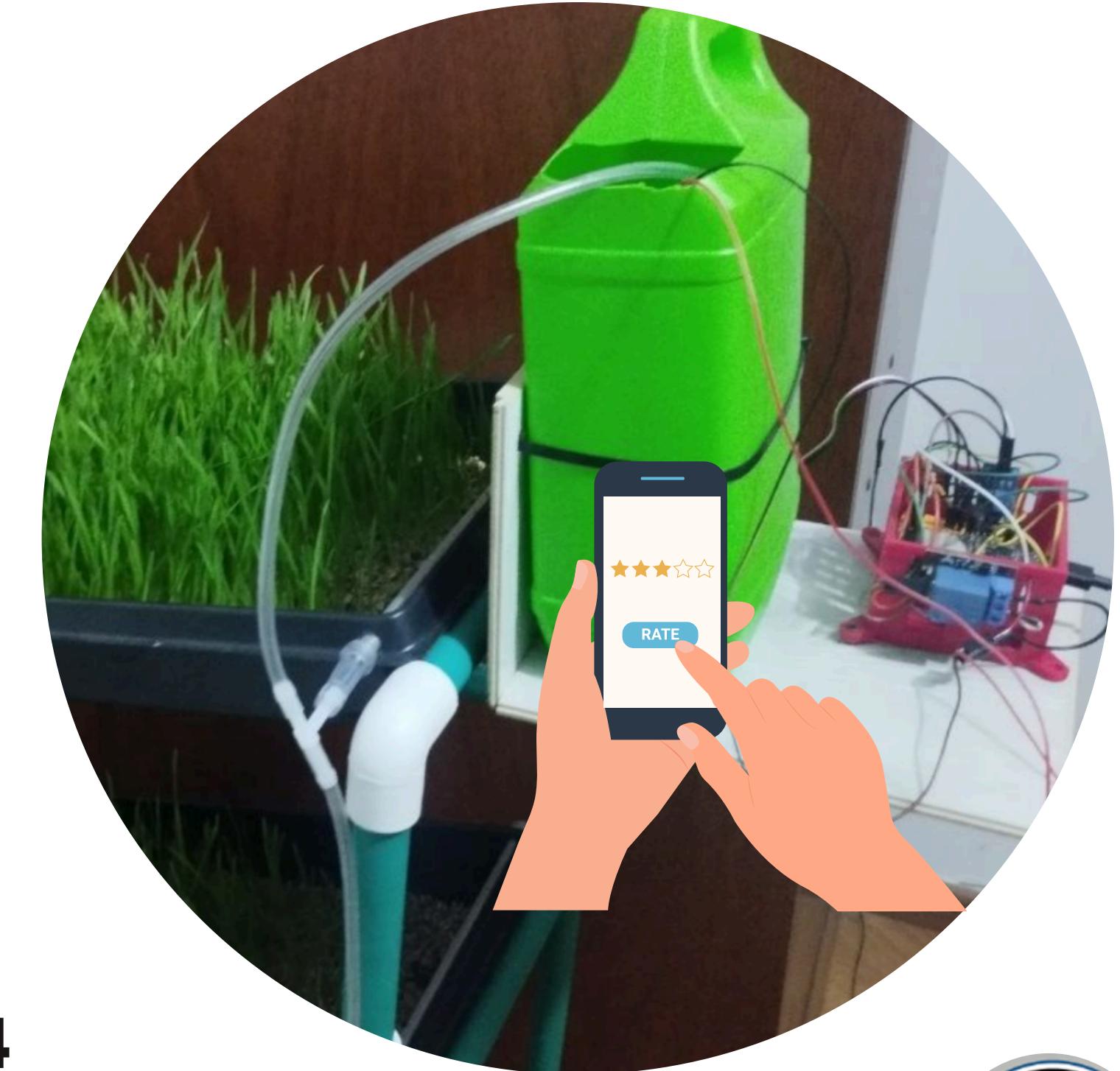


Desarrollo de un Prototipo IoT para el control del riego y monitoreo de variables ambientales en el cultivo de forraje verde

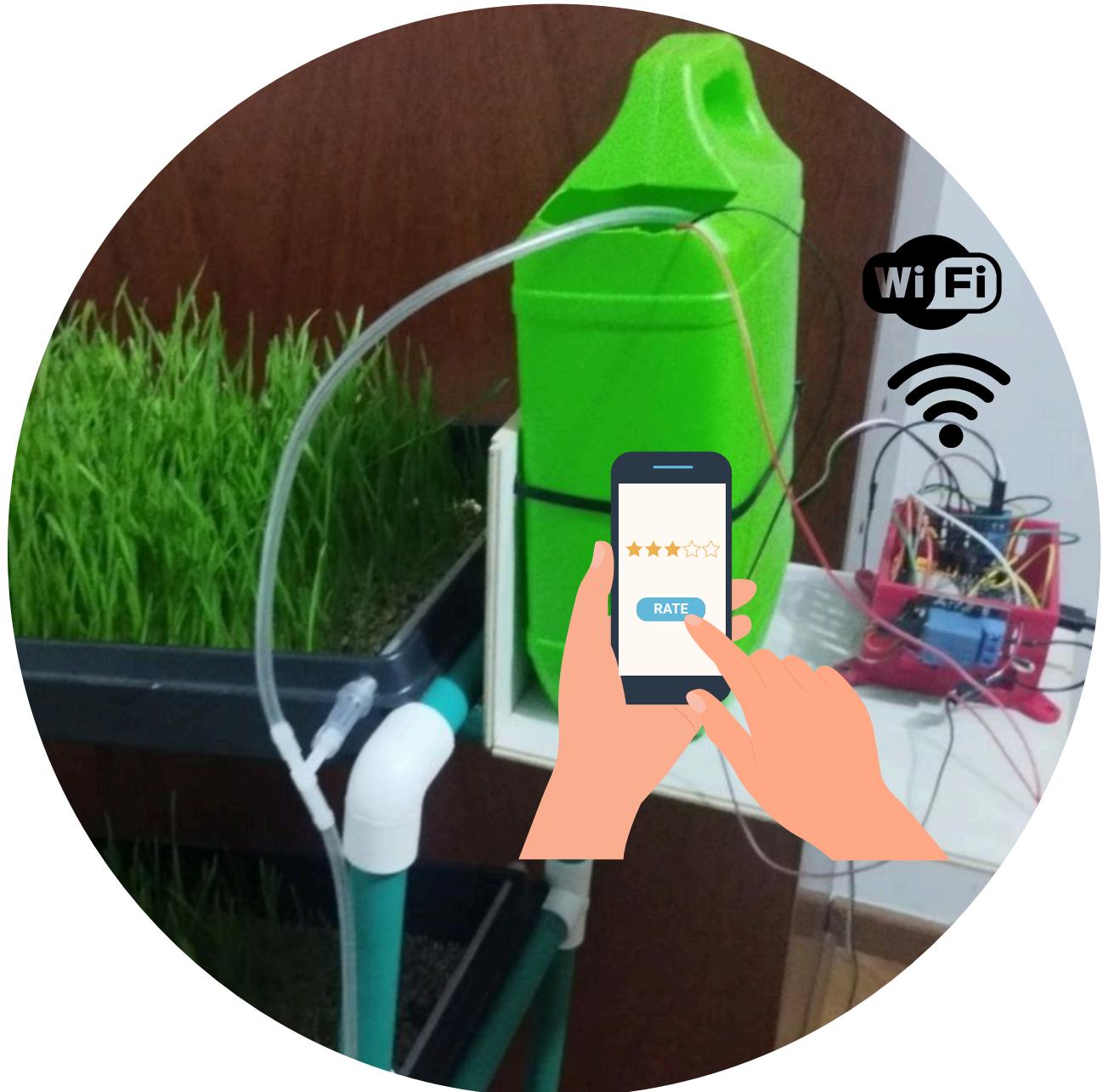


Facultad de Ingeniería de Sistemas 2024

Nicolás Mahecha Pérez



Introducción



Este proyecto tiene como propósito el desarrollo de un prototipo a través de la implementación de tecnología de **Internet de las cosas (IoT)** para el control del riego y monitoreo de variables ambientales en el cultivo de forraje verde

FVH-IoT



Planteamiento del problema



¿Cómo el desarrollo de un prototipo basado en tecnologías emergentes como el Internet de las Cosas (IoT) facilita el control del cultivo de forraje verde hidropónico mediante el monitoreo de variables ambientales como la humedad y la temperatura, y la automatización del sistema de riego?

Objetivos

Objetivo General

Desarrollar un prototipo IoT llamado **FVH-TEK IOT** para el control del riego y monitoreo de variables ambientales (temperatura, humedad) en el cultivo de forraje verde hidropónico

FVH-IoT



Objetivos

Objetivos Específicos

1. Realizar el levantamiento de información sobre el cultivo de forraje verde y el internet de las cosas.
2. Analizar la información recolectada para identificar las necesidades y definir los requerimientos funcionales y no funcionales.
3. Diseñar la arquitectura y escoger los componentes del prototipo.
4. Codificar el prototipo integrando las diferentes herramientas de software para controlar el riego y visualizar el estado de las variables ambientales del cultivo de forraje verde hidropónico.



Justificación - Alcance

FVH-TEK IOT busca implementar técnicas de agricultura moderna por medio de tecnología del internet de las cosas en el cultivo de forraje verde hidropónico, brindando una solución de control del riego automático remotamente, y un monitoreo constante de las variables del cultivo.

