

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:

NOMBRE DEL INFORMANTE 1

TÍTULOS Y GRADOS DEL INFORMANTE 1

PROFESOR INFORMANTE 2:

NOMBRE DEL INFORMANTE 2

TÍTULOS Y GRADOS DEL INFORMANTE 2

**IGNACIO JÉLVEZ HERNÁNDEZ**

VALDIVIA – CHILE

2020

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

**ÍNDICE**

[ÍNDICE DE TABLAS II](#_Toc57844542)

[ÍNDICE DE FIGURAS III](#_Toc57844543)

[RESUMEN IV](#_Toc57844544)

[ABSTRACT V](#_Toc57844545)

[1. Introducción 1](#_Toc57844546)

[1.1. Contexto 2](#_Toc57844547)

[1.2. Objetivos 3](#_Toc57844548)

[1.2.1. Objetivo General 3](#_Toc57844549)

[1.2.2. Objetivos Específicos 3](#_Toc57844550)

[1.3. Impacto esperado del proyecto 4](#_Toc57844551)

[2. Marco teorico 5](#_Toc57844552)

[3. Dispositivos y tecnologías 6](#_Toc57844553)

[3.1. Microcontroladores 6](#_Toc57844554)

[3.2. Conexión a internet y red 8](#_Toc57844555)

[3.3. Sensores 14](#_Toc57844556)

[3.4. Energía 21](#_Toc57844557)

[3.5. Bases de datos 25](#_Toc57844558)

[3.6. Tecnologías para el desarrollo 28](#_Toc57844559)

[4. Arquitectura y solución 33](#_Toc57844560)

[4.1. Descripción de la problemática 33](#_Toc57844561)

[4.2. Solución propuesta 34](#_Toc57844562)

[4.3. Metodología 35](#_Toc57844563)

[4.4. Especificación de requisitos 36](#_Toc57844564)

[4.4.1. Requerimientos funcionales 36](#_Toc57844565)

[4.4.2. Requerimientos no funcionales 37](#_Toc57844566)

[4.5. Casos de uso 37](#_Toc57844567)

[4.6. Modelo de procesos 38](#_Toc57844568)

[4.7. Descripción de los actores del sistema 39](#_Toc57844569)

[4.8. Arquitectura del sistema 39](#_Toc57844570)

[4.8.1. Hardware 39](#_Toc57844571)

[4.8.2. Software 41](#_Toc57844572)

[5. Pruebas 42](#_Toc57844573)

[5.1. Validación con soluciones existentes 42](#_Toc57844574)

[5.2. Plantas y crecimiento 42](#_Toc57844575)

[5.3. Retardo en toma de datos 42](#_Toc57844576)

[5.4. Consumo de energía 42](#_Toc57844577)

[6. Mejoras 43](#_Toc57844578)

[6.1. Cambios de sensores 43](#_Toc57844579)

[6.2. Mejoras de algoritmos 43](#_Toc57844580)

[6.3. Definición de la solución final 43](#_Toc57844581)

[7. Conclusión 44](#_Toc57844582)

[8. Referencias 45](#_Toc57844583)

[ANEXOS 46](#_Toc57844584)

ÍNDICE DE TABLAS

Tabla Página

Tabla 1: Comparación placas Arduino 7

Tabla 2: Comparación placas basadas en ESP8266 9

Tabla 3: Comparación tecnologías para red inalámbrica. 13

Tabla 4: Comparación entre bases de datos relacionales y no relacionales 27

Tabla 5: Precios Firebase Realtime Database 29

Tabla 6: Comparativa de características entre aplicaciones móviles hibridas y nativas 30

Tabla 7: Descripción caso de uso “Ver datos de los sensores” 37

Tabla 8: Flujo normal de los eventos caso de uso “Ver datos de los sensores” 37

Tabla 9: Descripción caso de uso "Ver gráficas de los sensores" 38

Tabla 10: Flujo normal de los eventos caso de uso "Ver gráficas de los sensores" 38

Tabla 11: Descripción caso de uso “Configurar sensores a mostrar” 38

Tabla 12: Flujo normal de los eventos caso de uso “Configurar sensores a mostrar” 38

ÍNDICE DE FIGURAS

Figura DDS DDDDDDDDDSF 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 7

Figura 5: Rapsberry Pi 3 B+ 8

Figura 6: ESP-01 9

Figura 7: NodeMCU 9

Figura 8: Módulo XBee-PRO ZB S2C 11

Figura 9: Módulo Z-Uno 11

Figura 10: SX1276 13

Figura 11: TTGO LoRa32 13

Figura 12: Sensor YL-69/YL-38 15

Figura 13: Sensor de humedad de suelo capacitivo 16

Figura 14: Sensor LM35 16

Figura 15: Sensor DS18B20. 17

Figura 16: Sensor DHT11 18

Figura 17: Sensor DHT22 18

Figura 18: Fotorresistencia LDR 19

Figura 19: Sensor ML8511 19

Figura 20: Sensor YF-S201 20

Figura 21: Sensor de PH Gravity 20

Figura 22: Opciones de alimentación de Arduino 22

Figura 23: Opciones de alimentación de NodeMCU 23

Figura 24: Esquema de alimentación de Arduino usando panel solar 24

Figura 25: Ejemplo estructura de datos en tabla SQL 25

Figura 26: Ejemplo estructura de documento JSON 26

Figura 27: Servicios disponibles en Firebase 28

Figura 28: Funcionamiento del sistema de control de temperatura 33

Figura 29: Flujo de datos del sistema 35

Figura 30: Diagrama de conexión usando Arduino UNO y ESP-01 40

Figura 31: Diagrama de conexión usando NodeMCU 40

RESUMEN

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus sed hendrerit lorem. Pellentesque sed lectus faucibus, mattis magna in, sagittis sapien. Quisque et libero ex. Integer eget urna vehicula, efficitur elit quis, auctor ipsum. Praesent mollis, ligula at congue convallis, mauris erat pharetra ex, non dictum orci urna et lacus. Proin pharetra sapien congue metus dictum accumsan. Nulla mollis metus eget velit tempus, a pellentesque eros ultricies. Nam tincidunt at enim id congue. Curabitur egestas non velit a imperdiet.

Proin ultricies, tellus ac mollis pulvinar, quam erat eleifend tellus, at efficitur leo felis et ipsum. Donec ut leo ornare, tincidunt lorem ut, iaculis felis. Pellentesque malesuada condimentum lacus. Quisque eget tincidunt massa. Nunc aliquam lobortis dapibus. Aliquam erat volutpat. Quisque interdum, erat ac lobortis ullamcorper, dolor erat venenatis sem, quis viverra dolor leo eu nulla. Aenean efficitur nisl nec mi elementum, nec mattis nibh semper. Ut auctor nunc quis magna varius ultricies.

Donec eu vehicula nulla, vitae tempus velit. Maecenas mi libero, euismod sed lacus quis, rhoncus posuere eros. Nulla vel volutpat erat, eu vehicula elit. Phasellus a dolor vitae ligula mattis dignissim ac eu augue. Praesent non enim sit amet mi sollicitudin dictum et sed dui. Aenean ullamcorper hendrerit pellentesque. Vivamus cursus turpis in sem facilisis, ac varius mi malesuada. Quisque sed nisl sed quam tincidunt hendrerit sit amet quis neque. Nulla lobortis, arcu vel fermentum pharetra, nisl lectus eleifend diam, id elementum dui velit ut sapien. Aliquam hendrerit metus vitae lorem tincidunt dapibus.

Phasellus libero nibh, commodo at dignissim nec, ultricies non enim. Donec fermentum vitae diam et consequat. Interdum et malesuada fames ac ante ipsum primis in faucibus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec hendrerit finibus elit, id blandit odio. Nulla porttitor justo pharetra, gravida enim ut, iaculis leo. Praesent tempus ex venenatis lorem bibendum posuere id nec urna. Etiam rutrum porttitor dolor, placerat pharetra mauris euismod congue. Nulla nisl nibh, aliquam non maximus in, accumsan et risus. Nunc dignissim fringilla turpis in dignissim. Curabitur facilisis, ipsum a ultricies luctus, nisl quam mollis mi, id ultricies nibh velit in nibh. In hac habitasse platea dictumst. Nam semper, metus porttitor aliquam molestie, neque elit mollis ex, et imperdiet nisl sem nec diam.

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus sed hendrerit lorem. Pellentesque sed lectus faucibus, mattis magna in, sagittis sapien. Quisque et libero ex. Integer eget urna vehicula, efficitur elit quis, auctor ipsum. Praesent mollis, ligula at congue convallis, mauris erat pharetra ex, non dictum orci urna et lacus. Proin pharetra sapien congue metus dictum accumsan. Nulla mollis metus eget velit tempus, a pellentesque eros ultricies. Nam tincidunt at enim id congue. Curabitur egestas non velit a imperdiet.

Proin ultricies, tellus ac mollis pulvinar, quam erat eleifend tellus, at efficitur leo felis et ipsum. Donec ut leo ornare, tincidunt lorem ut, iaculis felis. Pellentesque malesuada condimentum lacus. Quisque eget tincidunt massa. Nunc aliquam lobortis dapibus. Aliquam erat volutpat. Quisque interdum, erat ac lobortis ullamcorper, dolor erat venenatis sem, quis viverra dolor leo eu nulla. Aenean efficitur nisl nec mi elementum, nec mattis nibh semper. Ut auctor nunc quis magna varius ultricies.

