

Sensorización del huerto doméstico

Un sistema de riego distribuido

El problema a resolver

- ▶ Agricultores con huertos pequeños o medianos no usan los recursos hídricos de forma eficiente.
- ▶ Conlleva una sobreexplotación de los recursos hídricos.
- ▶ Un mayor riesgo de desertificación.

Las soluciones actuales no son eficaces

- ▶ La mayoría son programadores de riego simples que riegan mientras hay precipitaciones.
- ▶ Los sistemas de riego complejos exceden el presupuesto del pequeño agricultor.



Trabajos relacionados

Nombre	Cometido	Riego	C.A.	Coste	Insumos
Garduino	Sensorización	No procede	Si		No
Huerta inteligente FABLAB Santander	Invernadero	Goteo	Si		No
Urban Kitchen Garden	Invernadero hidropónico	Hidroponía	Si		No
Skydrop	Programador riego	Aspersión	No		No
Jardibric aquaflow	Programador riego	Múltiple	No		No
Cloud wifi hozelock	Programador riego	Múltiple	No		Si
Aspara	Huerto hidropónico	Hidroponía	No		Si
Homy Jungle	Planta personalizada	No procede	No		No
FarmBot	Huerto CNC	Aspersión	Si		No
Cell One	Microgreen	Aeroponía	No		No
Aquapioneers	Sistema híbrido	Aquaponía	Si		Si
Myfood	Sistema híbrido	Aquaponía	Si		No

