

# **PROYECTO FINAL**

**Sistema de control y riego  
automatizado de plantas**

**Martin Castro - LU 92179**

**Juan Comellas - LU 105160**

# Índice

<b>Introducción</b>	<b>3</b>
<b>Diseño y desarrollo</b>	<b>3</b>
Requerimientos:	3
Hardware y componentes utilizados	4
Conexión con internet	4
Conexión con firebase	4
Calibración de sensores	5
Parámetros y toma de datos	5
Página Web	6
<b>Funcionamiento</b>	<b>8</b>
<b>Complicaciones presentadas</b>	<b>1</b>
<b>Posibles mejoras</b>	<b>1</b>
<b>Conclusiones</b>	<b>2</b>

# Introducción

Actualmente países del primer mundo donde las condiciones climáticas no son favorables para el cultivo tradicional de plantas utilizan invernaderos verticales. Estos sistemas basados en el cultivo por aeroponía reducen hasta un 95% el consumo de recursos hídricos.

A partir de esta idea surgió la posibilidad de la implementación de un sistema de cuidado de plantas, donde mediante la medición de la humedad del suelo se pueda automatizar el riego. A su vez, como las plantas requieren de iluminación para su crecimiento es necesario contar con la misma incluso en momentos de poca claridad, por lo que se incorporó al mismo un sistema de iluminación..

Por otro lado no solo es de importancia para las plantas crecer con suficiente iluminación y humedad en el suelo, sino también que las distintas especies toleran distintos ambientes; por ejemplo la planta de maracuyá se cultiva a temperaturas entre 23°C y 28°C, con humedades en torno al 65-70%<sup>1</sup>, por lo cual las condiciones climáticas son necesarias medirlas.

Para llevar un control del cuidado de las plantas es importante saber las condiciones de las mismas, en el momento actual o hasta es interesante llevar un registro. Para esto se debe tener una comunicación entre los parámetros medidos y un servidor.

# Diseño y desarrollo

Antes de realizar cualquier tipo de desarrollo, realizamos un breve análisis de requerimientos del sistema. En este análisis, definimos el comportamiento esperado del proyecto y los componentes de hardware necesarios para su desarrollo.

## Requerimientos:

- Desarrollo de un sistema de riego e iluminación automatizado.
- Recopilación de métricas como la humedad del suelo, humedad del ambiente y temperatura para permitir al usuario llevar un control del estado de la planta.
- Desarrollar un sistema web que permita la visualización de toda la información recopilada por el sistema y permita setear parámetros al mismo.
- Desarrollar una interfaz gráfica amigable con el usuario.
- Utilizar una fuente de energía accesible para el usuario y que permita el funcionamiento del sistema de manera autónoma e indefinida.
- Utilizar Firebase como base de datos en tiempo real.

---

<sup>1</sup> [https://inta.gob.ar/sites/default/files/cartilla\\_maracuya\\_0.pdf](https://inta.gob.ar/sites/default/files/cartilla_maracuya_0.pdf)

## Hardware y componentes utilizados

- Un módulo ESP32 con conectividad WiFi y Bluetooth para el cual se desarrolló el sistema.
- Un sensor HL69 para medir la humedad del suelo.
- DTH11 para temperatura y humedad en el ambiente.
- Un Fotorresistor para tomar la luminosidad del ambiente.
- Una bomba de agua para poder regar las plantas.
- Una lámpara para poder iluminar las plantas en caso de ser necesario.
- Dos relé para poder encender la bomba de agua y la lámpara cuando sea necesario.

