

**CARRERA DE ESPECIALIZACIÓN EN  
INTERNET DE LAS COSAS**

**MEMORIA DEL TRABAJO FINAL**

**Monitoreo y gestión remota de  
controladores de clima y riego**

**Autor:**  
**Ing. Martín A. Brocca**

Director:  
Mg. Ing. Juan Carlos Brocca

Jurados:  
A ser designado  
A ser designado  
A ser designado

*Este trabajo fue realizado en la Ciudad Autónoma de Buenos Aires,  
entre junio de 2022 y mayo de 2023.*



## *Resumen*

En la presente memoria se describe el diseño, desarrollo e implementación de un sistema de control y monitoreo de clima y riego para un invernadero de tipo hogareño. El trabajo se compone de una colección de sensores y actuadores basados en hardware de bajo costo y una plataforma de software de código abierto que permite automatizar el proceso y adaptarlo a diferentes cultivos.

Para completar este proyecto se utilizaron técnicas de selección e integración de circuitos, desarrollo de firmware, evaluación de software y diseño de automatizaciones.



# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Motivación . . . . .	1
1.3. Estado del arte . . . . .	2
1.4. Objetivos y alcance . . . . .	5
<b>2. Introducción específica</b>	<b>7</b>
2.1. Tecnologías de comunicación . . . . .	7
2.1.1. Tecnologías Wi-Fi . . . . .	7
2.1.2. Protocolo MQTT . . . . .	7
2.1.3. Protocolo HTTP . . . . .	8
2.1.4. Protocolo SSL/TLS . . . . .	9
2.2. Componentes de hardware utilizado . . . . .	10
2.2.1. Raspberry Pi . . . . .	10
2.2.2. Módulos ESP . . . . .	11
2.2.3. Sensores y actuadores . . . . .	11
2.3. Tecnologías de software aplicadas . . . . .	12
2.3.1. ThingsBoard . . . . .	12
2.3.2. Arduino IDE . . . . .	12
2.3.3. Telegram . . . . .	13
2.4. Requerimientos . . . . .	13
<b>3. Diseño e implementación</b>	<b>15</b>
3.1. Arquitectura del sistema . . . . .	15
3.2. Detalle de los módulos de hardware . . . . .	15
3.3. Detalle del firmware desarrollado . . . . .	15
3.4. Selección y configuración del software . . . . .	15
3.5. Ciberseguridad del sistema . . . . .	15
<b>4. Ensayos y resultados</b>	<b>17</b>
4.1. Banco de pruebas . . . . .	17
4.2. Pruebas unitarias . . . . .	17
4.3. Pruebas de sistema . . . . .	17
4.4. Comparativa con el estado de arte . . . . .	17
<b>5. Conclusiones</b>	<b>19</b>
5.1. Resultados obtenidos . . . . .	19
5.2. Trabajo futuro . . . . .	19
<b>Bibliografía</b>	<b>21</b>



# Índice de figuras

1.1. Invernadero hogareño . . . . .	2
1.2. Invernadero inteligente controlado por IoT . . . . .	3
1.3. Sistema de control de irrigación de Growlink . . . . .	4
1.4. Sistema multisensor de Grodan . . . . .	4
2.1. Pila de protocolos para IoT. . . . .	7
2.2. Arquitectura del protocolo MQTT. . . . .	8
2.3. Proceso de autenticación de TLS . . . . .	9
2.4. Raspberry Pi. . . . .	10
2.5. Módulo de desarrollo ESP32. . . . .	11
2.6. Principales sensores y actuadores empleados en el invernadero. . . . .	12



# Índice de tablas

1.1. Análisis del estado del arte . . . . .	4
2.1. Especificaciones técnicas de la Raspberry Pi 4B. . . . .	10
2.2. Especificaciones de los microcontroladores ESP. . . . .	11



# Capítulo 1

## Introducción general

En este capítulo se introducen conceptos asociados al uso de Internet de las Cosas (IoT) en invernaderos, la motivación del cliente y cuál es el estado del arte. Asimismo, se explican los objetivos y el alcance establecidos.

### 1.1. Introducción

El término Internet de las Cosas (IoT) se refiere a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana [1]. IoT tiene una amplia variedad de campos de aplicación, entre los cuales se destaca la agricultura inteligente aplicada a invernaderos.

Los invernaderos modernos son sistemas de cultivo intensivo diseñados para alcanzar una alta eficiencia y productividad. Debido a su capacidad para mantener las condiciones ambientales en niveles óptimos o subóptimos, facilitan la producción de plantas a lo largo de todo el año en forma independiente a las condiciones climáticas externas [2].

