.5.

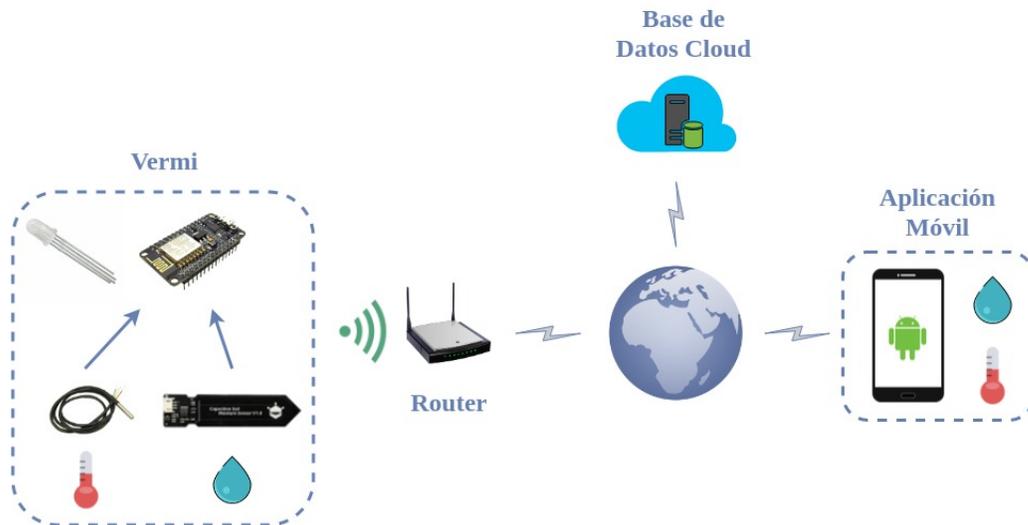


Figura 4.5: Arquitectura de hardware de la Vermi. De izquierda a derecha se observa en primer lugar el hardware que compone a la Vermi, con los sensores de temperatura, humedad y el led indicador. Luego, se puede observar que el ESP8266 se conecta al *router* del usuario mediante *wifi*, pudiendo así registrar la información en la base de datos *cloud*. Finalmente, las lecturas de temperatura y humedad se despliegan en el *smartphone* del usuario.

4.5.1. Led indicador

Para informar sobre el estado general de la Vermi al usuario, se utiliza un led RGB el cual se comporta de acuerdo a lo descrito en el diagrama de comportamiento mostrado en la Figura 4.3.

El led RGB es controlable con las salidas digitales del ESP8266, cabe destacar que el voltaje de salida máximo del ESP8266 es de 3.3[V] y 12[mA]. Esto produce buenos resultados en los colores verde y azul. Sin embargo, el color rojo brilla significativamente menos que el verde y el azul, por lo cual se construye el circuito auxiliar de la Figura 4.6 para equiparar el brillo del color rojo a los otros dos colores.

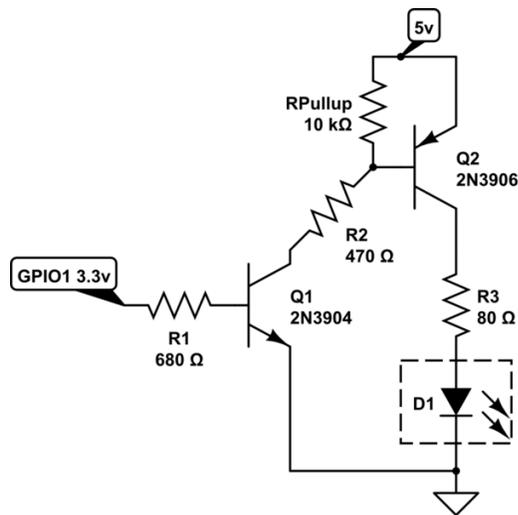


Figura 4.6: Circuito auxiliar que permite regular los 5[V] que se requieren para un adecuado brillo del led rojo utilizando una señal de control de baja potencia, en este caso de 3.3[V]. Fuente <https://electronics.stackexchange.com/questions/87967/rgb-led-brightness-levels?rq=1>

4.5.2. Sensor de Temperatura

El sensor de temperatura DS18B20 es un sensor digital [36]. Puede ser alimentado en un rango de 3.3[V] a 5[V] y cuenta con una interfaz de señal de un único cable para la transmisión de datos.

Como muestra la Figura 4.7 este sensor cuenta con una cubierta impermeable, lo cual lo hace idóneo para utilizar en condiciones de humedad alta.



Figura 4.7: Termómetro Digital DS18B20 con su cubierta impermeable. Posee 3 cables, el rojo para la alimentación, el negro para la tierra y el amarillo para la transmisión de datos.

Debido a que se trata de un sensor digital, no existe atenuación de la señal en grandes distancias. El DS18B20 posee un número serial único, por lo cual es posible transmitir a través de un único cable las medidas de múltiples DS18B20 posicionados en distintos lugares sin que exista interferencia.

4.5.3. Sensores de Humedad

Sensor de Humedad Resistivo YL-69

El sensor de humedad resistivo YL-69 es un sensor analógico [37]. Puede ser alimentado en un rango de 3.3[V] a 5[V] y está compuesto por dos electrodos entre los cuales se aplica una diferencia de tensión y una placa amplificadora de señal, como se muestra en la Figura 4.8.

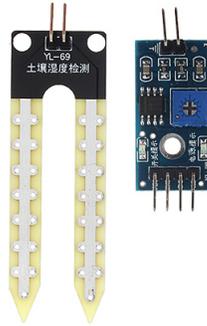


Figura 4.8: Sensor de humedad resistivo YL-69. Se observan los electrodos del sensor (izquierda) y el amplificador de señal (derecha). Fuente <https://www.fasttech.com/product/4738201-yl-69-soil-humidity-moisture-detection-sensor>

El sustrato del cual se busca obtener una medida representativa de la humedad contiene iones disueltos, por lo cual puede considerarse al sustrato como un elemento conductor. Entonces, para un sustrato de área A , largo l y conductividad σ se tiene que su resistencia $R_{sustrato}$ está dada por:

$$R_{sustrato} = \frac{l}{\sigma \cdot A} \quad (4.1)$$

Cuando el sensor está inserto en un sustrato permeable con iones disueltos, la diferencia de tensión entre los electrodos cambia en forma tal que siempre circule una corriente de 35[mA] entre ambos [37]. Luego, como el sensor es resistivo, se cumple la Ley de Ohm para una corriente constante de magnitud 35[mA]:

$$\begin{aligned} V &= I \cdot R \\ V_{YL-69} &= I_{YL-69} \cdot R_{sustrato} \\ V_{YL-69} &= 0,035 \cdot R_{sustrato} \end{aligned} \quad (4.2)$$

De acuerdo a la ecuación para la resistencia del sustrato (4.1), al aumentar la cantidad de agua en el sustrato (por ejemplo después de un riego), la conductividad del sustrato σ

aumenta lo cual disminuye el valor de la resistencia $R_{sustrato}$. Luego, de acuerdo a la ley de Ohm (4.2), si la resistencia del sustrato disminuye, el voltaje entre los electrodos del sensor V_{YL-69} debe disminuir tal que la corriente de 35[mA] se mantenga constante. Finalmente, el voltaje del sensor V_{YL-69} es amplificado y enviado al microcontrolador.

Sensor de Humedad Capacitivo

El sensor de humedad capacitivo es un sensor analógico. Puede ser alimentado en un rango de 3.3[V] a 5[V] y se trata de un condensador, el cual está compuesto por dos placas aisladas, un dieléctrico que las separa y una placa amplificadora de señal como se muestra en la Figura 4.9.



Figura 4.9: Sensor de Humedad Capacitivo V1.0 fabricado por DFRobot. Fuente <https://wecl-stem.com/shop/058-77-2031/>

El sustrato del cual se busca obtener una medida representativa de la humedad contiene iones disueltos, los cuales provocan variaciones en la capacitancia del sensor C_{sensor} [38]. Si se considera que el sensor tiene un dieléctrico con permitividad ϵ el cual está inserto entremedio de las placas de área A , las cuales están separadas en una distancia d , su capacitancia está dada por:

$$C_{sensor} = \epsilon \cdot \frac{A}{d} \quad (4.3)$$

Cuando el sensor está inserto en un sustrato permeable con iones disueltos, la diferencia de tensión entre sus placas V_{sensor} cambia en forma tal que la carga Q almacenada por el condensador se mantenga siempre constante, es decir:

$$Q = C \cdot V \quad (4.4)$$

$$Q = C_{sensor} \cdot V_{sensor} \quad (4.5)$$

De acuerdo a la ecuación para la capacitancia del sensor (4.3), al aumentar la cantidad de agua en el sustrato, la permitividad ϵ aumenta lo cual aumenta la capacitancia del sensor. Luego, de acuerdo a la ecuación del condensador (4.5), si la capacitancia del sensor aumenta, el voltaje del sensor V_{sensor} debe disminuir tal que la carga almacenada Q se mantenga constante. Finalmente, el voltaje del sensor V_{sensor} es amplificado y enviado al microcontrolador.

4.5.4. Microcontrolador ESP8266

El ESP8266EX es un microchip de bajo costo desarrollado por la compañía china *Espressif Systems* [39] cuya característica principal es su capacidad de comunicarse mediante el protocolo TCP/IP. Este procesador es de 32 bits y su frecuencia máxima de reloj es de 160[Mhz].

El ESP8266MOD, también conocido como *ESP12-E Module*, es un módulo basado en el microchip ESP8266EX y está diseñado para ser una solución de conectividad robusta frente a condiciones externas, operando correctamente en un rango de temperaturas que va desde los -40 hasta los 125°C. El ESP8266MOD posee puertos periféricos digitales estándar, una antena Wifi de 2.4[Ghz], un led azul embebido y un amplificador de potencia de +25dBm. Su tamaño compacto, bajo precio y compatibilidad lo hacen un microchip ideal para prototipar proyectos de internet de las cosas, como se muestra en la Figura 4.10.



Figura 4.10: ESP8266MOD, también conocido como ESP12-E. Se observa la antena Wifi embebida. Fuente <https://es.aliexpress.com/item/32324005820.html>

Existen varias placas de desarrollo que utilizan al ESP8266MOD como su núcleo [40], ya que este hardware es de tipo *open-source*. Dentro de estas opciones, el NodeMCU 1.0 es una de las soluciones más baratas y completamente integradas del ESP8266MOD. La placa denominada NodeMCU 1.0 agrega al ESP8266MOD reguladores de voltaje, botones embebidos de *reset* y *flash* y un puerto micro USB para comunicación serial con el computador, como se muestra en la Figura 2.4, lo cual lo hace fácil de programar.

Los pines del NodeMCU 1.0 se muestra en la Figura 4.11.

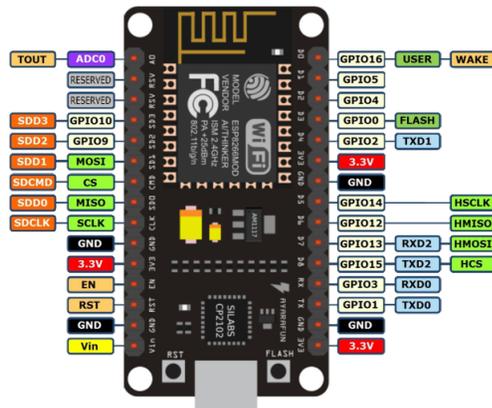


Figura 4.11: Pines del NodeMCU 1.0. Se aprecia el ESP8266MOD con su antena *wifi*, las funciones de sus pines, el puerto micro USB y los botones de *flash* y *reset*. Fuente <https://iotbytes.wordpress.com/nodemcu-pinout/>

4.5.5. Fuente de Alimentación y Consumo de Corriente

El artefacto inteligente en desarrollo cae dentro de la categoría de un electrodoméstico de línea blanca (como las lavadoras, refrigeradores o microondas). A fin de concebir un artefacto lo más completo posible, es deseable que su estética sea similar a otros objetos de su categoría. Dicho lo anterior, se decide cambiar el transformador de teléfono Android por una fuente de voltaje de propósito general con entrada de 220[Vac] como la mostrada en la Figura 4.12.