Soluciones

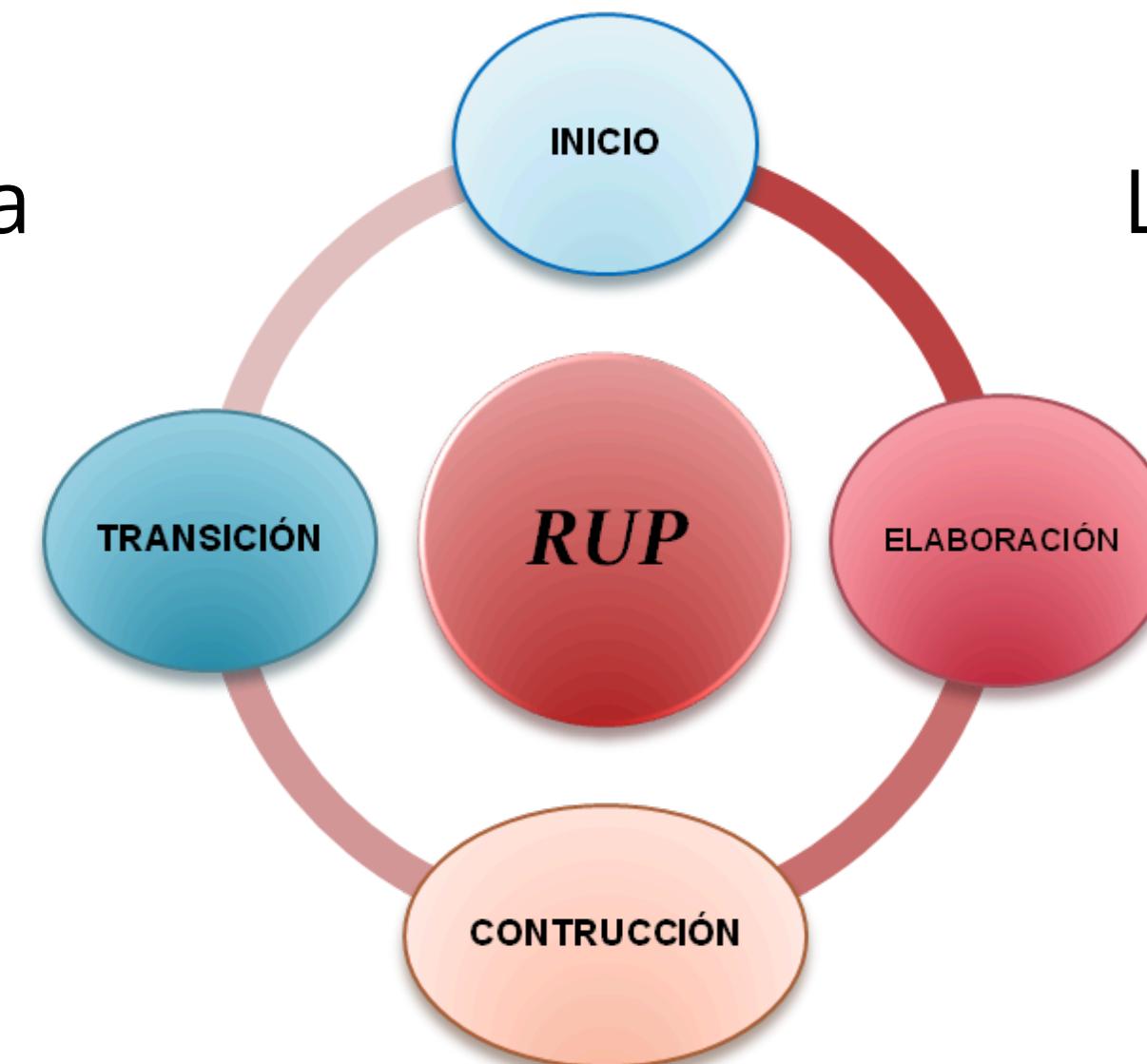
- Control del riego automático y remotamente
- Control del uso del agua en el riego
- Monitoreo de variables ambientales (Humedad, temperatura)
- Almacenamiento de datos recopilados por los sensores
- Seguridad de la información
- Decisiones Informadas



Solución al planteamiento del problema

Metodología RUP

1. Investigación Aplicada
2. Requerimientos
3. Diseño
4. Pruebas



Lenguajes de Programación
Internet de las Cosas
Base de Datos NoSQL
Dashboard Web



Metodología RUP

Fase Inicio

- Requerimientos
- Casos de Uso

Fase Elaboración

- Desarrollo Arquitectura
- Diagrama de Hardware
- Diagrama de Software
- Viabilidad del prototipo

Fase Construcción

- Recolectar datos de los sensores
- Almacenar datos de los sensores en la base de datos Firebase
- Implementar el control del riego

Fase Transición

- Pruebas del prototipo
- Identificación de errores y fallas



Requerimientos funcionales

RF-01	ESP-32
Descripción	Guarda información de los sensores de humedad del suelo, temperatura y humedad ambiental. Guarda información del sistema de riego. Envía la información almacenada al servicio de Realtime Database de Google Firebase.
RF-02	FIREBASE
Descripción	Realtime Database: recibe la información de los sensores y las variables del sistema.
RF-03	HYDROHARMONY IOT - APP WEB
Descripción	Visualización grafica de las variables del cultivo, control remoto del sistema de riego.



Requerimientos no funcionales

RNF-01	ESP-32 - PROGRAMACION C++
Descripción	Los módulos o scripts son programados con IDE Arduino. C++ Conectado a internet con una red Wifi. Envía información a la web cada ciclo. Envía información a la base de datos cada ciclo.
RNF-02	FIREBASE – BASE DE DATOS NOSQL - DOCUMENTAL
Descripción	Activar los servicios Realtime Database. (Autorización almacenar datos Autenticación de usuario)
RNF-03	HYDROHARMONY IOT - APP WEB – PROGRAMACION REACT
Descripción	Configuración de la interfaz gráfica en web. REACT Configuración de los widgets para las lecturas de los sensores. Configuración del botón que activara y desactivara el sistema de riego. Uso de autenticación de Firebase. Visualización de los datos de los sensores. Generación de alertas del sistema. Programación Responsive.



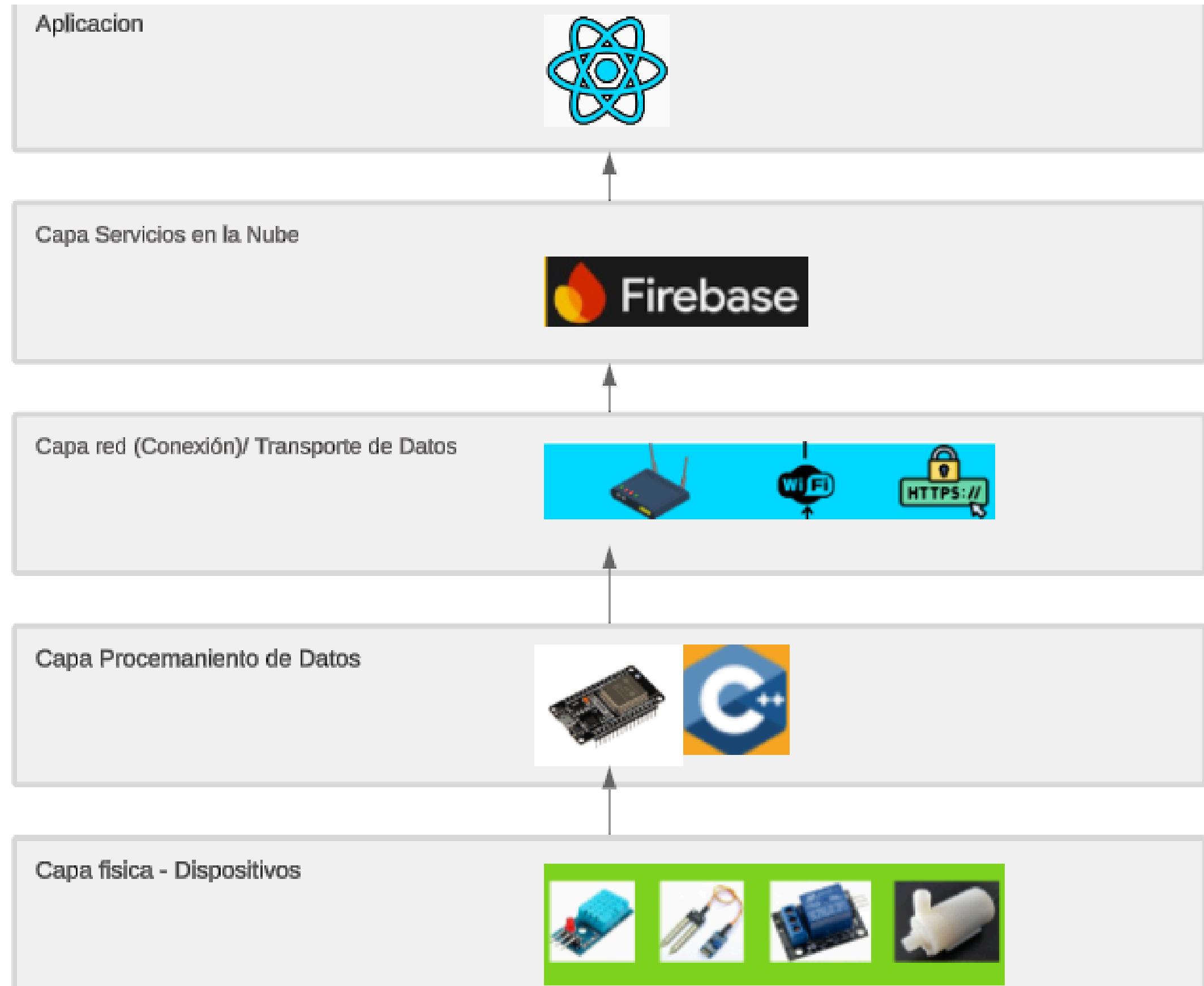
Desarrollo del Proyecto

Con el levantamiento de información realizado bajo la metodología RUP y la investigación del cultivo de forraje hidropónico y el internet de las cosas se logra identificar y establecer cuáles son los módulos a estructurar para el desarrollo del prototipo **FVH-TEK IOT**

El prototipo desarrollara un sistema IoT con el ESP32 programado en C++ para controlar el riego automático mediante HTTP y WiFi. Los datos de sensores serán enviados a Firebase, donde se almacenarán y visualizarán en una aplicación web hecha en React. Los usuarios podrán monitorear variables ambientales y activar el riego en tiempo real desde el dashboard web.