## Conexión con internet

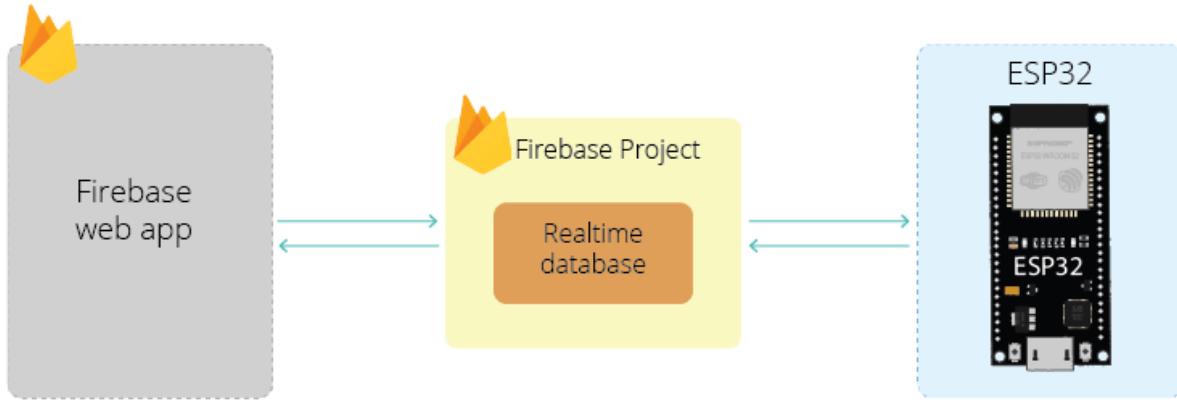
Una vez definidos los componentes a utilizar y los requerimientos del sistema, se comenzó el desarrollo para permitir la conexión del módulo ESP32 a internet y la manipulación de datos en firebase.

Para lograr comunicarse con internet se utilizó la librería *WiFi.h*, la cual permite mediante la sentencia *WiFi.begin(nombre\_red, contraseña)* conectarse con una red WiFi cercana para obtener una dirección IP y comenzar la comunicación a través de internet.

```
20:33:50.938 -> Conectado con la IP: 192.168.1.108
```

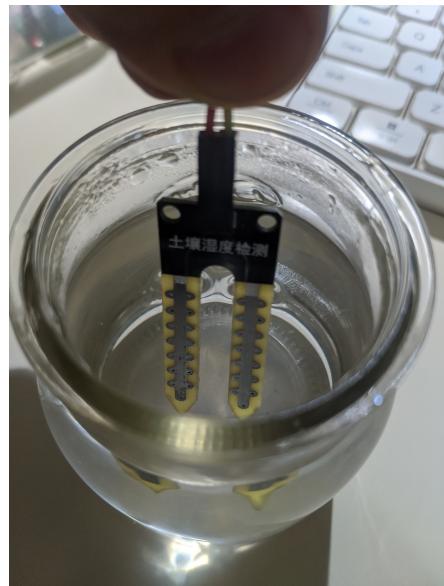
## Conexión con firebase

La idea principal del proyecto es que nuestro módulo ESP32 pueda almacenar los datos medidos en una base de datos en firebase. Además, debe poder recuperar parámetros que se encuentran almacenados en la misma. Por otro lado, disponemos de un sistema web que se encarga de visualizar los datos recabados por el ESP32 y de setear los valores necesarios para automatizar el riego. Esta comunicación la realizamos mediante la librería **Firebase\_ESP\_Client.h**, que encapsula la comunicación con la base de datos y hace uso del protocolo HTTP y los verbos GET, PUT y POST para la actualización de la misma. A continuación se muestra un pequeño diagrama que representa la comunicación del sistema:



## Calibración de sensores

Una de las dificultades que encontramos a la hora de utilizar el sensor de humedad en el suelo y el fotoresistor es que se debía encontrar los valores adecuados para el funcionamiento del sistema. En el caso del resistor se tuvieron que hacer pruebas midiendo la luminosidad del ambiente durante distintas etapas del día para encontrar los valores que más se aproximen a lo esperado. Por otro lado, en el caso del sensor de humedad en el suelo hubo que realizar experimentos midiendo la humedad en el aire para obtener el 0% y sumergiéndolo en agua para poder encontrar el 100% de humedad como se muestra en la imagen.



## Parámetros y toma de datos

Como explicamos anteriormente, nuestro sistema permite recuperar los parámetros que definen el riego del sistema desde firebase. Estos datos son necesarios para saber cual es el momento ideal de regar y pueden variar según el tipo de planta. El sistema de riego funciona de la siguiente manera<sup>2</sup>:

---

<sup>2</sup> <https://www.revistaespacios.com/a18v39n37/a18v39n37p27.pdf>

- Si la humedad del suelo es menor a la establecida como mínima dentro de los parámetros del sistema, se procede a regar.
- Sino, si la humedad del suelo actual se encuentre entre la máxima y mínima definidas como parámetro, el sistema siempre procederá a regar en los siguientes casos:
  - Cuando el valor medido de la humedad ambiente sea menor al parámetro de humedad mínima que necesita la planta.
  - Sino cuando la temperatura en el ambiente es mayor al parámetro de temperatura máxima establecido que tolera la planta .

