



CARRERA DE ESPECIALIZACIÓN EN INTERNET DE LAS COSAS

MEMORIA DEL TRABAJO FINAL

Sistema de riego y control de huertas

Autor:

Lic. Gustavo Hernan Siciliano

Director:

Mg. Ing. Osvaldo Ivani (FIUBA)

Jurados:

Nombre del jurado 1 (pertenencia)

Nombre del jurado 2 (pertenencia)

Nombre del jurado 3 (pertenencia)

*Este trabajo fue realizado en la ciudad de Avellaneda,
entre mayo de 2023 y agosto de 2023.*

Resumen

La presente memoria describe el diseño y desarrollo de un prototipo tecnológico que realice acciones de monitoreo y cuidado sobre un conjunto de cultivos en una huerta. La idea nace a partir de un proyecto de investigación originado en la UNLa (Universidad Nacional de Lanús), que apunta a formar conocimiento sobre agricultura 4.0 y a trabajar en torno al concepto de empleo verde. El producto será desplegado y utilizado en el predio de la universidad, con fines tanto académicos como socio culturales.

Para el desarrollo de este trabajo fueron importantes los conocimientos sobre los módulos ESP y sus posibilidades de integración con diferentes sensores.

Además, se destacan los aprendizajes adquiridos en desarrollo web y bases de datos a gran escala. Finalmente, fue fundamental la comprensión de los distintos protocolos de comunicación y la seguridad con la que deben contar para operar correctamente.

Agradecimientos

A mi mamá Amada que siempre va a estar en mis recuerdos.

A mi papá Gustavo y mi hermana Lucia que siempre están cuando los necesito.

A mi novia Carolina que, además de incentivarme para volver a estudiar, me hace muy feliz y me llena de amor y alegría.

A mi familia y amigos que tanto quiero y aprecio.

A Damian Reboredo por su soporte y buena predisposición para charlar y debatir sobre los detalles de la capa física del proyecto.

Índice general

Resumen	I
1. Introducción general	1
1.1. Marco de trabajo en la universidad	1
1.2. Motivación	2
1.2.1. Prácticas profesionales para estudiantes	2
1.2.2. Nuevos conocimientos para la carrera	2
1.2.3. Oportunidades de proyectos de investigación	2
1.2.4. Oportunidades de congresos y conferencias	2
1.2.5. Impacto positivo en la comunidad de la UNLa	2
1.3. Estado del arte	3
1.3.1. Sistema de riego automatizado y monitoreo de variables ambientales	3
1.3.2. Sistema de riego automatizado basado en IoT	3
1.3.3. Sistema automatizado para riego en huertos urbanos y plantas	4
1.4. Objetivos y alcance	4
2. Introducción específica	7
2.1. Componentes de hardware	7
2.1.1. NodeMCU ESP32	7
2.1.2. Sensor de temperatura y humedad relativa DHT22	8
2.1.3. Sensor de humedad en suelo capacitivo analógico V1.2	9
2.1.4. ADC ADS1115	9
2.2. Herramientas de software	9
2.2.1. Backend con Java y SpringBoot	9
2.2.2. Frontend con JavaScript y React	9
3. Diseño e implementación	11
3.1. Arquitectura del sistema	11
3.2. Diseño y arquitectura de la capa física	12
3.3. Arquitectura integración del Backend con el Broker MQTT	13
3.4. Desarrollo base del Backend y Frontend en UNLa	13
3.5. Despliegue del sistema en UNLa	14
4. Ensayos y resultados	15
4.0.1. Prueba de concepto del dispositivo	15
4.1. Banco de pruebas	15
4.2. Pruebas de componentes	15
4.3. Pruebas del backend	15
4.4. Pruebas del frontend	15
4.5. Valor agregado del proyecto	15

5. Conclusiones	17
5.1. Resultados obtenidos	17
5.2. Trabajo futuro	17

Índice de figuras

2.1. Diagrama de bloques ESP32.	7
2.2. Diagrama pines ESP32.	8
3.1. Arquitectura del sistema IoT.	11
3.2. Diagrama del dispositivo.	12
3.3. Arquitectura backend e integración con Mosquitto del dispositivo.	13
3.4. Mapa de la huerta en UNLa.	14

Índice de tablas

1.1. Nuevas funcionalidades sobre el proyecto de Cadavid y Garcia . .	3
1.2. Nuevas funcionalidades sobre el proyecto de Díaz, Sierra y David .	3
1.3. Nuevas funcionalidades sobre el proyecto de Rocío González . . .	4

Capítulo 1

Introducción general

Este capítulo explica el marco contextual del problema y el rol de la UNLa en el proyecto. Además, detalla la motivación del equipo de profesores y estudiantes en el trabajo. Luego, presenta una descripción del estado del arte en relación a proyectos similares, que sirvieron como fuente de información. Finalmente, describe los objetivos y el alcance establecido para el presente documento.