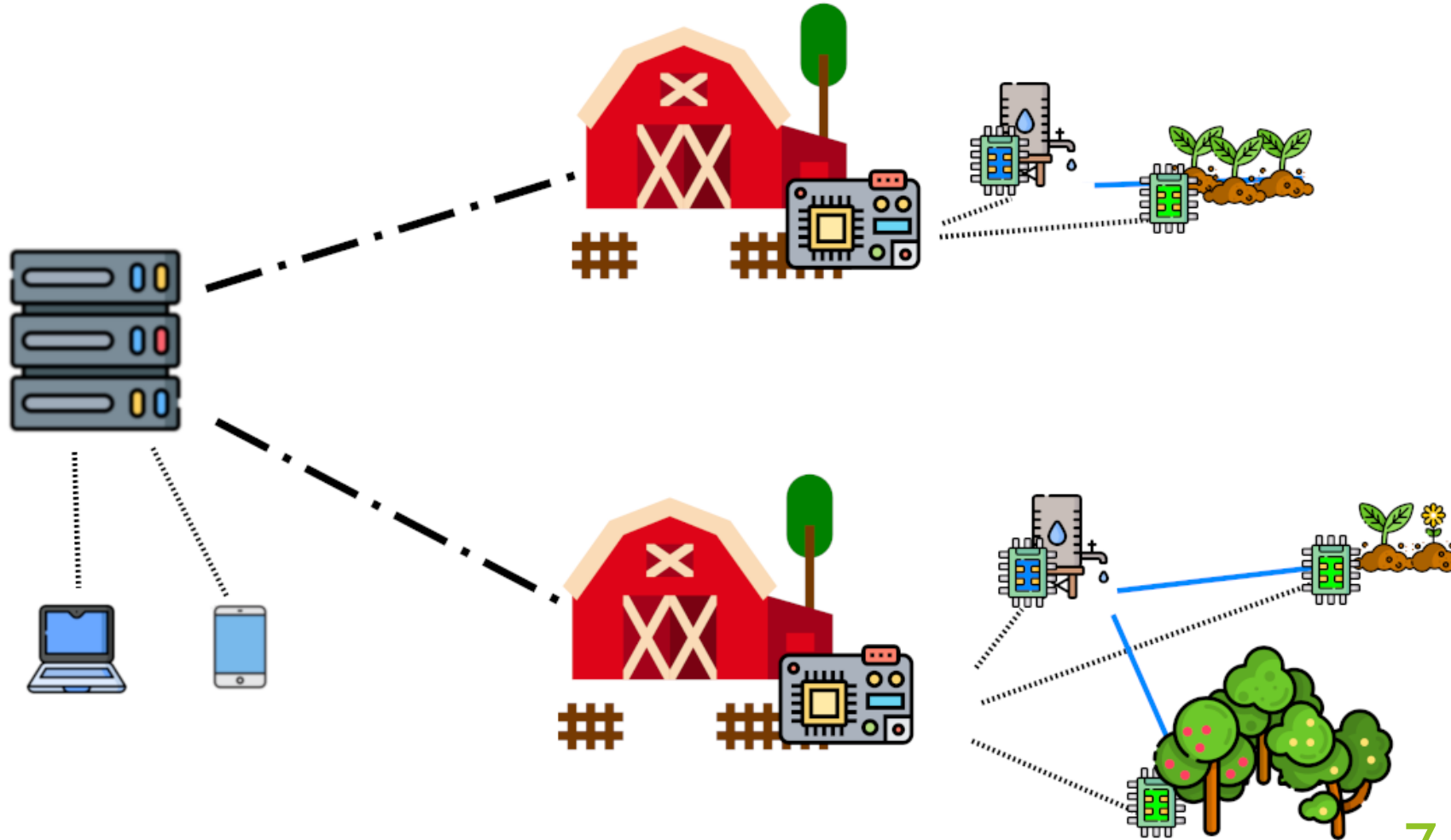
Planificación

	TIEMPO ESTIMADO	TIEMPO REALIZADO
Introducción	30 horas	30 horas
Análisis de requisitos	30 horas	50 horas
Planificación	10 horas	2 horas
Diseño de arquitectura	40 horas	90 horas
Implementación	150 horas	120 horas
Pruebas	20 horas	20 horas
Total		312 horas

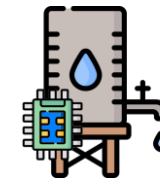
Objetivo a alcanzar

- ▶ Crear un sistemas de riego inteligente que sea de software y hardware libre.
- ▶ Que garantice el uso eficiente de los recursos hídricos disponibles.
- ▶ Centrado en la facilidad de uso y que aporte versatilidad para adaptarse a diferentes entornos como huertos domesticos, huertos urbanos comunitarios o incluso pequenas huertas.