Donec eu vehicula nulla, vitae tempus velit. Maecenas mi libero, euismod sed lacus quis, rhoncus posuere eros. Nulla vel volutpat erat, eu vehicula elit. Phasellus a dolor vitae ligula mattis dignissim ac eu augue. Praesent non enim sit amet mi sollicitudin dictum et sed dui. Aenean ullamcorper hendrerit pellentesque. Vivamus cursus turpis in sem facilisis, ac varius mi malesuada. Quisque sed nisl sed quam tincidunt hendrerit sit amet quis neque. Nulla lobortis, arcu vel fermentum pharetra, nisl lectus eleifend diam, id elementum dui velit ut sapien. Aliquam hendrerit metus vitae lorem tincidunt dapibus.

Phasellus libero nibh, commodo at dignissim nec, ultricies non enim. Donec fermentum vitae diam et consequat. Interdum et malesuada fames ac ante ipsum primis in faucibus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec hendrerit finibus elit, id blandit odio. Nulla porttitor justo pharetra, gravida enim ut, iaculis leo. Praesent tempus ex venenatis lorem bibendum posuere id nec urna. Etiam rutrum porttitor dolor, placerat pharetra mauris euismod congue. Nulla nisl nibh, aliquam non maximus in, accumsan et risus. Nunc dignissim fringilla turpis in dignissim. Curabitur facilisis, ipsum a ultricies luctus, nisl quam mollis mi, id ultricies nibh velit in nibh. In hac habitasse platea dictumst. Nam semper, metus porttitor aliquam molestie, neque elit mollis ex, et imperdiet nisl sem nec diam.

# Introducción

El crecimiento de las plantas se ve afectado principalmente por factores climáticos [https://repository.agrosavia.co/handle/20.500.12324/22033], como la temperatura, la humedad o las precipitaciones, así como también la calidad del aire, que afecta a la parte aérea de la planta, gases como el dióxido de carbono (CO2), el dióxido de azufre (SO2) o el monóxido de carbono (CO) puede afectar, e incluso provocar daños a las plantas cuando se encuentran en concentraciones altas.

Las plantas también se pueden ver afectadas a 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.

<http://bibliotecadigital.ciren.cl/bitstream/handle/123456789/32023/Manual%20de%20papa%20para%20la%20araucan%c3%ada%20Manejo%20de%20cultivo%2cenfermedades%20y%20almacenaje%2011-05-2020.pdf?sequence=1&isAllowed=y>]

Los tubérculos y en especial las papas no están exentas de estas afecciones, las papas que se producen en el sur de Chile se ven afectadas principalmente por:

* 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 estas y por lo tanto el rendimiento de los cultivos puede ser modificado por acción humana, y para ello se utilizan diferentes técnicas.

* Para la prevención de enfermedades se utilizan mejoras genéticas ….
* Nuevas técnicas de cultivos, tales como la hidroponía o la aeroponia….

|  |  |
| --- | --- |
| Figura 1: Cultivo Hidropónico | Aeroponics Farming - Smartenup - Innovative Research And Technologies.  Figura 2: Cultivo Aeropónico |

* Control de ¿Los ciclos de sueño de la planta? ….
* Monitoreo y análisis de datos….

## Contexto

Aeroponics es un invernadero dedicado a la producción de semillas de varias especies de tubérculos de papa, mediante la técnica de la Aeroponía (ver figura). La planta de producción se encuentra ubicada en el sector de Cayumapu, comuna de Valdivia y cuenta con alrededor de 8 invernaderos. Actualmente trabajan 14 personas, distribuidos entre gerentes, supervisores, equipo técnico y asistente contable.

Para que la planta tenga un desarrollo óptimo debe tener una temperatura estable, lo que se logra con estos aspersores que mantienen la temperatura, pero esta solo se realiza de forma estimativa, ya que la temperatura se toma en los estanques que almacena el agua y no directamente en las plantas. La medición constante de la temperatura permitirá mejorar el sistema de riego y mantener una temperatura más estable. Otras variables de interés, cuyas mediciones sería importante incorporar en el también son humedad, caudal de agua, radiación solar, etc.



Figura 3: Plantas de papa al interior del invernadero Aeroponics

## Objetivos

### 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.

### 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.

## 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 invertir en estas tecnologías.

# Marco teorico

Se efectuó una 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

* DECIR COSAS DE PAPERS CON CITAS

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

* DECIR COSAS DE PAPERS CON CITAS [SACAR COSAS DEL ANTEPROYECTO]

# 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 diversos microcontroladores, tecnologías para interconectar los dispositivos con internet, sensores y opciones para alimentarlos de energía que se estudiaron, 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.

## 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 [MICROCONTROLADOR]. Estos son de especial utilidad en sistemas de control y monitoreo.

Microcontrolador: https://ti.tuwien.ac.at/ecs/teaching/courses/mclu/theory-material/Microcontroller.pdf

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 mas fácil la integración con otros sistemas y facilita obtener sensores compatibles.

**Arduino** [**https://www.arduino.cc/en/Guide/Introduction**](https://www.arduino.cc/en/Guide/Introduction)

Arduino 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 Ivrea, Italia, (https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino) 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 (la placa más popular, ver figura), 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 siguiente tabla 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 |
| N° Pines Digitales | 14 | 20 | 54 | 22 |
| N° 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 |

**Rapsberry**

[REF: https://raspberrypi.cl/que-es-raspberry/ ] Rapsberry Pi es una computadora de bajo consto y tamaño compacto, puede utilizarse conectándola a un monitor, teclado y mouse, cuenta con un sistema operativo Linux.

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 Rapsberry Pi 3 B+ (Ver Figura), que puede conseguirse por alrededor de 30 USD, además de 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+

<https://www.xataka.com/basics/arduino-raspberry-pi-que-cuales-sus-diferencias>

<https://www.arrow.com/es-mx/research-and-events/videos/the-top-10-development-platforms-dev-kits-2018>

## 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 la necesidad.

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.

* + 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, en el módulo ESP-01 (Ver figura), 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, por ejemplo, como se verá más adelante este es uno de los usos que se le da a ese dispositivo.

Existen varias placas basadas en módulos ESP, entre las más populares tenemos el ESP-01, ESP-12, NodeMCU (Ver figura) y Wemos D1

|  |  |
| --- | --- |
| Figura 6: ESP-01 | NodeMCU v2 - Lua based ESP8266 development kit  Figura 7: NodeMCU |

En la siguiente tabla 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 *protoboard* | 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 |

* + 1. **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.

* + 1. **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 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 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 prototipaje son los módulos Xbee (ver figura), 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 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

* + 1. **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 si, por lo que deben estar 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), esta es una placa de expansión con certificación Z-Wave basada en Arduino, también se puede usar en otras placas como Rapsberry.

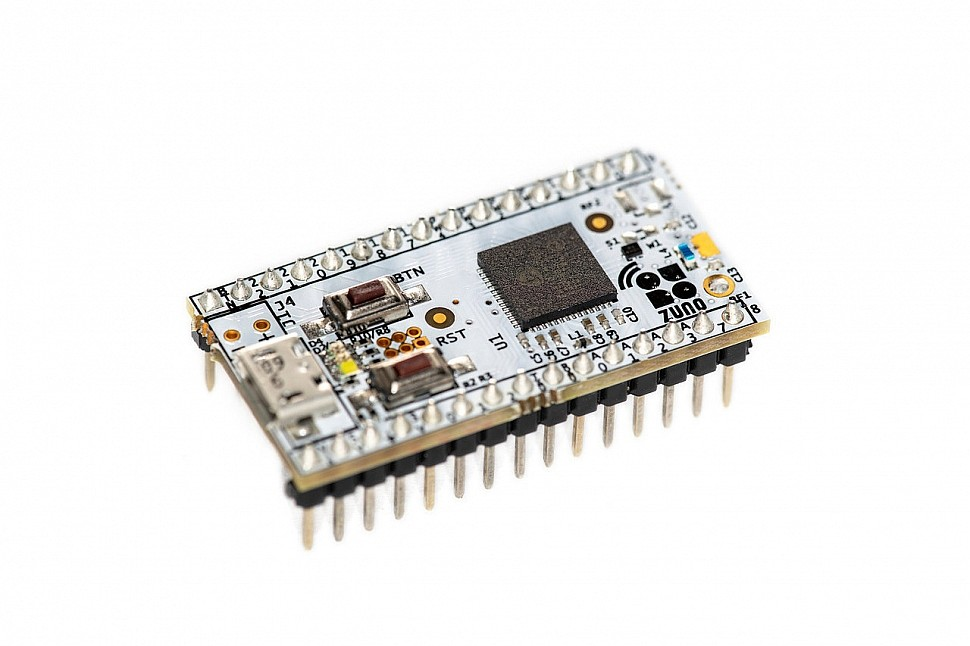


Figura 9: Módulo Z-Uno

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 US$ 60.

* + 1. **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.
  + 1. **Lora**

LoraWAN es la capa de red de estándar abierto gobernada por LoRa Alliance. Sin embargo, no está realmente abierto ya que el chip subyacente 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. Lora es una buena red para soluciones IoT, se pueden construir soluciones propias, donde el dueño 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 o TTGO-LoRa (ver figura), 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 |

* + 1. **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.

* + 1. **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 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 del sur.
* Narrowband: No existe una red disponible en Chile para su uso.

En la siguiente tabla 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 enviaran 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.

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.

## 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 mas económicas, existe mas variedad de dispositivos y una comunidad de usuarios y desarrolladores que aporta con mejoras y software para utilizar, pero se requiere de mas conocimiento para utilizarlos, puesto que son los usuarios quieres deben realizar las soluciones y no una empresa que ofrece todo en un paquete *plug and play*.

* + 1. **Soluciones Comerciales**

<https://www.apogeeinstruments.com/sq-522-ss-modbus-digital-output-full-spectrum-quantum-sensor/#product-tab-description>

<https://www.detectores.cl/tienda/detector-dioxido-carbono-3-salidas-analogas/>

COLOCAR SENSOR DE CO2 INDUSTRIAL O COMERCIAL

* + 1. **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.