Arquitectura Capas

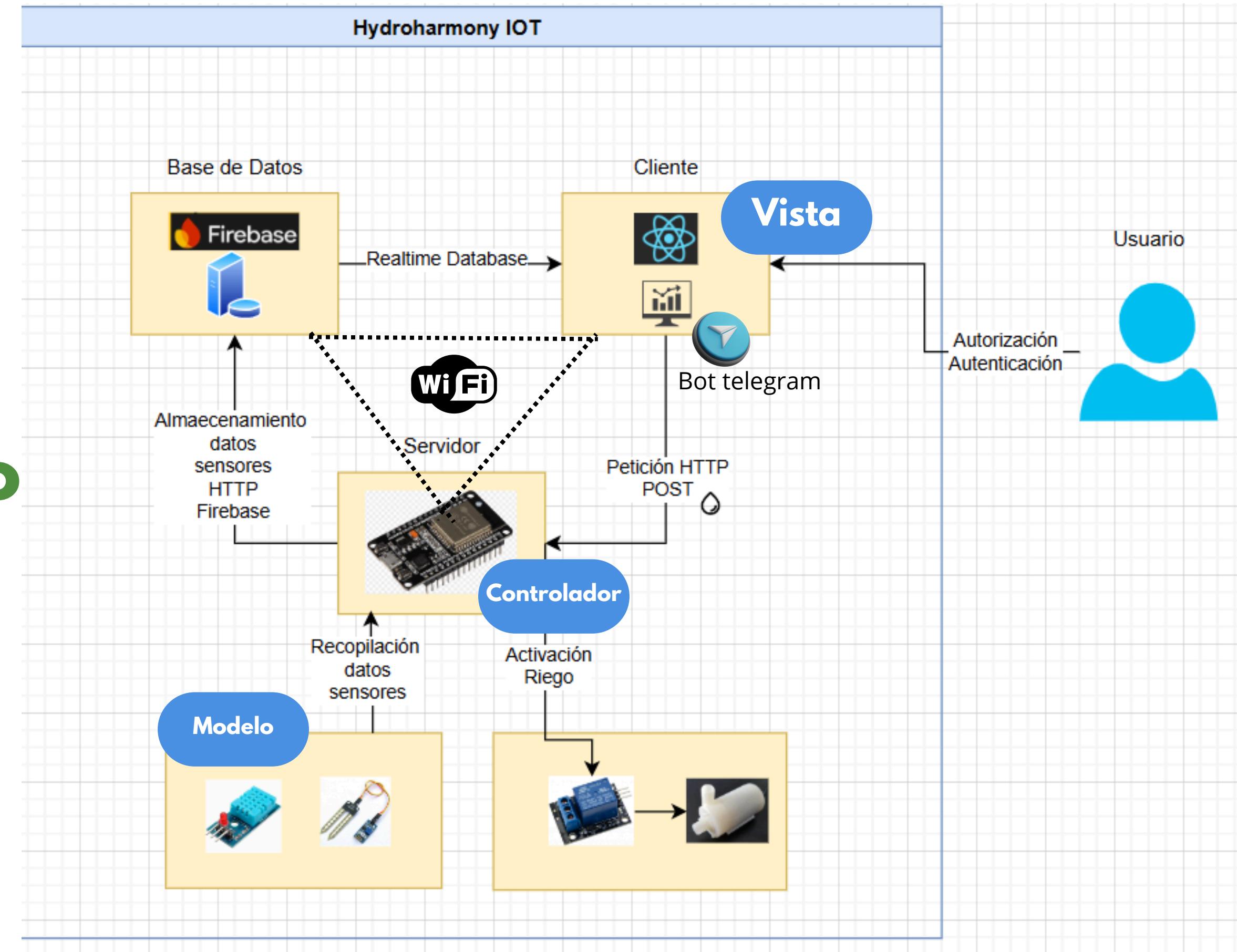


Arquitectura Software Cliente - Servidor

Patrón de Diseño

MVC

Modelo Vista
Controlador



Patrón Diseño

Componentes (React)

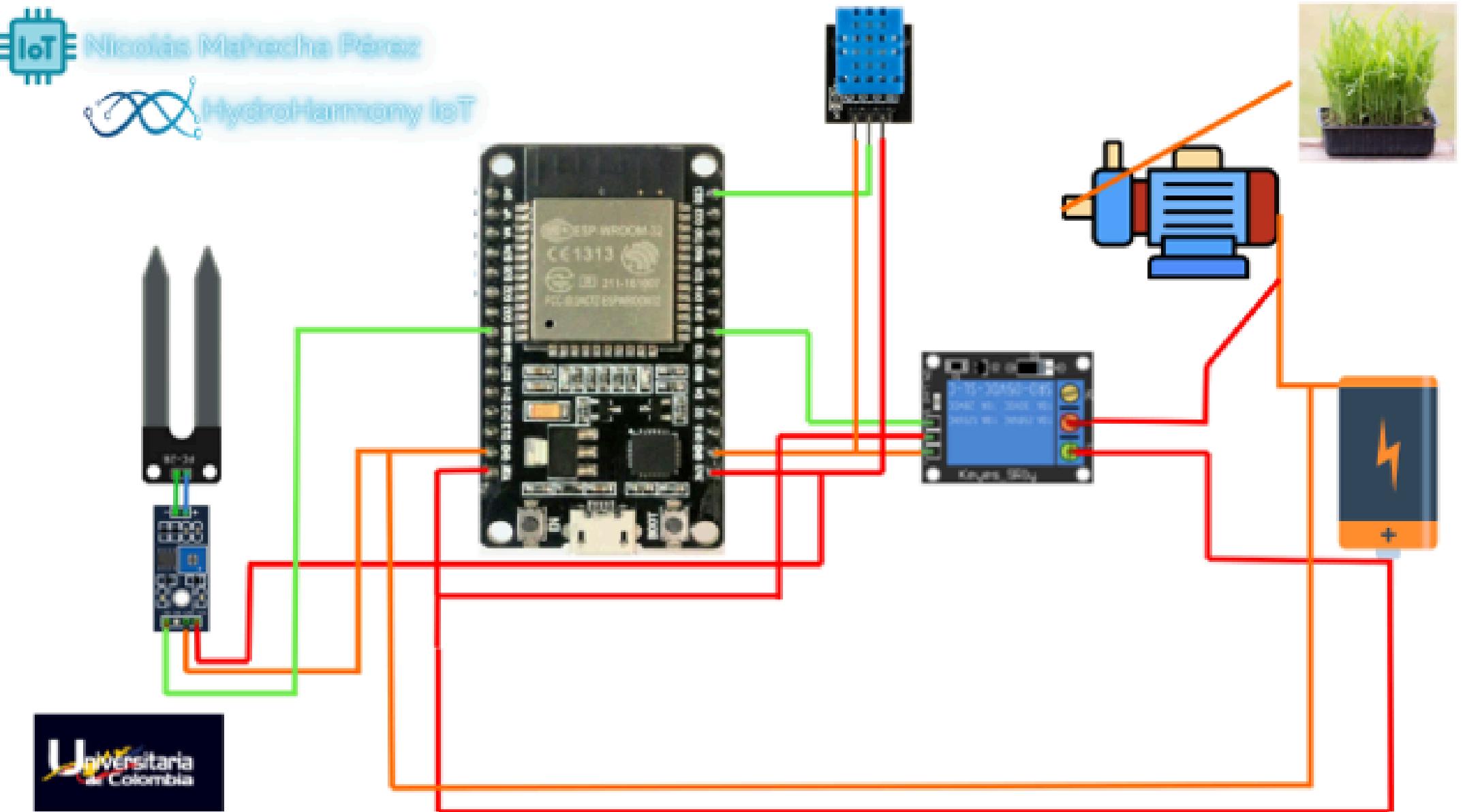


```
src
  assets
  Components
    Alert
    Card
    Dashboard
    Header
    IrrigationRecords
    LineChart
    LoadingSpinner
    LoginRegister
    Logout
    Sidebar
```

A screenshot of a file explorer showing a directory structure. The 'src' folder contains 'assets' and 'Components'. The 'Components' folder is expanded, listing ten React components: Alert, Card, Dashboard, Header, IrrigationRecords, LineChart, LoadingSpinner, LoginRegister, Logout, and Sidebar. Each component name is preceded by a blue right-pointing arrow indicating it is a file.

Arquitectura Hardware

- PLACA ESP-32
- Relay PIN D5
(Bomba de agua)
- DTH11 PIN 23



Base de Datos

FIREBASE

NOSQL

DOCUMENTAL



```
1 // firebaseConfig.ts
2 import { initializeApp } from 'firebase/app';
3 import { getAuth } from 'firebase/auth';
4 import { getDatabase } from 'firebase/database';
5
6 // Tu configuración de Firebase
7 const firebaseConfig = {
8   apiKey: "AIzaSyB6yQkpIwN2VkJzXNXwaa0kxHk6CKycmOk",
9   authDomain: "hydroharmony-iot.firebaseioapp.com",
10  projectId: "hydroharmony-iot",
11  storageBucket: "hydroharmony-iot.appspot.com",
12  messagingSenderId: "484029961213",
13  appId: "1:484029961213:web:94aad0170beaa577507f94",
14  databaseURL: "https://hydroharmony-iot-default.firebaseio.com"
15 };
16
17 // Inicializa Firebase
18 const appFirebase = initializeApp(firebaseConfig);
19
20 // Inicializa Auth y Realtime Database
21 const auth = getAuth(appFirebase);
22 const database = getDatabase(appFirebase);
23
24 export { appFirebase, auth, database };
25
```