La aplicación de IoT a invernaderos ha demostrado una mejora sustancial en la eficiencia de la gestión de los cultivos al mismo tiempo que ha logrado acelerar la producción y reducir sus costos [3]. Además de las ventajas mencionadas, los invernaderos pueden impactar de forma positiva a los entusiastas de la jardinería, ya que proporcionan beneficios tanto físicos como anímicos especialmente durante las temporadas invernales o de baja temperatura [4].

### 1.2. Motivación

El cliente de este proyecto es un jubilado reciente que decidió comenzar una nueva vida en una finca rural. Su intención es convertir la propiedad en una residencia sustentable, capaz de producir diferentes tipos de plantas: por un lado, hortalizas para abastecer el consumo familiar y por otro, especies de árboles para reforestación.

Para lograrlo con mínima intervención humana en el proceso, surge la necesidad de instalar un invernadero que automatice el control de los cultivos y además tenga la capacidad de adaptar las condiciones climáticas y de riego a los diversos tipos de plantas a sembrar.

En la figura 1.1 se observa un vivero hogareño típico de producción y dimensiones similares al propuesto.



FIGURA 1.1. Invernadero hogareño<sup>1</sup>.

### 1.3. Estado del arte

Los invernaderos son estructuras diseñadas para controlar y proteger a las plantas del clima y otros factores ambientales adversos. Tradicionalmente, la temperatura, humedad e iluminación se controlaban de forma manual, lo que requería una gran cantidad de mano de obra y recursos.

Sin embargo, debido a los avances tecnológicos se ha popularizado el desarrollo de invernaderos capaces de ajustar las condiciones ambientales mediante el uso de sensores, actuadores y controladores. Estos dispositivos responden en función de configuraciones preprogramadas o a partir de datos en tiempo real.

Así el despliegue de este tipo de sistemas, denominados invernaderos inteligentes, se ha extendido enormemente en los últimos años debido a la eficiencia obtenida durante la producción y al incremento en la resiliencia de los cultivos [5].

Algunas posibles aplicaciones de IoT en viveros incluyen:

- Monitoreo y control del clima: distintos sensores miden la temperatura, humedad, iluminación y otros factores ambientales en el invernadero, para que luego diferentes actuadores automáticos ajusten el clima y creen las condiciones óptimas para el crecimiento de las plantas.
- Riego a demanda: la medición continua de la humedad del suelo permite activar sistemas de riego automatizados para mantener los niveles óptimos de humedad. Se logra así reducir el desperdicio de agua y disminuir los costos asociados.
- Seguimiento del crecimiento de las plantas: los sensores IoT pueden medir el crecimiento de las plantas y proporcionar información útil para la gestión del cultivo. Esto puede ayudar a identificar problemas de crecimiento temprano y a tomar medidas que eviten problemas mayores a futuro.

<sup>1</sup>Imagen bajo licencia de <https://www.istockphoto.com/>

- Control de plagas y enfermedades: monitorear sus niveles en el vivero y activar sistemas de control cuando se detecten problemas ayuda a reducir el uso de pesticidas y otros productos químicos.
- Fertirrigación: el sistema puede administrar fertilizantes o nutrientes al suelo de forma optimizada y precisa a través del riego, en base a configuraciones acordes a la plantación en curso o mediante sensores que midan las características del agua, el pH o la conductividad eléctrica entre otras.
- Automatización de tareas: los sistemas IoT pueden automatizar muchas tareas en el vivero, como la siembra, el trasplante y la recolección de plantas. Esto puede reducir los costos de mano de obra y mejorar la eficiencia de la producción.

En la figura 1.2 se representa un invernadero inteligente con sus respectivos sensores y actuadores.



FIGURA 1.2. Invernadero inteligente controlado por IoT<sup>2</sup>.

En el mercado internacional se encuentran diferentes proveedores que ofrecen soluciones para el desarrollo de invernaderos inteligentes. A la hora de comparar distintas opciones es necesario considerar los niveles de automatización requeridos, la facilidad de uso y las opciones de personalización. Adicionalmente, se debe considerar el costo y la compatibilidad con la infraestructura existente.

En la tabla 1.1 se observa una breve comparación entre los principales proveedores comerciales de servicios. Allí se observa que en general las soluciones presentadas ofrecen características similares siendo el costo el mayor diferenciador.

<sup>2</sup>Imagen adaptada de *Internet of Things Empowered Smart Greenhouse Farming* [6].

TABLA 1.1. Análisis del estado del arte.

