

## CARRERA DE ESPECIALIZACIÓN EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

### Robot de exploración ambiental

**Autor:**  
**Ing. Gonzalo Carreño**

Director:  
Esp. Ing. Sergio Alberino (UTN.BA)

Jurados:  
Nombre del jurado 1 (pertenencia)  
Nombre del jurado 2 (pertenencia)  
Nombre del jurado 3 (pertenencia)

*Este trabajo fue realizado en la Ciudad Autónoma de Buenos Aires,  
entre marzo de 2023 y noviembre de 2023.*



## *Resumen*

El presente trabajo describe un emprendimiento personal en el que se desarrolla un dispositivo robótico de exploración ambiental, controlable a distancia con las funciones básicas de desplazamiento, medición y reporte de parámetros ambientales tales como presión, temperatura, humedad y luminosidad.

El sistema cuenta con una arquitectura base robusta y flexible sobre la cual otros sensores y actuadores pueden ser adaptados para poder interactuar con el medio ambiente, y logra por tanto brindar una solución que ayuda a incrementar la oferta de robots exploradores en la industria argentina pudiendo ser utilizados en casos de uso de IoT, como por ejemplo en aplicaciones de explotación de petróleo, Smart Mining, Smart Farming, etc.

Para su implementación se utilizaron conceptos y herramientas tales como buenas prácticas en el diseño y desarrollo de firmware, la utilización de sistemas operativos de tiempo real como plataforma de ejecución base, protocolos de comunicaciones para sistemas embebidos, y técnicas y frameworks de testing para asegurar la calidad del producto final.



## *Agradecimientos*

Esta sección es para agradecimientos personales y es totalmente **OPCIONAL**.



# Índice general

<b>Resumen</b>	<b>I</b>
<b>1. Introducción general</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Estado del arte . . . . .	1
1.3. Alcance y objetivos . . . . .	2
1.4. Requerimientos . . . . .	2
<b>2. Introducción específica</b>	<b>5</b>
2.1. Tecnologías de hardware utilizadas . . . . .	5
2.1.1. Espressif ESP32 . . . . .	5
2.1.2. Sensor de temperatura y humedad DHT11 . . . . .	5
2.1.3. Sensor de presión BMP280 . . . . .	6
2.1.4. Fotoresistor como sensor de luminosidad . . . . .	6
2.1.5. Joystick analógico . . . . .	6
2.1.6. Display LCM1602A . . . . .	7
2.1.7. Motores de corriente continua . . . . .	7
2.1.8. Módulos L298N . . . . .	7
2.1.9. Pilas de Li-Ion 3.7 V . . . . .	8
2.1.10. Baterías AA 1.5 V . . . . .	8
2.1.11. Plaquetas genéricas PCB . . . . .	9
2.1.12. Cables de conexión DuPont . . . . .	9
2.1.13. Board Pin Headers para montaje de componentes y cables . .	10
2.1.14. Interruptores de On-Off . . . . .	10
2.1.15. Portapilas . . . . .	10
2.1.16. Ruedas . . . . .	11
2.2. Tecnologías de software utilizadas . . . . .	11
2.2.1. Marco de trabajo ESP-IDF . . . . .	11
2.2.2. Plataforma Docker . . . . .	12
2.2.3. Visual Studio Code . . . . .	12
2.2.4. Sistema operativo Ubuntu . . . . .	13
<b>3. Diseño e implementación</b>	<b>15</b>
3.1. Arquitectura de software del sistema . . . . .	15
3.1.1. Arquitectura y diseño de componentes en la versión v1.0 . .	15
3.1.2. Arquitectura y diseño de componentes en la versión v2.0 . .	17
3.2. Implementación de los módulos . . . . .	19
3.2.1. Control de la red WIFI . . . . .	20
3.2.2. Control del joystick analogico . . . . .	20
3.2.3. Medición de valor de luminosidad . . . . .	21
3.2.4. Medición de temperatura y humedad . . . . .	21
3.2.5. Medición de presión . . . . .	22

3.2.6. Control de motores DC . . . . .	22
3.2.7. Control del display . . . . .	23
3.3. Arquitectura de hardware . . . . .	24
3.3.1. Conexionado lógico . . . . .	24
3.3.2. Conexionado físico del robot . . . . .	25
3.4. Ciclo de desarrollo . . . . .	26
<b>4. Ensayos y resultados</b> . . . . .	<b>27</b>
4.1. Proceso de desarrollo y aseguramiento de calidad . . . . .	27
4.2. Verificación técnica de los diferentes módulos . . . . .	27
4.2.1. Verificación del módulo de joystick . . . . .	27
4.2.2. Verificación del módulo de control del display . . . . .	28
4.2.3. Verificación del módulo de control de motores . . . . .	28
4.2.4. Verificación del módulo de medición de temperatura y humedad . . . . .	28
4.2.5. Verificación del modulo de medicion de presion atmosferica	28
4.2.6. Verificación del módulo de medición de luminosidad . . . . .	28
4.2.7. Verificación del módulo de comunicación UTP sobre WiFi .	28
4.3. Pruebas funcionales del producto . . . . .	28
4.3.1. Prueba del módulo de medición de temperatura y humedad	28
4.3.2. Prueba del módulo de medición de presión . . . . .	29
4.3.3. Prueba del módulo de medición de luminosidad ambiental	29
4.3.4. Prueba del control y desplazamiento del robot . . . . .	29
4.3.5. Prueba del módulo de visualización de display . . . . .	30
4.4. Reportes de testing . . . . .	30
4.5. Documentación del producto . . . . .	30
<b>5. Conclusiones</b> . . . . .	<b>31</b>
5.1. Próximos pasos . . . . .	31
<b>Bibliografía</b>	<b>33</b>

# Índice de figuras

2.1.	Microcontrolador ESP32-WROOM-32D . . . . .	5
2.2.	Sensor DHT11. . . . .	6
2.3.	Sensor BMP280. . . . .	6
2.4.	Fotoresistor. . . . .	6
2.5.	Joystick analógico. . . . .	7
2.6.	Display LCM1602A. . . . .	7
2.7.	Motor de corriente continua. . . . .	7
2.8.	Interruptores de On-Off . . . . .	8
2.9.	Baterias Li-Ion . . . . .	8
2.10.	Baterías AA . . . . .	9
2.11.	Plaquetas genericas . . . . .	9
2.12.	Cables DuPont . . . . .	9
2.13.	Pines . . . . .	10
2.14.	Interruptores de On-Off . . . . .	10
2.15.	Porta Pilas . . . . .	11
2.16.	Ruedas . . . . .	11
2.17.	Proceso de desarrollo utilizando ESP-IDF <sup>1</sup> . . . . .	12
3.1.	Arquitectura global. . . . .	16
3.2.	Arquitectura global. . . . .	18
3.3.	Conexionado joystick. . . . .	21
3.4.	Conexionado fotorresistor. . . . .	21
3.5.	Circuito del conexionado DHT11. . . . .	22
3.6.	Conexionado BMP280. . . . .	22
3.7.	Conexionado motores. . . . .	23
3.8.	Conexionado display. . . . .	24
3.9.	Conexionado del robot. . . . .	24
3.10.	Conexionado del joystick. . . . .	25
3.11.	Conexionado físico del robot. . . . .	25



# Índice de tablas

3.1. Configuración de AP WiFi . . . . .	20
3.2. Conexionado joystick . . . . .	20
3.3. Conexionado fotoresistor . . . . .	21
4.1. Resultados de mediciones de temperatura y humedad . . . . .	29
4.2. Resultados de mediciones de presión ambiental . . . . .	29



*Dedicado a mi familia y mis amigos.*



# Capítulo 1

## Introducción general

Esta sección presenta la motivación, alcance, objetivos y requerimientos del producto en el marco del estado del arte y su importancia en la industria.

### 1.1. Motivación

La motivación del presente trabajo fue primeramente volcar y unificar en un emprendimiento personal los conceptos aprendidos en la especialización de Sistemas Embebidos, con una arquitectura robusta que pueda ser extrapolada a otros casos de uso de valor en la industria como por ejemplo la exploración de suelos en el agro, la exploración submarina para la perforación de pozos de petróleo, o los mencionados más adelante en el estado del arte. Por otra parte, se buscó desarrollar un producto que pueda contribuir a aumentar la oferta de dispositivos robóticos exploradores en Argentina.

### 1.2. Estado del arte

Los robots exploradores son dispositivos robotizados capaces de moverse de forma autónoma, y/o controlados a distancia, que han sido creados con el fin de reconocer y explorar un lugar o entorno donde una persona no pueda o deba acceder ya sea por motivos de capacidad, practicidad o seguridad. Por este motivo, en función de las necesidades de desplazamiento, existen diferentes sistemas de motricidad, como son por ejemplo, los bípedos, cuadrúpedos, con ruedas, tracción oruga, acuáticos/sumergibles, aéreos, etc. En cuanto a la forma de control, los hay manejados por control remoto cableado o inalámbrico, habiendo equipos más sofisticados, que gracias a aplicaciones de Inteligencia Artificial, están preparados para desplazarse y tomar decisiones de forma autónoma. Algunos de los tipos de robots exploradores más conocidos son los espaciales, de minas, de rescate en catástrofes, de tuberías, submarinos, y de suelos.