Consumo de Corriente

A fin de conocer el consumo de corriente del circuito de la Vermi, se utiliza un multímetro digital en modo amperímetro conectado en serie con la Vermi. Para ello, se reemplaza el fusible de la fuente de voltaje por las puntas de prueba del multímetro y se obtiene el valor RMS de la corriente alterna que consume el aparato, mostrado en la Tabla 4.2.

Consumos	I_{rms} [A]
Fuente de Voltaje	0.65
Circuito Completo	1.0

Tabla 4.2: Consumos de corriente de la Vermi 2.0

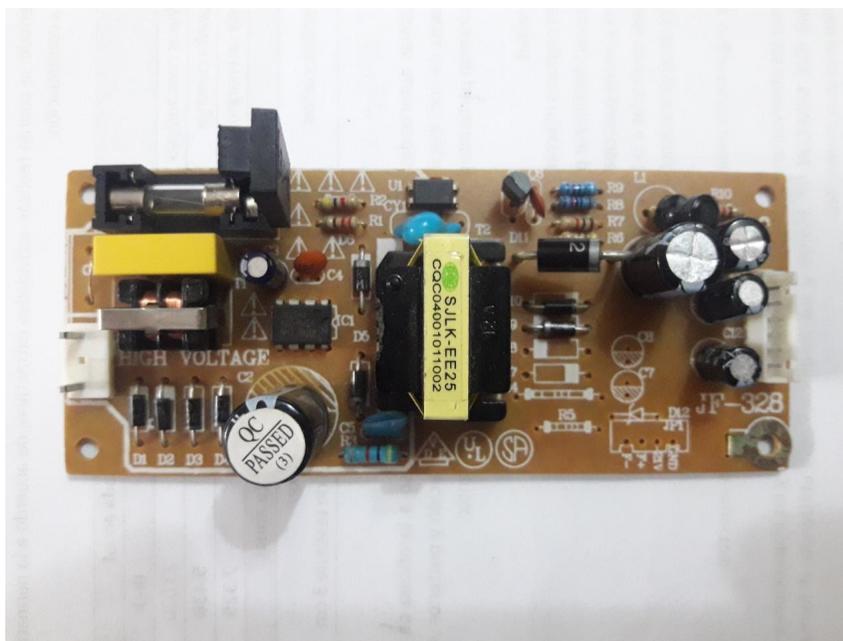


Figura 4.12: Fuente de Voltaje de propósito general con entrada de 220[Vac] y salidas de voltajes DC de 5[V] y 12[V]. Esta fuente cuenta con un fusible de 2[A] como protección. Esta fuente es capaz de alimentar cargas inductivas sin problemas de inestabilidad en los voltajes de alimentación.

4.6. Obtención del Porcentaje de Humedad Relativa del Vermicompost

Para obtener la humedad relativa en forma de porcentaje, se realiza un ajuste lineal de acuerdo a la documentación que provee DFRobot [41] para el sensor de humedad capacitivo, obteniéndose la Figura 4.20. Para realizar el ajuste lineal se procede como sigue:

1. Se registra el valor obtenido por la función *analogRead()* cuando el sensor de humedad está expuesto al aire. Este valor llamado *highestVal* corresponde al caso límite en el cual el sustrato se encuentra completamente seco, es decir, con una humedad del 0% (*lowerPerc*).
2. Se registra el valor obtenido por la función *analogRead()* cuando el sensor de humedad está inserto dentro de un vaso de agua. Este valor llamado *lowerVal* corresponde al caso límite en el cual el sustrato se encuentra completamente húmedo, es decir, con una humedad del 100% (*highestPerc*).

Entonces, utilizando la fórmula principal de la ecuación de la recta es posible obtener el % de humedad en función del valor que arroja la función *analogRead()*. Primero, calculando la pendiente:

$$\begin{aligned}
m &= \frac{\text{highestVal} - \text{lowerVal}}{\text{lowerPerc} - \text{highestPerc}} \\
m &= \frac{780 - 360}{0 - 100} \\
m &= -4,2
\end{aligned} \tag{4.6}$$

Luego, de la fórmula principal se tiene que:

$$\begin{aligned}
\text{percentage} &= m \cdot (\text{analogRead}() - \text{lowerVal}) + \text{highestPerc} \\
\text{percentage} &= -4,2 \cdot (\text{analogRead}() - 360) + 100 \\
\text{percentage} &= -4,2 \cdot \text{analogRead}() + 1512 + 100 \\
\text{percentage} &= -4,2 \cdot \text{analogRead}() + 1612
\end{aligned} \tag{4.7}$$

En la ecuación (4.7) *percentage* corresponde a la humedad relativa del vermicompost en forma de porcentaje. Este cálculo se lleva a cabo en el microcontrolador NodeMCU 1.0.

4.7. Diseño e Implementación del Electrodoméstico

En lo sucesivo se detalla todo el trabajo relativo al diseño e implementación del electrodoméstico en términos de los prototipos desarrollados de cara a la obtención del prototipo funcional.

4.7.1. La Vermi 1.0 - Primer Prototipo

En la Figura 4.13 se muestra el primer prototipo del electrodoméstico creado, el cual recibe el nombre de Vermi 1.0.

Para esta primera versión se escogen componentes ya utilizados previamente por el autor en otros proyectos. Este circuito tiene por objetivo validar el traspaso de datos a la base de datos, para lo cual es necesario:

- Construir el módulo de sensado incluyendo los siguientes componentes:
 - Sensor de Humedad analógico YL-69
 - Sensor de Temperatura digital DS18B20
 - ESP8266
 - Led indicador de encendido
 - Transformador de teléfono Android (5[V], 1[A])

- Switch On/Off
- Montar en la Vermi 1.0 el módulo de sensado
- Verificar que las mediciones sean fidedignas
- Lograr almacenar los datos históricamente en la base de datos



Figura 4.13: Primer prototipo de la Vermi instalado en una cocina. Está construido en terciado de pino insigne, un material barato y fácil de trabajar en una CNC. Tiene un sellado interior hecho con aceite de linaza, protegiendo la madera de la humedad. Fuente: Camila Martín.

En la Figura 4.14 se muestra el módulo de sensado de la Vermi 1.0.

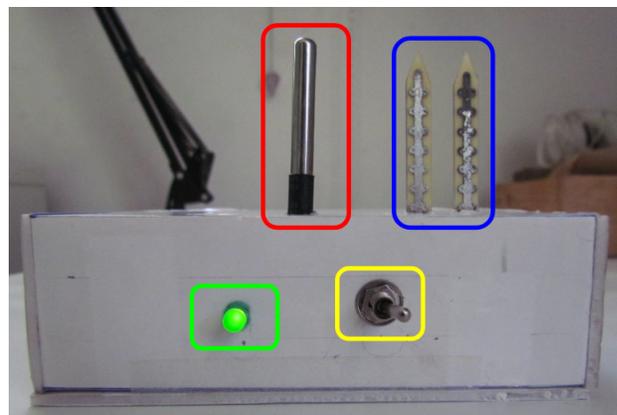


Figura 4.14: Módulo de sensado de la Vermi 1.0. Se tiene el sensor de temperatura DS18B20, el sensor de humedad YL-69, el led indicador de encendido y el Switch On/Off. Dentro de la caja fabricada con cartón pluma se encuentra el ESP8266, por la parte trasera sale el cable del cargador de teléfono Android.

Resultados

Se logra una conexión exitosa a la base de datos, pudiendo visualizar en tiempo cuasi-real el ingreso de nuevos datos en la consola que Firebase provee para este fin, lo cual se ilustra en la Figura 4.15:



Figura 4.15: Estructura del documento JSON en la base de datos Firebase que guarda los datos históricos. Posee 3 campos que se actualizan en tiempo cuasi-real llamados: *hum*, *temp* y *timestamp* y 1 campo secundario llamado *logs*, el cual almacena los 3 campos principales separados por un espacio a lo largo del tiempo.

Con el fin de analizar los datos capturados por el módulo de sensado, se programó un script de Python llamado *drawResults.py* el cual se encuentra disponible en el repositorio de GitHub del autor [19]. Se obtuvieron las curvas mostradas en las Figuras 4.16 y 4.17.

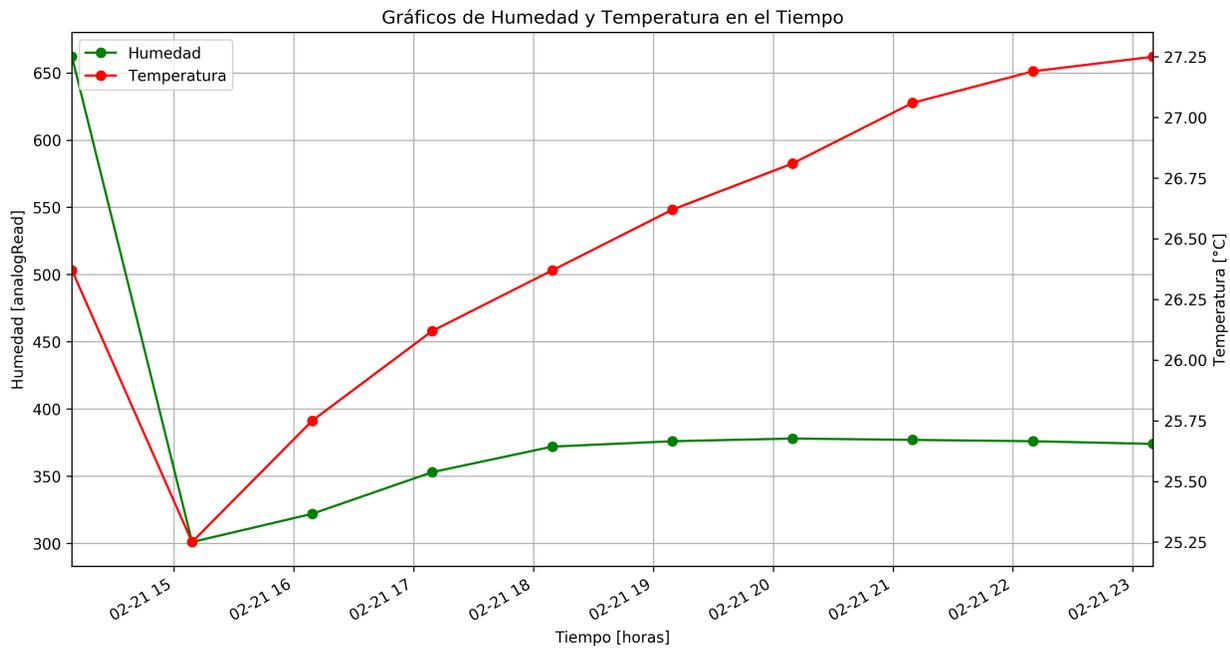


Figura 4.16: Gráfico de Humedad y Temperatura en el tiempo (horas) para el día 21 de Febrero (02-21) del 2019 construido con 10 muestras tomadas en la Vermi 1.0. Cada marca en el eje del tiempo corresponde a las horas del día a partir de las 3 de la tarde. El valor representativo de la Humedad está dado por el sensor YL-69. Los puntos graficados son datos reales unidos por una línea de tendencia.