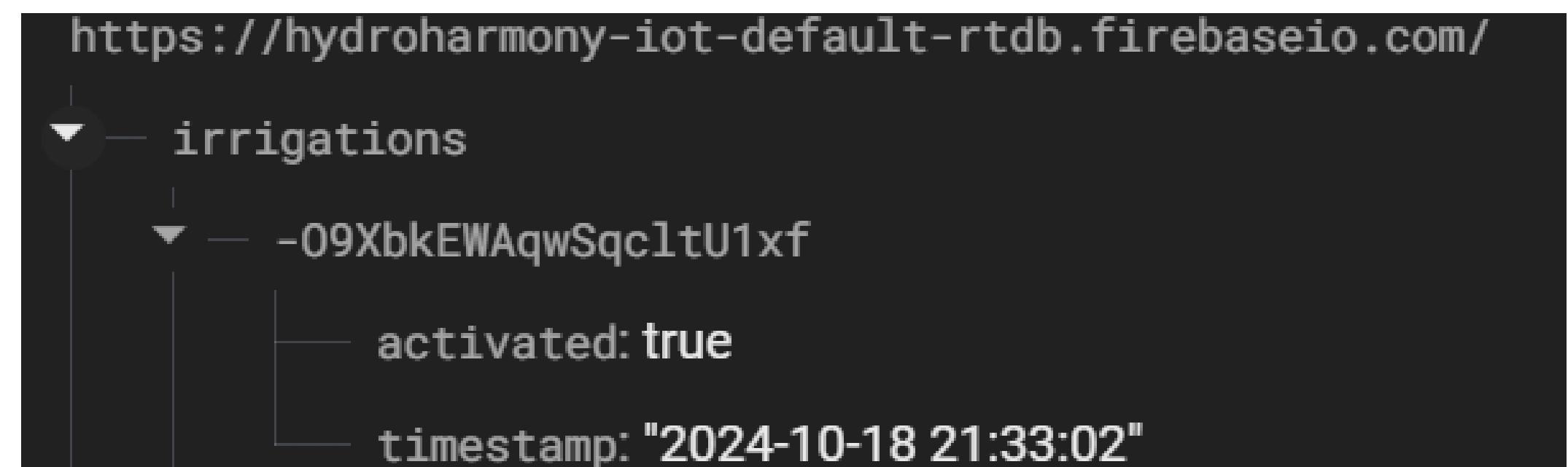
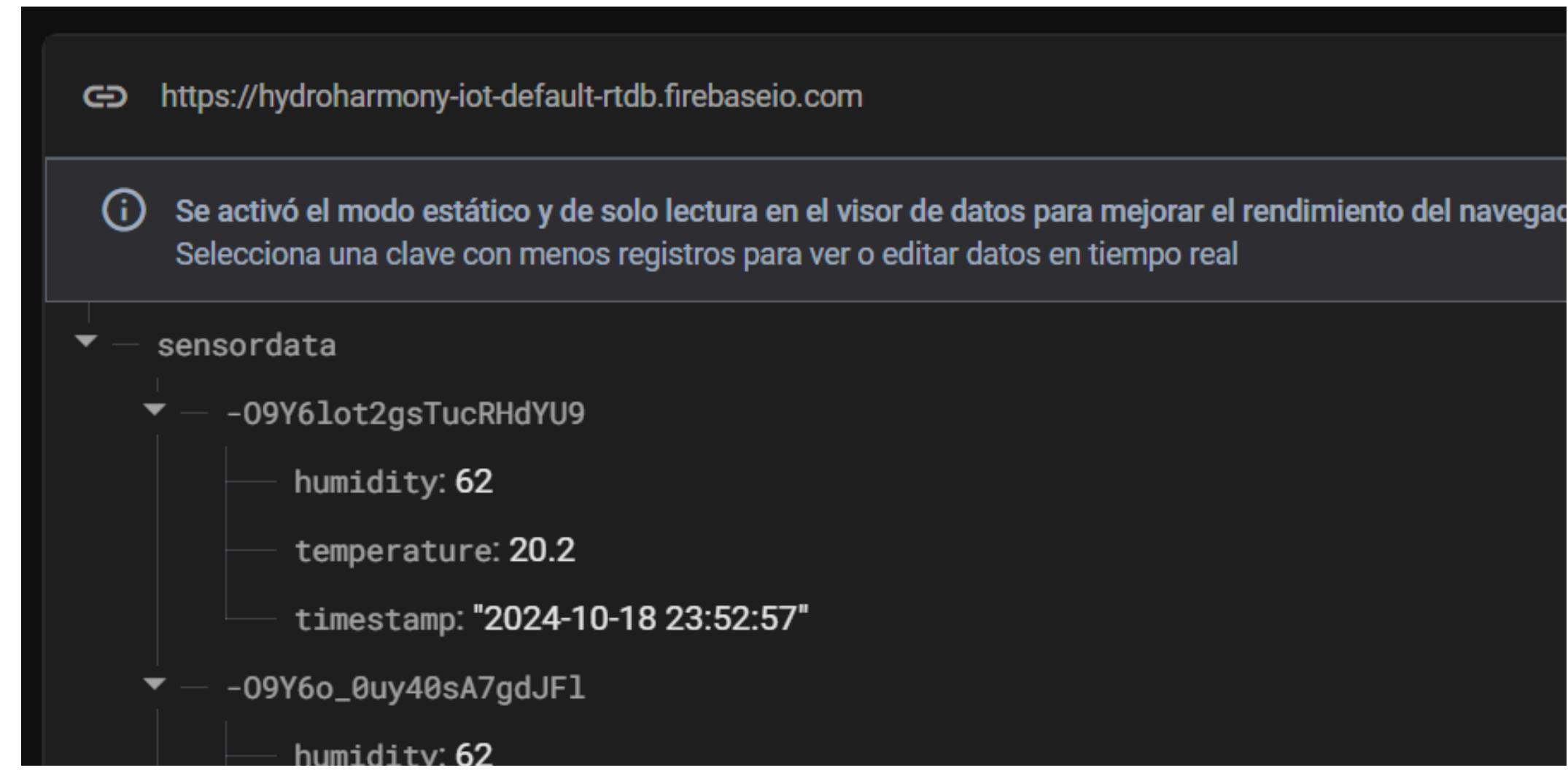