Solución propuesta



Resumen elicitación de requisitos



Actores del sistema

- ▶ Agricultor
- ▶ Autómata

Objetivos del sistema

- ▶ Gestión de zonas de riego
- ▶ Gestión meteorológica
- ▶ Gestión de usuarios

Requisitos de información sobre

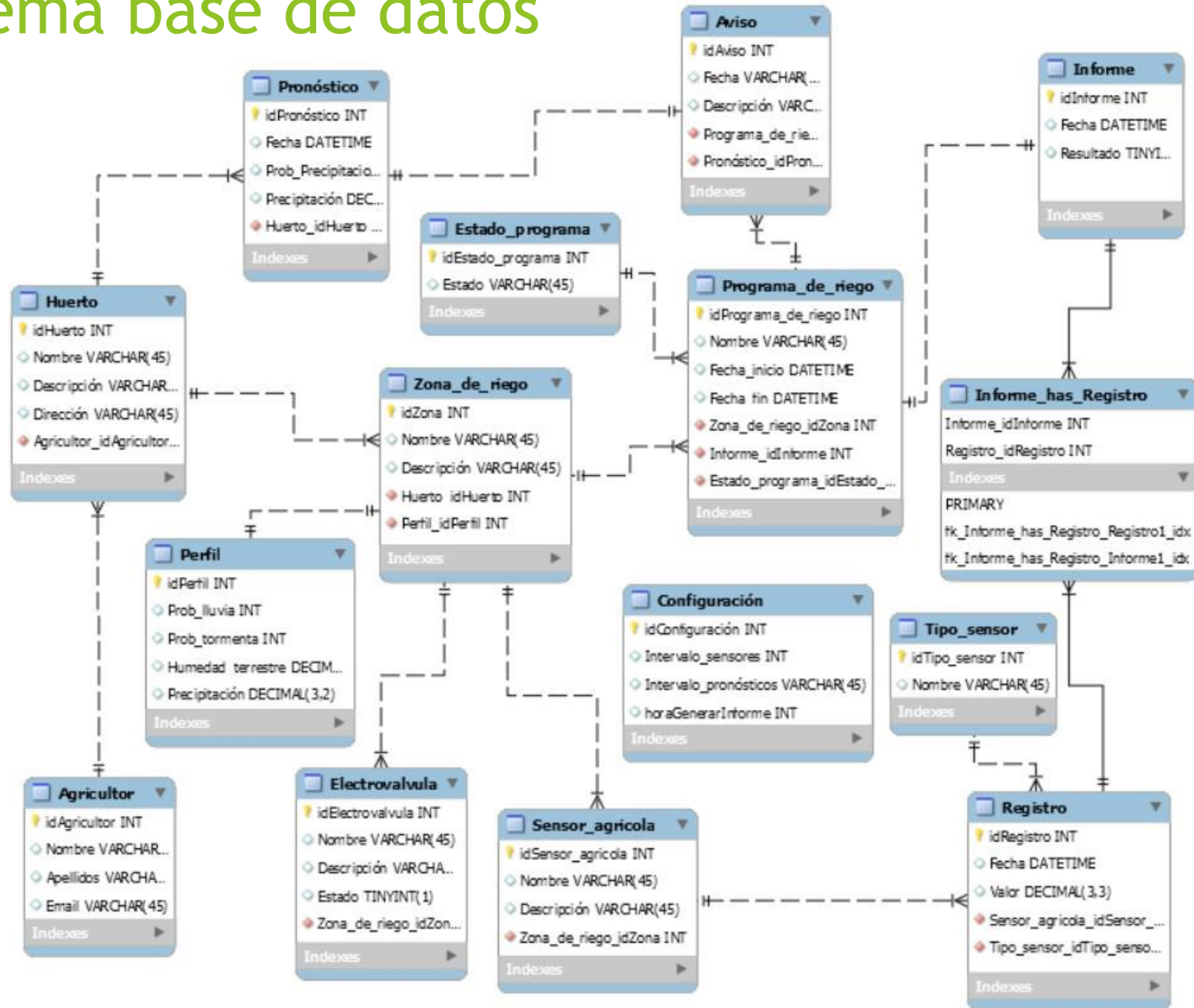
- ▶ Zonas de riego
- ▶ Sensor agrícola
- ▶ Electroválvula
- ▶ Programas de riego
- ▶ Pronósticos
- ▶ Avisos
- ▶ Informes

Casos de uso relevantes

- ▶ Proceso de riego
- ▶ Cancelar riego
- ▶ Abortar riego
- ▶ Ver informes
- ▶ Ver avisos



Esquema base de datos



Canales de comunicación



- ▶ ¿Cómo comunicar los sensores con el servidor?
- ▶ Existen múltiples tecnologías como lora/sigfox, wifi, bluetooth.
- ▶ Para el proyecto se ha optado por wifi, ya que brinda la posibilidad de usar múltiples protocolos de comunicación.

Protocolos de comunicación

- ▶ Se ha estudiado diferentes formas de transportar la información.
- ▶ Tras estudiarlo bien se optó por MQTT.