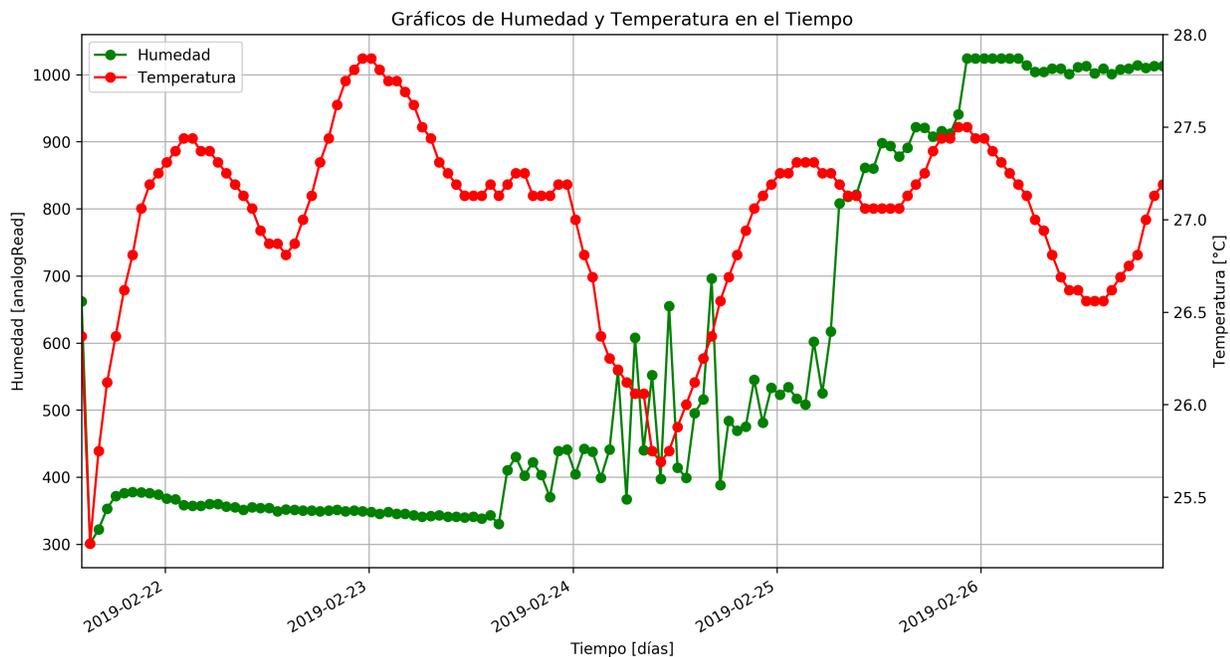


Figura 4.17: Gráfico de Humedad y Temperatura en el tiempo (días) para el mes de Febrero de 2019 construido con 128 muestras tomadas en la Vermi 1.0. Cada marca en el eje del tiempo corresponde a las 00:00 horas de día correspondiente. El valor representativo de la Humedad está dado por el sensor YL-69. Los puntos graficados son datos reales unidos por una línea de tendencia.

Análisis

De la Figura 4.16, se desprende que la curva de humedad presenta un comportamiento estable, el primer punto es un *outlier*³ relacionado a la instalación del hardware. La estabilización de esta curva ocurre a 4 horas desde la instalación. Por otro lado, la curva de temperatura presenta una estabilización veloz, del orden de 1 hora.

En la Figura 4.17 la curva de temperatura contiene suficiente detalle como para observar variaciones durante el día y la noche, los valles de esta curva corresponden a horas de la madrugada, mientras que las crestas corresponden a las calurosas horas de la tarde en un día de verano. Se observa que los peaks de temperatura ocurren cercanos a las 00:00 horas de cada día; esto tiene su origen en 2 causas interesantes:

- La fermentación de los desechos orgánicos aumenta la temperatura del vermicompost.
- La cantidad de agua retiene el calor del día y lo libera lentamente durante la noche.

El día 24 se agregan desechos congelados a la Vermi a fin de disminuir la temperatura del vermicompost, observándose un descenso de la temperatura de aproximadamente 2 grados y variaciones difusas en la curva de humedad.

Del 24 al 26, la curva de humedad no presenta un comportamiento estable como se esperaría y finalmente se estabiliza en valores por sobre 1000, esto para el sensor de humedad significa que el vermicompost está totalmente seco. Dado el comportamiento anormal de la curva de humedad descrito anteriormente, se revisó el sensor de humedad del módulo de sensado, lo cual se ilustra en la Figura 4.18.



Figura 4.18: Sensores de Humedad YL-69. Al centro se observa un sensor YL-69 nuevo y a ambos lados otros dos utilizados en la Vermi 1.0, los cuales presentan signos de corrosión.

De la Figura 4.18 se puede concluir que las lecturas anormales del sensor de humedad YL-69 se explican por la corrosión de sus electrodos provocada por la humedad del vermicompost. Se requiere del sensor de humedad capacitivo, el cual tiene un aislante protector anticorrosión.

En la Figura 4.19 se muestran los resultados obtenidos con el sensor capacitivo y la función

³Dato que difiere significativamente de los demás.

de lectura de sensores analógicos de Arduino `analogRead()` y en la Figura 4.20 se grafican los mismos datos incluyendo el ajuste lineal para la obtención del porcentaje de humedad relativa del vermicompost. Es posible notar que la humedad muestra un comportamiento estable como se espera, luego, se corrobora un correcto funcionamiento del sensor capacitivo de DFRobot. En ambas Figuras se observa que la línea de tendencia es en algunos períodos de tiempo una recta, esto es debido a que durante esos períodos de tiempo la Vermi estuvo *offline* producto de una conexión inestable al *router*. Esta inestabilidad en la conexión se debe a la calidad del *router* de Camila, en el cual se realizaron las pruebas. Tras cambiar de *router* y realizar algunas modificaciones al algoritmo de la Vermi 1.0, esta situación no volvió a repetirse.

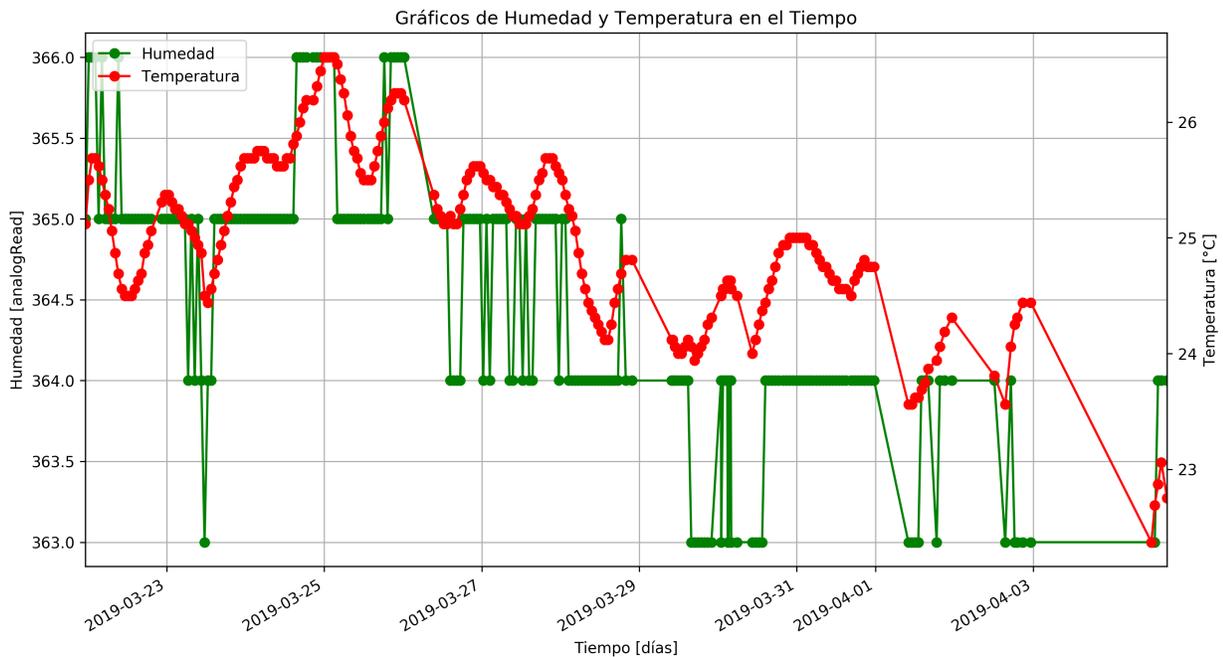


Figura 4.19: Gráfico de Humedad y Temperatura en el tiempo (días) para el mes de Marzo de 2019 construido con 241 muestras tomadas en la Vermi 1.0. El valor representativo de la Humedad está dado por el sensor de humedad capacitivo de DFRobot. Los puntos graficados son datos reales unidos por una línea de tendencia.

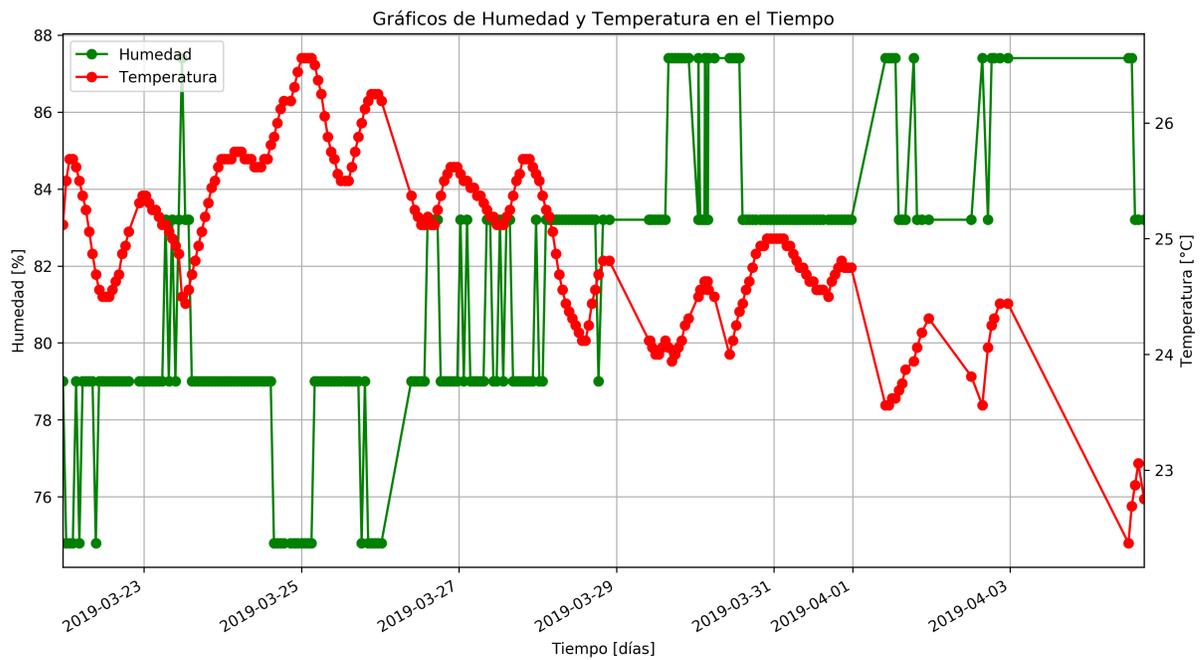


Figura 4.20: Gráfico de Humedad y Temperatura en el tiempo (días) para el mes de Marzo de 2019 construido con 241 muestras tomadas en la Vermi 1.0. El valor representativo de la Humedad está dado por la medida del sensor de humedad capacitivo de DFRobot incluyendo el ajuste lineal para la obtención del porcentaje de humedad relativa del vermicompost. Los puntos graficados son datos reales unidos por una línea de tendencia.

4.7.2. La Vermi 2.0 - Segundo Prototipo

El segundo prototipo del electrodoméstico recibe el nombre de Vermi 2.0 y se muestra en la Figura 4.21. Tiene por objetivos principales el implementar una interfaz que permita al usuario final de la Vermi conectar el objeto a su red local en forma sencilla e informar sobre el estado general de la Vermi utilizando el led indicador.



Figura 4.21: Segundo prototipo de la Vermi (Vermi 2.0). Corresponde a una maqueta construida con couché forrado que cuenta con un sistema de 2 cajones removibles que permiten la cosecha del hummus sólido y líquido. Fuente: Camila Martín.

