



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA RED DE SENsoRES INALÁMBRICA PARA LA MEJORA DE PRODUCCIÓN DE MINITUBÉRCULOS DE PAPA

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:
CHRISTIAN LAZO RAMIREZ
INGENIERO EN GESTIÓN INFORMATICA
DOCTOR EN INGENIERÍA TELEMATICA

PROFESOR INFORMANTE 1:
MARÍA ELIANA DE LA MAZA WERNER
INGENIERO CIVIL EN INFORMÁTICA
MAGISTER EN INFORMÁTICA EDUCATIVA

PROFESOR INFORMANTE 2:
JORGE ANTONIO MORALES VILUGRON
DIPLOMADO EN CIENCIAS DE LA INGENIERÍA
MAGISTER EN ADMINISTRACIÓN DE EMPRESAS, MBA

IGNACIO JÉLVEZ HERNÁNDEZ

VALDIVIA – CHILE
2021

ÍNDICE

ÍNDICE DE TABLAS	III
ÍNDICE DE FIGURAS	IV
RESUMEN.....	VI
ABSTRACT	VII
1. Introducción	1
1.1. Contexto	2
1.2. Objetivos.....	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos.....	3
1.3. Impacto esperado del proyecto	4
2. Marco teórico	5
3. Dispositivos y tecnologías.....	7
3.1. Microcontroladores.....	7
3.2. Conexión a internet y red	9
3.3. Sensores.....	15
3.4. Energía.....	26
3.5. Bases de datos.....	30
3.6. Tecnologías para el desarrollo.....	33
4. Arquitectura y solución.....	37
4.1. Descripción de la problemática	37
4.2. Solución propuesta	38
4.3. Metodología.....	39
4.4. Especificación de requisitos	42
4.4.1. Requerimientos funcionales	42
4.4.2. Requerimientos no funcionales	42
4.5. Casos de uso	43
4.6. Modelo de procesos.....	46
4.7. Descripción de los actores del sistema	47
4.8. Arquitectura del sistema	47
4.8.1. Hardware	48
4.8.2. Software	51
5. Pruebas	59
5.1. Validación con soluciones existentes	59
5.2. Plantas y crecimiento.....	60
5.3. Retardo en toma de datos	64
5.4. Consumo de energía	65
5.5. Limitantes	65
6. Mejoras.....	70
6.1. Cambios de sensores.....	70
6.2. Mejoras de algoritmos	71
7. Conclusiones y trabajo futuro	76
7.1. Conclusión.....	76
7.2. Trabajo futuro.....	77
8. Referencias.....	78

ANEXOS.....	83
Anexo A: Visita a invernadero	83
Anexo B: Plantas luego de 2 meses.....	85

ÍNDICE DE TABLAS

Tabla	Página
Tabla 1: Comparación placas Arduino.....	8
Tabla 2: Comparación placas basadas en ESP8266.....	10
Tabla 3: Comparación tecnologías para red inalámbrica.....	14
Tabla 4: Comparación entre bases de datos relacionales y no relacionales	32
Tabla 5: Precios Firebase Realtime Database	34
Tabla 6: Comparativa de características entre aplicaciones móviles híbridas y nativas..	35
Tabla 7: Niveles de Madurez Tecnologica NASA y Unión Europea	39
Tabla 8: Descripción caso de uso “Ver datos de los sensores”.....	43
Tabla 9: Flujo normal de los eventos caso de uso “Ver datos de los sensores”	43
Tabla 10: Descripción caso de uso "Ver gráficas de los sensores"	44
Tabla 11: Flujo normal de los eventos caso de uso "Ver gráficas de los sensores".....	44
Tabla 12: Descripción caso de uso “Configurar sensores a mostrar”	45
Tabla 13: Flujo normal de los eventos caso de uso “Configurar sensores a mostrar”	46
Tabla 14: Datos obtenidos por ambos prototipos.....	62
Tabla 15: Valores iniciales y finales experimento "Prototipo en refrigerador"	65
Tabla 16: Límite de envío de datos utilizando plan gratis en Firebase	67
Tabla 17: Costos al enviar datos durante un año.....	67
Tabla 18: Especificaciones técnicas Huawei B310.....	68

ÍNDICE DE FIGURAS

Figura	Página
Figura 1: Cultivo Hidropónico	2
Figura 2: Cultivo Aeropónico	2
Figura 3: Plantas de papa al interior del invernadero Aeroponics	3
Figura 4: Arduino UNO	8
Figura 5: Rapsberry Pi 3 B+	9
Figura 6: ESP-01	10
Figura 7: NodeMCU	10
Figura 8: Módulo XBee-PRO ZB S2C	11
Figura 9: Módulo Z-Uno	12
Figura 10: SX1276	13
Figura 11: TTGO LoRa32	13
Figura 12: Dispositivo <i>Insta Weather</i>	16
Figura 13: Sistema de Monitoreo Libelium <i>Plug & Sense!</i>	17
Figura 14: Dispositivo Arable Mark 2	17
Figura 15: Sensor cuántico de espectro completo Modbus SQ-522-SS	18
Figura 16: Interfaz Software Sitrad	19
Figura 17: Vista de la información en sala de control y software Sitrad	19
Figura 18: Sensor YL-69/YL-38	20
Figura 19: Sensor de humedad de suelo capacitivo	21
Figura 20: Sensor LM35	21
Figura 21: Sensor DS18B20	22
Figura 22: Sensor DHT11	23
Figura 23: Sensor DHT22	23
Figura 24: Fotorresistencia LDR	24
Figura 25: Sensor ML8511	24
Figura 26: Sensor YF-S201	25
Figura 27: Sensor de PH Gravity	25
Figura 28: Opciones de alimentación de Arduino	27
Figura 29: Opciones de alimentación de NodeMCU	28
Figura 30: Esquema de alimentación de Arduino usando panel solar	29
Figura 31: Ejemplo estructura de datos en tabla SQL	30
Figura 32: Ejemplo estructura de documento JSON	31
Figura 33: Servicios disponibles en Firebase	33
Figura 34: Funcionamiento del sistema de control de temperatura	37
Figura 35: Flujo de datos del sistema	39
Figura 36: Diagrama de secuencia caso de uso "Ver datos de los sensores"	44
Figura 37: Diagrama de secuencia caso de uso "Ver gráficas de los sensores"	45
Figura 38: Diagrama de secuencia caso de uso "Configurar sensores a mostrar"	46
Figura 39: Diagrama de proceso del prototipo	47
Figura 40: Prototipo usando Arduino UNO, vista interior	48
Figura 41: Prototipo usando Arduino UNO, vista exterior	48

Figura 42: Prototipo usando NodeMCU, vista superior.....	49
Figura 43: Diagrama de conexión usando Arduino UNO y ESP-01	50
Figura 44: Diagrama de conexión usando NodeMCU	50
Figura 45: Formato sketch en lenguaje de programación Arduino.....	51
Figura 46: Interfaz generada por Native Base en sistema iOS y Android	52
Figura 47: Diagrama de secuencia del prototipo.....	53
Figura 48: Vista de los datos en la interfaz de Firebase Realtime Database.....	54
Figura 49: <i>VirtualDom</i> de React Native.....	54
Figura 50: React Native Bridge.....	55
Figura 51: Estructura de archivos del proyecto aplicación móvil.....	55
Figura 52: Iconos de navegación en la aplicación móvil	56
Figura 53: Vista inicio de la aplicación.....	56
Figura 54: Vista gráficos de la aplicación.....	57
Figura 55: Vista configuración de la aplicación	58
Figura 56: Dispositivos midiendo al interior y a la intemperie.....	60
Figura 57: Datos obtenidos por el prototipo en la intemperie.....	61
Figura 58: Datos obtenidos por el prototipo en un al interior de una casa.....	62
Figura 59: Plantas a 3 días de iniciado el experimento	63
Figura 60: Crecimiento de plantas al exterior, luego de 8 días	63
Figura 61: Crecimiento de plantas al interior, luego de 8 días.....	63
Figura 62: Gráfica Resultados del experimento “Prototipo en refrigerador”	64
Figura 63: Conjunto de datos enviado por los sensores	66
Figura 64: Almacenamiento usado en Google Cloud luego de las pruebas.....	66
Figura 65: Niveles de señal de la red WiFi	69
Figura 66: Conexión del prototipo con sensores extra.....	70
Figura 67: Datos mostrados en la aplicación con sensores extra	71
Figura 68: Ejemplo definición intervalo algoritmo fijo	73
Figura 69: Ejemplo definición intervalo algoritmo variable.....	74
Figura 70: Ejemplo definición intervalo algoritmo adaptativo	75

RESUMEN

La aeroponía es una técnica que consiste en plantar las plantas en estructuras que las mantienen colgando en el aire, donde las plantas son alimentadas mediante aspersores con soluciones de agua y nutrientes, estas nuevas técnicas de cultivos requieren mayor atención de los agricultores, por lo que diversos invernaderos se encuentran en proceso de automatizar y digitalizar algunas de sus labores.

Este proyecto se enfoca en entregar un prototipo que permita extraer datos de interés respecto del estado de las plantas y permitir revisar la información en tiempo real por los operarios que lo requieran.

Para determinar el prototipo a realizar se realizó una investigación, para definir las variables a medir y que pueden influir en el crecimiento de las plantas, los sensores y microcontroladores a utilizar, así como también las tecnologías para redes de sensores inalámbricas.

El proyecto se realizó en 2 etapas (incrementos), donde se desarrolló una aplicación móvil que muestra la información en tiempo real, así como también un prototipo de extracción de datos.

La aplicación móvil se desarrolla con el framework React Native y es posible instalarla en dispositivos iOS y Android, para el prototipo se desarrollan dos versiones, una utilizando una placa NodeMCU y otra con un Arduino UNO, los cuales capturan información de sensores, como son temperatura, humedad y luz, la cual es enviada mediante una conexión a una red WiFi y almacenada en la base de datos en la nube Firebase Realtime Database.

Se efectuaron pruebas en condiciones simuladas y una revisión bibliográfica de soluciones similares para validar el prototipo, así como también detectar sus limitaciones y futuras mejoras.

ABSTRACT

Aerponics is a technique that consists of planting plants in structures that keep them hanging in the air, where plants are fed by sprinklers with water and nutrient solutions, these new cultivation techniques require more attention from farmers, so various greenhouses are in the process of automating and digitizing some of their work.

This project focuses on delivering a prototype that allows the extraction of data of interest regarding the status of the plants and allows the information to be reviewed in real time by the operators who require it.

To determine the prototype to be carried out, an investigation was carried out, to define the variables to be measured and that can influence the growth of the plants, the sensors and microcontrollers to be used, as well as the technologies for wireless sensor networks.

The development was carried out in 2 stages (increments), where a mobile application was developed that displays the information in real time, as well as a data extraction prototype.

The mobile application is developed with the React Native framework and it is possible to install it on iOS and Android devices, for the prototype two versions are developed, one using a NodeMCU board and other with an Arduino UNO, which capture information from sensors, such as temperature, humidity and light, which is sent through a connection to a Wi-Fi network and stored in the Firebase Realtime Database in the cloud.

Tests under simulated conditions and a bibliographic review of similar solutions were carried out to validate the prototype, it was also sought to detect its possible limitations and future improvements.

1. INTRODUCCIÓN

El crecimiento de las plantas, como en todos los seres vivos, se ve afectado por diversos factores, entre los cuales principalmente corresponden a factores climáticos tales como: la temperatura, la humedad o las precipitaciones, así como también la calidad del aire, que afecta a la parte aérea de la planta (Marín Morales, 1977). Gases como el dióxido de carbono (CO_2), el dióxido de azufre (SO_2) o el monóxido de carbono (CO) pueden afectar, e incluso provocar daños a las plantas cuando se encuentran en concentraciones altas.

Así como las personas, las plantas también pueden verse afectadas por enfermedades, en las raíces se pueden provocar infecciones que hacen que estas se pudran, lo que las imposibilita para absorber agua y nutrientes del suelo. Las infecciones en las hojas pueden provocar manchas, tizones, etc., lo que interfiere en la fotosíntesis, y por ende en su crecimiento.

Dentro de ellas, los tubérculos y en especial las papas no están exentas de estas afecciones (Mendez L. & Inostroza F., 2009), en análisis realizados a las papas que se producen en el sur de Chile se ha encontrado que estas ven afectadas principalmente por las siguientes enfermedades:

- Costra negra: se producen manchas negras en el tubérculo, la planta tiende a producir tubérculos deformes y menos cantidad.
- Sarna: Afecta la piel del tubérculo alterando su apariencia y calidad, sin disminuir el rendimiento.
- Pudrición seca y Pudrición húmeda: pudrición que se produce en el almacenaje de la semilla o cuando se encuentra sembrada, los tubérculos son inutilizables.
- Tizón: uno de los hongos más dañinos para estos tubérculos, puede dañar la planta en pocos días y provoca una especie de pudrición en las papas.

El crecimiento de las plantas y el rendimiento de los cultivos puede ser modificado por acción humana, y para ello se utilizan distintas técnicas, tales como:

- Mejoras genéticas, principalmente para la prevención de enfermedades (Stange, 2020), donde se modifican los genes de las plantas mediante diferentes técnicas, como el cruzamiento de especies o mutaciones inducidas al azar que les otorgan características a las plantas que las plantas antecesoras no tenían, como resistencia a patógenos y a la sequía, mayor cantidad de vitaminas, etc.
- Nuevas técnicas de cultivos (Durán, Martínez, & Navas, 2000), como la hidroponía (ver Figura 1); que consiste en colocar las plantas en sistemas que las sustenten y donde sus raíces se mantengan sumergidas en agua, por lo general en contenedores plásticos donde se mantienen soluciones a base de agua y nutrientes minerales, o los cultivos aeropónicos (ver Figura 2); que son sistemas hidropónicos más modernos, donde las raíces se mantienen “colgando” en el aire y los nutrientes se proveen por pequeños aspersores, esto permite optimizar el uso de agua y proveer de más oxígeno a las raíces.



Figura 1: Cultivo Hidropónico



Figura 2: Cultivo Aeropónico

- Control de los ritmos circadianos de las plantas (Van leperen & Velez-Ramirez, 2011), existen plantas que pueden crecer más rápido si se ven expuestas a luz durante todo el tiempo, en cambio otras pueden dañarse.
- Monitoreo y análisis de datos (CodexVerde, 2019), el monitoreo permite producir plantas de forma más eficiente, ya que se controlan las variables que afectan a los cultivos, reduce riesgos (como plantas que se sequen) y la recolección y análisis de datos permite generar estadísticas y efectuar estudios del comportamiento de la producción.

1.1. Contexto

Aeroponics es una empresa dedicada a la producción de semillas de varias especies de tubérculos de papa, para ello se utiliza la técnica de la Aeroponía (ver Figura 3). Cuentan con una planta de producción se encuentra ubicada en el sector de Cayumapu, comuna de Valdivia y esta cuenta con 8 invernaderos. Actualmente trabajan 14 personas, distribuidos entre gerentes, supervisores, equipo técnico y asistente contable.

Para que las plantas tengan un desarrollo óptimo se debe tener una temperatura estable, lo que se logra con aspersores que mantienen la temperatura, pero la regulación de la temperatura solo se realiza de forma estimativa, ya que la temperatura se mide en los estanques que almacenan el agua (que se encuentran fuera de los invernaderos) y no directamente en las plantas. La medición constante de la temperatura en la planta permitirá mejorar el sistema de riego y mantener una temperatura más estable. Otras variables de interés, cuyas mediciones sería importante incorporar, también son humedad, caudal de agua, radiación solar, entre otras.



Figura 3: Plantas de papa al interior del invernadero Aeroponics

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un prototipo de extracción de datos de un invernadero mediante dispositivos IoT, capaz de registrar información útil para los agricultores sobre los parámetros de interés (temperatura, humedad, caudal de agua, radiación UV, etc.) que podrían mejorar la producción de semillas de papas.

1.2.2. Objetivos Específicos

1. Adquirir conocimientos respecto al uso de soluciones IoT en proyectos de *Smart-farming*, sensores utilizados y tecnologías de conexión entre ellos e internet.
2. Diseñar una arquitectura para la solución que tenga en consideración los requerimientos de la empresa.
3. Implementar una solución que permita recuperar datos sobre los parámetros de interés y almacenarlos.
4. Validar los resultados entregados por la solución mediante la comparación de los datos obtenidos con soluciones existentes.

1.3. Impacto esperado del proyecto

Este proyecto pretende apoyar la automatización de los procesos al interior del invernadero, mediante la obtención de datos en tiempo real para verificar el estado de las plantas, también en un futuro el análisis de los datos que se almacenen debería permitir aumentar la producción, reducir las pérdidas, utilizar menos agua y fertilizantes, etc.

Aeroponics tiene una capacidad de producción potencial anual de un millón de minitubérculos, basado en 18.000 plantas en la suma de todos sus invernaderos. Esto, en tres temporadas de producción.

En un principio el proyecto será aprovechado por este productor, pero en un futuro se podría llegar a otros productores locales que estén dispuestos a colaborar e invertir en estas tecnologías.

2. MARCO TEÓRICO

Para realizar este trabajo se efectuó una breve revisión bibliográfica para determinar el estado del arte en el cultivo de tubérculos utilizando la técnica de la Aeroponía, así como también el uso de microcontroladores y sensores en la agricultura, a partir de ello es posible indicar lo siguiente:

Con respecto a la producción de tubérculos utilizando Aeroponía:

En (Tunio, y otros, 2020) se hace hincapié en la necesidad de nuevos tipos de cultivos, que puedan hacer frente a los problemas para producir alimento, como son la mala calidad de los suelos, la mala calidad del agua y las plagas, esto principalmente en los países en vías de desarrollo. En la producción de papas la Aeroponía permite tasas de crecimiento más altas y tubérculos de papa saludables, uniformes y vigorosos. Este sistema puede producir hasta 10 veces el rendimiento de los sistemas de producción convencionales. Además, este tipo de cultivos brinda protección contra plagas y enfermedades transmitidas por el suelo. En los invernaderos donde se utiliza esta técnica la temperatura no debe ser superior a los 30°C ni inferior a 4°C en el día, en la noche la temperatura óptima es de 10°C a 15°C, la humedad se debe mantener estable ya que las variaciones bruscas afectan el crecimiento de los tubérculos. También se considera medir pH y nutrientes como Nitrógeno, Fósforo, Calcio y Magnesio; que son los que se introducen en la solución de alimento para las plantas.

En (Ritter, y otros, 2001) se comparan diferentes tipos de cultivos, concretamente los cultivos tradicionales, la hidroponía y la Aeroponía, donde se destaca que la hidroponía permite producir tubérculos de mejor calidad en menos tiempo que los cultivos tradicionales, además se evita la propagación de enfermedades encontradas en el suelo y se optimiza la producción de tubérculos al mejorar la disponibilidad del agua y los minerales, puesto que estos llegan directamente a la raíz, la Aeroponía tiene aún más puntos a favor, puesto que optimiza la aireación de las raíces, lo que provoca que el rendimiento de las plantas sea aún mayor.

Con respecto a la aplicación de técnicas de monitoreo utilizando sensores y microcontroladores en la agricultura:

En (Lakhiar, Jianmin, Chandio, Buttar, & Qureshi, 2018) se indica que las mediciones en los sistemas aeropónicos deben tener dos frentes, uno respecto al sistema de bombas y pulverizadores, ya que una falla en estos podría provocar que las plantas mueran, el otro es mediciones directamente en las plantas, donde se utilizan sensores de monitoreo de temperatura, intensidad de luz, pH y conductividad eléctrica, principalmente.

Destaca el uso de un sistema de sensores y actuadores, donde se midieron parámetros como temperatura, presión, humedad, nivel de agua y pH, un arduino captura y envía la información a un nodo coordinador utilizando redes móviles 3G, el cual activa una alarma sonora en el caso que algún parámetro se salga de los rangos normales.

La mayoría de los estudios que han diseñado sistemas aeropónicos usan una red de sensores inalámbricos utilizando ZigBee, Bluetooth, redes 3G o Wi-Fi y como

controladores sistemas Arduino o Raspberry. Por lo general no se utilizan bases de datos en la nube ni técnicas de big data para hacer análisis de datos. Se indica que el uso de la nube brinda muchas ventajas al usuario, como la reducción del costo inicial, la asignación de los recursos bajo demanda, reducción de complejidad en la programación back-end, además de un desarrollo fácil y rápido.

El uso de redes inalámbricas de sensores y actuadores permite al agricultor monitorear varios parámetros sin usar instrumentos de laboratorio, además de poder controlar el sistema de forma remota. El almacenamiento de los datos permite a investigadores tener una mayor comprensión de cómo los parámetros clave se pueden correlacionar con el crecimiento de las plantas.

En general se encontraron sistemas donde se miden variables como PH, temperatura y humedad. En (Kerns & Lee, 2017) se implementó un sistema de monitoreo utilizando una Raspberry PI Zero donde se midieron estos parámetros, y además se activa una bomba para inyectar nutrientes cuando alcanzan un nivel bajo, esto mediante un relé. También se incluye el uso de tecnologías web para la visualización de los datos en tiempo real.

Otras soluciones como la encontrada en (Jagadesh, Karthik, Manikandan, Nivetha, & Prasanth Kumar, 2018) incorporan además paneles fotovoltaicos para el suministro de energía, así como también automatización de ventiladores para expulsar humedad y la activación automática de los atomizadores.

3. DISPOSITIVOS Y TECNOLOGÍAS

Actualmente se encuentran diversas alternativas para realizar un prototipo y captar cientos de variables diferentes, almacenarlas, procesarlas y visualizarlas o hacer estudios de todo tipo, en las siguientes secciones se describirán los microcontroladores, tecnologías para interconectar los dispositivos con internet, sensores y opciones de alimentación de energía que se estudiaron para el uso en el prototipo, finalmente se indica cuales se seleccionaron y las razones para ello. También se encuentra un detalle de tecnologías disponibles para el desarrollo del prototipo y una aplicación móvil.

3.1. Microcontroladores

Un microcontrolador es un circuito integrado, que cuenta con todos los componentes para operar de forma autónoma, tales como memoria, CPU, bus de datos, pines de control, etc. (Weiss & Gridling, 2007). Estos son de especial utilidad en sistemas de control y monitoreo.

Actualmente se encuentran multitud de placas de microcontroladores en el mercado, pero existen 2 empresas que se han popularizado en los últimos años, utilizarlas hace más fácil la integración con otros sistemas y facilita el obtener sensores compatibles, puesto que hay una mayor variedad que se encuentran probados estos microcontroladores.

Arduino

Arduino (Arduino, 2018) es una plataforma de creación de electrónica de código abierto, basada en hardware y software libre, lo que permite que cualquier persona pueda utilizar las placas, desarrollar software o crear sus propias placas personalizadas.

El proyecto nació el año 2005 en el Interaction Design Institute Ivrea en Italia (Kushner, 2011), con el objetivo de permitir que principiantes y profesionales creen dispositivos que permitan interactuar con el entorno.

Las placas Arduino cuentan con un conjunto de pines de entrada y salida programables, los que se pueden utilizar para conectar dispositivos como sensores, botones, etc. Estas se programan utilizando el lenguaje de programación Arduino (basado en Wiring, que tiene una sintaxis similar a C/C++) dentro del software Arduino IDE, aunque también se pueden utilizar otras plataformas.

Esta plataforma se hizo especialmente popular debido a su sencillez a la hora de realizar los prototipos, su bajo costo y la posibilidad de programar en computadores con los sistemas operativos más famosos (Windows, Mac y Linux).

Entre las placas más populares tenemos Arduino UNO (ver Figura 4), Arduino Zero, Arduino Mega, Arduino Nano, también se pueden ampliar sus capacidades mediante el

uso de *shields*, las que se montan sobre la placa y añaden funciones como WiFi, conectividad 3G, relés, pantallas, etc.



Figura 4: Arduino UNO

En la tabla 1 se resumen las principales características de estas placas, esto será de utilidad para la decisión de la placa a utilizar.

Tabla 1: Comparación placas Arduino

Modelo Arduino	UNO	Zero	Mega	Nano
Microcontrolador	ATmega328P	ATSAMD21G18	ATmega2560	ATmega328
Voltaje Operación	5 V	3.3 V	5 V	5 V
Nº Pines Digitales	14	20	54	22
Nº Pines Analógicos	6	6	16	8
Salida 3.3V	SI	SI	SI	SI
Salida 5V	SI	SI	SI	SI
Frecuencia de reloj	16 MHz	48 MHz	16 MHz	16 MHz
Precio	~ 15 USD	~ 40 USD	~ 40 USD	~ 20 USD

Raspberry

Raspberry Pi es una computadora de bajo costo y tamaño compacto, puede utilizarse conectándola a un monitor, teclado y ratón, cuenta con un sistema operativo Linux (Raspberry PI, s.f.).

