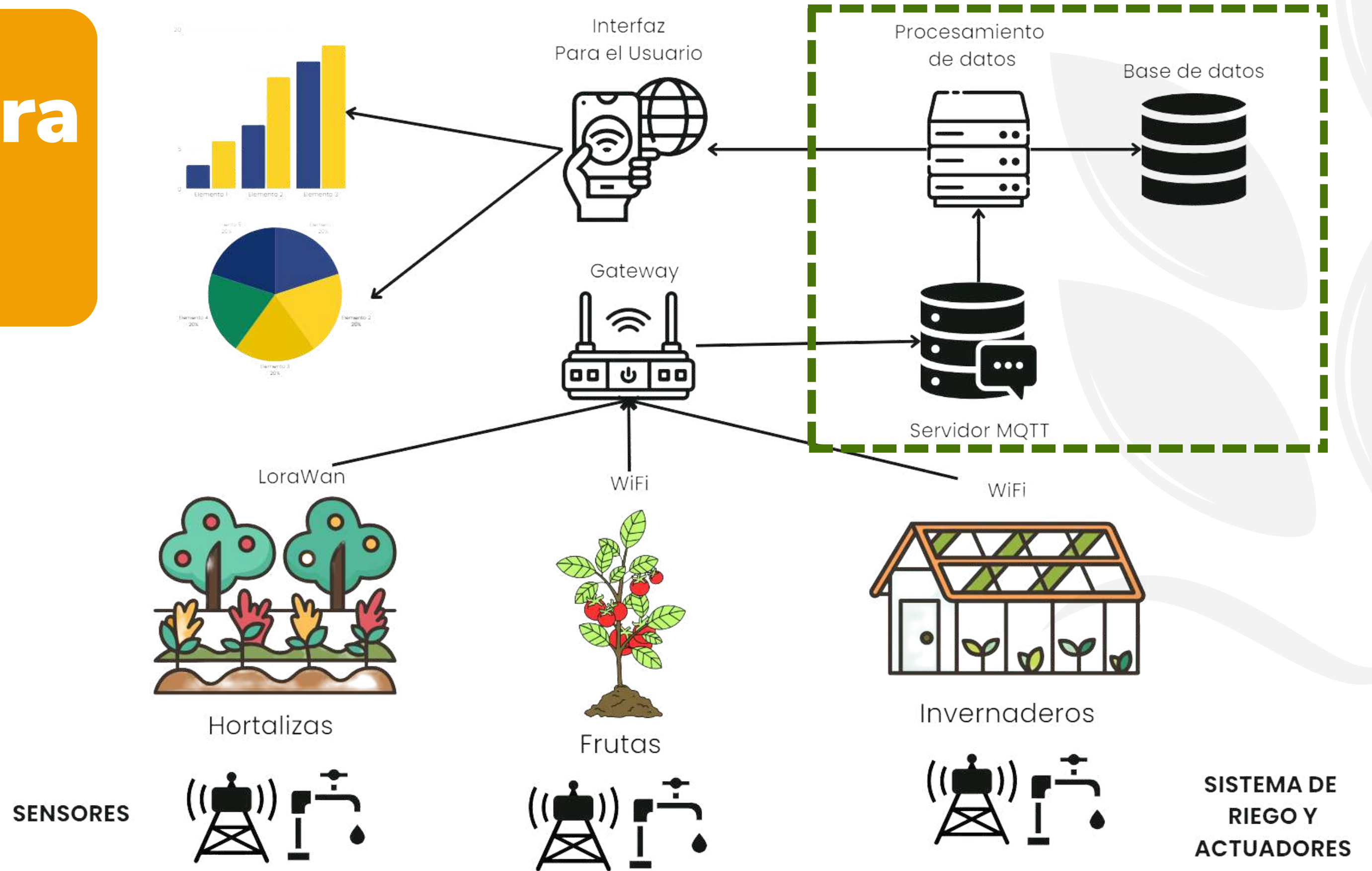




SMART FARMING

CREACIÓN DEL BACKEND PARA UNA APLICACIÓN WEB DE UN
PROTOTIPO DE MONITOREO DE AGRICULTURA INTELIGENTE UTILIZANDO
EQUIPOS DE DESARROLLO DE BAJO COSTO Y TECNOLOGÍA DE
COMUNICACIÓN LORAWAN Y WIFI.

Arquitectura completa





GRANJA URKUWAYKU



Objetivos



01

BASE DE DATOS

Diseñar e implementar el servidor de base de datos

02

MQTT

Enlazar servidor MQTT con servicio de capa de acceso de datos (Backend) para la recolección y preparación de los datos en crudo.

