

UNIVERSIDAD NACIONAL
DE INGENIERÍA
FACULTAD DE CIENCIAS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN

CC221 – ARQUITECTURA DE COMPUTADORES
SECCIÓN: B



PROYECTO FORMATIVO
Dispositivo con aplicativo móvil para
el monitoreo remoto de un cultivo

Integrante	Escuela	Código
Muñoz Diaz, Jose Estalin	Ciencia de la Computación	20222718J
Villalobos Palomino, Jarem Alexssander	Ciencia de la Computación	20234159K
Guevara Tay, Gabriel Rodrigo	Ciencia de la Computación	20221389B
Taipe Cotrina, Álvaro Jesús	Ciencia de la Computación	20221565E

Docente: Cruz Salazar, César Martín
Fecha: 28 de noviembre de 2024

Índice

1. Introducción	2
2. Objetivos	2
2.1. Objetivo general	2
2.2. Objetivos Específicos	2
3. Fundamento teórico	2
3.1. Hardware	2
3.1.1. Microcontrolador	2
3.1.2. Sensor BME280	5
3.1.3. Sensor MQ135	6
3.1.4. Sensor ECH2O EC-5	7
3.2. Software	8
3.2.1. Kotlin	8
3.2.2. Android Studio	8
4. Hoja de ruta	9
4.1. Configuración de Hardware	9
4.1.0. Recursos a usar	9
4.1.1. Diagrama de Conexiones	9
4.1.2. Soporte del ESP32 en Arduino IDE	10
4.1.3. Conexión con Firebase	10
4.2. Programación de aplicación	10
4.2.0. Recursos a usar	10
4.2.1. Estructura de la aplicación	11
4.2.2. Visualización de Datos y Gráficos : Gráfica en tiempo real	13
5. Conclusiones	14

1. Introducción

El proceso de cultivar plantas en un entorno controlado requiere la observación y moderación de múltiples condiciones ambientales necesarias para el crecimiento de las plantas de interés. El presente proyecto está orientado a facilitar la labor de registrar e inspeccionar indicadores cuantitativos que corresponden con la situación de un ambiente de cultivo.

En su centro se encuentra un microcontrolador ESP32^{devkitv4} que, en conjunto con sensores colocados en el ambiente a monitorear, constituye un bloque fundacional en el monitoreo de un sitio de cultivo. Además, sus capacidades de comunicación inalámbrica permiten integrar la recolección de datos con servicios en la nube sobre los cuales es posible construir un sistema de alertas frente a posibles condiciones inadecuadas.

El presente documento describe el trabajo realizado según las ideas mencionadas.

2. Objetivos

2.1. Objetivo general

Facilitar el monitoreo de las condiciones ambientales en un sitio de cultivo según las necesidades de una planta de interés.

2.2. Objetivos Específicos

- Fabricar un dispositivo de observación que mida indicadores cuantitativos útiles en el lugar de cultivo.
- Recolectar las observaciones en una base de datos.
- Conectar los datos reunidos con un aplicativo móvil que permita comparar las observaciones más recientes con sus anteriores, así como con perfiles de condiciones necesarias para el cultivo.

3. Fundamento teórico

3.1. Hardware

3.1.1. Microcontrolador

ESP32 es una familia de microcontroladores que pertenece a Espressif caracterizada por su variedad de módulos integrados orientados a telecomunicaciones por Bluetooth y Wi-Fi. ^{ESP32_espressif}

La serie ESP32-DevKitC, en particular, es notable por tener una abundancia de pines expuestos para facilitar su conexión y uso. El presente prototipo utiliza un microcontrolador ESP32-DevKitC V4, mismo que tiene integrados un botón de reinicio, así como un puerto Micro-USB conectado a un puente USB-a-UART^{devkitv4}. Una imagen del microcontrolador con detalles anotados es provista en la figura 1.

Entre sus componentes, son de interés un módulo ESP32-WROOM-32 integrado con un microprocesador ESP-D0WDQ6, una interfaz serial de periféricos con 4 MB de memoria flash integrada, una antena y un oscilador de cristal con frecuencia 40 MHz,^{esp32wroom32doc} tal como se puede observar en la figura 2.