En pos del primer objetivo, se implementa un servidor web local en el ESP8266 que permite el ingreso y almacenamiento de las credenciales de red del usuario. A este servidor web se puede acceder desde cualquier dispositivo con interfaz *wifi*, un ejemplo de esto se muestra en la Figura 4.22. Los pasos a seguir para el ingreso de credenciales se detallan en Anexos - Conexión al Wifi.

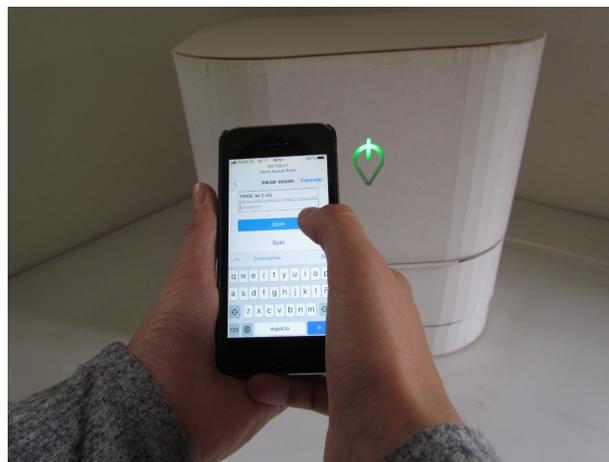


Figura 4.22: Conexión al servidor web del ESP8266 desde un iPhone para entregar credenciales de la red local del usuario final. Fuente: Camila Martín.

Para alcanzar el segundo objetivo, la Vermi 2.0 tiene embebido un led RGB mostrado en la Figura 4.22 a modo de interfaz de usuario, el cual va respondiendo a medida del

avance de procesos internos que se ejecutan a nivel de la Vermi de acuerdo al diagrama de comportamiento ilustrado en la Figura 4.3.

Los sensores se disponen dentro de la Vermi 2.0 como muestra la Figura 4.23.



Figura 4.23: Sensores de Temperatura (izquierda) y Humedad (derecha) instalados en la Vermi 2.0. Se observa una malla diseñada para permitir un arneado del hummus. Fuente: Camila Martín.

El circuito se dispone en la parte trasera de la Vermi 2.0 como muestra la Figura 4.24. El led RGB es cableado hacia adelante por una canaleta que va desde atrás hacia adelante, como muestra la Figura 4.25.

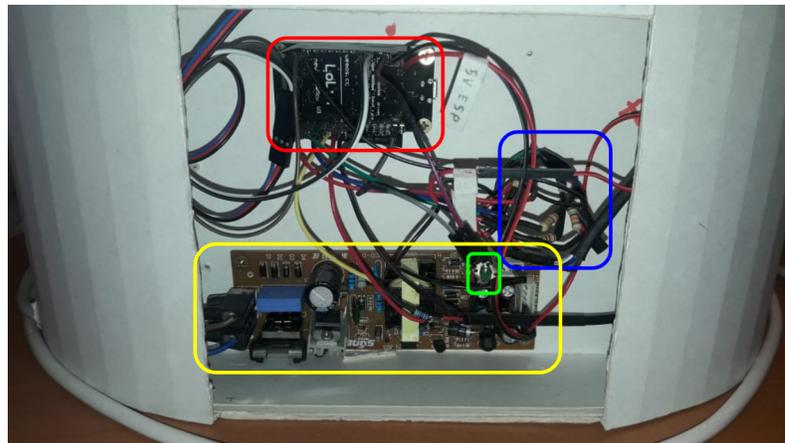


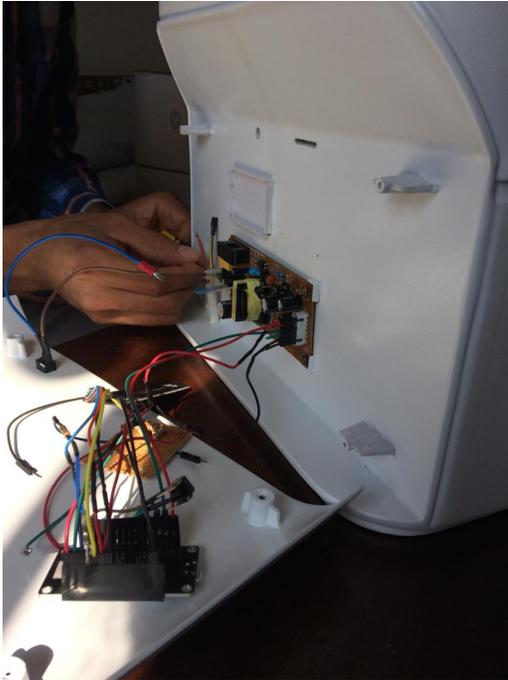
Figura 4.24: Parte trasera de la Vermi 2.0, se muestra el circuito situado por dentro con sus componentes afirmados por tornillos para madera de 5mm. Este espacio es cubierto por una tapa. Se tiene el microcontrolador **ESP8266**, un **botón de reset** accesible, la **fuentes de voltaje** que alimenta a toda la circuitería y el **circuito auxiliar del led RGB**.



Figura 4.25: Vermi 2.0 sin su carcasa externa, se muestra el cableado del led RGB y el propio led RGB. La carcasa externa tiene un logo transparente hecho con mica translúcida que queda situado justo por delante del led. Fuente: Camila Martín.

4.7.3. Versión Final

Para mostrar la versión final del electrodoméstico creado se fabrican dos prototipos, uno funcional con el cual los usuarios pueden interactuar físicamente y otro visual, el cual permite observar la estética final de este artefacto. El prototipo funcional se muestra en la Figura 4.26 y el prototipo visual se muestra en la Figura 4.27. Detalles a nivel de modelo 3D se muestran en la Figura 4.28.



(a) Instalación del circuito en el prototipo funcional. Los componentes eléctricos se fijan con tornillos para madera de 3 mm.



(b) Bandeja recolectora de hummus sólido con arnero.



(c) Bandeja recolectora de hummus líquido.



(d) Led indicador encendido en rojo frente a la condición de humedad baja.

Figura 4.26: Prototipo funcional del electrodoméstico. Este prototipo contiene vermicompost en su interior, se conecta a la red local del usuario mediante wifi solicitando sus credenciales, envía datos de temperatura y humedad cada 1 hora a la base de datos y activa su led indicador frente a condiciones de operación fuera de rango. Fuente: Camila Martín.



Figura 4.27: Prototipo visual del electrodoméstico instalado en una cocina. Sólo para fines estéticos. Fuente: Camila Martín.



(a) Vista frontal.



(b) Vista trasera.



(c) Modelo 3D general.

Figura 4.28: Modelos 3D de la versión final del electrodoméstico. Fuente: Camila Martín.

4.8. Aplicación Móvil

La aplicación se encuentra diseñada para ser mostrada en un teléfono SM-J710MN (Samsung Galaxy J7 2016). También se muestra correctamente en el SM-J700M (Samsung Galaxy J7 2015) ya que ambos teléfonos tienen resoluciones de pantalla similares.

Para ejecutar la aplicación en los teléfonos antes mencionados y de acuerdo a los alcances del trabajo, se debe descargar el visor de aplicaciones de Thunkable, acceder con la cuenta de Thunkable y seleccionar la aplicación a visualizar en el menú.

La aplicación tiene 5 pantallas principales, cada una con diferentes propósitos y a las cuales es posible acceder mediante una barra de navegación situada en la parte inferior de la aplicación. Las pantallas junto con su descripción y referencia se muestran en la Tabla 4.3. El contenido visual fue creado en su completitud por Camila Martín.

Las pantallas *Home* y *Dashboard* son dinámicas (Figuras 4.31 y 4.33), es decir, varían su contenido de acuerdo a la información que contiene la base de datos en cierto instante. Por otro lado, las pantallas *Problems*, *Information* y *Community* son estáticas (Figuras 4.34, 4.36 y 4.38) sirviendo para desplegar información enfocada en la educación del usuario.

Pantalla	Descripción	Figura
<i>Home</i>	Primera pantalla que se muestra cuando se abre la aplicación. Aquí se informa del estado general de la Vermi mediante el visor que se observa en la parte superior, en la parte inferior se despliega un dato curioso relativo al vermicompostaje o una notificación según corresponda.	[4.31]
<i>Dashboard</i>	En la segunda pantalla se muestra el <i>Dashboard</i> de la aplicación, el cual funciona como un visor en tiempo cuasi-real de la humedad y temperatura del vermicompost. Las medidas se actualizan cada 1 hora desde el circuito y la aplicación responde frente a estos cambios periódicos en la base de datos.	[4.33]
<i>Problems</i>	Se muestran 6 botones con los problemas más típicos que se presentan al realizar el proceso de vermicompostaje, se pueden presionar para acceder a información más detallada sobre como abordar cada uno de ellos.	[4.34]
<i>Info</i>	Se muestran 3 botones deslizables que conducen a otras pantallas con información sobre las lombrices, los beneficios que el abono producido tiene para el suelo y los desechos permitidos y no permitidos que pueden agregarse a la Vermi.	[4.36]
<i>Community</i>	Se muestran dos botones. A la izquierda se accede a la página web del proyecto y a la derecha, como proyección futura, se encuentra el acceso a lo que será la comunidad, en la cual se podrá intercambiar lombrices, abono y experiencias.	[4.38]

Tabla 4.3: Tabla con el nombre de las 5 pantallas principales que componen la aplicación móvil junto con su descripción y la referencia a la Figura correspondiente.

4.8.1. Sistema de Notificaciones

Las notificaciones comunican al usuario cuando la Vermi se encuentra fuera de los rangos adecuados de funcionamiento, ya sean de temperatura o de humedad, así como también entregarle información útil para que proceda a resolver el problema por sí mismo.

Frente a operación fuera de los rangos, las notificaciones se despliegan de dos formas:

1. Mediante una alerta dentro de la aplicación, como muestra la Figura 4.30.
2. Mediante un espacio dedicado a notificaciones en la pantalla *Home*, como muestra la Figura 4.31.

Para conocer los rangos fuera de los cuales deben mostrarse estas notificaciones, estos se encuentran almacenados como valores fijos en la base de datos, tal como muestra la Figura 4.29. Tanto la alerta como la notificación contienen el mismo texto y ambas se despliegan frente a cambios en los parámetros *hum* y *temp* de la base de datos, conformando así un sistema de notificaciones y alertas en tiempo cuasi-real.

La comunicación con la base de datos se realiza mediante las *cloud variables*. A nivel de la aplicación, una *cloud variable* es una referencia directa a una variable de la base de datos. Una *cloud variables* cambia su valor al momento de ocurrir alguna actualización en la base de datos. En este caso, se utilizan las siguientes *cloud variables*:

- *hum*: Se actualiza periódicamente con el valor de la humedad del vermicompost.
- *maxSoilHumidity*: Valor fijo que define el umbral superior para el rango de humedad adecuado.
- *minSoilHumidity*: Valor fijo que define el umbral inferior para el rango de humedad adecuado.
- *maxSoilTemp*: Valor fijo que define el umbral superior para el rango de temperatura adecuado.
- *minSoilTemp*: Valor fijo que define el umbral inferior para el rango de temperatura adecuado.
- *temp*: Se actualiza periódicamente con el valor de la temperatura del vermicompost.
- *timestamp*: Se actualiza periódicamente con datos de hora y fecha de las mediciones.

La aplicación se encuentra programada internamente en pseudolenguaje, el cual permite acceder a las *cloud variables* para realizar acciones. Entonces, programando en este pseudolenguaje es posible desplegar las alertas comparando los valores umbrales con los valores actuales para ambos parámetros del vermicompost.

sustentavermiv2

```
hum: 87.4 ×
logs
maxSoilHumidity: 85
maxSoilTemp: 28.5
minSoilHumidity: 70
minSoilTemp: 7
temp: 13.44
timestamp: "14:34:58 e1 24/6/2019"
```

Figura 4.29: Rangos adecuados de funcionamiento para humedad y temperatura almacenados en la base de datos bajo los nombres *maxSoilHumidity*, *minSoilHumidity*, *maxSoilTemp* y *minSoilTemp*.