03

API

Diseñar APIs de comunicación para entrega de datos a la aplicación de frontend

04

DOCUMENTACIÓN

Generar documentación de las actividades desarrolladas

Entorno utilizado



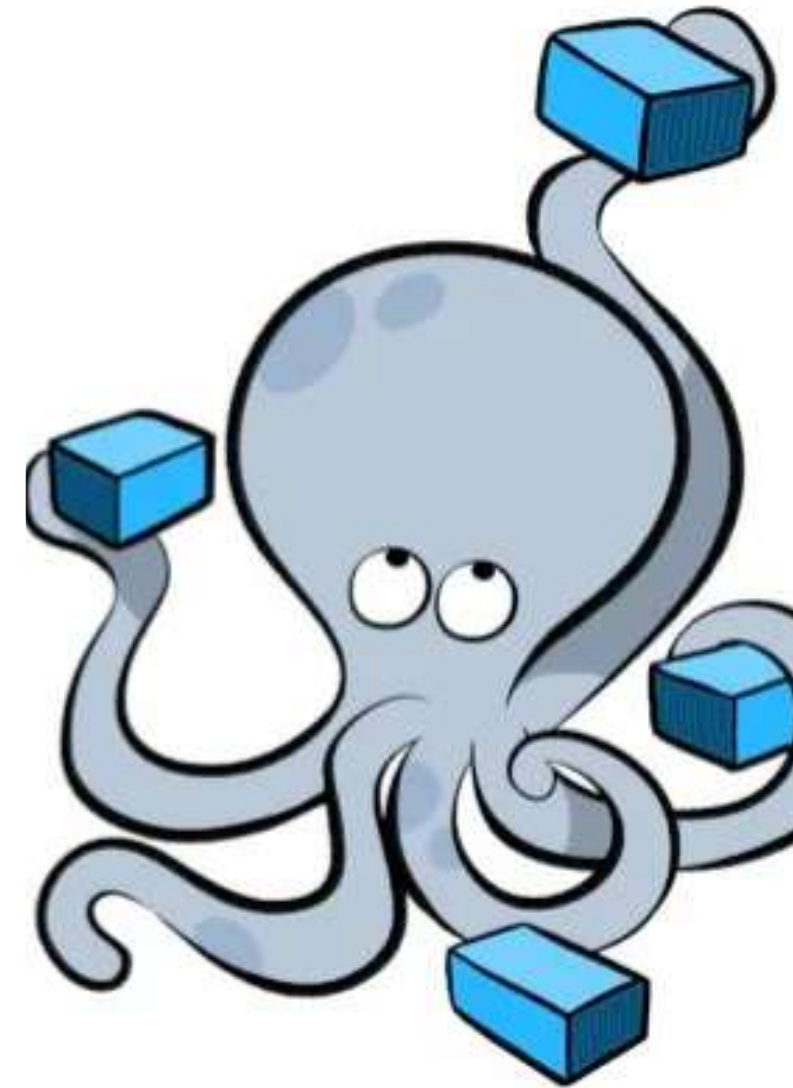
Computador

Macbook pro 13-pul, inicios del
2011

MacOS High Sierra (2017)