* + - 1. **Sensores de humedad**

**YL-69/YL-38**

Este sensor se utiliza para medir la humedad del suelo (ver figura), 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. [https://www.taloselectronics.com/blogs/tutoriales/sensor-de-humedad-del-suelo-yl38-y-yl69, revisado 14 nov]

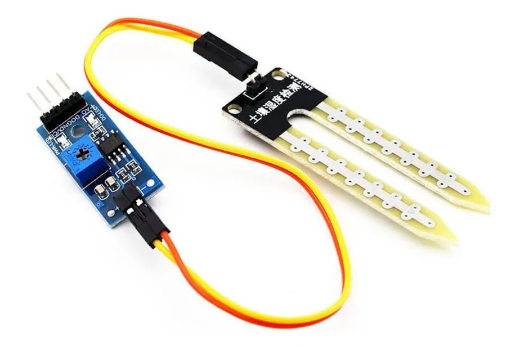


Figura 12: Sensor YL-69/YL-38

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 (ver figura) es capaz de medir la humedad del suelo donde es insertado, mediante detección capacitiva. Esta 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.

[https://maxelectronica.cl/temperatura-y-humedad/519-sensor-capacitivo-de-humedad-de-suelo-v12.html, revisado 14 nov]

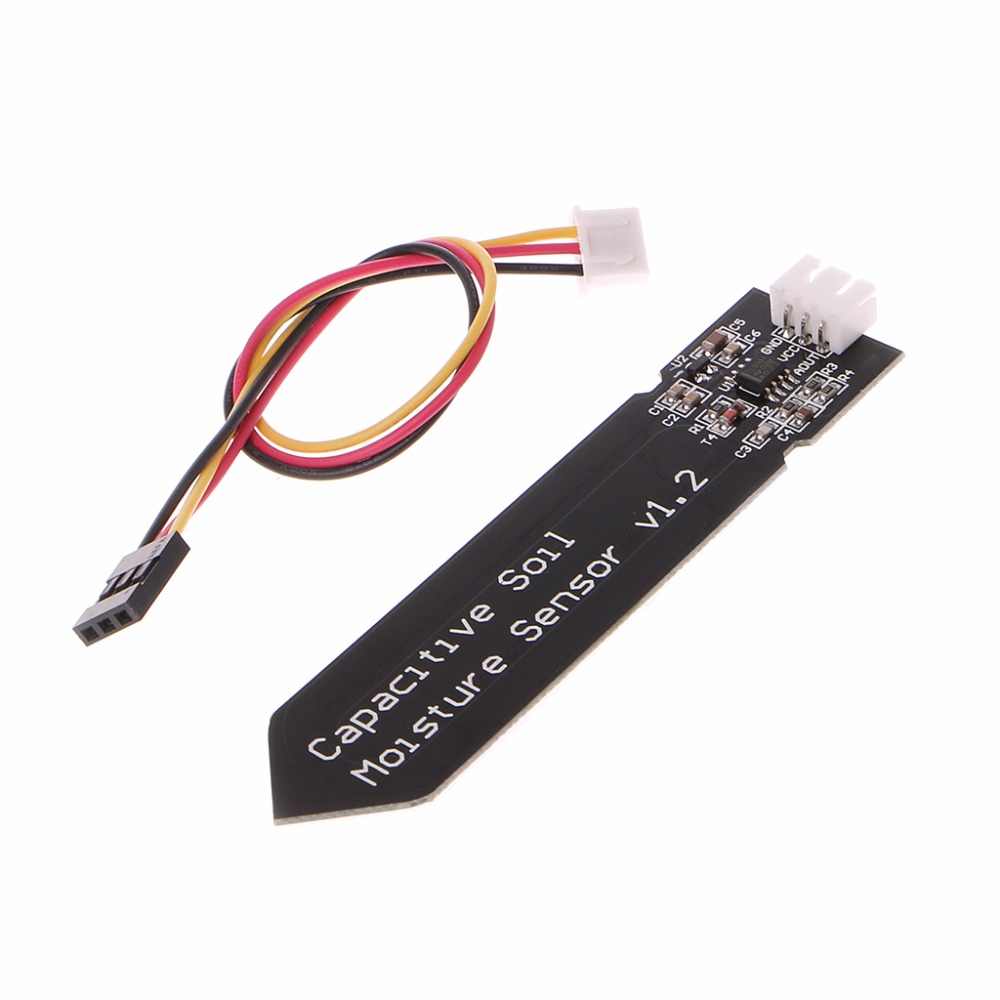


Figura 13: Sensor de humedad de suelo capacitivo

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
  + - 1. **Sensores de temperatura**

**LM35**

[https://www.hwlibre.com/lm35/] El LM35 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 14: Sensor LM35

Características:

* Envía señal analógica
* Calibrado para grados Celsius.
* Tensión de precisión garantizada de 0.5ºC a 25ºC.
* Baja corriente de alimentación (60 μA).
* Voltaje de trabajo entre 4 y 30v.
* Precio: ~ 1 USD

**DS18B20**

[https://www.luisllamas.es/temperatura-liquidos-arduino-ds18b20/] El sensor de temperatura DS18B20 es un sensor idóneo para medir temperatura en ambientes húmedos o debajo del agua, puesto que viene en forma de sonda impermeable (ver figura).



Figura 15: Sensor DS18B20.

Características:

* Alimentación de 3V a 5V
* Utiliza el protocolo 1-Wire
* Precisión de ±0,5ºC en temperaturas entre -10°C y 85°C
* Resolución programable: 9-bit, 10-bit, 11-bit o 12-bit (por defecto)
* Precio: ~ 4 USD
  + - 1. **Sensores de temperatura y humedad**

**DHT11**

[https://www.hwlibre.com/dht11/] El DHT11 (ver figura) 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 0ºC a 50ºC
* Precisión para medir temperatura a 25ºC de unos 2ºC de variación
* La resolución para medir temperatura es de 8-bit, 1ºC
* 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 0-50ºC
* Precio: ~ 2 USD

**DHT22**

[https://www.hwlibre.com/dht22/] El DHT22 (ver figura) 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 -40ºC a 125ºC
* Precisión para medir temperatura a 25ºC de 0.5ºC de variación
* La resolución para medir temperatura es de 8-bit, 0,1ºC
* 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 0-50ºC
* 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 16: Sensor DHT11 | Figura 17: Sensor DHT22 |

* + - 1. **Sensores de luz y radiación UV**

**Fotorresistencia LDR**

[https://aprendiendoarduino.wordpress.com/tag/fotoresistencia/] Una fotorresistencia o LDR (ver figura) es un componente electrónico cuya resistencia varía en función de la luz.



Figura 18: 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**

[https://maxelectronica.cl/luz-color/167-modulo-ml8511-sensor-de-luz-y-radiacion-ultravioleta-uv-a-uv-b.html] El módulo ML8511 (ver figura) es un sensor de luz ultravioleta (UV), funciona al emitir una señal analógica en relación con la cantidad de luz UV detectada.

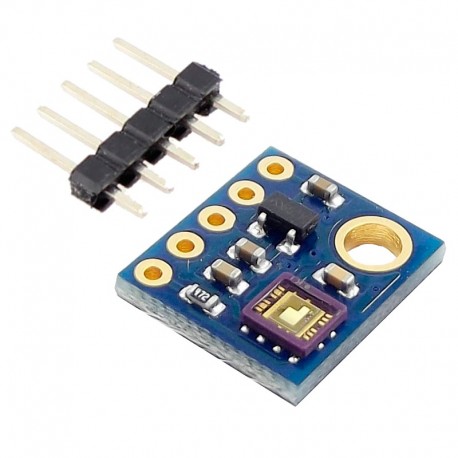


Figura 19: Sensor ML8511

Características:

* Alimentación: 3.3V
* Salida analógica
* Longitud de onda captada: 280-390nm
* Precio: ~ 10 USD
  + - 1. **Otros sensores**

**Flujo de agua - YF-S201**

[https://naylampmechatronics.com/blog/47\_tutorial-sensor-de-flujo-de-agua.html]

El sensor YF-S201 (ver figura) es un sensor de flujo de agua, comúnmente utilizado para medir el consumo de agua, internamente tiene un rotor cuyas paletas tienen un imagen 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 mas sencillos y que sirve para flujos de agua bajos.



Figura 20: 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**

[https://www.mcielectronics.cl/shop/product/kit-sensor-analogo-para-ph-gravity-11001]

El pH es una medida de acidez o alcalinidad de una disolución. El pH indica la concentración de iones hidronio [H3O+] 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 imagen)



Figura 21: Sensor de PH Gravity

Características:

* Alimentación: 5V.
* Rango de medición: 0-14pH.
* Medición de Temperatura: 0-60°C.
* Precisión: ± 0.1 pH (25°c).
* Tiempo de respuesta: ≤ 1 min.
* Precio: ~ 30 USD
  + 1. **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.

## 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 mas adecuada en cada caso y que se ajusta a las condiciones de los invernaderos.