1.1. Marco de trabajo en la universidad

Este trabajo se originó en el contexto del laboratorio de investigación y desarrollo de la carrera de Licenciatura en Sistemas de la UNLa. En él se llevan adelante propuestas de diversos tipos, pero todas con un factor en común, generar conocimiento y lograr impacto positivo en la comunidad. Por este motivo se desarrollan sistemas orientados a pequeñas y medianas organizaciones. Siempre se busca brindar experiencia profesional a estudiantes, generar conocimientos nuevos en la carrera y/o trabajar en proyectos de bajo costo (o gratuitos) para las organizaciones que lo necesiten.

En el laboratorio se llevan adelante diferentes tareas para definir en qué tipo proyectos se puede incursionar. En ese sentido se iniciaron investigaciones orientadas a temáticas sobre soberanía alimentaria [1] y empleo verde [2]. El foco principal se puso en investigar las nuevas tecnologías y propuestas en torno a la agricultura 4.0 [3]. Finalmente se inició una serie de cursos en la Universidad de Chile en donde se participó en talleres sobre estas tópicos. Siendo los talleres del proyecto Piwkeyewün [4] los que inspiraron el inicio del proyecto en la UNLa.

Gracias a toda esta investigación y preparación quedó expuesta la necesidad de iniciar un proyecto sobre IoT (Internet de las cosas). Que tuviera una orientación a la agricultura de uso común, con un equipo de trabajo formado por estudiantes y profesores. De esta forma, el laboratorio de la carrera se propuso el objetivo de desarrollar un prototipo de genere acciones de cuidado y monitoreo sobre diversos cultivos en una huerta. Con vistas de tenerlo productivo en el mismo predio de la UNLa o en la sede de Abremate. En ambos casos la idea es tener el sistema funcionando para generar muestras a estudiantes y visitantes.

1.2. Motivación

A continuación se detalla el listado de motivaciones por parte del equipo de estudiantes y profesores de la UNLa. Las cuales no están relacionadas con el alcance del producto en sí, sino que con necesidades y verticales internas de trabajo de la carrera.

1.2.1. Prácticas profesionales para estudiantes

Se buscó que en el proyecto puedan colaborar estudiantes para completar sus trabajos finales de la carrera. En concreto participaron:

- Damian Reboredo: investigación y prototipo de la capa física.
- Luciano Otegui: investigación y desarrollo base del backend de la capa lógica.
- Guido Contento: investigación y desarrollo base del frontend de la capa lógica.

1.2.2. Nuevos conocimientos para la carrera

Con la documentación del proyecto se quiere capitalizar el conocimiento adquirido para sumarlo en la carrera. Esto podría ser a través de cursos o talleres, materias nuevas, agregado de temas en materias existentes y/o documentación bibliográfica general. Al finalizar el trabajo se contemplará cuál o cuáles de las opciones anteriores se podrían sumar.

1.2.3. Oportunidades de proyectos de investigación

De la misma manera que se apunta a estudiantes, también se espera la participación de profesores de la carrera en proyectos de investigación. Esto se relaciona con fortalecer y agregar nuevos conocimientos en el cuerpo docente. En este caso el autor de este trabajo es el profesor a cargo del proyecto. De esta forma el rol del docente se centra en guiar al equipo de desarrollo, buscar perfeccionamiento profesional y capitalizar las experiencias adquiridas en el marco de la carrera.

1.2.4. Oportunidades de congresos y conferencias

Otro punto importante es la posibilidad de participar en congresos y/o conferencias de informática. Una vez finalizado el proyecto se comenzará a investigar este tipo de oportunidades para enviar propuestas de charlas. Se busca representar a la UNLa y participar de comunidades tecnológicas.

1.2.5. Impacto positivo en la comunidad de la UNLa

Finalmente no se puede perder de vista el motivo inicial del proyecto. La iniciativa del producto y la idea de tenerlo funcionando en la universidad viene asociada con el hecho de tener huertas en la UNLa. Se quiere que con estas, no solo se pueda usar el sistema, sino que también se pueda proveer de alimentos a quienes lo necesiten. Para esto se va a estudiar el caso con las áreas que correspondan en la universidad. Por otro lado, también se espera que el producto pueda ser utilizado en organizaciones con lazos fuertes con la UNLa, tanto para temas de estudio, como de alimentación y nutrición.

1.3. Estado del arte

En esta sección se detallan proyectos de investigación y desarrollo similares al del presente trabajo. Cada uno fue analizado por el equipo del laboratorio y se hizo foco en buscar puntos de valor agregado.

1.3.1. Sistema de riego automatizado y monitoreo de variables ambientales

Este proyecto consistió en el diseño e implementación de un sistema IoT en los cultivos urbanos de la fundación mujeres empresarias Maria Poussepin [5]. Fue desarrollado Valeria Cadavid y Marco Garcia en la Universidad Católica de Colombia. La solución consiste en un módulo formado por una placa de arduino, sensores de humedad, una placa LCD y un sistema de riego accionado por una electroválvula. Además, se utilizó una placa ESP8266 [6] para enviar los datos vía Wi-Fi a un servidor y visualizarlos en un gráfico. Sus bases se utilizaron inicialmente como información para buscar valor agregado. La propuesta de Cadavid y Garcia es muy similar a la del presente trabajo pero cuenta con diferencias, las cuales se presentan en la tabla número 1.1.