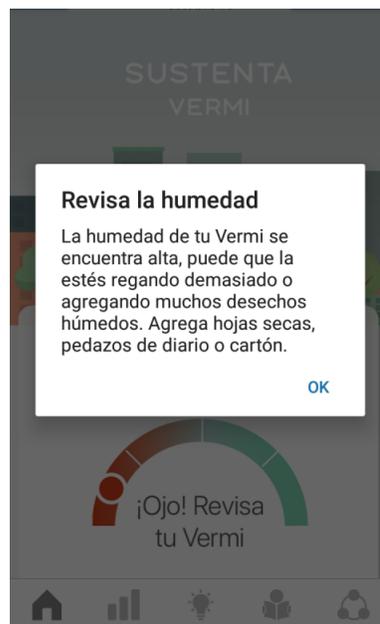
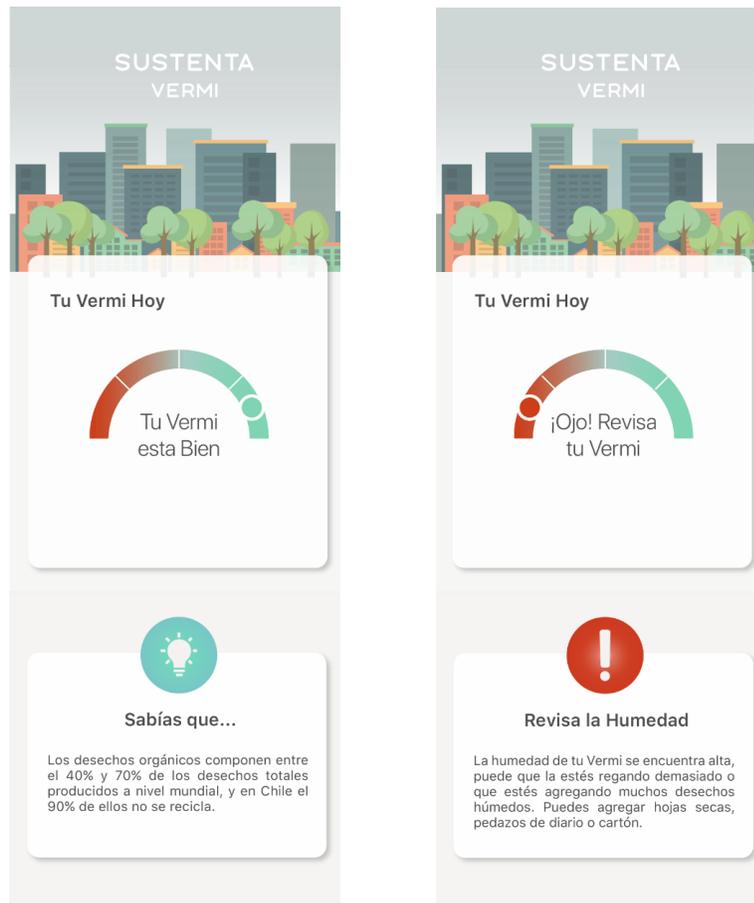


Figura 4.30: Alerta desplegada dentro de la aplicación para el caso en el cual la humedad se encuentra alta.

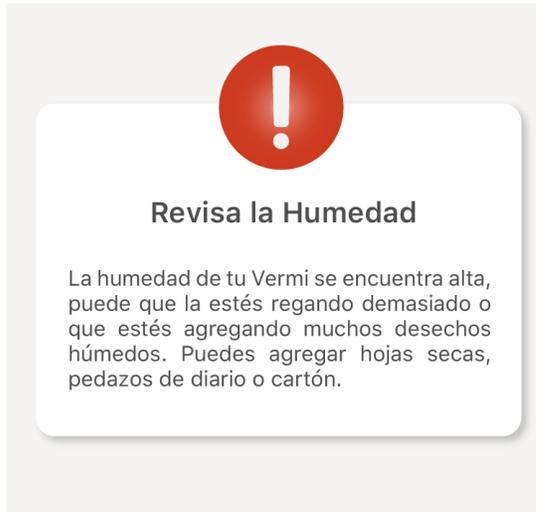
4.8.2. Pantallas



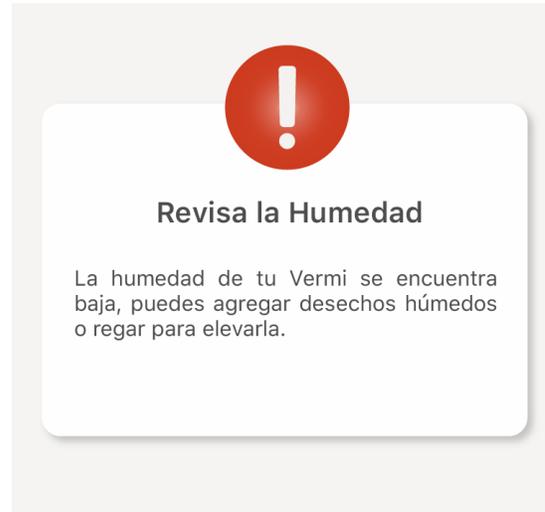
(a) La Vermi se encuentra dentro de rangos adecuados de funcionamiento. Se muestra un dato curioso en la parte inferior.

(b) La Vermi se encuentra fuera de rangos adecuados de funcionamiento. Se muestra una notificación en la parte inferior según sea el caso, en esta subfigura la humedad se encuentra alta por ejemplo.

Figura 4.31: *Home* es la primera pantalla que se muestra cuando se abre la aplicación. Aquí se informa del estado general de la Vermi mediante el visor que se observa en la parte superior, en la parte inferior se despliega un dato curioso relativo al vermicompostaje o una notificación según corresponda.



(a) Notificación de humedad alta.



(b) Notificación de humedad baja.



(c) Notificación de temperatura alta.



(d) Notificación de temperatura baja.

Figura 4.32: Notificaciones de funcionamiento fuera de rango según cada caso. Se tienen 4 escenarios posibles definidos por umbrales inferiores y superiores para la humedad y temperatura respectivamente.



Figura 4.33: En la segunda pantalla se muestra el *Dashboard* de la aplicación, el cual funciona como un visor en tiempo cuasi-real de la humedad y temperatura del vermicompost. Las medidas se actualizan cada 1 hora desde el circuito y la aplicación responde frente a estos cambios periódicos en la base de datos.

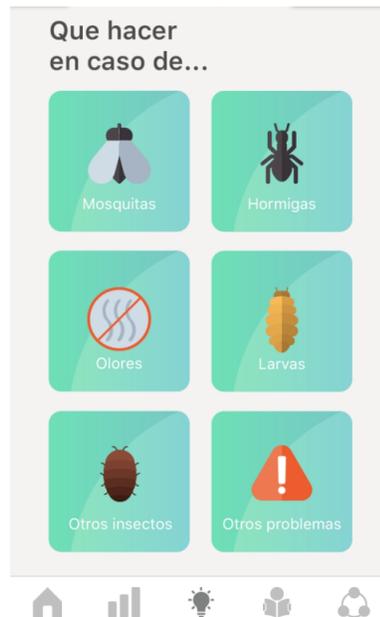


Figura 4.34: En la pantalla *Problems* se muestran 6 botones con los problemas más típicos que se presentan al realizar el proceso de vermicompostaje, se pueden presionar para acceder a información más detallada sobre como abordar cada uno de ellos.



(a) Mosquitas



(b) Hormigas



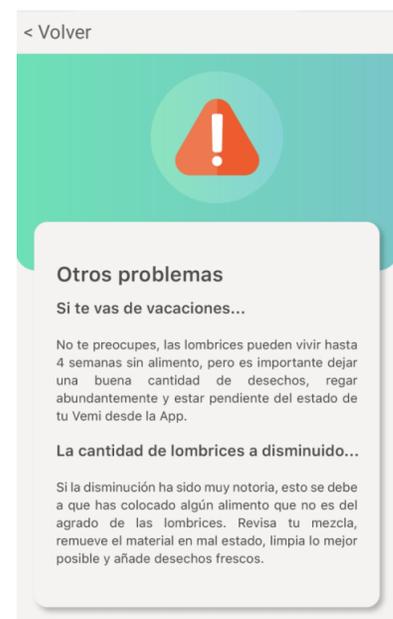
(c) Malos Olores



(d) Larvas



(e) Otros insectos



(f) Otros problemas

Figura 4.35: Información detallada sobre los problemas más típicos que pueden ocurrir. Al presionar <Volver, se vuelve a la pantalla *Problems*. Para una mejor visualización de la figura recurrir a la versión digital de este documento.

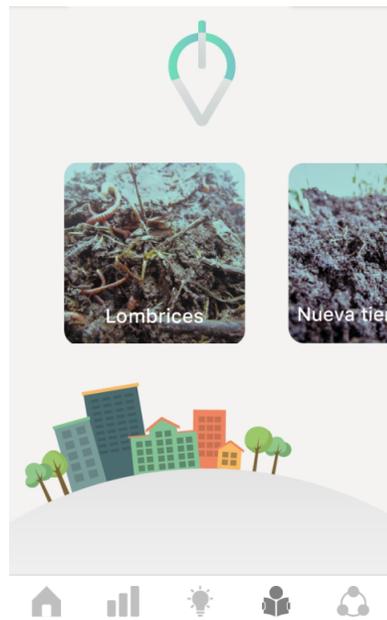


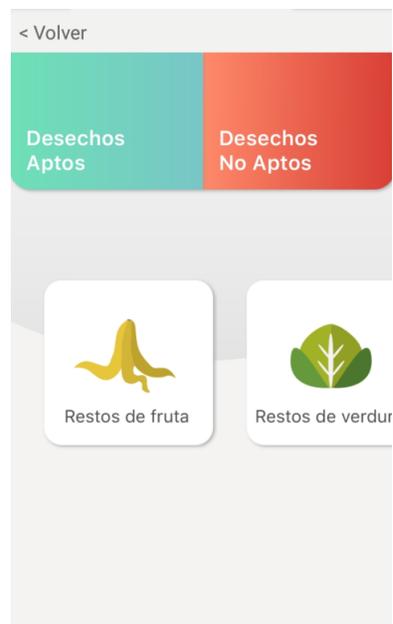
Figura 4.36: En la pantalla *Information* se muestran 3 botones deslizables que conducen a otras pantallas con información sobre las lombrices, los beneficios que el abono producido tiene para el suelo y los desechos permitidos y no permitidos que pueden agregarse a la Vermi.



(a) Lombrices



(b) Importancia para el suelo



(c) Desechos aptos y no aptos

Figura 4.37: Información general que muestran los 3 botones de la pantalla *Information*. Al presionar <Volver, se vuelve a la pantalla *Information*.



Figura 4.38: En la pantalla *Community* se muestran dos botones. A la izquierda se accede a la página web del proyecto y a la derecha, como proyección futura, se encuentra el acceso a lo que será la comunidad, en la cual se podrá intercambiar lombrices, abono y experiencias.

4.9. Resultados obtenidos

En este trabajo de memoria se logra crear en forma multidisciplinaria un sistema de vermicompostaje con internet de las cosas orientado a mejorar la experiencia del usuario. El sistema fue probado con usuarios finales como muestra la Figura 4.39 para evaluar el mejoramiento de la experiencia del usuario. En el trabajo de memoria de Camila Martín titulado *Artefacto con internet de las cosas que permite mejorar la experiencia de vermicompostaje* se utiliza una herramienta para medir la emocionalidad de los usuarios al utilizar el sistema creado y determinar el mejoramiento de la experiencia. La herramienta indica un mejoramiento de la experiencia del usuario tanto desde el punto de vista de la emocionalidad como de la usabilidad del artefacto.