Tanto en el ámbito académico como en la industria existen trabajos, proyectos, e implementaciones comerciales similares al presente trabajo, como por ejemplo:

- El prototipo robótico de exploración minera publicado en varios artículos [1], [2], e impulsado por el Instituto de Automática de la Facultad de Ingeniería de la Universidad Nacional de San Juan en el marco de un convenio con la Comisión Nacional de Energía Atómica y el Gobierno argentino [3].
- El robot de exploración terrestre denominado Geobot [4] desarrollado por los ingenieros Nelson Dario García Hurtado y Melvin Andrés González

Pino, de la universidad de Pamplona, capaz de realizar reconocimiento de zonas y manipulación de muestras de manera autónoma o asistida.

- El robot minero MIN-SIS 1.0 SDG-STR [5] desarrollado por los ingenieros Hernán L. Helguero Velásquez y Rubén Medinaceli Tórrez de la Universidad Técnica de Oruro, capaz de detectar gases, almacenar datos locales y enviar video e imágenes al puesto de mando.
- Spot [6], desarrollado por Boston Dynamics, un robot explorador cuadrupeado de propósito general capaz de explorar, almacenar y enviar información en tiempo real.
- BIKE [7], desarrollado por Waygate Technologies, un robot con ruedas magnéticas, muy utilizado en la industria de petróleo y gas entre otras, capaz de desplazarse por el interior de tuberías para poder realizar inspecciones y comunicar hallazgos.

### 1.3. Alcance y objetivos

A continuación se detallan las funcionalidades incluidas en el alcance del trabajo.

- Sistema de desplazamiento terrestre.
- Operaciones de exploración
  - Medición de humedad ambiental.
  - Medición de temperatura ambiental.
  - Medición de presión ambiental.
  - Medición de luminosidad ambiental.
- Visualización de estado de exploración (lecturas de los sensores).
- Sistema de control por medio de un joystick cableado.

Queda fuera del alcance:

- Locomoción por cualquier otro medio que no sea terrestre.
- Cualquier otra función no contemplada en este alcance.

### 1.4. Requerimientos

A continuación se listan los requerimientos del producto:

1. Requerimientos funcionales
  - a) El sistema debe contar con funciones de desplazamiento para poder moverse hacia adelante y atrás, y poder girar radialmente un ángulo de 360 grados.
  - b) El sistema debe ser capaz de realizar las siguientes operaciones de exploración:
    - Medición de humedad ambiental.
    - Medición de temperatura ambiental.

- Medición de luminosidad ambiental.
  - Medición de presión ambiental.
- c) El sistema debe poder ser controlado a distancia mediante un joystick para que el dispositivo pueda realizar sus movimientos. En caso de que alguna de sus operaciones de exploración requiera algún otro mecanismo de control, el mismo también será integrado en el joystick.
- d) El sistema debe proveer un mecanismo de visualización de las operaciones de exploración al usuario que controla el dispositivo para poder ver el estado y lectura de las operaciones de exploración.
- La interfaz de usuario debe permitir visualizar las lecturas de cada uno de los sensores.
  - Debe haber una pequeña leyenda de la magnitud que se está midiendo y la unidad utilizada junto con el valor.

## 2. Requerimientos no funcionales

- a) La arquitectura del producto debe ser robusta y tolerante a fallas.
- b) A fin de maximizar la mantenibilidad, la arquitectura del producto debe estar modularizada para permitir que los diferentes módulos puedan ser integrados y orquestados separadamente.



## Capítulo 2

# Introducción específica

Esta sección presenta una breve introducción técnica a las herramientas hardware y software utilizadas en el trabajo.

### 2.1. Tecnologías de hardware utilizadas

#### 2.1.1. Espressif ESP32

ESP32 [8] es una serie de microcontroladores embebidos en un chip con Wi-Fi y Bluetooth integrados, de bajo costo y consumo, desarrollado por *Espressif Systems*. Emplea dos cores Xtensa® 32-bit LX6 CPU, incluye interruptores de antena, amplificador de potencia, amplificador de recepción de bajo ruido, un co-procesador ULP (Ultra Low Power), módulos de administración de energía y varios periféricos. En la siguiente imagen (2.1) se puede apreciar la placa ESP32-WROOM-32D [9] utilizada para el desarrollo del presente trabajo.



FIGURA 2.1. Microcontrolador ESP32-WROOM-32D.

#### 2.1.2. Sensor de temperatura y humedad DHT11

El DHT11 [10] es un sensor digital de temperatura y humedad relativa de bajo costo y fácil uso. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica). Entre otras aplicaciones se lo suele utilizar principalmente en aplicaciones relacionadas al control automático de temperatura, aire acondicionado y monitoreo ambiental en agricultura. En la figura 2.2 se puede apreciar una imagen del componente.



FIGURA 2.2. Sensor DHT11.

### 2.1.3. Sensor de presión BMP280

El BMP280 [11] es un sensor de presión barométrica absoluta, especialmente factible para aplicaciones móviles que puede ser utilizado con I2C o SPI. Permite alta precisión y linealidad, estabilidad a largo plazo, alta robustez a un muy bajo consumo. En la figura 2.3 se puede apreciar una imagen del componente.



FIGURA 2.3. Sensor BMP280.

### 2.1.4. Fotoresistor como sensor de luminosidad

El fotoresistor es una resistencia eléctrica que varía su valor en función de la cantidad de luz que incide sobre su superficie. Cuando el fotoresistor no está expuesto a radiaciones luminosas, los electrones están firmemente unidos en los átomos que lo conforman, por lo que alcanza su máxima resistencia eléctrica, y cuando sobre él inciden radiaciones luminosas, esta energía libera los electrones con lo cual el material se vuelve más conductor, y se disminuye su resistencia. En la figura 2.4 se puede apreciar una imagen del componente.



FIGURA 2.4. Fotoresistor.

### 2.1.5. Joystick analógico

El módulo de joystick analógico [12] está construido sobre el montaje de dos potenciómetros en un ángulo de 90 grados. Los potenciómetros están conectados a una palanca corta centrada por resortes. Este módulo produce una salida de alrededor de 2,5 Volts cuando la palanca se encuentra en reposo (en el centro), mientras que al desplazarse hará que la salida varíe de 0 a 5 Volts dependiendo de su posición. La obtención de los valores en Volts se obtiene tras convertir las lecturas en niveles lógicos mediante el módulo ADC (conversor analógico digital)

[13] del microcontrolador ESP32. En la figura 2.5 se puede apreciar una imagen del componente.



FIGURA 2.5. Joystick analógico.

#### 2.1.6. Display LCM1602A

El display LCM1602A [14] consta de una pantalla de cristal líquido que permite representar dos filas con hasta 16 caracteres alfanuméricos en cada una y dado que se encuentra integrada a una interfaz adaptadora I2C puede ser controlada por este protocolo. En la figura 2.6 se puede apreciar una imagen del componente.



FIGURA 2.6. Display LCM1602A.

#### 2.1.7. Motores de corriente continua

El motor DC (corriente continua) [15] es un electromotor que transforma energía eléctrica en energía mecánica. Estos motores operan con un voltaje entre 3 y 6 Volts, corriente de 150 mA, permiten una velocidad de entre 90 y 200 RPM y un torque de entre 0,15 Nm y 0,60 Nm. En la figura 2.7 se puede apreciar una imagen del componente.



FIGURA 2.7. Motor de corriente continua.

#### 2.1.8. Módulos L298N

Los módulos L298N son controladores de motores de puente H duales muy utilizados en proyectos de robótica y automoción para controlar motores de corriente continua (DC) y motores paso a paso. El módulo se basa en el IC L298N, que es un controlador de motor de doble puente H fabricado por STMicroelectronics. Este módulo posee dos puentes H que permiten controlar 2 motores DC o un motor paso a paso bipolar/unipolar. El módulo permite controlar el sentido de giro y velocidad mediante señales TTL (transistor-transistor logic) que se pueden obtener de microcontroladores. Tiene integrado un regulador de voltaje de 5V

encargado de alimentar la parte lógica del L298N cuya configuración se hace a través de un Jumper y se puede usar para alimentar el microcontrolador.

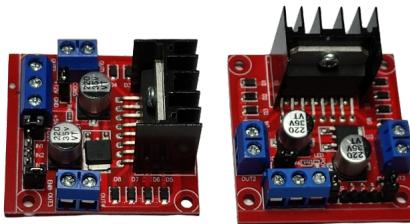


FIGURA 2.8. Interruptores de On-Off

### 2.1.9. Pilas de Li-Ion 3.7 V

Las pilas 18650 de 3000 mAh son baterías recargables que utilizan tecnología de iones de litio y se emplean comúnmente en una variedad de dispositivos electrónicos, como por ejemplo linternas de alta potencia, laptops, vehículos eléctricos (como bicicletas y scooters), etc. Las baterías de iones de litio son conocidas por su alta densidad de energía, su baja tasa de autodescarga y su larga vida útil en comparación con otras tecnologías de baterías recargables. Con 3000 mAh (miliamperios-hora), estas baterías ofrecen una duración prolongada, lo que es ideal para dispositivos que requieren mucha energía. Al ser recargables, son más ecológicas y económicas a largo plazo en comparación con las baterías desechables. Además, son generalmente seguras y tienen una buena estabilidad térmica, reduciendo el riesgo de explosiones o incendios.



