

# 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 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. Protocolo MQTT	7
2.1.2. Protocolo HTTP	8
2.1.3. Protocolo SSL/TLS	8
2.1.4. Tecnologías Wi-Fi	9
2.2. Componentes de hardware utilizado	9
2.2.1. Raspberry Pi	9
2.2.2. Módulos ESP	10
2.2.3. Sensores y actuadores	12
2.3. Tecnologías de software aplicadas	12
2.3.1. ThingsBoard	12
2.3.2. Arduino IDE	13
2.3.3. Telegram	13
2.4. Requerimientos	13
<b>3. Diseño e implementación</b>	<b>15</b>
3.1. Arquitectura de la solución	15
3.1.1. Componentes del sistema	15
3.1.2. Protocolos de comunicación	16
3.2. Detalle de los módulos de hardware	17
3.2.1. Módulos sensores de humedad del suelo	18
3.2.2. Módulo controlador del riego	19
3.2.3. Módulo sensor de temperatura y humedad	20
3.2.4. Módulo controlador de clima	21
3.3. Desarrollo del firmware	23
3.4. Selección y configuración del software	23
3.5. Ciberseguridad del sistema	23
<b>4. Ensayos y resultados</b>	<b>25</b>
4.1. Banco de pruebas	25
4.2. Pruebas unitarias	25
4.3. Pruebas de sistema	25
4.4. Comparativa con el estado de arte	25

IV

5. Conclusiones 27

5.1. Resultados obtenidos 27

5.2. Trabajo futuro 27

Bibliografía 29

## 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 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. Ubicación de los protocolos de IoT en la pila TCP/IP . . . . .	7
2.2. Arquitectura del protocolo MQTT . . . . .	8
2.3. Proceso de autenticación de TLS . . . . .	9
2.4. Raspberry Pi . . . . .	10
2.5. Módulos de desarrollo ESP empleados en el proyecto . . . . .	11
2.6. Principales sensores y actuadores empleados en el invernadero . . . . .	12
3.1. Arquitectura del sistema. . . . .	16
3.2. Protocolos de comunicación entre módulos. . . . .	17
3.3. Esquema de conexión de módulos sensores de humedad del suelo . . . . .	18
3.4. Módulos de sensores de humedad del suelo empleados en el proyecto . . . . .	19
3.5. Conexión del módulo de control de riego . . . . .	20
3.6. Módulo de control de riego . . . . .	20
3.7. Conexión del sensor de temperatura y humedad . . . . .	21
3.8. Módulo completo en su caja protectora . . . . .	21
3.9. Conexión del módulo de control de clima . . . . .	22
3.10. Módulo de control del clima . . . . .	22

# 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

# 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

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/>

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 **soluciones** 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/>

## 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]

## 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 son: MQTT [12], HTTP [13], SSL/TLS [14] e IEEE 802.11 [15]. Según el modelo TCP/IP [16] los tres primeros se encuentran en la capa de aplicación mientras que el cuarto se ubica en la capa de enlace. Esto se esquematiza en la figura 2.1.



FIGURA 2.1. Ubicación de los protocolos de IoT en la pila TCP/IP.

#### 2.1.1. 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 [17].

Su funcionamiento se basa en el paradigma de publicación-suscripción [18], el cual consiste en desvincular un cliente que publica un mensaje (publicador) de otros clientes que reciben el mensaje (suscriptores). Otra característica importante a mencionar es que, al ser un protocolo asíncronico, 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 [17].

En la figura 2.2 se observa la arquitectura del protocolo.



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.

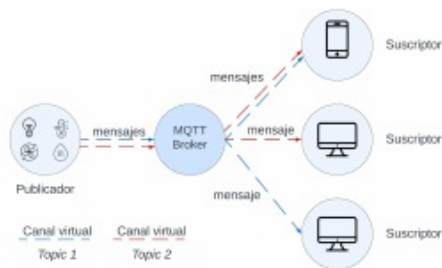


FIGURA 2.2. Arquitectura del protocolo MQTT<sup>2</sup>.

### 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]

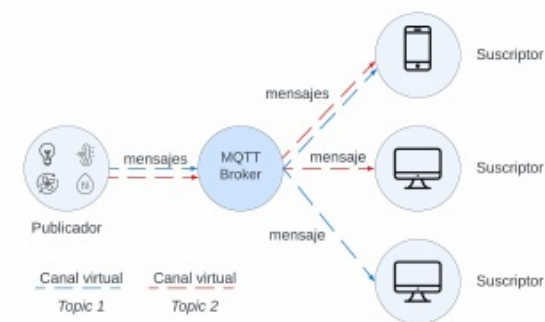


FIGURA 2.2. Arquitectura del protocolo MQTT.