Figura 4.39: Prueba del sistema de vermicompostaje con internet de las cosas.

El sistema consiste en el electrodoméstico inteligente mostrado en la Figura 4.26 y en la aplicación móvil asociada descrita en la sección previa. Este electrodoméstico es habitado por lombrices vivas las cuales sobreviven en su interior sin problemas. El electrodoméstico envía datos de la temperatura y humedad del vermicompost cada 1 hora a partir de la primera conexión a internet. La aplicación móvil responde dinámicamente frente a nuevos datos, mostrando información relativa al vermicompostaje y notificando al usuario si la temperatura o la humedad del vermicompost se encuentran fuera de los rangos ideales, la notificación incluye información sobre qué acción de control ejercer para estabilizar las condiciones ideales del vermicompost.

Se logra diseñar e implementar una arquitectura de internet de las cosas a nivel de hardware y software que permite incorporar esta nueva tecnología al sistema de vermicompostaje, la cual funciona correctamente en este despliegue real en campo. Con respecto a los componentes de la arquitectura de software, se utilizan librerías de Arduino de tipo *open-source*, la base de datos Firebase y la plataforma web orientada al desarrollo de aplicaciones móviles

Thunkable. En lo que hardware respecta, se ha hecho uso del microcontrolador NodeMCU 1.0, el sensor de humedad capacitivo de DF Robots, el sensor de temperatura DS18B20, un led RGB y una fuente de alimentación además de de electrónica de bajo costo adquirible en tiendas especializadas.

Se logra con éxito la programación del microcontrolador NodeMCU 1.0 en C++ para obtener la temperatura en grados celsius y la humedad en % de humedad relativa del vermicompost, conectar el microcontrolador en forma inalámbrica a la red local del usuario, traspasar las medidas a una base de datos *cloud* y finalmente desplegar estos datos en la aplicación móvil asociada al electrodoméstico. Esta aplicación es de uso intuitivo ya que sus gestos táctiles responden a gestos naturales de los usuarios al observar ciertas figuras en la pantalla. La aplicación despliega las lecturas de temperatura y humedad en tiempo cuasi-real así como también alertas y notificaciones frente a operación fuera de los rangos adecuados.

Todo el proceso de desarrollo de hardware y software se encuentra plenamente documentado en este mismo documento y en el repositorio de github del autor[19], tal que se facilite la robustez y escalabilidad de la solución.

Capítulo 5

Proyecciones

En caso de darle continuidad a este proyecto, las proyecciones listadas y discutidas en el presente capítulo abordan aspectos claves para continuar con el desarrollo de un electrodoméstico inteligente, ecológico, escalable, amigable con el usuario y con proyecciones comerciales.

5.1. Proyecciones de Software

5.1.1. Arquitectura del Proyecto

Actualmente, el circuito es lo que genera cambios en la base de datos y la aplicación responde a estos cambios. Como proyección, se propone crear una *Application Programming Interface* (API) y montarla en un servidor *cloud*, en forma tal que las acciones de notificación tanto de hardware como de software sean emitidas desde ahí, para esto es requerido que la API tenga acceso a la base de datos, como muestra la Figura 5.1.

Esto reduce la carga computacional del ESP8266 y hace al sistema más robusto, ya que el comportamiento del software es más fiable que el comportamiento del hardware. Por otro lado, errores en el software son más fáciles de detectar al centralizar la lógica de las operaciones. Esto contribuye también a la escalabilidad, ya que para realizar la incorporación de cambios y otras funcionalidades bastará con editar la API, realizando cambios menores en la programación del hardware y de la aplicación. A su vez, la API es multiplataforma, pudiendo crear una visualización no solamente en forma de aplicación móvil, si no que también en forma de aplicación web.

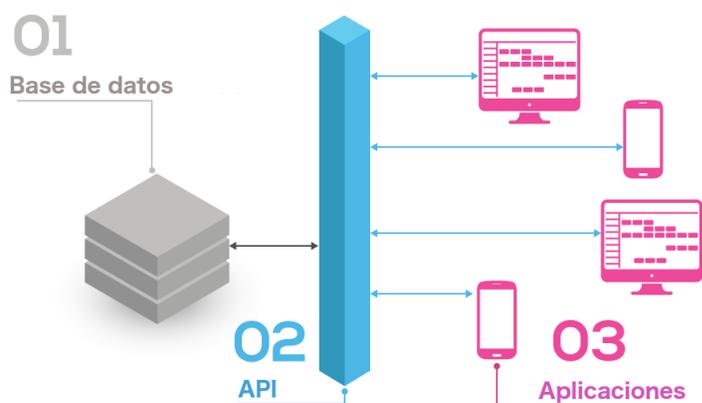


Figura 5.1: Arquitectura de Software incluyendo una API, la cual vincula a la base de datos con las aplicaciones y maneja la lógica de negocio. Fuente https://cianbox.com/blog/novedades-de-agosto-y-septiembre-2018/14986416433514_api/

5.1.2. Base de Datos

Aspectos claves del funcionamiento tanto de la aplicación como del hardware de la Vermi son controlables vía software. En particular, como la comunicación con la base de datos es bidireccional, es posible crear un booleano que sirva para representar el estado de la Vermi y que, por una parte, dé la orden de desplegar las alertas correspondientes en la aplicación y, por otro lado, definir el comportamiento del indicador de acuerdo a los valores de temperatura y humedad.

Se propone también como proyección el tener los textos de las notificaciones en una base de datos, para modificarlos únicamente ahí y no en cada imagen desplegada. La centralización ayuda a optimizar el tiempo dedicado a actualizaciones de la aplicación.

5.1.3. Aplicación Móvil Nativa

El mockup funcional creado es útil para incorporar modificaciones sobre la marcha de acuerdo a indicaciones tanto de los profesores como de los mismos usuarios. Permite cambiar en forma rápida aspectos de estética para concebir una primera versión de la aplicación, la cual, por ser funcional, es posible probarla frente a usuarios reales y obtener de inmediato su retroalimentación sin realizar desarrollos más intensivos. Así, por ejemplo, es posible observar los gestos táctiles que los usuarios realizan sobre la pantalla del *smartphone* e incorporarlos en la próxima versión, ayudando esto a enriquecer la experiencia del usuario.

Luego de algunas iteraciones, el siguiente paso es desarrollar una aplicación nativa para Android y iOS, para este propósito existen diversos *frameworks*, aplicaciones muy populares como Facebook e Instagram se encuentran desarrolladas en *React Native* por ejemplo. El desarrollo de una aplicación nativa permite agregar nuevas funcionalidades como por ejemplo la creación de cuentas de usuario, la implementación del foro de intercambio de lombrices u otros supproductos del vermicompost, la generación de gráficos dinámicos de acuerdo a

hora y fecha a elección e inclusive notificaciones desplegadas en la barra de notificaciones del *smartphone*.

Con la aplicación nativa ya desarrollada, el siguiente paso es publicarla en la Play Store y App Store.

Es posible contactar también a expertos en el desarrollo de aplicaciones para recibir retroalimentación de profesionales y así crear una mejor aplicación. La empresa chilena *2brains* se presenta como una opción, ofreciendo una reunión y revisión gratuita de la aplicación desarrollada en este trabajo.

5.1.4. Predicción de Series de Tiempo

En el contexto del internet de las cosas, sensores de diversas variables del mundo real se encuentran recolectando datos secuencialmente a lo largo del tiempo.

Para realizar una toma de decisiones informada en base a los datos recolectados, como puede ser la emisión de alertas tempranas o preventivas, un problema de interés es realizar predicciones del comportamiento futuro de ciertas variables tomando en cuenta la dimensión temporal de los datos y también la correlación que pueda existir entre estas mismas variables.

Dentro del contexto del estudio de series de tiempo, en la estadística clásica, realizar predicciones sobre el futuro a partir de observaciones previas recibe el nombre de extrapolación. Dentro de las diversas áreas del aprendizaje de máquina, este tópico recibe la denominación de predicción de series de tiempo (*Time Series Forecasting*) [42]. La predicción de series de tiempo incluye la dimensión temporal de los datos en su modelamiento, considerando al tiempo en el cual estos son recolectados como una fuente adicional de información.

Así, una serie de tiempo es una serie de observaciones recolectadas a intervalos regulares de tiempo. Previamente a la predicción de una serie de tiempo, es necesario realizar un análisis de la serie de tiempo, esto comprende extraer estadística descriptiva y otras características de los datos, lo cual permite comprender de mejor forma su comportamiento y así realizar predicciones más precisas.

Algunos modelos de predicción de series de tiempo asumen que la serie de tiempo a predecir es estacionaria [43], esto significa que su media, varianza y covarianza son constantes en el tiempo, como muestra la Figura 5.2.

Dentro de la fase de análisis de la serie de tiempo es clave determinar la ventana de tiempo en la cual la serie de tiempo es estacionaria, para así obtener un desempeño adecuado de la predicción [44]. Para este propósito existen diversas técnicas, tanto gráficas como analíticas, que permiten demostrar si una serie de tiempo es o no estacionaria en cierta ventana temporal.

Para el caso del presente trabajo, se tiene una serie de tiempo multivariable (en particular, de dos variables), la cual es construida a partir de muestreos periódicos cada 1 hora de la temperatura y la humedad del vermicompost. Como proyección, se propone encontrar un modelo predictivo en un intervalo temporal determinado mediante la experimentación el cual

permita emitir alertas tempranas, las cuales se anticipen a la situación en la cual la Vermi quede operando fuera de los rangos adecuados de temperatura o de humedad, solicitando al usuario que ejerza alguna acción de control preventiva para anteponerse a la situación. Como ejemplo, el usuario podría ser notificado con algunas horas o días de anticipación de que la Vermi está perdiendo humedad en forma importante debido a las altas temperaturas de los días de verano y que debe ser regada prontamente.

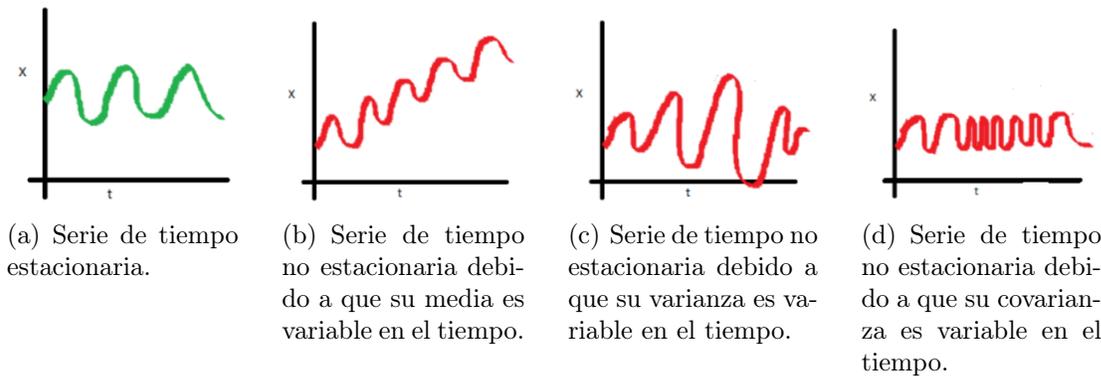


Figura 5.2: Ejemplos de series de tiempo estacionaria (verde) y no estacionarias (rojo). Fuente: Aishwarya singh - A Gentle Introduction to Handling a Non-Stationary Time Series in Python.

5.2. Proyecciones de Hardware

5.2.1. Control Inteligente de Humedad

A mediados del proyecto se considera la opción de incorporar un estanque de agua a la Vermi y una bomba sumergible controlada mediante una activación de tipo On/Off de acuerdo a la operación fuera de rangos.