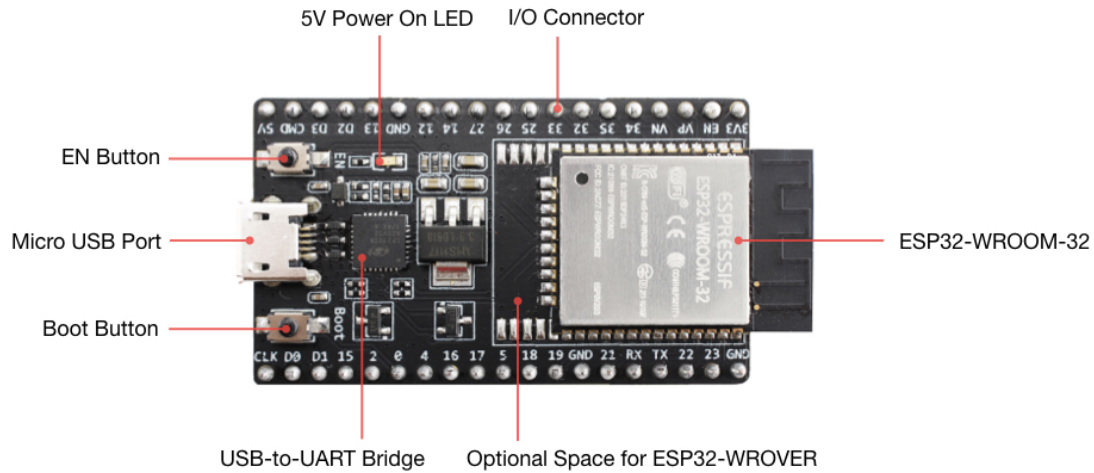


Figura 1: Descripción del ESP32-DevKitC V4^{devkitv4}

Componente	Descripción
ESP32-WROOM-32	Módulo con el ESP32.
EN	Botón de reinicio.
Boot	Botón de descarga. Mantenerlo presionado y luego presionar <i>EN</i> inicia el <i>Modo de descarga de firmware</i> , que opera a través del puerto serial.
USB-to-UART Bridge	Un chip de puente USB-UART. Provee velocidades de transferencia de hasta 3 MB s^{-1} .
Micro USB Port	Interfaz USB. Provee energía a la placa y puede comunicar a una computadora con el módulo ESP32-WROOM-32.
5V Power On LED	Se enciende cuando está conectado a la placa un suministro de energía de 5 V externo o por USB.
I/O	La mayoría de los pines del módulo ESP están conectados con los que sobresalen de la placa. Pueden poseer funcionalidad I2C, SPI, etc.