Docker y docker-compose

Docker

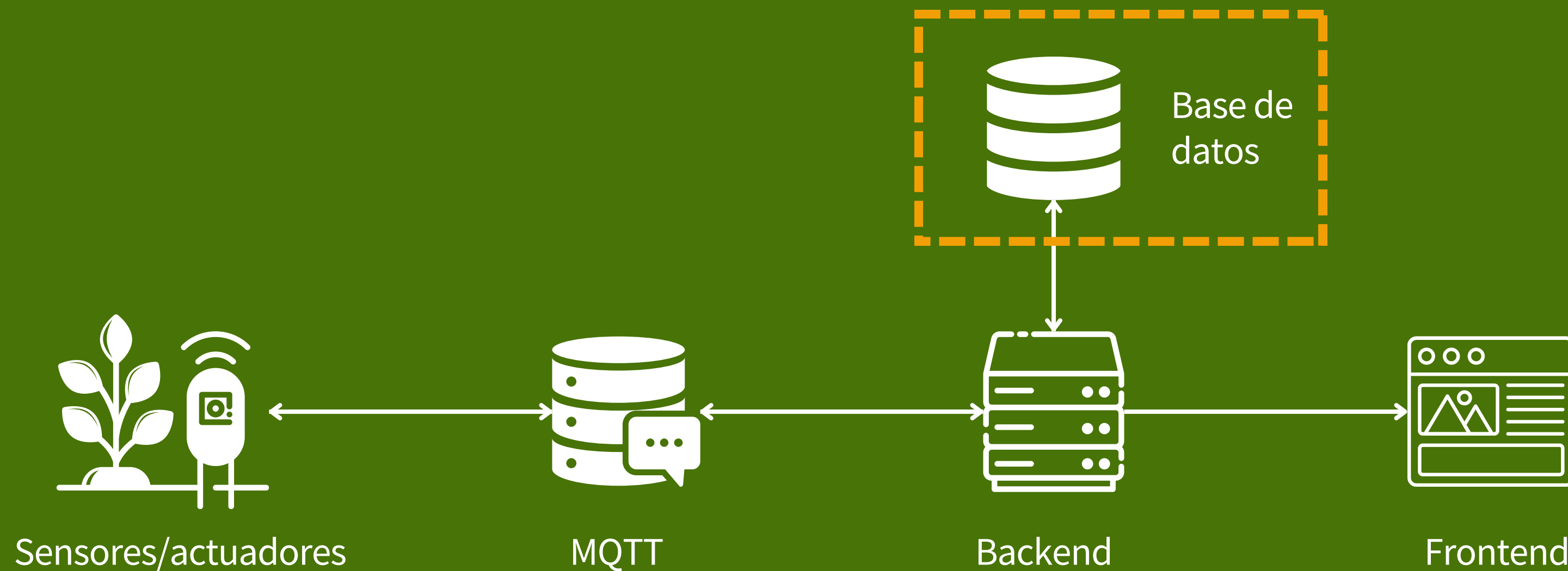


Visual Studio

Editor de Código

01

Base de datos



Modelos



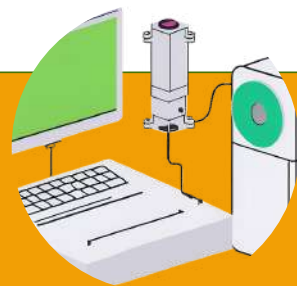
”

Actuador



”

Mensaje



”

Regla



”

Sensor

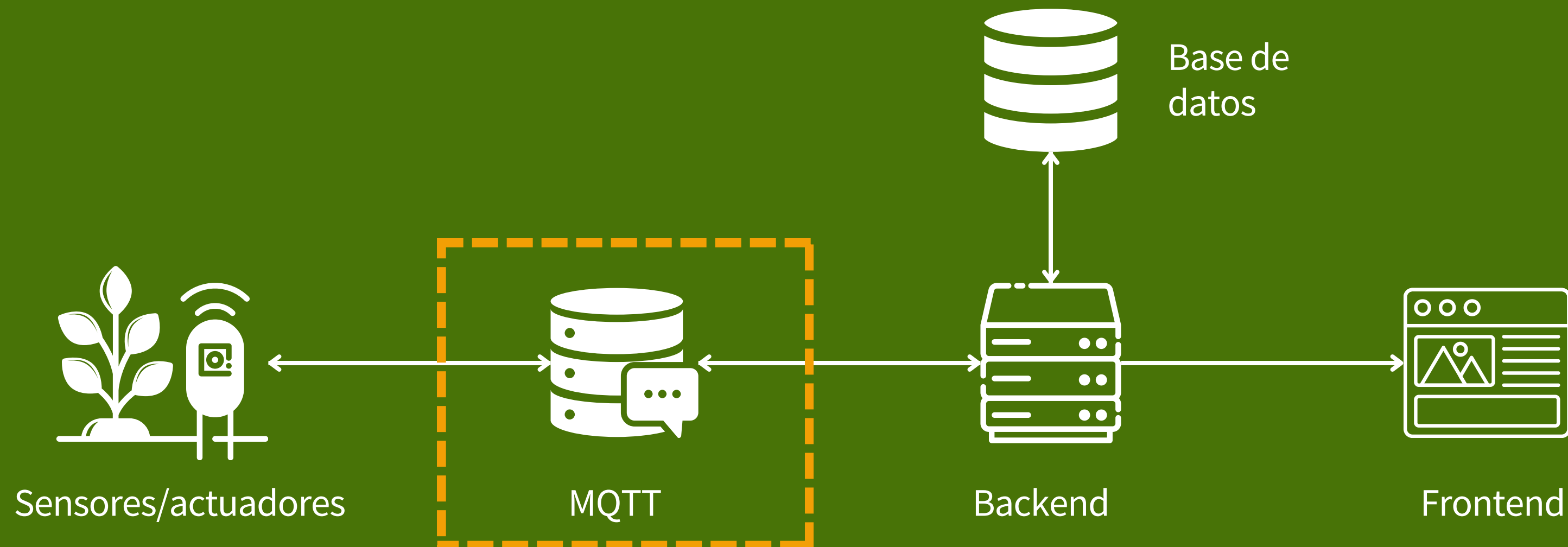
Servidor Mongo DB y problemas

```
mongodb:
  container_name: mongodb
  image: mongo:4.4.11
  restart: always
  volumes:
    - ./mongodb_data:/data/db
  networks:
    - mqtt_net
  healthcheck:
    test: ["CMD", "mongo", "--eval", "db.adminCommand('ping')"]
    interval: 30s
    timeout: 10s
    retries: 3
```