Como proyección, y si los usuarios lo estiman conveniente, se propone implementar tanto el estanque como la bomba, creando así un sistema de control inteligente para la humedad. La bomba puede ser controlable a través de una variable *cloud* de tipo booleana en la base de datos, la cual represente su estado, cambiando entre On/Off según corresponda.

5.2.2. Placa de Circuito Impreso

Se elabora un esquemático en el software libre Fritzing [45] para facilitar la construcción de una placa PCB, el cual se muestra en la Figura 5.3.

Esto permite reducir el tamaño de la electrónica, lo cual aumenta la cantidad de espacio disponible para los cajones de cosecha, extiende la vida útil del circuito y facilita el reemplazo de componentes dañados fácilmente.

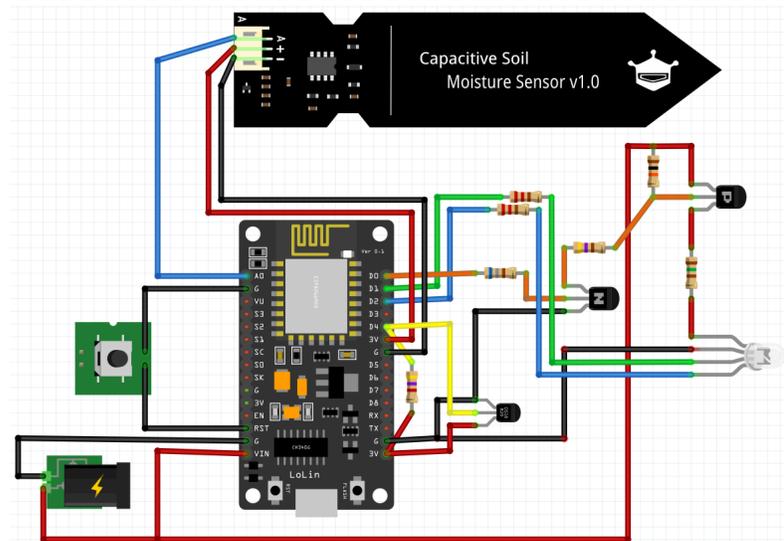


Figura 5.3: Esquemático del circuito a implementar en la Vermi hecho en Fritzing.

5.2.3. Etiqueta de Eficiencia Energética

La etiqueta de eficiencia energética [46] es una herramienta pensada para facilitar la decisión de compra de un consumidor. Se diseñó para indicar el consumo de energía de un artefacto, de tal modo que permita comparar la eficiencia entre productos que pertenezcan a una misma categoría.

En Chile, el año 2007 comienza la obligatoriedad para etiquetar ampollitas incandescentes, fluorescentes compactas y refrigeradores, los que en conjunto representan un 60 % del gasto en electricidad de un hogar.

La etiqueta varía dependiendo del tipo de artefacto, sin embargo, en la Figura 5.4 se muestra una etiqueta de eficiencia energética genérica.

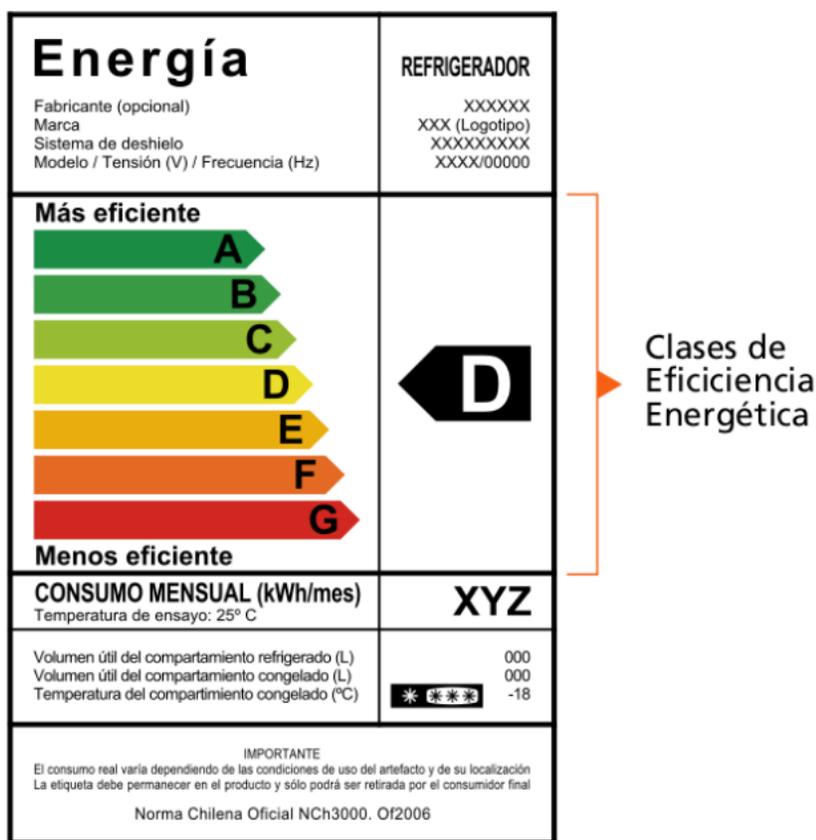


Figura 5.4: La etiqueta de eficiencia energética tiene dos partes fundamentales. En la parte izquierda, la matriz, que contiene las categorías energéticas, unidades y conceptos sobre los que se va a dar información. En la parte derecha, la ficha, que contiene los datos específicos que corresponden al electrodoméstico que lleva la etiqueta. Por medio de la escala de clases o categorías, se ofrece información sobre el grado de eficiencia energética del electrodoméstico en comparación con otros aparatos similares. La escala utiliza un código de colores y letras. La gama va desde el color verde y la letra A para los equipos más eficientes, hasta el color rojo y la letra G para los equipos menos eficientes.

Durante el mes de mayo del presente año se realizaron visitas a la Superintendencia de

Electricidad y Combustibles (SEC) y también al Ministerio de energía para dilucidar la posibilidad de emitir una etiqueta de eficiencia energética con respaldo legal para la Vermi. De las visitas se concluye que hay 2 motivos principales por los cuales no es posible emitir esta etiqueta:

1. La etiqueta se utiliza para comparar artefactos eléctricos y la Vermi es el primer artefacto en su tipo que permite realizar el proceso de vermicompostaje a nivel doméstico y que consume energía eléctrica, por lo tanto, no es posible efectuar comparación alguna.
2. La Vermi no es un producto de consumo masivo, por lo cual no existe normativa legal que permita diseñarle una etiqueta en forma adecuada.

5.3. Otras Proyecciones

Otras proyecciones cuya implementación es importante son:

- Mejor calibración del sensor de humedad: Es posible lograr un mejor desempeño del sensor de humedad realizando una calibración con otro sensor de humedad externo, esto tiene como resultado mediciones más fiables, precisas y no aproximadas mediante un ajuste lineal.
- Mejoramiento de la interfaz de conexión al *wifi*: Esto contribuye a una mejor experiencia del usuario, la consistencia de colores, formas y tipografía es importante para la concepción de un producto integral desde el punto de vista del diseño.
- Creación de un manual de usuario: Una vez concebido el prototipo funcional, el paso siguiente es realizar pruebas de experiencia de usuario con potenciales usuarios finales de la Vermi. Esto con el fin de ir creando en conjunto un manual de usuario del producto, tal como se hizo con las instrucciones de conexión al *wifi*.
- Desarrollar un modelo de negocios para evaluar las proyecciones comerciales del electrodoméstico creado.

Capítulo 6

Conclusión

En el presente trabajo se ha logrado llevar a cabo la creación de un sistema de vermicompostaje con internet de las cosas en forma multidisciplinaria en conjunto con la Diseñadora Industrial, Camila Martín Canales. Se obtiene un artefacto inteligente y funcional con su aplicación móvil asociada que permite realizar el proceso de vermicompostaje a nivel doméstico y en espacios reducidos y que involucra en forma activa al usuario con el proceso en cuestión, mejorando su experiencia. Como es un proyecto conjunto, la comunicación y la coordinación entre los miembros del equipo es importante, prefiriéndose metodologías de trabajo Ágiles para el desarrollo del trabajo.

Desde el punto de vista de la ingeniería, se logra la incorporación del internet de las cosas al sistema de vermicompostaje diseñando e implementado una arquitectura de internet de las cosas. La arquitectura se desarrolla a nivel de software y hardware y soporta un despliegue real en campo.

A lo largo del trabajo se desarrollan dos prototipos, uno funcional y otro estético llamados Vermi 1.0 y Vermi 2.0 para realizar pruebas de concepto. Estos permitieron la definición completa de las arquitecturas de hardware y software, así como también el evaluar la implementación de ciertas funcionalidades y definir su incorporación de cara al prototipo final.

Como conclusiones más particulares sobre la experiencia, se destaca que el reemplazar hardware con software cuando sea posible en general contribuye en forma importante a la reducción de costos de implementación, así como también a la escalabilidad y robustez de la solución. Un buen ejemplo de esto es obtener la hora y la fecha de una medición desde servidores en internet orientados a este propósito en lugar de utilizar un módulo de reloj.

A la hora de realizar proyectos que representen desafíos de innovación, el *hands on learning* se presenta como la mejor estrategia para un desarrollo más flexible. Es clave comenzar por resolver lo más sencillo con las herramientas del estado del arte adecuadas, mejor aún si son conocidas en forma previa, lo cual requiere horas de investigación previa antes de proceder. Una vez que se comienza a proceder, hacer prototipos verticales de construcción fácil y rápida orientados a la validación de las ideas entrega mejores resultados que teorizar y apostar a concebir inmediatamente una solución final.

El trabajo se encuentra documentado en su completitud, tanto desde el punto de vista del ingeniería como del diseño, permitiendo esto dar continuidad al proyecto de cara hacia el futuro.

Capítulo 7

Glosario

- Access Point: Dispositivo proveedor de conectividad wifi para otros dispositivos inalámbricos.
- Aplicación Móvil: Software creado para funcionar en un dispositivo móvil como un smartphone.
- Back log: Lista de tareas que permiten desarrollar un proyecto basándose en las necesidades del cliente.
- Bluetooth: Protocolo de comunicación inalámbrica diseñado para operar en distancias cortas. Transmite en la banda ISM, en particular en la banda de 2.4[Ghz].
- Cloud: Disponibilidad de un cierto recurso computacional en forma remota y de acuerdo a la demanda.
- Dashboard: Tipo de interfaz gráfica de usuario destinada a informar sobre el estado parámetros relevantes de un proceso.
- Dieléctrico: Material aislante.
- Dirección IP: Dirección numérica que identifica a cada dispositivo conectado a una red que utiliza el protocolo de internet para comunicarse.
- Drag and Drop: Posibilidad de mover objetos virtuales desplegados en pantalla utilizando el mouse u otro dispositivo similar.
- Electrodoméstico: Artefacto eléctrico que permite realizar y agilizar tareas domésticas de rutina diaria.
- GSM: Global System for Mobile communications.
- Hardware: Componentes físicas que permiten el funcionamiento de un sistema eléctrico.
- HTML: Hypertext Markup Language.
- HTTP: Hypertext Transfer Protocol.
- Hummus: Fertilizante orgánico de alta calidad obtenido del proceso de vermicompostaje.
- IDE: Integrated Development Environment.
- Internet de las Cosas: Conjunto de artefactos de diversa índole capaz de generar datos de parámetros físicos y transferirlos a través de una red sin intervención humana.
- ISM: Bandas de radio destinadas a uso industrial, médico y científico.