Cuadro 1: ESP32-DevKitC V4 con el módulo ESP32-WROOM-32^{devkitv4}

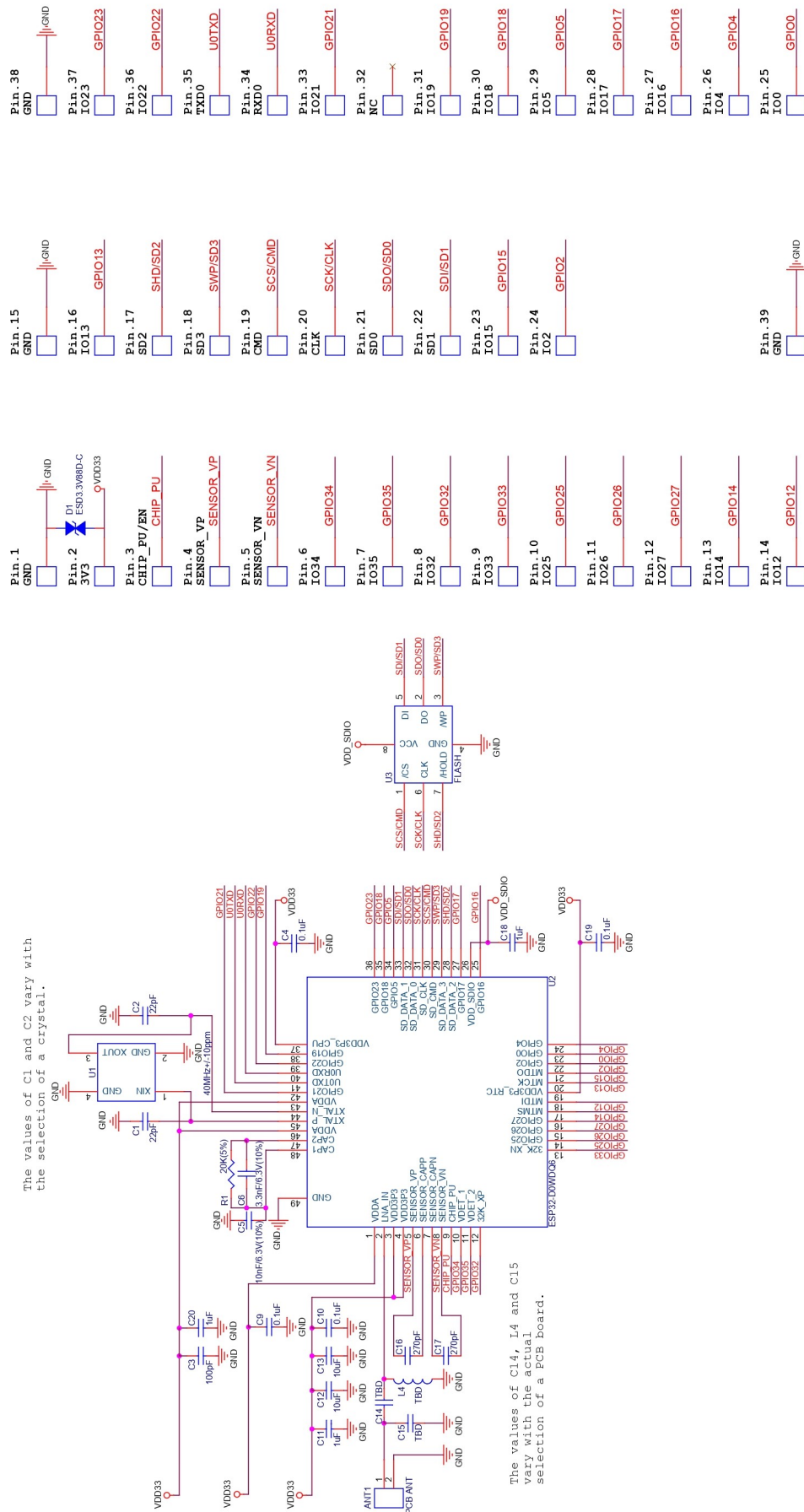


Figura 2: Esquema del ESP32-WROOM-32esp32wroom32doc

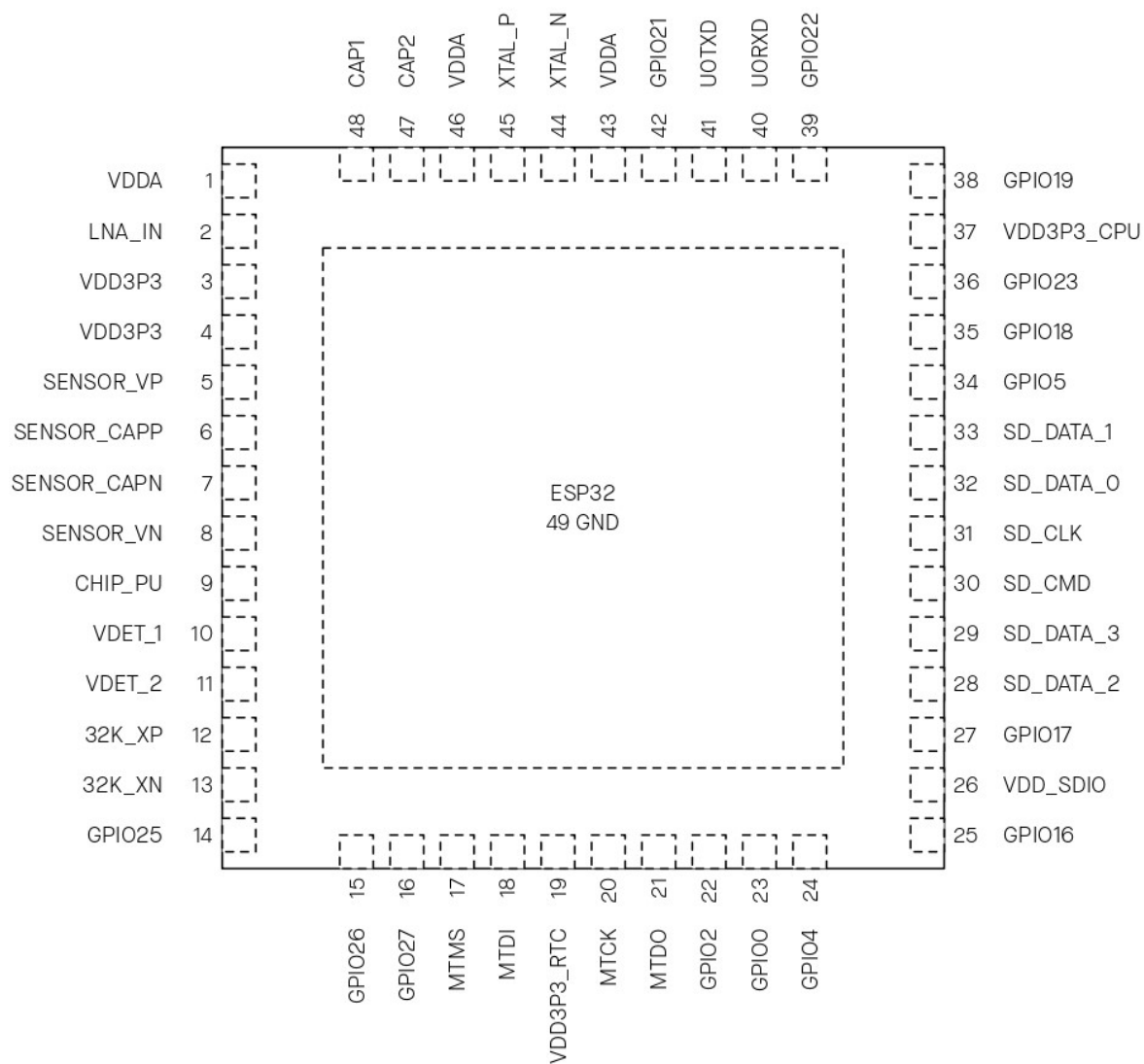


Figura 3: Diagrama de pines del ESP32-D0WDQ6^{esp32d0wdq62doc}

3.1.2. Sensor BME280

El sensor BOSCH-BME280 es capaz de medir humedad relativa, presión barométrica y temperatura ambiente.^{boschbme280descr} Algunas características se encuentran en la [tabla 2](#).

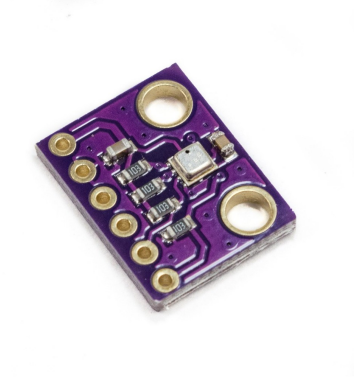


Figura 4: Sensor BME280^{bme280image}

Rango de operación	Presión: 300 a 1100 hPa Temperatura: -40 a 85°C
Voltaje de entrada V_{DD}	1,71 a 3,6 V
Voltaje de entrada V_{DDIO}	1,2 a 3,6 V
Interfaz	I ² C y SPI
Consumo de corriente	1,8 μA @ 1 Hz (humedad, temperatura) 2,8 μA @ 1 Hz (presión, temperatura) 3,6 μA @ 1 Hz (humedad, presión, temperatura) 0,1 μA (inactivo)
Sensor de humedad	
Tiempo de respuesta	1 s
Tolerancia de precisión	$\pm 3\%$ humedad relativa
Histéresis	$\leq 2\%$ humedad relativa
Sensor de presión	
Ruido RMS	0,2 Pa
Error de sensibilidad	$\pm 0,25\%$
Coefficiente de temperatura ajustado	$\pm 1,5 \frac{\text{Pa}}{\text{K}}$
Base	LGA de 8 pines con tapa de metal
Dimensiones	$2,5 \times 2,5 \times 0,93 \text{ mm}^3$

Cuadro 2: Información técnica del BME280^{boschbme280techdata}

3.1.3. Sensor MQ135

El sensor MQ135 mide la calidad del aire utilizando SnO_2 , cuya conductividad varía cuando entra en contacto con impurezas en el aire.^{mq135winson} En la tabla 3 son mencionadas algunas características del sensor.