Base de Datos

FIREBASE

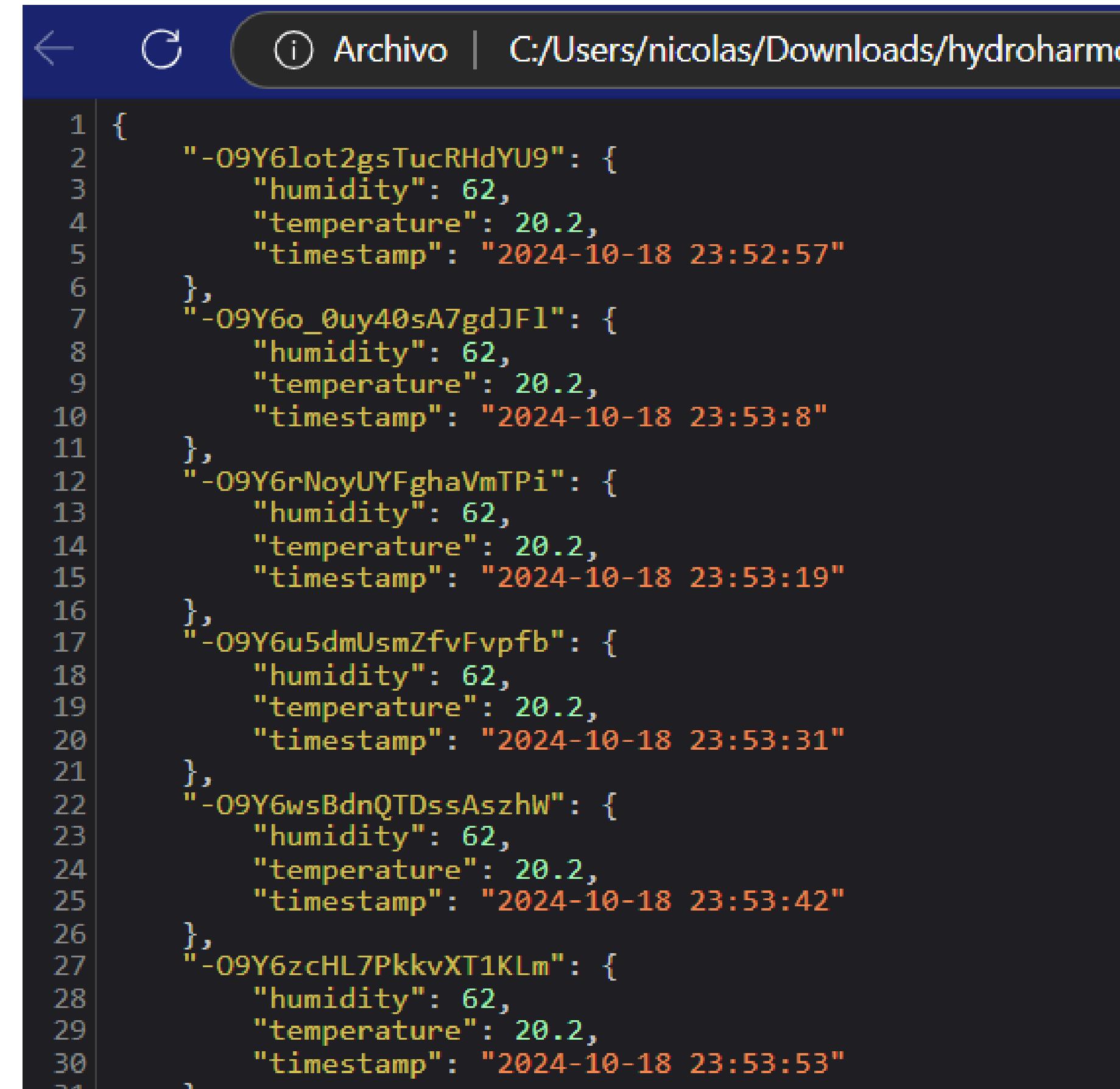
NOSQL

DOCUMENTAL



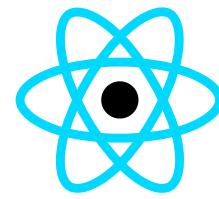
Base de Datos

FIREBASE NOSQL DOCUMENTAL

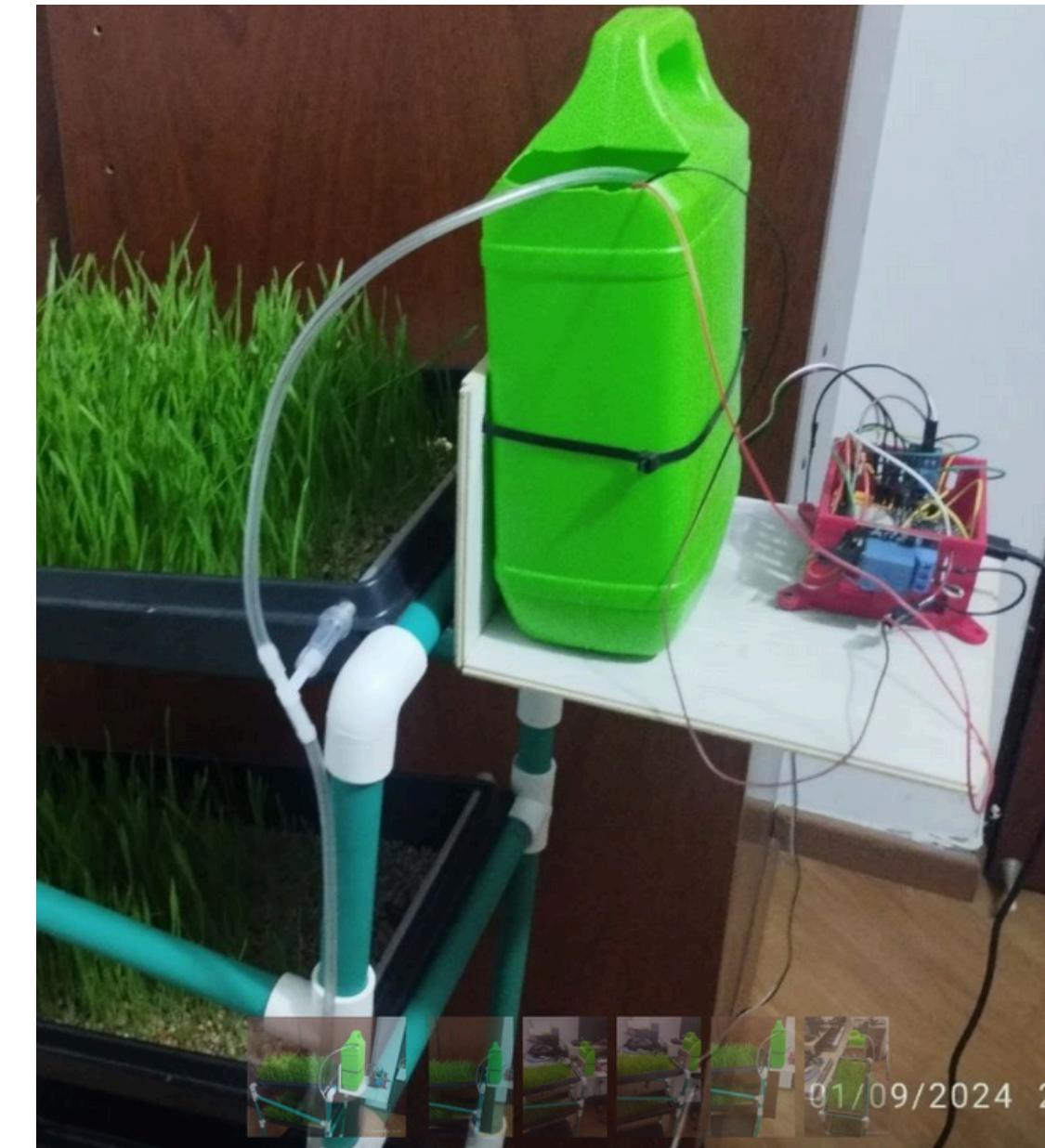
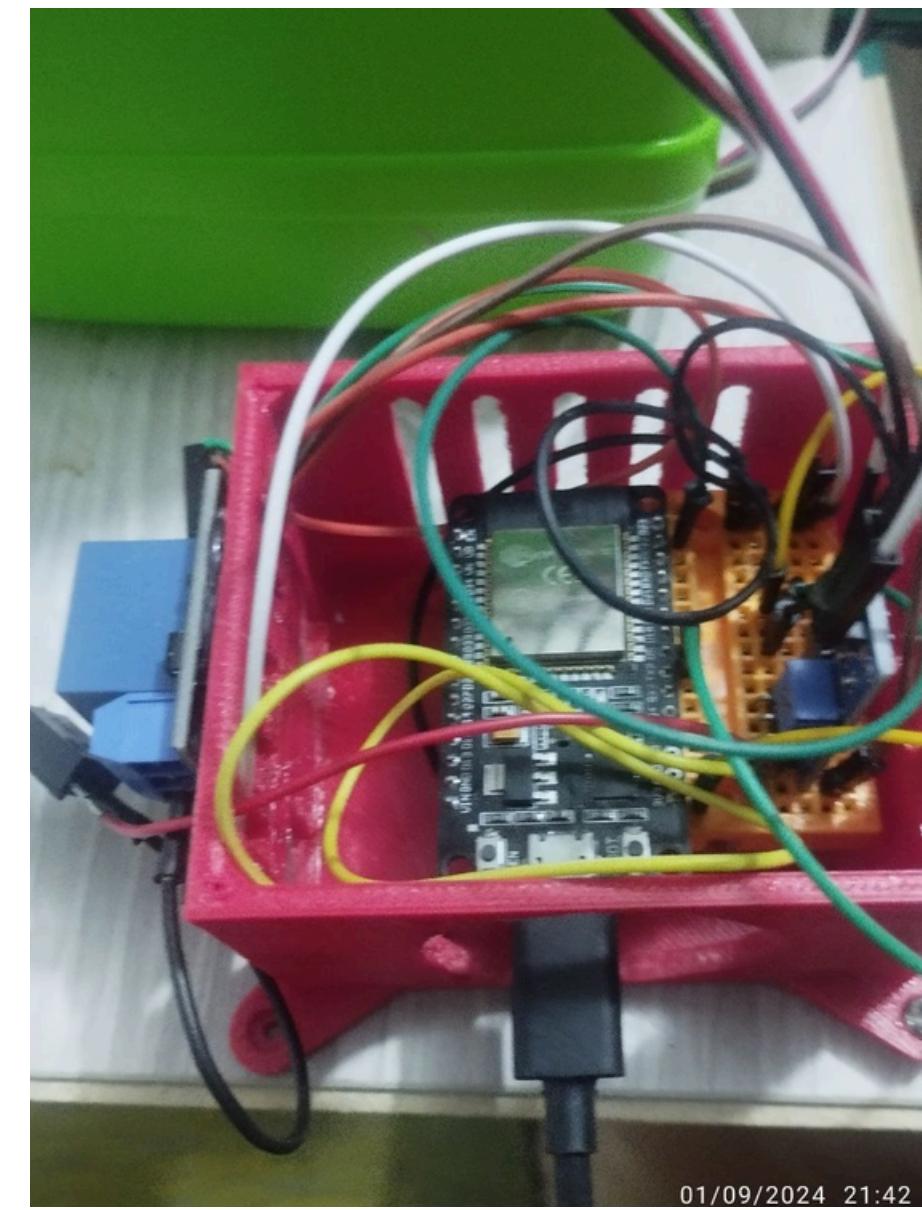


A screenshot of a code editor showing a JSON array of documents from a Firebase database. The array contains 30 documents, each representing a measurement with fields: humidity, temperature, and timestamp. The timestamp values are identical for all documents.

```
1  {
2      "-09Y6lot2gsTucRHdYU9": {
3          "humidity": 62,
4          "temperature": 20.2,
5          "timestamp": "2024-10-18 23:52:57"
6      },
7      "-09Y6o_0uy40sA7gdJF1": {
8          "humidity": 62,
9          "temperature": 20.2,
10         "timestamp": "2024-10-18 23:53:8"
11     },
12     "-09Y6rNoyUYFghaVmTPi": {
13         "humidity": 62,
14         "temperature": 20.2,
15         "timestamp": "2024-10-18 23:53:19"
16     },
17     "-09Y6u5dmUsmZfvFvpfb": {
18         "humidity": 62,
19         "temperature": 20.2,
20         "timestamp": "2024-10-18 23:53:31"
21     },
22     "-09Y6wsBdnQTDssAszhW": {
23         "humidity": 62,
24         "temperature": 20.2,
25         "timestamp": "2024-10-18 23:53:42"
26     },
27     "-09Y6zcHL7PkkvXT1KLm": {
28         "humidity": 62,
29         "temperature": 20.2,
30         "timestamp": "2024-10-18 23:53:53"
31 }
```