TABLA 1.1. Nuevas funcionalidades.

Nueva funcionalidad	Mejora
Uso únicamente de placa ESP32	Reducción de costos
Comandos a la placa desde un servidor	Customización para el usuario
Diseño modular de los sensores	Permite agregar sensores diferentes

1.3.2. Sistema de riego automatizado basado en IoT

Este trabajo se centró en utilizar variables ambientales para cultivos de berenjena en la finca la esperanza del municipio de Chinú-Córdoba [7]. Fue llevado adelante por Ramírez Díaz, Eliécer Jesús Vergara Sierra, Jesús David en la Universidad de Córdoba en Colombia. Su diseño consiste en una placa Arduino que, a través de sensores de humedad, recolecta información de una huerta. Estos valores se envían a un servidor web y cuando están por debajo de un valor fijo se acciona una electrobomba para regar los cultivos. Al igual que el proyecto anterior, este se utilizó para la generación de ideas y la búsquedas de valor agregado. En la tabla 1.2 se muestran las diferencias con el presente trabajo.

TABLA 1.2. Nuevas funcionalidades.

Nueva funcionalidad	Mejora
Uso de Placa ESP32	Reducción de costos
Comunicación con WiFi	Eliminación del cableado
Rangos de aceptación dinámicos	Customización para el usuario

1.3.3. Sistema automatizado para riego en huertos urbanos y plantas

En este proyecto se propuso diseñar e implementar un sistema de riego automatizado de bajo costo. La autora menciona que su foco se puso en lograr una gestión de manera automática del suministro de agua, para de esta forma, lograr un riego óptimo sobre los cultivos [8]. Fue desarrollado por Rocío González en la Universidad Técnica Federico Santa María en Chile. Este proyecto también utilizó una placa Arduino, pero, a diferencia de los anteriores, fue conectada con varios tipos de sensores. Entre ellos sensores de humedad del sustrato, humedad ambiente, temperatura ambiente y luminosidad. Los datos visualizan en una pantalla LCD. Además, a través de una placa ESP8266 con módulo Wi-Fi se enviaban los datos a un servidor. Cuando los valores se colocan por debajo un número seleccionado, se activa una electrobomba para asegurar el riego. Como los proyectos anteriores, este sirvió de ayuda para buscar mejoras y valor agregado. En la tabla 1.3 se presentan las diferencias con el actual trabajo.

TABLA 1.3. Nuevas funcionalidades.

Nueva funcionalidad	Mejora
Uso de Placa ESP32	Reducción de costos
Comandos a la placa desde un servidor	Customización para el usuario
Conexión editable con el servidor	Permite cambiar de servidor fácilmente

1.4. Objetivos y alcance

El foco del desarrollo de este documento se centra en tomar las pruebas de concepto desarrolladas en el laboratorio de software para finalizarlas e integrarlas.

A continuación se listan los requerimientos y alcances de cada capa.

1. Capa física

- Desarrollar el prototipo de la prueba de concepto.
- Agregar la autenticación por parte del dispositivo.
- Agregar la toma de datos del sistema por parámetros (Wi-Fi, rutas del servidor y claves)
- Desarrollar la integración con el comando de apertura de la válvula.

2. Capa de comunicación

- Integrar el proyecto con un broker IoT.
- Agregar una capa de seguridad en la transferencia de los paquetes.
- Tener una estrategia de backup de a información.

3. Capa Lógica

- Desarrollar las secciones de gestión de huertas y dispositivos.
- Desarrollar al menos un gráfico de información de los datos obtenidos.

- c) Evaluar y documentar magnitud de los datos y posible cambio de motor de base de datos.

Por otro lado, se detallan los requerimientos generales del sistema.

1. Requerimientos del dispositivo

- a) Debe tener un código interno para ser identificado unívocamente en el software de control.
- b) Debe contar con una placa ESP32 más dos sensores de humedad (uno de ambiente y otro de suelo).
- c) La placa ESP32 debe usar los sensores para medir los porcentajes correspondientes de las plantas de su sector.
- d) Tiene que poder activar la apertura o cierre de una válvula de agua usando un comando interno.

2. Requerimientos de integración del backend del software

- a) Debe contar con un endpoint rest API para consultar las mediciones de un dispositivo.
- b) Debe contar con un endpoint rest API para solicitar a un dispositivo la apertura de la válvula de agua durante 30 segundos.

3. Requerimientos del sistema para los usuarios

- a) Debe permitir a un usuario loguearse al sistema usando su mail y contraseña.
- b) Debe permitir a un usuario desloguearse del sistema.
- c) Debe permitir a un usuario recuperar y cambiar su contraseña.
- d) Debe contemplar permisos para cada rol del sistema:
 - 1) Administrador: acceso todas las funcionalidades.
 - 2) Responsable: acceso a las funcionalidades de administración de sus huertas.
 - 3) Visitante: acceso a las funcionalidades de visualización de sus huertas asociadas.
- e) Debe tener una sección para ver, modificar y eliminar huertas del sistema.
- f) Debe tener una sección para ver, modificar y eliminar usuarios del sistema.
- g) Debe tener una sección para administrar huertas vinculando el código de un dispositivo por sector.
- h) Debe tener una sección para administrar los porcentajes de aceptación de humedad por sector.
- i) Debe tener una sección para visualizar las mediciones de un dispositivo por sector.