Figura 5: Sensor MQ-135^{mq135image}

Parámetro	Condición
Condiciones de trabajo	
Voltaje en circuito	$(5,0 \pm 0,1) \text{ V AC o DC}$
Voltaje de calentamiento	$(5,0 \pm 0,1) \text{ V AC o DC}$
Resistencia de calentador	$33 \Omega \pm 5 \%$
Condiciones de ambiente	
Temperatura de uso	$-10 \text{ a } 45 ^\circ\text{C}$
Humedad relativa permitida	$< 95 \%$
Valores de concentración de oxígeno permitidos	21 % en condiciones normales, mínimo de $> 2 \%$

Cuadro 3: Condiciones de trabajo normales y de ambiente permitidos del MQ-135^{mq135hanwei}

3.1.4. Sensor ECH2O EC-5

El sensor EC-5 mide la humedad del suelo mediante la resistencia entre dos electrodos, la cual varía con proporcionalidad inversa respecto a la humedad del suelo cuando el sensor está inserto en él.^{ec5humedadsuelosensor} Algunas especificaciones pueden encontrarse en la tabla 4



Figura 6: Sensor Decagon EC-5^{ec5image}

Parámetro	Valor
Tiempo de medición	10 ms
Precisión	$\pm 2\%$
Requisitos de energía	2,5 a 3,6 V _{DC} @ 10 mA
Rango de temperaturas	-40 a 60 °C
Rango de medición	De 0 hasta saturarse

Cuadro 4: Especificaciones técnicas del ECH2O EC-5^{ec5humedadsuelosensor}

3.2. Software

3.2.1. Kotlin

Kotlin es un lenguaje de programación desarrollado por JetBrains que tiene soporte oficial de Google para desarrollo de aplicativos en Android. Por defecto es ejecutado en la Máquina Virtual de Java, por lo que la interoperabilidad Kotlin-Java es alta.^{JetBrains}

Kotlin es considerado estable y moderno en comparación con Java, un lenguaje conocido por complicaciones de retrocompatibilidad, por lo que es una opción preferible para desarrollar aplicativos en la plataforma Android.^{Moskala_Wojda_2017b}

3.2.2. Android Studio

Android Studio es el entorno de desarrollo integrado oficial utilizado en el desarrollo de aplicativos para Android. Su calidad tiene relación con los componentes que comparte con el ampliamente reconocido editor IntelliJ IDEA, también desarrollado por JetBrains. Entre sus herramientas más críticas, destaca el acceso a servicios y herramientas que provee Google.^{android_developers}

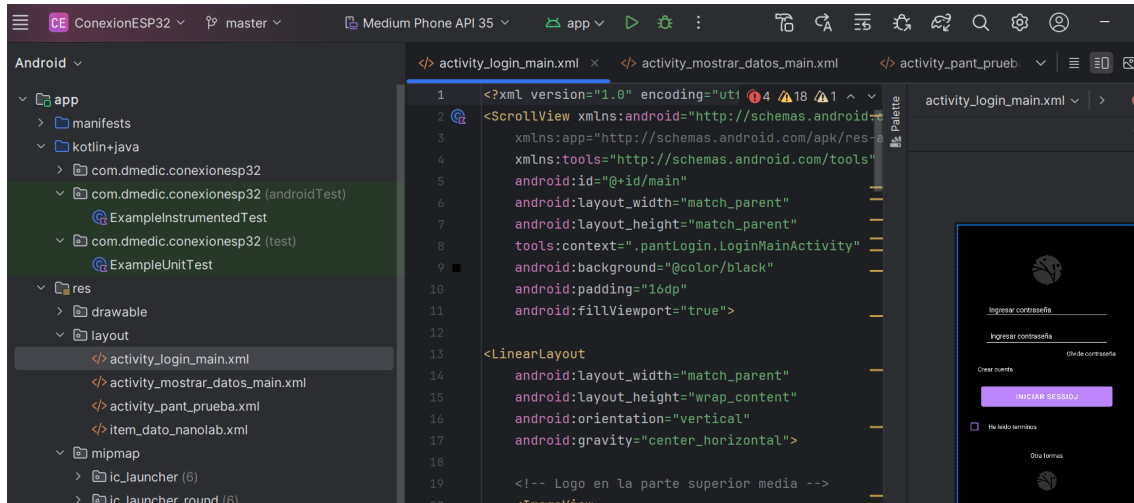


Figura 7: Ejemplo de programa en Android Studio

4. Hoja de ruta

4.1. Configuración de Hardware

4.1.0. Recursos a usar

- ESP32: Controlador principal con conectividad Wi-Fi.
- Sensor BME280: Sensor para medir temperatura, humedad y presión.
- Sensor MQ135: Sensor para la detección de la concentración de CO2 en el aire.
- Sensor de Humedad de Suelo: Sensor analógico para medir el nivel de humedad del suelo.
- Protoboard y Jumpers: Para facilitar las conexiones de prueba.
- Fuente de Alimentación: Batería o adaptador USB de 5V.
- Cables Dupont: Para realizar las conexiones entre los sensores y el ESP32.