### 2.1.2. Protocolo HTTP

El *Hypertext Transfer Protocol* (HTTP)[19] 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 [20].

En la actualidad, HTTP se utiliza en conjunto con la arquitectura REST (*Representational State Transfer*) [21] 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*) [22]. Dichas funciones a su vez se traducen en los métodos HTTP POST, GET, PUT y DELETE respectivamente [23].

### 2.1.3. Protocolo SSL/TLS

*Secure Socket Layer/Transport Layer Security* (SSL/TLS) 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.

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 [24] 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

### 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.

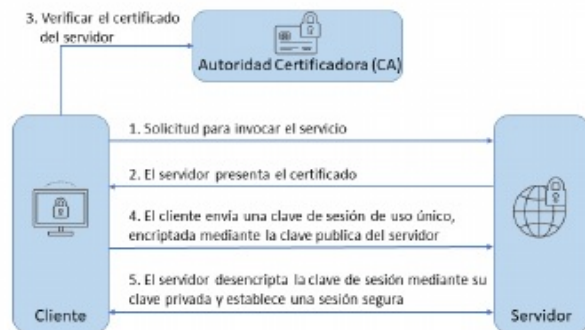


FIGURA 2.3. Proceso de autenticación de TLS<sup>3</sup>.

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

la información [25]. 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 [26] 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 [25].

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>1</sup>.

### 2.1.4. 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 [27]:

- **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.

## 2.2. 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 desarrolladas por la Raspberry Pi Foundation [28]. Una de sus principales características es proveer

<sup>1</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>.

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* y aplicaciones industriales [29].

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[30]
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 y 5,0 GHz IEEE 802.11ac
Bluetooth	5.0 y BLE
Ethernet	Gigabit, con soporte opcional para POE
USB	2 puertos 3.0 y 2 puertos 2.0
GPIO	Conector de 40 pines
HDMI	2 puertos micro-HDMI
Alimentación	5 V USB y GPIO
Temperatura de operación	0 °C a 50 °C



FIGURA 2.4. Raspberry Pi<sup>3</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

<sup>2</sup>Imagen tomada de <https://datasheets.raspberrypi.com/>.

### 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].

en IoT tanto en el ámbito profesional o industrial como en el de los aficionados [31] [32].

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 [31]	ESP8266 [32]
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 y con menores prestaciones que el ESP32. Constituye una opción económica para soluciones en las que los requerimientos de conectividad, cantidad de periféricos y demandas computacionales no son tan exigentes.

En la figura 2.5a se observa un módulo de desarrollo de la familia ESP32, mientras que el ESP8266 puede verse en la figura 2.5b.



(A) Módulo de desarrollo ESP32.



(B) Módulo de desarrollo ESP8266.

FIGURA 2.5. Módulos de desarrollo ESP empleados en el proyecto.



- 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

### 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  $-40^{\circ}\text{C}$  a  $80^{\circ}\text{C}$  y de 0 a 100 % de humedad relativa [33].
- Sensor capacitivo de humedad del suelo: **módulo analógico** compuesto de un material resistente a la corrosión que mide la humedad del suelo **indirectamente** por medio de la **capacitancia** observada. Opera con una **alimentación** de 3,3 a 5,5 VDC y entrega un valor de tensión que varía entre 0 V para un suelo seco a aproximadamente 3,15 V en un suelo completamente húmedo [34].
- 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 [35][36].

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 **forman** parte del sistema [37]. 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.

terceros entre los que encuentran los 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:

- Cloud: similar a la edición profesional, pero alojada en la nube de Things-Board. En este caso el proveedor se encarga del manejo de los componentes de la plataforma.
- Community edition: se encuentra bajo licencia Apache 2.0 [38] y es 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 los módulos ESP32/ESP8266 [39] 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 [40].

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 [41].

### 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 tienen licencias abiertas, 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 [42].

## 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.



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

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 [43].
- 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 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 del software seleccionado.

### 3.1. Arquitectura de la solución

Para la implementación del prototipo propuesto en el trabajo se requirió la construcción de diferentes subsistemas encargados de las múltiples funciones dentro del invernadero inteligente. Cada uno de ellos opera en forma independiente del resto y se comunican con una aplicación central mediante una red inalámbrica. Para garantizar el acceso de los usuarios desde Internet se desarrolló una interfaz de acceso remoto.

