

Aplicación web para la gestión de módulos hidropónicos inteligentes

Autor: José Antonio López Pérez

Tutores: Lidia Sánchez González,
Jesús Fernández Fernández

Índice

01

Descripción del problema

02

Descripción de la solución

03

Evaluación

04

Gestión del proyecto software

05

Demostración de funcionamiento

06

Conclusiones y trabajos futuros

01

Descripción del problema

Agricultura sostenible e hidroponía

La hidroponía ahorra entre un 80% y un 95% de agua
frente al cultivo tradicional.

Descripción del problema

- La hidroponía es una solución sostenible de cultivo, pero requiere monitoreo constante.
- Los sistemas actuales son costosos o están orientados a grandes explotaciones.
- Necesidad de automatización de procesos mediante interfaces accesibles e intuitivas

02

Descripción de la solución

Objetivos del proyecto

Concepto de módulo inteligente, sistema objetivo y planteamiento de la solución.

Descripción de la solución

Arquitectura por capas

Microservicios virtualizados



Origen de datos

Módulos hidropónicos
API de envío



--- Kafka ---

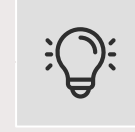
Persistencia

Base de datos
Backend
Recomendaciones



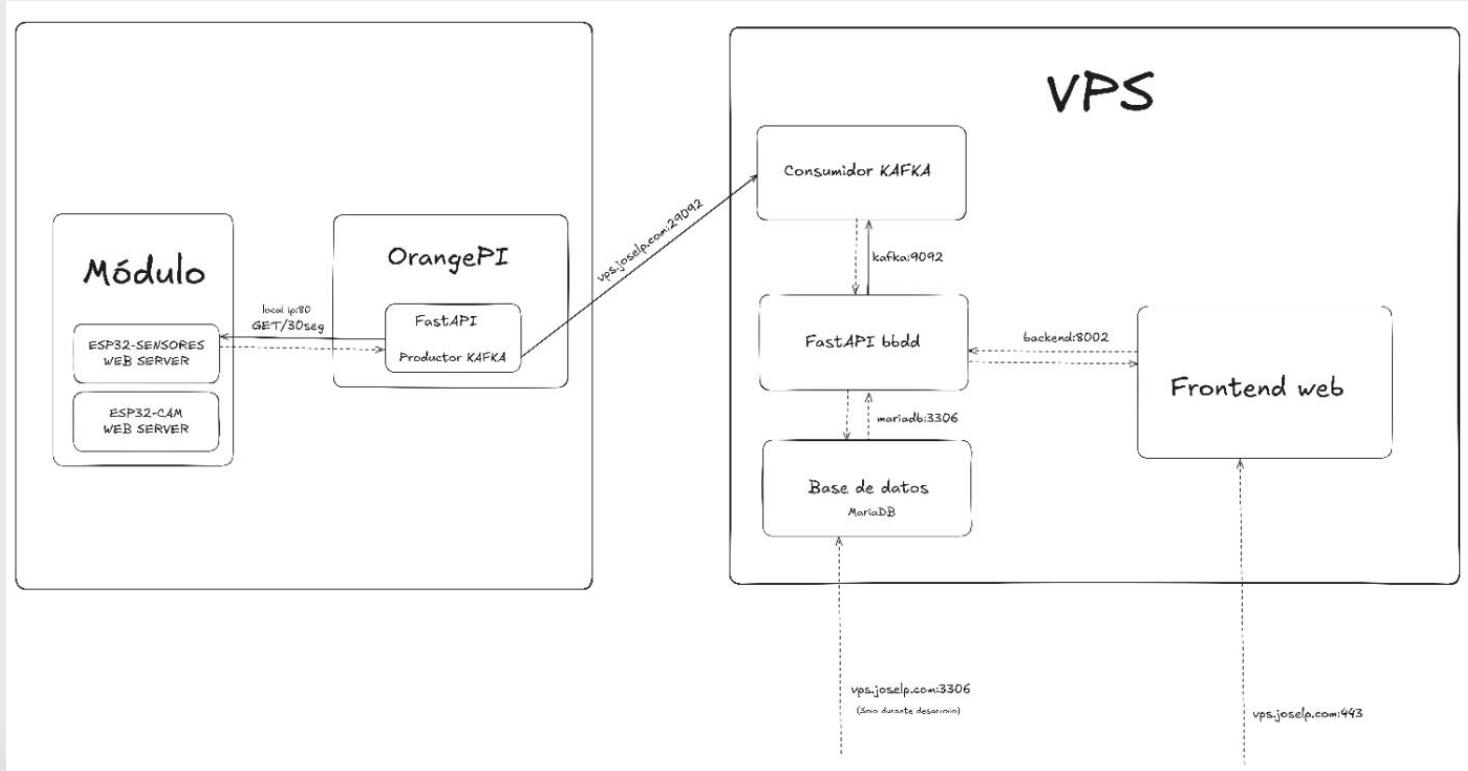
Explotación

Servicio web

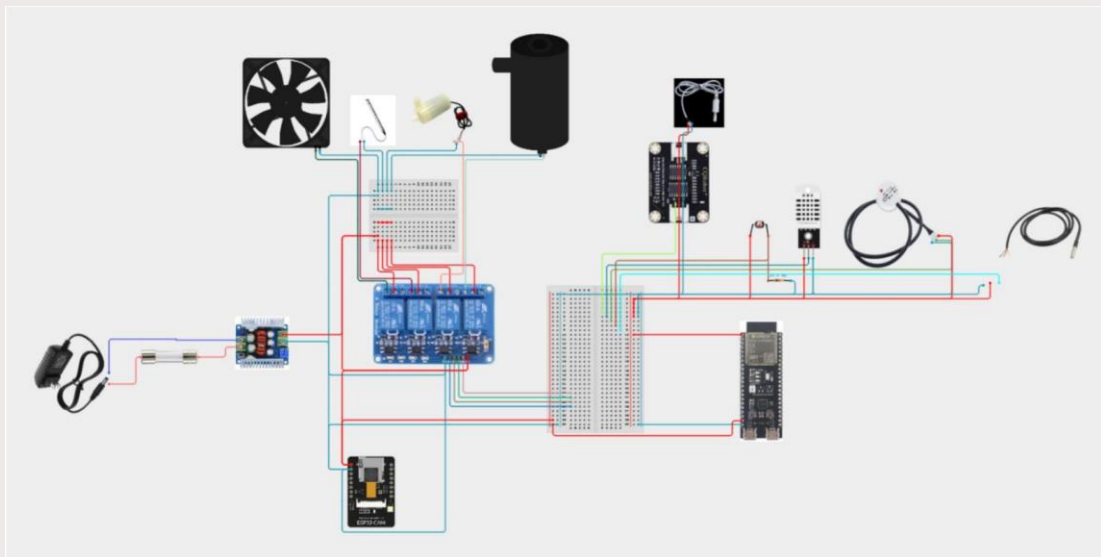
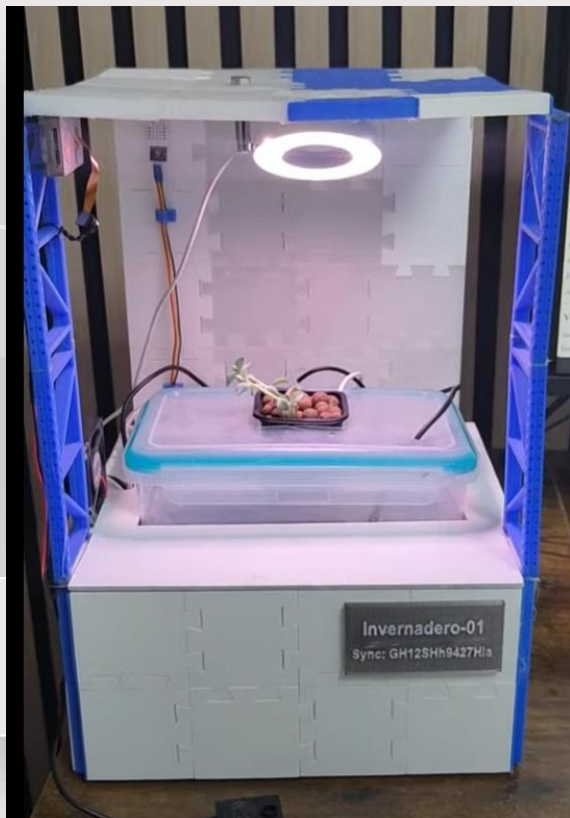