Funcionalidad	Argus Controls [7]	Grodan [8]	Growlink [9]
Gestión de clima	Sí	Sí	Sí
Control de riego	Sí	Sí	Sí
Fertirrigación	Sí	Sí	Sí
Gestión de energía	No	Sí	Sí
Tamaño de mercado	Grande	Grande	Pequeño
Costo	\$\$\$\$	\$\$\$	\$\$

A manera de ejemplo, algunas soluciones comerciales ofrecen kits que pueden costar más de USD 200 por sensor, con gastos adicionales asociados al transporte y almacenamiento de datos. De esta forma, las redes inalámbricas de sensores pueden llegar a requerir presupuestos mayores a USD 10 000 por invernadero [10]. Las figuras 1.3 y 1.4 muestran diferentes componentes que se encuentran en el mercado.

FIGURA 1.3. Sistema de control de irrigación de Growlink<sup>3</sup>.FIGURA 1.4. Sistema multisensor de Grodan<sup>4</sup>.

Una solución alternativa es utilizar kits de IoT como los provistos por Arduino

<sup>3</sup>Imagen bajo licencia de <https://www.growlink.com/>

<sup>4</sup>Imagen bajo licencia de <https://www.grodan.com/>

[11], una plataforma de hardware y software de código abierto. Además, es necesario contar con una comunidad de usuarios que colaboren en el proceso de desarrollo. En contrapartida, para aplicar esta solución se requieren conocimientos de programación y electrónica no siempre disponibles para el cultivador promedio [10].

## 1.4. Objetivos y alcance

El propósito de este trabajo es el desarrollo de una plataforma capaz de controlar el clima y riego de un invernadero mediante el uso de sensores y actuadores. Estos dispositivos se comunican con una aplicación instalada en un servidor local que administra los parámetros y las alarmas del sistema.

Durante el proyecto se construyó un prototipo completo de invernadero con los siguientes elementos:

- Aplicación para el monitoreo, control de dispositivos, gestión de alarmas y automatización.
- Control de usuarios, permisos y accesos a la plataforma.
- Interfaz gráfica para acceso y control de la plataforma.
- Análisis, investigación y elección del hardware para los sensores y actuadores.

El trabajo no incluyó:

- Instalación en sitio de los sistemas desarrollados.
- Implementación de métodos de control basados en condiciones climatológicas externas.
- Desarrollo o implementación de modelos analíticos o predictivos de las condiciones del vivero.
- Diseño o instalación de conexiones que no sean por Wi-Fi (LTE/5G).



## Capítulo 2

# Introducción específica

En este capítulo se describen las tecnologías y los protocolos de comunicación, los componentes de hardware, las herramientas de software y los requerimientos para la realización del proyecto.

### 2.1. Tecnologías de comunicación

Los principales protocolos de IoT empleados en el trabajo y su posicionamiento en la pila de comunicaciones se muestran resaltados en la figura 2.1.

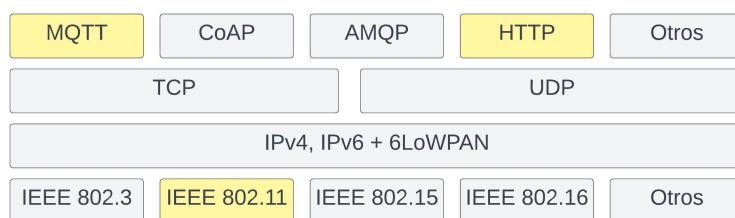


FIGURA 2.1. Pila de protocolos para IoT<sup>1</sup>.

#### 2.1.1. Tecnologías Wi-Fi

El estándar IEEE 802.11 para redes inalámbricas de área local (WLAN) es conocido comercialmente como Wi-Fi y presenta dos modos de operación [13]:

- Infraestructura: uno o más *access points* (AP) actúan como puente entre la red cableada y la red inalámbrica. Todas las comunicaciones entre los dispositivos conectados a la red se realizan a través de los APs.
- Ad-hoc: cada nodo puede realizar una conexión directa con otro, sin necesidad de un AP central. En este caso, los nodos se organizan en una red donde todos son capaces de enrutar los paquetes.

#### 2.1.2. Protocolo MQTT

MQTT fue desarrollado en 1999 con el objetivo principal de crear un protocolo muy eficiente desde el punto de vista del uso del ancho de banda y de muy bajo consumo de energía. Por estas razones es adecuado para el uso en IoT [14].