Esta fue creada el 2012 por la Raspberry Pi Foundation, originalmente pensado para promover y enseñar las ciencias básicas de la computación en las escuelas y universidades de Reino Unido.

Actualmente el modelo más popular es la Raspberry Pi 3 B+ (ver Figura 5), que puede conseguirse por alrededor de 30 USD, además de usarse como pequeño computador puede utilizarse para hacer proyectos de electrónica, ya que cuenta con puertos GPIO, entre sus principales características se tienen:

- Procesador ARM BCM2837B0.
- 1GB RAM LPDDR2.
- WiFi 2.4GHz y 5 GHz, Bluetooth 4.2 y Ethernet.
- 40 pines GPIO digitales, para uso análogo se requiere un conversor.
- Requiere 5V/2.5A para poder funcionar.
- Puerto HDMI, 4 USB 2.0.



Figura 5: Rapsberry Pi 3 B+

3.2. Conexión a internet y red

Si bien la mayoría de los modelos Rapsberry cuentan con conexión WiFi y Bluetooth, y existen modelos Arduino que son compatibles con WiFi mediante el uso de un *shield*, existen otras alternativas para formar una red entre los dispositivos que se encontrarán realizando las mediciones. En esta sección se describirán las principales tecnologías inalámbricas, además de dispositivos que podrían permitir conectarse a las placas de microcontroladores mencionados en la sección anterior para dotarlos de conectividad inalámbrica que no tienen, o en su defecto ser reemplazados por estos porque cuentan con un microcontrolador integrado y pines de entrada/salidas suficientes para las necesidades del prototipo.

Nota: En las tecnologías que no se indican dispositivos, esto es debido a que en Chile no tiene buena disponibilidad de estas redes no se buscaron dispositivos compatibles.

3.2.1. WiFi

WiFi es una de las tecnologías inalámbricas más conocidas, se encuentra estandarizado por la IEEE, el último estándar disponible es el 802.11ax. Una de sus mayores virtudes es la interoperabilidad entre todos los dispositivos certificados, así como la cantidad de dispositivos compatibles. El alcance de las redes WiFi depende de varios factores como son la banda de frecuencia, la salida de potencia de radio, sensibilidad del receptor, ganancia y tipo de antena, etc. También es posible ampliar su alcance con el uso de antenas o repetidores direccionales.

Entre sus características destacan:

- Alcance Geográfico: Alrededor de 100 metros.
- Transmisión de datos: Hasta 600Mbit/s con 802.11n, 6 Gbps con 802.11ac y 9Gbps con 802.11ax.
- Opera en la banda de frecuencia de 2.4 GHz y 5GHz.
- Utiliza el protocolo TCP/IP

Entre los dispositivos que se pueden usar para el uso de esta tecnología tenemos los basados en el ESP8266:

El ESP8266 es un chip de bajo costo WiFi, fabricado por la empresa china Espressif, el primer chip se hizo conocido el año 2014, el módulo ESP-01, este se encontraba diseñado para ser usado como un dispositivo independiente, con 2 pines GPIO digitales donde se pueden conectar sensores o actuadores, o como complemento a un Arduino, como se verá más adelante este es uno de los usos que se le da a este dispositivo en el prototipo.

Existen varias placas basadas en módulos ESP, entre las más populares tenemos el ESP-01 (Figura 6), ESP-12, NodeMCU (Figura 7) y Wemos D1.



Figura 6: ESP-01



Figura 7: NodeMCU

En la tabla 2 se encuentra una comparativa entre las placas basadas en el ESP8266.

Tabla 2: Comparación placas basadas en ESP8266

Modelo Placa	ESP-01	ESP-12	NodeMCU	Wemos D1
Pines GPIO	2	11	11	11
Pines análogos	0	1	1	1
Uso con <i>protoboard</i>	Se requiere adaptador	Se requiere adaptador	Listo para usar	Difícil (tiene formato Arduino)
Operación	Independiente o como complemento a Arduino.	Independiente	Independiente	Independiente
Conexión PC	Conversor Serial	Conversor serial	Micro USB	Micro USB
Precio	~ 3 USD	~ 6 USD	~ 6 USD	~ 6 USD

3.2.2. Bluetooth

Tal como WiFi, Bluetooth es una de las tecnologías más usadas en la electrónica de consumo, se ha utilizado crecientemente en entornos industriales, principalmente por su bajo coste.

Entre sus características se cuentan:

- Alcance Geográfico: Alrededor de 10 metros.
- Transmisión de datos: hasta 50 Mbit/s
- Opera en la banda ISM de 2.4GHz.

A las placas Arduino se les puede añadir conectividad bluetooth mediante el uso de módulos como HC-05 o HC-06 para el envío de mensajes entre estos.

3.2.3. Zigbee

Zigbee es una tecnología inalámbrica diseñada para aplicaciones donde se requiere una baja tasa de envío de datos y maximizar la vida útil de las baterías, se caracteriza por la poca necesidad de componentes electrónicos en la construcción de los dispositivos, en el ámbito donde se prevé tenga más fuerza en los próximos años es en la domótica.

Entre sus características se cuentan:

- Alcance Geográfico: hasta 100 metros en exteriores y alrededor de 30 metros en interiores.
- Transmisión de datos: hasta 256 kbit/s.
- Opera en las bandas: 868 MHz en Europa, 915 MHz en Estados Unidos y 2,4 GHz en todo el mundo.
- Soporta 3 tipos de topologías de red: estrella, malla y árbol.

Los módulos más utilizados para usar esta red en prototipado son los módulos XBee (ver Figura 8), estos se pueden utilizar de forma independiente, ya que cuentan con módulos de entrada/salida y conversores análogos, o mediante conexión serial con Arduino, estos se deben programar directamente con un software que provee Digi, su fabricante. Estos módulos se pueden encontrar a partir de los 30 USD aprox.



Figura 8: Módulo XBee-PRO ZB S2C

3.2.4. Z-Wave

Protocolo de comunicaciones inalámbricas utilizado principalmente para domótica, se considera la competencia más directa de Zigbee.

Entre sus características se cuentan:

- El alcance de la comunicación de los nodos es de alrededor de los 30 metros.
- Los dispositivos Z-Wave son todos compatibles entre sí, siempre que estos se encuentren certificados para funcionar.
- Se soportan saltos de información de hasta 4 veces entre nodos de la red.
- Funciona en las bandas 800-900 MHz.

Una de las placas que se puede utilizar para esta red es Z-Uno (ver Figura 9), esta es una placa de expansión con certificación Z-Wave basada en Arduino, también se puede usar en otras placas como Raspberry.

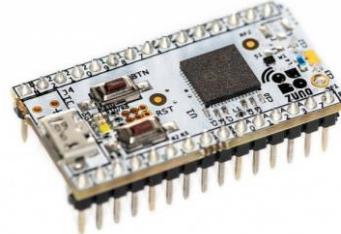


Figura 9: Módulo Z-Uno

Entre sus características se tiene:

- Transmisión a 9.6/40/100 kbps.
- Funciona en diferentes bandas según la región donde se compre.
- Precio: Alrededor de USD 60.

3.2.5. SIGFOX

Sigfox es una solución de conectividad celular pensada para comunicaciones de baja velocidad y consumo de energía, está basado en una infraestructura de antenas y estaciones base, ofrecida por el proveedor de servicios de el mismo nombre, para usarlo se debe estar en una ubicación con cobertura y un dispositivo que se encuentra apto para conectarse a la red administrada por SIGFOX Network Operator (SNO). Es un negocio similar a los de los operadores de telecomunicaciones celulares, pero enfocado en IoT.

Características:

- Utiliza bandas ISM.
- Transmisión de datos: 10 bps a 1000 bps.
- Consumo de energía: Se han logrado autonomías de hasta 15 años usando baterías.
- Gestión basada en la nube.
- Alcance Geográfico: 30-50 km en zonas rurales, 3-10 km en zonas urbanas.

- En Chile SNO tienen disponibilidad en algunas zonas, se encuentra operado por WND Chile.

3.2.6. Lora

LoraWAN es la capa de red de estándar abierto gobernada por LoRa Alliance. Sin embargo, no está realmente abierto ya que el chip necesario para implementar una pila completa de Lora WAN solo está disponible a través de Semtech. Básicamente, LoRa es la capa física: el chip. LoRaWAN es la capa MAC: el software que se coloca en el chip para permitir la conexión en red. Con una red LoRa se pueden construir soluciones propias, donde el desarrollador puede configurar y administrar su propia red.

Características:

- Transmisión de Datos: de 250bps a 50Kbps
- Alcance geográfico: de 10 a 15Km
- Duración de baterías: de 10 a 20 años
- Encriptación: AES 128

Para poder utilizar LoRa se pueden usar dispositivos que funcionan de forma independiente como el SX1276 (Figura 10) o TTGO-LoRa (Figura 11), a los cuales se les puede conectar directamente los sensores o módulos específicos para Arduino, además de *shields*, todos estos se encuentran a partir de los 20 USD, además se requiere el uso de un Gateway, los que tienen un costo cercano a los 100 USD.



Figura 10: SX1276

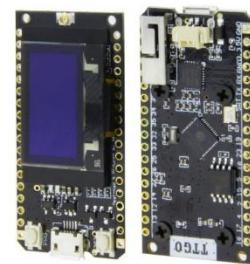


Figura 11: TTGO LoRa32

3.2.7. Narrowband

NB-IOT es una iniciativa del 3GPP, organización encargada de definir diferentes estándares de sistemas de tecnología celular, este se creó como respuesta a SIGFOX para implementarse en redes LTE, para el uso en dispositivos que requieran un bajo consumo de energía y baja transferencia de datos. En Chile solo se han realizado pruebas en medidores de consumo de agua, utilizando la red de 700MHz de Telefónica.

3.2.8. Resumen

Después de revisar las principales placas de desarrollo disponibles y las tecnologías de red para la realización del prototipo se seleccionaron las que se utilizaran en este, para el uso del microcontrolador se utilizará Arduino, debido a su bajo coste, además que su propósito es claramente el desarrollo de prototipos, a diferencia de Rapsberry que está enfocado en ser usado como pequeños computadores para tareas definidas, en específico se utilizará la placa Arduino Uno, debido a que cuenta con un número de pines suficiente para los requerimientos y es la de menor costo de las placas Arduino revisadas, además la mayoría de documentación existente se basa en el uso de esta placa, lo que facilitará el desarrollo.

Para la selección de las tecnologías de red podemos descartar:

- Z-Wave: Existe solo un dispositivo que se puede utilizar y tiene un alto costo, la red tiene poco alcance (cerca de 30 metros) y se requiere utilizar un controlador que cuente con esta tecnología para interconectar los dispositivos.
- SIGFOX: En Chile existe un operador que provee esta tecnología, pero no se encuentra disponible en muchas zonas de la zona sur.
- Narrowband: No existe una red disponible en Chile para su uso.

En la tabla 3 se tiene un resumen de las tecnologías de red que tienen factibilidad:

Tabla 3: Comparación tecnologías para red inalámbrica.

Tecnología	WiFi	Bluetooth	Zigbee	LoRa
Alcance	~ 100 metros	~ 10 metros	~ 30-100 metros	~ 10 km
Consumo Energía (Standby)	> 35 mA @ 3.3V	> 40 mA @ 3.3V	> 33 mA @ 3.3V	> 1.6 mA @ 3.3V
Velocidad transmisión datos	Alta, ~50 Mbps en 2.4 GHz	Baja, < 0.05 Mbps	Baja, < 0.256 Mbps	Baja, ~ 250bps - 50Kbps
Variedad dispositivos	Alta	Baja	Baja	Media
Precio	< 6 USD	~ 3 USD	~ 30 USD	> 20 USD

Como se puede apreciar en la tabla la tecnología Bluetooth se puede descartar debido a su bajo alcance, LoRa tiene un excelente consumo energético, incluso cuando se está en *standby* sin opciones para reducir el consumo de energía, pero los dispositivos tienen un costo más alto y el rango de alcance de la señal es muchísimo más alto que lo que se requiere para invernaderos que están a unas decenas de metros de distancia, Zigbee se encuentra similar en características a WiFi en general, aunque WiFi tiene más velocidad de transmisión (que para el uso que se le va a dar no es relevante, debido a que se envíaran unos pocos *kilobytes* cada cierto tiempo), los módulos Zigbee son más caros y se pueden obtener con solo un fabricante, además un punto de acceso WiFi para conectar los prototipos y subir la información a internet tienen un menor costo que los con tecnología Zigbee, por lo que WiFi parece la mejor elección para este caso.

Los dispositivos WiFi a utilizar para realizar prototipos serán 2, un ESP-01 conectado a un Arduino UNO y un NodeMCU de forma independiente, de esta manera se podrán probar las capacidades de ambos y definir cuál puede resultar mejor, NodeMCU tiene a favor que tiene un costo muy bajo, pero Arduino cuenta con un mayor número de pines analógicos, por lo que se podría conectar un mayor número de sensores de este tipo sin necesidad de utilizar un conversor.

Finalmente se puede agregar que el entorno de programación para todos estos dispositivos es el mismo (Arduino IDE), por lo que el código usado en Arduino junto con ESP-01 será fácilmente portable a NodeMCU y viceversa.

3.3. Sensores

En general podemos encontrar 2 tipos de soluciones, las comerciales y el *open hardware*, las comerciales son sensores que ofrecen empresas para operar con dispositivos específicos, normalmente utilizando software propietario que no se puede modificar. Las soluciones open hardware son más económicas, existe más variedad de dispositivos y una comunidad de usuarios y desarrolladores que aporta con mejoras y software para utilizar, pero se requiere de más conocimiento para utilizarlos, puesto que son los usuarios quienes deben realizar las soluciones y no una empresa que ofrece todo en un paquete *plug and play*.

3.3.1. Soluciones Comerciales

En esta sección se muestran algunas soluciones IoT comerciales de monitoreo para la agricultura, que son destacadas en Chile y en el mundo.

Soluciones en Chile

Savtec (Savtec, s.f.) es una empresa proveniente de Viña del Mar, cuentan con sistemas de IoT en diversas áreas, en el agro implementan sistemas de telemetría, donde se miden variables como humedad del suelo y ambiente, además de temperatura, los que sirven para controlar de manera automatizada sistemas de regadío. Para comunicarse con internet usan tecnologías como XBee, LoRA o WiFi. La información obtenida por los sensores se puede ver en una interfaz web dispuesta para ello.

Instacrops es una empresa chilena con presencia en 8 países de Latinoamérica, ofrecen sistemas para medir parámetros del clima y riego, instalan los sistemas electrónicos en una caja estanca resistente al agua conectada a paneles solares, la conexión a internet se realiza mediante conexión 3G. Además utilizan drones para captar imágenes que sirvan para medir parámetros de la plantación como el número de plantas o la detección enfermedades en ellas. La información puede visualizarse en aplicaciones móviles y cuentan con sistemas de inteligencia artificial para optimizar el uso de agua en regadíos. Una solución similar a la desarrollada en este proyecto que esta empresa ofrece es *Insta Weather* (ver Figura 12), que monitorea variables como temperatura ambiente, humedad

relativa, presión barométrica y punto de rocío (Instacrops, s.f.), una solución de este tipo tiene un costo a partir de los 400 USD aproximadamente, incluso llegando a precios sobre los 2000 USD en modelos con funcionalidades más avanzadas (Chile Desarrollo Sustentable, 2015).



Figura 12: Dispositivo *Insta Weather*

Soluciones en el mundo

Libelium (Libelium, s.f.) es una empresa española, desarrolla todo tipo de sensores, para *Smart cities*, agricultura, domótica, salud, etc. En agricultura cuentan con sistemas para medir diversos parámetros, incluyendo nutrientes específicos como potasio, sodio o calcio, sistemas de regadío o redes de estaciones meteorológicas. Además utilizan sistemas de conexión inalámbrica como ZigBee, WiFi, 4G, LoRaWAN o Sigfox y cuentan con su propia plataforma cloud. Un sistema como el de la Figura 13, con conexión WiFi, con sensores de temperatura, humedad y presión puede costar alrededor de 700USD, además ofrecen decenas de sensores que se pueden añadir al sistema, con costos desde los 50USD cada uno.



Figura 13: Sistema de Monitoreo Libelium *Plug & Sense!*

Arable (Arable, 2020) es una empresa de California USA, ofrecen un dispositivo (ver Figura 14) que mide diferentes parámetros en los cultivos, como el clima, radiación solar, salud de las plantas, presión, etc. y los sube a una plataforma de analítica de datos propia, lo que permite tomar decisiones en tiempo real. Los dispositivos usan conectividad 2G o LTE-M para enviar los datos. Cada uno de estos dispositivos cuesta 1600 USD más una suscripción de 600 USD anual (Albrecht, 2020).



Figura 14: Dispositivo Arable Mark 2

Apogee (Apogee, Apogee Instruments, s.f.) es una empresa estadounidense dedicada a la producción de sensores ambientales de todo tipo, donde se encuentran sensores de temperatura, UV, oxígeno, de radiación, etc.

De igual manera a las soluciones que se indicaron anteriormente, el costo de estos sensores es elevado, un sensor de temperatura puede costar alrededor de 300 USD o un sensor de

humedad 200 USD, estos sensores se pueden encontrar integrados en soluciones de otras empresas, estas además de la venta de sensores ofrecen sistemas de monitoreo.

Un sensor que Aeroponics se encuentra interesado en implementar es el sensor cuántico de espectro completo (ver Figura 15), el cual tiene un costo cercano a los 500USD (Apogee, Descripción SQ-522-SS, s.f.), esto permite medir la radiación fotosintéticamente activa, que es un factor que altera el crecimiento de las plantas, se ha demostrado que el aumento de radiación cuando es absorbido por las plantas puede aumentar el rendimiento de los cultivos (De La Casa & Ovando, 2011).



Figura 15: Sensor cuántico de espectro completo Modbus SQ-522-SS

Otro sensor que se desea incorporar en el futuro es uno que sea capaz de medir CO₂, ya que se ha probado que a mayor nivel de CO₂ se acelera el proceso de fotosíntesis y por lo tanto el crecimiento y producción de la planta, cuando el nivel de CO₂ baja se debe ventilar el invernadero (Nutricontrol, 2020), Libelium ofrece sensores calibrados de este tipo por valores cercanos a los 600 USD.

Una solución comercial que se utiliza actualmente en el invernadero es ofrecida por la empresa Full Gauge (Sitrad, s.f.), este consta de un software llamado Sitrad (ver Figura 16) el cual se encuentra diseñado para administración a distancia de instalaciones de refrigeración, calentamiento, climatización y calentamiento solar. Este software se puede integrar con sensores de la misma empresa.

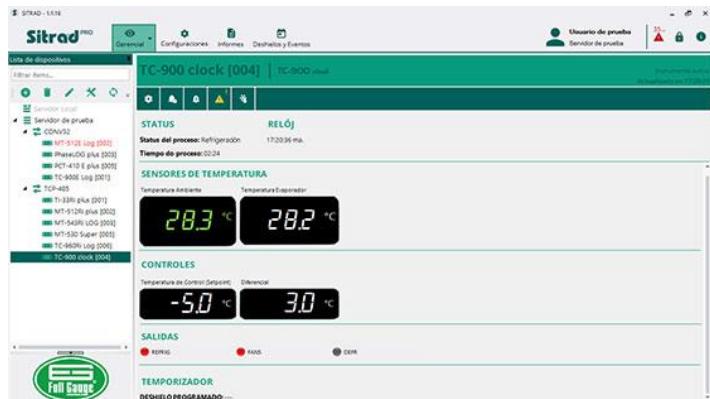


Figura 16: Interfaz Software Sitrad

En el invernadero este sistema se implementó recientemente para obtener información de temperatura y humedad principalmente, esto para cada invernadero (se tiene un dispositivo midiendo en cada invernadero), esta información puede ser vista desde una sala de control o el software Sitrad (ver Figura 17). Este software es de uso gratuito, pero cada sensor o controlador que funciona con el sistema cuesta alrededor de 100 USD, y cada punto de conexión cerca de 300 USD, el sistema completo que se encuentra implementado en el invernadero tiene un valor comercial de al menos 5000 USD.

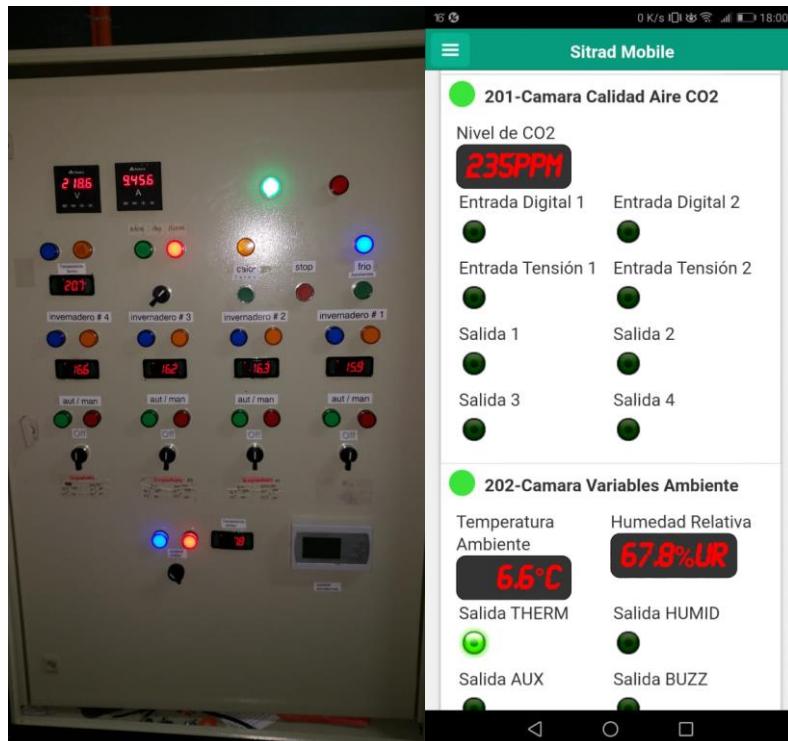


Figura 17: Vista de la información en sala de control y software Sitrad

Como se puede ver los sistemas de monitoreo comerciales tienen un valor muchísimo más alto que un sistema open hardware, pero también requieren menor intervención por parte del usuario, ya que es una empresa externa que se encarga del desarrollo del producto y el mantenimiento, además de ofrecer el software, aunque una solución desarrollada desde cero puede satisfacer mejor las necesidades de los agricultores, si estas son muy específicas o se quiere tener un mayor control sobre el sistema.

3.3.2. Soluciones *Open Hardware*

En la siguiente sección se muestran algunos sensores que pueden ser de utilidad en el desarrollo del prototipo, estos son principalmente sensores de humedad, luz y temperatura, para finalmente indicar los que seleccionaron para el desarrollo de los prototipos.

3.3.2.1. Sensores de humedad

YL-69/YL-38

Este sensor (Talos Electronics, s.f.) se utiliza para medir la humedad del suelo (ver Figura 18), consiste en una sonda YL-69 con dos terminales separados adecuadamente y un módulo YL-38 que contiene un circuito comparador LM393, un led de encendido y otro de activación de salida digital.

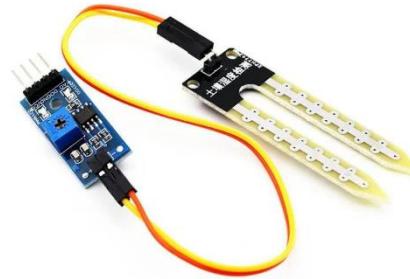


Figura 18: Sensor YL-69/YL-38

Características:

- Tensión de Alimentación: 3.3 V / 5 V.
- Doble Salida: Análoga y digital.
- Corriente: 35 mA.
- Vida útil electrodo sumergido: 3 a 6 meses.
- Precio: ~ 3 USD

Sensor de humedad de suelo capacitivo

Este sensor de humedad (MaxElectronica, Sensor Capacitivo de Humedad de Suelo v1.2, s.f.) (ver Figura 19) es capaz de medir la humedad del suelo donde es insertado, mediante detección capacitiva. Está construido con un material resistente a la corrosión, por lo que

tiene una alta durabilidad, la señal analógica que entrega es proporcional a la humedad del suelo.