GreenhouseIOT

Descripción de la solución



Módulo hidropónico



Controles

IP del ESP32: **192.168.1.201**

Hora actual: **19:03**

Cambiar nombre del invernadero:

Luces

Modo de luces: **Auto**

Estado actual: **OFF**

Hora de encendido:

Hora de apagado:

Encender luces:

Apagar luces:

Oxígeno

Modo de oxígeno: **Auto**

Estado actual: **OFF**

Hora de encendido:

Hora de apagado:

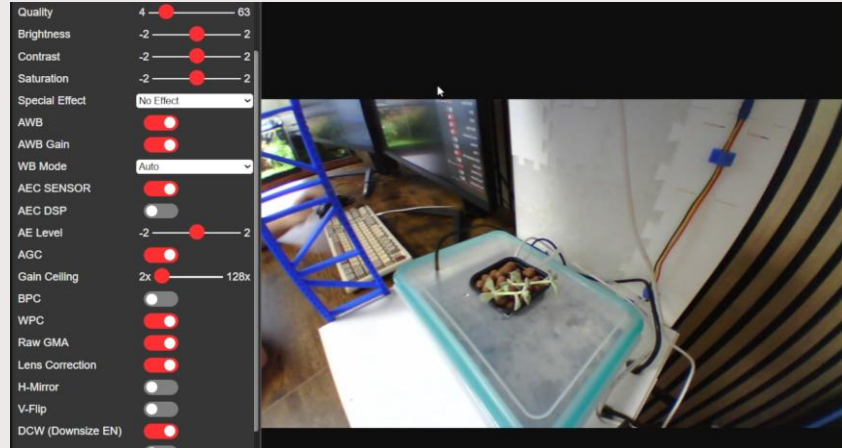
Encender oxígeno:

Apagar oxígeno:

Ventilador

Modo de ventilador: **Auto**

Módulo hidropónico



FastAPI 0.1.0 OAS 3.1

/openapi.json

Data Source

POST /send/sensordata Receive Data

default

GET /get-config/{gh_id} Get Remote Config

Schemas

HTTPValidationError > Expand all object

SensorData > Expand all object

ValidationError > Expand all object

Servicio de envío de datos

- Solicita al módulo valores de sus sensores
- Solicita imágenes
- Recupera y actualiza configuraciones
- Despliegue local y envío por Kafka

Backend y consumidor Kafka

- Los datos son recogidos por un consumidor y almacenados en la base de datos
- Gestiona accesos a base de datos
- Estrategias de protección
- Implementa funcionalidad clave para el servicio web



Base de datos

- Base de datos relacional MariaDB
- Servicio virtualizado
- Gestión a través de HeidiSQL



```
- name: Lanzar servicio db (MariaDB)
  docker_container:
    name: mariadb
    image: mariadb:latest
    state: started
    restart_policy: always
    networks:
      - name: mariadb_network
    env:
      MARIADB_ROOT_PASSWORD: "{{ mariadb_root_password }}"
    published_ports:
      - "3306:3306"
```

Servicio de recomendaciones

- Altos requerimientos hardware
- Ollama + LLaVA
- Toma los últimos datos e imágenes para obtener sugerencias y mejorar el cultivo
- Envuelto en FastAPI + Caddy



User



User

Do you know who drew this painting?



LLaVA

The painting depicts a woman, commonly believed to be Mona Lisa, the famous Italian Renaissance artist. The woman's enigmatic smile and the painting's intricate details have made it one of the most famous and mysterious works of art in the world. It is housed in the Louvre Museum in Paris, and it is known for its use of oil paint and its enigmatic smile.