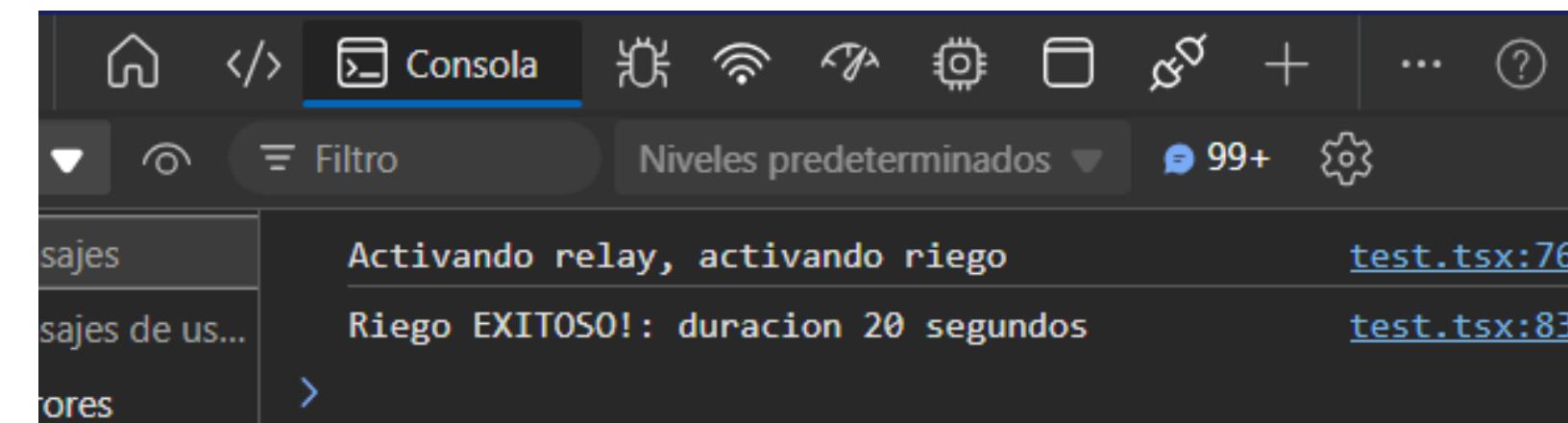


React



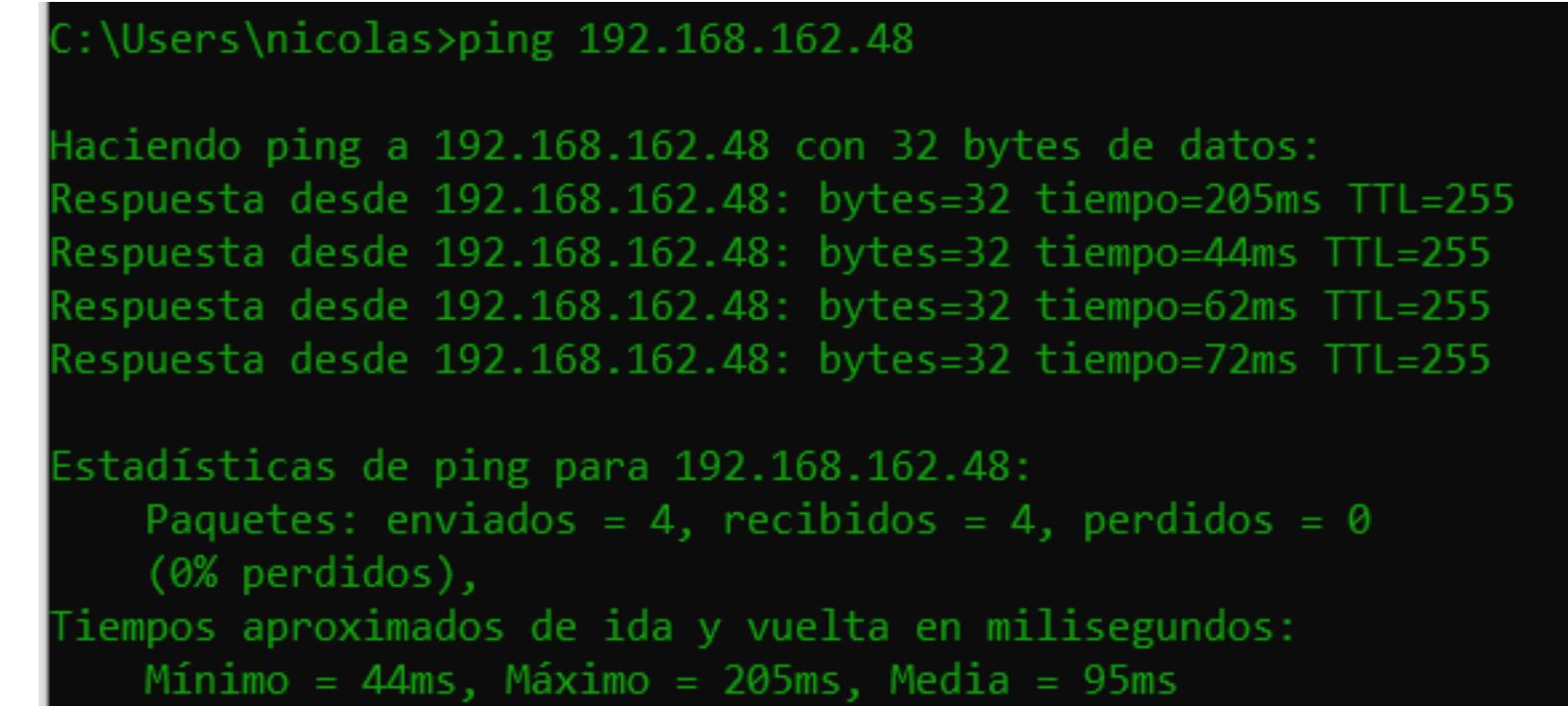
El sistema de riego se activa mediante un botón por medio del cliente (React web) que envía una solicitud POST vía HTTP al servidor (ESP32), que acciona el relé para encender la bomba de agua.





A screenshot of a terminal window titled "Consola". The logs show:

```
sajes          Activando relay, activando riego      test.tsx:76
sajes de us...  Riego EXITOSO!: duracion 20 segundos  test.tsx:83
ores          >
```



A screenshot of a terminal window showing the output of a "ping" command:

```
C:\Users\nicolas>ping 192.168.162.48

Haciendo ping a 192.168.162.48 con 32 bytes de datos:
Respuesta desde 192.168.162.48: bytes=32 tiempo=205ms TTL=255
Respuesta desde 192.168.162.48: bytes=32 tiempo=44ms TTL=255
Respuesta desde 192.168.162.48: bytes=32 tiempo=62ms TTL=255
Respuesta desde 192.168.162.48: bytes=32 tiempo=72ms TTL=255

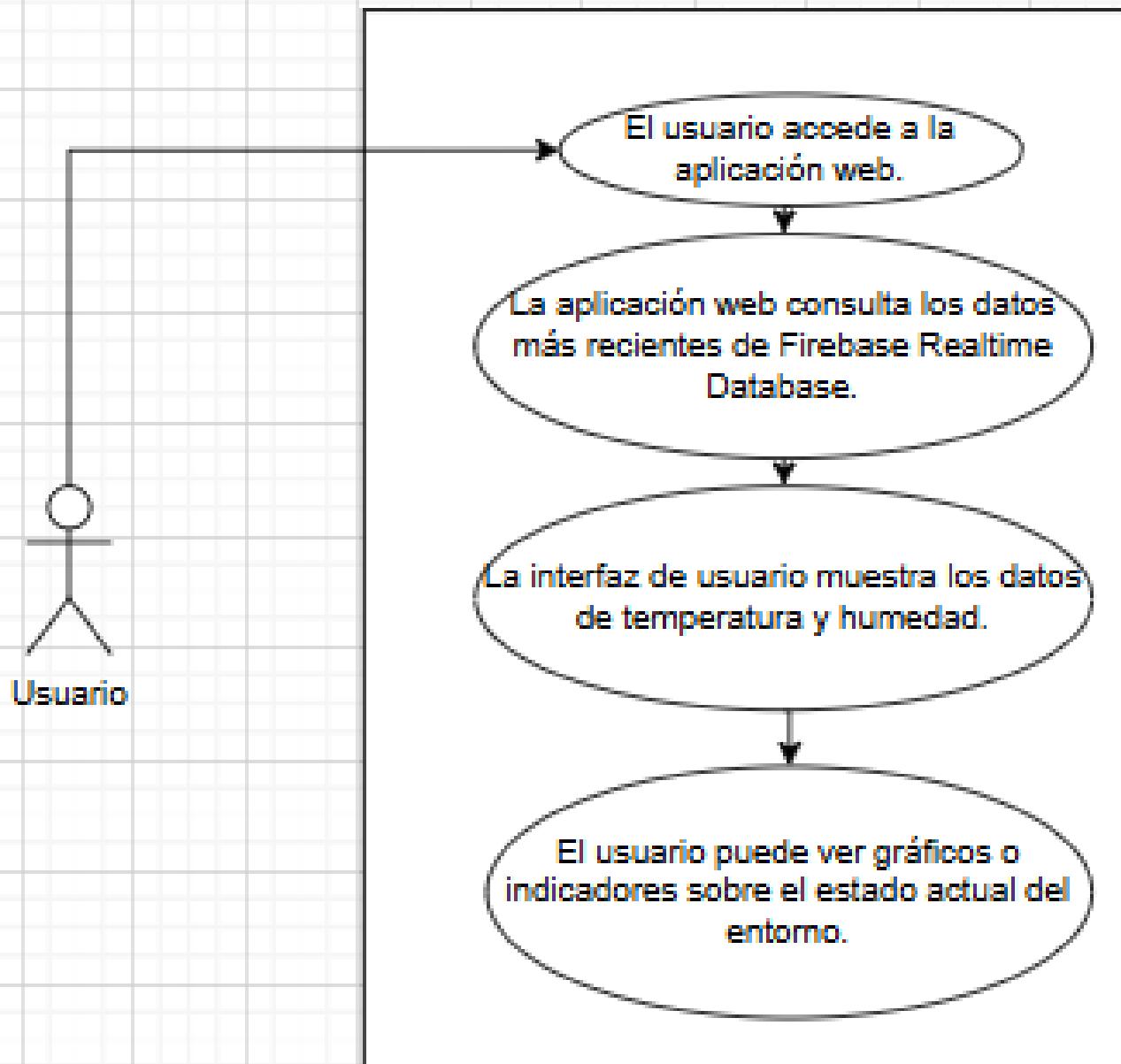
Estadísticas de ping para 192.168.162.48:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
                (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 44ms, Máximo = 205ms, Media = 95ms
```

El sistema de riego se activa mediante una solicitud POST enviada vía HTTP al ESP32, que acciona el relé para encender la bomba de agua.