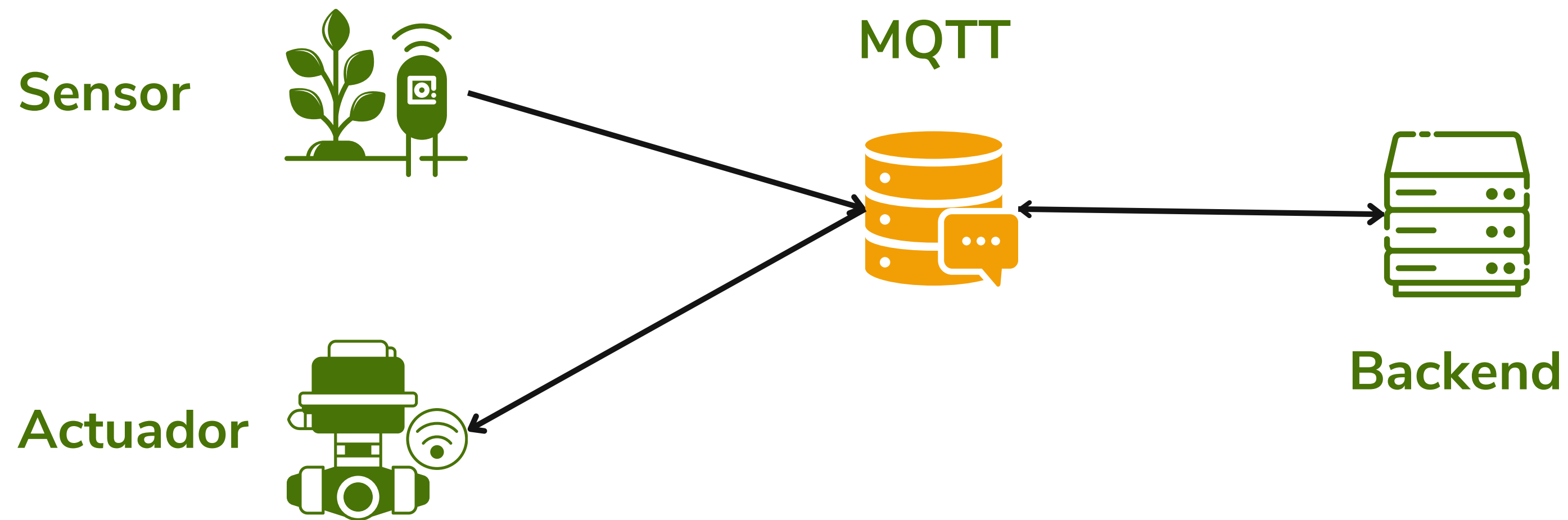


02

MQTT

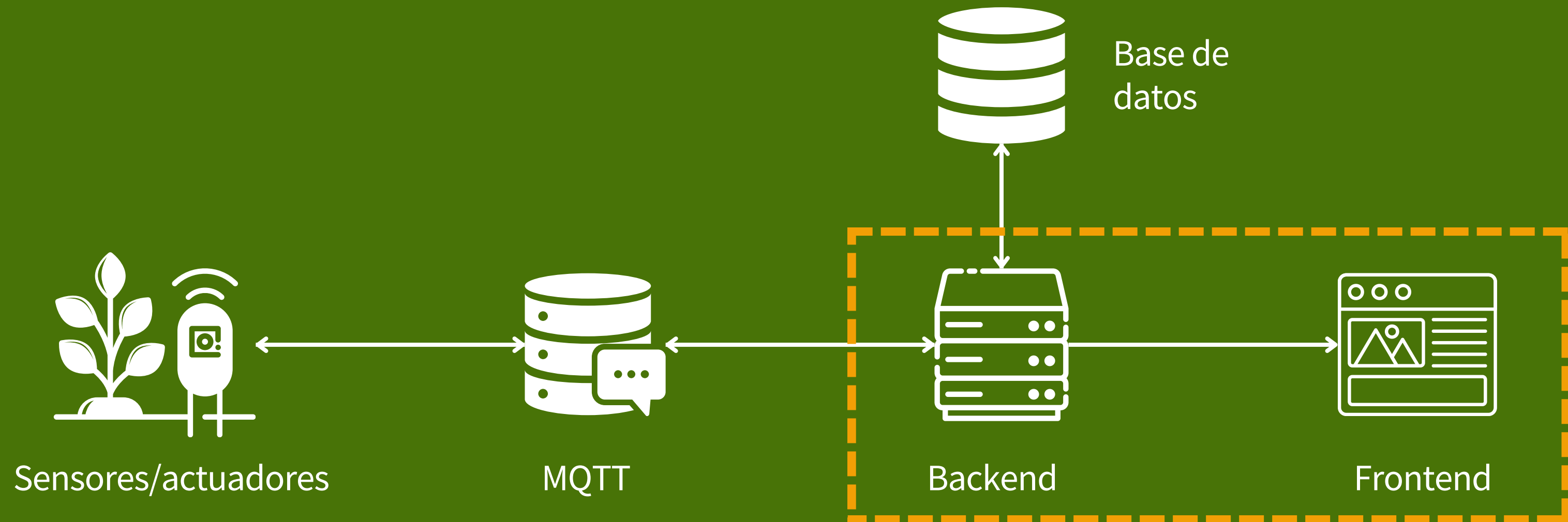


MQTT



03

API



Rutas

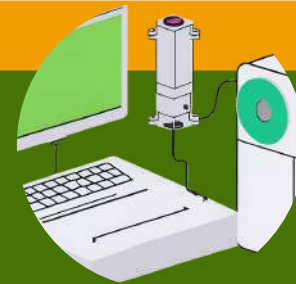


Actuadores



Mensajes

express



Reglas



Sensores

04

Scrum

Documentación