Capítulo 2

Introducción específica

En este capítulo se detallan los componentes y tecnologías que se seleccionaron para utilizar en el trabajo.

2.1. Componentes de hardware

Para la realización de la capa física hay cuatro componentes indispensables. Estos se especifican en las próximas subsecciones:

2.1.1. NodeMCU ESP32

Este es un microcontrolador desarrollado Espressif [9] que se destaca por sus módulos Wi-Fi, Bluetooth y Bluetooth LE. Es importante señalar que en el mundo de ESP existen dos familias, ESP8266 y ESP32. Entre ambas se opta por ESP32 porque es la nueva generación fabricada por Espressif. Además, cuenta con más documentación [10] y posee mejoras en el hardware. Las cuales están asociadas a una mayor capacidad de procesamiento, de integración y otra versión de placa Wi-Fi con mayores prestaciones. En la hoja de datos de la documentación oficial [11] es posible observar su descripción de hardware. En la figura 2.1, extraída de dicha documentación, se presenta el diagrama de bloques del ESP32.

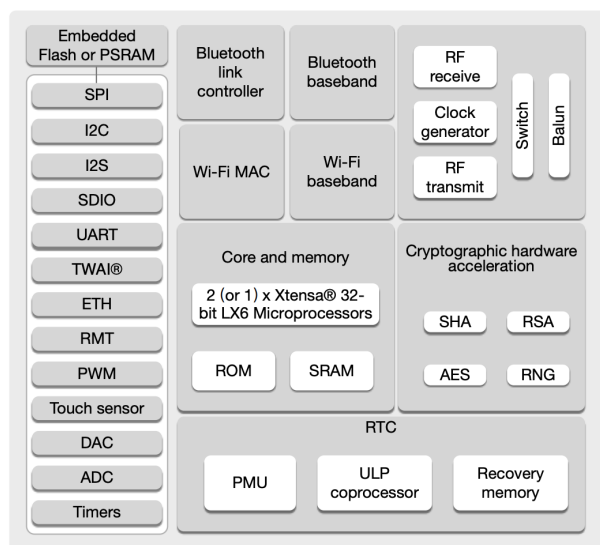


FIGURA 2.1. Diagrama de bloques ESP32.

A su vez, en la familia de ESP32 existen diferentes tipos de módulos [12]. Cada uno presenta pequeñas variaciones de hardware, comandos de integración de firmware y cantidad de pines. En particular, para este trabajo se utiliza el módulo de WROOM. Más específicamente, uno de los tableros de ESP32-devkit [13]. Además, existen diferentes versiones de tableros de desarrollo de WROOM para ESP32-DEVKIT. Por ello, para realizar las conexiones con los sensores, es fundamental tener conocimiento sobre la usabilidad de cada uno de los pines de la placa. En la figura 2.2 [14] es posible observar un diagrama de pines del módulo.

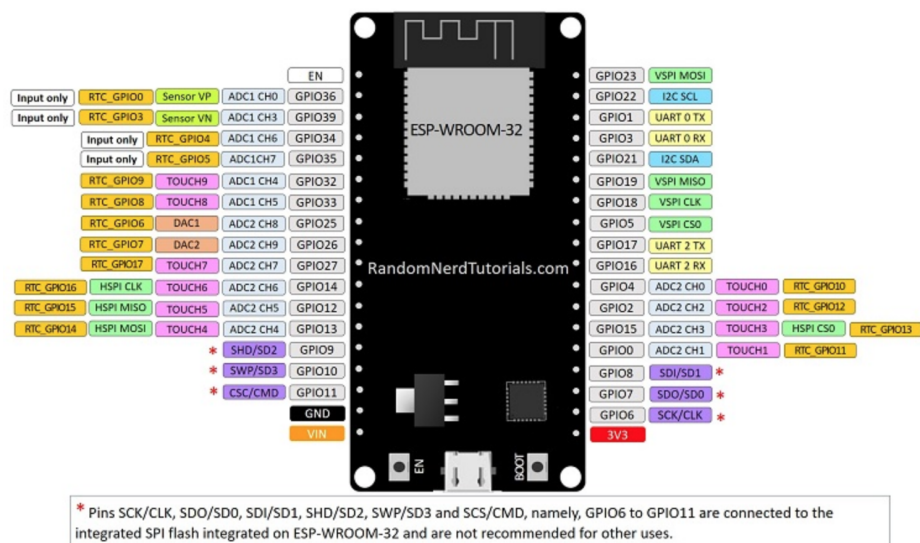


FIGURA 2.2. Diagrama pines ESP32.

La selección de este módulo se basó no solo en sus excelentes características de hardware, sino también en las ventajas que ofrece en términos de software. Espressif, el fabricante de la placa, proporciona una amplia variedad de ejemplos de código C en sus repositorios de GitHub [15].