En cualquier otro caso la bomba permanecerá apagada.

```
15:30:29.954 -> Humedad del aire minima: 30%
15:30:29.954 -> Humedad del suelo minima: 10%
15:30:29.954 -> Humedad del suelo maxima: 60%
15:30:29.954 -> Temperatura maxima: 35.00°C
```

Por otro lado, contamos con todas las medidas realizadas por nuestros sensores. En nuestro caso tomamos medidas de la **humedad del suelo, humedad ambiente, temperatura y sensación térmica**. Además, contamos con la **luminosidad** medida por un fotoresistor para saber cuándo encender o apagar la lámpara.

```
20:33:56.475 -> Humedad: 55% - Temperatura: 24.40°C - Sensacion termica: 24.33°C
20:33:56.475 -> LUMINOSIDAD: 1899
20:33:56.475 -> Encender lampara
20:33:56.475 -> Humedad del suelo: 1%
20:33:56.475 -> Regando...
```

## Página Web

Para la visualización de datos y la carga de los parámetros de riego en el sistema, desarrollamos una aplicación web con el framework de desarrollo Angular CLI. Esta simple página web consta de tres secciones:

- **Control de planta:** esta sección nos permite ver el estado de la planta en tiempo real mediante una tabla y guardar los valores para mantener un registro.

Parámetro	Estado Actual
Estado de la bomba	APAGADA
Estado de la lámpara	APAGADA
Última Actualización	01/12/2021 14:29
Humedad del Suelo	56%
Humedad ambiente	58%
Sensación térmica	24.63°C
Temperatura ambiente	24.60°C
<b>Guardar valores</b>	

- **Setear parámetros:** esta sección nos permite setear los valores necesarios para controlar el sistema de riego:

Humedad del ambiente mínima:  
30

Humedad del suelo máxima:  
60

Humedad del suelo mínima:  
10

Temperatura máxima:  
35

**Actualizar parámetros**

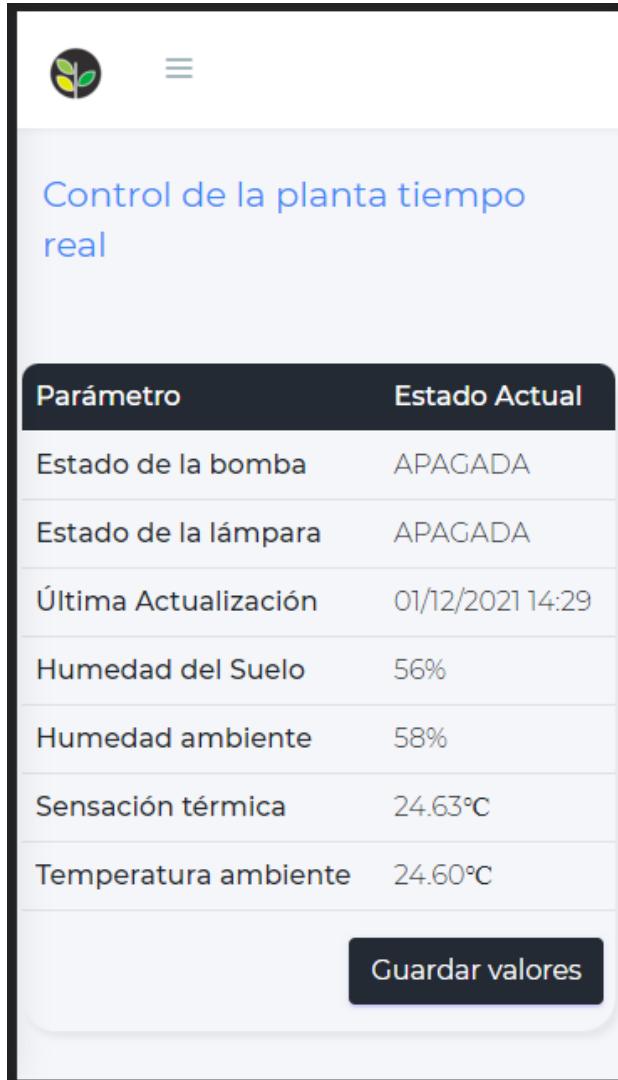
- **Saves:** esta sección nos permite visualizar todos parámetros almacenados para mantener un control en el tiempo:

Parámetro	Estado	Parámetro	Estado
Estado de la bomba	APAGADA	Estado de la bomba	APAGADA
Estado de la lámpara	APAGADA	Estado de la lámpara	APAGADA
Última Actualización	01/12/2021 14:01	Última Actualización	01/12/2021 14:07
Humedad del Suelo	31%	Humedad del Suelo	91%
Humedad ambiente	53%	Humedad ambiente	54%
Sensación térmica	28.25°C	Sensación térmica	28.00°C
Temperatura ambiente	27.60°C	Temperatura ambiente	27.30°C

Parámetro	Estado	Parámetro	Estado
Estado de la bomba	APAGADA	Estado de la bomba	APAGADA
Estado de la lámpara	APAGADA	Estado de la lámpara	APAGADA
Última Actualización	01/12/2021 14:07	Última Actualización	01/12/2021 14:07
Humedad del Suelo	90%	Humedad del Suelo	90%
Humedad ambiente	54%	Humedad ambiente	54%
Sensación térmica	27.88°C	Sensación térmica	27.88°C
Temperatura ambiente	27.20°C	Temperatura ambiente	27.20°C

Otra funcionalidad importante a destacar ,es que la aplicación web está desarrollada para que funcione de manera correcta en dispositivos móviles.



## Funcionamiento

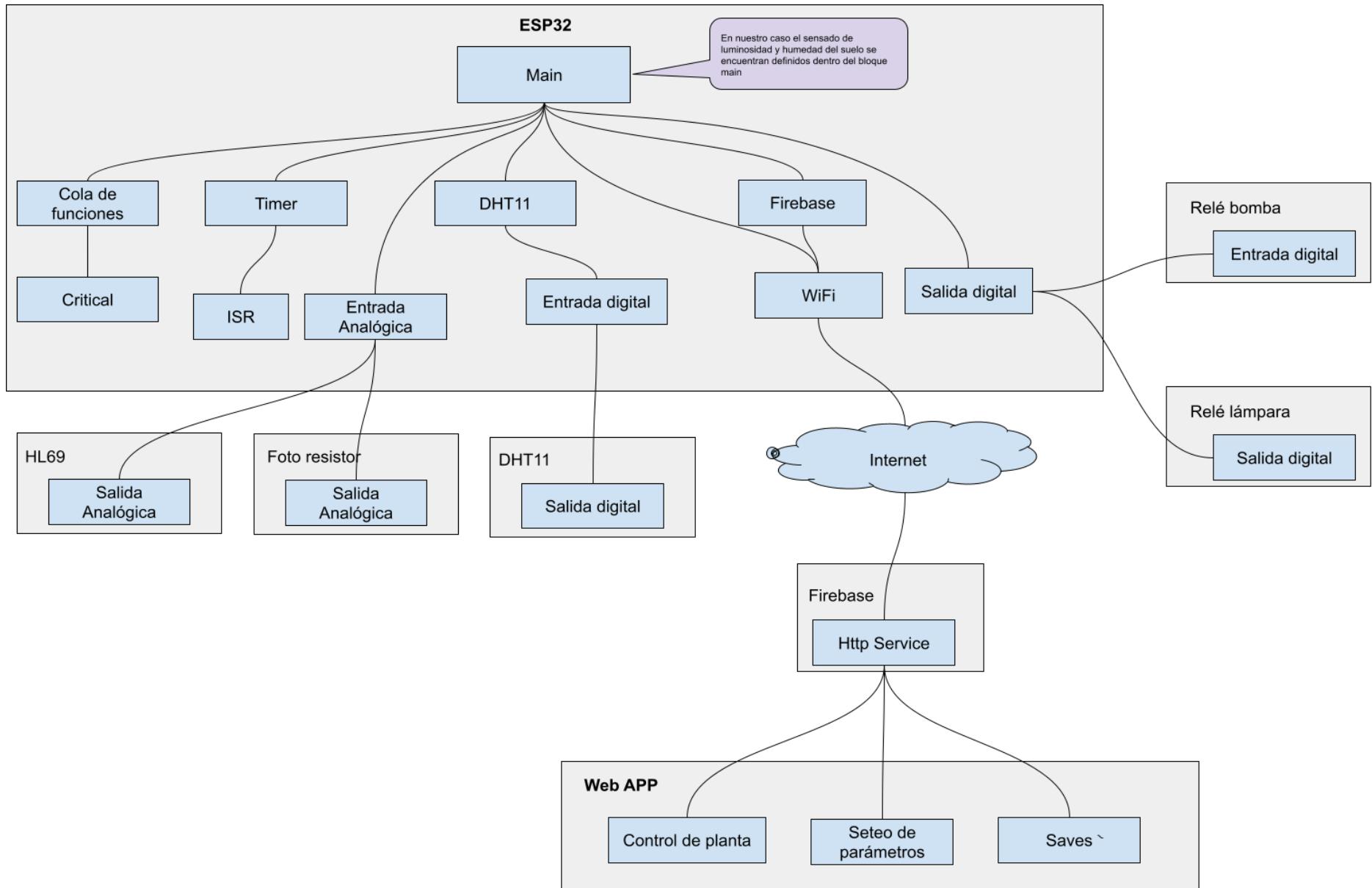
En líneas generales el funcionamiento del proyecto es bastante simple, primero inicializamos la conexión WiFi y luego se procede a la conexión con Firebase mediante la cual se obtienen los parámetros mínimos que necesita la planta para regarla cuando sea necesario. En caso de no poder obtener los parámetros, se utilizan unos definidos por defecto.