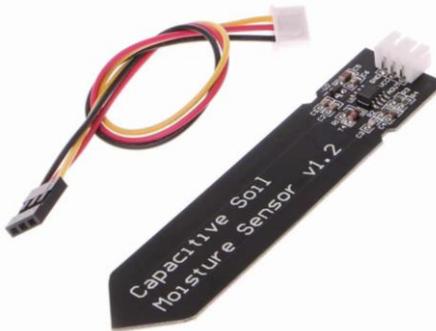


Figura 19: Sensor de humedad de suelo capacitivo

Características:

- Voltaje de alimentación: 3.3V - 5V DC
- Corriente operación: 5mA
- Voltaje de la señal de salida: 0 a 5V (Analógico)
- Vida útil: 3 años mín.
- Precio: ~ 3 USD

3.3.2.2. Sensores de temperatura

LM35

El LM35 (Hardwarelibre, s.f.) (ver Figura 20) es uno de los sensores de temperatura más populares, envía una señal analógica proporcional a la temperatura ambiental. La circuitería viene encapsulada en plástico.



Figura 20: Sensor LM35

Características:

- Envía señal analógica
- Calibrado para grados Celsius.
- Tensión de precisión garantizada de 0.5°C a 25°C.
- Baja corriente de alimentación (60 µA).

- Voltaje de trabajo entre 4 y 30v.
- Precio: ~ 1 USD

DS18B20

El sensor de temperatura DS18B20 (Llamas, DS18B20, 2016) es un sensor idóneo para medir temperatura en ambientes húmedos o debajo del agua, puesto que viene en forma de sonda impermeable (ver Figura 21).



Figura 21: Sensor DS18B20.

Características:

- Alimentación de 3V a 5V
- Utiliza el protocolo 1-Wire
- Precisión de $\pm 0,5^{\circ}\text{C}$ en temperaturas entre -10°C y 85°C
- Resolución programable: 9-bit, 10-bit, 11-bit o 12-bit (por defecto)
- Precio: ~ 4 USD

3.3.2.3. Sensores de temperatura y humedad

DHT11

El DHT11 (Hardwarelibre, DHT11, s.f.) (ver Figura 22) es un sensor que incluye medición de temperatura y humedad, tiene una alta fiabilidad y estabilidad debido a su señal digital calibrada.

Características:

- Alimentación de 3,5V a 5V
- Consumo de corriente de 2,5mA
- Señal de salida digital
- Rango de temperatura de 0°C a 50°C
- Precisión para medir temperatura a 25°C de unos 2°C de variación
- La resolución para medir temperatura es de 8-bit, 1°C
- La humedad puede medir desde 20% RH hasta los 90% RH

- Con precisión para la humedad del 5% RH para temperaturas que se encuentren entre 0-50°C
- Precio: ~ 2 USD

DHT22

El DHT22 (Hardwarelibre, DHT22, s.f.) (ver Figura 23) es una versión mejorada del DHT11, puesto que capta un mayor rango de temperatura y humedad con precisión, además puede realizar hasta dos mediciones por segundo, a diferencia del DHT11 que realiza solo una.

Características:

- Alimentación de 3,5V a 5V
- Consumo de corriente de 2,5mA
- Señal de salida digital
- Rango de temperatura de -40°C a 125°C
- Precisión para medir temperatura a 25°C de 0,5°C de variación
- La resolución para medir temperatura es de 8-bit, 0,1°C
- La humedad puede medir desde 0% RH hasta los 100% RH
- Con precisión para la humedad del 2-5% RH para temperaturas que se encuentren entre 0-50°C
- La resolución es de 0,1% RH, no puede captar variaciones por debajo de esa
- Frecuencia de muestreo de 2 muestras por segundo: 2Hz
- Precio: ~ 4 USD

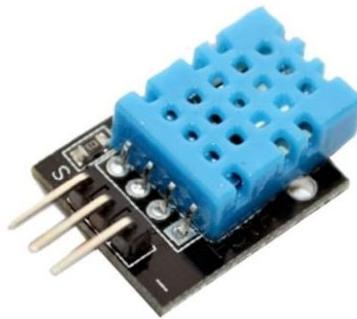


Figura 22: Sensor DHT11

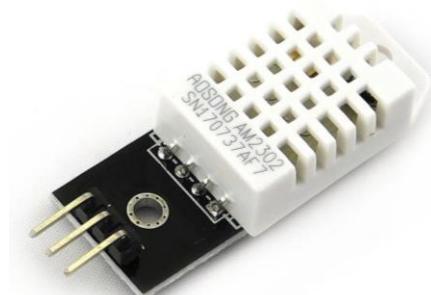


Figura 23: Sensor DHT22

3.3.2.4. Sensores de luz y radiación UV

Fotorresistencia LDR

Una fotorresistencia o LDR (Crespo, 2019) (ver Figura 24) es un componente electrónico cuya resistencia varía en función de la luz.



Figura 24: Fotorresistencia LDR

La resistencia va disminuyendo a medida que aumenta la luz, las mediciones son estables ya que es un dispositivo relativamente lento, tienen un costo muy bajo, alrededor de 1USD cada 5 unidades.

ML8511

El módulo ML8511 (MaxElectronica, Módulo ML8511, s.f.) (ver Figura 25) es un sensor de luz ultravioleta (UV), funciona al emitir una señal analógica en relación con la cantidad de luz UV detectada.

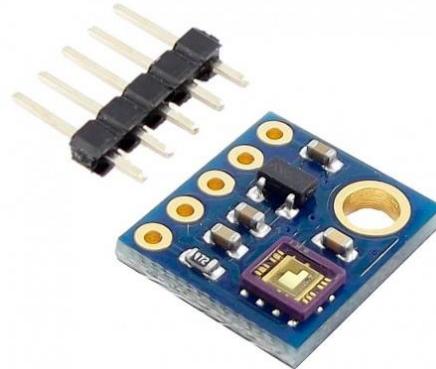


Figura 25: Sensor ML8511

Características:

- Alimentación: 3.3V
- Salida analógica
- Longitud de onda captada: 280-390nm
- Precio: ~ 10 USD

3.3.2.5. Otros sensores

Flujo de agua - YF-S201

El sensor YF-S201 (Mechatronics, 2016) (ver Figura 26) es un sensor de flujo de agua, comúnmente utilizado para medir el consumo de agua, internamente tiene un rotor cuyas paletas tienen un imán y con un sensor de efecto hall se detecta el campo magnético del

imán de las paletas (se detecta el número de giros de estas). Existen varios sensores que funcionan de la misma manera, pero este es uno de los más sencillos y que sirve para flujos de agua bajos.



Figura 26: Sensor YF-S201

- Para utilizar en tubos de 1/2"
- Funciona mediante un sensor de efecto hall que detecta el giro de aspas al interior.
- Alimentación: 5V a 24v DC
- Corriente de operación: 15mA (5V)
- Rango de Flujo: 1~30 Litros/min
- Precio: ~ 10 USD

PH - Sensor de PH Gravity

El pH es una medida de acidez o alcalinidad de una disolución. El pH indica la concentración de iones hidronio $[H_3O^+]$ presentes en determinadas sustancias. El pH se considera un factor de crecimiento en las plantas. Existen multitud de sensores que se comercializan en formato de sonda (ver Figura 27). Este sensor (MCI electronics, s.f.) se encuentra diseñado para funcionar con Arduino, cuenta con un conector BNC y una interfaz para conectarse con el conector analógico del Arduino.



Figura 27: Sensor de PH Gravity

Características:

- Alimentación: 5V.
- Rango de medición: 0-14pH.
- Medición de Temperatura: 0-60°C.
- Precisión: ± 0.1 pH ($25^\circ C$).
- Tiempo de respuesta: ≤ 1 min.

- Precio: ~ 30 USD

3.3.3. Resumen

Para realizar una primera versión del prototipo se analizarán 3 variables: luz, temperatura y humedad, para lo cual se utilizará el sensor DHT11 y fotorresistencia LDR, esto debido al bajo costo de los sensores (es más barato utilizar un sensor DHT11 que usar sensores de temperatura y humedad económicos por separado), los rangos de temperatura y humedad que pueden medir con precisión se encuentran dentro de los rangos esperados para el interior de un invernadero, por lo que se evaluará el rendimiento de estos sensores y se verificará si es necesario reemplazarlos o complementarlos con el uso de sensores más avanzados, como el de humedad de suelo capacitivo y el sensor sonda DS18B20. Debido a que las pruebas no se podrán realizar en instalaciones en invernadero no se utilizará el sensor de flujo de agua en un principio, debido a que se requiere una instalación de riego para poder probarse.

3.4. Energía

Para alimentar de energía a las placas Arduino y NodeMCU existen varias alternativas, las que se revisarán en esta sección, sus ventajas y desventajas, para así seleccionar la más adecuada en cada caso y que se ajusta a las condiciones de los invernaderos.

3.4.1. Opciones de alimentación de las placas

Para alimentar el Arduino UNO tenemos 3 opciones (ver Figura 28) (GeekFactory, 2017):

1. Alimentar mediante el conector *jack*: Al alimentar por este puerto se debe utilizar un voltaje de 7 a 12 volts DC, menor voltaje pueden provocar que no funcione correctamente, y mayores que el regulador de voltaje del Arduino se sobrecaliente o se queme. Lo recomendable es utilizar un adaptador AC/DC de 7V/1A.
2. Alimentar mediante el puerto USB: Al alimentar por este puerto solo se admiten 5 volts, por lo que se puede utilizar un pc, una batería o un adaptador AC/DC, un fusible del Arduino límite la corriente a 500 mA, por lo que esta forma de alimentación es bastante segura. En contra tiene que existen dispositivos que se pueden conectar que requieren más de 5 volts, por lo que en este caso hay que alimentarlos de manera externa con otra fuente de energía.
3. Alimentar mediante los pines: Se puede alimentar mediante el pin VIN o el pin de 5V. Por el pin VIN se puede alimentar de 7 a 12 volts, pero no se cuenta con protección de inversión de polaridad o daño sobre corriente, por lo que se debe utilizar una alimentación estable, tampoco se debe aplicar simultáneamente voltaje en el Jack, esto dañará la placa. Por el pin 5V se puede alimentar utilizando una fuente estabilizada y regulada a 5 volts cuando no hay cable USB o un adaptador conectados al puerto Jack, tampoco se cuenta con ningún tipo de protección en este pin.

En resumen, el método más seguro es utilizar el puerto USB o en su defecto el Jack, siempre utilizando una fuente de energía estable para tener un funcionamiento correcto y no dañar la placa.

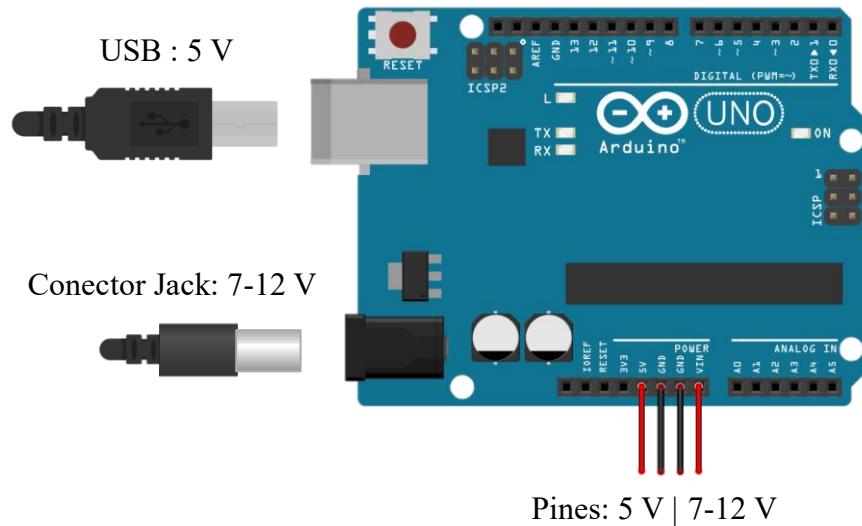


Figura 28: Opciones de alimentación de Arduino

Para alimentar el NodeMCU tenemos 2 opciones (ver Figura 29) (Del Valle L. , 2017):

1. Alimentar el puerto Micro USB: Se recomienda que se utilice una fuente de 5V/1A para obtener el funcionamiento óptimo.
2. Alimentar mediante los pines: Se puede alimentar con 3.3V utilizando cualquier pin de 3.3V, pero si se alimenta utilizando un pin de 3.3V el módulo no podrá entregar más que ese voltaje, por lo que si se requiere conectar algo que requiera 5V el NodeMCU se debe alimentar con 5V. Todos los pines se encuentran protegidos contra subidas de voltaje. También se puede alimentar usando el pin VIN con 5-9V.

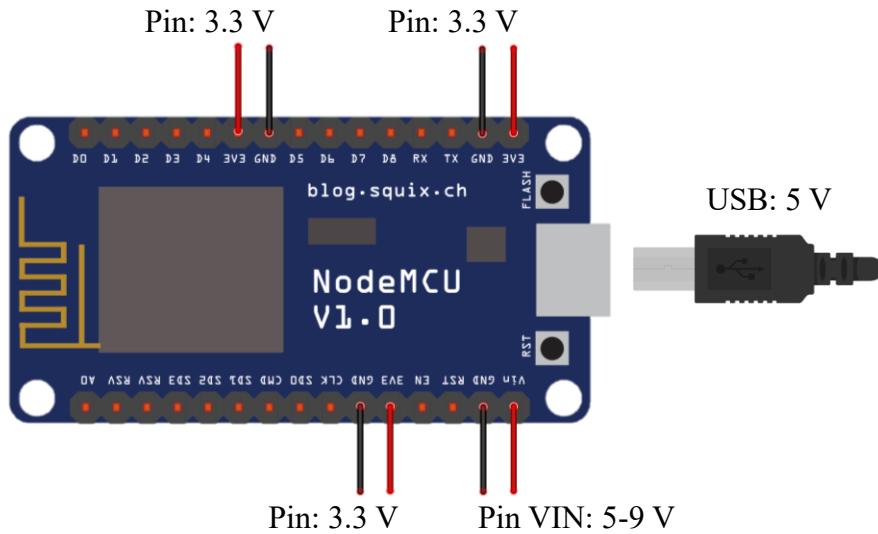


Figura 29: Opciones de alimentación de NodeMCU

3.4.2. Baterías

Para el uso de baterías se tienen 3 opciones (Llamas, Opciones para alimentar Arduino con baterías, 2016):

- Utilizar un grupo de baterías AA o AAA: Es una opción sencilla y económica, siempre que se opte por baterías recargables, por ejemplo 4 pilas AA proporcionan 9V con una intensidad máxima de 2A, aunque su capacidad total es en torno a los 1000mAh, por lo que se pueden descargar rápidamente. Esta es una solución óptima para pequeños proyectos con un consumo energético muy bajo.
- Utilizar una batería de 9V: Son una buena opción para Arduino, son fáciles de conseguir y usar, hay adaptadores para conectar la batería al puerto *jack* del Arduino, en NodeMCU es más complejo ya que se deben usar el pin VIN. Como desventajas tienen una baja capacidad (~500mAh), además tienen una baja intensidad de corriente, por lo que al conectar varios sensores podrían tenerse problemas, como no son recargables se vuelve costoso a largo plazo, además de generar contaminación innecesaria.
- Utilizar baterías recargables de 5V (*powerbanks*): El uso de estas baterías es especialmente interesante, puesto que proporcionan el voltaje apropiado para ambos módulos y para el uso de una gran cantidad de componentes (como sensores, leds, pantallas o motores), como ya entregan un voltaje regulado no es necesario utilizar reguladores externos, además existen con altas capacidades, por lo que se pueden dejar funcionando por mucho tiempo. Cómo funcionan a través de puerto USB cuentan con protección por sobrecarga en Arduino y NodeMCU, pero son más caras que las opciones anteriores y la intensidad máxima es un poco baja (~1A), aunque suficiente para la mayoría de los usos.

3.4.3. Paneles Solares

Para la utilización de cualquiera de los módulos con un panel solar, se requiere conocer el consumo de energía para elegir un tamaño de panel que pueda generar energía suficiente, así como también una batería que pueda entregar al menos dos días de carga, también se requiere un controlador de carga de la batería y un circuito regulador de voltaje (ver Figura 30) (Christoulakis, 2020).

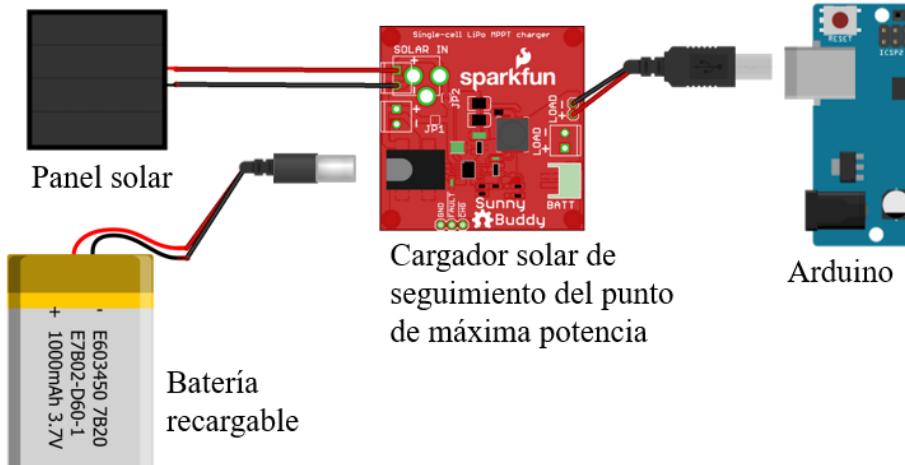


Figura 30: Esquema de alimentación de Arduino usando panel solar

3.4.4. Resumen

Luego de revisar las alternativas para alimentar los módulos Arduino y NodeMCU y los posibles usos de cada uno de ellos es necesario aplicar la viabilidad de utilizar estos métodos de alimentación en el contexto de un invernadero, donde puede haber alto porcentaje de humedad y aspersores funcionando sobre los dispositivos. Como se indicó anteriormente, se tienen las siguientes opciones:

- Conectado directamente a la corriente: Es una excelente alternativa cuando se tiene conexión a la red eléctrica, los adaptadores son baratos y es seguro siempre que se cuente con protección para la humedad en las tomas de alimentación.
- Mediante baterías: También es una buena alternativa, siempre que se utilicen baterías recargables, de lo contrario se estará generando demasiada basura, ya que hay casos donde el consumo de energía es elevado y deben ser reemplazadas en lapsos cortos de tiempo. La mejor opción en este caso es utilizar baterías recargables de 5V, puesto que entregan el voltaje regulado y existen opciones con altas capacidades, lo que permite recargarlas menos veces. Con este método se resguarda la integridad de los componentes, puesto que se pueden instalar en una caja resistente a la humedad.

- Panel solar: Esta opción es la ideal cuando se tienen condiciones para instalar los paneles solares, esto en un invernadero donde existe alta humedad es difícil, puesto que los paneles quedarían descubiertos y se podrían dañar con el tiempo, la mejor opción es instalarlos fuera y cablear hacia los módulos, pero esto puede ser peligroso cuando existen demasiados, puesto que puede haber accidentes con el uso de cables.

A partir de lo expuesto anteriormente el primer prototipo se realizará utilizando una batería de 5V recargable, también es probable que se utilice la conexión directa a la red eléctrica.

Todo esto será realizado conectando directamente los dispositivos a sus respectivos puertos USB, en este caso es la mejor opción, considerando que estos puertos cuentan con protección de sobre corriente en ambos módulos, además no se tendrán altos consumos de energía para requerir de voltajes mayores y los sensores a utilizar requieren alimentación en 5V o 3.3V.

3.5. Bases de datos

Para la realización del proyecto se requiere obtener datos de los sensores, almacenarlos y visualizarlos en tiempo real, por lo que se analizará qué base de datos es una mejor opción para el IoT, a continuación, se describen y se entregan pros y contras de las bases de datos relacionales y no relacionales aplicadas al contexto del proyecto.

3.5.1. Bases de datos relacionales (SQL)

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, almacenando datos en tablas (Ver Figura 31). En una base de datos relacional, cada fila de la tabla es un registro con un ID único llamado clave. Las columnas de la tabla contienen atributos de los datos, y cada registro generalmente tiene un valor para cada atributo, lo que facilita el establecimiento de las relaciones entre los puntos de datos (Oracle, s.f.).

id	temperatura	humedad	luz	luz_uv
1	23.2	74	57	7
2	24.5	75	68	7.2

Figura 31: Ejemplo estructura de datos en tabla SQL

Una de las principales características de las bases de datos relacionales es que estas cumplen con las características ACID, esto es:

- Atomicidad: Solo se ejecutan transacciones completas, o se ejecutan todas las acciones o no se ejecuta ninguna.

- Consistencia: Solo se ejecutan operaciones que no van a romper la integridad de la base de datos.
- Aislamiento: Una operación no puede afectar a otras que se estén ejecutando.
- Durabilidad: Luego de realizada una operación, esta no se podrá deshacer, aunque falle el sistema.

Entre otras características se tienen:

- Escalabilidad vertical, esto es que para aumentar la capacidad de la base de datos se requiere aumentar las capacidades de la máquina donde se está alojando esta y no aumentando el número de máquinas.
- El rendimiento al realizar consultas puede ser más bajo que en las no relacionales, cuando existen consultas que tienen uniones entre varias tablas de la base de datos, por lo que al aumentar la complejidad de la base de datos se requiere más capacidad de procesamiento.
- No es flexible, debido a que se debe tener una estructura de datos previamente definida, para añadir un nuevo dato esta se debe modificar.

3.5.2. Bases de datos no relacionales (NoSQL)

Las bases de datos no relacionales se diferencian de las bases de datos relacionales tradicionales en que almacenan sus datos en forma no tabular. En cambio, las bases de datos no relacionales suelen basarse en estructuras de datos como documentos (ver Figura 32), pero existen varios tipos de bases de datos NoSQL que poseen otras estructuras (mongoDB, s.f.).

```
{
  "temperatura": "23.2",
  "humedad": "74",
  "luz": "57",
  "luz_uv": "7",
  ...
}
```

Figura 32: Ejemplo estructura de documento JSON

Entre sus características se puede destacar:

- Flexibilidad: Las bases de datos no relacionales permiten hacer cambios en el modelo de datos sin necesidad de hacer migraciones, por lo que añadir nuevos datos es más sencillo que en las SQL.
- Escalabilidad horizontal: Se puede aumentar la capacidad aumentando el número de máquinas, esto es muy utilizado para almacenar información en aplicaciones con Big Data, también se utiliza para tener bases de datos distribuidas.
- Poco soporte a la transaccionalidad, esto puede afectar la integridad de los datos, pero las hace más rápidas al ejecutar consultas.

3.5.3. Resumen

Luego de revisar los dos principales tipos de bases de datos, es posible comparar los beneficios y contras del uso de estas bases de datos en un proyecto de IoT (Rautmare & Bhalerao, 2016).

Tabla 4: Comparación entre bases de datos relacionales y no relacionales

Tipo de BD	Relacionales	No Relacionales
ACID	Si	Soportado, pero en pocos SGBD.
Escalabilidad	Vertical	Horizontal
Flexibilidad	No	Muy flexible
Rendimiento	Alto, decae cuando se complejiza la DB.	Alto
Tolerancia a fallos	No	Si
Madurez	Alta	Media (se han hecho populares hace pocos años).

A partir de lo expuesto en la Tabla 4 se puede concluir que las bases de datos no relacionales son la mejor opción para el problema que se espera resolver con el prototipo, por lo siguiente:

- Las bases de datos SQL soportan transacciones ACID, mientras que las no relacionales no siempre, esto no es de especial importancia en este proyecto, debido a que no se efectuarán transacciones que puedan afectar la integridad de los datos, ya que solo se escribirán datos de los sensores y se obtendrán para la visualización de los datos.
- La escalabilidad horizontal es más importante en IoT, debido a que cuando crece el número de sensores comúnmente se distribuye la carga de datos entre distintas máquinas, además estos datos pueden obtenerse de distintas ubicaciones, por lo que es útil tener los datos almacenados cerca.
- La flexibilidad de los datos será útil en las iteraciones y mejoras del prototipo, puesto que si se añaden sensores que midan parámetros nuevos no será necesario hacer migraciones en la base de datos.
- Debe existir tolerancia a fallos, por ejemplo, sin un sensor tuvo un problema para entregar un dato se debe poder visualizar los datos de los demás sensores, lo que con una base SQL podría terminar en que no se suban los nuevos datos medidos.
- La diferencia de rendimiento en este caso no es tan relevante, puesto que si se utilizara SQL la base de datos no tendría muchas tablas con relaciones entre ellas.