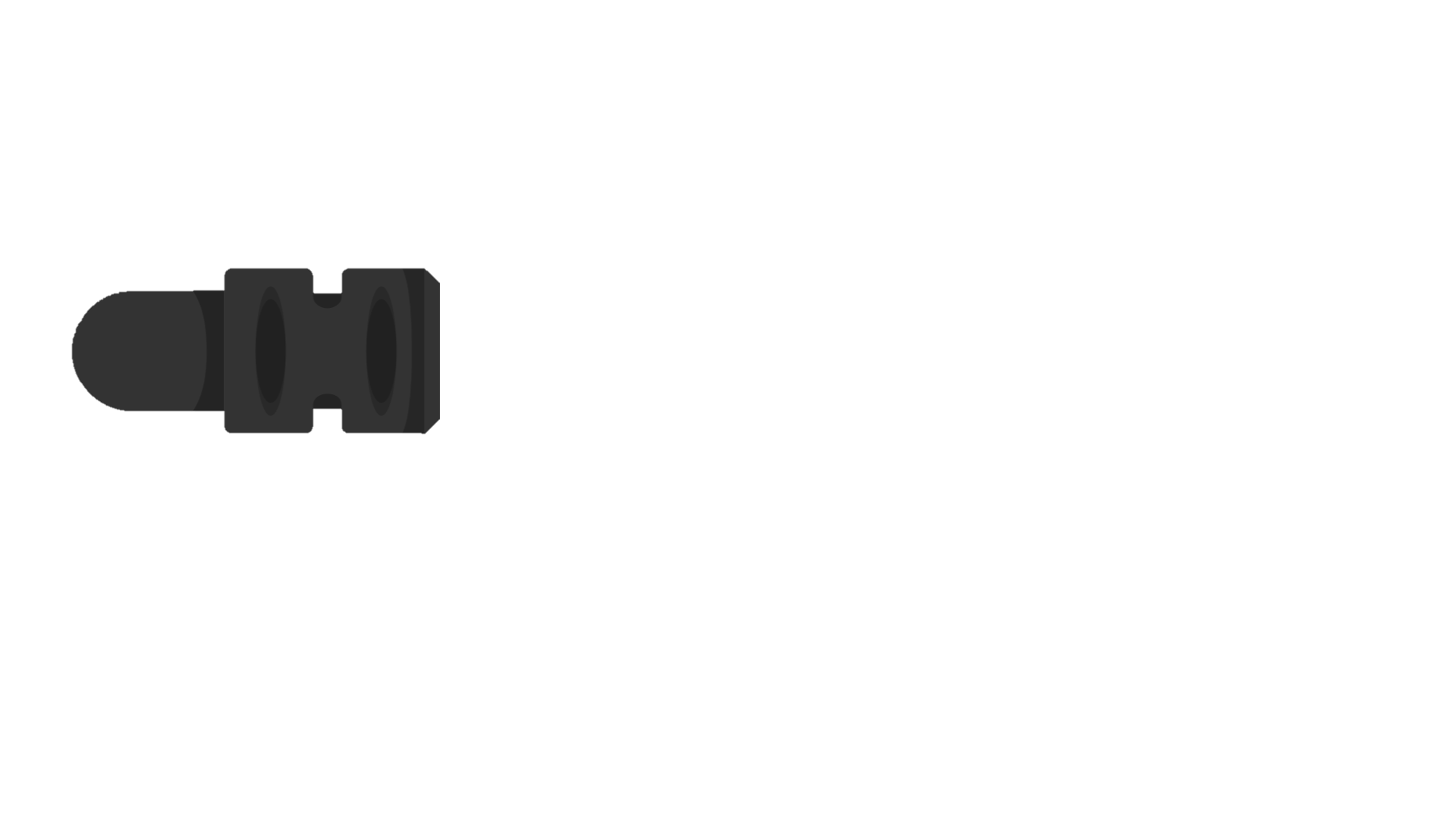
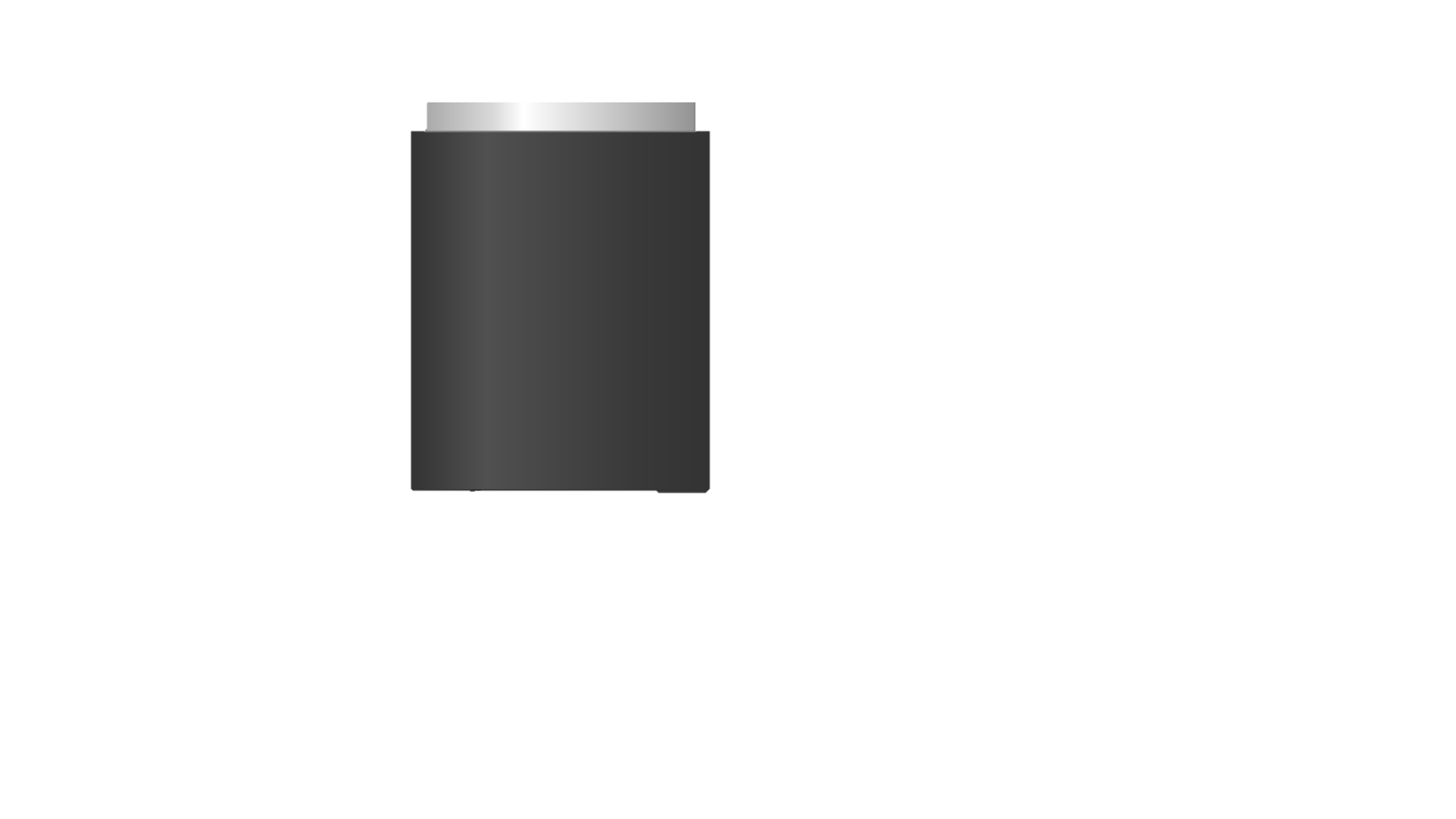
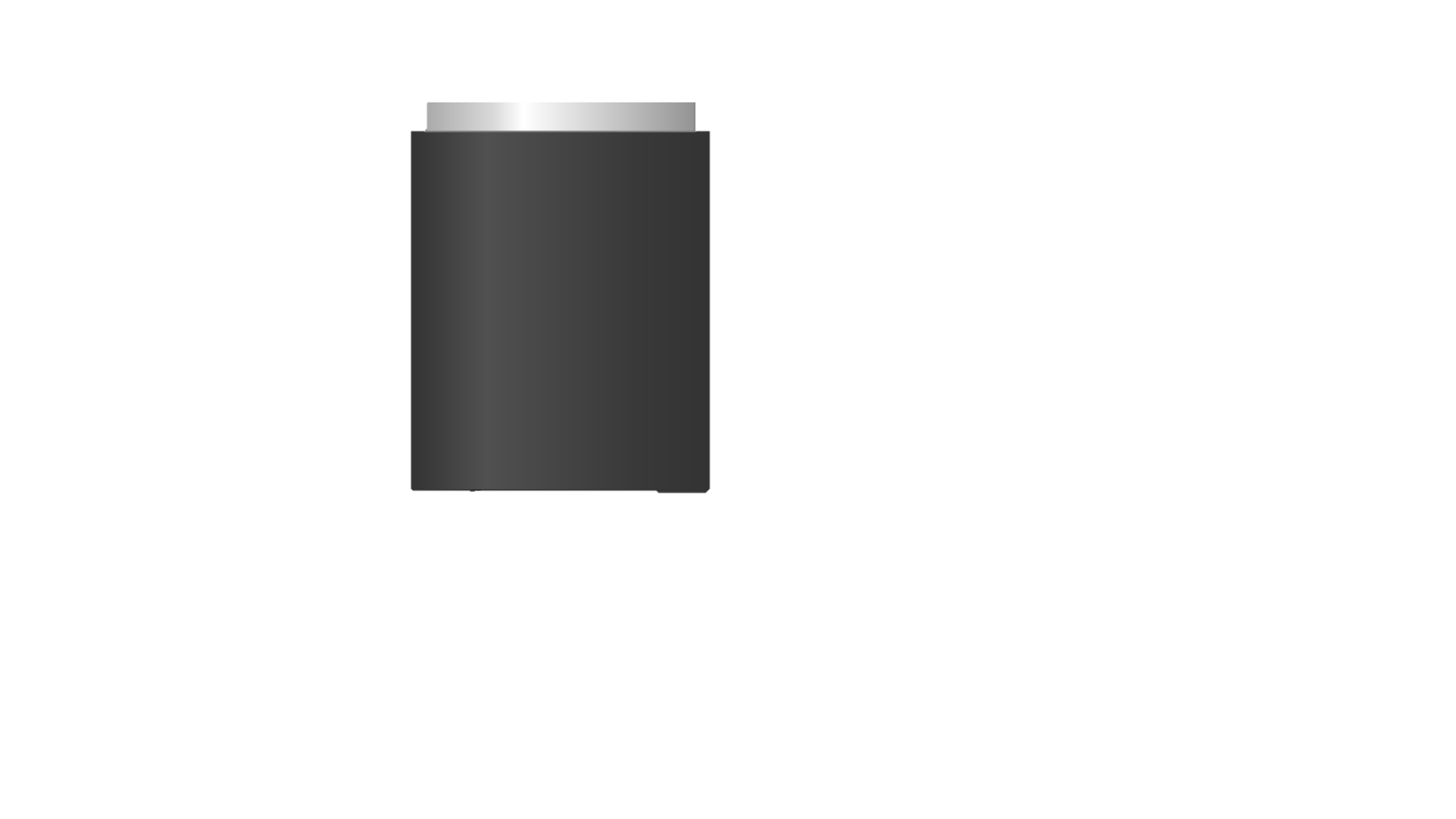