FIGURA 2.9. Baterias Li-Ion

### 2.1.10. Baterías AA 1.5 V

Las pilas AA de 1.5V con 2600 mWh son baterías de tamaño AA que proporcionan una tensión de 1.5 voltios y tienen una capacidad energética de 2600 miliwatts-hora (mWh). Ofrecen alta capacidad pudiendo durar más tiempo entre recargas o reemplazos en comparación con las baterías de menor capacidad, y versatilidad, ya que al ser AA son compatibles con la mayoría de los dispositivos electrónicos que requieren alimentación de bajo voltaje. Las pilas AA de 1.5V con una alta capacidad como 2600 mWh son ideales para dispositivos que requieren un mayor consumo de energía, tales como cámaras digitales, linternas, juguetes electrónicos, mandos a distancia, controles de videojuegos, etc.



FIGURA 2.10. Baterías AA

### 2.1.11. Plaquetas genéricas PCB

Las plaquetas electrónicas genéricas del tipo PCB (Printed Circuit Board), también conocidas como placas de desarrollo, son herramientas utilizadas principalmente en la educación, el diseño y el prototipado de circuitos electrónicos que constan de material aislante con patrones de cobre en una o ambas caras, donde se pueden soldar componentes para crear circuitos permanentes. Permiten el montaje de circuitos electrónicos de manera más duradera y fiable que el uso de un Protoboard y son asequibles de forma online a un bajo costo.



FIGURA 2.11. Plaquetas genericas

### 2.1.12. Cables de conexión DuPont

Los cables Dupont son cables de conexión eléctrica utilizados comúnmente en prototipado y proyectos de electrónica. Estos cables son muy populares en la comunidad de los entusiastas y estudiantes de electrónica debido a su facilidad de uso y versatilidad. Están hechos de hilos de cobre con una cubierta de plástico aislante. Los conectores suelen ser de metal chapado para asegurar una buena conductividad. Los hay del tipo Hembra-Hembra, Hembra-Macho y Macho-Macho. Entre sus usos más comunes se encuentran la conexión de dispositivos en Proto-board sin necesidad de soldadura y la interconexión entre sensores, actuadores y terminales en pines o bahías de plaquetas impresas.



FIGURA 2.12. Cables DuPont

### 2.1.13. Board Pin Headers para montaje de componentes y cables

Los pines de cabecera son componentes electrónicos que consisten en filas de pines metálicos montados en una base plástica. Estos pines se utilizan para realizar conexiones eléctricas entre diferentes componentes y placas en proyectos electrónicos. La cantidad de pines puede variar en número, desde unos pocos pines hasta decenas, dependiendo de las necesidades del proyecto. Pueden encontrarse en varias configuraciones, como ser Single Row -una sola fila de pines-, Double Row -dos filas de pines, aumentando la capacidad de conexión en un espacio compacto-, Right Angle -pines que se extienden en ángulo recto respecto a la base, útiles para conexiones horizontales-. Se los puede implementar con dos tipos posibles de montajes: Through-Hole (Agujero Pasante) -donde los pines se insertan a través de agujeros en la placa y se sueldan en el lado opuesto-; o bien Surface-Mount (Montaje Superficial), donde los pines se sueldan directamente en la superficie de la placa, sin pasar a través de ella.



FIGURA 2.13. Pines

### 2.1.14. Interruptores de On-Off

Los interruptores de encendido-apagado (on-off) son componentes fundamentales en circuitos electrónicos que permiten controlar el flujo de corriente eléctrica. Son utilizados para conectar o desconectar la corriente en un circuito, funcionando como un medio sencillo y efectivo para gestionar la energía de dispositivos electrónicos. Se los puede encontrar en diferentes tipos, Interruptores Basculantes (Rocker Switches) -se operan basculando de una posición a otra-, interruptores deslizantes (Slide Switches) -se operan deslizando un pequeño botón a lo largo de una pista-, interruptores de palanca (Toggle Switches) -utilizan una palanca que se mueve hacia arriba y hacia abajo para cambiar entre las posiciones on y off, e interruptores de pulsador (Push-Button Switches) -al presionar el botón se conecta o desconecta el circuito-.



FIGURA 2.14. Interruptores de On-Off

### 2.1.15. Portapilas

Los porta pilas son componentes diseñados para alojar y conectar baterías en un circuito electrónico. Estos dispositivos, generalmente hechos de plástico resistente con contactos metálicos (normalmente de níquel o acero) para asegurar

una buena conductividad, facilitan la instalación y el reemplazo de baterías, asegurando una conexión segura y confiable. Se los puede encontrar comúnmente para los tipos de pilas y baterías más utilizadas, como por ejemplo AA, AAA, 9V, 18650, CR2032, etc. Además, suelen venir en tres posibles configuraciones, un solo compartimento (para una sola batería), múltiples compartimentos (para conectar varias baterías en serie o en paralelo), y con interruptor (algunos portapilas incluyen un interruptor para encender o apagar la conexión de las baterías al circuito).



FIGURA 2.15. Porta Pilas

### 2.1.16. Ruedas

Las ruedas de plástico son componentes genéricos utilizados en la comunidad hobbista y estudiantil de electrónica para el armado de sistemas con desplazamiento. Se las puede comprar online a un bajo costo, usualmente vienen en kit de dos o cuatro unidades, y son compatibles con los motores de DC utilizados para moverlas.



FIGURA 2.16. Ruedas

## 2.2. Tecnologías de software utilizadas

### 2.2.1. Marco de trabajo ESP-IDF

*Espressif Systems* proporciona recursos básicos de hardware y software para ayudar a los desarrolladores de aplicaciones a realizar sus ideas utilizando el hardware de la serie ESP32. El framework de software de Espressif está destinado al desarrollo de aplicaciones de IoT (Internet de las cosas) con Wi-Fi, Bluetooth, administración de energía y varias otras características del sistema. Sus componentes son:

1. Toolchain, utilizado para compilar el código para ESP32.

2. Build tools, que provee utilidades como CMake [16] y Ninja [17] para construir la aplicación completa para ESP32.
3. ESP-IDF [18], que brinda la API de desarrollo para ESP32 y scripts para ejecutar Toolchain.

Además de las herramientas mencionadas se utilizó el conjunto de bibliotecas y drivers provistos por el proyecto ESP-IDF-Lib [19] basados en el framework ESP-IDF.

En la figura 2.17 se puede apreciar una imagen del proceso de desarrollo y despliegue usando el framework ESP-IDF.

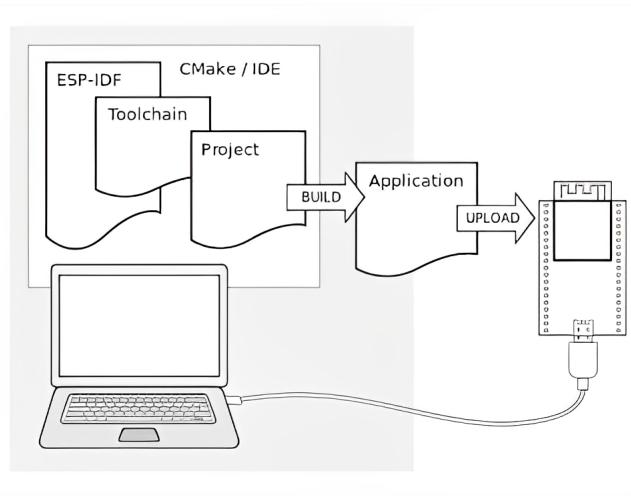


FIGURA 2.17. Proceso de desarrollo utilizando ESP-IDF<sup>1</sup>.

### 2.2.2. Plataforma Docker

Docker [21] es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que contenedores livianos independientes se ejecuten en paralelo de manera aislada evitando la sobrecarga de iniciar y mantener máquinas virtuales.

### 2.2.3. Visual Studio Code

Visual Studio Code [22] es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

---

<sup>1</sup>Imagen tomada de [20]

### **2.2.4. Sistema operativo Ubuntu**

Ubuntu [23] es una distribución Linux basada en Debian GNU/Linux y patrocinado por Canonical, que incluye principalmente software libre y de código abierto. Puede utilizarse en ordenadores y servidores, está orientado al usuario promedio, con un fuerte enfoque en la facilidad de uso y en mejorar la experiencia del usuario.



## Capítulo 3

# Diseño e implementación

Esta sección presenta los detalles técnicos del diseño e implementación de las diferentes funcionalidades del producto, la arquitectura hardware y software, y finalmente la interfaz de usuario para el control y reporte de las operaciones del robot.

El proyecto fue realizado siguiendo una metodología basada en crear un prototipo funcional con una arquitectura escalable que cumpla con alcance básico especificado en el plan de proyecto, y una vez conseguido agregar en lo posible y de acuerdo a la capacidad, funcionalidades adicionales. De esta forma, una vez que se logró el alcance básico, considerado como la versión v1.0 del producto, se continuó con el desarrollo y se agregó la funcionalidad adicional de control inalámbrico por medio de un joystick, en lo que se considera la versión v2.0 del mismo.

En las siguientes sub secciones se detallan los detalles del diseño e implementación realizados para ambas versiones.

### 3.1. Arquitectura de software del sistema

Con el fin de poder lograr una arquitectura de software modular, se implementaron diferentes componentes software y servicios que abstraen el acceso a los módulos de hardware (explicados en la siguiente sección) y permite un escalamiento fácil de funcionalidades en la expansión de las funcionalidades del robot entre su versión v1.0 y la v2.0.