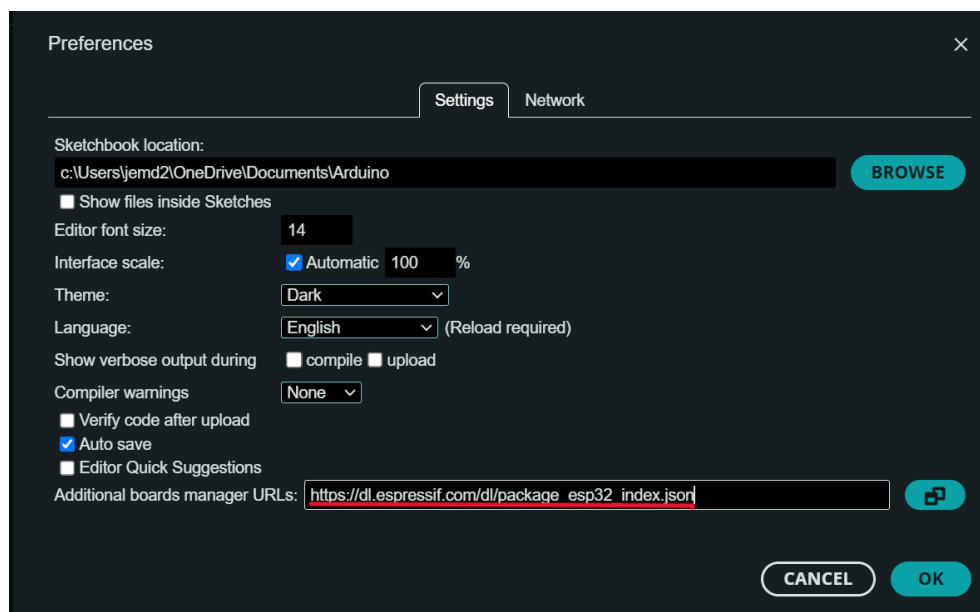
4.1.1. Diagrama de Conexiones

- ESP32: Este será el controlador principal y estará encargado de leer los datos de cada sensor.
- BME280: Este sensor utiliza la comunicación I2C para conectarse con el ESP32. Las conexiones necesarias son:
 1. VCC a 3,3V del ESP32.
 2. GND a GND del ESP32.
 3. SCL al pin D22 del ESP32.
 4. SDA al pin D21 del ESP32.

- MQ135: Para el sensor MQ135, se utilizará la salida analógica para la detección de CO₂.
 1. VCC a 5 V del ESP32.
 2. GND a GND del ESP32.
 3. AOUT a un pin analógico del ESP32.
- Sensor de Humedad de Suelo: El sensor de humedad de suelo puede ser analógico o digital. Se optará por la salida analógica para una medición más precisa.
 1. VCC a 3,3 V del ESP32.
 2. GND a GND del ESP32.
 3. AOUT a un pin analógico del ESP32.

4.1.2. Soporte del ESP32 en Arduino IDE

Para poder programar en ESP32 con Arduino, es necesario agregar un URL en el gestor de placas lo siguiente: https://dl.espressif.com/dl/package_esp32_index.json



4.1.3. Conexión con Firebase

Para realizar una conexión a Firebase desde tu ESP32, es utilizada la biblioteca Firebase ESP32 Client.

4.2. Programación de aplicación

4.2.0. Recursos a usar

- Kotlin: lenguaje principal de desarrollo ¹

¹<https://kotlinlang.org/>

- Firebase Realtime Database: almacenamiento de observaciones y perfiles ²
- Androidplot: biblioteca para la creación de gráficos ³

4.2.1. Estructura de la aplicación

La aplicación está estructurada 4 actividades principales:

- Actividad de visualización de datos y gráficas (MainActivity) : Se muestran los datos recibidos por la placa y su interpretación respecto al perfil elegido por el usuario.
- Actividad de elección de perfiles (ProfileActivity) : Se muestran todos los perfiles disponibles para su selección en **MainActivity**.
- Actividad de modificación de perfiles (DisplayProfileActivity): Se muestra todos los datos del perfil seleccionado, así como la opción de modificarlo o eliminarlo.
- Actividad de creación de perfiles (NewProfileActivity): Se muestra un campo para poder crear un nuevo perfil existente.