- JSON: Javascript Object Notation.
- Mockup: Modelo (opcionalmente funcional) creado para realizar evaluaciones de diseño.
- NTP: Network Time Protocol.
- Outlier: En estadística es un dato que difiere significativamente de los demás.
- Router: Hardware orientado a la conectividad de redes que permite la conexión entre redes de dispositivos.
- Smartphone: Teléfono inteligente.
- SMS: Short Message Service.
- Software: Set de instrucciones que pueden ser interpretadas por un sistema eléctrico a fin de realizar tareas específicas.
- Station: Dispositivo conectado a una red wifi.
- SSID: Service Set Identifier. Nombre visible de una red wifi.
- Tiempo cuasi-real: Cercano al monitoreo en tiempo real.
- Tiempo real: Monitoreo de un fenómeno dentro del orden de los milisegundos.
- Válvula Solenoide: Válvula electromecánica accionada por un electroimán. Esta válvula permite el control del paso de un fluido por un conducto mediante la circulación de corriente.
- Vermicompost: Mezcla compuesta por tierra húmeda, lombrices rojas y desechos orgánicos en proceso de degradación.
- Vermicompostaje: Proceso mediante el cual desechos orgánicos son degradados por lombrices rojas y otros microorganismos, obteniéndose como resultado hummus y biomasa de lombrices.
- Wifi: Protocolo de comunicación inalámbrico diseñado para operar a nivel de redes de área local. Transmite en la banda ISM, en particular en la banda de 2.4[Ghz] y 5[Ghz].

Capítulo 8

Bibliografía

- [1] Kevin Ashton. That 'Internet of Things' Thing. *RFID Journal*, June 22, 2009.
- [2] Gobierno de España Ministerio de Medio Ambiente y Medio Rural y Marino. Manual de Compostaje. *Centro de Publicaciones de la Secretaría General Técnica*, 2009.
- [3] Viviana Meruane. Beauchef Proyecto. <http://ingenieria.uchile.cl/facultad/ingenieria-y-ciencias-2030/beauchef-proyecto/124181/proyecto>. [Online accessed 9-January-2019].
- [4] Pruthvi R. & Professors: Ravim Yadahalli & Mahendra Naik Students: Abhijith G. R., Adarsh B. Anium S. R. Automated Vermicompost Plant. http://www.kscst.iisc.ernet.in/spp/37_series/spp37s/synopsis_exhibition/147_37S1004.pdf, 2011. [Online accessed 8-January-2019].
- [5] Vishal G. Thengane & Mohit B. Gawande. Arduino Based Tomato Ripening Stages Monitoring System. *International Journal of Innovative Research in Science, Engineering and Technology*, 2018.
- [6] Mayuresh Budukh Soham Dhodapkar Amol V. Dhumane Aqsa Mahir, Tanmay Banavalikar. Soil Monitoring System using Zigbee for Smart Agriculture. *International Journal of Science Technology & Engineering*, 2018.
- [7] SD. Mazaruddin G. V. Satyanarayana. Wireless Sensor Based Remote Monitoring System for Agriculture Using Zigbee and GPS. *Conference on Advances in Communication and Control Systems*, 2013.
- [8] Uo Qian & Xie Min. Temperature and Humidity detection system of communication system based on Raspbeery pi. *International Conference on Intelligent Transportation, Big Data and Smart City*, 2018.
- [9] Romain Bourdon. WampServer. <http://www.wampserver.com/en/>. [Online accessed 8-January-2019].

- [10] Seema Rawat & Tanupriya Choudhury Harsh Vardhan Bhatnagar, Praveen Kumar. Implementation model of Wi-Fi based Smart Home System. *International Conference on Advances in Computing and Communication Engineering*, 2018.
- [11] Google. Firebase products. <https://firebase.google.com/products>. [Online accessed 17-June-2019].
- [12] Srijita Gayen & Saurabh Bilgaiyan Sourabh Sarkar. Android based Home Security Systems using Internet of Things(IoT) and Firebase. *International Conference on Inventive Research in Computing Applications*, 2018.
- [13] Backblaze Inc. Application of Scrum Methods to Hardware Development. <https://www.backblaze.com/blog/wp-content/uploads/2015/08/Scrum-for-Hardware-Development-V3.pdf>, 2015.
- [14] Arie van Bennekum Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Ken Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland & Dave Thomas Kent Beck, Mike Beedle. Manifesto for agile software development. <https://agilemanifesto.org/iso/en/manifesto.html>, 2001. [Online accessed 30-December-2018].
- [15] Alex Brown & Joe Justice. Scrum for full scale manufacturing. <https://34slpa7u66f159hfp1fh19aur1-wpengine.netdna-ssl.com/wp-content/uploads/2015/06/Scrum-for-Full-Scale-Manufacturing.pdf>. [Online accessed 4-August-2019].
- [16] Node-RED. Node-RED Flow-based programming for the Internet of Things. <https://nodered.org/>, 2018. [Online accessed 30-December-2018].
- [17] Thunkable. Drag and Drop App Builder for iOS and Android. <https://thunkable.com/#/>, 2018. [Online accessed 30-December-2018].
- [18] Google. Google Keep. <https://keep.google.com/>, 2018. [Online accessed 30-December-2018].
- [19] Matías Zamora. Sustenta. <https://github.com/matias6942/Sustenta>, 2019.
- [20] Amigos de la Tierra. Manual de compostaje. http://www.madrid.org/cs/BlobServer?blobkey=id&blobwhere=1202767672745&blobheader=application/pdf&blobheadername1=Content-Disposition&blobheadervalue1=filename=MANUAL_COMPOST_ADT_2008.pdf&blobcol=urldata&blohtable=MungoBlobs. [Online accessed 9-July-2019].
- [21] Servicio Hidrográfico y Oceanográfico de la Armada de Chile. Procedimiento para sincronizar servidores y computadores con la hora oficial de Chile. <https://www.horaoficial.cl/doc/sincronizacion.pdf>. [Online accessed 5-July-2019].
- [22] Thunkable. Thunkable variables. <https://docs.thunkable.com/variables>. [Online accessed 17-June-2019].

- [23] Stack Exchange KidCode. What are the advantages and disadvantages of firebase for a database? <https://dba.stackexchange.com/questions/81349/what-are-the-advantages-and-disadvantages-of-firebase-for-a-database/82048/>. [Online accessed 17-June-2019].
- [24] w3Schools. Json introduction. https://www.w3schools.com/js/js_json_intro.asp. [Online accessed 17-June-2019].
- [25] Google. Firebase api reference. <https://firebase.google.com/docs/reference>. [Online accessed 17-June-2019].
- [26] Google. Firebase pricing. <https://firebase.google.com/pricing>. [Online accessed 17-June-2019].
- [27] Arduino. Arduino - Software. <https://www.arduino.cc/en/Main/Software>. [Online accessed 8-January-2019].
- [28] Ivan Grokhotkov y Michael Molinari. Arduino core for esp8266 wifi chip. <https://github.com/esp8266/Arduino>. [Online accessed 29-June-2019].
- [29] tzapu. Esp8266 wifi connection manager with web captive portal. <https://github.com/tzapu/WiFiManager>. [Online accessed 29-June-2019].
- [30] Ed Coyne. Firebase arduino. <https://github.com/FirebaseExtended/firebase-arduino>. [Online accessed 29-June-2019].
- [31] Benoît Blanchon. Arduino json. <https://github.com/bblanchon/ArduinoJson>. [Online accessed 29-June-2019].
- [32] Paul Stoffregen. Arduino time library. <https://github.com/PaulStoffregen/Time>. [Online accessed 29-June-2019].
- [33] Paul Stoffregen. Library for dallas/maxim 1-wire chips. <https://github.com/PaulStoffregen/OneWire>. [Online accessed 29-June-2019].
- [34] Miles Burton. Arduino library for dallas temperature ics. <https://github.com/milesburton/Arduino-Temperature-Control-Library>. [Online accessed 29-June-2019].
- [35] Ivan Grokhotkov. Esp8266 library that calls functions periodically. <https://github.com/esp8266/Arduino/tree/master/libraries/Ticker>. [Online accessed 29-June-2019].
- [36] Maxim Integrated. Programmable resolution 1-wire digital thermometer. <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>, 2018. [Online accessed 28-December-2018].
- [37] Sara Santos. Guide for soil moisture sensor yl-69 or hl-69 with arduino. <https://randomnerdtutorials.com/>

- guide-for-soil-moisture-sensor-yl-69-or-hl-69-with-the-arduino/, 2017. [Online accessed 28-December-2018].
- [38] Berinie Chen. Using capacitive soil moisture sensors on the raspberry pi. <https://www.switchdoc.com/2018/11/tutorial-capacitive-moisture-sensor-grove/>. [Online accessed 26-June-2019].
- [39] ESPRESSIF SYSTEMS. Esp8266ex. <https://www.espressif.com/en/products/hardware/esp8266ex/overview>. [Online accessed 2-August-2019].
- [40] Ivan Grokhotkov. Boards - generic esp8266 module. <https://arduino-esp8266.readthedocs.io/en/latest/boards.html#generic-esp8266-module>. [Online accessed 2-August-2019].
- [41] DFRobot. Capacitive soil moisture sensor. https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193. [Online accessed 26-June-2019].
- [42] Jason Brownlee. What is time series forecasting? <https://machinelearningmastery.com/time-series-forecasting/>. [Online accessed 12-July-2019].
- [43] Aishwarya Singh. A gentle introduction to handling a non-stationary time series in python. <https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/>. [Online accessed 12-July-2019].
- [44] Jason Brownlee. Time series forecasting as supervised learning. <https://machinelearningmastery.com/time-series-forecasting-supervised-learning/>. [Online accessed 12-July-2019].
- [45] Friends of Fritzing. Fritzing. <https://fritzing.org/home/>. [Online accessed 8-July-2019].
- [46] Ministerio de Energía. Manual de etiquetado energético - guía para las empresas. <http://www.minenergia.cl/ganamostodos/docweb/Manual%20Etiquetado.pdf>. [Online accessed 27-June-2019].

Capítulo 9

Anexos

9.1. Conexión al Wifi

A continuación se lista la secuencia de pasos que permiten conectar la Vermi a la red Wifi del usuario final. Estas instrucciones han sido redactadas en conjunto con usuarios potenciales de la Vermi, buscando ser lo más intuitivas posibles.

1. **Enchufe la Vermi y enciéndala. El indicador comenzará a parpadear en verde**, indicando que está lista para configurar la conexión Wifi.
2. Vaya al menú de **Ajustes** de su dispositivo y **Conéctese a la red wifi Vermi Access Point** e introduzca la contraseña **Vermi123**. *Ignore las advertencias de internet no disponible. Si su dispositivo es marca Apple vaya al paso 4.*
3. **Abra el navegador web** (Google Chrome, Mozilla Firefox) y escriba **192.168.4.1** en la barra de navegación. Espere a que cargue la página en el navegador.
4. Haga click en el botón **Configure Wifi**.
5. De la lista de redes disponibles, **seleccione la red de su hogar e ingrese su contraseña. Se verá el mensaje Credentials Saved y el indicador se mantendrá fijo en verde**, indicando que la conexión a la red fue exitosa. El indicador rojo fijo indica valores anormales de temperatura o humedad.
6. Consulte la aplicación móvil para más detalles.

9.2. Costo de Hardware y Materiales

Item	Cantidad	Precio [CLP]	Total [CLP]
ESP8266	1	\$3,500	\$3,500
Sensor Humedad Capacitivo DFRobot	1	\$9,990	\$9,990
Sensor de Temperatura DS18B20	1	\$2,290	\$2,290
Jumpers Hembra-Macho	2	\$1,500	\$3,000
Led RGB	1	\$250	\$250
Switch On/off	1	\$200	\$200
Cordón Blanco 2 x 0.75	1	\$800	\$800
Enchufe Blanco 220v 2 Patas	1	\$700	\$700
Pulsador Reset	1	\$200	\$200
Fuente de Switching	1	\$3,000	\$3,000
Soporte Leds	1	\$50	\$50
Soldadura - Termorretráctil - Resistencias	1	\$6,890	\$6,890
Transistores	2	\$50	\$100
Costo Total Hardware			\$30,970

Tabla 9.1: Tabla de costos de hardware y materiales eléctricos.

9.3. Repositorio de Github

<https://github.com/matias6942/Sustenta>