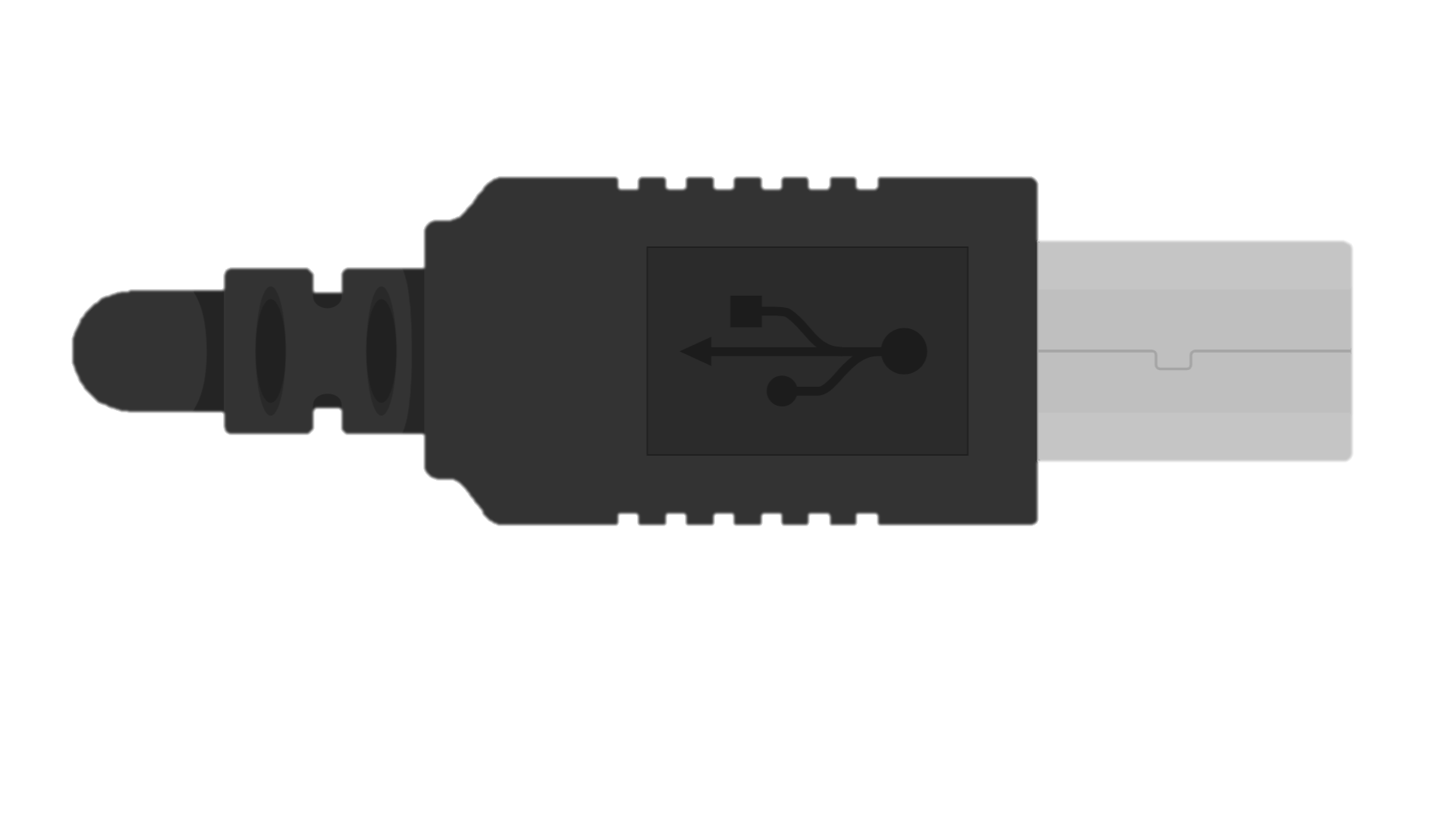
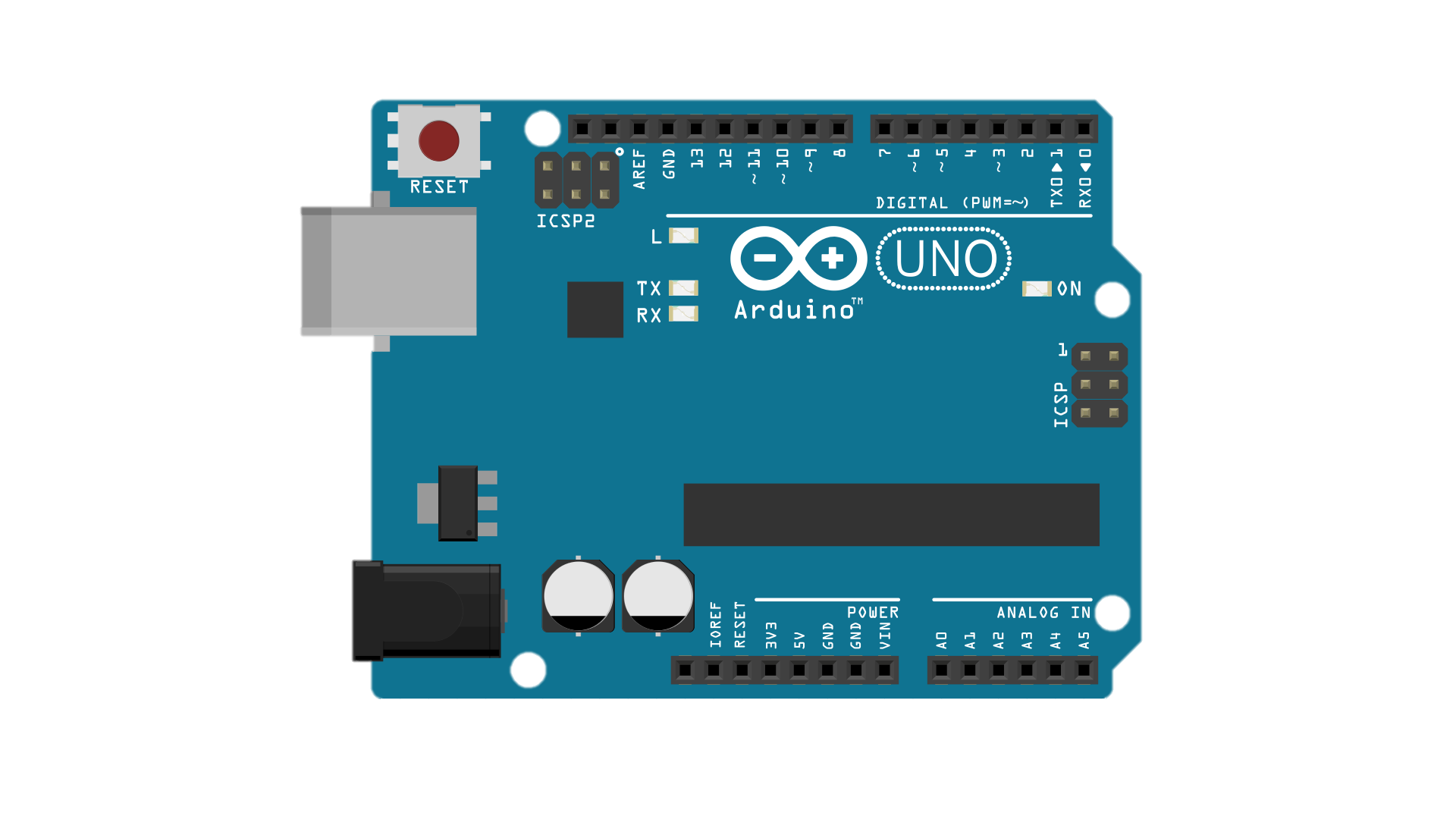
* + 1. **Opciones de alimentación de las placas**