En la aplicación, se han usado dos estructuras de datos para abstraer los datos de los sensores y de los perfiles

Clase Profile

```

1 package com.mypackage.esppapplication.models
2
3 import android.os.Parcelable
4 import kotlinx.parcelize.Parcelize
5
6 @Parcelize
7 data class Parameter(
8     var maxValue : Double = 0.0,
9     var minValue : Double = 0.0
10 ) : Parcelable
11
12 @Parcelize
13 data class Profile(
14     var id : String? = null,
15     var name : String = "defaultProfile",
16     var alcohol : Parameter = Parameter(),
17     var altitud : Parameter = Parameter(),
18     var co2 : Parameter = Parameter(),
19     var humedadesuelo : Parameter = Parameter(),
20     var presion : Parameter = Parameter(),
21     var temperatura : Parameter = Parameter()) : Parcelable

```

Clase SensorValues

```

1 package com.mypackage.esppapplication.models
2
3 class SensorValues (
4     val alcohol : Double = 0.0,
5     val altitud : Double = 0.0,
6     val co2 : Double = 0.0,
7     val estado : Int = 1,
8     val humedadesuelo : Double = 0.0,
9     val presion : Double = 0.0,
10     val temperatura : Double = 0.0)

```

²<https://firebase.google.com/>

³<https://github.com/halfhp/androidplot>

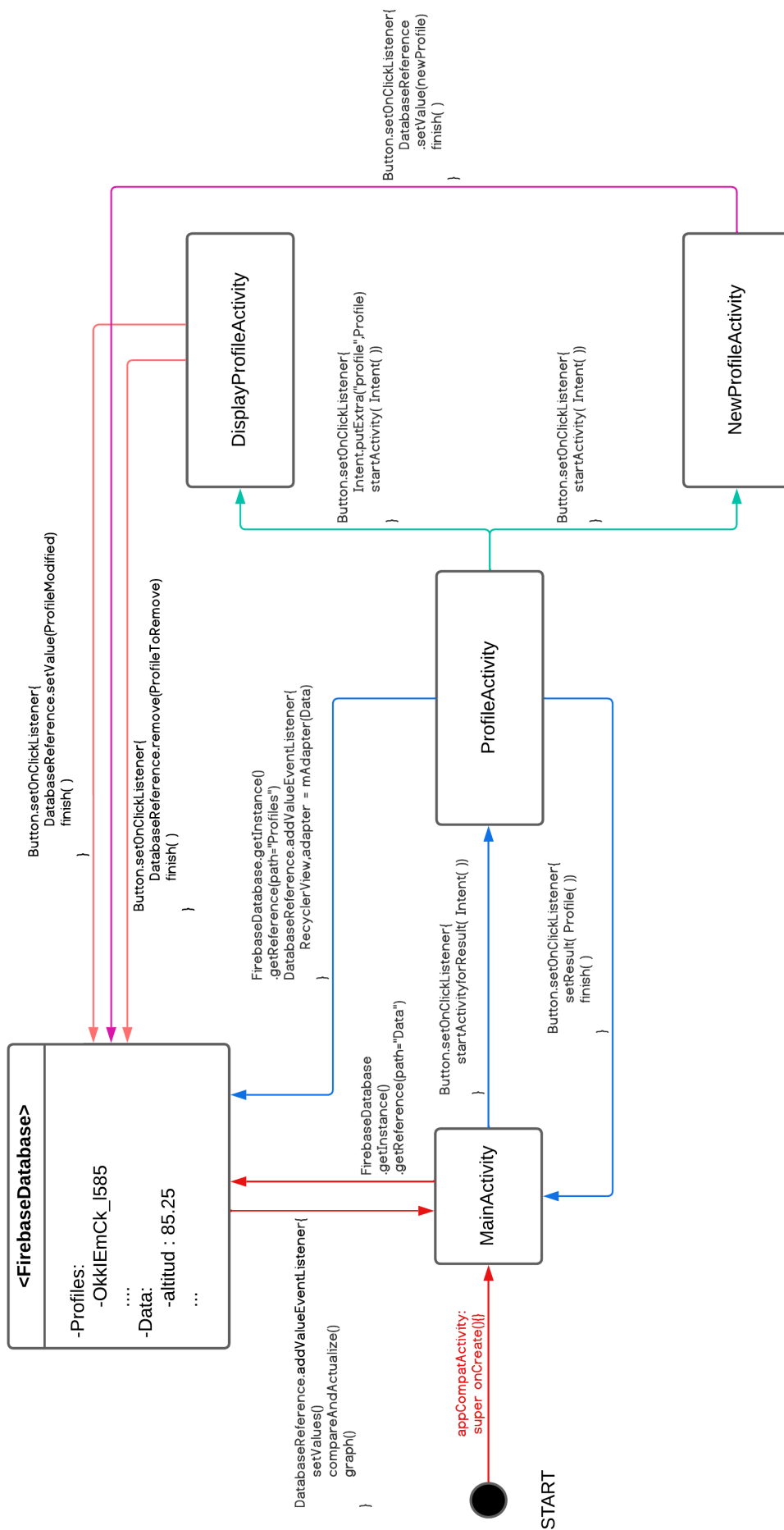


Figura 8: Diagrama de funcionamiento de la aplicación

4.2.2. Visualización de Datos y Gráficos : Gráfica en tiempo real