{ **REST:API** }

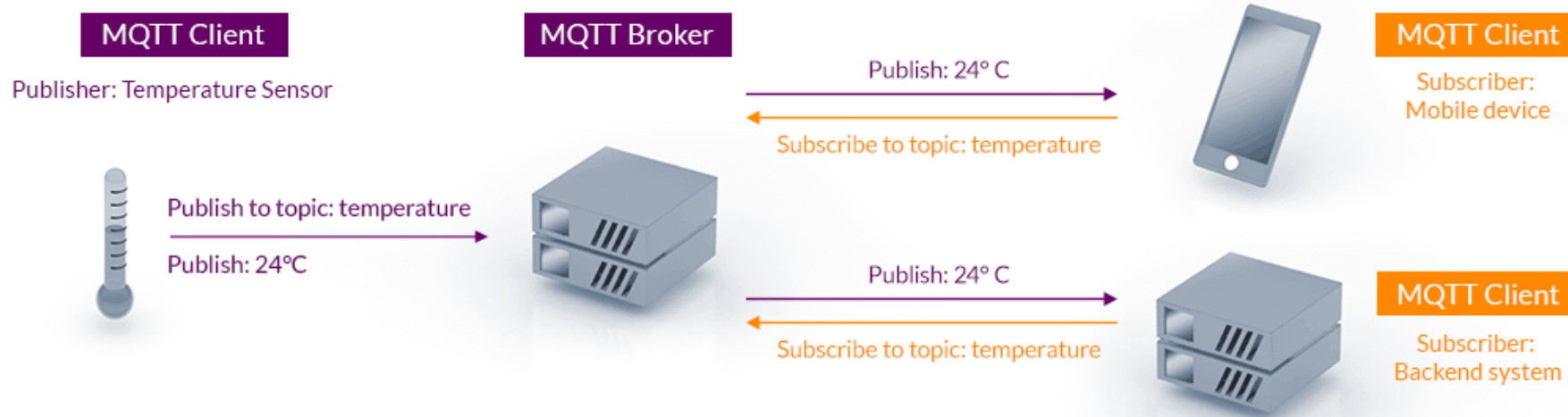


Websockets

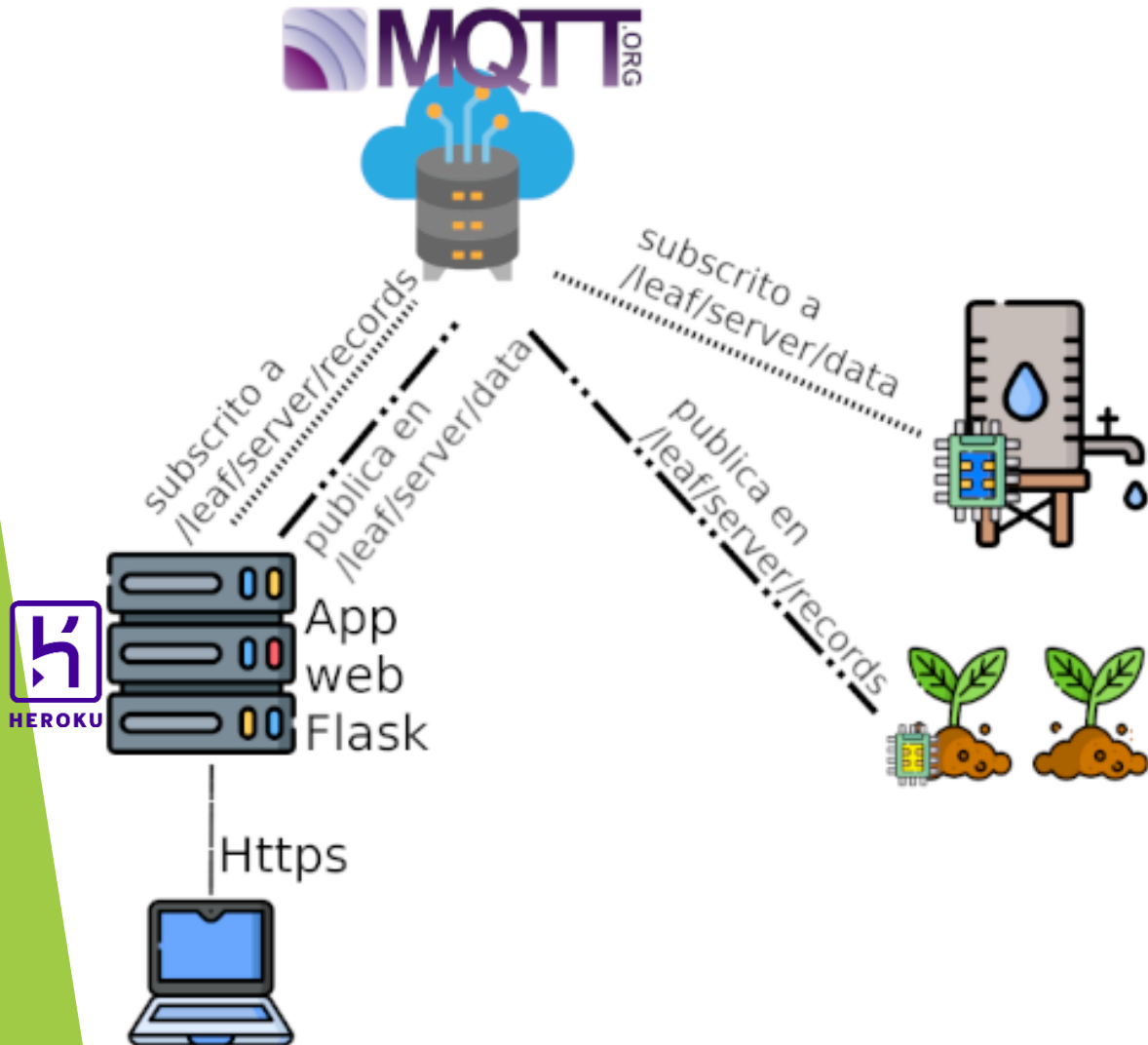


Message Queing Telemetry Transport

- ▶ Protocolo de comunicación M2M, enfocado al internet de las cosas
- ▶ Basado en TCP/IP
- ▶ Mensajería push con patrón publicador/suscriptor (pub-sub)
- ▶ Mensajes organizados en *topics* de forma jerárquica