#### 3.1.1. Arquitectura y diseño de componentes en la versión v1.0

A continuación podemos apreciar la arquitectura de software del sistema en su versión v1.0 y el detalle de sus componentes y servicios.

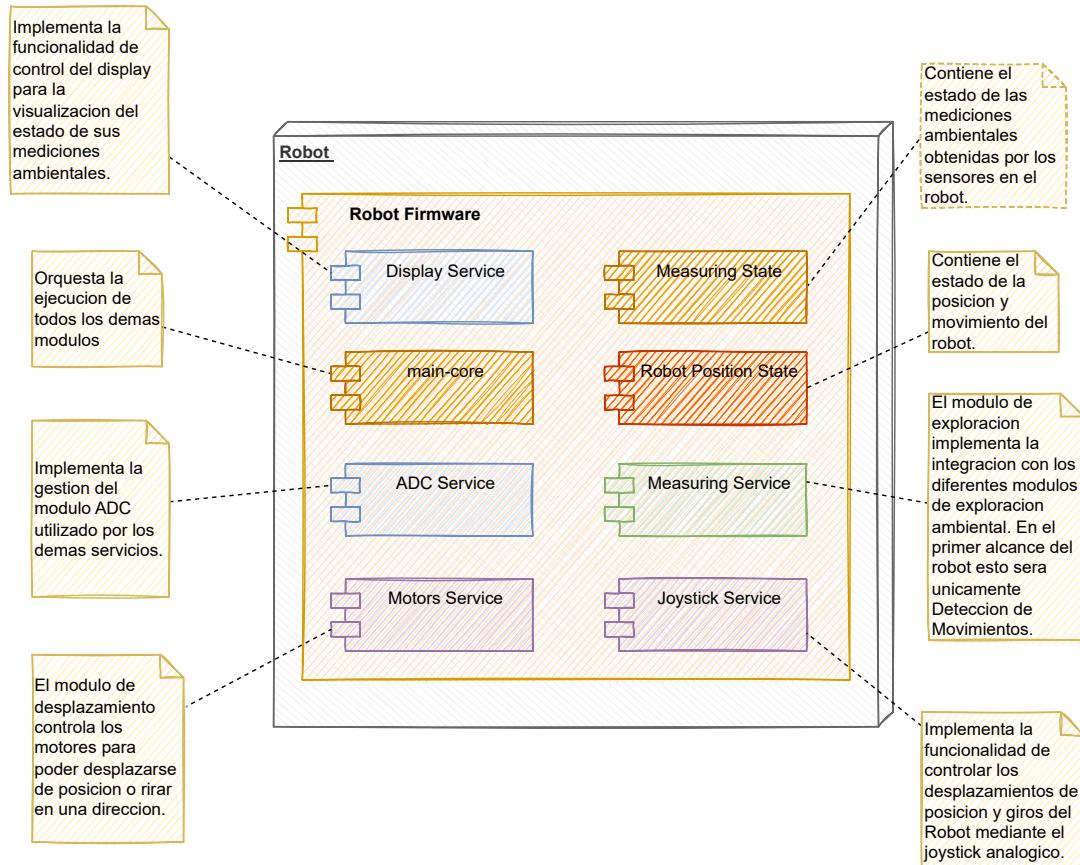


FIGURA 3.1. Arquitectura global.

Los componentes y servicios de software del Robot en la versión v1 son los siguientes:

### 1. Componentes de software

- a) main-core: orquesta las diferentes tareas desde las que se invocan los demás componentes de software.
- b) Measuring Service: abstrae el acceso a los módulos de medición de temperatura, humedad, presión y luminosidad.
- c) Measuring State: mantiene el estado de cada uno de los parámetros ambientales medidos.
- d) Robot Position State: mantiene el estado del movimiento actual del robot.
- e) ADC Service: abstrae el acceso a los módulos que hacen uso de los canales que requieren conversiones ADC (Analogico/Digital), como el joystick analógico y el fotoresistor.
- f) Motors Service: abstrae el acceso al módulo de control de motores.
- g) Display Service: abstrae el acceso al módulo de control del display mediante Ic2.
- h) Joystick Service: abstrae el acceso al módulo de control del joystick.

### 2. Servicios (tareas)

- a) Display Task
- b) Measuring Task
- c) Joystick Task
- d) Motors Task

### 3.1.2. Arquitectura y diseño de componentes en la versión v2.0

Luego de expandir su funcionalidad, los componentes y servicios de software se separaron físicamente en el hardware del robot y el del joystick, agregando además los necesarios para la comunicación inalámbrica. A continuación podemos apreciar la arquitectura de software del sistema en su versión v2.0 y el detalle de sus componentes y servicios.

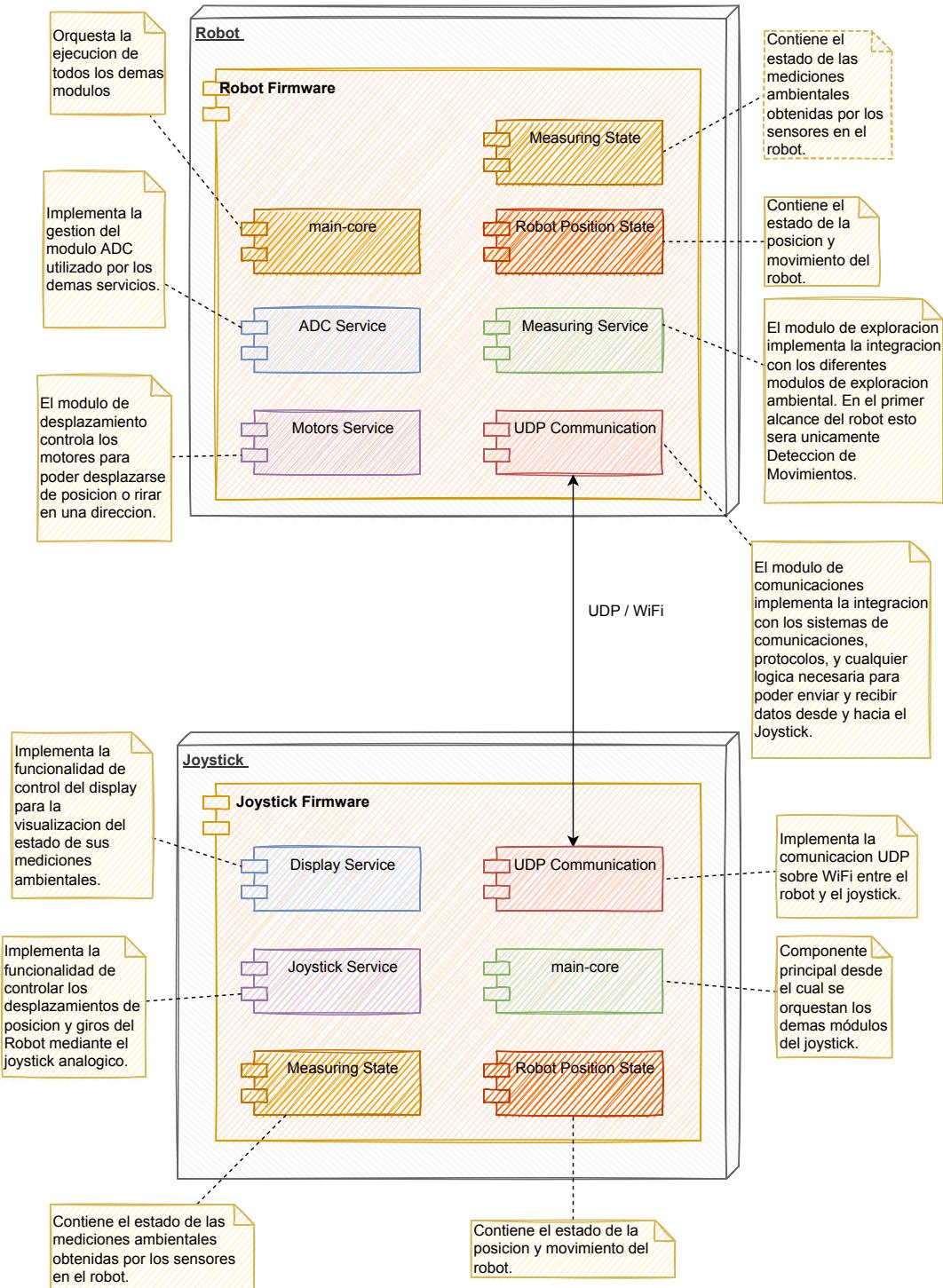


FIGURA 3.2. Arquitectura global.

Los componentes de software y servicios del robot son:

1. Componentes de software
  - a) main-core
  - b) Measuring Service
  - c) Measuring State

- d) Robot Position State
  - e) ADC Service
  - f) Motors Service
  - g) WIFI Service
  - h) UDP Service
2. Servicios (tareas)
- a) Measuring Task
  - b) Motors Task
  - c) UDP Server Task

Los componentes de software y servicios del joystick son:

1. Componentes de software
- a) main-core
  - b) Measuring State
  - c) Robot Position State
  - d) UDP Service
2. Servicios (tareas)
- a) Display Task
  - b) Joystick Task
  - c) UDP Server Task

## 3.2. Implementación de los módulos

Los macro componentes de hardware presentes en la arquitectura de la versión v2.0 son el robot y el joystick, los cuales constituyen dos sistemas embebidos independientes implementados con dos microcontroladores ESP32, integrados entre sí por medio de una red WiFi y una comunicación UDP. Los modulos hardware desarrollados son los siguientes.La arquitectura de hardware del sistema esta compuesta por los siguientes módulos:

- En el robot:
  - Control de los motores DC.
  - Control de los sensores de medición (DHT11, BMP280 y fotoresistor).
  - Gestión de la comunicación inalámbrica vía WIFI (en la versión v2.0).
- En el joystick:
  - Control del display.
  - Control del joystick.
  - Gestión de la comunicación inalámbrica vía WIFI (en la versión v2.0).