<sup>1</sup>Gráfico creado en base a una imagen tomada de [12]

Su funcionamiento se basa en el paradigma de publicación-suscripción que consiste en desvincular un cliente que publica un mensaje (publicador) de otros clientes que reciben el mensaje (suscriptores). Sumado a esto, MQTT es un protocolo asincrónico, lo que significa que el cliente puede seguir operando mientras espera un nuevo mensaje.

Un componente principal del protocolo es el *broker* cuya función primaria es la de recibir los mensajes de los publicadores y enviarlos a los suscriptores. Para realizar esta tarea, el *broker* utiliza temas o *topics* para agrupar clientes que necesitan recibir los mismos mensajes. De esta manera, el *topic* es un canal virtual que conecta a los publicadores con sus suscriptores [14].

En la figura 2.2 se observa la arquitectura del protocolo.

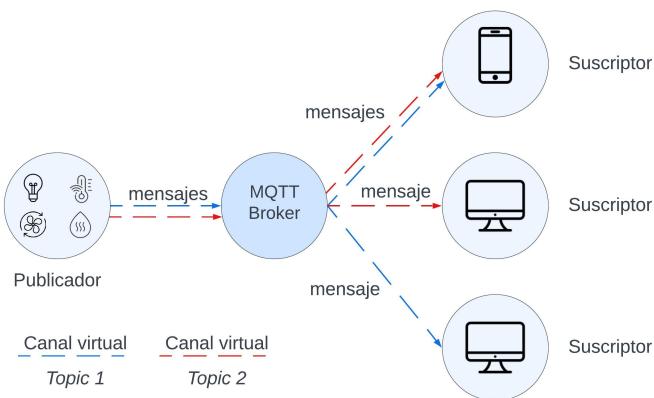


FIGURA 2.2. Arquitectura del protocolo MQTT<sup>2</sup>.

### 2.1.3. Protocolo HTTP

El *Hypertext Transfer Protocol* (HTTP)[15] es un protocolo utilizado en la Web para el desarrollo de aplicaciones y está basado en el paradigma cliente-servidor. Aquí el cliente emplea un agente intermediario (por lo general un *browser*) para realizar un pedido de información y el servidor proporciona una respuesta. Esto se conoce con el nombre de modelo *request/response*.

HTTP es un protocolo que no guarda información de estado, esto significa que el servidor no es capaz de reconocer la relación entre múltiples pedidos de un mismo usuario [16].

En la actualidad, HTTP se utiliza en conjunto con la arquitectura REST (*Representational State Transfer*) [17] para facilitar la interacción entre distintas entidades sobre servicios basados en red. Esta asociación permite que los dispositivos interactúen mediante funciones estándares de tipo CRUD (*create, read, update, delete*) [18]. Dichas funciones a su vez se traducen a los métodos HTTP POST, GET, PUT y DELETE respectivamente [19].

<sup>2</sup>Gráfico creado en base a una imagen tomada de [14]

### 2.1.4. Protocolo SSL/TLS

*Secure Socket Layer/Transport Layer Security* (SSL/TLS) [20] es un protocolo criptográfico que proporciona seguridad de extremo a extremo de los datos enviados entre aplicaciones a través de Internet. TLS evolucionó a partir de *Secure Socket Layer* (SSL), que fue desarrollado originalmente por Netscape Communications Corporation en 1994 para proteger las sesiones web.

SSL 1.0 nunca se lanzó públicamente, mientras que SSL 2.0 fue reemplazado rápidamente por SSL 3.0 que proporcionó las bases para la posterior creación de TLS.

Cabe señalar que TLS no protege los datos en los sistemas finales, simplemente garantiza la entrega segura de datos a través de Internet y al mismo tiempo evita posibles escuchas y/o alteraciones del contenido. TLS normalmente se implementa sobre TCP [21] para cifrar los protocolos de la capa de aplicación, como por ejemplo HTTP.

TLS utiliza una combinación de criptografía simétrica y asimétrica que proporciona un buen compromiso entre rendimiento y seguridad al momento de transmitir la información [22]. Para mayor protección es deseable que un cliente que se conecta a un servidor pueda validar la veracidad de la clave pública ofrecida por este. Normalmente dicha verificación se lleva a cabo por medio de un certificado digital X.509 [23] emitido por un tercero de confianza denominado Autoridad Certificadora (CA). Dicha CA está encargada de afirmar la autenticidad de la clave pública. En los casos en los que no se dispone de una CA, un servidor puede usar un certificado autofirmado en el que el cliente debe confiar explícitamente [22].

En la figura 2.3 se detalla el esquema de autenticación y verificación con certificados e inicio de una conexión segura.