<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/alimentar-el-arduino-la-guia-definitiva/> Explica opciones para conectar arduino (puertos y voltajes)

Para alimentar el Arduino UNO tenemos 3 opciones (ver figura):

1. Alimentar mediante el conector *jack*: Al alimentar por este puerto se debe utilizar un voltaje de 7 a 12 volts DC, voltajes menos 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 limite 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.



USB : 5 V

Conector Jack: 7-12 V

Pines: 5 V |

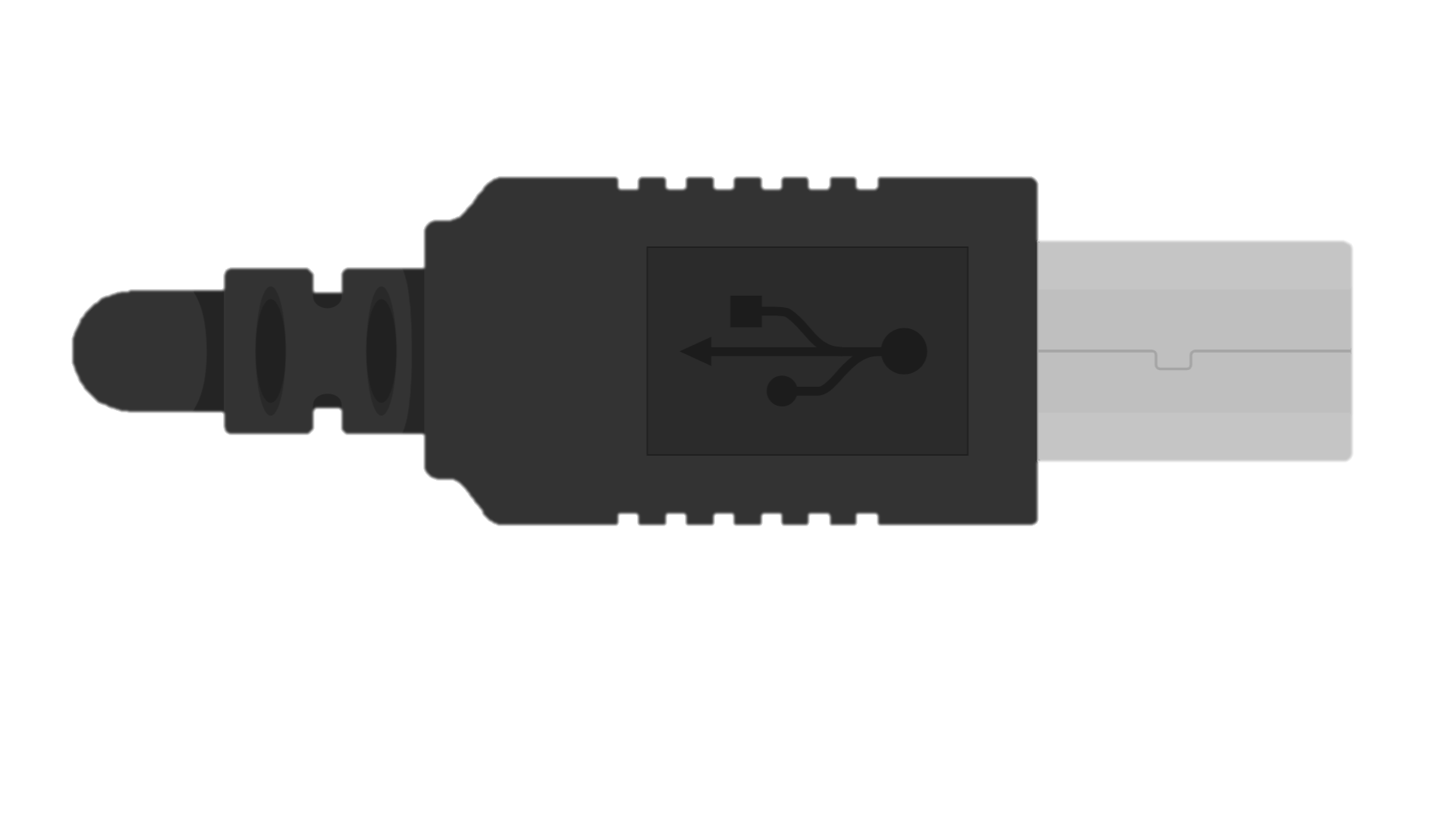
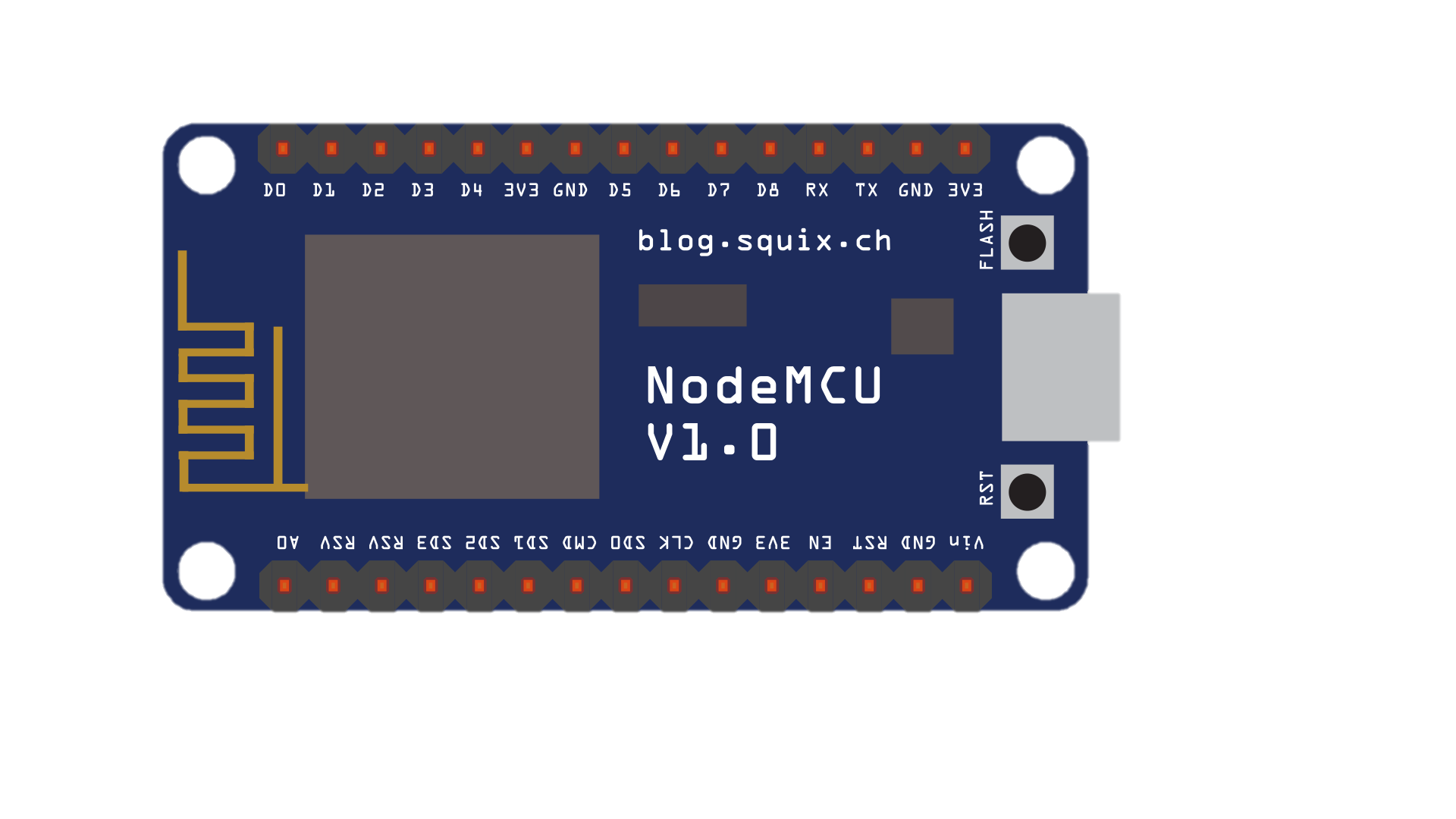
7-12 V

Figura 22: Opciones de alimentación de Arduino

Para alimentar el NodeMCU tenemos 2 opciones (ver figura):

https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/

1. Alimentar el puerto MicroUSB: Se recomienda que se utilice una fuente de 5V/1A para obtener el funcionamiento optimo.
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.



USB: 5 V

Pin: 3.3 V

Pin: 3.3 V

Pin: 3.3 V

Pin VIN: 5-9 V

Figura 23: Opciones de alimentación de NodeMCU

* + 1. **Baterías**

<https://www.luisllamas.es/alimentar-arduino-baterias/> Opciones usando baterías

Para el uso de baterías se tienen 3 opciones:

* 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 al 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 señores, 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. Como 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.
  + 1. **Paneles Solares**

<https://create.arduino.cc/projecthub/iasonas-christoulakis/how-to-power-your-arduino-using-solar-panel-4d29d8> Tutorial para usar un panel solar

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).



Figura 24: Esquema de alimentación de Arduino usando panel solar

* + 1. **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 (puede servir, pero las condiciones de humedad complican el “enchufarlo”): 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.

## 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 que 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.

* + 1. **Bases de datos relacionales (SQL)**

[ORACLE: <https://www.oracle.com/cl/database/what-is-a-relational-database/>]

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). 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **temperatura** | **humedad** | **luz** | **luz\_uv** |
| 1 | 23.2 | 74 | 57 | 7 |
| 2 | 24.5 | 75 | 68 | 7.2 |

Figura 25: 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 maquina 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.

[INFO DEL PAPER EN INGLES] -> https://sci-hub.do/10.1109/ICACA.2016.7887957

* + 1. **Bases de datos no relacionales (NoSQL)**

[MONGO: <https://www.mongodb.com/non-relational-database>]

Las bases de datos no relacionales 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), pero existen varios tipos de bases de datos NoSQL que poseen otras estructuras.

{

"temperatura": "23.2",

"humedad”: "74",

"luz”: "57",

"luz\_uv": "7",

…

}

Figura 26: 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 numero 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.
  + 1. **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.

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 anterior 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 maquinas, 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).