La integración de los mismos se realizó mediante el diseño y construcción de una placa integradora central, que conecta los dispositivos hardware con el microcontrolador ESP-32.

### 3.2.1. Control de la red WIFI

El módulo de red WiFi está integrado en el chip ESP32 en el cual se soportan múltiples features [24] por lo cual a nivel hardware no fue necesario realizar ningún conexiónado. A nivel software, la gestión del módulo WiFi se encuentra incluida en el SDK ESP-IDF, y el acceso al mismo se realiza desde el módulo ADC 2 [13]. Por este motivo, cuando el sistema embebido utiliza el módulo WiFi, el uso del ADC2 queda restringido a esta funcionalidad por lo cual cualquier otro dispositivo que deba hacer uso del ADC debe ser configurado para utilizar el ADC1, como por ejemplo los módulos de joystick y detección de luminosidad, explicados en las siguientes secciones.

La configuración del *soft access point WiFi* implementado se realizó en base a los parámetros de red detallados en la siguiente tabla 3.1:

TABLA 3.1. Configuración de AP WiFi

Parámetro	Valor
SSID	Robot
password	Robot

El desarrollo de este módulo se basó en el ejemplo provisto por Espressif [25]. El código fuente del prototipo realizado puede apreciarse en el siguiente enlace [26].

### 3.2.2. Control del joystick analogico

Para el desarrollo de este prototipo se utilizó el SDK ESP-IDF y se configuró el módulo ADC1 de acuerdo a los siguientes detalles en la tabla 3.3:

TABLA 3.2. Conexionado joystick

Channel	Unit	Pin GPIO
6	1	34
7	1	35

A continuación se puede apreciar el conexionado físico en la figura 3.3.

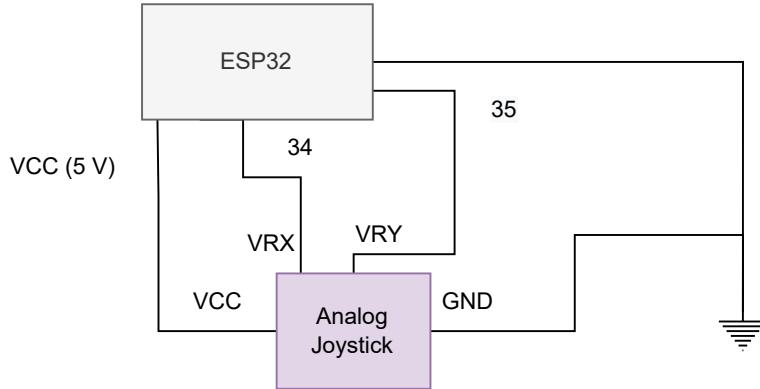


FIGURA 3.3. Conexionado joystick.

El desarrollo de este modulo se baso en el ejemplo provisto por Espressif [27]. El codigo fuente del prototipo realizado puede apreciarse en el siguiente enlace [28].

### 3.2.3. Medición de valor de luminosidad

Para el desarrollo de este prototipo se utilizó el SDK de ESP-IDF y se configuró el módulo ADC1 de acuerdo a los detalles provistos en la siguiente tabla 3.3

TABLA 3.3. Conexionado fotoresistor

Channel	Unit	Pin GPIO
0	1	36

A continuación se puede apreciar un diagrama de su conexionado físico en la figura 3.4.

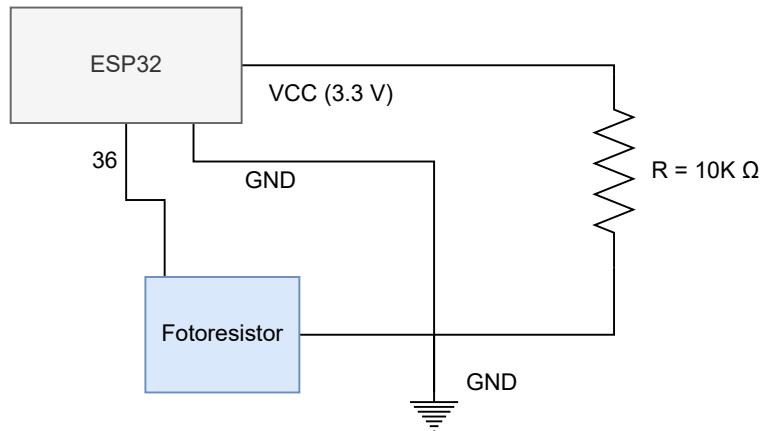


FIGURA 3.4. Conexionado fotoresistor.

El desarrollo de este módulo se basó en el ejemplo provisto por Espressif [27]. El código fuente del prototipo realizado puede apreciarse en el siguiente enlace [29].

### 3.2.4. Medición de temperatura y humedad

Para el desarrollo de este modulo se utilizó la biblioteca de código ESP-IDF-Lib Components Library [19] que provee el soporte para gestionar el DHT11. Para acceder a las lecturas del dispositivo se abstrajo el mismo mediante el componente

Measuring Service, el cual es invocado por la tarea Measuring Task y el estado de la lectura es almacenado en el componente Measuring State. A continuación se puede apreciar el conexionado del prototipo en la figura 3.5.

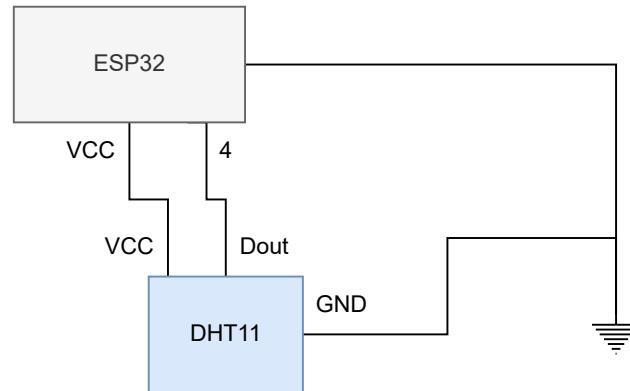


FIGURA 3.5. Circuito del conexionado DHT11.

El desarrollo de este módulo se basó en el ejemplo provisto por la librería ESP-IDF-Lib [30]. El código fuente del prototipo realizado puede apreciarse en el siguiente enlace [31].

### 3.2.5. Medición de presión

Para el desarrollo de este módulo se utilizó el framework ESP-IDF y la biblioteca de código ESP-IDF Components que provee el soporte para gestionar el dispositivo BMP280 por medio del protocolo I2C. El driver es inicializado en el componente main-core para ser posteriormente invocado desde el Measuring Service en la tarea Measuring Task. Sus lecturas son guardadas en el Measuring State. A continuación se puede apreciar el conexionado del prototipo en la figura 3.6.

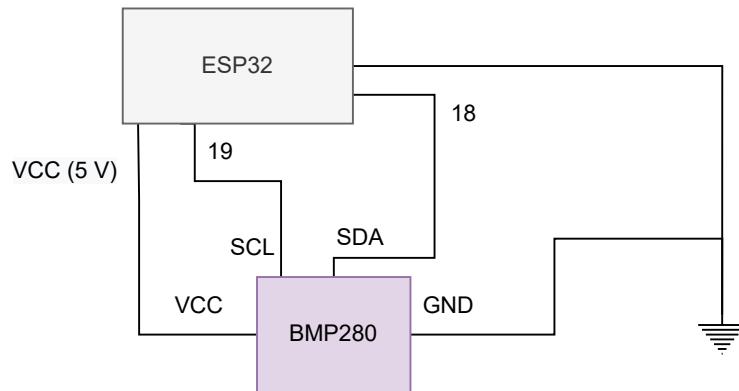


FIGURA 3.6. Conexión BMP280.

El desarrollo de este módulo se basó en el ejemplo provisto por la librería ESP-IDF-Lib [32]. El código fuente del prototipo realizado puede apreciarse en el siguiente enlace [33].

### 3.2.6. Control de motores DC

Para la implementación del módulo de control de los motores de corriente continua se utilizaron a nivel hardware dos módulos puentes L298N [34] que permiten integrar y controlar dos motores cada uno. Con el fin de alimentar los módulos

con una fuente de poder de amperaje y voltaje consistente se utilizaron dos baterías de Li-Ion de 3.7 V y 2000 mA/h conectadas en serie activadas mediante un interruptor. A nivel driver en el ESP32 se utilizó el módulo de control de motores por modulación de pulsos (MCPWM) [35] que por medio de la configuración de sus unidades y del duty cycle [36] se puede controlar el sentido y velocidad de rotación de los motores. Los puentes L298N brindan además una tensión de salida de 5V, y esto fue utilizado para la alimentación del ESP32 conectando su pin Vin.

En el siguiente diagrama puede apreciarse el conexionado lógico para el control de los motores con los componentes mencionados.