Una vez conectado el sistema con Firebase se procede a iniciar los relays utilizando dos salidas digitales, iniciar el driver del sensor DHT11 asignándole una entrada digital y por último se inicia el driver de timer para que ocurra una interrupción cada 1 segundo.

Para realizar un correcto funcionamiento del sistema, utilizamos una cola de funciones que se encarga de ejecutar las funciones encoladas en el loop del hilo principal del sistema. Por otro lado, el driver de timer implementado se encarga de encolar su función de callback cuando ocurre una interrupción para evitar problemas de sincronización.

La función de callback que encola el driver de timer se encarga de, en primer lugar, medir con el DHT11 la temperatura y humedad del ambiente, tomar medidas de la luminosidad mediante el fotoresistor y la humedad del suelo a través de sensor HL69. Una vez medidos los valores se procede a chequear si es necesario regar. Para esto, se acude a una variable donde el riego se lleva a cabo por una cierta cantidad de segundos dependiendo del estado del ambiente y humedad del suelo. Por otro lado se tendrá una variable que irá aumentando en 1 por cada segundo y al llegar a X segundos, se chequeará en el servidor “ntp” la fecha y hora local, se enviarán a Firebase los valores obtenidos y si los parámetros definidos para la planta cambiaron, los mismos serán recuperados para actualizarlos en el sistema.

A continuación, se presenta un diagrama de bloques realizado para este proyecto:



# Complicaciones presentadas

- Energía del sistema
  - Se tomó la decisión de utilizar un cargador de celular.
- Sincronización
  - Modificamos el librería de fnqueue y critical propuestos por la cátedra.
- Maqueta
  - Para lograr una estructura a la altura del proyecto y siendo que se realizó a pequeña escala, la solución encontrada fue armar con maderas un soporte, el cual pueda mantener la protoboard en una parte resguardada del agua para cuidar los sensores.
- Calibración de sensores
  - Para el correcto funcionamiento del fotoresistor, hubo que medir por un período de días en distintos horarios la luminosidad en la habitación donde fue instalado el sistema embebido de manera tal que se lograra encontrar un valor adecuado para encender la lámpara.
- Aprendizaje de nuevas tecnologías en un periodo corto de tiempo
  - Librerías del ESP32
  - Firebase
  - Angular

# Posibles mejoras

A continuación se enumeran optimizaciones viables de realizarse a futuro:

- Implementar el riego por goteo
- Modificación para un posible cultivo por hidroponia.
- En caso de la interfaz de usuario
  - Tener predefinidos los parámetros de ciertas especies, de manera tal que el usuario pueda elegir qué planta/cultivo desea cuidar, olvidándose de setear manualmente los parámetros. En caso que no se encuentre la especie, pueda agregarla.
  - Carteles de comunicación y notificaciones hacia el usuario.
  - Imagen/video de la planta en tiempo real
- Agregar un módulo de clock para evitar consultar constantemente al servidor ntp la fecha y hora.

# Conclusiones

Mediante este transcurso del proyecto se pudo ir notando que este sistema es de bajo costo, viable incluso para ofrecerse como un sistema de cultivo hogareño a hobbistas o personas que quieran incursionar en el cuidado y/o cultivo de plantas.

Por otro lado es sumamente posible llevar el sistema a una implementación a mayor escala, instalándose en invernaderos e incluso modificándolo para el cultivo hidropónico de forma tal que zonas con escasos recursos hídricos y solares puedan también cosechar productos que de no ser así deben importar.