2.1.2. Sensor de temperatura y humedad relativa DHT22

Con este sensor se podrá realizar la obtención de valores de temperatura y humedad ambiente. Según se detalla en su hoja de datos [16] ofrece una señal digital confiable y estable. Este sensor cuenta con 4 pines, de los cuales se utilizarán los siguientes 3 (numeración de izquierda a derecha con el sensor de frente):

- Pin 1: VDD (alimentación). Deberá recibir de 3,3 a 6 V. Se recomienda colocar un capacitor de 100 nF entre este pin y el pin GND para su protección.
- Pin 2: DATA. Una vez establecida la conexión, este pin enviará los datos de temperatura y humedad relativa al módulo ESP.
- Pin 4: GND.

2.1.3. Sensor de humedad en suelo capacitivo analógico V1.2

Es un sensor que se utiliza para medir la humedad del sustrato. Como se menciona en su hoja de datos [17] este componente es una excelente opción ya que, al estar fabricado con un material resistente a la corrosión, posee una excelente vida útil. Este sensor cuenta con un conector de pines con especificaciones muy claras: VCC, GND y AOUT (datos analógicos). Se recomienda utilizar con una potencia de 3,3 a 5 V.

2.1.4. ADC ADS1115

Debido a que el sensor de humedad de suelo tiene mayores pruebas y confiabilidad usando Arduino (incluso en su hoja de datos) se va a agregar un convertidor analógico digital con el ESP32. Se utilizó como fuente de información su hoja de datos [18] y documentación adicional [19]. De esta forma se transformará la señal analógica, del sensor capacitivo, a digital para enviarla a la placa ESP32.

2.2. Herramientas de software

Para el desarrollo de software se trabajó con diferentes lenguajes de programación para el backend y el frontend. A su vez, en cada capa se definió el uso de frameworks que provean de funcionalidad base para agilizar el desarrollo. A continuación se detallan las siguientes tecnologías.

2.2.1. Backend con Java y SpringBoot

Se optó por trabajar con Java [20] como lenguaje de programación, utilizando las bibliotecas y funcionalidades que ofrece el framework de Spring Boot [21]. Con estas tecnologías se realiza:

- Conexión con la base de datos utilizando MySQL [22].
- Integración con la capa física.
- Desarrollo de los endpoints para integración con el frontend.
- Capa de validación y seguridad (para usuarios y dispositivos).

2.2.2. Frontend con JavaScript y React

Con respecto al frontend se definió utilizar JavaScript [23] como lenguaje de programación y el conjunto de bibliotecas de React [24] para el desarrollo de las vistas del sistema. Con estas herramientas se realiza:

- Bloqueo de URL's no permitidas sin autenticación.
- Definición de componentes visuales del sistema para su re utilización.
- Maquetado con herramientas pre definidas de estilos CSS.
- Formularios reactivos.

Capítulo 3

Diseño e implementación

En este capítulo se explica la solución del sistema, tanto desde un punto vista general, como particular, detallando cada una de sus partes.

3.1. Arquitectura del sistema

La arquitectura del trabajo contempla las tres capas de toda aplicación IoT. En la capa física se encuentra la placa ESP32 que centraliza las mediciones de los sensores de humedad ambiente y del sustrato. Dichos datos se envían por Wi-Fi a la capa de transferencia a través un Broker Cloud MQTT. Para establecer la conexión el ESP32 usa los datos de usuario y clave correspondientes del Broker. Luego, al momento de publicar, cada placa enviará sus mensajes a un tópico llamado `/metrics/«nodoId»`, donde `nodoId` es el identificador de la placa en el sistema. De esta manera se realizará la asociación de qué métricas se corresponden con una placa específica. Por otra parte, el Broker disponibilizará la información para el Backend del sistema, que actúa como subscriptor de todos los tópicos mediante `/metrics/#`. De esta forma, la capa de presentación obtiene las métricas de cada placa a través de la comunicación con el Backend. Finalmente, el Frontend del sistema permite que los usuarios finales puedan utilizar los datos de sus cultivos. Cada usuario tiene la posibilidad de gestionar su huerta y seleccionar rangos óptimos de aceptación de los porcentajes de humedad. En el caso de tener valores no deseados, el Backend publicará un mensaje en el Broker para dar aviso, a una placa en particular, que realice la apertura de una válvula de agua. Las placas van a recibir los comandos a ejecutar debido a que cada una a esta subscripta al tópico `/actions/«suNodoId»`. En la figura 3.1 se visualiza el diagrama de la arquitectura general del sistema.

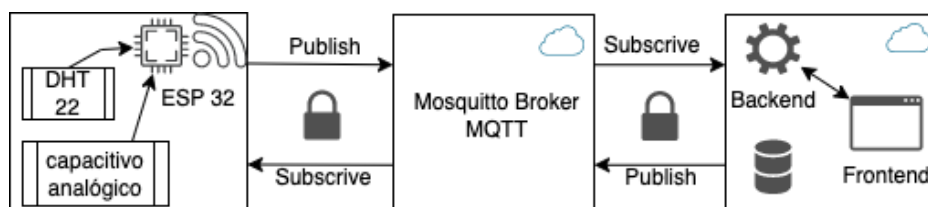


FIGURA 3.1. Arquitectura del sistema IoT.