En la siguiente sección se mostrará la tecnología a utilizar para el desarrollo de la base de datos no relacional (NoSQL).

3.6. Tecnologías para el desarrollo

3.6.1. Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y móviles desarrollada por Google, es una plataforma en la nube cuya función es desarrollar y facilitar la creación de aplicaciones de alta calidad de forma ágil, incluyendo múltiples servicios (ver Figura 33), la plataforma está disponible para plataformas como iOS, Android y web (Perez Cardona, 2016).

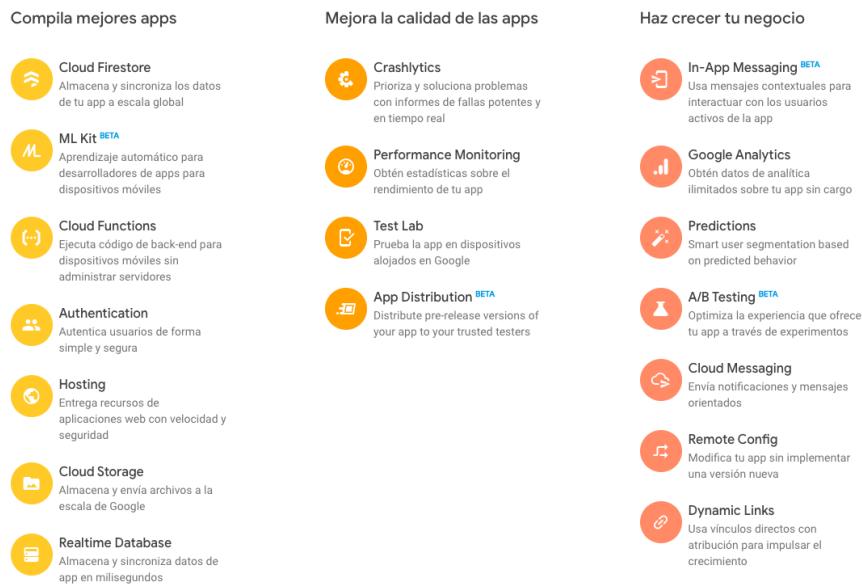


Figura 33: Servicios disponibles en Firebase

Para el desarrollo de aplicaciones tiene funciones como:

- Desarrollo: permite integrar funciones como detección de errores y *testing*, para aumentar la calidad de las aplicaciones.
- Analítica: permite tener un control del rendimiento y del uso de los usuarios de la aplicación, con un panel web de uso gratuito.
- Autenticación: permite autenticar usuarios utilizando únicamente código del lado del cliente (sin utilizar *backend*), incluye autenticación con servicios como Facebook, Github, Google o Microsoft, además de correo electrónico y contraseña.
- Almacenamiento: permite realizar cargas y descargas de archivos desde las aplicaciones conectadas con Firebase.

Además, se incluyen 2 servicios de bases de datos no relacionales:

- Firebase Cloud Firestone: es una base de datos NoSQL, se organiza en forma de documentos agrupados en colecciones, donde se pueden almacenar datos de tipo JSON.

- Realtime Database: Es una base de datos en tiempo real (Firebase, Firebase Realtime Database, s.f.), organizada en forma de árbol JSON, esta proporciona una API para que la información sea almacenada y sincronizada con los dispositivos. Cuenta con integración con aplicaciones Android, iOS, JavaScript, Java, Objetive-C y Swift, también la comunidad ha desarrollado la librería FirebaseArduino (Firebase Arduino, 2016) que permite la integración de esta base de datos con los microcontroladores basados en Arduino.

La sincronización en tiempo real permite acceder a los datos desde cualquier dispositivo en tiempo real, y que cuando se realiza una modificación a esta se almacena en la nube y se notifica al resto de dispositivos.

Como esta base de datos usa la infraestructura de Google, escala automáticamente según el uso que se le dé, se ofrece un uso básico de esta base de datos de forma gratuita

Para el uso de estos servicios existe un plan gratuito y uno de pago por uso, los precios por el uso de Realtime Database se muestran en la Tabla 5.

Tabla 5: Precios Firebase Realtime Database

	Plan gratuito	Pago por uso
Conexiones simultaneas	100	200.000 por base de datos
GB almacenados	1 GB	5 USD por cada GB
GB descargados	10 GB al mes	1 USD por cada GB
Varias bases de datos por proyecto	No	Si

En las pruebas se analizará el uso que se alcanza almacenando los datos de los sensores en esta base de datos, para estimar los costos que se pueda tener al crecer el número de sensores utilizados, y por ende los datos almacenados y el tráfico web.

3.6.2. Desarrollo móvil

Para el desarrollo de aplicaciones móviles se cuenta con 2 grandes alternativas, el desarrollo de aplicaciones nativas o híbridas.

Las aplicaciones nativas se encuentran desarrolladas utilizando el SDK de cada plataforma (iOS o Android) utilizando los lenguajes de programación Objetive-C o Swift en iOS y Java o Kotlin en Android, estas aplicaciones solo pueden ejecutarse para la plataforma que se desarrollaron y no puede utilizarse código fuente de una plataforma en otra.

Las aplicaciones híbridas son aplicaciones que se desarrollan para más de una plataforma, utilizando algún framework como Ionic, React Native u otros, utilizando tecnologías web que incluyen HTML, JavaScript y CSS, por lo general estas aplicaciones se desarrollan una sola vez y pueden utilizarse en los sistemas operativos iOS y Android.

Cada una de estas alternativas cuenta con beneficios y contras, los que deben tomarse en cuenta para decidir de qué forma se desarrollará una aplicación móvil, considerando que esta deberá estar disponible en los sistemas iOS y Android, en la Tabla 6 se resumen algunas de sus características (Gorka, 2019).

Tabla 6: Comparativa de características entre aplicaciones móviles híbridas y nativas

Tipo de aplicación	Hibrida	Nativa
Tiempo de desarrollo	Cerca de la mitad de una nativa, se desarrolla una aplicación y puede ser necesario hacer tareas para garantizar compatibilidad.	Doble, hay que hacer una aplicación desde cero para cada sistema.
Experiencia de uso	Experiencia similar a la nativa, se usan adaptaciones de las UI de cada sistema.	Transiciones fluidas, cambios de vista rápidos, desarrollo de la UI lo provee cada sistema.
Reutilización de código	Se reutiliza el mismo código en diferentes plataformas.	Se crean funciones duplicadas que realizan la misma función.
Rendimiento	Rendimiento un poco menor por el uso de entornos virtuales, optimizado por controles nativos (depende del framework)	El máximo rendimiento posible que ofrece el sistema operativo en el dispositivo.
Coste de desarrollo	Se mantiene al desarrollar para más plataformas.	Crece proporcionalmente según el número de plataformas a desarrollar
Uso recomendado	Aplicaciones sencillas, las funciones no varían dependiendo del dispositivo, no se requiere de extrema velocidad para mantener la experiencia de usuario.	Aplicaciones con gráficos 3D, uso intensivo de CPU, uso de elementos de hardware como lectores NFC, cámaras.

A partir de lo que se ve en la tabla se puede indicar que para este caso la mejor opción es el uso de aplicaciones híbridas, puesto que la aplicación no requiere de avanzadas funciones (principalmente se usará para ver datos de los sensores), lo que no requiere un uso intensivo de la CPU y por lo tanto la diferencia de rendimiento entre ambas opciones no debería ser considerado, además esto permitirá el desarrollo de una sola aplicación que funcionará en los 2 sistemas operativos, como el uso de aplicaciones híbridas por lo general utiliza tecnologías web, la compatibilidad con la base de datos de Firebase se encuentra garantizada.

Entre las tecnologías de desarrollo de aplicaciones híbridas más famosas tenemos (Sandoval, 2019):

- **Ionic:** Creado el 2013, es un framework que proporciona herramientas para el desarrollo de aplicaciones móviles multiplataforma, se basa en el uso de HTML, JavaScript y CSS, pero también se pueden utilizar frameworks de desarrollo web como Angular, React o Vue, cuenta con un conjunto de elementos de UI como formularios, menús y barras de navegación, este se encuentra construido sobre Angular JS. Utiliza complementos de Cordoba para acceder a funciones como cámara, GPS y linterna.
- **Xamarin:** Creado el 2011, es una plataforma para desarrollar apps para Android, iOS y Windows, desarrollada por Microsoft, genera una interfaz de usuario nativa, esta utiliza los lenguajes de programación C# y .NET.
- **React Native:** Creado el 2015, es un framework para desarrollar aplicaciones móviles en iOS y Android, se basa en el framework React, que utiliza el lenguaje de programación JavaScript y se encuentra desarrollado por Facebook. Este genera componentes de UI nativos para cada sistema, lo que proporciona interfaces fluidas y rápidas. Actualmente es el framework para desarrollo de aplicaciones móviles más popular, por lo que cuenta con una basta documentación y apoyo de la comunidad para solucionar prácticamente cualquier problema.
- **Flutter:** Creado el 2017, es un SDK desarrollado por Google, utiliza el lenguaje de programación Dart, utilizando una única base de código para Android e IOS, se encuentra pensado para ser utilizado en el próximo sistema operativo de Google por lo que las funcionalidades aún son reducidas, pero promete ser la base de las aplicaciones Android en un futuro.

A partir de lo indicado anteriormente, se puede indicar que los frameworks más factibles a utilizar son Ionic y React Native (Ravichandran, 2019), esto puesto a que como la base del código utiliza tecnologías web en un futuro se podrá reutilizar este código para la realización de una interfaz web si fuese necesario, e incorporar nuevas funcionalidades como análisis de datos.

Se utilizará el framework React Native principalmente por dos razones, ofrece un rendimiento mejor debido a que utiliza componentes nativos, en cambio Ionic puede tener elementos de UI nativos, pero utiliza complementos de Cordoba para algunas interacciones con el sistema, lo que penaliza el rendimiento, además existe experiencia previa con React Native, debido a que ya existen conocimientos en el framework React el tiempo de desarrollo debería ser más corto.

4. ARQUITECTURA Y SOLUCIÓN

En el siguiente capítulo se describe la problemática a resolver por este trabajo, junto con herramientas de ingeniería de software para describir la solución propuesta, también se muestra cómo se efectuó el proceso de desarrollo del prototipo.

4.1. Descripción de la problemática

En el invernadero Aeroponics uno de los parámetros más importantes que se requiere tener controlado es la temperatura, actualmente esta se mide de manera indirecta.

El sistema que provee de los nutrientes a la planta consta de una serie de micro aspersores que pulveriza una solución de agua con los nutrientes necesarios (alimento), para ello se utilizan bombas que alimentan la red de agua, se cuenta con sistemas para calentar y enfriar el agua, se realizan mediciones de temperatura en el estanque para determinar cuándo se debe calentar o enfriar el agua (ver Figura 34).

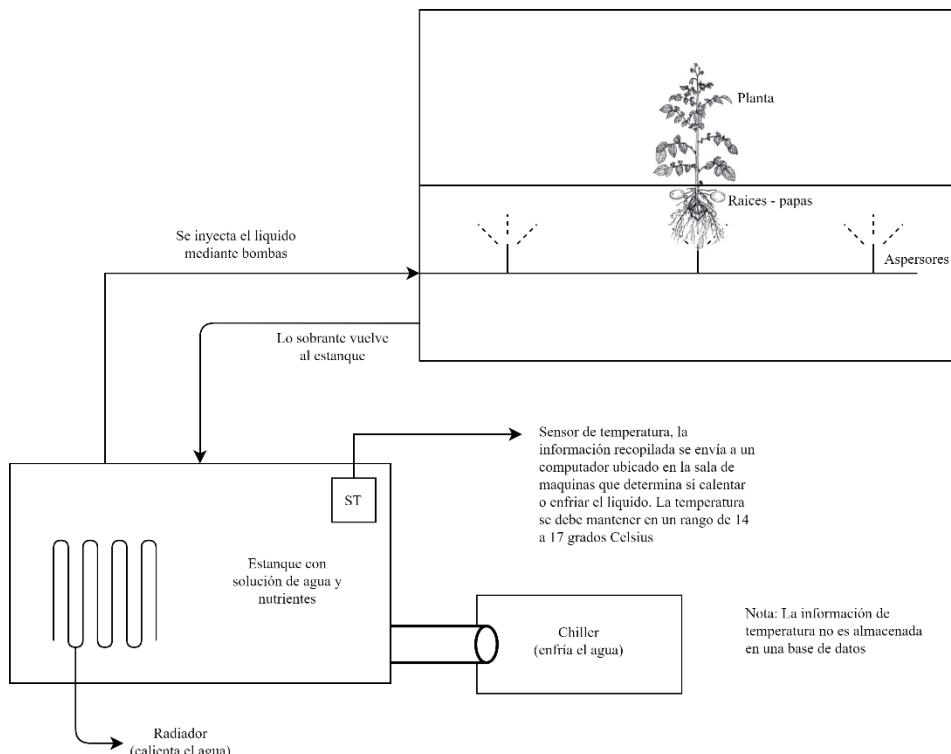


Figura 34: Funcionamiento del sistema de control de temperatura

Actualmente no se almacenan los datos de las mediciones de temperatura, solo se anotan esporádicamente de forma manual, también se tienen termómetros digitales en la superficie del invernadero, pero al igual que las mediciones al interior del estanque no se almacenan en ningún sistema computacional.

Como se ve en la Figura 34 no se efectúan mediciones de temperatura en la planta, sino que estas se realizan en el estanque, por lo que la temperatura real de la planta tiende a ser más alta que la medida en la época de verano y tiende a ser más baja en invierno, ya que en verano la temperatura ambiente en las cercanías de Valdivia puede ser muy superior a 17 grados y en invierno muy inferior a 14 grados, lo que no se tiene en consideración actualmente.

Actualmente existen diferencias en la producción entre los invernaderos, las que se explican en parte por diferencias entre tamaños y tipo de planta. Más aún, en la producción de la primera temporada del 2020 se han encontrado grandes diferencias. Existen plantas que producen 10 minitubérculos y otras que incluso llegan a los 30 minitubérculos.

Dada la ubicación de los invernaderos, alejada del centro urbano, y su tamaño resulta necesario diseñar un sistema de comunicación entre los nodos de la red y la subida de datos a internet.

Este sistema pretende solucionar las problemáticas indicadas, capturando información de parámetros como temperatura y humedad directamente desde las plantas, generar una visualización en tiempo real y almacenar los datos, para que en un futuro sirvan para realizar análisis de estas diferencias en la producción y así hacer modificaciones que permitan optimizar sus procesos.

4.2. Solución propuesta

Se propone desarrollar un prototipo de extracción que sea capaz de registrar información respecto de posibles parámetros de interés (en una primera versión estos serán temperatura, humedad e intensidad de luz), almacenar los datos en una base de datos en la nube y permitir visualizar los datos en tiempo real en dispositivos móviles Android e iOS.

A partir del estudio realizado anteriormente se realizarán 2 versiones del prototipo, una utilizando un Arduino uno y un ESP01 para conectarlo a internet, usando una conexión WiFi y otra versión utilizará un NodeMCU, ambos conectados a sensores de temperatura y humedad DHT11 y sensores de luz LDR con conversor digital.

Estos prototipos serán mejorados luego de pruebas que se realicen con los prototipos en situaciones simuladas y reales, tanto con el cambio de sensores como con optimizaciones de software. Finalmente se decidirá cuál versión del prototipo es mejor para cada caso que se presente.

La base de datos se alojará en servidores de Google, utilizando el servicio Firebase Realtime Database, los datos podrán ser revisados en tiempo real en aplicaciones para Android e iOS las que serán desarrolladas utilizando el framework React Native, la Figura 35 muestra el flujo que seguirán los datos en el sistema.

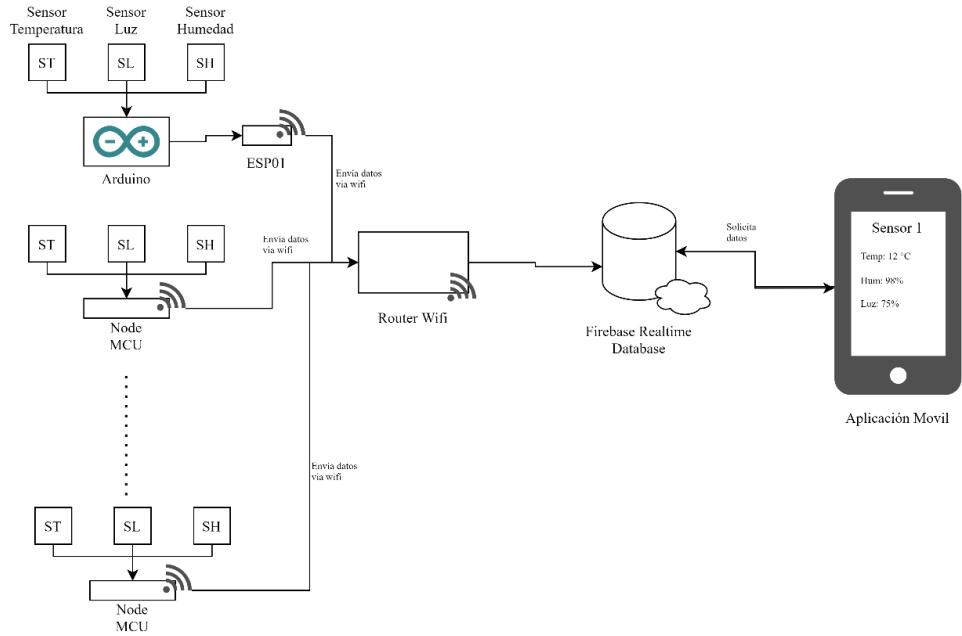


Figura 35: Flujo de datos del sistema

4.3. Metodología

Existe un método para definir el grado de madurez de tecnologías, llamado niveles de madurez tecnológica (European Space Agency, 2008), este concepto fue desarrollado por la NASA durante la década de 1970, existen 2 versiones de esto, una desarrollada por la NASA (NASA, 2012) y otra por la Unión Europea, estas se encuentran divididas por niveles, los que se muestran en la Tabla 7.

Tabla 7: Niveles de Madurez Tecnologica NASA y Unión Europea

Nivel	NASA	Unión Europea
TRL 1	Principios básicos observados y documentados	Principios básicos observados
TRL 2	Concepto de tecnología y/o aplicación formulados	Concepto de tecnología formulado
TRL 3	Prueba de concepto de función crítica demostrada en forma analítica y experimental y/o característica	Prueba experimental de concepto
TRL 4	Validación de componentes y/o placas de prueba en entornos de laboratorio	Tecnología validada en laboratorio
TRL 5	Validación de componentes y/o placas de prueba en un entorno relevante	Tecnología validada en un entorno relevante
TRL 6	Modelo de sistema/subsistema o demostración de prototipo en un entorno relevante.	Tecnología demostrada en un entorno relevante.
TRL 7	Demostración del prototipo del sistema en un entorno espacial	Demostración del prototipo del sistema en un entorno operativo.
TRL 8	Sistema real completado y "calificado para vuelo" mediante prueba y demostración	Sistema completo y calificado
TRL 9	Sistema real "probado en vuelo" a través de operaciones de misión exitosas	Sistema real probado en el entorno operativo

Para la realización del prototipo se espera llegar al nivel TLR3 o TLR4, esto es realizar una investigación y el desarrollo, además de realizar estudios en laboratorio para validar la solución tecnológica, donde se encontrarán integrados los componentes para comprobar que las piezas funcionan juntas, como se subentiende a partir de la tabla, esto no es una solución final y existen más pasos a realizar para llegar a ello (un TLR9).

Para definir el modelo de desarrollo a utilizar en la realización de este prototipo primero se describirán algunos de los principales modelos de desarrollo. Comúnmente se pueden dividir en dos tipos: las metodologías clásicas y las metodologías ágiles.

En las metodologías clásicas se definen los requisitos en un principio, con un plan relativamente detallado, entre sus principales tipos se tienen (Maida & Julian, 2015):

- Cascada: En este modelo se ordenan las etapas del proceso de desarrollo de software de tal manera que el inicio de cada etapa debe esperar a la finalización de la etapa anterior. Al final de cada etapa se efectúa una revisión para determinar si el proyecto está listo para pasar a la siguiente fase. El modelo de cascada data de la década de 1960.
- Prototipaje: En esta metodología se implementan parcialmente sistemas, con el objetivo de aprender de los requerimientos del sistema, se construyen tan rápido como sea posible. Esta metodología se usa comúnmente en sistemas complejos donde es difícil definir los requisitos en un principio.
- Espiral: Toma como bases las dos metodologías anteriores, cuenta con cuatro actividades: planificación, análisis de riesgo, ingeniería y evaluación del cliente, actividades que se van repitiendo hasta llegar a la solución final. En cada iteración el producto va ganando madurez, aproximándose cada vez más a la solución final.
- Incremental: En esta metodología se construye el proyecto en etapas incrementales en donde en cada una de ellas se añaden funcionalidades. Las etapas incluyen requerimientos, diseño, codificación, pruebas y entrega. La solución se va mejorando a través de múltiples iteraciones, las partes más importantes del sistema se entregan antes.

Las metodologías ágiles son un modelo más adaptativo, donde se trabaja sobre funcionalidades básicas que se definen en un inicio, por lo general no se fijan contratos estrictos, sino que se trabaja directamente con el cliente, estas metodologías se basan en el Manifiesto Ágil, el cual define cuatro valores:

1. Al individuo y sus interacciones más que al proceso y las herramientas.
2. Desarrollar software que funcione más que obtener una documentación exhaustiva.
3. La colaboración con el cliente más que la negociación de un contrato.
4. Responder a los cambios más que seguir una planificación.

Entre los principales tipos de metodologías ágiles se tienen:

- Scrum: Esta metodología se caracteriza por adoptar una estrategia de desarrollo incremental (Schwaber & Sutherland, 2020), en lugar de la planificación y ejecución completa del producto, el desarrollo de los incrementos se hace en

- bloque temporales cortos (*Sprints*), donde se desarrolla una parte del software y se entrega al cliente la solución.
- Programación extrema (XP): En esta metodología se produce un diseño iterativo e incremental, se programa en parejas (Copeland, 2001), existe integración entre el equipo de desarrollo y el cliente, existen cambios constantes en los requisitos del proyecto y el equipo de desarrollo se adapta a ello.
 - Desarrollo de Software Lean: Es una metodología que se basa en siete principios para desarrollar software, nacida a partir del sistema de producción Toyota, los principios son: optimizar el todo, eliminar desperdicios, calidad en la construcción, aprender constantemente, entregar rápido, involucrar a todo el mundo y seguir mejorando (Poppendieck, Poppendieck, & Poppendieck, 2003).

Para el desarrollo del prototipo se siguió una metodología de tipo incremental (Ortiz, 2012) , puesto que es la que mejor se adapta para la elaboración de un prototipo donde también es importante realizar documentación, en este proceso se siguió una serie de etapas, cada una dependiente de la anterior, estas son investigación, selección de tecnologías, desarrollo, pruebas, captura de datos, validación y correcciones, esto para una primera iteración, en la segunda iteración se suprimirán las etapas de investigación y selección de tecnologías, puesto que ya no son necesarias.

La investigación consiste en una revisión sistemática, donde se buscó responder a las preguntas ¿Qué tecnologías de red en IoT son factibles de utilizar para una pequeña red de sensores inalámbricos que han sido utilizados en *Smart Farm* para subir la información a la nube? y ¿Qué tipos de sensores son más idóneos para realizar mediciones en invernaderos?, parte de aquella investigación se puede ver en las conclusiones del capítulo 3: Dispositivos y Tecnologías, también se efectuó una investigación respecto del estado del arte en la producción de papas y sistemas de cultivos modernos, como es la Aeroponía, cuya revisión se puede encontrar en el capítulo 2: Marco Teórico.

La selección de tecnologías consiste en la búsqueda y selección de microcontroladores y sensores y su posterior adquisición, además de la selección de tecnologías de desarrollo para la aplicación móvil.

El desarrollo consiste en las conexiones de los dispositivos y la programación de los sensores, además de la realización de la aplicación móvil.

Las pruebas se realizarán en condiciones simuladas y reales, para determinar parámetros como el tiempo de respuesta de los sensores, consumo de energía y duración de las baterías y verificar cómo se comporta en condiciones climáticas más extremas (una lluvia, por ejemplo).