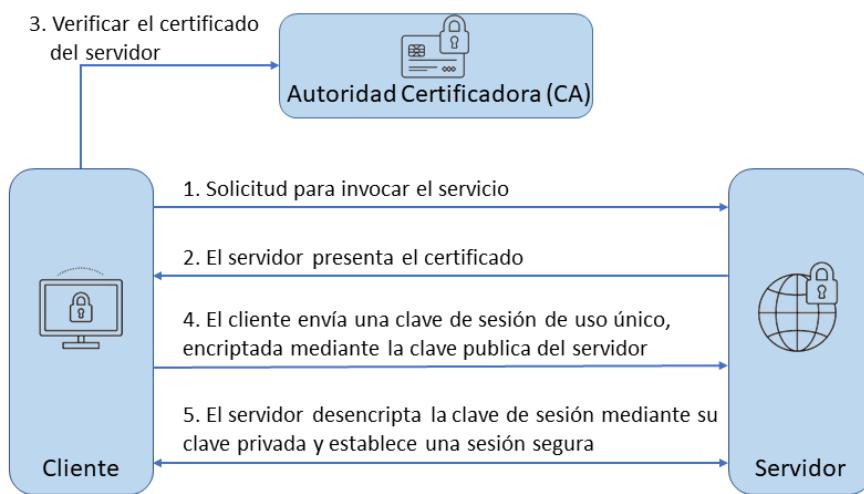


FIGURA 2.3. Proceso de autenticación de TLS<sup>3</sup>.

<sup>3</sup>Gráfico creado en base a una imagen tomada de <http://www.herongyang.com/PKI/HTTPS-Communication-Data-Encryption.html>

## 2.2. Componentes de hardware utilizado

### 2.2.1. Raspberry Pi

Se denomina así a una serie de computadoras monoplaca o computadoras de placa simple (SBC, por *Single Board Computer*) de bajo costo desarrollados por la Raspberry Pi Foundation [24]. Una de sus principales características es proveer un conjunto de pines de GPIO (*general purpose input/output* que permiten controlar componentes electrónicos y otros dispositivos en el ámbito de Internet de las Cosas. A pesar de su reducido tamaño, la Raspberry Pi ofrece una capacidad de procesamiento comparable a una computadora de escritorio y es por ello que su uso se ha expandido en proyectos que incluyen domótica, *edge computing* o aplicaciones industriales [25].

En la figura 2.4 se muestra una Raspberry Pi modelo 4B similar a la utilizada en el trabajo y en la tabla 2.1 se listan sus principales características:

TABLA 2.1. Especificaciones técnicas de la Raspberry Pi 4B.

Categoría	Especificación
Procesador	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64 bits SoC @ 1,8 GHz
Memoria SDRAM	1, 2, 4 u 8 GB LPDDR4-3200
Wi-Fi	2,4 GHz and 5,0 GHz IEEE 802.11ac
Bluetooth	5,0, BLE
Ethernet	Gigabit, con soporte opcional para POE
USB	2 puertos 3.0 ; 2 puertos 2.0
GPIO	Conector de 40 pines
HDMI	2 puertos micro-HDMI
Alimentación	5 V USB y GPIO
Temperatura	0 °C a 50 °C

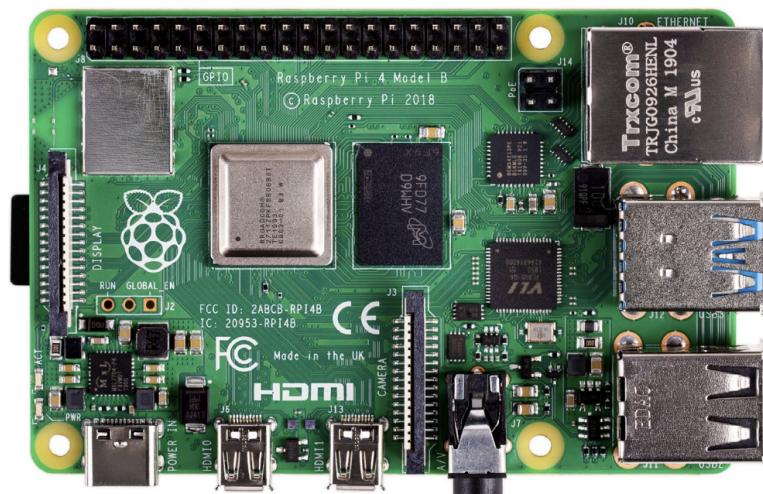


FIGURA 2.4. Raspberry Pi<sup>4</sup>.