3.2. Diseño y arquitectura de la capa física

Como se menciona en la sección 2.1 la solución del sistema embebido tiene tres partes fundamentales:

- Módulo ESP32.
- Sensor de humedad ambiente DHT22.
- Sensor del sustrato capacitivo analógico V1.2.

A pesar de esto, el diseño del dispositivo, como un todo, es más complejo, ya que se necesitan más componentes para poder operar en el mundo real. A continuación se analiza cada componente y sus conexiones entre sí. Esto se puede apreciar en la figura 3.2, diseñada por Damian Reboredo en UNLa [25].

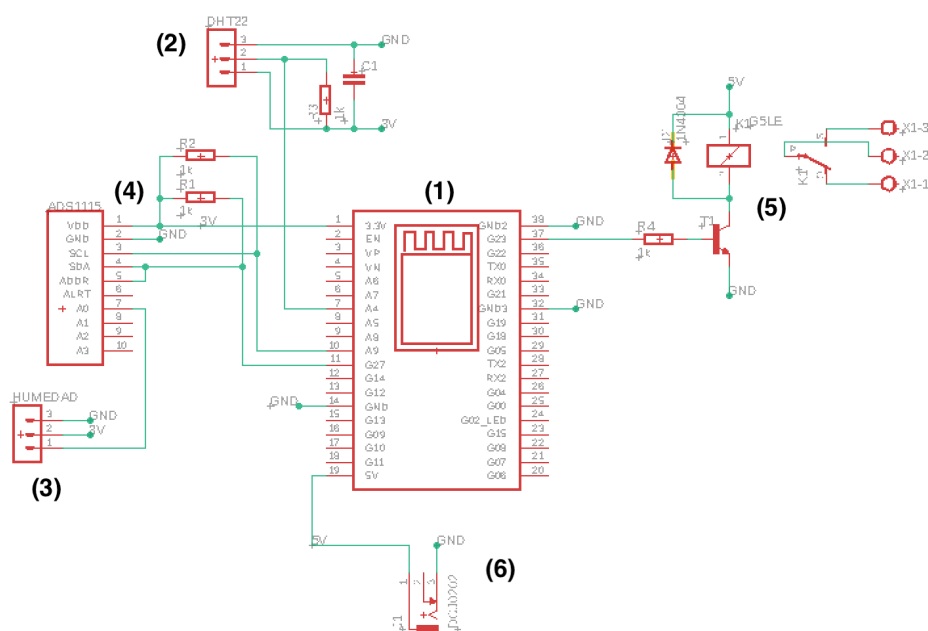


FIGURA 3.2. Diagrama del dispositivo.

Como se observa en la imagen, el diseño cuenta con seis secciones a destacar.

1. ESP32. Siendo esta la placa que centraliza la información de los sensores, distribuye la energía los componentes. Además, es el componente encargado de enviar los datos vía Wi-Fi. Para lograr esto, dentro contiene la información necesaria para realizar la comunicación con el Broker MQTT. Además, se encarga de recibir el comando de apertura de la válvula de agua y accionar dicha acción.

2. DHT22. Este sensor realiza la medición la temperatura y humedad ambiente. Es importante resaltar que podría no existir en todos los dispositivos, ya que si en una misma huerta se instalan entre dos y cuatro placas ESP32, solo una llevará el rol de medición de humedad ambiente. Esto último se debe tener en cuenta a la hora de realizar el plano de la huerta. Por otro lado, la programación del ESP32 tiene que estar preparada para contar o no con este sensor.

3. Sensor de humedad del sustrato. A diferencia del sensor DHT22, este está presente en cada una de las ESP32 de la huerta. En todos los casos se usa este componente para validar la humedad relativa del suelo en un conjunto de cultivos.

4. ADS1115. Se trata de un conversor analógico-digital utilizado para cambiar la señal analógica del sensor de humedad del sustrato. De esta forma se genera la interacción de información con la placa ESP32.
5. Válvula de agua. En este caso, como se puede ver en el diagrama de la figura 3.2, se armó el prototipo para accionar un relé. De esta forma se simula la acción de apertura de una manguera. Para el trabajo final se cambia esto último para trabajar con la válvula.
6. Fuente de alimentación. Para este punto, como en el anterior, se pensó inicialmente una fuente directa. Para el caso del producto final de este trabajo se cambia este enfoque hacia el uso de una batería. De esta manera se quiere prescindir de la existencia de un cableado de corriente eléctrica.

3.3. Arquitectura integración del Backend con el Broker MQTT

Para simplificar la capa de integración con el Broker MQTT se definió crear un servicio con NodeJS que funcione como un *Cron Job*. De esta forma este desarrollo tiene la responsabilidad de obtener la información publicada en el Broker, procesarla y persistirla en la base de datos. Luego, la aplicación en Java Spring queda con la responsabilidad de ejecutar la lógica de negocio del sistema y exponer la API Rest para consumo del Frontend. En la figura 3.3 se puede observar el diagrama de la arquitectura descrita.