#### 3.1.1. Componentes del sistema

En la figura 3.1 se observa el diagrama en bloques de la arquitectura diseñada, que está compuesta por los siguientes elementos:

- Sistema de monitoreo y control de clima, formado por dos módulos:
  - Sensores de temperatura y humedad con sus correspondientes microcontroladores.
  - Unidad de control de temperatura y humedad comprendida por un microcontrolador, relé y ventiladores.

Los sensores miden la temperatura y humedad ambiente en el invernadero y envían estos valores a la aplicación central por medio del microcontrolador. De acuerdo con los datos recibidos, la aplicación determina si es necesario emitir una señal para que la unidad de control encienda los ventiladores.

- Sistema de control de riego, dividido en dos partes:
  - Conjunto de sensores de humedad de suelo con sus respectivos microcontroladores.
  - Unidad de control de riego constituida por un microcontrolador, relés, bomba de agua y válvulas.

Los sensores envían las mediciones a la aplicación central que se encarga de procesarlas. En caso de ser necesario, se disparan las señales de encendido a través de la unidad de control. Primero se activa la válvula que corresponda

y luego se enciende la bomba de agua para comenzar el riego. El orden de estas actividades es importante para evitar daños en la bomba o las cañerías.

- Aplicación central. Constituye el cerebro del invernadero y es la encargada de almacenar los parámetros de configuración de los diversos sensores y actuadores, procesar los mensajes recibidos, disparar acciones y alertas y visualizar el estado general.
- Sistema de acceso remoto. Permite a los usuarios obtener reportes del sistema en forma segura desde Internet.

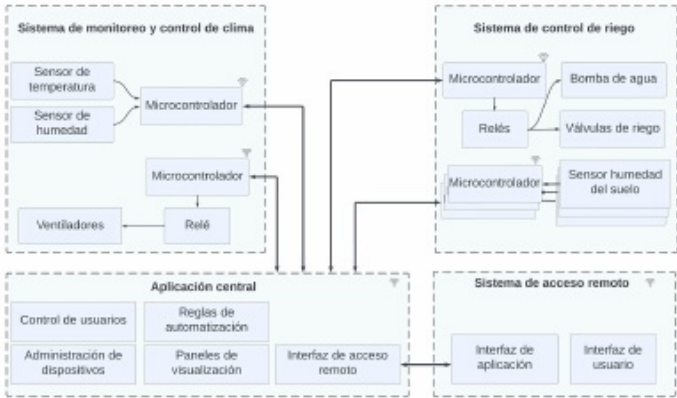


FIGURA 3.1. Arquitectura del sistema.

3.1.2. Protocolos de comunicación

En esta sección se describe cómo se comunican los sistemas con la aplicación central y los protocolos usados en cada caso. En la figura 3.2 se aprecia un esquema simplificado que ilustra dichas interacciones.

Si bien los módulos de hardware y el software soportan una gran variedad de protocolos, se implementó MQTT en la mayoría de los casos conforme a los requerimientos. En algunas situaciones donde esto no fue técnicamente posible, se utilizó HTTP. Adicionalmente, para garantizar la seguridad de las comunicaciones, se incorporó un certificado autofirmado TLS en el servidor.

## 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

### 3.2. Detalle de los módulos de hardware

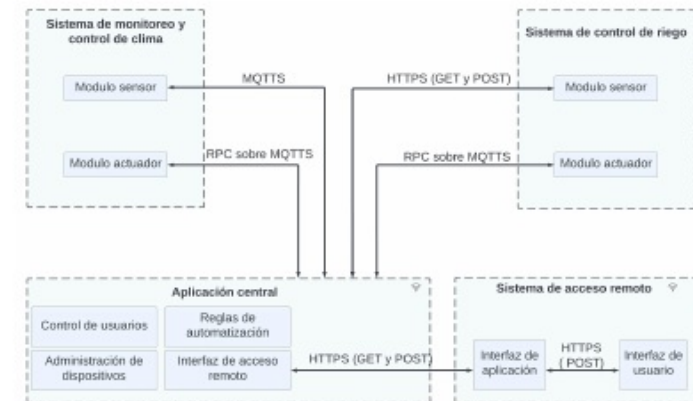


FIGURA 3.2. Protocolos de comunicación entre módulos.