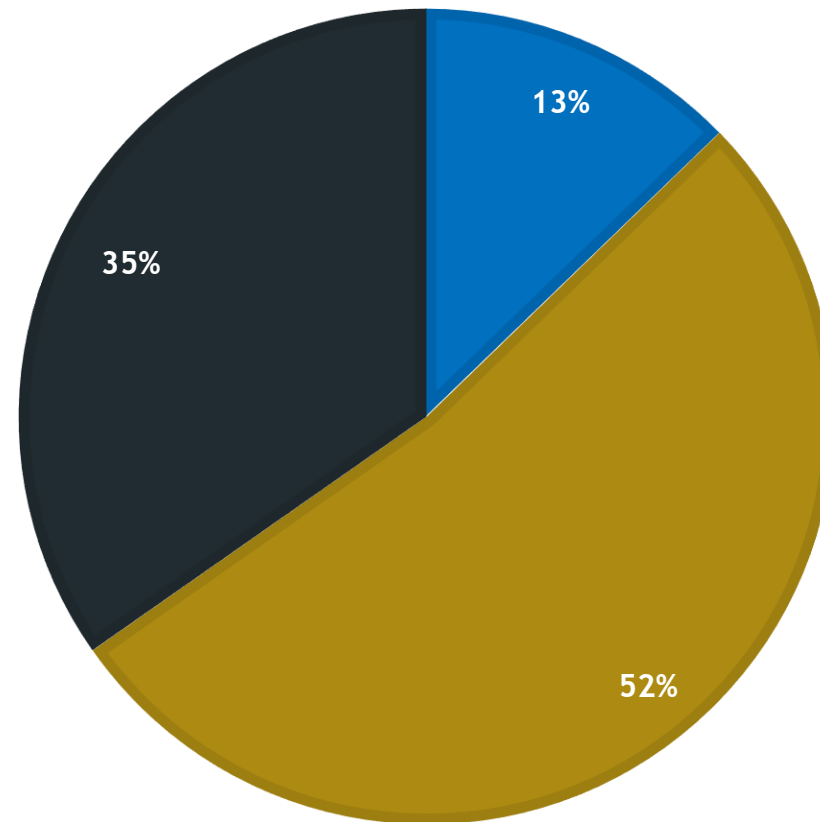
Resultados obtenidos



- ▶ El sistema se ha realizado en Python y Arduino/C++
- ▶ Cuenta con un app web alojada en heroku y un broker mqtt público.
- ▶ Además de placas microcontroladoras con antenas wifi para los sensores y actuadores
- ▶ Trás investigar se optó por suprimir el concentrador y comunicar directamente los sensores y actuadores con la app web