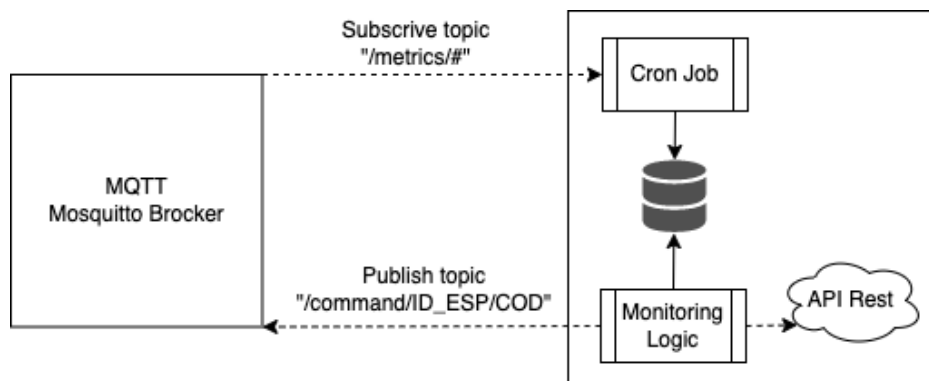


FIGURA 3.3. Arquitectura backend e integración con Mosquitto del dispositivo.

3.4. Desarrollo base del Backend y Frontend en UNLa

Por parte de la capa lógica, Luciano Otegui y Guido Contenido hicieron foco en generar la base del Backend y del Frontend para el trabajo. Se realizó el armado de casos de uso y diagrama de clases. Por otra parte, crearon la estructura mínima del sistema para una correcta interacción de usuarios finales. Finalmente, se definieron las vistas y estilos generales del sitio. Al igual que el caso anterior, el código base del backend [26] y del frontend [27] se alojaron en repositorios del dominio de la universidad.

3.5. Despliegue del sistema en UNLa

Para la puesta en producción se va a construir una huerta en uno de los predios de la UNLa. Para esto se seguirán las especificaciones de la figura 3.4. Como se puede apreciar, la solución contempla el armado de seis kits de dispositivos para tomar mediciones. De estos, solo uno tendrá el sensor DHT22, y todos contarán con el sensor de humedad del sustrato. En la imagen se observan unas líneas de color azul, estas representan la cañería de agua. Este montaje se llevará en conjunto con el equipo de mantenimiento de la UNLa según lo acordado al inicio del proyecto con la Universidad.

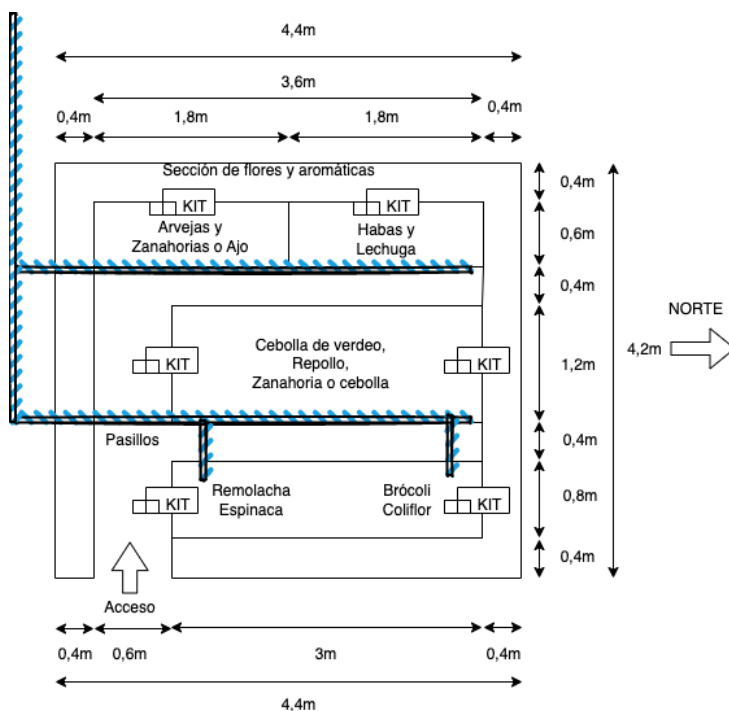


FIGURA 3.4. Mapa de la huerta en UNLa.

Capítulo 4

Ensayos y resultados

4.0.1. Prueba de concepto del dispositivo

En conjunto con Damian Reboredo se realizaron pruebas sobre la placa ESP32. Se logró llegar a un prototipo funcional donde los sensores descriptos anteriormente envían los datos a la placa. Además, se hizo una prueba de concepto para la apertura de la válvula. Finalmente, se realizó la programación para transmitir los datos a un servidor web, si bien los valores son fijos, se quería probar la conexión usando el protocolo HTTP. El código del proyecto se alojó en un repositorio de la organización de proyectos del laboratorio [25].