Visión

Manejo eficaz de los requisitos cambiantes del negocio y flexibilidad para adaptarse



Estructura

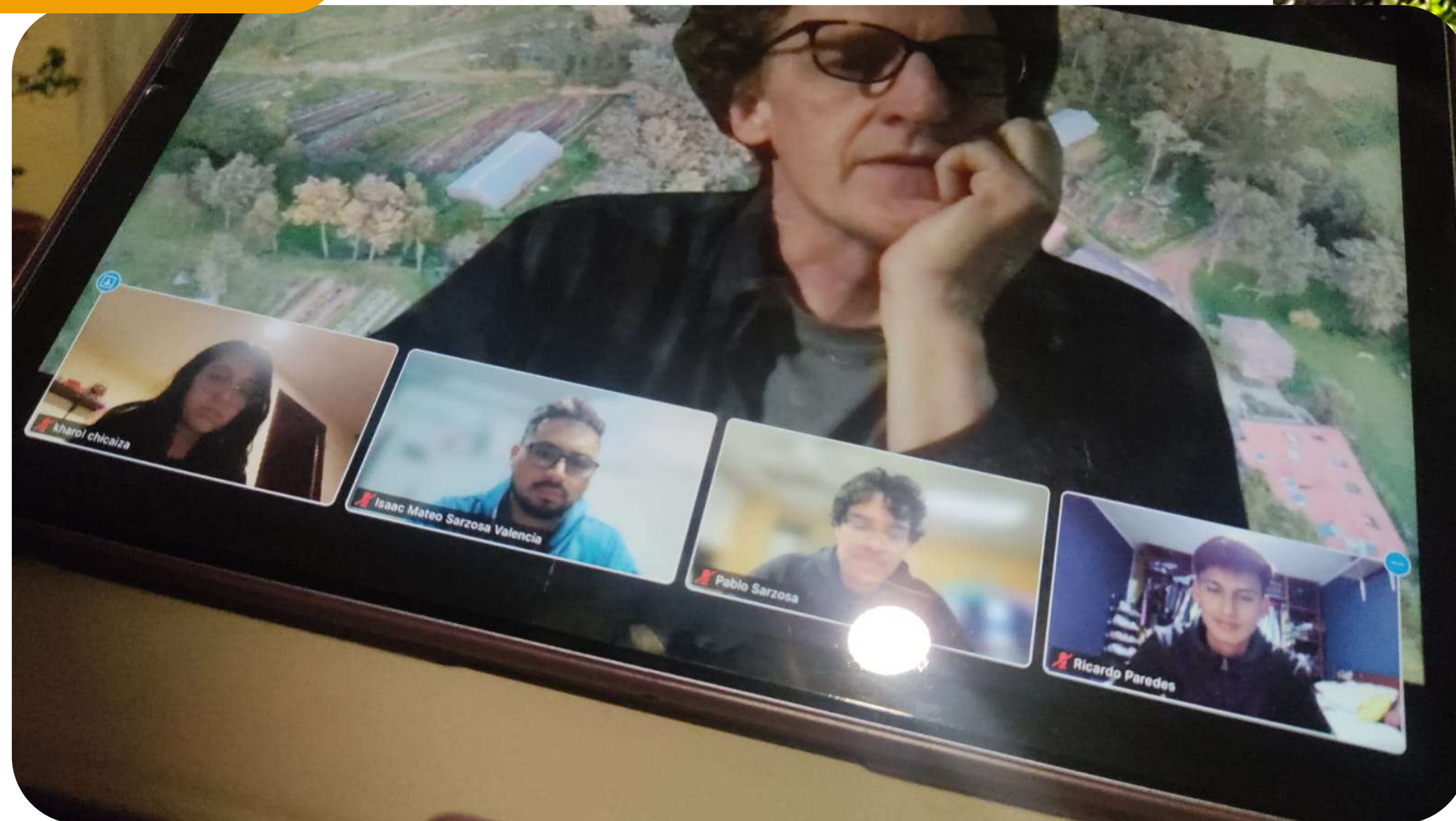
Los artefactos y reuniones de Scrum proporcionan una estructura clara y ágil en el desarrollo de software