Servicio web

- React + Vite + JavaScript
 - DaisyUI
 - Auth0
-
- Sincronización sencilla
 - Gráficas detalladas
 - Resumen de datos
 - Configuraciones remotas

03

Evaluación

Entre los criterios de evaluación se encuentran métricas como correctitud funcional, rendimiento, fiabilidad, usabilidad y escalabilidad.

Evaluación

Metodología de evaluación

- Pruebas locales basadas en los casos de uso
- Servicios evaluados de forma individual
- Rutina automática de test unitarios
- Encuesta a usuarios orientada en la resolución de una tarea

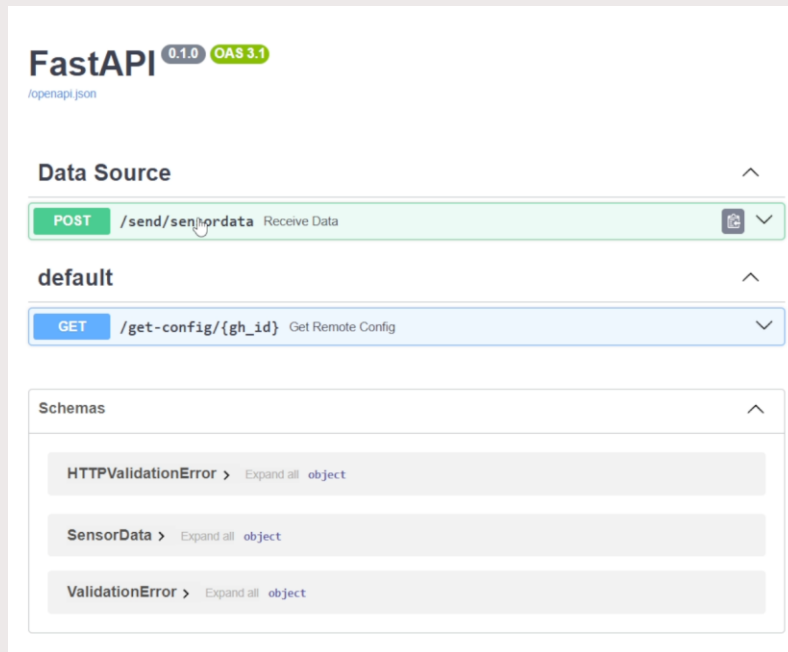
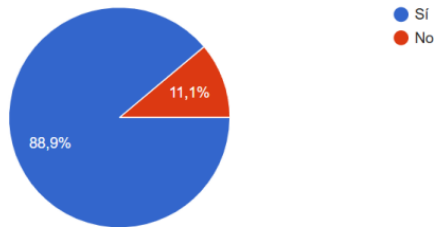
```
----- coverage: platform linux, python 3.9.23-final
Name                               Stmts  Miss  Cover
-----
app/const.py                        4      0   100%
app/controllers/db/connector.py     19      0   100%
app/controllers/db/db_queries.py   156      0   100%
app/controllers/detector.py         15      0   100%
app/kafka_module/consumer.py        38     13    66%
app/main.py                         33      2    94%
app/routers/db_router.py            79      0   100%
app/routers/greenhouses.py          29      0   100%
app/routers/reads.py               144     28    81%
app/routers/users.py                33      0   100%
-----
TOTAL                               550     43    92%
```


Evaluación

Herramientas de evaluación

¿Consiguió sincronizar el invernadero y configurarlo ?