Panorámica del sistema

■ Arduino/C++d **263** ■ Python **1083** ■ HTML/CSS/JINJA2 **716**

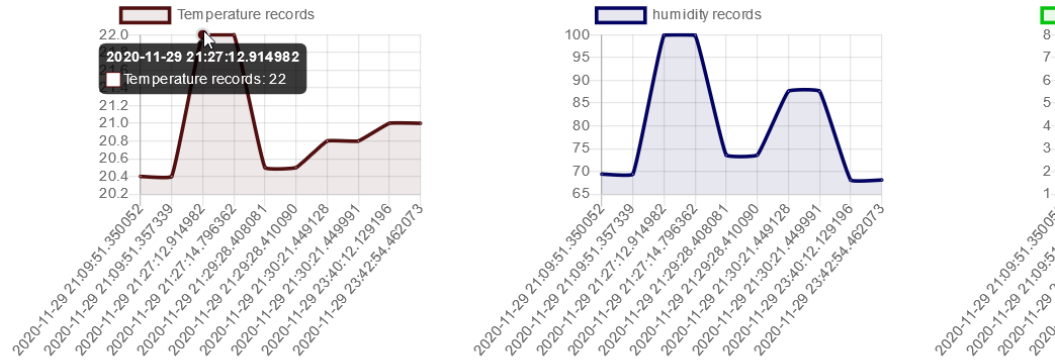


Plataforma web

Home Pricing Your farm Irrigation zones Irrigation program (Logout)

Statistics of the irrigation area

Aromáticas



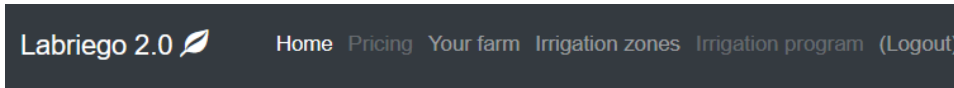
Activate solenoid valve



- La plataforma web cuenta con zonas de riego las cuales tienen asociadas un sensor agrícola y una electroválvula
- Está desarrollada en Python con el microframework Flask
- Con la extensión Flask-mqtt se comunica con los sensores y actuadores de los huertos a través de un servidor MQTT público

Plataforma web

- ▶ Aquí se muestra el formulario para crear una zona de riego, directamente se añade los identificadores del sensor agrícola y la electro-válvula que se asociará a la zona
- ▶ Mediante un identificador único como la mac del sensor o actuador se determina a que zona de riego pertenece en el caso de que no existiera se descarta.
- ▶ Para estandarizar los mensajes se utiliza un formato json