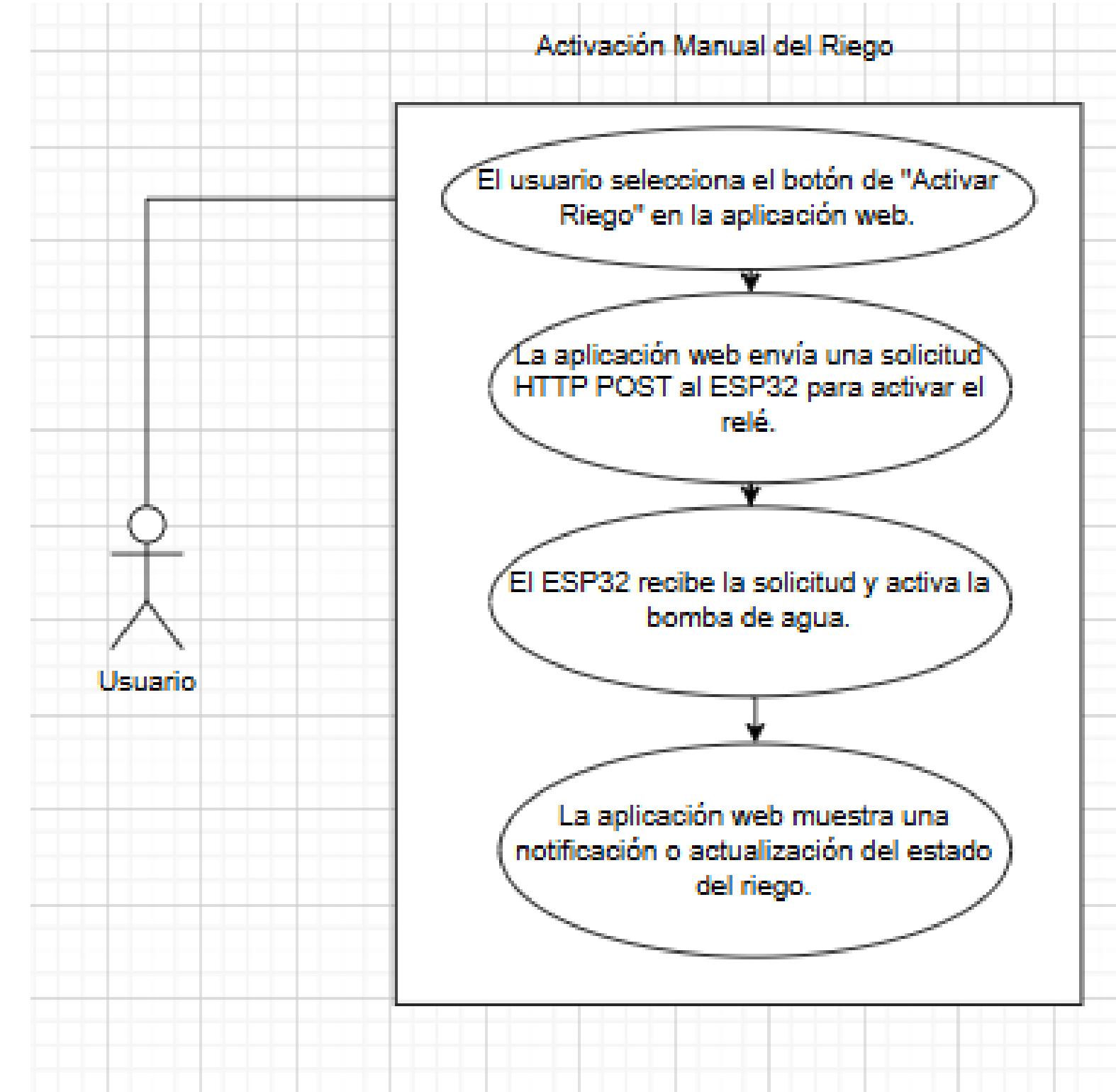


Casos de Uso

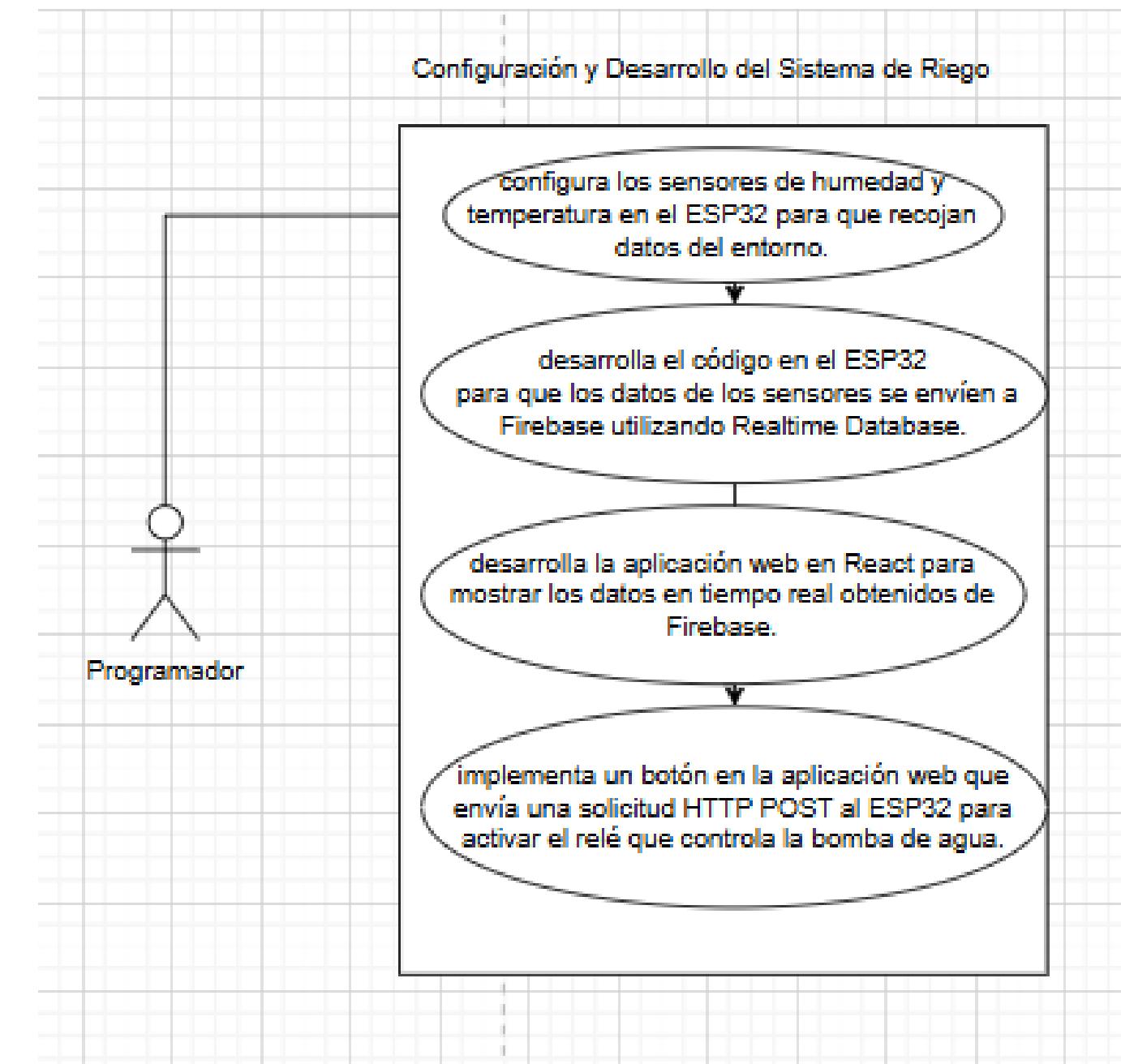
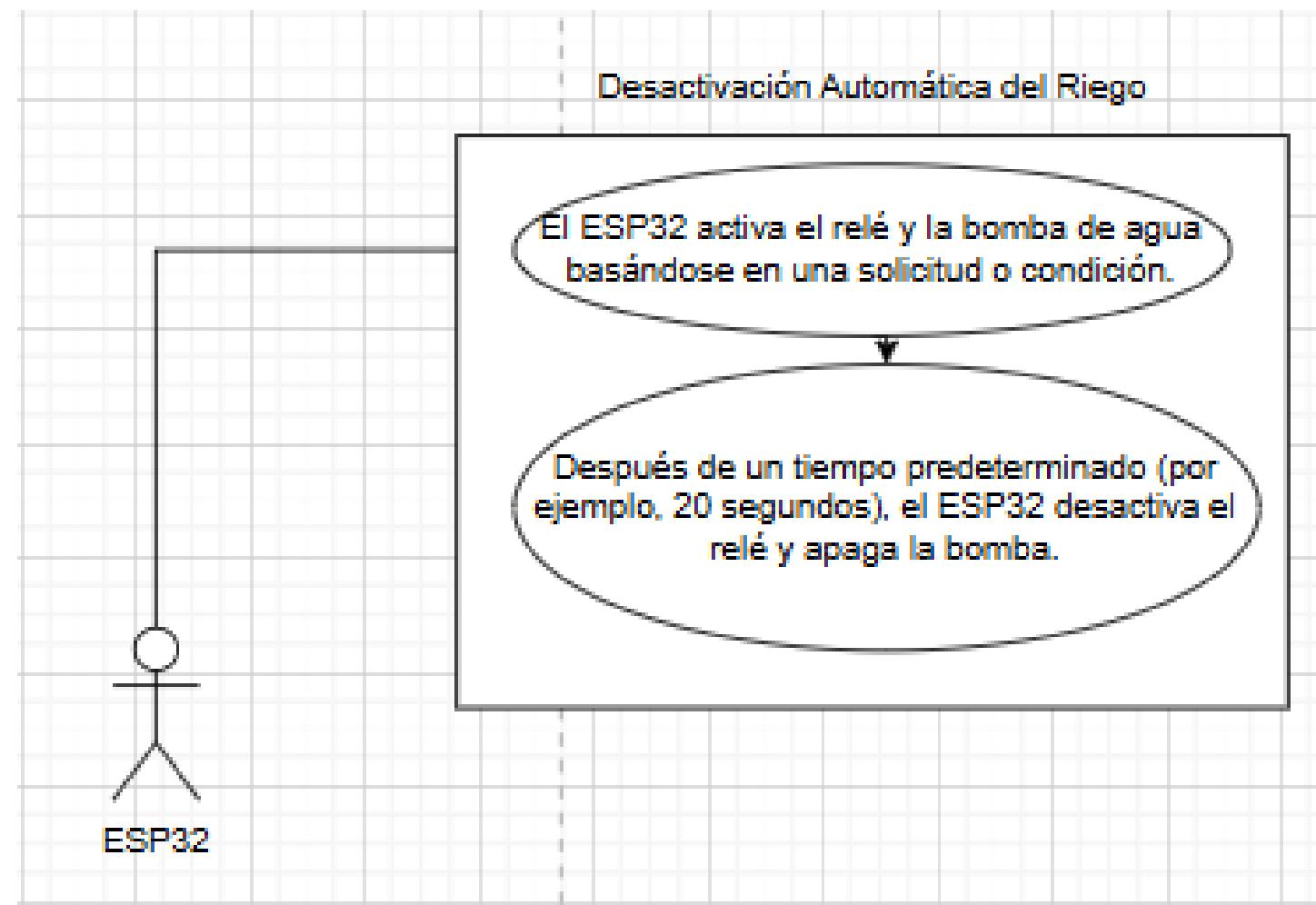
Monitoreo de Condiciones Ambientales