Requisitos
Sprint 1

Requisitos
Sprint 4
(extra)



Reuniones



Estación climática

Extra



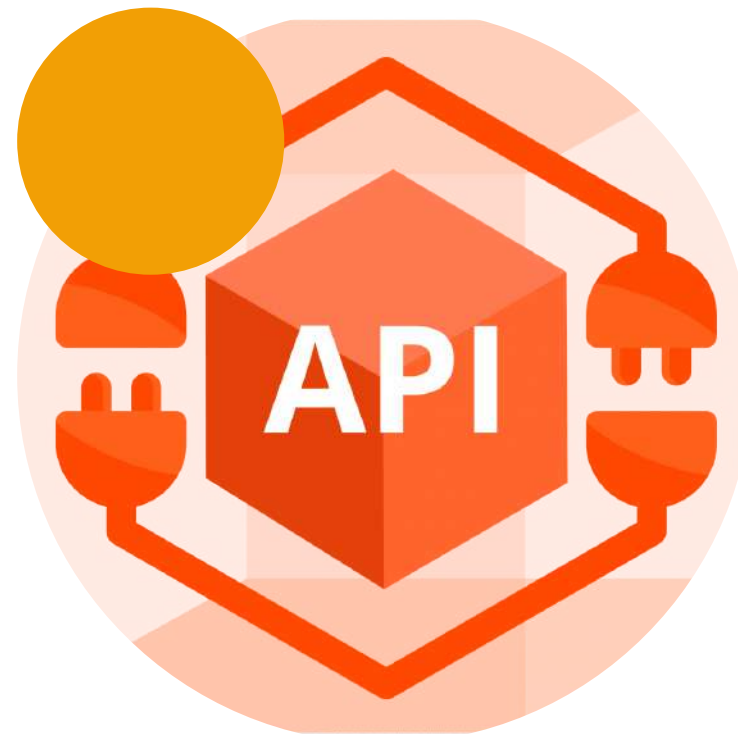
Resultados

Pruebas de carga



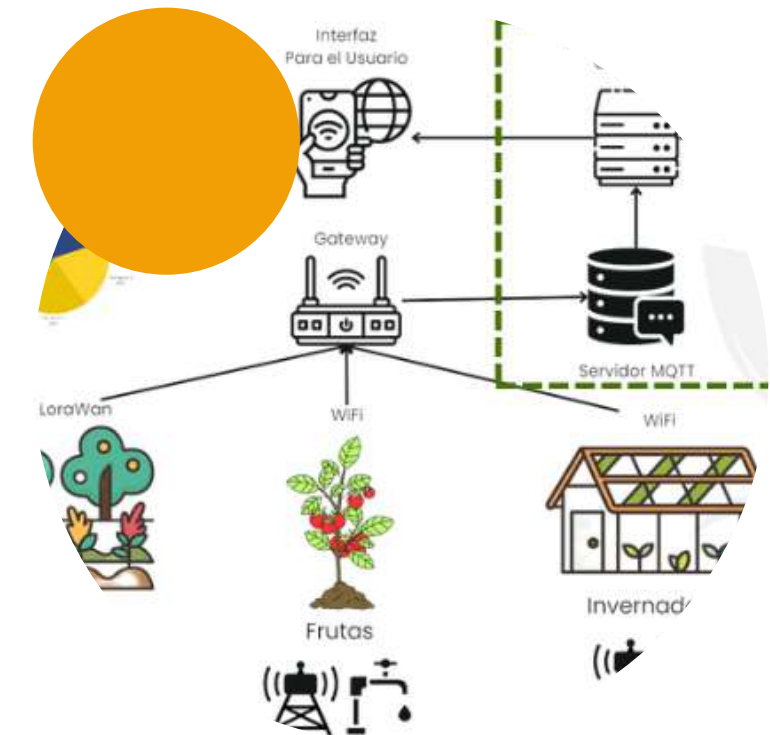
MQTT

10 Suscripciones por minuto
10 Publicaciones por minuto
Suma progresiva



API

10 Llamadas a la API por minuto
Progresiva

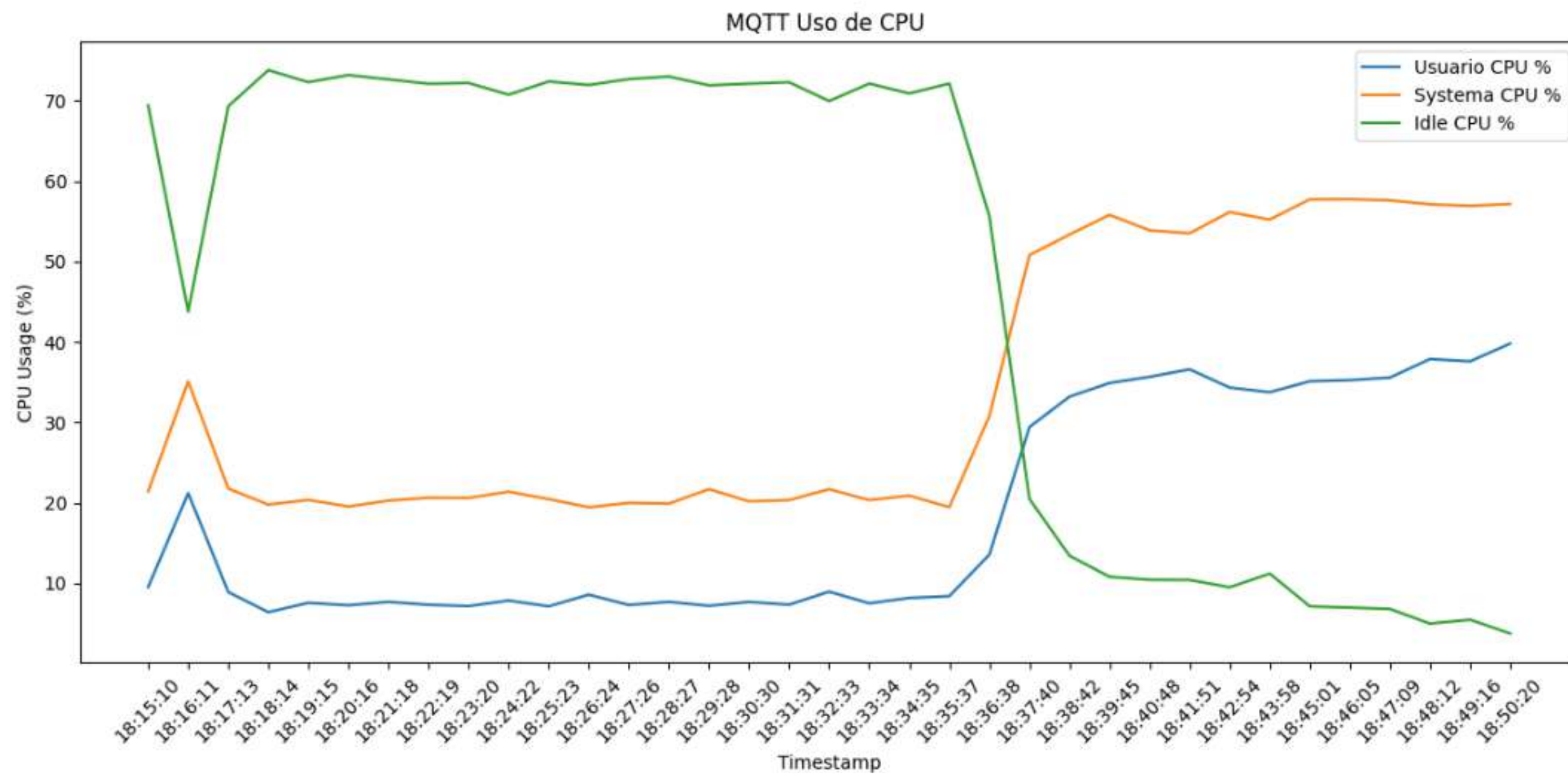


Funcionalidad completa

Combinación de pruebas MQTT y API

MQTT

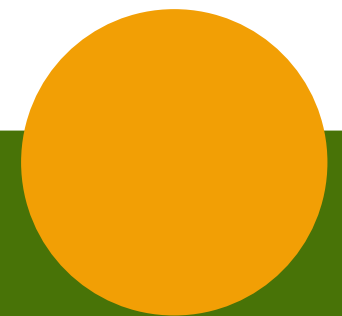