### 2.2.2. Módulos ESP

ESP es una familia de microcontroladores de baja potencia desarrollada por la empresa china Espressif. Estos chips cuentan con una amplia variedad de usos en IoT tanto en el ámbito profesional/industrial como en el de los aficionados [26] [27].

En la tabla 2.2 se observa una comparación entre los modelos ESP32-WROOM-32 y ESP8266 utilizados en este trabajo.

TABLA 2.2. Especificaciones de los microcontroladores ESP.

Características	ESP32 [26]	ESP8266 [27]
Procesador	2 x Xtensa 32 bits LX6	Tensilica L106 32 bits
Memoria ROM	448 KB	No dispone
Memoria SRAM	520 KB	160 KB
Wi-Fi	802.11 b/g/n	802.11 b/g/n
Bluetooth	4.2, BLE	No dispone
GPIOs	34 pines	17 pines
Consumo normal	~500 mA	~80 mA
Temperatura de operación	-40 °C a 85 °C	-40 °C a 125 °C
Sistema operativo	freeRTOS	freeRTOS
Costo en USD	\$ 6 a \$ 12	\$ 3 a \$ 6

El modelo ESP8266 es una versión anterior al ESP32 y ofrece una menor variedad de características que este último. Sin embargo, constituye una opción económica cuando los requerimientos de conectividad y rendimiento van de la mano con sus prestaciones.

En la figura 2.5 se observa un módulo de desarrollo de la familia ESP32.



FIGURA 2.5. Módulo de desarrollo ESP32.

### 2.2.3. Sensores y actuadores

Los principales sensores y actuadores empleados en el sistema del invernadero inteligente son los siguientes:

- DHT22: módulo básico y económico para determinar los valores de temperatura y humedad en forma digital. Utiliza un sensor de humedad capacitivo y un termistor para medir el aire circundante y entrega una señal digital en el pin de datos de acuerdo al valor calculado. Funciona con una alimentación de 3,3 a 6 VDC y su rango de medición es de -40 °C a 80 °C y de 0 a 100 % de humedad relativa [28].

- Sensor capacitivo de humedad del suelo: compuesto de un material resistente a la corrosión, mide la humedad del suelo indirectamente por medio de la capacidad observada. Opera con una alimentación de 3,3 a 5,5 VDC [29].
- Válvula solenoide de dos vías: dispositivo neumático para controlar el flujo de líquidos o gases que se acciona eléctricamente. Para poder operarlo, se utiliza un relé que es un instrumento electromecánico que actúa como interruptor controlado por un circuito eléctrico [30][31].

La figura 2.6 muestra imágenes de los componentes listados previamente.



FIGURA 2.6. Principales sensores y actuadores empleados en el invernadero.

## 2.3. Tecnologías de software aplicadas

### 2.3.1. ThingsBoard

ThingsBoard es una plataforma de código abierto que permite el desarrollo, administración y expansión de proyectos de IoT. Esta aplicación permite la gestión de las comunicaciones, el almacenamiento y la visualización de los datos que provienen de los sensores u otros dispositivos que formen parte del sistema [32]. Se ofrece en las siguientes versiones:

- Professional Edition: es la versión comercial, con varios tipos de licenciamiento y costos según sea el sistema a implementar. Esta edición posee soporte técnico, ilimitada cantidad de dispositivos a conectar y soporte para almacenamiento híbrido entre otras funcionalidades.
- Cloud: similar a la edición profesional, pero alojada en la nube de ThingsBoard. En este caso el proveedor se encarga del manejo de los componentes de la plataforma.
- Community edition: es la edición libre de licenciamientos y la que se utilizó en este trabajo. Si bien no tiene limitaciones en cuanto a la cantidad de dispositivos a conectar, carece de ciertas funcionalidades como por ejemplo la descarga de datos de dispositivos desde la interfaz web o la ejecución programada de tareas (*scheduler*).

### 2.3.2. Arduino IDE

Arduino IDE (*Integrated Development Environment*) es un software de código abierto que se utiliza para escribir y cargar código a placas Arduino. Sin embargo por medio de la instalación de paquetes de expansión, el IDE soporta hardware de

terceros entre los que encuentran las módulos ESP32/ESP8266 [33] empleados en este proyecto.

El código de los programas se realiza en lenguaje C o C++ y los archivos resultantes se denominan *sketches*. Estos son compilados y cargados en las placas desde el mismo IDE [34].

Este software es una herramienta fácil de utilizar tanto por usuarios experimentados como por principiantes. Es frecuentemente empleada por aquellos que se inician en la programación electrónica y la robótica o al momento de construir prototipos interactivos [35].