Las interacciones principales entre los bloques componentes son:

- Sistema de monitoreo y control de clima: las comunicaciones se realizan exclusivamente con aplicación central. El módulo sensor envía las mediciones realizadas por medio de MQTT y en caso de requerir una acción, la aplicación central comanda el encendido de los ventiladores por medio de un mensaje enviado por RPC [13] sobre MQTT.
- Sistema de control de riego: las comunicaciones se realizan con la aplicación central de manera bidireccional. El módulo sensor efectúa dos conexiones, una para el envío de las mediciones y otra para recibir valores de atributos tales como la duración del tiempo de riego. Debido a limitaciones en la configuración de la persistencia de los mensajes en las colas de MQTT, se optó por utilizar llamadas HTTP (GET y POST) para realizarlas. Al igual que en el control de clima, la aplicación central inicia el riego por medio de mensajes RCP sobre MQTT hacia el controlador de la bomba y de las válvulas.
- Sistema de acceso remoto: la interfaz se comunica con la aplicación por medio de pedidos HTTP GET y POST para consultar el reporte de estado de los diferentes componentes. A continuación este se envía hacia la interfaz de usuario por medio de una solicitud HTTP POST.

### 3.2. Detalle de los módulos de hardware

En esta sección se describen en detalle los esquemas de conexión de los distintos módulos y las consideraciones de diseño y construcción empleadas.



3.2.1. Módulos sensores de humedad del suelo

En el proyecto se desarrollaron dos configuraciones diferentes de módulos para medir la humedad del suelo en macetas de diverso tipo y tamaño. Ambas opciones utilizan el microcontrolador ESP8266 pero incorporan distinta cantidad de sensores. En la figura 3.3a se muestra el esquema de conexión para la versión simple (con un único sensor) y en la figura 3.3b se ilustra la configuración doble.

Si bien en el prototipo los sensores se conectaron a una fuente de alimentación, la configuración y conexión del sistema está optimizada para el uso de baterías. Esto se debe a que las sondas pueden estar desplegadas en múltiples ubicaciones dentro del invernadero y no siempre es posible conectarlas a la red eléctrica.

El ahorro de energía necesario para permitir el uso de baterías se logra a través de ciclos de apagado en los periodos donde no se realizan lecturas. A este mecanismo se lo que conoce como *deep sleep* y una vez que el microcontrolador se encuentra en este estado se requiere un pulso eléctrico en el pin de *reset* para que retorne al modo activo. Para habilitar la reactivación se interconectaron los pines D0 (GPIO16) y RST del chip ESP8266. Se logra así una reducción del consumo de energía a valores muy pequeños que rondan los 0,3 mA durante el período de inactividad.

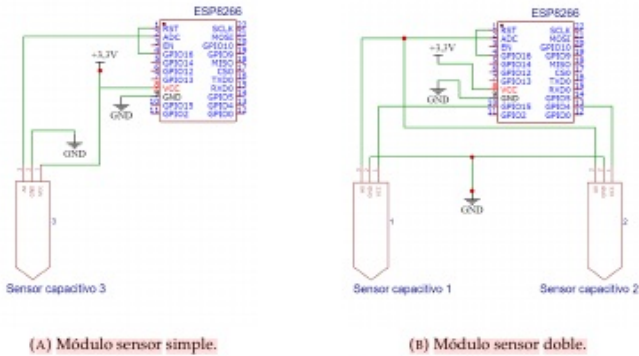


FIGURA 3.3. Esquema de conexión de módulos sensores de humedad del suelo.

La integración de los componentes de los sensores se realizó en forma manual por medio de placas de circuitos impresos (PCB) experimentales. En las figuras 3.4a y 3.4b se muestran los componentes empleados y un módulo ensamblado.

Dado que los sensores están expuestos a salpicaduras, se los recubrió con tubos adhesivos termocontraíbles que, al aplicarles calor, generan una protección a prueba de agua. Para el resguardo del chip ESP8266 se utilizó una caja de polipropileno transparente sellada. En las figuras 3.4c y 3.4d se muestran los componentes y sus protecciones.

## 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.



(A) Módulo con un sensor de humedad del suelo.



(B) Módulo con dos sensores de humedad del suelo.



(C) Detalle de protección de circuitos en los sensores.



(D) Módulo en su caja protectora.

FIGURA 3.4. Módulo de sensores de humedad del suelo empleados en el proyecto.

### 3.2.2. Módulo controlador del riego

Se compone de un microcontrolador ESP32, una placa de interfaz de relé de cuatro canales, una pantalla LCD/OLED SSH1106 y un regulador de voltaje DC-DC *step down* LM2596. El esquema de conexiones entre estos componentes se detalla en la figura 3.5.

El módulo se alimenta con una fuente de 12 VDC y para energizar a los circuitos electrónicos el regulador LM2596 reduce la tensión a 5 VDC.