La captura de datos se realizará paralelo a la etapa de pruebas, donde se capturarán datos durante al menos una semana, para verificar que los datos concuerden con la realidad y que los dispositivos funcionen correctamente durante tiempos prolongados

La validación se realizará de forma teórica, con la revisión de documentos e investigaciones que comparen dispositivos similares a los utilizados con soluciones comerciales, que tienen costos muchísimo más altos.

En la etapa de correcciones se reflexionará respecto de los resultados obtenidos y se efectuarán mejoras, tanto cambios de sensores por otros más completos, como también correcciones a los algoritmos utilizados, para mejorar el consumo de energía o calibrar de mejor manera los sensores, por ejemplo.

4.4. Especificación de requisitos

Para la realización de este proyecto se tienen principalmente 2 módulos, el dispositivo de toma de datos y la aplicación móvil, por lo que los requerimientos funcionales y no funcionales para cada uno de ellos se muestran en las siguientes secciones.

Los requisitos expuestos se obtuvieron de reuniones periódicas que tenía el profesor patrocinante con encargados del invernadero, las que se formalizaron en reuniones semanales entre el profesor patrocinante y el estudiante.

4.4.1. Requerimientos funcionales

4.4.1.1. Requerimientos del dispositivo de toma de datos

- RF1: El dispositivo debe medir temperatura y humedad al menos una vez cada minuto.
- RF2: El dispositivo se debe poder conectar a una red WiFi.
- RF3: El dispositivo se debe poder alimentar mediante la red eléctrica o baterías externas.
- RF4: El dispositivo debe mostrar los datos en una pantalla integrada.
- RF5: El dispositivo debe enviar los datos a una base de datos que almacene la información.
- RF6: El dispositivo debe reconectarse automáticamente a WiFi luego de una pérdida de conexión.
- RF7: El dispositivo deberá indicar una pérdida de conexión.

4.4.1.2. Requerimientos de la aplicación móvil

- RF8: La aplicación debe mostrar la última medición disponible.
- RF9: La aplicación debe permitir seleccionar los sensores a mostrar.
- RF10: La aplicación debe mostrar gráficas con datos anteriores.

4.4.2. Requerimientos no funcionales

4.4.2.1. Requerimientos del dispositivo de toma de datos

- RNF1: El dispositivo deberá poder enviar datos por períodos prolongados.

4.4.2.2. Requerimientos de la aplicación móvil

- RNF2: La aplicación deberá ser compatible con los sistemas operativos iOS y Android en versiones recientes.

4.5. Casos de uso

A continuación, se describen los casos de uso detectados a partir de los requerimientos que se indicaron en la sección anterior, las interacciones de usuario se realizan solo en la aplicación móvil y son tres: ver datos de los sensores, ver gráficas de los sensores y configurar sensores a mostrar.

4.5.1. Ver datos de los sensores

Descripción

Tabla 8: Descripción caso de uso “Ver datos de los sensores”

Caso de uso:	Ver datos de los sensores
Actores:	Usuario
Precondición:	Cuenta con la aplicación instalada y conexión a internet, hay sensores enviando información.
Resumen:	El usuario accede a la aplicación para visualizar el estado actual en las plantas.
Tipo:	Primario

Flujo normal de los eventos

Tabla 9: Flujo normal de los eventos caso de uso “Ver datos de los sensores”

Actor	Sistema
El usuario accede a la aplicación	Se muestran las secciones de la aplicación
El usuario selecciona la sección “Monitoreo”	Se muestra información asociada a cada sensor (temperatura, humedad, luz).
Busca el sensor que desea y visualiza la información.	

Diagrama de secuencia

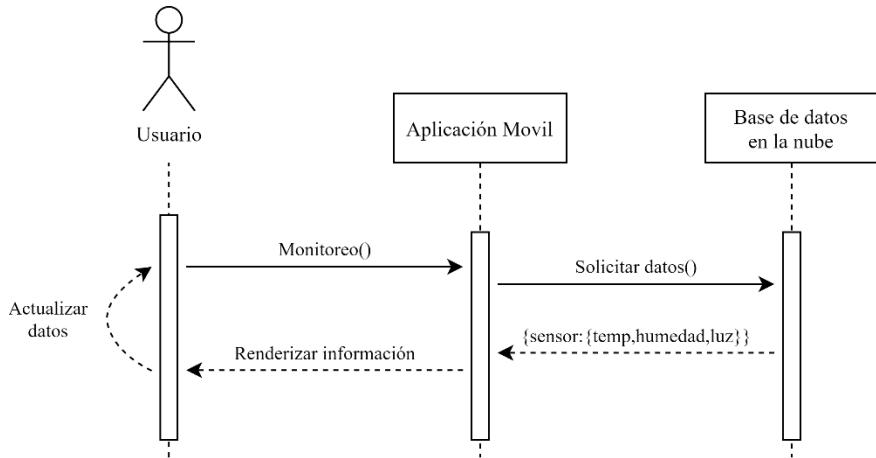


Figura 36: Diagrama de secuencia caso de uso “Ver datos de los sensores”

4.5.2. Ver gráficas de los sensores

Descripción

Tabla 10: Descripción caso de uso "Ver gráficas de los sensores"

Caso de uso:	Ver gráficas de los sensores
Actores:	Usuario
Precondición:	Cuenta con la aplicación instalada y conexión a internet, existe información histórica.
Resumen:	El usuario ingresa a la aplicación para ver estadísticas históricas de los sensores.
Tipo:	Secundario

Flujo normal de los eventos

Tabla 11: Flujo normal de los eventos caso de uso "Ver gráficas de los sensores"

Actor	Sistema
El usuario accede a la aplicación	Se muestran las secciones de la aplicación
El usuario selecciona la sección “Gráficos”	Se muestra la lista de sensores
El usuario selecciona el sensor	Se muestran gráficos con datos de la última hora
El usuario puede cambiar la vista a día o semana	

Diagrama de secuencia

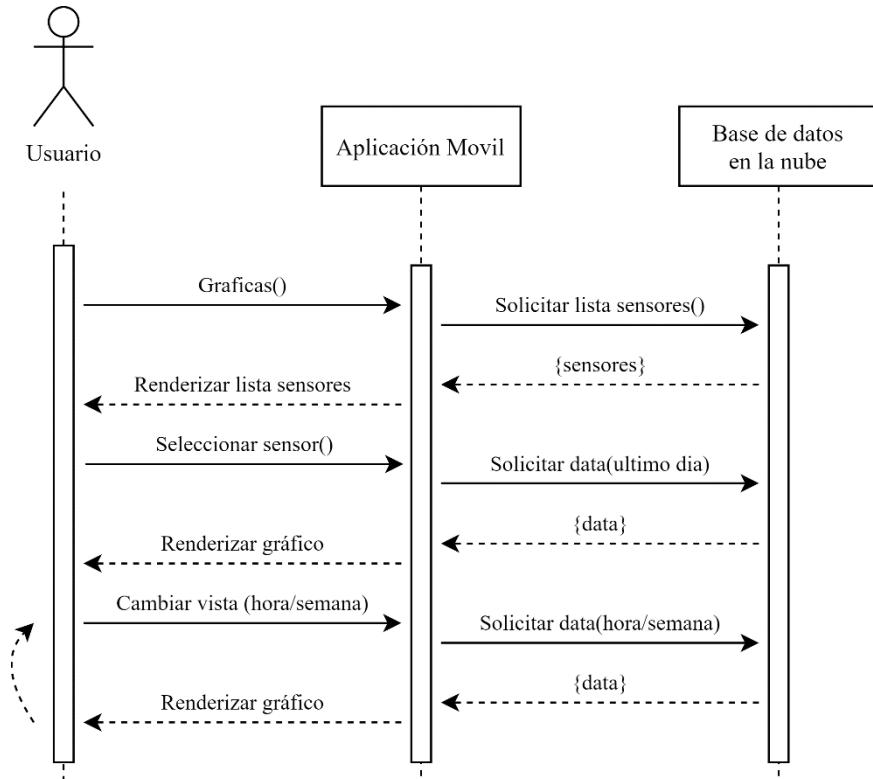


Figura 37: Diagrama de secuencia caso de uso "Ver gráficas de los sensores"

4.5.3. Configurar sensores a mostrar

Descripción

Tabla 12: Descripción caso de uso “Configurar sensores a mostrar”

Caso de uso:	Configurar sensores a mostrar
Actores:	Usuario
Precondición:	Cuenta con la aplicación instalada y conexión a internet, hay sensores enviando información.
Resumen:	El usuario entra a la aplicación para seleccionar los sensores de los cuales quiere ver información.
Tipo:	Secundario

Flujo normal de los eventos

Tabla 13: Flujo normal de los eventos caso de uso “Configurar sensores a mostrar”

Actor	Sistema
El usuario accede a la aplicación	
	Se muestran las secciones de la aplicación
El usuario selecciona la sección “Configuración”	
	Se muestran los sensores junto a <i>checkbox</i>
El usuario marca los sensores que desea ver	
	Se almacena la configuración en la base de datos, se actualizan las demás vistas de la aplicación con los sensores seleccionados

Diagrama de Secuencia

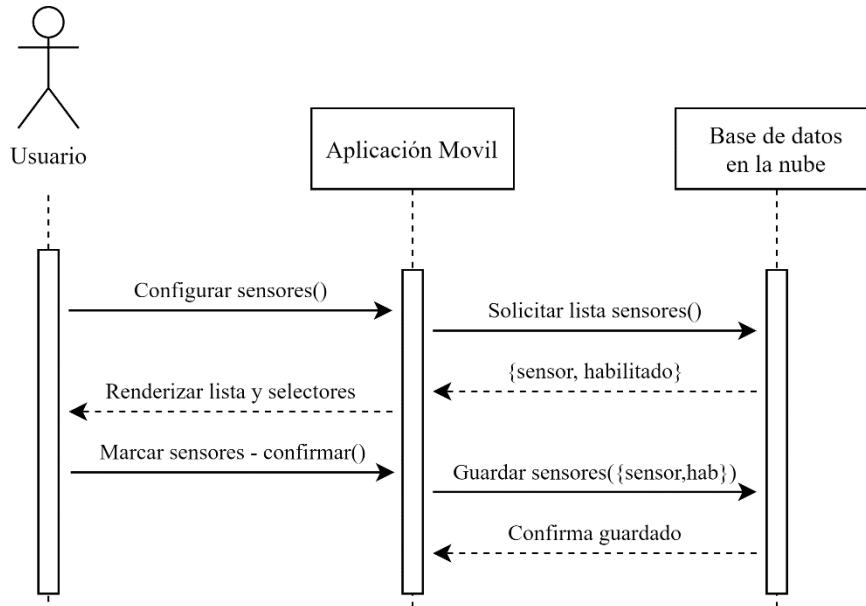


Figura 38: Diagrama de secuencia caso de uso "Configurar sensores a mostrar"

4.6. Modelo de procesos

En la Figura 39 se puede observar el modelo de procesos seguido en el prototipo para la extracción de datos, se pueden identificar las principales acciones realizadas y las relaciones entre ellas, lo que permite entender de mejor manera el funcionamiento del sistema.

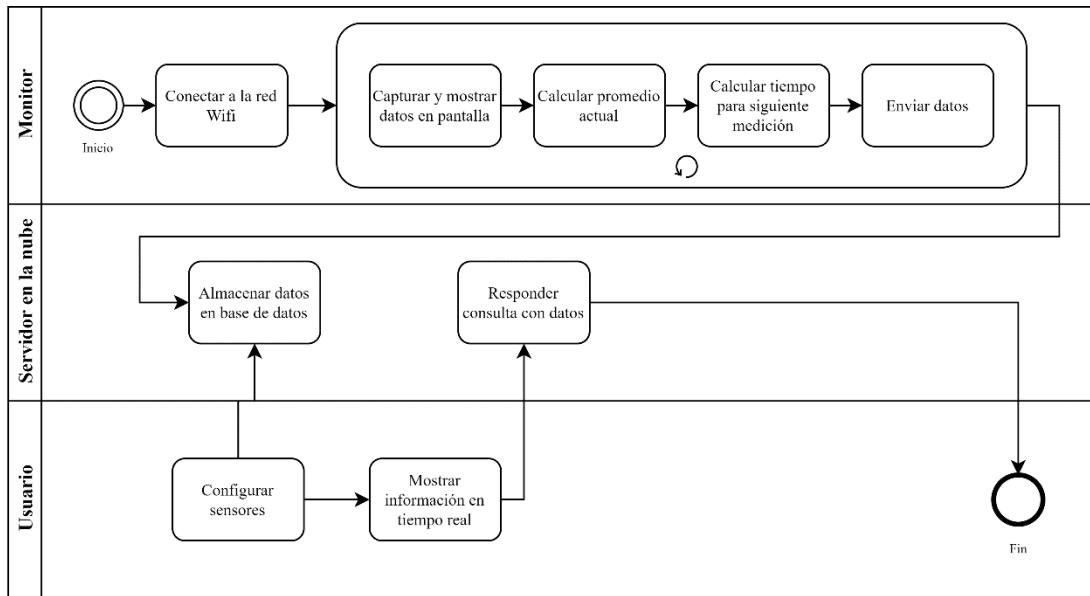


Figura 39: Diagrama de proceso del prototipo

4.7. Descripción de los actores del sistema

Como se pudo ver en el diagrama de proceso del prototipo en el sistema se pueden identificar 3 actores principalmente:

- Monitor: Estos son todos los dispositivos que se encargan de almacenar la información, estos pueden ser Arduino o NodeMCU que captan información de los sensores y envían la información mediante la conexión WiFi integrada.
- Servidor en la nube: Se encarga de almacenar la información enviada por los sensores y de responder a las solicitudes que efectúa la aplicación móvil, esto se encuentra gestionado por servidores de Google mediante la base de datos Firebase Realtime Database.
- Usuario: Es la persona (puede ser cualquier operario del invernadero o algún usuario que tenga acceso a la aplicación) quién puede revisar el estado en tiempo real de los sensores, configurar qué sensor se va a visualizar o ver gráficas.

4.8. Arquitectura del sistema

En esta sección se mostrará la solución desarrollada, esta se encuentra dividida principalmente en dos partes, la sección de Hardware donde se muestran los dispositivos utilizados y cómo se efectuaron las conexiones, la sección de software describe el funcionamiento de los prototipos y la aplicación móvil.

4.8.1. Hardware

4.8.1.1. Dispositivos utilizados

Como se indicó anteriormente, inicialmente se crearon 2 prototipos que efectúan la misma función, uno que utiliza un Arduino UNO como controlador y un ESP01 para la conexión WiFi (Figura 40 y Figura 41), y otro prototipo que utiliza un NodeMCU (Figura 42) como controlador (que incluye WiFi incorporado)

Ambos prototipos se instalaron dentro de cajas plásticas, el prototipo Arduino utiliza una caja sencilla, en cambio el prototipo NodeMCU utiliza una caja estanca resistente al agua, con una cara transparente, para que el sensor de luz pueda detectar la luz proveniente del exterior.

La versión del prototipo que utiliza un Arduino UNO se encuentra conectado directamente a la red eléctrica, utilizando un adaptador de 5V/1A, en cambio la versión que utiliza un NodeMCU se encuentra conectado a una batería recargable de 5000 mAh, marca Anker.

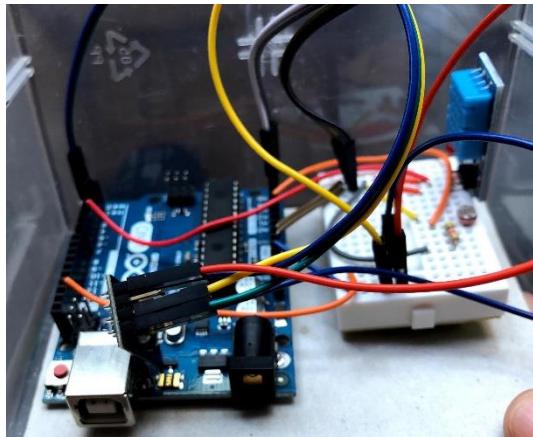


Figura 40: Prototipo usando Arduino UNO, vista interior



Figura 41: Prototipo usando Arduino UNO, vista exterior

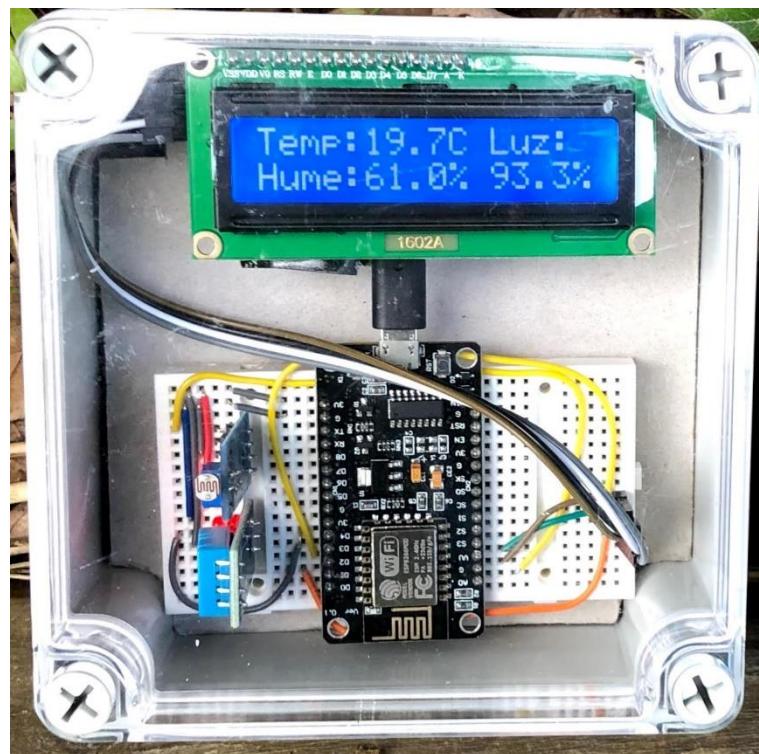


Figura 42: Prototipo usando NodeMCU, vista superior

4.8.1.2. Diagramas de conexión de dispositivos

Las siguientes figuras muestran las conexiones realizadas entre las placas y los sensores, dado que las imágenes son ilustrativas se utilizó una *protoboard* más grande que la que realmente se utilizó para que el circuito se vea de forma más clara.

En el prototipo utilizando Arduino (Figura 43) se tiene (de izquierda a derecha), un sensor de temperatura y humedad integrado DHT11, un sensor LDR con su respectiva resistencia y el ESP01, además del Arduino UNO y un *display* LCD de 16x2 caracteres.

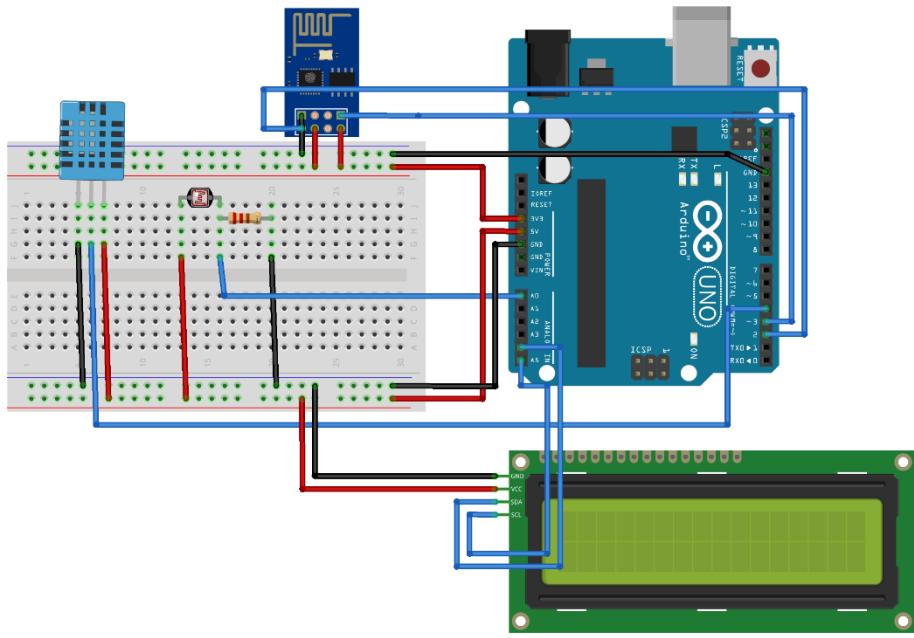


Figura 43: Diagrama de conexión usando Arduino UNO y ESP-01

En el prototipo utilizando NodeMCU (Figura 44) se tiene (de izquierda a derecha), un sensor LDR con conversor digital, un sensor de temperatura y humedad DHT11, un *display* LCD de 16x2 caracteres y el NodeMCU.

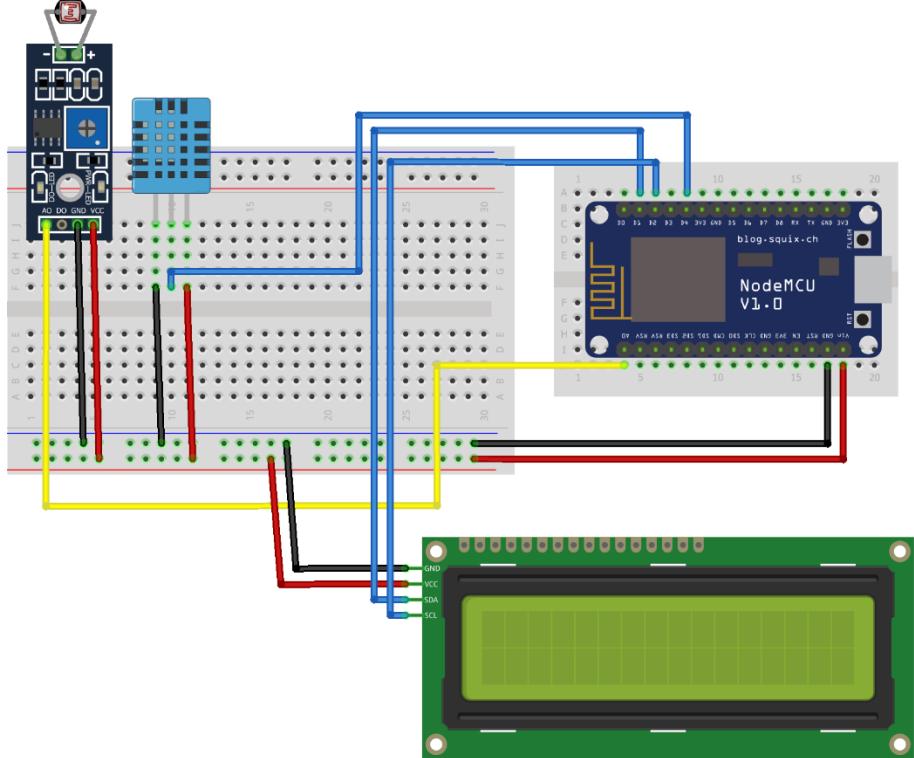


Figura 44: Diagrama de conexión usando NodeMCU

4.8.2. Software

4.8.2.1. Tecnologías utilizadas

Para el desarrollo de los scripts para ejecutar en los prototipos, así como también en la aplicación móvil se utilizaron varias tecnologías, las más importantes se describen a continuación:

Prototipos

- Arduino: Se utilizó el Arduino IDE y el lenguaje de programación Arduino, el Arduino IDE es una aplicación multiplataforma (Windows, Linux, Mac) que se puede utilizar para escribir y cargar programas en placas Arduino, aunque también se puede utilizar con placas de desarrollo de otros fabricantes (como el nodeMCU), el lenguaje de programación Arduino es una variación del lenguaje de programación C++, los programas escritos en este lenguaje se les llaman sketches y cuentan con la estructura que se muestra en la (Figura 45).

```
// Declaración de variables y librerías
void setup() {
    //Código de configuración
    //Este código se ejecutará solo una vez
}
void loop() {
    //Código de ejecución
    //Este código se ejecutará en un ciclo infinito
}
```

Figura 45: Formato sketch en lenguaje de programación Arduino

- Arduino JSON (ArduinoJson, s.f.): Es una librería para serialización y deserialización de documentos JSON, se encuentra especialmente creada para Arduino, por lo que se encuentra optimizada para este hardware (tiene bajo uso de memoria y de CPU), esta librería se utilizó para enviar los datos desde el ESP01 al Arduino en formato JSON, así como también enviar los datos en formato JSON a Firebase, utilizando la librería que se indica a continuación.
- FirebaseArduino (Firebase Arduino, 2016): Es una librería para simplificar la conexión a la base de datos Firebase Realtime Database desde clientes Arduino, puesto que Google no ofrece una librería oficial, ofrece funciones para realizar llamadas a la API REST de Firebase, por lo que todas las interacciones que se realizan entre los prototipos y la base de datos se realizan mediante el uso de esta librería.