### 2.3.3. Telegram

Telegram es una aplicación de mensajería multiplataforma rápida, simple y gratuita. Está basada en la nube y cuenta con sincronización constante, lo que significa que se puede acceder a los mensajes desde diferentes dispositivos simultáneamente.

Tanto el código de los clientes de Telegram como el de su API son abiertos, lo que permite crear otras aplicaciones a partir de ellos. Adicionalmente cuenta con una API para *bots* que facilita la implementación de herramientas basadas en Telegram, la integración de servicios o la realización pagos [36].

## 2.4. Requerimientos

A continuación se listan los principales requerimientos funcionales, no funcionales y de documentación del proyecto:

- Requerimientos funcionales:
  1. El estado del sistema podrá ser consultado desde Internet.
  2. La aplicación soportará múltiples usuarios de forma concurrente.
  3. La aplicación permitirá crear roles de usuarios con diferentes permisos.
- Requerimientos no funcionales:
  1. El rango de tensión de alimentación de los nodos será de 3,3 a 5 VDC.
  2. El sistema de riego operará con una tensión de alimentación de 12 VDC.
  3. El sistema estará basado en software de código abierto.
  4. El firmware deberá desarrollarse en plataformas de código abierto.
  5. El trabajo se realizará sobre dispositivos de bajo costo y fácil reposición.
  6. Los datos se almacenarán localmente.
  7. La aplicación soportará MQTT.
  8. Los sensores de humedad del suelo tendrán una protección IP65 [37].
- Requerimientos de documentación:

1. Los manuales y/o guías estarán redactados en inglés.

## Capítulo 3

# Diseño e implementación

En este capítulo se presentan los detalles del diseño de los nodos sensores y actuadores que conforman el trabajo, como así también los de la implementación de la aplicación Thingsboard.

- 3.1. Arquitectura del sistema**
- 3.2. Detalle de los módulos de hardware**
- 3.3. Detalle del firmware desarrollado**
- 3.4. Selección y configuración del software**
- 3.5. Ciberseguridad del sistema**



## **Capítulo 4**

# **Ensayos y resultados**

En este capítulo se explica la metodología de pruebas aplicada tanto a los componentes individuales como al sistema implementado, finalizando con una comparativa con el estado del arte.

- 4.1. Banco de pruebas**
- 4.2. Pruebas unitarias**
- 4.3. Pruebas de sistema**
- 4.4. Comparativa con el estado de arte**



## **Capítulo 5**

# **Conclusiones**

En este capítulo se muestran las conclusiones sobre el trabajo realizado. A su vez se presentan algunas modificaciones o mejoras como posible trabajo futuro

### **5.1. Resultados obtenidos**

### **5.2. Trabajo futuro**

Acá se indica cómo se podría continuar el trabajo más adelante.



# Bibliografía

- [1] Scott Eldridge y Lyman Chapin Karen Rose. «The Internet of Things: An Overview». En: (2015).
- [2] Krishna Nemali. «History of Controlled Environment Horticulture: Greenhouses». En: *HortScience* 57.2 (2022), págs. 239 -246. DOI: [10.21273/HORTSCI16160-21](https://doi.org/10.21273/HORTSCI16160-21). URL: <https://journals.ashs.org/hortsci/view/journals/hortsci/57/2/article-p239.xml>.
- [3] Chrysanthos Maraveas y Thomas Bartzanas. «Aplicación de internet de las cosas (IoT) para entornos de invernadero optimizados». En: *Magna Scientia UCEVA* 2 (dic. de 2022), págs. 253-268. DOI: [10.54502/msuceva.v2n2a11](https://doi.org/10.54502/msuceva.v2n2a11).
- [4] John W. Bartok. *Greenhouses for Homeowners and Gardeners*. Natural Resource, Agriculture, y Engineering Service (NRAES), 2000-06.
- [5] Intel. *¿Qué son los invernaderos inteligentes?* <https://agrofacto.com/invernaderos-inteligentes/m>. 2022. (Visitado 20-03-2023).
- [6] Rakiba Rayhana, Gaozhi Xiao y Zheng Liu. «Internet of Things Empowered Smart Greenhouse Farming». En: *IEEE Journal of Radio Frequency Identification* 4.3 (2020), págs. 195-211. DOI: [10.1109/JRFID.2020.2984391](https://doi.org/10.1109/JRFID.2020.2984391).
- [7] Argus Controls. <https://arguscontrols.com/>. (Visitado 20-03-2023).
- [8] Grodan. <https://www.grodan.com/>. (Visitado 20-03-2023).
- [9] Growlink. <https://www.growlink.com/>. (Visitado 20-03-2023).
- [10] Chet Udell y Alan Dennis Lloyd Nackley. <https://diggermagazine.com/the-smart-greenhouse/>. 2020. (Visitado 20-03-2023).
- [11] Arduino. <https://www.arduino.cc/>. (Visitado 20-03-2023).
- [12] Nitin Naik. «Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP». En: *2017 IEEE International Systems Engineering Symposium (ISSE)*. 2017, págs. 1-7. DOI: [10.1109/SysEng.2017.8088251](https://doi.org/10.1109/SysEng.2017.8088251).
- [13] Shanna L. «Comparative Analysis of Infrastructure and Ad-Hoc Wireless Networks». En: *ITM Web of Conferences* 25, 01009 (2019).
- [14] www.survivingwithandroid.com. <https://www.survivingwithandroid.com/mqtt-protocol-tutorial/>. (Visitado 20-03-2023).
- [15] The Internet Society. *Hypertext Transfer Protocol – HTTP/1.1*. RFC. URL: \url{https://www.ietf.org/rfc/rfc2616.txt} (visitado 20-03-2023).
- [16] <https://www.oreilly.com/library/view/javaserver-pages-3rd/0596005636/ch02s01.html>. (Visitado 20-03-2023).
- [17] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Inf. téc. 2000. Cap. Chapter 5: Representational State Transfer (REST).