18 respuestas

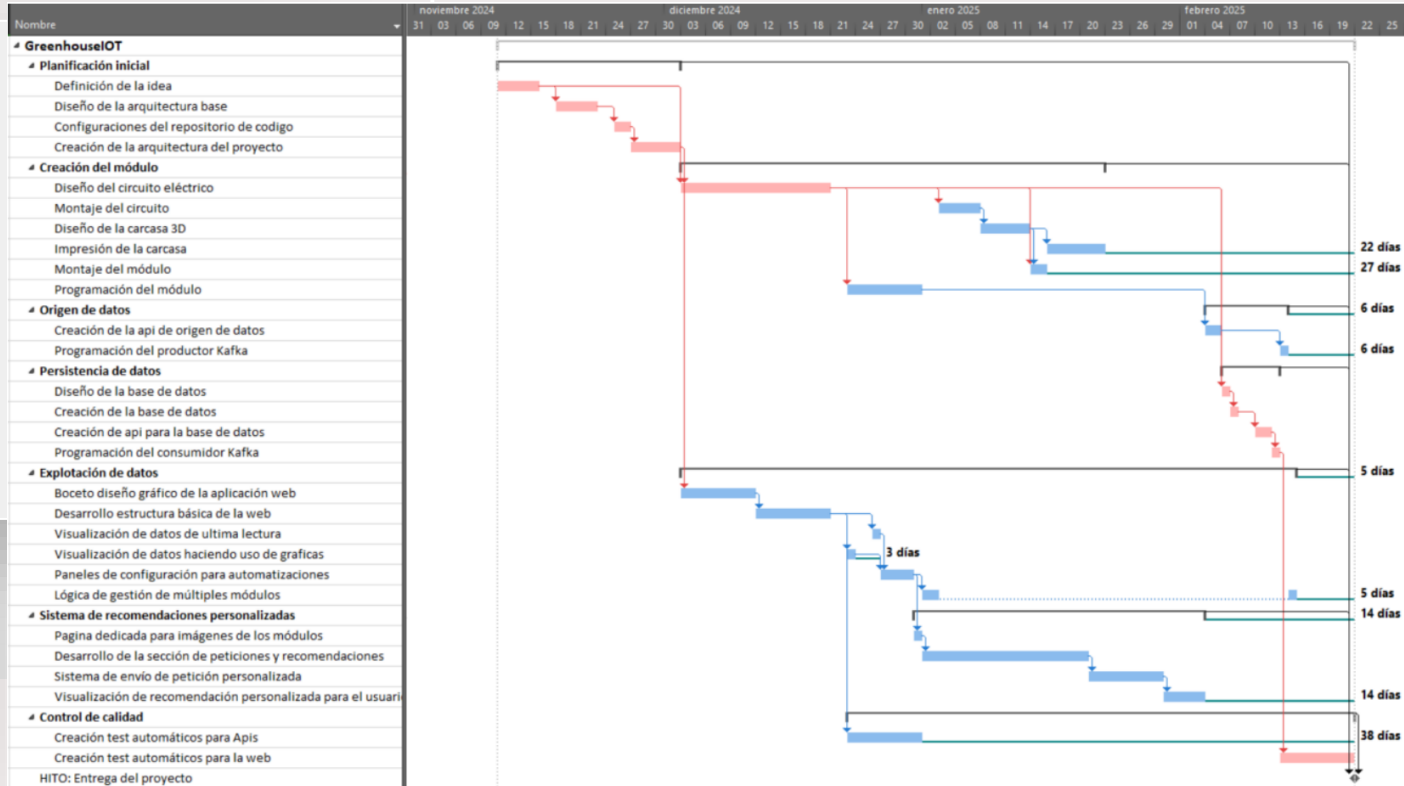


04

Gestión del proyecto

Tareas y presupuestos

Diagrama de Gantt



Presupuestos

Concepto	Coste aproximado
Recursos humanos	28.000 €
Hardware	2.000 €
Otros costes	19.000 €

Total: 49.000 €



06

Conclusiones

El proyecto GreenhouseIoT ha alcanzado los objetivos planteados, desarrollando una solución integrada para la gestión y automatización de módulos hidropónicos inteligentes a través de una aplicación web.

Futuras líneas de trabajo

- Seguimiento de las necesidades de los usuarios mediante un portal de sugerencias.
- Creación de un servicio que simule un módulo virtual para mejorar la sección de guías de usuario.
- Panel y rango administrador.
- Incremento de sensores.



Aplicación web para la gestión de módulos hidropónicos inteligentes

Autor: José Antonio López Pérez

Tutores: Lidia Sánchez González,
Jesús Fernández Fernández