Aplicación móvil:

- React Native (Facebook, s.f.): Es un framework escrito en Javascript, basado en el framework ReactJS, creado para utilizar componentes web para crear aplicaciones

móviles. Este framework se utiliza como base para la creación de la aplicación móvil. En la sección aplicación móvil de este capítulo se explica con mayor detalle el funcionamiento de este framework.

- Expo (Expo, s.f.): Es un framework para crear aplicaciones React Native, otorga herramientas para desarrollar, construir y desplegar aplicaciones. Permite hacer cosas como probar aplicaciones sin necesidad de instalarlas directamente en los dispositivos, enviar notificaciones Push y manejar almacenamiento local. Este framework se utilizó en el proceso de desarrollo para probar la aplicación desarrollada directamente en la Aplicación Móvil Expo, así como también generar los archivos ejecutables de las aplicaciones iOS y Android, evitando el uso de Xcode/Android Studio.
- Native Base (NativeBase, s.f.): Es una librería de componentes de interfaz nativos para aplicaciones React Native y Vue Native, cuenta con componentes como botones, listas, textos e iconos, que se renderizan como componentes nativos para cada sistema operativo, en la Figura 46 se visualiza como se muestra una interfaz sencilla en sistema operativo iOS y Android utilizando los mismos componentes.

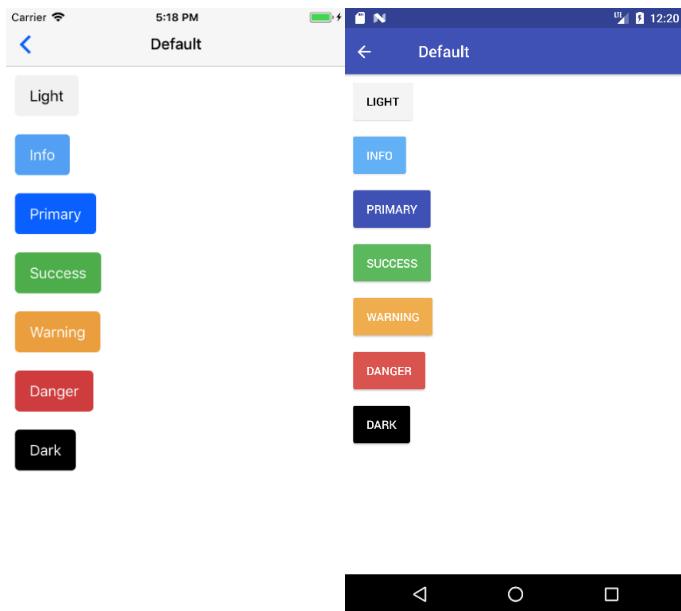
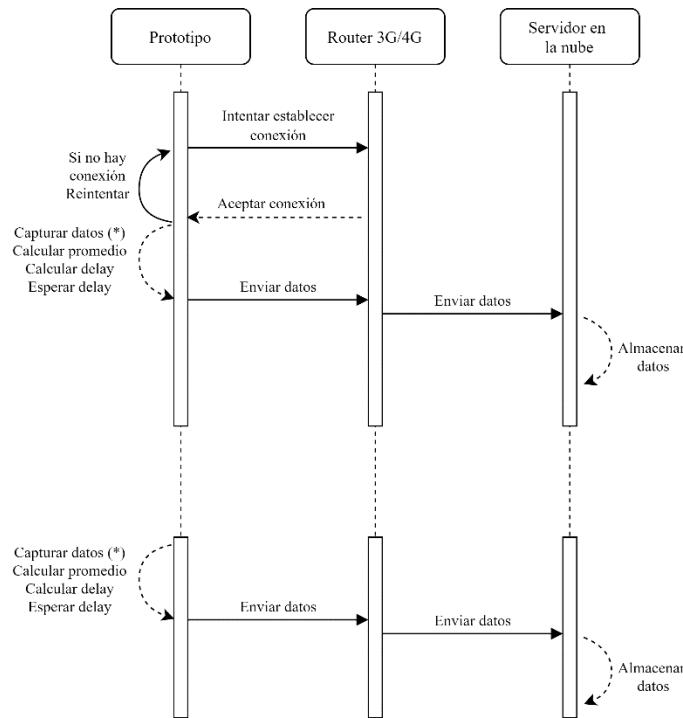


Figura 46: Interfaz generada por Native Base en sistema iOS y Android

Para la realización de las consultas a la base de datos que se requieren en la aplicación se hace uso de la API web de Firebase (Firebase, Firebase JavaScript SDK Reference, s.f.), esto es directamente ofrecido por Google.

4.8.2.2. Diagrama de secuencia del prototipo

En la Figura 47 se muestra el funcionamiento del prototipo, aplica para la versión del prototipo utilizando Arduino, así como también para la versión que utiliza el NodeMCU.



* En NodeMCU la información capturada se envía directamente al router vía WiFi, en Arduino se envía un JSON mediante conexión serial con los datos al ESP01, quien envía los datos al router.

Figura 47: Diagrama de secuencia del prototipo

Hay que recordar que los diagramas de secuencia correspondientes a la aplicación móvil se encuentran en la sección casos de uso.

4.8.2.3. Firebase

Como ya se indicó anteriormente Firebase cuenta con una multitud de servicios, pero en este proyecto el relevante es Realtime Database, ya que es la base de datos que se utiliza para almacenar los datos que generan los prototipos.

Los datos son enviados y recibidos por los clientes en formato JSON, estos datos pueden ser accedidos desde una visualización web que provee el servicio (ver Figura 48). Además, los datos pueden ser descargados para efectuar análisis de datos, en el capítulo de pruebas se efectuaron algunos análisis a partir de datos exportados.

The screenshot shows the Firebase Realtime Database interface. On the left is a sidebar with various icons. The main area has tabs for 'Datos', 'Reglas', 'Copias de seguridad', and 'Uso'. Below these tabs, there's a message about activating static mode. The database structure is displayed under 'test-arduino-d6faa':

```

test-arduino-d6faa
  nameSensors
    sensor1
      2020-09-17T22:50:23Z
      2020-09-17T22:50:37Z
      2020-09-17T22:50:47Z
        humidity: 68
        light: 66.01563
        temperature: 28.1
        timestamp: 1600383000
    sensor2
    sensor3
    sensor4
    sensor5
  
```

At the bottom, it says 'Ubicación de la base de datos: Estados Unidos (us-central)'.

Figura 48: Vista de los datos en la interfaz de Firebase Realtime Database

También es posible añadir autenticación, lo que será de utilidad en futuras iteraciones, cuando sea necesario restringir el acceso.

4.8.2.4. Aplicación móvil

Como ya se indicó anteriormente, para desarrollar la aplicación móvil se utilizó el framework React Native, un framework para desarrollar aplicaciones móviles utilizando una sola base de código.

El funcionamiento de este framework se basa en el uso del lenguaje Javascript, donde se define el código HTML, que es convertido en componentes nativos de interfaz, esto es llamado el *VirtualDOM* (Jimenez, 2019) (ver Figura 49).

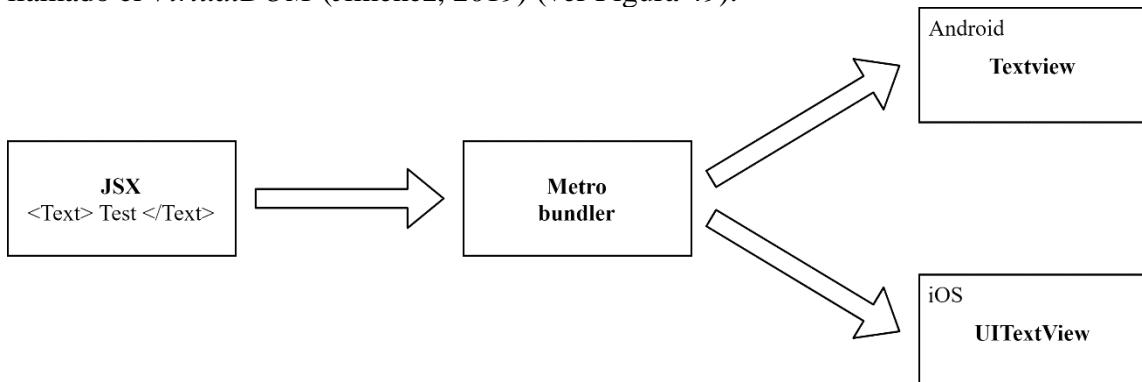


Figura 49: *VirtualDom* de React Native

Esto permite que el rendimiento de la interfaz sea equivalente al de una aplicación nativa de cada sistema operativo, además de un diseño diferenciado para cada sistema.

Aun cuando el rendimiento de la interfaz es nativo, hay elementos que no se ejecutan de forma nativa, sino que se ejecuta mediante dos hilos, uno que ejecuta módulos nativos como la interfaz y el uso de servicios del sistema operativo y otro que ejecuta una máquina virtual de Javascript. La comunicación entre ambos hilos se realiza mediante un bridge (ver Figura 50).

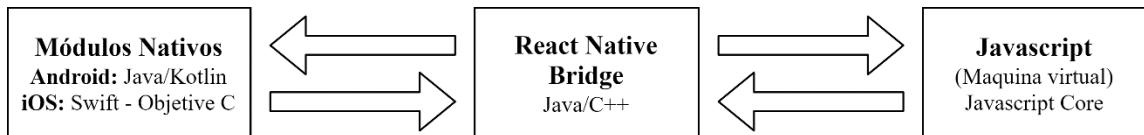


Figura 50: React Native Bridge

A continuación, se muestra en más detalle el desarrollo de la aplicación móvil. La Figura 51 muestra los principales archivos del proyecto.

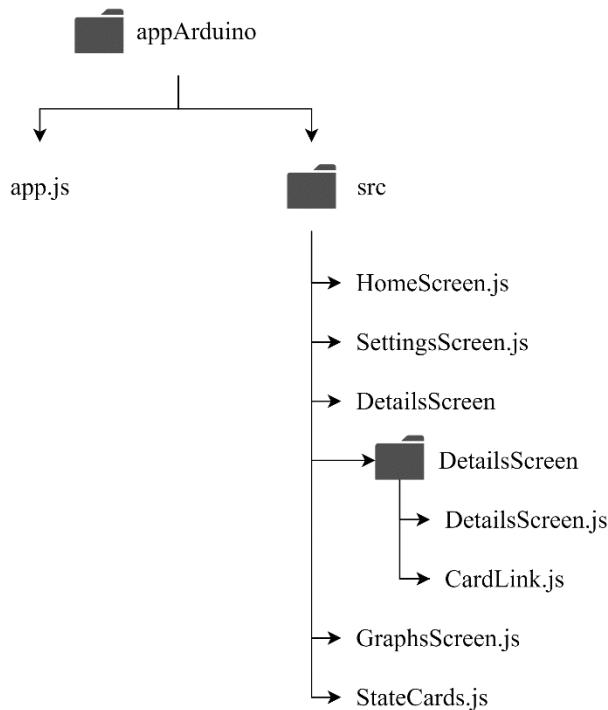


Figura 51: Estructura de archivos del proyecto aplicación móvil

Como se ha indicado anteriormente, la aplicación cuenta con tres vistas principales, la vista de datos en tiempo real, gráficos y configuración, el funcionamiento de cada una de ellas se detalla a continuación.

Navegación

Eso se controla en el archivo App.js, se utiliza la librería React Navigation, que proporciona navegación, ya que React no maneja esto, como lo haría un navegador web. La navegación por las distintas vistas se maneja mediante iconos seleccionables en la parte inferior de la aplicación, como se muestra en la Figura 52.

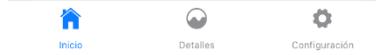


Figura 52: Iconos de navegación en la aplicación móvil

Vista Inicio

Esto se controla en los archivos HomeScreen.js y StateCards.js, donde se obtiene la última información disponible de los sensores, el usuario puede deslizar para ver la información si son varios sensores y puede deslizar hacia abajo para actualizar la información. También se almacena la información obtenida la última vez cuando no hay conexión a internet. En la Figura 53 se muestra la vista tanto en sistema Android como iOS, respectivamente.

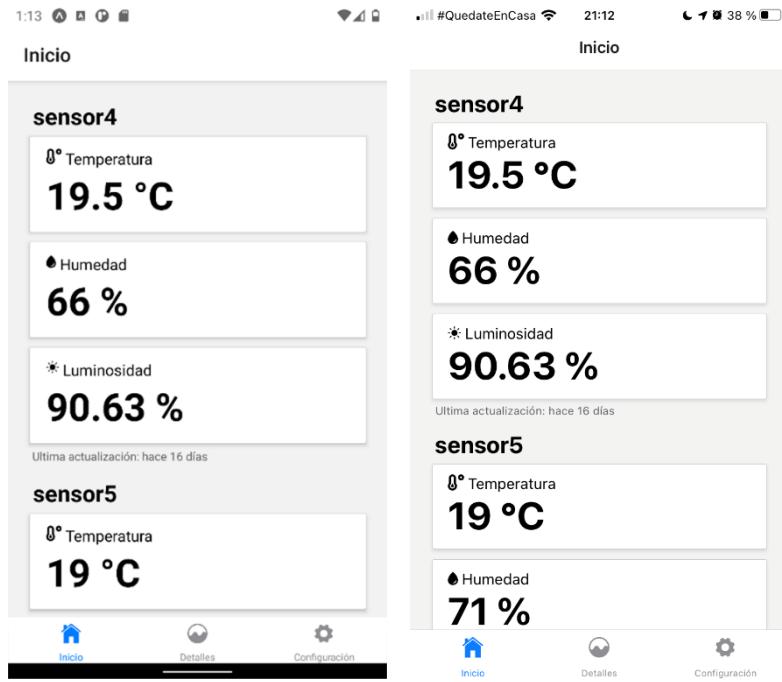


Figura 53: Vista inicio de la aplicación

Vista Gráficos

Esto es controlado en los archivos DetailsScreen.js, CardLink.js y GraphsScreen.js, estos se encargan de obtener los sensores disponibles, gestionar las vistas de gráficos (hora, día, semana), obtener la información necesaria y desplegar los gráficos, en la Figura 54 se muestra un ejemplo de estas vistas.

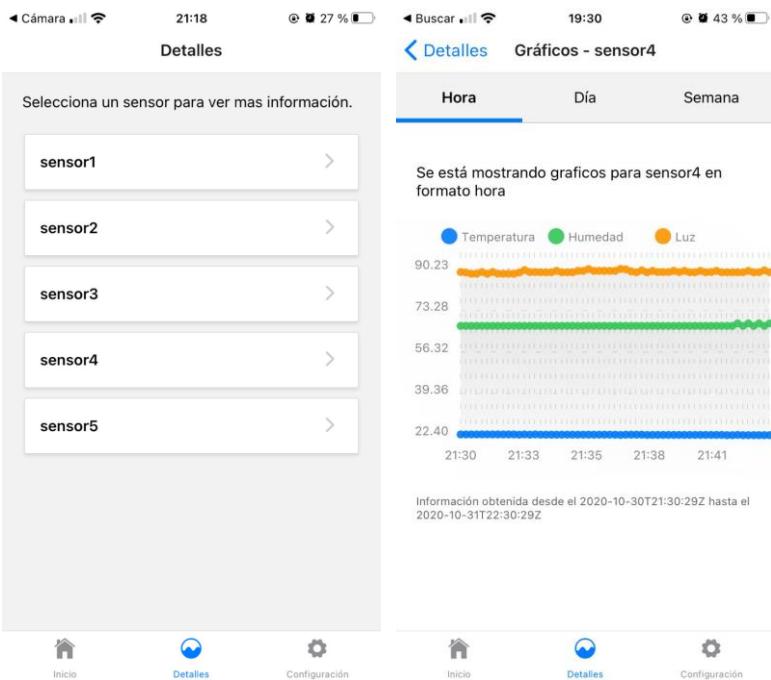


Figura 54: Vista gráficos de la aplicación

Vista Configuración

Esto es controlado por el archivo SettingsScreen.js, este archivo se encarga de obtener los sensores almacenados en la base de datos, listarlos y mostrar selectores para cada uno de ellos, cuando se marcan los seleccionados esta información se almacena en la base de datos, para definir los que se mostrarán en la aplicación. En la Figura 55 se muestra el funcionamiento de esta vista.

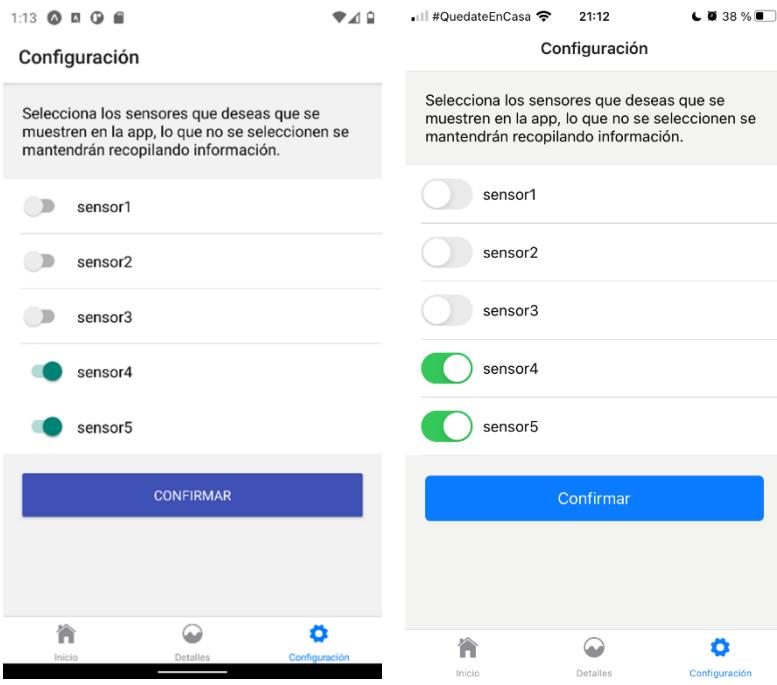


Figura 55: Vista configuración de la aplicación

5. PRUEBAS

En este capítulo se encuentran pruebas realizadas al prototipo en condiciones simuladas, puesto que no fue posible trabajar directamente en el invernadero, debido a la situación sanitaria actual.

5.1. Validación con soluciones existentes

Se realizó una revisión bibliográfica para determinar que la solución desarrollada tiene un grado de confiabilidad alto, tanto en la recolección de datos, como en su funcionamiento, esto nos permite determinar si la solución que se desarrolló es lo suficientemente confiable para usarla de manera permanente. Se seleccionaron 3 artículos que guardaban cierta similitud con este proyecto (se usan microcontroladores o sensores similares).

En (Palacios, y otros, 2017) se realiza un sistema similar, utilizando un Arduino y sensores DHT, además de sensores de luz LDR, como diferencias se tienen que se implementó un sistema de alertas; el cual es fácil de implementar dentro de la aplicación móvil, además utiliza la tecnología Zigbee, la cual se descartó en este proyecto debido a su menor alcance y mayor costo. También se usó una plataforma en la nube.

Para validar el funcionamiento se hizo un estudio de las condiciones en que crecen o no las plantas, donde los sensores midieron valores fuera de los rangos las plantas murieron o no crecieron adecuadamente, lo que demostró que los valores obtenidos eran correctos y la solución fue confiable.

En (Espinoza, Villavicencio, & Díaz, 2014) también se desarrolló un sistema similar, utilizando un Arduino como microcontrolador, en este caso además de medir información al interior del invernadero se consideró medir meteorología al exterior del invernadero (mediante el uso de una veleta y un anemómetro). La medición de temperatura y humedad se realizó con un sensor de similares especificaciones al DHT11 (HMZ-433A1).

Al ser un trabajo relativamente antiguo (2014) se tiene que se utilizó otras tecnologías, como pantallas con conector RCA o el uso de un sistema que requiere un computador conectado al Arduino para almacenar información, un punto a favor es el uso de una tarjeta microSD que permite obtener la información del sensor sin necesidad de obtenerla de una base de datos. Se compararon algunos sensores comerciales y se llegó a la conclusión que los datos tienen precisión similar, aunque los sensores open hardware tienden a actuar más lento, respecto a esto último, para el prototipo que se desarrolla en el presente proyecto se realizan pruebas en las siguientes secciones.

Uno de los principales puntos a favor del uso de soluciones open hardware es la reducción de costos, en un porcentaje muy alto, puesto que las soluciones comerciales son mucho más caras (del orden de 20 veces mayor).

En (Mora & Rosas, 2019) también se realizó una solución similar, utilizando un NodeMCU y sensores DHT11, en este caso se hicieron comparaciones con otros

dispositivos validados que se utilizaban en un invernadero, donde se detectó que el sensor dht11 tiene una precisión de $\pm 5\%$.

Con respecto al NodeMCU se comprobó que el dispositivo es confiable para las mediciones y subida de datos sin errores. También se utilizó un servidor en la nube, pero en este caso de Amazon AWS (EC2), donde se destacó la confiabilidad y alta disponibilidad de estos servicios. Los datos generados permitieron tomar mejores decisiones para el cultivo de frijoles (porotos).

En base a estos estudios se puede indicar que la solución desarrollada para el presente proyecto es confiable, debido a los buenos resultados obtenidos con soluciones muy similares, puesto que los sensores no presentan errores en las mediciones que pudieran generar problemas en la producción, así como los microcontroladores y tecnologías de red pueden funcionar sin problemas durante mucho tiempo. Finalmente hay que recalcar que los sensores open hardware son una muy buena opción para ahorrar costos, en especial en pequeños productores.

5.2. Plantas y crecimiento

Se efectuaron pruebas con 2 prototipos, que midieron datos para el crecimiento de plantas en 2 ambientes distintos, con el objetivo de verificar que las variables que se están midiendo pueden afectar al crecimiento de las plantas, aun cuando ya se tienen multitud de estudios que comprueban que esto es cierto, estas pruebas permiten comprobar la fiabilidad de los sensores y la capacidad de los microcontroladores de funcionar de forma continua durante largos períodos de tiempo.

Primero se realizó una búsqueda para encontrar las plantas que pueden usarse en experimentos y que pueden crecer más rápidamente, se encontró que estos son el trigo y semillas de poroto, estos fueron sembrados en pequeños frascos de vidrio, donde fuera posible ver su crecimiento de forma fácil a lo largo del tiempo, se instalaron en 2 ubicaciones distintas, uno al interior de una casa con iluminación durante las 24 horas del día y otro a la intemperie con iluminación natural, cada uno de estos frascos acompañado de un prototipo que capture los datos. En la Figura 56 se encuentra una foto de cada dispositivo.



Figura 56: Dispositivos midiendo al interior y a la intemperie

Las pruebas se realizaron durante un plazo de 8 días (suficiente para verificar que las plantas hayan crecido), entre el 20 y 28 de octubre del 2020, ambos prototipos en las mismas fechas.

En la Figura 57 se encuentran graficados los datos obtenidos por el prototipo ubicado en la intemperie.