## Tecnologías para el desarrollo

* + 1. **Arduino**

[LENGUAJE DE PROGRAMACION E IDE]

**3.6.2. Firebase**

[DESCRIPCION <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarroladores-google-seo-sem/>]

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 apps de alta calidad de forma ágil, incluyendo múltiples servicios (ver figura), la plataforma esta disponible para plataformas como iOS, Android y web.

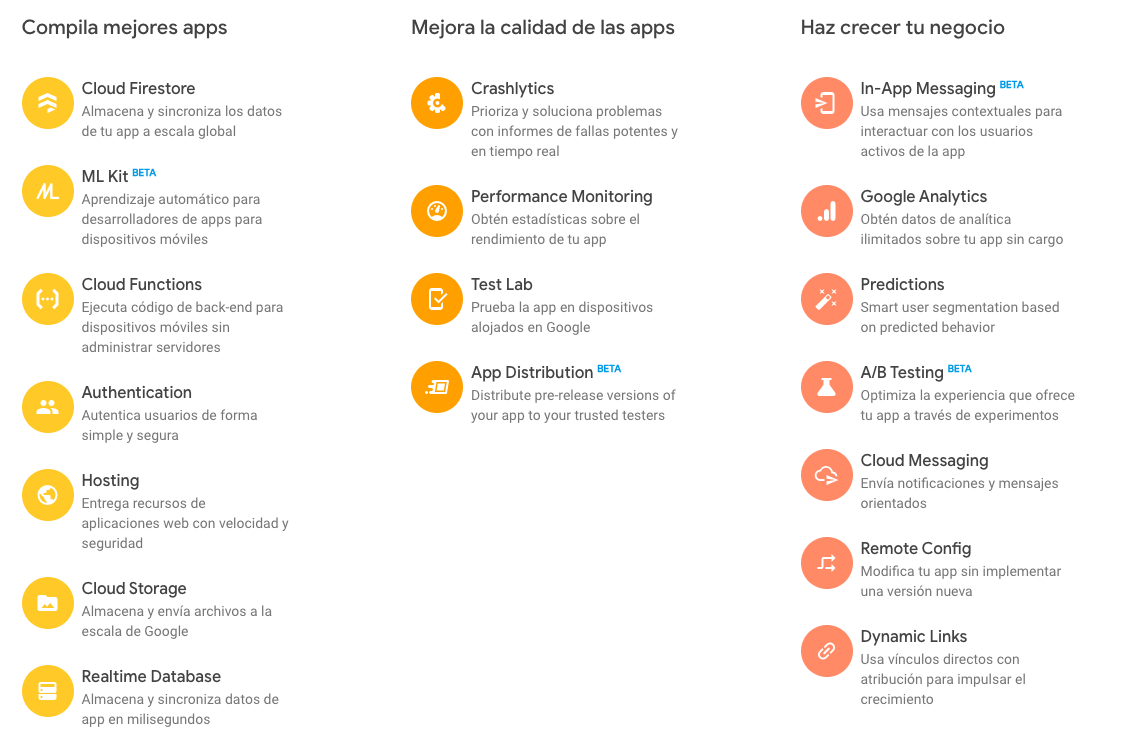


Figura 27: Servicios disponibles en Firebase

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: [Google https://firebase.google.com/docs/database?hl=es-419 ] Es una base de datos en tiempo real, 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 [https://github.com/FirebaseExtended/firebase-arduino] 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.

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.

<https://medium.com/@sebastianpaduano/firebase-qu%C3%A9-es-y-c%C3%B3mo-usarlo-bfdaed2a3f76>

<https://openwebinars.net/blog/usar-angular-firebase-desarrollo-iot/>

**3.6.3. Desarrollo móvil**

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

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 hibridas son aplicaciones que se desarrollan para mas 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 que forma se desarrollará una aplicación móvil, considerando que esta deberá estar disponible en los sistemas iOS y Android, en la tabla se resumen algunas de sus características.

Tabla 6: Comparativa de características entre aplicaciones móviles hibridas 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. |

[Fuente tabla <https://blog.irontec.com/apps-hibridas-vs-apps-nativas-un-breve-analisis-comparativo-de-tecnologias-moviles/>]

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 hibridas, puesto que la aplicación no requiere de avanzadas funciones (principalmente se usara 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 hibridas 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 hibridas más famosas tenemos:

<http://www.pixelgrafia.com/post/105_los-5-mejores-marcos-de-aplicaciones-hibridas-para-crear-aplicaciones-moviles-en-2020>

[buen resumen https://medium.com/hackernoon/top-10-best-mobile-app-development-frameworks-in-2019-612b95cf930f]

* **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, cuanta 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. Esta genera componentes de UI nativos para cada sistema, lo que proporciona interfaces fluidas y rápidas. Actualmente es el framework para desarrollo de aplicaciones móviles mas 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 aun 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 maás factibles a utilizar son Ionic y React Native, 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.

[Ionic vs React native https://blog.logrocket.com/react-native-vs-ionic/#:~:text=React%20Native%20provides%20better%20performance,components%2C%20hence%20providing%20better%20performance. ]

# 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.

## 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 cuando se debe calentar o enfriar el agua (ver figura).

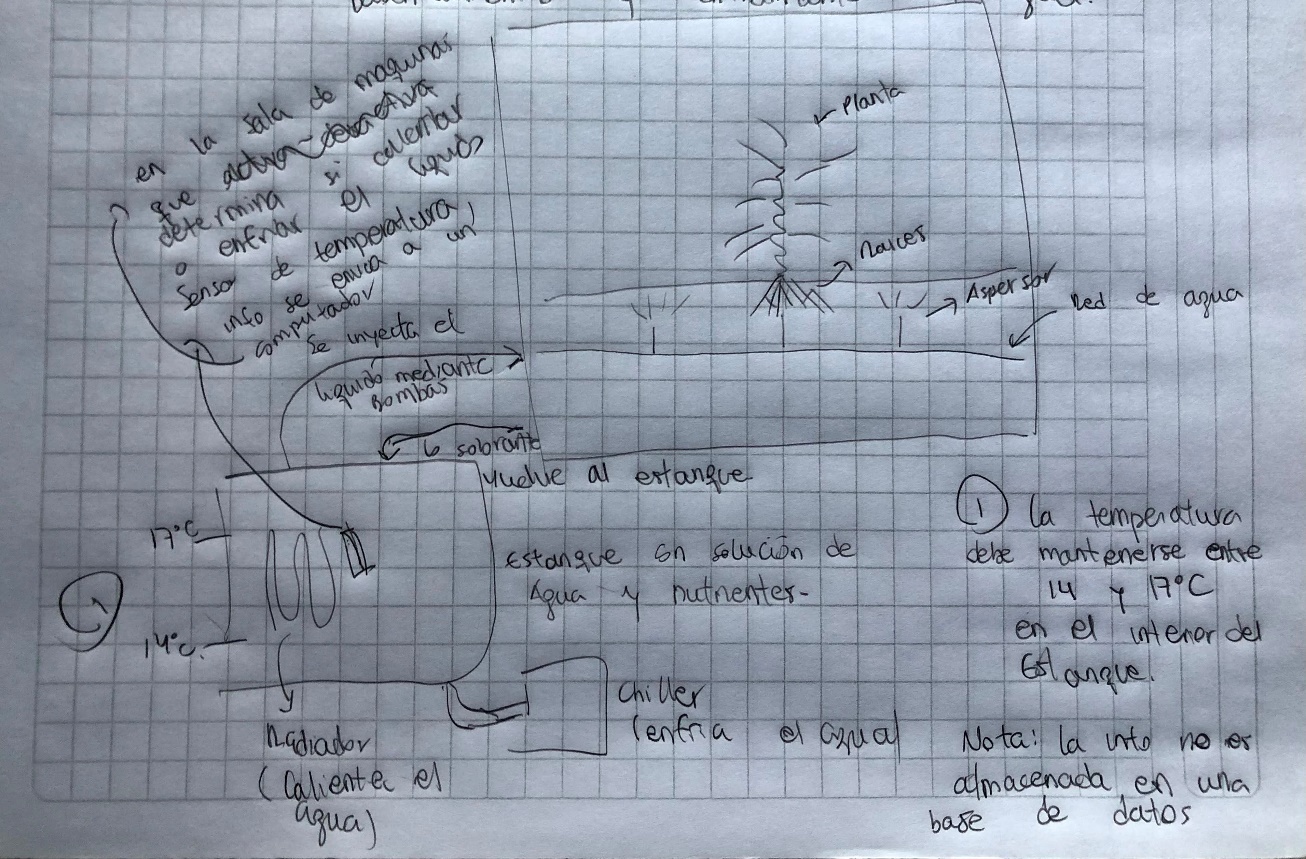


Figura 28: Funcionamiento del sistema de control de temperatura

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 no se efectúan mediciones de temperatura en la planta, sino que en el estanque, por lo que la temperatura real de la planta tiende a ser mas 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.

## 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 tipo 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 utilizara 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 cual 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 muestra el flujo que seguirán los datos en el sistema.



Figura 29: Flujo de datos del sistema

## Metodología

Para el desarrollo del prototipo se siguió una metodología de tipo cascada, donde 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.

La investigación consiste en una revisión sistemática, donde se busco 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 microncontroladores y sensores y su posterior adquisición, además de la selección 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 app 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 mas altos.

En la etapa de correcciones se reflexionará respecto de los resultados obtenidos y se efectuaran 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.

## 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.

<https://github.com/jelbo96/cerveceriaIngDeSoftware/blob/master/Documentos/Especificacion%20de%20requisitos.pdf> - tambien p32 tesis matamala

### Requerimientos funcionales

* + - 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.
* RNF6: El dispositivo debe reconectarse automáticamente a WiFi luego de una perdida de conexión.
* RNF7: El dispositivo deberá indicar una pérdida de conexión.
  + - 1. **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 graficas con datos anteriores.