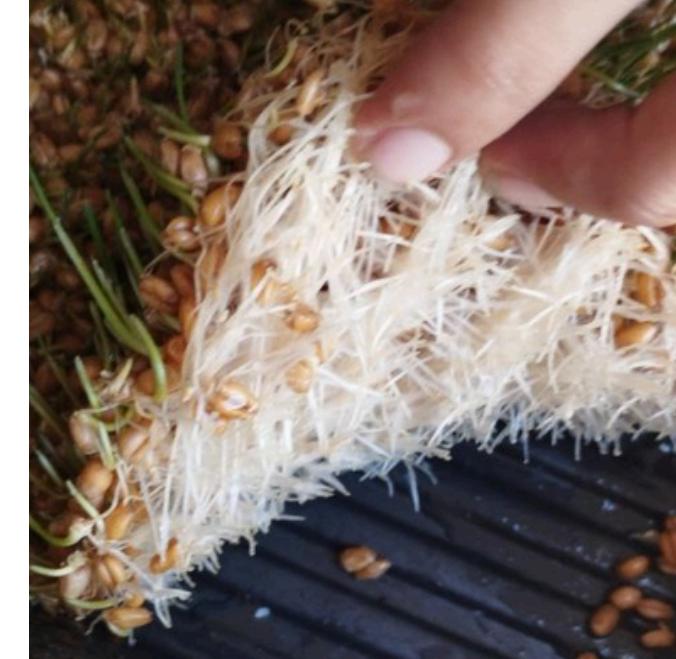
Activación Manual del Riego



Casos de Uso



Registros Fotográficos



Registros Fotográficos



Registros Fotográficos



Presupuesto

Presupuesto de componentes eléctricos y electrónicos (hardware)				
Componentes varios	Unidad	Cantidad	Precio Unidad (\$)	Precio total (\$)
ESP-32 Módulo wifi	c/u	1	\$ 32.900	\$ 32.900
Sensor de Humedad DTH-11	c/u	1	\$ 5.800	\$ 5.800
Sensor de Humedad Suelo	c/u	2	\$ 9.000	\$ 18.000
Relay 5V	c/u	1	\$ 8.000	\$ 8.000
Motor bomba de agua	c/u	1	\$ 16.000	\$ 16.000
Batería 12V	c/u	1	\$ 40.000	\$ 40.000
Protoboard	c/u	1	\$ 15.000	\$ 15.000
	c/u	1		\$ 0
		TOTAL		\$ 135.700

Presupuesto de infraestructura física.				
Componentes varios	Unidad	Cantidad	Precio Unidad (\$)	Precio total (\$)
Bandejas 40*30 plásticas	c/u	3	\$ 10.000	\$ 30.000
Tubos PVC	c/u	4	\$ 10.000	\$ 40.000
Tanque de agua	c/u	1	reciclado	reciclado
Mangueras	c/u	1	reciclado	reciclado
			TOTAL	\$ 70.000



Presupuesto

Módulo	Descripción	Horas	Costo por Hora	Costo Total (COP)	Sistema IoT (ESP32)					
Plataforma Web										
Dashboard de Visualización	visualización gráfica de datos históricos y en tiempo real.	10	\$ 20,000	\$ 200,000	Monitoreo de Sensores	Captura de datos ambientales mediante sensores (DHT11).	5	\$ 40,000	\$ 200,000	
Historial de Datos	Acceso a registros históricos de temperatura y humedad.	5	\$ 20,000	\$ 100,000	Control de Actuadores	Activación y desactivación del sistema de riego a través de un relé.	5	\$ 40,000	\$ 200,000	
Gestión de Alertas	Configuración y administración de alertas y notificaciones.	5	\$ 20,000	\$ 100,000	Programacion	placa microntrolador esp32	5	\$ 40,000	\$ 200,000	
Administración de Usuarios	Gestión de usuarios, roles y	5	\$ 20,000	\$ 100,000	Total Estimado		40		\$ 900,000	



Conclusiones

El prototipo **HydroHarmony IOT** basado en IoT facilita el control del cultivo de forraje verde hidropónico al:

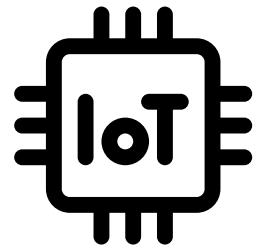
- Monitorear en tiempo real variables ambientales (humedad y temperatura).
- Proporcionar datos precisos para una gestión proactiva del cultivo.
- Automatizar el riego según las necesidades específicas de las plantas, optimizando recursos.
- Promover prácticas agrícolas sostenibles e innovadoras, mejorando la eficiencia y la producción.



Conclusiones

- Aprendizaje del FVH: Profundización en técnicas de cultivo de forraje verde hidropónico, requiriendo atención a detalles para el éxito.
- Impacto del IoT: Uso efectivo de tecnología IoT para monitoreo y control, mejorando la gestión del entorno de cultivo.
- Desafíos de Hardware:
 - Gestión energética y estabilidad del suministro.
 - Programación y calibración del ESP32, garantizando datos precisos.
- Desarrollo de Interfaz: Integración compleja de la página web con el hardware, necesaria para control remoto y visualización en tiempo real.

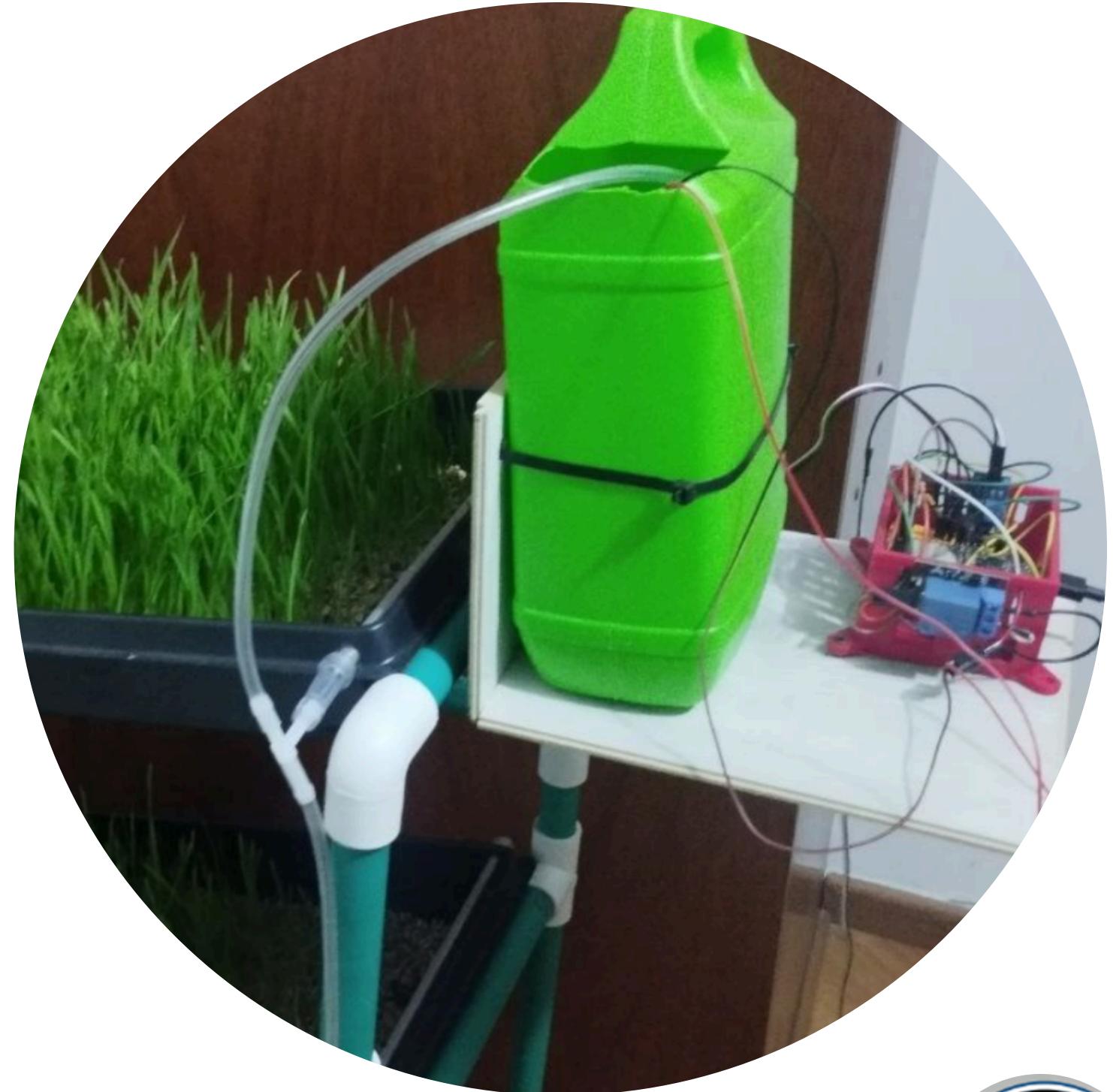




¡Muchas gracias!

Facultad de Ingeniería de Sistemas 2024

Nicolás Mahecha Pérez



FVH-IOT IoT