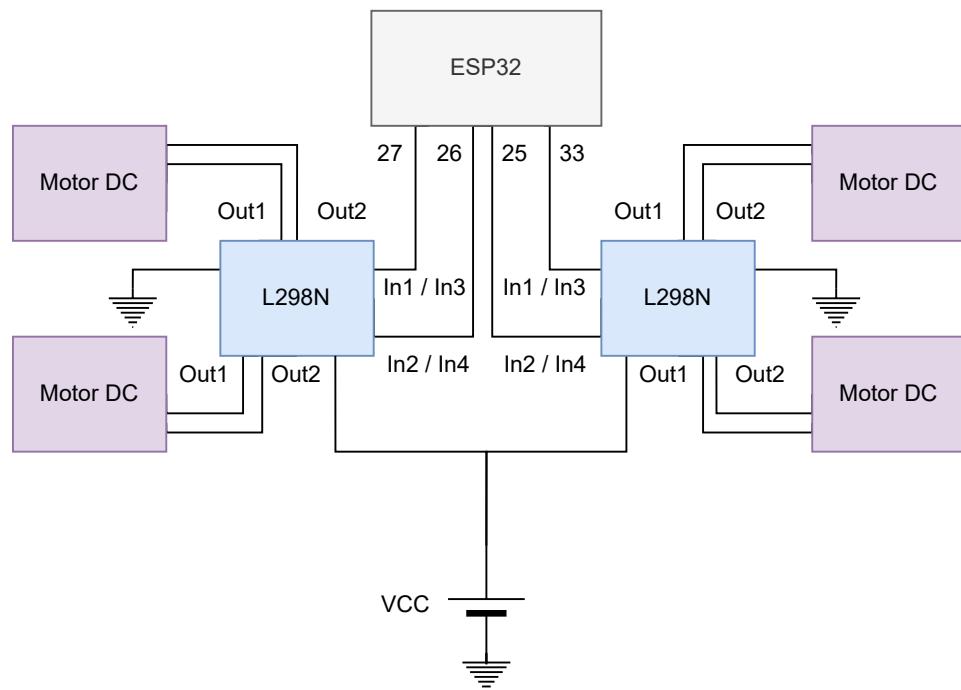


FIGURA 3.7. Conexión de motores.

El desarrollo de este módulo se basó en el ejemplo provisto por Espressif [37]. El código fuente del prototipo realizado puede apreciarse en el siguiente enlace [38].

### 3.2.7. Control del display

Para la implementación del módulo de visualización de valores observados se integró un display de dos líneas y dieciséis caracteres LCM1602A por medio de un driver I2C que facilita su control. Al basarse en el protocolo I2C el display comparte las mismas líneas SCL y SDA que el sensor BMP280.

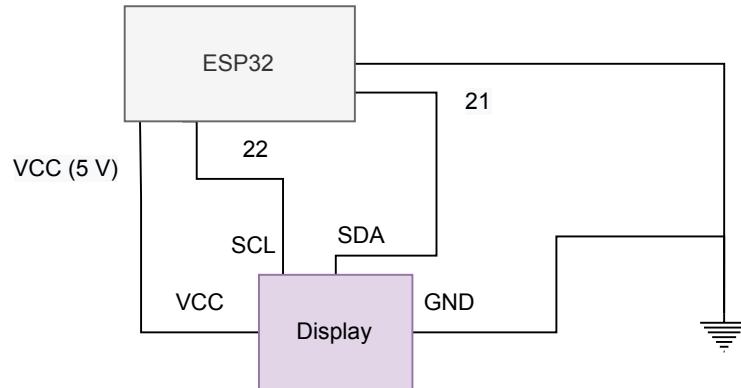


FIGURA 3.8. Conexión display.

El desarrollo de este módulo se basó en el ejemplo provisto en el enlace [39]. El código fuente del prototipo realizado puede apreciarse en el siguiente enlace [40].

### 3.3. Arquitectura de hardware

#### 3.3.1. Conexión lógico

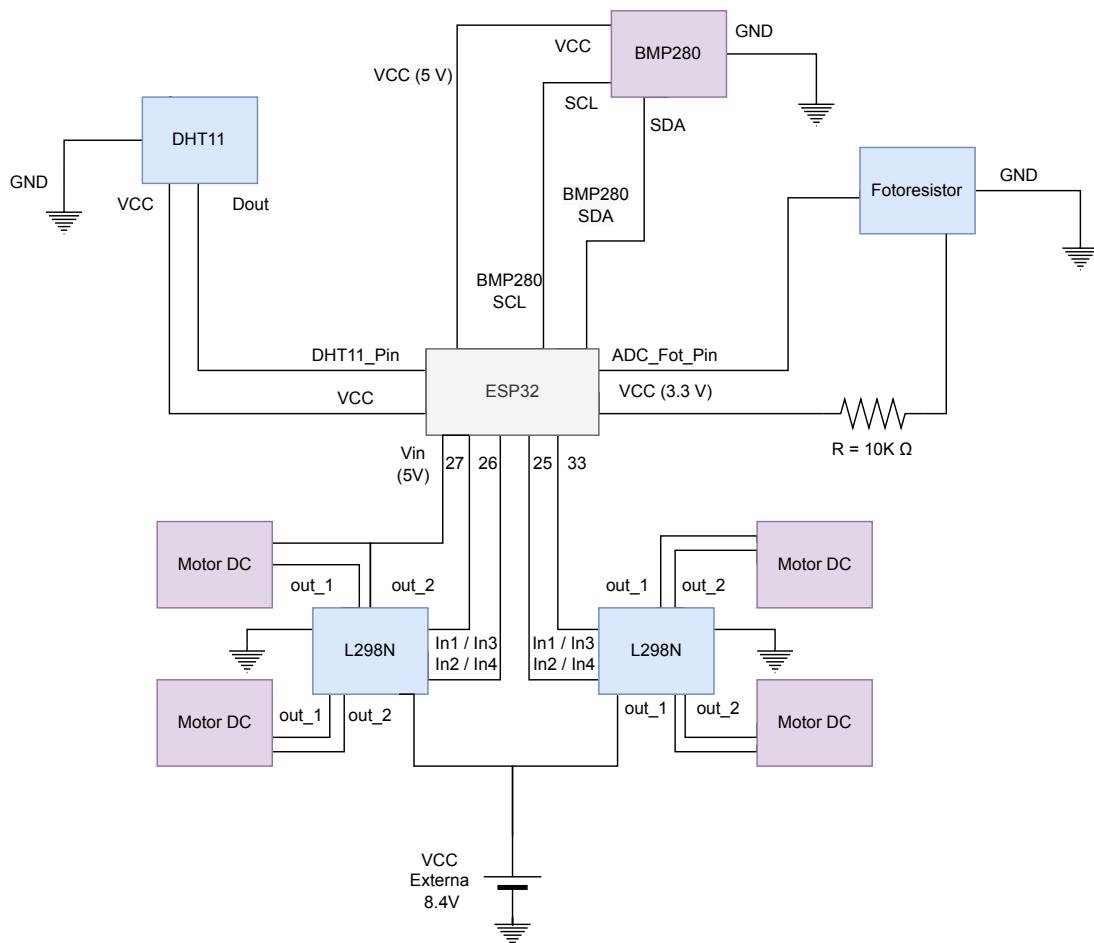


FIGURA 3.9. Conexión del robot.

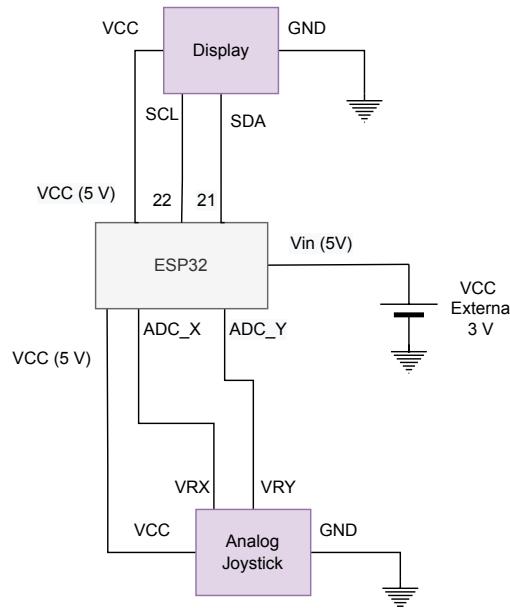


FIGURA 3.10. Conexionado del joystick.