Para la construcción del prototipo se realizó la integración del microcontrolador con la pantalla LCD mediante una placa PCB experimental. Tanto para el módulo regulador de tensión como para el conjunto de relés se emplearon circuitos preensamblados. En la figura 3.6 se muestran los componentes, su conexionado y la versión final de la unidad dentro de una caja protectora.

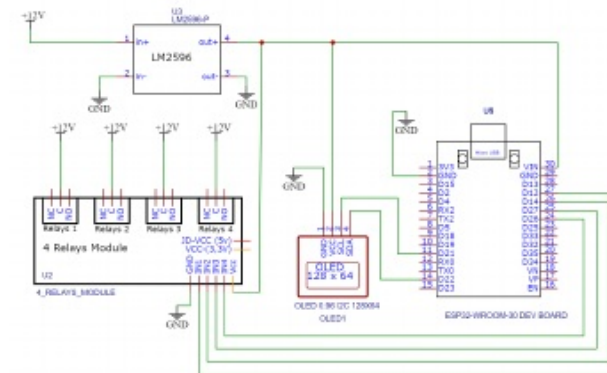
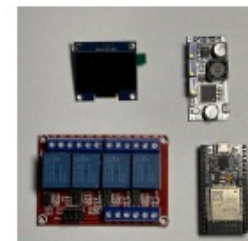
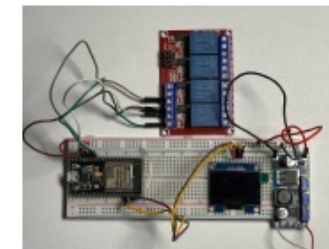


FIGURA 3.5. Conexión del módulo de control de riego.



(A) Detalle de los componentes.



(B) Conexionado.



(C) Módulo finalizado en su caja protectora.

FIGURA 3.6. Módulo de control de riego.

### 3.2.3. Módulo sensor de temperatura y humedad

Está compuesto por un microcontrolador ESP8266, un sensor DHT22 y una pantalla LCD/OLED SSH1106 para visualizar los valores de temperatura y humedad *in situ* en tiempo real. El esquema de conexión de los componentes se puede ver en la figura 3.7.

A diferencia de las sondas de humedad del suelo, el sensor de temperatura y humedad está pensado para instalarse en una ubicación fija con acceso a la red



## 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. 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.
- [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.
- [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.
- [13] Shanna L. «Comparative Analysis of Infrastructure and Ad-Hoc Wireless Networks». En: *ITM Web of Conferences* 25, 01009 (2019).
- [14] [www.survivingwithandroid.com](http://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: <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).

eléctrica. Por este motivo no se consideró configurarlo para soportar *deep sleep*.

La construcción del módulo se realizó sobre placa PCB experimental en forma similar a los demás sistemas. Para proteger los componentes se utilizó una caja de polipropileno transparente. El sensor DHT22 quedó expuesto para medir las condiciones ambientales como muestra la figura 3.8.

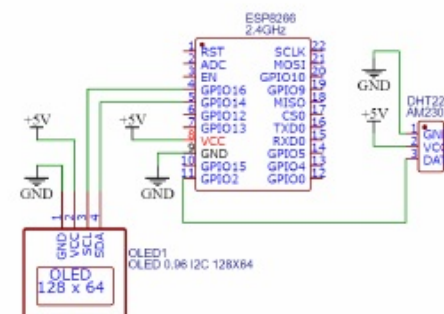


FIGURA 3.7. Conexión del sensor de temperatura y humedad.



FIGURA 3.8. Módulo completo en su caja protectora.

### 3.2.4. Módulo controlador de clima

Es el responsable de accionar los ventiladores en el invernadero. Para el diseño se utilizó un chip ESP8266 conectado a un relé de una vía como se muestra en la figura 3.9.

Dado que la salida del microcontrolador es de 3,3 V para el prototipo se seleccionó un relé que pueda ser accionado con ese valor de tensión, para evitar el uso de componentes adicionales tales como convertidores de tensión DC-DC *step up*.



- [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. 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>. (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).

En la figura 3.10 se ilustra el proceso de construcción del módulo.

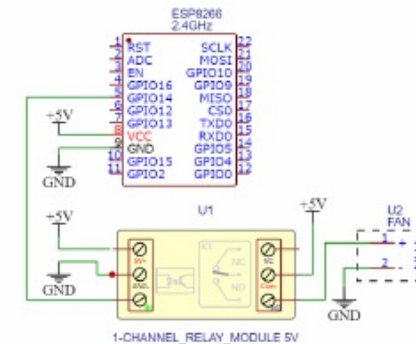


FIGURA 3.9. Conexión del módulo de control de clima.



FIGURA 3.10. Módulo de control del clima.