Análisis



CONEXIONES

Cambio: 220

Desconexión: 350

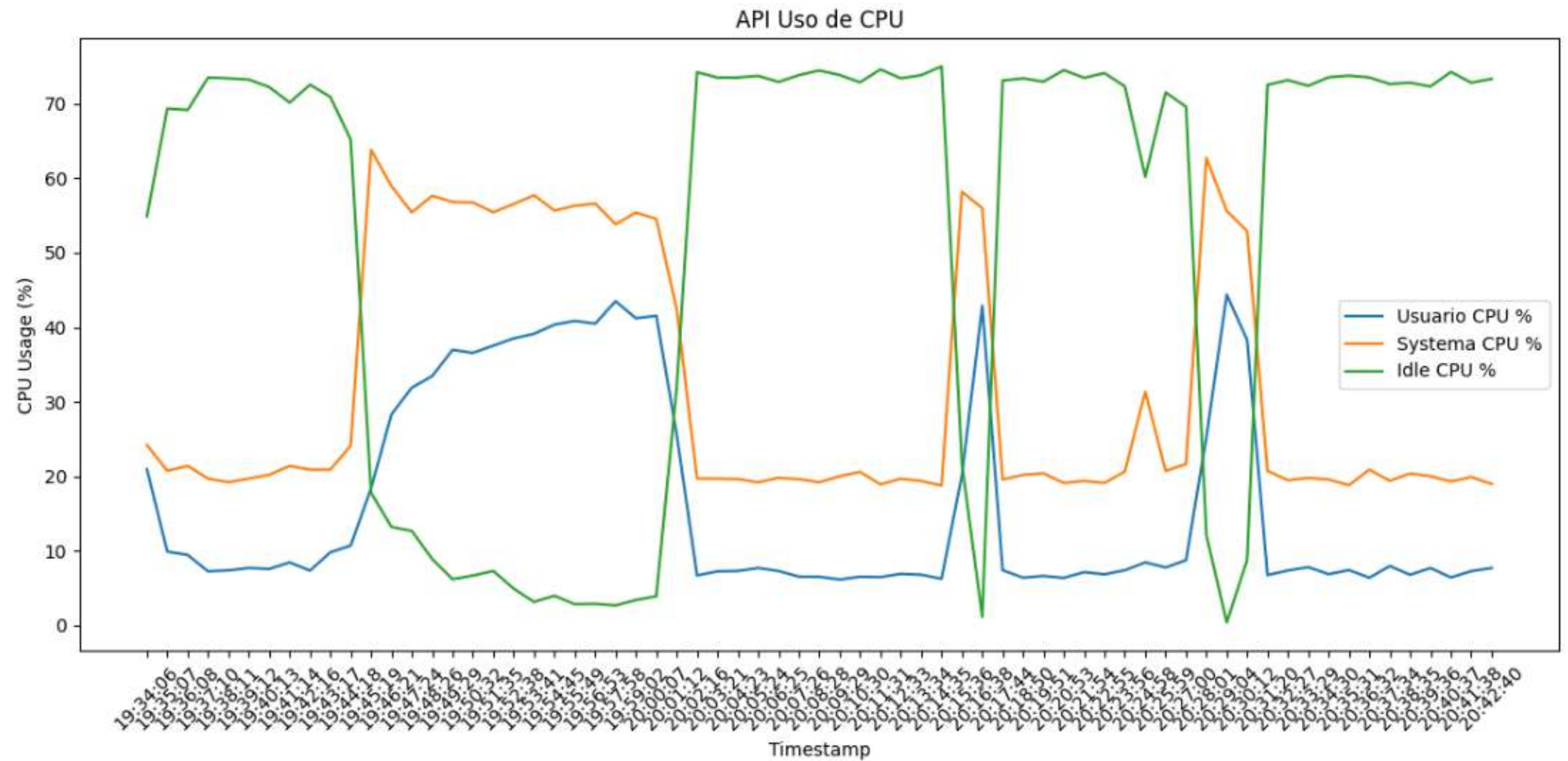


API

Análisis

LLAMADAS

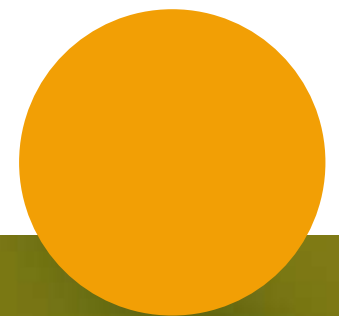
Desconexión: 1382





Funcionalidad completa

ERROR después de las 20 llamadas a la API



Conclusiones

Docker compose

EFICIENTE

- Diferentes configuraciones
- Más pruebas
- Más rápido

MQTT - API

FUNCIONAL

- Operando correctamente
- Datos en tiempo real

MONGO DB

FUNCIONAL

- Escalable y flexible
- Inclusión progresiva de componentes
- Compatibilidad

Pruebas

CONSTANTES

- Correcta integración
- Flexibilidad en el desarrollo

Recomendaciones

Cambio a servidor

- Más adecuado
- Servidor en la nube
- Migración fácil

Diseño flexible

- Cambios sin reestructuras
- Modificaciones según la necesidad

Documentación

- Detallada y actualizada
- Mantenimiento y evolución

Comunicación

- Clara y efectiva
- Identificar problemas
- Adaptado a necesidades

11/09/2024



GRACIAS