

Flujo completo del sistema

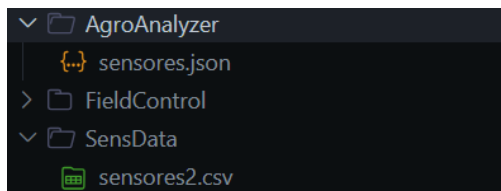
1. SensData genera lecturas de sensores en formato CSV.
2. AgroAnalyzer recibe esos datos, los convierte a JSON y los guarda en una base de datos.
3. FieldControl consulta los datos más recientes para tomar decisiones de riego.

1. SensData → AgroAnalyzer (File Transfer)

El módulo SensData exporta un archivo sensores.csv con lecturas como:

```
id_sensor,fecha,humedad,temperatura
S001,2025-05-22,45,26.4
S002,2025-05-22,50,25.1
S003,2025-05-22,47,27.3
```










Una ruta Camel detecta automáticamente este archivo, lo convierte a JSON y lo guarda en la carpeta de AgroAnalyzer como sensores.json.



2. AgroAnalyzer → MySQL (Shared Database)

AgroAnalyzer toma el JSON y lo transforma en objetos Java (SensorLectura). Luego, cada lectura se inserta en la base de datos MySQL en la tabla lecturas.

La tabla contiene columnas como: id, id_sensor, fecha, humedad, temperatura.

						id	id_sensor	fecha	humedad	temperatura
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	S001	2025-05-22 00:00:00	45	26.4		
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	S002	2025-05-22 00:00:00	50	25.1		
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	S003	2025-05-22 00:00:00	47	27.3		

3. FieldControl → MySQL (Consulta directa)

FieldControl simula una consulta por sensor. Por ejemplo, al colocar S001.txt en la carpeta FieldControl/requests/, se activa una ruta Camel que:

- Lee el ID del sensor
- Ejecuta una consulta SQL para obtener la lectura más reciente
- Genera un archivo JSON con la respuesta en FieldControl/responses/



1. ¿Qué patrón aplicaste en cada fase del flujo y por qué?

Se usó Shared Database permitió que AgroAnalyzer guarde lecturas en MySQL y FieldControl las consulte.

2. ¿Qué riesgos observas al usar una base de datos compartida?

Una base compartida puede generar dependencia entre sistemas. Si falla o cambia, todos se ven afectados. También hay riesgo de conflictos si varios procesos acceden al mismo tiempo, y se complica el control de seguridad y acceso.

3. ¿Cómo ayuda el RPC simulado a representar un flujo síncrono?

El RPC simulado permite que un módulo envíe una solicitud y reciba respuesta inmediata, como si fuera una llamada directa. Aunque no usa red real, Camel lo simula bien, ideal para decisiones en tiempo real.

4. ¿Qué limitaciones tienen los patrones clásicos frente a arquitecturas modernas?

Los patrones clásicos son útiles, pero menos escalables. No manejan errores distribuidos

ni ofrecen trazabilidad avanzada. Las arquitecturas modernas usan eventos y servicios desacoplados que son más flexibles y resistentes.

Github:

<https://github.com/WillyC4/AgroAnalyzer>