- [18] Jasenka Dizdarević et al. «A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration». En: *ACM Comput. Surv.* 51.6 (ene. de 2019). ISSN: 0360-0300. DOI: [10.1145/3292674](https://doi.org/10.1145/3292674). URL: <https://doi.org/10.1145/3292674>.
- [19] Dimitrios Glaroudis, Athanasios Iossifides y Periklis Chatzimisios. «Survey, comparison and research challenges of IoT application protocols for smart farming». En: *Computer Networks* 168 (2020), pág. 107037. ISSN: 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2019.107037>. URL: <https://www.sciencedirect.com/science/article/pii/S1389128619306942>.
- [20] E. Rescorla T. Dierks. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC. 2008. URL: <https://www.rfc-editor.org/rfc/rfc5280>.
- [21] J. Postel. *Transmission Control Protocol*. RFC 793. Internet Engineering Task Force, 1981, pág. 85. URL: <http://www.rfc-editor.org/rfc/rfc793.txt>.
- [22] Internet Society. «TLS Basics». En: (). URL: <https://www.internetsociety.org/deploy360/tls/basics/>.
- [23] D. Cooper et. al. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC. 2008. URL: <https://www.rfc-editor.org/rfc/rfc5280>.
- [24] [https://es.wikipedia.org/wiki/Raspberry\\_Pi](https://es.wikipedia.org/wiki/Raspberry_Pi). (Visitado 20-03-2023).
- [25] <https://opensource.com/resources/raspberry-pi>. (Visitado 20-03-2023).
- [26] *ESP32 Series*. Datasheet. V4.2. Espressif. Ene. de 2023.
- [27] *ESP8266EX*. Datasheet. V6.9. Espressif. Feb. de 2023.
- [28] *Digital output relative humidity and temperature sensor/module*. DHT22. Aosong Electronics Co.,Ltd. URL: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>.
- [29] <https://smartbitbn.com/product/soil-humidity-sensor/>. (Visitado 20-03-2023).
- [30] <https://www.amazon.com/dp/B07BW21Z5M/>. (Visitado 20-03-2023).
- [31] <https://es.wikipedia.org/wiki/Rel%C3%A1o>. (Visitado 20-03-2023).
- [32] <https://thingsboard.io/docs/getting-started-guides/what-is-thingsboard/>. (Visitado 20-03-2023).
- [33] *Soporte de Arduino IDE para hardware de terceros*. URL: <https://github.com/per1234/ino-hardware-package-list/blob/master/ino-hardware-package-list.tsv>.
- [34] *Arduino Integrated Development Environment (IDE) v1*. URL: <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>.
- [35] *What is arduino software (IDE), and how use it ?* URL: <https://andprof.com/tools/what-is-arduino-software-ide-and-how-use-it/>.
- [36] *Telegram - Preguntas Frecuentes*. URL: <https://telegram.org/faq?setln=es#p-que-es-telegram-que-puedo-hacer-aqui>.
- [37] <https://www.iec.ch/ip-ratings>. (Visitado 20-03-2023).