Create a irrigation zone

Name

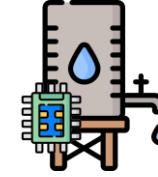
Description

MAC agricultural sensor

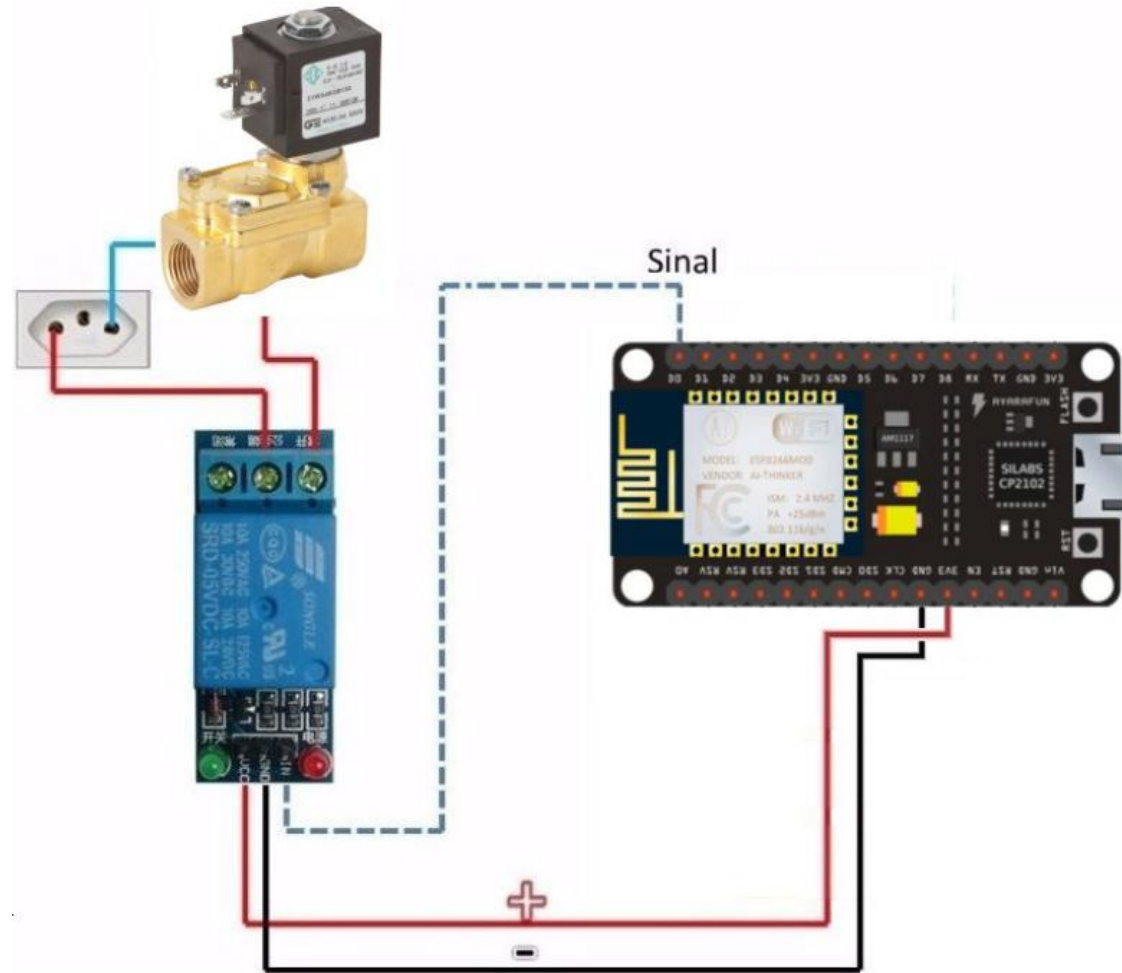
MAC solenoid valve

Submit

Electro-válvula



- Se utilizo un relé a modo de interruptor.
- Gobernado por un nodeMCU placa que integra un ESP8266 un chip con antena wifi