### 3.3.2. Conexionado físico del robot

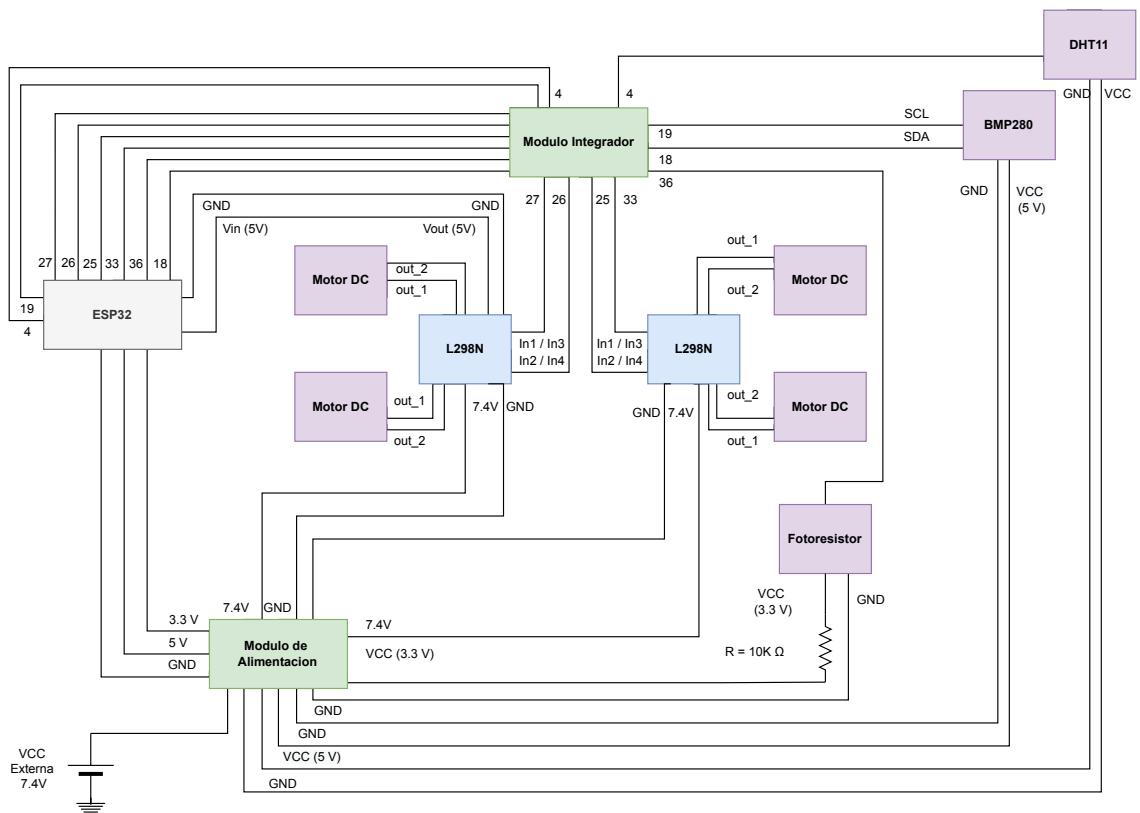


FIGURA 3.11. Conexionado físico del robot.

### **3.4. Ciclo de desarrollo**

Durante el ciclo de desarrollo se utilizaron las herramientas explicadas en el capítulo anterior, y se creó por cada prototipo una imagen Docker extendiendo la de espressif/idf [41]. El proceso de compilación y despliegue está explicado en la documentación técnica del producto [42].

## Capítulo 4

# Ensayos y resultados

Esta sección presenta los diferentes prototipos realizados para determinar la viabilidad de cada una de las funcionalidades provistas, la metodología de desarrollo, testing, y finalmente los entregables finales del trabajo.

### 4.1. Proceso de desarrollo y aseguramiento de calidad

Para el proceso de desarrollo se realizaron pruebas de concepto de las diferentes funcionalidades utilizando como materiales la bibliografía encontrada en internet, las hojas de datos y los ejemplos de código provistos por el SDK y librerías empleadas. Una vez logrado el objetivo funcional de componente se optimizó y encapsuló cada módulo para ser integrado de manera individual a un prototipo integrador sin afectar el funcionamiento de cualquier otro módulo. De esta manera se desarrolló un prototipo integrador como la sumatoria de todos los módulos de forma incremental, probándose por regresión que los módulos ya integrados previamente siguieran funcionando de forma óptima.

Una vez logrado el prototipo integrador con todas las funcionalidades de la versión v1.0 se prosiguió a la expansión en hardware del mismo para crear la versión v2.0 extrayendo los módulos de joystick y display -que serían posteriormente agregados al sistema embebido del joystick- e incorporando los módulos de conectividad UDP sobre WiFi.

Tras lograr la versión v2.0 se repite el proceso de control de calidad de los diferentes módulos ya integrados.

A continuación se detallan las diferentes pruebas realizadas

### 4.2. Verificación técnica de los diferentes módulos

Todos los módulos fueron probados mediante una inspección visual durante el proceso de pruebas de concepto.

#### 4.2.1. Verificación del módulo de joystick

Se verificó visualmente que los valores del joystick analógico puedan ser leídos apropiadamente, y que sean representativos y relevantes con la dirección del movimiento de la palanca sobre sus coordenadas X e Y.

#### **4.2.2. Verificación del módulo de control del display**

Se verificó visualmente que el display representa los caracteres programados en la prueba de concepto con una intensidad de luz aceptable para poder leerlos apropiadamente.

#### **4.2.3. Verificación del módulo de control de motores**

Se verificó visualmente que individualmente el motor pudiera girar en ambos sentidos. Luego, al implementarse los cuatro motores con sus ruedas, se probó que se pueda realizar los giros en todas las direcciones.

#### **4.2.4. Verificación del módulo de medición de temperatura y humedad**

Se verificó visualmente que los valores obtenidos por el sensor DHT11 fueran cercanos a lo esperado en relación a la temperatura en el interior del lugar de experimentación y la humedad en un valor cercano a lo reportado en Google.

#### **4.2.5. Verificación del modulo de medicion de presion atmosferica**

Se verificó visualmente que el valor obtenido por el sensor BMP280 fuera cercano a lo esperado en relación al valor reportado por Google.

#### **4.2.6. Verificación del módulo de medición de luminosidad**

Se verificó visualmente que los valores obtenidos del fotorresistor, tras ser transformados a valores absolutos porcentuales, guardan relación con el nivel de luminosidad ambiental del interior del lugar de experimentación.

#### **4.2.7. Verificación del módulo de comunicación UTP sobre WiFi**

Por medio de un dos programas UDP, uno cliente y uno servidor, se probó el establecimiento de la comunicación UDP entre dos ESP32. Luego se incorporó el servicio de comunicaciones UDP en el robot y mientras que desde el programa cliente se enviaban las acciones representando las direcciones del movimiento a realizar (FORWARD, BACKWARD, LEFT, RIGHT) y se observó visualmente como el robot giraba sus ruedas en función de los comandos enviados. Finalmente se incorporó el módulo de comunicaciones en el joystick para enviar los comandos al robot cuando se realizaba dicho

### **4.3. Pruebas funcionales del producto**

#### **4.3.1. Prueba del módulo de medición de temperatura y humedad**

Se compararon los valores medidos por el módulo de medición de temperatura y humedad basado en el sensor DHT11 con los obtenidos a través de un dispositivo de medición de temperatura y humedad. Se realizó la medición en diferentes contextos:

- En el interior de una vivienda.
- En el exterior durante el día.
- En el exterior durante la noche.

En la siguiente tabla se pueden apreciar los resultados.

TABLA 4.1. Resultados de mediciones de temperatura y humedad

Contexto	Temp. Robot	Temp. Ref.	Hume. Robot	Hume. Ref.
Interior	22.0	23.6	44.0 - 45.0	58.5
Exterior (dia)	17.0	14.0	47.0 - 53.0	62.9 - 64.0
Exterior (noche)	-	-	-	-

En el siguiente video se puede apreciar el experimento [43].

#### 4.3.2. Prueba del módulo de medición de presión

Se compararon los valores medidos por el módulo de medición de presión basado en el sensor BMP280 con los obtenidos a través de un dispositivo manómetro digital. Se realizó la medición en el interior de la vivienda dos días distintos.

En la siguiente tabla pueden apreciarse los resultados obtenidos:

TABLA 4.2. Resultados de mediciones de presión ambiental

Contexto	Presión. Robot	Presión. Ref.
Dia 1	1013	1018.9
Dia 2	1003.9	998

En el siguiente video se puede apreciar el experimento [44].

#### 4.3.3. Prueba del módulo de medición de luminosidad ambiental

Se compararon los valores medidos por el módulo de medición de luminosidad basado en un fotoresistor percibidos por el ojo humano sin utilizar ningún dispositivo de medición. Se realizó la medición en diferentes escenarios

- En interiores con luz concentrada sobre el sensor.
- En interiores con luz ambiental no concentrada.
- En interiores con a oscuras
- En el exterior de día.
- En el exterior de noche.

Los resultados mostraron que los valores porcentuales indicados por el módulo de medición de luminosidad son consistentes con los niveles de luz detectados por el ojo humano. En el siguiente video se puede apreciar el experimento [45].

#### 4.3.4. Prueba del control y desplazamiento del robot

Se verificó el control del desplazamiento del robot de forma visual por medio de accionar el joystick en las diferentes coordenadas (X;Y) y se controló que:

- la dirección del movimiento del robot sea acorde al accionamiento del joystick

- el tiempo de respuesta en el movimiento del robot y tras accionar del joystick sea mínimo, permitiendo una buena experiencia de usuario

En el siguiente video puede apreciarse dicho experimento [46].

#### **4.3.5. Prueba del módulo de visualización de display**

Se verificó el funcionamiento del display visualizando las lecturas de los valores censados y transmitidos por el robot. Se controló que:

- las lecturas sean nítidas y entendibles
- las unidades de medida están presentes
- haya un detalle de lo que se está midiendo acompañando las lecturas y la unidad de medida
- el nivel de luminosidad sea óptimo para permitir la lectura independiente-mente de la iluminación ambiental
- se presentan las lecturas de todos los valores observados

En el siguiente video se puede apreciar el experimento [47].