La visualización de datos y gráficos se realiza de manera dinámica en la actividad principal de la aplicación. Los parámetros que se visualizan en la aplicación son:

- Alcohol
- Altitud
- CO_2
- Humedad
- Presión
- Temperatura

La generación de gráficos se realizará mediante la librería *AndroidPlot*, utilizando un objeto *XYPlot* para visualizar la información enviada por la placa, cada vez que ésta se conecte a la base de datos.

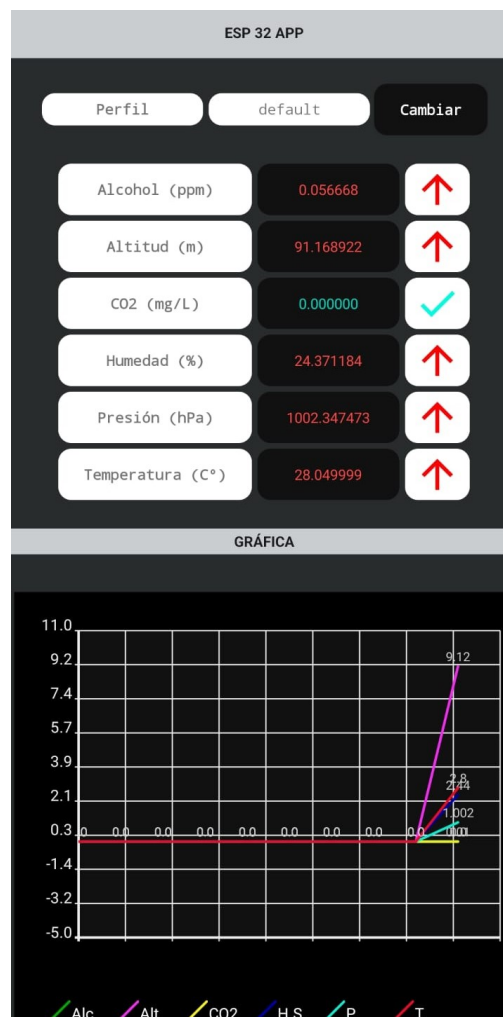


Figura 9: Visualización de parámetros y gráfico en tiempo real

5. Conclusiones

Las medidas de interés han sido recogidas con éxito durante las pruebas efectuadas en el desarrollo del dispositivo de observación. Asimismo, el uso de servicios en la nube para almacenar las observaciones obtenidas ha sido extendido con la capacidad de almacenar los perfiles objetivo, por lo que dicha información puede ser solicitada sin depender de un solo dispositivo de observación o un solo dispositivo de usuario.

El aplicativo móvil, aún en una etapa temprana de desarrollo, ha demostrado su utilidad en comunicar al usuario la comparación entre las condiciones actuales y las del perfil objetivo, así como en ilustrar los cambios en las medidas a través del tiempo y en proveer soporte para el registro y uso de distintos perfiles de monitoreo. El prototipo desarrollado hasta la fecha del presente documento cumple su rol como herramienta que transmite a su usuario una perspectiva cuantitativa actualizada de las condiciones ambientales en un sitio de cultivo.