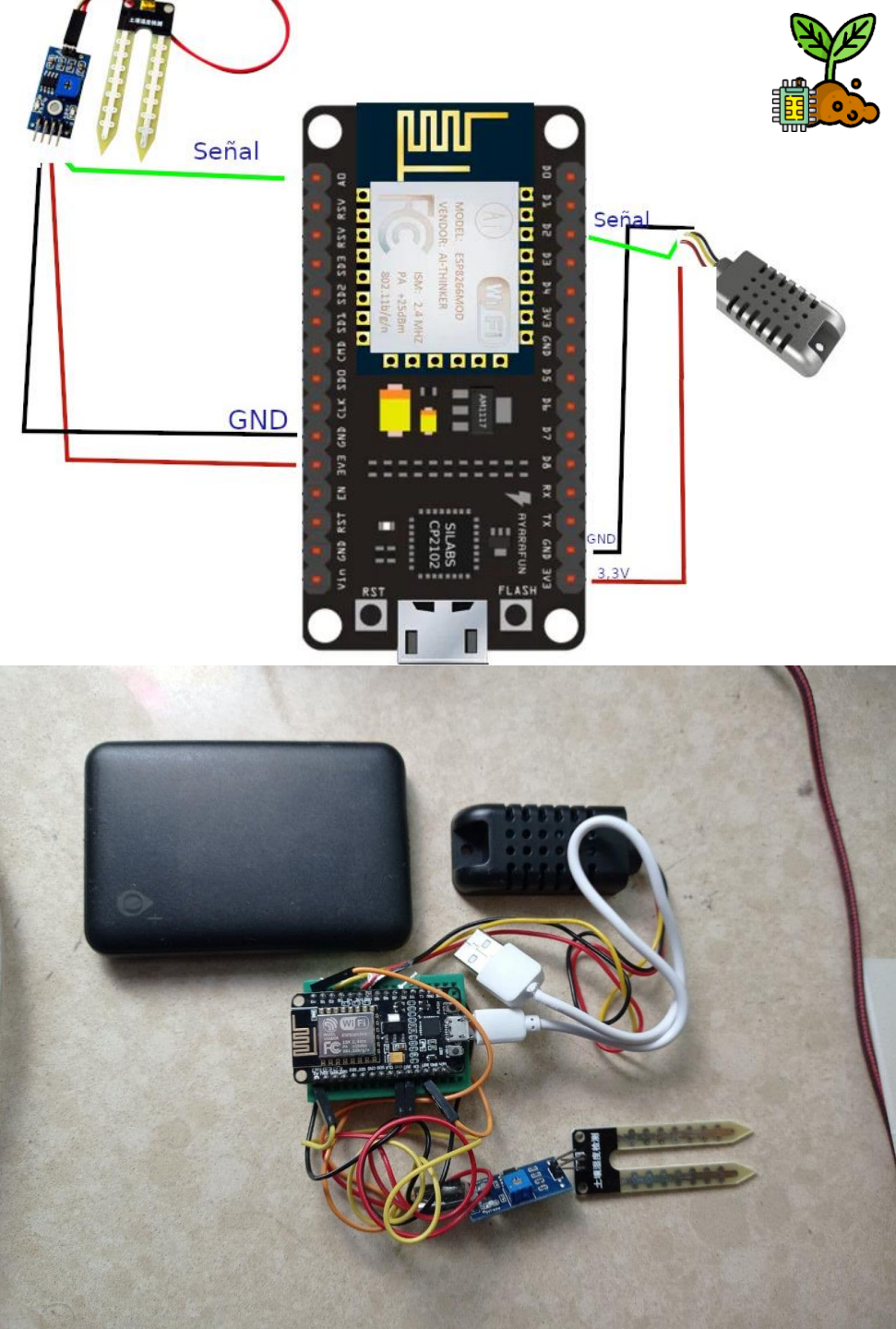


Sensor agrícola

- ▶ Compuesto de sensor de temperatura y humedad ambiental DHT21 y sensor de humedad terrestre
- ▶ Para la alimentación se utilizó una batería de 6000mah
- ▶ Para ahorrar batería se optó por controlar el tiempo de ejecución de la placa mediante hibernación de la misma
- ▶ La información se envía en formato json, para ello se utiliza la librería arduinoJSON



18/21





Sensor agrícola

Aquí se muestra el prototipo del sensor agrícola

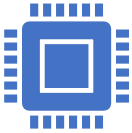


Pruebas realizadas

- ▶ En el primero se ha probado la integridad de la base de datos.
- ▶ En el segundo el acceso anónimo a zonas que requieren acceso con un usuario logueado.
- ▶ En la tercera se ha probado el acceso a un apartado sin el rol de usuario correcto.
- ▶ En la cuarta se ha probado el acceder a un apartado que no existe.
- ▶ En la quinta se ha probado el acceso a un apartado sin el rol de administrador.
- ▶ En la sexta se ha probado el envío y recepción de mensajes a través de un servidor MQTT para su posterior guardado en la base de datos.

```
82 > def test_make_unique_email(self): ...
92
93 > def test_anonymous_access(self): ...
100
101 > def test_unauthorized_access(self): ...
104
105 > def test_show_zone_not_exists(self): ...
109
110 > def test_unauthorized_access_to_admin(self): ...
115
116 > def test_mqtt_received(self): ...
```

Conclusiones



A lo largo del proyecto he aprendido desarrollo en microcontroladores, arquitectura de sistemas distribuidos, protocolos de comunicación, diferentes tipos de telecomunicaciones. Esto me ha llevado a darme cuenta de las posibilidades de aplicación que tiene una placa de desarrollo con antena wifi de no más de un euro.



Resulta paradójico la cantidad de dificultades que tuve durante el desarrollo del mismo, dar mil vueltas a una misma cosa para acabar en el punto de partida. Aunque no sin antes haber adquirido nuevas competencias con las que realizar el mismo viaje más rápido.



Con lo que me quedo del proyecto es con las posibilidades que hay, con las formas tan diferentes de interconectar cosas. De dar sentido a un dato, de saber como poder abrir una conexión desde un servidor externo a un cliente detrás de un router nat de tres maneras diferente. De la necesidad de trabajar en grupo para poder llegar más lejos.

- ▶ MQTT brinda un mundo de posibilidades de interconexión
- ▶ La comunicación wifi consume demasiada electricidad para IoT, se debe buscar alternativas como lora o sigfox a largo plazo.
- ▶ Hay un mercado emergente de placas de desarrollo de bajo coste con la que se puede realizar multitud de aplicaciones