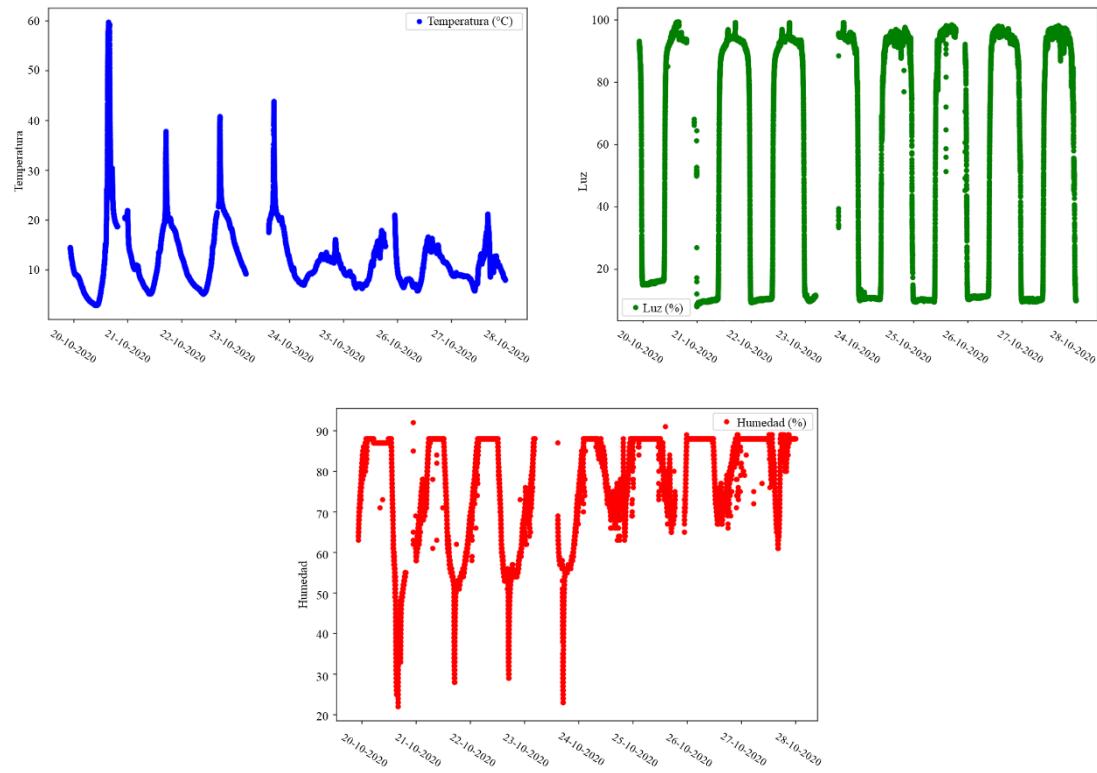


Figura 57: Datos obtenidos por el prototipo en la intemperie

Como se puede ver en los primeros 4 días la temperatura llega a niveles inusualmente altos y la humedad a niveles muy bajos, esto es debido a que la caja resistente al agua provoca un efecto invernadero muy exagerado cuando se encuentra directamente a la luz del sol, por lo que las condiciones a las que realmente se ve expuesta las plantas fueron las que se ven en los últimos 4 días aproximadamente.

En la Figura 58 se encuentran graficados los datos obtenidos por el prototipo ubicado al interior de casa.

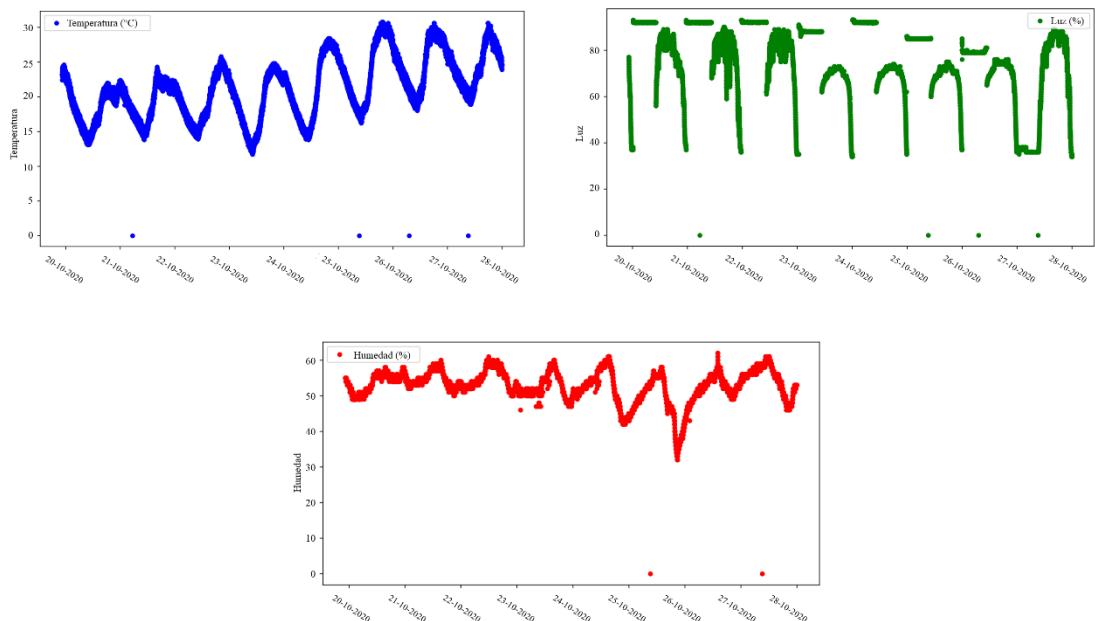


Figura 58: Datos obtenidos por el prototipo en un al interior de una casa

Resumiendo, los datos obtenidos se pueden ver en la tabla.

Tabla 14: Datos obtenidos por ambos prototipos

		Interior	Exterior
Temperatura (°C)	Media	21.24	11.94
	Mediana	21.1	10.5
	Moda	21.4	8.7
	Desviación estándar	4.39	6.52
Humedad (%)	Media	52.95	77.21
	Mediana	53	84
	Moda	55	88
	Desviación estándar	4.38	13.54
Luz (%)	Media	76.61	56.54
	Mediana	79	85.44
	Moda	92	10.35
	Desviación estándar	15.36	40.06

Como se puede ver las condiciones son considerablemente distintas entre ambos prototipos utilizados, donde al exterior las temperaturas son más bajas, la humedad más alta y la luz más baja, esto debido a que en el interior se tenía calefacción y luz artificial las 24 horas. A continuación, se analizará el crecimiento que tuvieron las plantas en los días que transcurrió el experimento.

En la Figura 59 se muestran fotografías del crecimiento de las plantas a los 3 días de iniciado el experimento.



Figura 59: Plantas a 3 días de iniciado el experimento

En la figura de la izquierda se ve el crecimiento de las plantas al interior y a la derecha al exterior, como ya se ve las plantas crecieron más al interior.

En las Figura 60 y Figura 61 se muestran fotografías del crecimiento de las plantas luego de 8 días de iniciado el experimento.



Figura 60: Crecimiento de plantas al exterior, luego de 8 días



Figura 61: Crecimiento de plantas al interior, luego de 8 días

Como se puede ver en las fotografías, la diferencia de crecimiento es apreciable, en ambas plantas, los porotos en el exterior apenas tienen brotes mientras que los que crecieron en el interior las plantas ya se están desarrollando, con raíces y planta con forma. En las semillas de trigo (fotografías de la derecha), se puede ver que los brotes que crecieron al exterior dieron brotes de alrededor de 1cm, con raíces de 3cm aprox., en cambio los que crecieron al interior tienen brotes de más de 10cm, con raíces de unos 10cm.

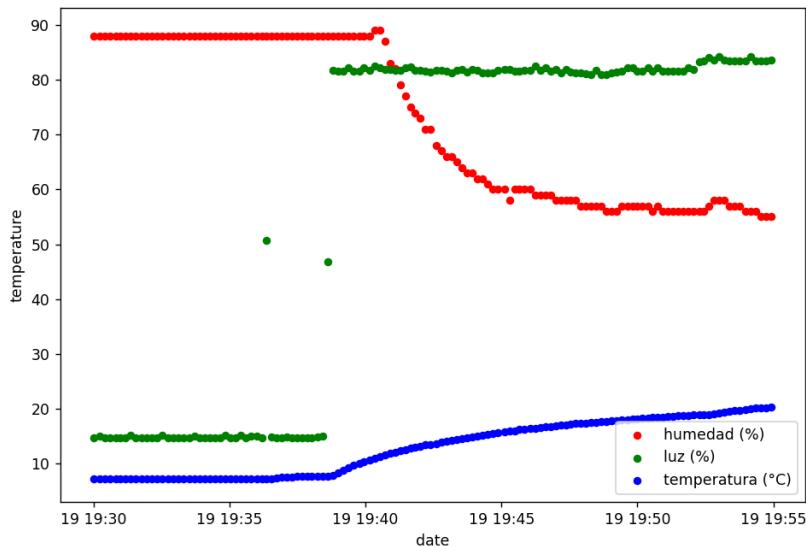
A partir de la prueba realizada se puede concluir que los prototipos están preparados para obtener información durante largos períodos de tiempo, aunque existieron problemas con el que se alimentaba con baterías, puesto que la duración de batería no fue suficiente y se debió recargar varias veces.

También se puede concluir que los parámetros que se controlan si afectan al crecimiento de las plantas, puesto que las plantas del interior tuvieron un crecimiento mucho más acelerado, donde las condiciones medidas también fueron diferentes, con temperaturas más altas y niveles de humedad más bajos. Por lo tanto, el prototipo sirve para medir estas variables y estas deberían tener relación con el crecimiento de las plantas.

5.3. Retardo en toma de datos

Para determinar el valor se realizó una prueba sencilla, insertar los prototipos en un refrigerador con una temperatura aproximada de 5 °C y sin luz, para luego retirarlo al exterior, con una temperatura aproximada de 20°C y luz natural. El objetivo de esto es determinar si existe algún retraso en las mediciones cuando existen cambios bruscos de temperatura y de luz (cuando ocurre un problema en la iluminación del invernadero, por ejemplo).

En la Figura 62 se muestran los resultados del experimento.



En la gráfica se puede ver claramente cuando ocurre el cambio, ya que el sensor de luz responde inmediatamente desde un 15% hacia un 80% aproximadamente. Se puede ver que la temperatura y humedad no varían instantáneamente, la variación de humedad demora siete minutos, pero demora unos dos minutos en comenzar a cambiar, en cambio la temperatura demora aproximadamente 20 minutos para llegar a la temperatura de las nuevas condiciones. En la Tabla 15 se pueden ver los valores iniciales y finales obtenidos.

Tabla 15: Valores iniciales y finales experimento "Prototipo en refrigerador"

Hora	Temperatura	Humedad	Luz
19:34	7,2 °C	88 %	14,7 %
19:54	20 °C	56 %	84,1 %

A partir de los datos obtenidos se puede indicar que estos retardos son inherentes al funcionamiento de los sensores y no a retardos en la obtención de los datos por los algoritmos, además los casos obtenidos en el caso de la iluminación son casi instantáneos, lo que es positivo, en la temperatura y humedad no es así, pero no se espera que haya cambios tan bruscos de temperatura o humedad en las condiciones reales.

5.4. Consumo de energía

Esta prueba se obtuvo directamente del uso del dispositivo y no se utilizaron instrumentos para determinar el consumo de energía en tiempo real, por ejemplo.

Se realizaron pruebas con el uso del prototipo NodeMCU, utilizando una batería recargable de 5000 mAh/18.5 Wh donde no se utilizaba ninguna optimización en el uso de energía (el NodeMCU cuenta con modos específicos para esto), se obtuvieron datos cada 10 segundos desde los sensores, enviándose directamente a la base de datos.

En un caso se mantuvo la pantalla LCD conectada y activa durante todo el tiempo, donde se obtuvo una autonomía aproximada de 27 horas, en cambio prescindiendo del uso de la pantalla se obtuvo una autonomía aproximada de 50 horas.

Aun cuando la batería utilizada es relativamente pequeña y podría mejorarse la autonomía utilizando baterías con mayor capacidad, el tiempo de funcionamiento fue bajo, por lo que nos lleva a pensar que se requiere realizar optimizaciones en este aspecto, reduciendo el envío de datos redundantes y utilizando los modos de ahorro de energía que tiene el NodeMCU.

5.5. Limitantes

A continuación, se detallan posibles limitantes que se detectaron en el desarrollo del proyecto.

Capacidad de la base de datos

A partir de los datos ya generados en la base de datos se determinará el espacio que ocupa un conjunto de datos generado por cada sensor en cada iteración, un conjunto de datos consiste en lo que se muestra en la figura:

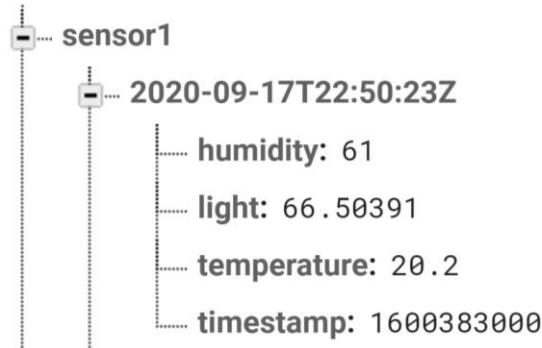


Figura 63: Conjunto de datos enviado por los sensores

Es posible hacer una estimación del tamaño de este conjunto mediante la expresión:

$$\text{Tamaño conjunto} = \frac{\text{Tamaño total datos}}{\text{Número de datos}} \quad (1)$$

Se generó un script para determinar el número de datos obtenidos y se obtuvo el tamaño total de los datos desde Google Cloud (ver Figura 64).

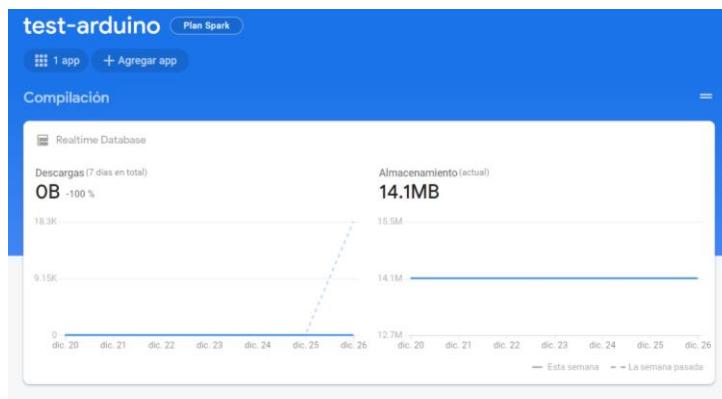


Figura 64: Almacenamiento usado en Google Cloud luego de las pruebas

Como se puede ver en la figura se está ocupando un total de 14,1MB, se tienen un total de 153.165 datos, por lo que aplicando la expresión (1) se tiene:

$$Tamaño \ conjunto = \frac{14.100 [KB]}{153.165} \quad (2)$$

$$Tamaño \ conjunto = 0.092 [KB] \ ó 92 [Bytes] \quad (3)$$

Se tienen que cada dato ocupa 92 Bytes, esto significa que se puede tener un sensor que envié alrededor de 55 millones de conjuntos de datos sin pagar por un plan en Google Cloud (recordar que el límite de almacenamiento gratuito es 1GB)

Como se sabe que no se tendrá un solo sensor y suponiendo que cada uno de estos sensores enviará datos cada un minuto en la Tabla 16 se tienen algunas simulaciones de cuantos sensores se pueden tener por cuánto tiempo sin pagar.

Tabla 16: Límite de envío de datos utilizando plan gratis en Firebase

Número de sensores	Tiempo enviando datos
1	104 años
10	10.4 años
50	1.7 años
100	1 año
1.000	37 días
5.000	7 días
10.000	4 días

Suponiendo que se encontrara en las últimas 3 situaciones, 1.000, 5.000 o 10.000 sensores, en un año se tendría que incurrir en los costos que se indican en la Tabla 17, aproximadamente.

Tabla 17: Costos al enviar datos durante un año

Número de sensores	Costo al enviar datos en un año
1.000	50 USD
5.000	250 USD
10.000	500 USD

Esto nos indica que el uso de la base de datos Firebase Realtime Database no debería ser una limitación para el sistema, puesto que se prevé la instalación futura de decenas o cientos de sensores, donde los costos no deberían ser muy altos.

Capacidad de la red WiFi

Para la realización de las pruebas se utilizó un router 4G Huawei B310, en una zona rural, por lo que los niveles de congestión de la red WiFi son equivalentes a las que se obtendrían en el invernadero.

Algunas de las especificaciones técnicas de este router son las que se muestran en la Tabla 18:

Tabla 18: Especificaciones técnicas Huawei B310

Fabricante y modelo	Huawei – B310s 518
Bandas 3G/4G	3G: 850/1700 AWS/1900/2100 4G: B1, B2, B4 AWS, B5, B7, 28
Conectividad WiFi	IEEE 802.11b/g/n (2.4 GHz)
Puertos	Ethernet, línea telefónica, dos conectores para antenas externas.
Velocidad Máxima	LTE Cat 4 150/50 Mbps
Máximo dispositivos conectados	32
Alcance máximo teórico	200 metros

Como se ve en la tabla solo se soportan 32 dispositivos conectados a la red WiFi, por lo que se deberían utilizar repetidores o router 4G industriales para soportar un número mayor de sensores.

Como cada dato enviado ocupa un ancho de banda cercano a 100 Bytes con una conexión estable de 0.0256 megabits sería suficiente para poder enviar la información de estos 32 sensores sin pérdida de datos, aun cuando se enviaran datos cada segundo por todos los sensores, lo que de antemano se sabe que no será así. Es conocido que las velocidades de las redes 4G son muy superiores a esto por lo que no sería una limitante con esta cantidad de sensores.

También se realizaron pruebas para determinar la distancia máxima que soportan los prototipos conectados a este router, esto se realizó mediante un pequeño script que imprime la señal de la red WiFi en pantalla, en la Figura 65 se encuentran los datos obtenidos.

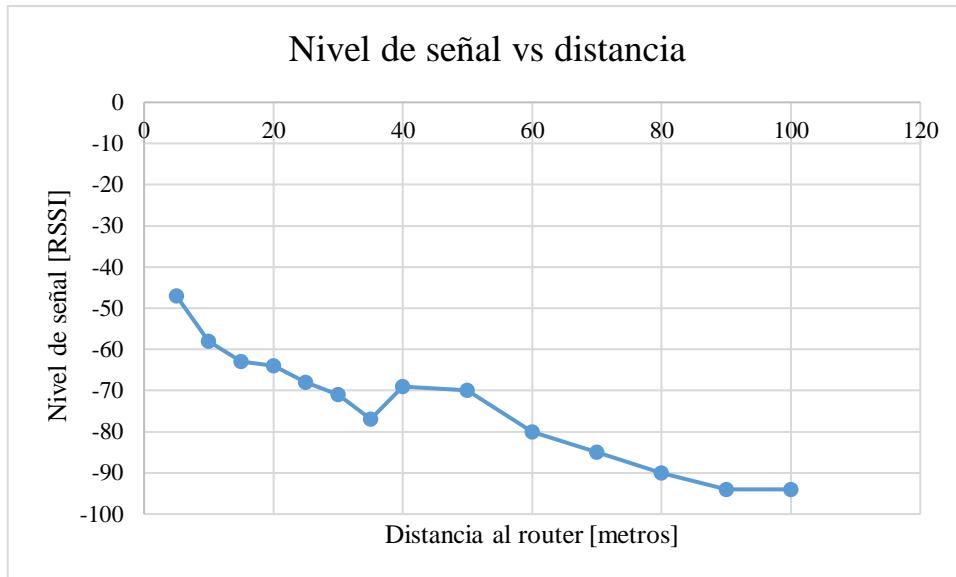


Figura 65: Niveles de señal de la red WiFi

Según la escala RSSI un nivel entre 0 y -80 se puede establecer conexión, aunque en las pruebas realizadas el dispositivo envía datos hasta los 80 metros, la señal era muy débil e inestable luego de los 100 metros y a los 120 metros ya no era detectable por el dispositivo. Esto nos indica que un router de este tipo debería funcionar en el invernadero, puesto que las dimensiones son menores a los 80 metros, estas distancias con la red WiFi se consiguen debido a que la congestión en zonas rurales es muy baja.

6. MEJORAS

Dado que la primera versión del prototipo tiene aspectos que se pueden mejorar, estos fueron abordados para la realización de una segunda versión, se consideran principalmente en 2 aspectos, el cambio de sensores y mejoras en los algoritmos utilizados para la medición, los cuales se describen en las siguientes secciones.

6.1. Cambios de sensores

Debido a los datos obtenidos en las pruebas realizadas se detectó que existen variaciones en las mediciones cuando se utiliza el sensor DHT11 en la caja estanca y se encuentra expuesto directamente a la luz del sol, además que las condiciones de las plantas no son necesariamente las mismas que en la caja que se encuentra a unos centímetros, por lo que se decidió añadir dos sensores que realizan básicamente la misma función, un sensor de humedad de suelo capacitivo y un sensor DS18B20, estos permiten ser instalados directamente en el ambiente que se ven expuestas las raíces, puesto que son resistentes al agua.

Estos sensores se añadieron como extras al DHT11 en el prototipo que utiliza Arduino, el diagrama de conexión es el que se muestra en la Figura 66.

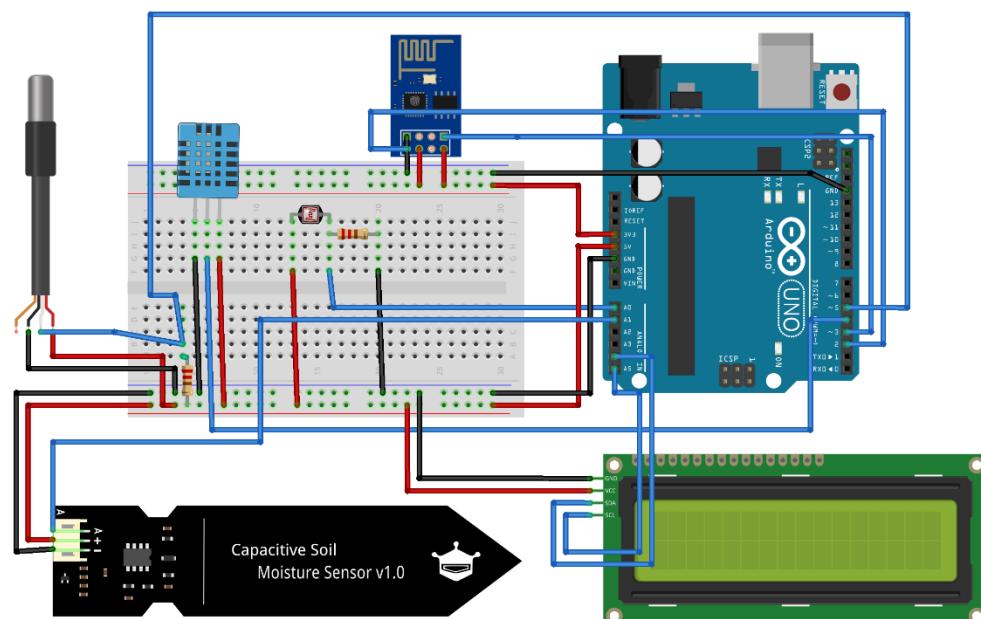


Figura 66: Conexión del prototipo con sensores extra

Desde la aplicación móvil se visualizan estos datos cuando están disponibles, en la Figura 67 se muestra un sensor que cuenta con estos datos y otro que no.

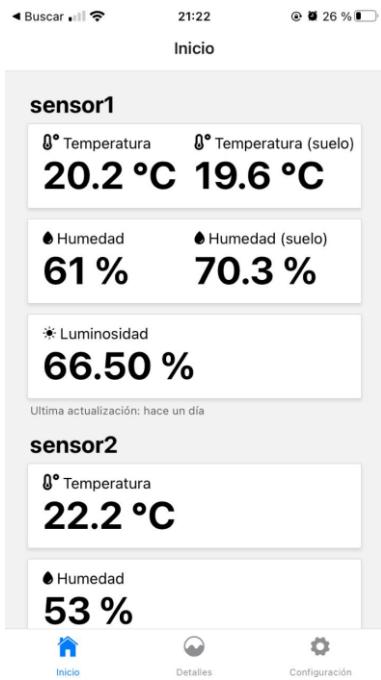


Figura 67: Datos mostrados en la aplicación con sensores extra

Los datos obtenidos por estos sensores extra también serán almacenados en la base de datos y serán de utilidad para futuros análisis.

6.2. Mejoras de algoritmos

Como se indicó en el capítulo anterior, los datos fueron obtenidos utilizando intervalos fijos de tiempo de 10 segundos, esto tiene principalmente 2 implicancias

- El consumo de energía: cuando el microcontrolador requiere obtener datos consume mayor energía y más aún cuando esta es enviada por WiFi, actualmente no se considera el uso de modos de ahorro de energía, por lo que esto es un aspecto que considerar en la revisión de su funcionamiento.
- La acumulación de información redundante: es sabido que no existirán grandes variaciones en los datos obtenidos del sensor para los parámetros indicados, por lo que se debe estudiar una nueva manera para definir los intervalos de tiempo en que se obtienen los datos, donde se considere obtener más datos cuando existe una variabilidad más alta y menos datos cuando esta sea más baja.

Lo indicado en los puntos anteriores se estudiará y se aplicará al prototipo realizado, los cambios realizados se detallan a continuación.

6.2.1. Modos de ahorro de energía

El NodeMCU cuenta con tres modos de ahorro de energía (ESPloradores, 2017), los que se describen a continuación:

- *Deep-Sleep*: Este es el modo que genera mayor ahorro de energía, puesto que deja la placa en suspensión, la única parte que funciona en este modo es el reloj en tiempo real, que reinicia la placa luego de ocurrir el tiempo de reposo.
- *Modem-Sleep*: este modo de ahorro permite desactivar la conexión WiFi establecida con un punto de acceso (router), cuando no sea necesario su uso y permite volver a activarla cuando se necesite.
- *Light-Sleep*: este modo de ahorro permite mantener la conexión WiFi, pero reduce el consumo de energía en los momentos en los que no hay envío de información, este es el modo que reduce en menor medida el consumo de energía.