### **4.4. Reportes de testing**

...

### **4.5. Documentación del producto**

Se desarrolló la documentación del producto compuesta de los siguientes entre-gables

- Documentación técnica [42].
- Manual de usuario [48].

## Capítulo 5

# Conclusiones

Conclusiones del trabajo...

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se puedo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

### 5.1. Próximos pasos

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



# Bibliografía

- [1] Latam Mining. *Robots y minería: Gobierno argentino quiere implementarlos.* URL: <https://www.latam-mining.com/robots-y-mineria-gobierno-argentino-quiere-implementarlos/>.
- [2] Diario de Cuyo. *Gobierno pone la mira en el desarrollo de robots para la actividad minera.* URL: <https://www.diariodecuyo.com.ar/politica/Gobierno-pone-la-mira-en-el-desarrollo-de-robots-para-la-actividad-minera-20200202-0052.html>.
- [3] Universidad Nacional de San Juan. *Robots en la minería.* URL: [http://www.unsj.edu.ar/home/noticias\\_detalles/4810/1](http://www.unsj.edu.ar/home/noticias_detalles/4810/1).
- [4] Ing. Nelson Dario García Hurtado e Ing. Melvin Andrés González Pino. *Robot de exploración terrestre Geobot.* URL: [https://www.unipamplona.edu.co/unipamplona/portalIG/home\\_40/recursos/01\\_general/revista\\_1/09102011/v01\\_09.pdf](https://www.unipamplona.edu.co/unipamplona/portalIG/home_40/recursos/01_general/revista_1/09102011/v01_09.pdf).
- [5] Ing. Hernán L. Helguero Velásquez1 e Ing. Rubén Medinaceli Tórrez. *Robot Minero: Sistema Detector de Gases utilizando Sensores en Tiempo Real MIN – SIS 1.0 SDG-STR.* URL: [http://www.scielo.org.bo/scielo.php?script=sci\\_arttext&pid=S2519-53522020000100003](http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S2519-53522020000100003).
- [6] Boston Dynamics. *Spot.* URL: <https://www.bostondynamics.com/products/spot>.
- [7] Waygate Technologies. *BIKE - An advanced crawler robot for remote visual inspection.* URL: <https://www.bakerhughes.com/waygate-technologies/robotic-inspection/bike>.
- [8] Espressif. *ESP32.* URL: <https://www.espressif.com/en/products/socs/esp32>.
- [9] Espressif. *ESP32-WROOM-32D Datasheet.* URL: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d\\_esp32-wroom-32u\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32d_esp32-wroom-32u_datasheet_en.pdf).
- [10] Mouser. *DHT11 datasheet.* URL: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>.
- [11] Bosch. *BMP280 datasheet.* URL: <https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>.
- [12] Handson Technology. *PS2 Joy Stick for Arduino/Raspberry.* URL: <http://www.handsontec.com/dataspecs/accessory/PS2-Joystick.pdf>.
- [13] Espressif. *Analog to Digital Converter (ADC).* URL: <https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html>.
- [14] Handson Technology. *I2C Serial Interface 1602 LCD Module.* URL: [http://www.handsontec.com/dataspecs/module/I2C\\_1602\\_LCD.pdf](http://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf).
- [15] Adafruit. *DC Gearbox Motor - TT Motor -200RPM - 3 to 6VDC.* URL: [https://media.digikey.com/pdf/Data%20Sheets/Adafruit%20PDFs/3777\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/Adafruit%20PDFs/3777_Web.pdf).

- [16] CMake. *CMake*. URL: <https://cmake.org/>.
- [17] CMake. *Ninja*. URL: <https://cmake.org/cmake/help/latest/generator/Ninja.html>.
- [18] Espressif. *ESP-IDF Programming Guide | Get Started*. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/>.
- [19] Readthedocs by Ruslan V. Uss. *ESP-IDF Components library*. URL: <https://esp-idf-lib.readthedocs.io/en/latest/>.
- [20] Espressif Programing Guide. *ESP-IDF Get Started*. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>.
- [21] Docker. *Docker*. URL: <https://docker.com/>.
- [22] Visualstudio. *Visualstudio Code*. URL: <https://code.visualstudio.com/>.
- [23] Ubuntu. *Ubuntu*. URL: <https://ubuntu.com/>.
- [24] Espressif. *ESP-IDF WiFi*. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/wifi.html>.
- [25] Espressif. *ESP-IDF - Wi-Fi SoftAP Example*. URL: [https://github.com/espressif/esp-idf/tree/v4.4/examples/wifi/getting\\_started/softAP](https://github.com/espressif/esp-idf/tree/v4.4/examples/wifi/getting_started/softAP).
- [26] Gonzalo Carreno. *POC ESP32-WiFi v4.4*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/tree/main/pocs/esp32-wifi-ap-v4.4](https://github.com/kronleuchter85/cese_proyecto_especializacion/tree/main/pocs/esp32-wifi-ap-v4.4).
- [27] Espressif. *ESP-IDF ADC1 Example*. URL: <https://github.com/espressif/esp-idf/tree/v4.0.3/examples/peripherals/adc>.
- [28] Gonzalo Carreno. *POC ESP32-joystick*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/tree/main/pocs/esp32-joystick](https://github.com/kronleuchter85/cese_proyecto_especializacion/tree/main/pocs/esp32-joystick).
- [29] Gonzalo Carreno. *POC ESP32-photoresistor*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/tree/main/pocs/esp32-photoresistor](https://github.com/kronleuchter85/cese_proyecto_especializacion/tree/main/pocs/esp32-photoresistor).
- [30] UncleRus. *ESP32 - Example for dht driver*. URL: <https://github.com/UncleRus/esp-idf-lib/tree/master/examples/dht/default>.
- [31] Gonzalo Carreno. *POC ESP32-DHT11*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/tree/main/pocs/esp32-dht11](https://github.com/kronleuchter85/cese_proyecto_especializacion/tree/main/pocs/esp32-dht11).
- [32] UncleRus. *ESP32 - Example for bmp280 driver*. URL: <https://github.com/UncleRus/esp-idf-lib/tree/master/examples/bmp280/default>.
- [33] Gonzalo Carreno. *POC ESP32-BMP280*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/tree/main/pocs/esp32-bmp280](https://github.com/kronleuchter85/cese_proyecto_especializacion/tree/main/pocs/esp32-bmp280).
- [34] Espressif. *L298N Dual H-Bridge Motor Driver*. URL: <https://www.handsontec.com/datasheets/L298N%20Motor%20Driver.pdf>.
- [35] Espressif. *ESP-IDF Motor Control Pulse Width Modulator (MCPWM)*. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/mcpwm.html>.
- [36] Espressif. *MCPWM*. URL: <https://docs.espressif.com/projects/esp-idf/en/v4.2/esp32/api-reference/peripherals/mcpwm.html>.
- [37] Espressif. *MCPWM brushed dc motor control Example*. URL: [https://github.com/espressif/esp-idf/tree/v4.2/examples/peripherals/mcpwm/mcpwm\\_brushed\\_dc\\_control](https://github.com/espressif/esp-idf/tree/v4.2/examples/peripherals/mcpwm/mcpwm_brushed_dc_control).

- [38] Gonzalo Carreno. *POC ESP32-motor-pwm*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/tree/main/pocs/esp32-motor-pwm](https://github.com/kronleuchter85/cese_proyecto_especializacion/tree/main/pocs/esp32-motor-pwm).
- [39] ESP32 Tutoriales. *ESP32 I2C LCD with ESP-IDF*. URL: <https://esp32tutorials.com/i2c-lcd-esp32-esp-idf/>.
- [40] Gonzalo Carreno. *POC ESP32-DHT11*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/tree/main/pocs/esp32-display](https://github.com/kronleuchter85/cese_proyecto_especializacion/tree/main/pocs/esp32-display).
- [41] Espressif. *Espressif Docker Image*. URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/tools/idf-docker-image.html>.
- [42] Gonzalo Carreno. *Robot de exploracion ambiental - Documentacion Tecnica*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/blob/main/docs/Documentacion-Tecnica.pdf](https://github.com/kronleuchter85/cese_proyecto_especializacion/blob/main/docs/Documentacion-Tecnica.pdf).
- [43] Gonzalo Carreno. *Robot de exploracion ambiental - Probando control de desplazamiento*. URL: <https://prueba-display-resultados>.
- [44] Gonzalo Carreno. *Robot de exploracion ambiental - Probando control de desplazamiento*. URL: <https://prueba-display-resultados>.
- [45] Gonzalo Carreno. *Robot de exploracion ambiental - Probando control de desplazamiento*. URL: <https://prueba-display-resultados>.
- [46] Gonzalo Carreno. *Robot de exploracion ambiental - Probando control de desplazamiento*. URL: <https://prueba-desplazamiento>.
- [47] Gonzalo Carreno. *Robot de exploracion ambiental - Probando control de desplazamiento*. URL: <https://prueba-display-resultados>.
- [48] Gonzalo Carreno. *Robot de exploracion ambiental - Manual de usuario*. URL: [https://github.com/kronleuchter85/cese\\_proyecto\\_especializacion/blob/main/docs/Manual-De-Usuario-vFinal.pdf](https://github.com/kronleuchter85/cese_proyecto_especializacion/blob/main/docs/Manual-De-Usuario-vFinal.pdf).