### Requerimientos no funcionales

* + - 1. **Requerimientos del dispositivo de toma de datos**
* RNF1: El dispositivo deberá poder enviar datos por periodos prolongados.
  + - 1. **Requerimientos de la aplicación móvil**
* RNF2: La aplicación deberá ser compatible con los sistemas operativos iOS y Android en versiones recientes.

## Casos de uso

<https://github.com/jelbo96/cerveceriaIngDeSoftware/blob/master/Documentos/Documentacion%20Analisis%20y%20Dise%C3%B1o.pdf>

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 graficas de los sensores y configurar sensores a mostrar.

**4.5.1. Ver datos de los sensores**

**[p34 tesis Matamala tiene ideas para parrafos]**

**Descripción**

Tabla 7: Descripción caso de uso “Ver datos de los sensores”

|  |  |
| --- | --- |
| **Caso de uso:** |  |
| **Actores:** |  |
| **Precondición:** |  |
| **Resumen:** |  |
| **Tipo:** |  |

**Flujo normal de los eventos**

Tabla 8: Flujo normal de los eventos caso de uso “Ver datos de los sensores”

|  |  |
| --- | --- |
| **Actor** | **Sistema** |
|  |  |
|  |  |
|  |  |

**4.5.2. Ver gráficas de los sensores**

**Descripción**

Tabla 9: Descripción caso de uso "Ver gráficas de los sensores"

|  |  |
| --- | --- |
| **Caso de uso:** |  |
| **Actores:** |  |
| **Precondición:** |  |
| **Resumen:** |  |
| **Tipo:** |  |

**Flujo normal de los eventos**

Tabla 10: Flujo normal de los eventos caso de uso "Ver gráficas de los sensores"

|  |  |
| --- | --- |
| **Actor** | **Sistema** |
|  |  |
|  |  |
|  |  |

**4.5.3. Configurar sensores a mostrar**

**Descripción**

Tabla 11: Descripción caso de uso “Configurar sensores a mostrar”

|  |  |
| --- | --- |
| **Caso de uso:** |  |
| **Actores:** |  |
| **Precondición:** |  |
| **Resumen:** |  |
| **Tipo:** |  |

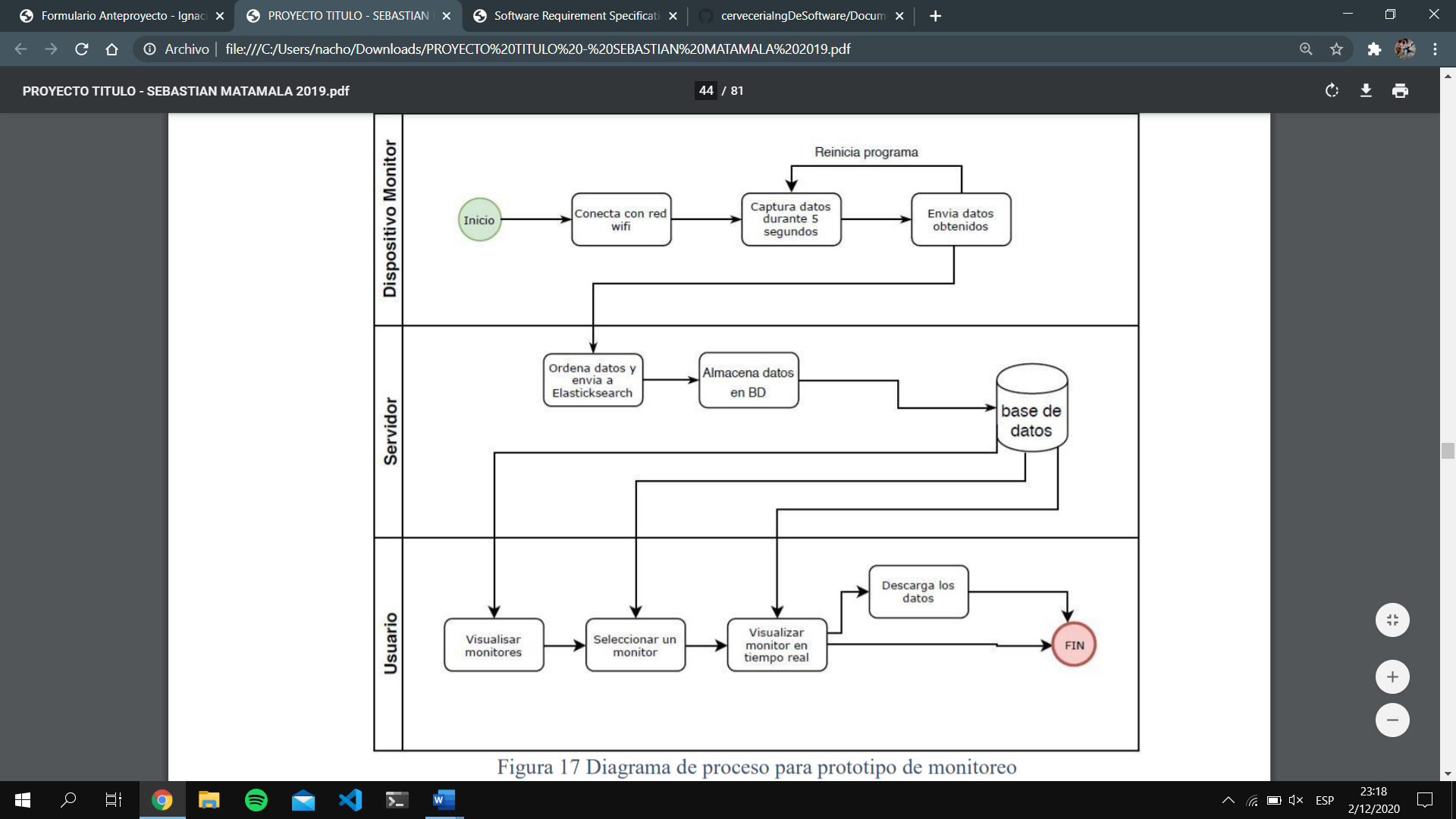
**Flujo normal de los eventos**

Tabla 12: Flujo normal de los eventos caso de uso “Configurar sensores a mostrar”

|  |  |
| --- | --- |
| **Actor** | **Sistema** |
|  |  |
|  |  |
|  |  |

## Modelo de procesos

[Ejemplo de lo que hay que hacer]



## Descripción de los actores del sistema

## Arquitectura del sistema

Arquitectura del sistema

### Hardware

**4.8.1.1. Dispositivos utilizados**

**4.8.1.2. Diagramas de conexión de dispositivos**

Imagen que contiene circuito, computadora

Descripción generada automáticamente

Figura 30: Diagrama de conexión usando Arduino UNO y ESP-01

Diagrama

Descripción generada automáticamente

Figura 31: Diagrama de conexión usando NodeMCU

**4.8.1.3. Diagrama de despliegue**

(se muestra diagrama sobre cómo se conecta, recibe datos y los envía a internet -> al otro lado un teléfono móvil recibiendo la info)

### 4.8.2. Software

**4.8.2.1. Tecnologías utilizadas**

(arduino, javascript, firebase ,angular, etc)

**4.8.2.2. Diagrama de despliegue**

(Mostrar como interactúan los elementos software (apis, bases de datos, tiempo real, etc))

**4.8.2.3. Aplicación móvil**

(Hacer diagramas y explicar como funciona internamente react native , framework para la interfaz, capturas de la interfaz, etc)

* Navegación
* Secciones de la interfaz (3, que tienen que ver con los casos de uso)

(En esta parte mostrar capturas y describir que info tienen y como funcionan)

# Pruebas

## Validación con soluciones existentes

<https://scholar.google.cl/scholar?hl=es&as_sdt=0%2C5&q=agricultura+precisi%C3%B3n+aplicada+cultivos+invernaderos&btnG=>

<http://repositorio.urp.edu.pe/bitstream/handle/URP/2294/TESIS_MORA-ROSAS.pdf?sequence=1&isAllowed=y>

* En base a información entregada por terceros se puede indicar que mi solución da información valida, que sirve……, entregar estudios que lo validen y demostrar que aspectos técnicos de mi solución son mejores.

## Plantas y crecimiento

Hola

## Retardo en toma de datos

Hola

## Consumo de energía

Hola, se realizaron pruebas con una batería de 5000mah, donde no se considero ninguna optimización en el consumo de energía, obteniendo datos cada 10 segundos se obtuvo una autonomía de 2 dias sin el uso de pantalla lcd, y de 1 día utilizando esta pantalla.

# Mejoras

Dado que la primera versión del prototipo tiene aspectos que se pueden mejorar, estos fueron abordados para la realización de una segunda versión, 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.

## Cambios de sensores

[DECIR QUE SENSORES SE AÑADIERON/QUITARON Y PORQUE]

## Mejoras de algoritmos

Como se indico en el capitulo 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 mas aun 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 mas datos cuando existe una variabilidad mas 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.

* + 1. **Modos de ahorro de energía**

[Acá mostrar los modos de ahorro de energía disponibles en el NodeMCU y cual se va a utilizar/utilizo]

* + 1. **Algoritmos para definir intervalos en la obtención de datos**

[FIJO, VARIABLE, DINAMICO] Explicarlos mediante pseudocodigo, decir que el dinámico es el mejor y mostrar las pruebas que se realizaron y los resultados obtenidos

## Definición de la solución final

# Conclusión

hOLA

# Referencias

# ANEXOS

[PONER FOTOS DE LA VISITA, QUE SE VEAN LAS TEMPERATURAS DE LOS MEDIDORES QUE TIENEN Y LO QUE MEDIA EN ESE INSTANTE LA APP]