Para el prototipo utilizando el NodeMCU se decidió utilizar el modo *Deep-Sleep* debido a que es el método que genera un mayor ahorro energético, lo que permitirá recargar las baterías luego de un número mayor de días, además se ajusta a las necesidades del proyecto, puesto que puede ajustarse el tiempo de reposo de forma dinámica, almacenando datos en la memoria RTC (Cherukupalli , 2019), que cuenta con una capacidad de 8kb, esta memoria solo se borra al reiniciar el dispositivo mediante el botón integrado o desconectarlo de su fuente de energía

Según las especificaciones del microcontrolador NodeMCU su consumo de energía medio en modo activo es de unos 80mA y en modo *deep-sleep* es de unos 8mA (Del Valle L. , 2020), es posible calcular un tiempo aproximado de duración de batería mediante la siguiente expresión.

$$Duración \text{ (horas)} = \left(\frac{\text{Capacidad batería (mAh)}}{I_{\text{promedio}}} \right) \times 0,7 \quad (4)$$

El valor 0.7 se aplica debido a la probabilidad de que la capacidad de la batería no sea el 100%. Aplicando esta expresión a la batería que se utiliza en el prototipo (de 5000mAh) se tiene.

$$Duración \text{ (horas)} = \left(\frac{5000}{80} \right) \times 0,7 = 43,7 \quad (5)$$

Como se puede ver la duración de la batería estimada es similar a la autonomía obtenida sin utilizar la pantalla LCD (50 horas). A partir de esto es posible obtener una autonomía teórica, considerando que el dispositivo se mantendrá encendido un segundo durante cada minuto (enviando un conjunto de datos por minuto) el consumo medio sería de unos 9.2mA, si se aplica a la expresión.

$$Duración = \left(\frac{5000}{9,2} \right) \times 0,7 = 380 \text{ horas ó } 15,8 \text{ días} \quad (6)$$

Si se utilizara una batería con una capacidad mayor, como 10000mAh se podría lograr una autonomía mayor a un mes, lo que permitiría tener los dispositivos funcionando por un tiempo prolongado sin la necesidad de recargar baterías.

6.2.2. Algoritmos para definir intervalos en la obtención de datos

Para obtener los datos y evitar que se obtengan datos demasiados redundantes, que llenen la base de datos innecesariamente y aumente el consumo de energía se definirá una estrategia para definir la espera del microcontrolador entre mediciones, para esto podemos utilizar principalmente tres estrategias.

Un concepto que es posible asociar a esta optimización del prototipo es el de *Edge Computing* (Pastor, 2018), el cual consiste en realizar análisis y optimización de los datos en los nodos de una red, y no directamente en los servidores, esto permite obtener un mejor tiempo de respuesta en los sistemas IoT, puesto que se libera carga en la red y se realiza menos procesamiento en los servidores, las mejores capacidades de los microcontroladores permiten realizar este procesamiento.

- Algoritmo fijo: Este es el que se utilizó en la primera versión del prototipo, donde se define un tiempo fijo para la captura y envío de cada conjunto de datos, en este caso fue de 10 segundos. En la Figura 68 se encuentra un ejemplo en código Arduino.

```
const int intervalo = 10000; //intervalo en milisegundos
void setup() {
}
void loop(){
    delay(intervalo);
}
```

Figura 68: Ejemplo definición intervalo algoritmo fijo

- Algoritmo variable: En esta opción se modifica el tiempo del intervalo según alguna condición, en este caso el valor de los datos que generan los sensores. En la Figura 69 se encuentra un ejemplo en código Arduino.

```

void setup() {
}
void loop(){
    if(temperatura > 40 || humedad > 90 || luz > 95){
        intervalo = valor
    }
    else if(otra_condicion){
        Intervalo = otro valor
    }
    delay(intervalo)
}

```

Figura 69: Ejemplo definición intervalo algoritmo variable

- Algoritmo adaptativo: En este algoritmo se modifica el tiempo del intervalo a partir de las condiciones de los sensores, acelerando o desacelerando según se requiera, por ejemplo, si existen variaciones entre una medición y otra se capturan más datos y si no existen variaciones se evita capturar datos, de manera de evitar información redundante.

Para la realización del algoritmo se calcula un “promedio” que se calcula mediante la siguiente expresión.

$$Promedio_i = (1-\alpha) * Promedio_{i-1} + \alpha * valor_{sensores} \quad (7)$$

Donde α es un valor arbitrario, que determina que tanta relevancia se les dará a los datos nuevos, y $valor_{sensor}$ es el valor obtenido por la última medición de los sensores (temperatura, humedad, luz, etc.)

A partir del promedio generado y la diferencia obtenida con los nuevos datos se determina el tiempo del intervalo para la siguiente medición. En la Figura 70 se encuentra un ejemplo simplificado en código Arduino.

```

const int promedio = 30;
const int intervalo = 30000; //30seg
const int intervalo_minimo = 10000; //10seg
const int intervalo_maximo = 120000; //2min

void setup(){
}

void loop(){
    diferencia = calc_diferencia_porcentual(promedio_i, valor_sensores)
    if(diferencia > 100 && intervalo > intervalo_minimo){
        //Reducir intervalo 10%
    }
    else if(diferencia <= 100 && intervalo > intervalo_minimo){
        //Reducir intervalo 0.1*diferencia
    }
    else if(diferencia < 10 && intervalo < intervalo_maximo){
        //aumentar intervalo 0.1*diferencia
    }
    else{
        //mantener intervalo
    }
    delay(intervalo)
}

```

Figura 70: Ejemplo definición intervalo algoritmo adaptativo

Nota: La diferencia porcentual se calcula mediante la expresión:

$$Diferencia \% = \frac{|valor_{sensores} + Promedio_i|}{|Promedio_i|} * 100 \quad (8)$$

Lo que esto produce es que cuando existen variaciones de los datos obtenidos con el promedio almacenado se obtienen más datos, si la condición se mantiene así el promedio se estabiliza y se comienzan a obtener menos datos nuevamente.

El algoritmo adaptativo es el que se aplicó en la solución final, puesto que permite obtener menos datos redundantes, reduce el consumo de energía y entrega más información cuando ocurren cambios importantes en poco tiempo.

7. CONCLUSIONES Y TRABAJO FUTURO

7.1. Conclusión

Hay que recordar que el objetivo general del proyecto es desarrollar un prototipo de extracción de datos de un invernadero mediante dispositivos IoT, capaz de registrar información útil para los agricultores sobre los parámetros de interés (temperatura, humedad, caudal de agua, radiación UV, etc.) que podrían mejorar la producción de semillas de papas. El objetivo general se puede indicar que se cumplió y el desarrollo del proyecto permitió poner en práctica conocimientos adquiridos durante la formación académica del aspirante a este proyecto de título.

A continuación, se listan los objetivos específicos indicados al inicio de este trabajo y se analiza brevemente su cumplimiento.

- Adquirir conocimientos respecto al uso de soluciones IoT en proyectos de *Smart-farming*, sensores utilizados y tecnologías de conexión entre ellos e internet.

Este objetivo fue cumplido mediante la revisión de bibliografía, donde se conoció respecto a las tecnologías utilizadas, donde destaca el uso de open hardware, con microcontroladores como Arduino, tecnologías de red como WiFi, Zigbee o LoRa, y sensores de diverso tipo, además de las soluciones comerciales

- Diseñar una arquitectura para la solución que tenga en consideración los requerimientos de la empresa.

Este objetivo fue cumplido, previo a ello fue necesario realizar una extensa revisión de tecnologías, la que se puede revisar en el capítulo 3 de este trabajo, la solución diseñada consideró la utilización de sensores y microcontroladores enviando datos conectados mediante una red WiFi, los datos son almacenados en una base de datos en la nube y es posible acceder a ellos mediante la aplicación móvil desarrollada.

- Implementar una solución que permita recuperar datos sobre los parámetros de interés y almacenarlos.

La solución fue ejecutada a partir de la arquitectura del sistema previamente definida, con un funcionamiento correcto, se desarrolló la aplicación móvil para visualización de datos en tiempo real y se desarrolló el prototipo en dos versiones con microcontroladores distintos.

Hubo sensores que no pudieron ser implementados y probados como el de caudal de agua, debido a que no se contó con una instalación adaptada a esto, se espera que en un futuro se tenga acceso a las instalaciones del invernadero para poder realizar las respectivas pruebas.

- Validar los resultados entregados por la solución mediante la comparación de los datos obtenidos con soluciones existentes.

Este objetivo fue cumplido parcialmente mediante la comparación teórica con otras soluciones, no así práctica, debido al alto costo que tienen las soluciones comerciales. También se realizaron pruebas para determinar la fiabilidad del prototipo en condiciones simuladas y se comprobó que los parámetros medidos tienen relación con el crecimiento de las plantas.

7.2. Trabajo futuro

Luego de realizar el prototipo se detectaron algunas mejoras que podrían aplicarse en las siguientes iteraciones y que mejorarán la calidad final del producto.

- Realizar una aplicación web o utilizar algún sistema que permita visualizar los datos en una pantalla más grande, visualizar gráficas más avanzadas y métricas que permitan realizar análisis de los datos, así como también descargar la información sin necesidad de acceder a la base de datos.
- Efectuar pruebas más exhaustivas y durante más tiempo en el invernadero, donde se puedan detectar otras mejoras.
- Detectar otras variables que sean de importancia en el crecimiento de las plantas y añadir los sensores correspondientes a la solución
- Añadir sistemas de autenticación que permitan subir la aplicación directamente a las tiendas de aplicaciones móviles.
- Cambiar los *display LCD* por otros de tipo OLED que reduzcan el consumo energético.
- Añadir un sistema de alertas (notificaciones) en la aplicación móvil, que se envíen cuando existen situaciones anómalas.

Afortunadamente el prototipo se puede seguir mejorando a futuro para entregar información confiable y a tiempo a los encargados del invernadero.

8. REFERENCIAS

- Albrecht, C. (2020). *Arable Launches New Mark 2 Sensor to Monitor Climate and Plant Conditions on Farms*. Recuperado el 7 de Enero de 2021, de <https://thespoon.tech/arable-launches-new-mark-2-sensor-to-monitor-climate-and-plant-conditions-on-farms/>
- Apogee. (s.f.). *Apogee Instruments*. Recuperado el 5 de Enero de 2021, de <https://www.apogeeinstruments.com>
- Apogee. (s.f.). *Descripción SQ-522-SS*. Recuperado el 5 de Enero de 2021, de <https://www.apogeeinstruments.com/sq-522-ss-modbus-digital-output-full-spectrum-quantum-sensor/#product-tab-description>
- Arable. (2020). *Arable - Solutions Weather*. Recuperado el 5 de Enero de 2021, de https://www.arable.com/solutions_weather/
- Arduino. (2018). *What is Arduino?* Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
- ArduinoJson. (s.f.). *ArduinoJson*. Obtenido de <https://arduinojson.org/>
- Cherukupalli , P. (2019). *ESP32 Deep Sleep with Arduino IDE and Wake Up Sources*. Obtenido de <https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources>
- Chile Desarrollo Sustentable. (2015). *Instacrops: La innovación local reconocida en el extranjero que evita millonarias pérdidas al agro*. Recuperado el 5 de Enero de 2020, de <http://www.chiledesarrollosustentable.cl/noticias/noticia-pais/instacrops-la-innovacion-local-reconocida-en-el-extranjero-que-evita-millonarias-perdidas-al-agro/>
- Christoulakis, I. (2020). *How to power your Arduino using Solar Panel*. Obtenido de <https://create.arduino.cc/projecthub/iasonas-christoulakis/how-to-power-your-arduino-using-solar-panel-4d29d8>
- CodexVerde. (2019). *La importancia de monitorear cultivos ante el cambio climático que enfrenta Chile*. Obtenido de <https://codexverde.cl/la-importancia-de-monitorear-cultivos-ante-el-cambio-climatico-que-enfrenta-chile/>
- Copeland, L. (2001). *Extreme Programming*. Obtenido de <https://www.computerworld.com/article/2585634/extreme-programming.html>
- Crespo, E. (2019). *LDR con Arduino*. Recuperado el 15 de Noviembre de 2020, de <https://aprendiendoarduino.wordpress.com/tag/fotoresistencia/>
- De La Casa, A., & Ovando, G. (2011). *Eficiencia en el uso de la radiación en papa estimada a partir de la cobertura del follaje*. doi:10.31047/1668.298x.v28.n1.2777
- Del Valle, L. (2017). *NodeMCU tutorial paso a paso desde cero*. Obtenido de <https://programarfácil.com/podcast/nodemcu-tutorial-paso-a-paso/>
- Del Valle, L. (2020). *ESP8266 Deep Sleep, cuánto consumen NodeMCU y Wemos D1 Mini*. Obtenido de <https://programarfácil.com/esp8266/esp8266-deep-sleep-nodemcu-wemos-d1-mini/>
- Durán, J. M., Martínez, E., & Navas, L. M. (2000). *Los cultivos sin suelo: de la hidroponía a la aeroponía*. Universidad Politécnica de Madrid, Madrid. Obtenido de https://www.mapa.gob.es/ministerio/pags/biblioteca/revistas/pdf_vrural/Vrural_200_101_40_43.pdf

- Espinoza, O., Villavicencio, G., & Díaz, S. (2014). Paquete tecnológico para el monitoreo ambiental en invernaderos con el uso de hardware y software libre. *Terra Latinoamericana*, 77-84.
- ESPloradores. (2017). *Modos de ahorro de energía -DEEP SLEEP-*. Obtenido de <https://www.esploradores.com/practica-9-modos-de-ahorro-de-energia-deep-sleep/>
- European Space Agency. (2008). *Technology readiness levels handbook for space applications*.
- Expo. (s.f.). *Expo*. Obtenido de <https://expo.io/>
- Facebook. (s.f.). *React Native*. Obtenido de <https://reactnative.dev/>
- Firebase Arduino. (2016). *Firebase Arduino Documentation*. Obtenido de <https://firebasearduino.readthedocs.io/en/latest/>
- Firebase. (s.f.). *Firebase JavaScript SDK Reference*. Obtenido de <https://firebase.google.com/docs/reference/js>
- Firebase. (s.f.). *Firebase Realtime Database*. Obtenido de <https://firebase.google.com/docs/database?hl=es-419>
- GeekFactory. (2017). *Alimentar el Arduino: La guía definitiva*. Obtenido de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/alimentar-el-arduino-la-guia-definitiva/>
- Gorka, R. (2019). *¿Apps híbridas o apps nativas? Un breve análisis comparativo de tecnologías móviles*. Obtenido de <https://blog.irontec.com/apps-hibridas-vs-apps-nativas-un-breve-analisis-comparativo-de-tecnologias-moviles>
- Hardwarelibre. (s.f.). *DHT11*. Recuperado el 15 de Noviembre de 2020, de <https://www.hwlibre.com/dht11/>
- Hardwarelibre. (s.f.). *DHT22*. Recuperado el 15 de Noviembre de 2020, de <https://www.hwlibre.com/dht22/>
- Hardwarelibre. (s.f.). *LM35*. Recuperado el 14 de Noviembre de 2020, de <https://www.hwlibre.com/lm35/>
- Instacrops. (s.f.). *Instaweather*. Recuperado el 5 de Enero de 2021, de <https://www.instacrops.com/project/insta-weather>
- Jagadesh, M., Karthik, M., Manikandan, A., Nivetha, S., & Prasanth Kumar, R. (2018). IoT Based Aeroponics Agriculture Monitoring System. *International Journal of Creative Research Thoughts*.
- Jimenez, A. (2019). *React Native: ¿Qué es y para que sirve este framework de programación?* Obtenido de <https://openwebinars.net/blog/react-native-que-es-para-que-sirve/>
- Kerns, S. C., & Lee, J.-L. (2017). Automated Aeroponics System Using IoT for Smart Farming. *European Scientific Journal*.
- Kushner, D. (2011). *The Making of Arduino - How five friends engineered a small circuit board that's taking the DIY world by storm*. Obtenido de <https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>
- Lakhiar, A. I., Jianmin, G., Chandio, F. A., Buttar, N. A., & Qureshi, W. A. (2018). Monitoring and Control Systems in Agriculture Using Intelligent Sensor Techniques: A Review of the Aeroponic System. *J. Sensors*. doi:10.1155/2018/8672769

- Libelium. (s.f.). *Libelium*. Recuperado el 5 de Enero de 2021, de <http://www.libelium.com/>
- Llamas, L. (2016). *DS18B20*. Recuperado el 14 de Noviembre de 2020, de <https://www.luisllamas.es/temperatura-liquidos-arduino-ds18b20/>
- Llamas, L. (2016). *Opciones para alimentar Arduino con baterías*. Obtenido de <https://www.luisllamas.es/alimentar-arduino-baterias/>
- Maida, E. G., & Julian, P. (2015). *Metodologías de desarrollo de software*. Obtenido de : <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>
- Marín Morales, J. G. (1977). *Factores que afectan el crecimiento de las plantas*. Instituto Colombiano Agropecuario. Obtenido de <https://repository.agrosavia.co/handle/20.500.12324/22033>
- MaxElectronica. (s.f.). *Módulo ML8511*. Recuperado el 15 de Noviembre de 2020, de <https://maxelectronica.cl/luz-color/167-modulo-ml8511-sensor-de-luz-y-radiacion-ultravioleta-uv-a-uv-b.html>
- MaxElectronica. (s.f.). *Sensor Capacitivo de Humedad de Suelo v1.2*. Recuperado el 14 de Noviembre de 2020, de <https://maxelectronica.cl/temperatura-y-humedad/519-sensor-capacitivo-de-humedad-de-suelo-v12.html>
- MCI electronics. (s.f.). *Kit Sensor análogo para PH gravity*. Recuperado el 16 de Noviembre de 2020, de <https://www.mcielectronics.cl/shop/product/kit-sensor-analogo-para-ph-gravity-11001>
- Mechatronics, N. (2016). *Tutorial sensor de flujo de agua*. Recuperado el 16 de Noviembre de 2020, de https://www.naylampmechatronics.com/blog/47_tutorial-sensor-de-flujo-de-agua.html
- Mendez L., P., & Inostroza F., J. (2009). *Manual de la papa para La Araucanía*: . Ministerio de Agricultura, Instituto de Investigaciones Agropecuarias, Temuco. Obtenido de <http://bibliotecadigital.ciren.cl/bitstream/handle/123456789/32023/Manual%20de%20papa%20para%20la%20araucan%c3%ada%20Manejo%20de%20cultivo%20enfermedades%20y%20almacenaje%2011-05-2020.pdf?sequence=1&isAllowed=y>
- mongoDB. (s.f.). *What is a Non-Relational Database?* Obtenido de <https://www.mongodb.com/non-relational-database>
- Mora, H., & Rosas, J. (2019). *Diseño, desarrollo e implementación de una red de sensores inalámbricos (WSN) para el control, monitoreo y toma de decisiones aplicado en la agricultura de precisión basado en internet de las cosas (IOT).–Caso de estudio cultivo de frijol*.
- NASA. (2012). *Technology Readiness Level*. Obtenido de https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html
- NativeBase. (s.f.). *NativeBase*. Obtenido de <https://nativebase.io/>
- Nutricontrol. (2020). *Como influye el dioxido de carbono en el cultivo en invernadero*. Recuperado el 5 de Enero de 2021, de <http://nutricontrol.com/2015/como-influye-el-dioxido-de-carbono-co2-en-el-cultivo-en-invernadero/>
- Oracle. (s.f.). *¿Qué es una base de datos relacional?* Obtenido de <https://www.oracle.com/cl/database/what-is-a-relational-database/>

- Ortiz, M. (2012). *Modelo Incremental*. Obtenido de <http://isw-udistrital.blogspot.com/2012/09/ingenieria-de-software-i.html>
- Palacios, J., Ponce, K., Maya, E., Peluffo, D., Negrete, K., & Domínguez, H. (2017). Diseño de una red de sensores con tecnología 802.15.4 basado en el concepto de agricultura de precisión aplicada a cultivos de hortalizas bajo invernadero: Una prueba piloto. *Environmental Science*.
- Pastor, J. (2018). *Edge Computing: qué es y por qué hay gente que piensa que es el futuro*. Recuperado el 12 de Enero de 2020, de <https://www.xataka.com/internet-of-things/edge-computing-que-es-y-por-que-hay-gente-que-piensa-que-es-el-futuro>
- Perez Cardona, M. (2016). *Firebase, qué es y para qué sirve la plataforma de Google*. Obtenido de <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>
- Poppendieck, M., Poppendieck, T., & Poppendieck, T. D. (2003). *Lean Software Development: An Agile Toolkit*. (A.-W. Professional, Ed.)
- Raspberry PI. (s.f.). *¿Que es Raspberry Pi?* Obtenido de <https://raspberrypi.cl/que-es-raspberry/>
- Rautmare, S., & Bhalerao, D. M. (2016). MySQL and NoSQL database comparison for IoT application. *IEEE International Conference on Advances in Computer Applications*. Coimbatore. doi:10.1109/ICACA.2016.7887957
- Ravichandran, A. (2019). *React Native vs. Ionic: Which one is right for you?* Obtenido de <https://blog.logrocket.com/react-native-vs-ionic/>
- Ritter, E., Angulo, B., Riga, P., Herrán, C., Rellosa, J., & San José, M. (2001). Comparison of hydroponic and aeroponic cultivation systems for the production of potato minitubers. *Potato Research*, 127-135. doi:10.1007/BF02410099
- Sandoval, W. (2019). *Los 5 mejores marcos de aplicaciones híbridas para crear aplicaciones móviles en 2020*. Obtenido de http://www.pixelgrafia.com/post/105_los-5-mejores-marcos-de-aplicaciones-hibridas-para-crear-aplicaciones-moviles-en-2020
- Savtec. (s.f.). Savtec. Recuperado el 5 de Enero de 2021, de <http://www.savtec.cl/>
- Schwaber, K., & Sutherland, J. (2020). *The 2020 Scrum Guide*. Obtenido de <https://www.scrumguides.org/scrum-guide.html>
- Sitrad. (s.f.). *Sitrad Pro*. Recuperado el 12 de Enero de 2021, de <https://www.sitrad.com/es/sitrad-pro/#que-es>
- Strange, C. (2020). *Mejoramiento genético de las plantas: ¿qué beneficios tiene para los seres humanos?* Universidad de Chile. Obtenido de <https://www.uchile.cl/noticias/163779/mejoramiento-genetico-de-las-plantas-que-beneficios-tiene>
- Talos Electronics. (s.f.). *Sensor de humedad del suelo YL38 y YL69*. Recuperado el 14 de Noviembre de 2020, de <https://www.taloselectronics.com/products/sensor-de-humedad-del-suelo-yl38-y-yl69>
- Tunio, M. H., Gao, J., Shaikh, S. A., Lakhiar, I. A., Qureshi, W. A., Solangi, K. A., & Chandio, F. A. (2020). Potato production in aeroponics: An emerging food growing system in sustainable agriculture for food security. *Chilean journal of agricultural research*. doi:10.4067/S0718-58392020000100118
- Van leperen, W., & Velez-Ramirez, A. (2011). *Plants under continuous light*. doi:10.1016/j.tplants.2011.02.003

Weiss, B., & Gridling, G. (2007). *Introduction to Microcontrollers*. Vienna University of Technology.

ANEXOS

Anexo A: Visita a invernadero

Aun cuando no fue posible efectuar pruebas directamente en el invernadero, como fue inicialmente planeado, se realizó una visita a las instalaciones. A continuación, se muestran fotografías, donde se instaló un prototipo para que realizara mediciones.



En la cuarta fotografía se muestra un instrumento existente, con una temperatura registrada de 16.7° y Humedad relativa de 68%. La temperatura y humedad medida por el prototipo en ese instante se muestra en la captura de pantalla.



Anexo B: Plantas luego de 2 meses

Luego de realizado el experimento de crecimiento de las plantas, estas fueron trasplantadas en la intemperie, la siguiente fotografía muestra cómo se encontraban las plantas luego de 2 meses.



Como se ve en la fotografía la planta de porotos que más creció (marcada en rojo) se encuentra cerca de unos 70 centímetros, en cambio las plantas de trigo (marcadas en azul) crecieron alrededor de unos 50 centímetros.