4.1. Banco de pruebas

4.2. Pruebas de componentes

4.3. Pruebas del backend

4.4. Pruebas del frontend

4.5. Valor agregado del proyecto

Capítulo 5

Conclusiones

5.1. Resultados obtenidos

5.2. Trabajo futuro

Bibliografía

- [1] PATRICIA AGOSTO y MARIELLE PALAU, *Hacia la construcción de la soberanía alimentaria*, Paraguay y Argentina, 2015.
- [2] ORGANIZACIÓN DE LAS NACIONES UNIDAS PARA LA ALIMENTACIÓN Y LA AGRICULTURA, *Estudio del empleo verde, actual y potencial, en el sector de bioenergías*, Salta, Argentina, 2019.
- [3] AJAY KUMAR KAVITI, AMIT KUMAR THAKUR, ANITA GEHLOT y RAJESH SINGH, *Internet of Things for Agriculture 4.0*, 2022.
- [4] UNIVERSIDAD DE CHILE, *Proyecto Piwkeyewün*, URL: <https://pueblosindigenas.ing.uchile.cl/proyecto-piwkeyewun/>, Link 2023.
- [5] CORTES-CADAVID VALERIA y VARGAS-GARCÍA, MARCO FABIAN, *Diseño e implementación de un sistema de riego automatizado y monitoreo de variables ambientales mediante Iot en los cultivos urbanos de la fundación mujeres empresarias Maria Poussepin*, URL: <https://repositorioslatinoamericanos.uchile.cl/handle/2250/3430048>, Universidad Católica de Colombia, Colombia, Link 2023.
- [6] ALLDATASHEET, *EP8266 Datasheet*, URL: <https://www.alldatasheet.com/datasheet-pdf/pdf/297157/PCA/EP8266.html>, Link 2023.
- [7] RAMÍREZ DÍAZ, ELIÉCER JESÚS VERGARA SIERRA y JESÚS DAVID, *Sistema de riego automatizado basado en iot utilizando variables ambientales para cultivos de berenjena en la finca la esperanza del municipio de Chinú-Córdoba*, URL: <https://repositorio.unicordoba.edu.co/handle/ucordoba/2706>, Universidad de Córdoba, Colombia, Link 2023.
- [8] BURRAMUÑO GONZÁLEZ ROCÍO ELENA, *Sistema automatizado para riego en huertos urbanos y plantas*, URL: <https://repositorio.usm.cl/handle/11673/50443?show=full>, Universidad Técnica Federico Santa María, Chile, Link 2023.
- [9] ESPRESSIF, *ESP32*, URL: <https://www.espressif.com/en/products/modules/esp32>, Link 2023.
- [10] ESPRESSIF, *ESP-IDF PROGRAMMING GUIDE*, URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>, Link 2023.
- [11] ESPRESSIF, *DATASHEET ESP32*, URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, Link 2023.
- [12] ESPRESSIF, *MODULES ESP32*, URL: https://docs.espressif.com/projects/esp-at/en/release-v2.2.0.0_esp32c3/Compile_and_Develop/How_to_understand_the_differences_of_each_type_of_module.html, Link 2023.
- [13] ESPRESSIF, *ESP32 DEVKIT OVERVIEW*, URL: <https://www.espressif.com/en/products/devkits/esp32-devkitc/overview>, Link 2023.

- [14] RANDOMNERDTUTORIALS, ESP32 PINOUT GPIOS, URL: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>, Link 2023.
- [15] ESPRESSIF, GITHUB ESP-IDF, URL: <https://github.com/espressif/esp-idf/tree/master/examples>, Link 2023.
- [16] ALLDATASHEET, DHT22 DATASHEET, URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>, Link 2023.
- [17] SIGMAELECTRONICA, SENSOR DE HUMEDAD DE SUELO CAPACITIVO V1.2 DATASHEET, URL: <https://www.sigmaelectronica.net/wp-content/uploads/2018/04/sen0193-humedad-de-suelos.pdf>, Link 2023.
- [18] TI, ADS1115 DATASHEET, URL: <https://www.ti.com/lit/ds/symlink/ads1114.pdf?ts=1684021319030>, Link 2023.
- [19] PROGRAMAR FÁCIL, ADS1115 TUTORIAL DE USO, URL: <https://programarfacil.com/blog/arduino-blog/ads1115-convertidor-analogico-digital-adc-arduino-esp8266/>, Link 2023.
- [20] ORACLE, JAVA, URL: <https://www.java.com/es/>, Link 2023.
- [21] SPRING, SPRING BOOT, URL: <https://spring.io/>, Link 2023.
- [22] ORACLE, MYSQL, URL: <https://www.mysql.com/>, Link 2023.
- [23] JAVASCRIPT, URL: <https://developer.mozilla.org/es/docs/Web/JavaScript>, Link 2023.
- [24] META, REACT, URL: <https://es.react.dev/>, Link 2023.
- [25] UNLA, DISPOSITIVOS ESP32, URL: <https://github.com/unla-investigacion-desarrollo/agroecologia-iot-dispositivo>, Link 2023.
- [26] UNLA, BACKEND SPRING, URL: <https://github.com/unla-investigacion-desarrollo/agroecologia-iot>, Link 2023.
- [27] UNLA, FRONTEND REACT, URL: <https://github.com/unla-investigacion-desarrollo/agroecologia-iot-front>, Link 2023.