

**Integrantes:** Alex Caicedo Ramos

**Fecha:** 30 de octubre, 2025

## Prototipo de Integración:

**Patrones:** File Transfer, Shared Database, RPC (simulado)

**Stack:** Java 17 · Maven · Apache Camel 4.x · SQLite · Logback

## Resumen:

Se implementó un prototipo funcional de integración entre SensData, AgroAnalyzer y FieldControl. El sistema:

- Detecta automáticamente un CSV con lecturas de sensores (File Transfer),
- Copia el archivo a una carpeta de salida y convierte el contenido a JSON,
- Persiste todas las lecturas en una base de datos compartida (SQLite),
- Expone una consulta síncrona (RPC simulado) para recuperar el último valor por sensor,
- Registra la operación en un log centralizado con rotación.

El resultado cumple el caso de uso: disponibilidad del último valor para decisiones de riego y trazabilidad del flujo end-to-end.

## Arquitectura y patrones explicados:

### File Transfer (SensData → AgroAnalyzer)

- Un consumidor de archivos Camel detecta ./input/sensores.csv, copia el archivo a ./output/ con timestamp y convierte el contenido del CSV a JSON (arreglo de lecturas).
- Justificación: patrón sencillo, robusto y alineado con procesos por lotes de exportación horaria.

### Shared Database (AgroAnalyzer ↔ FieldControl)

- AgroAnalyzer inserta cada lectura en ./database/agrotech.db (SQLite), tabla lecturas(id\_sensor, fecha, humedad, temperatura).
- FieldControl consulta directamente la misma BD para obtener el último valor por sensor.
- Justificación: repositorio común, baja latencia de lectura y simplicidad de integración.

### RPC simulado (FieldControl → AgroAnalyzer)

- Llamada síncrona modelada con rutas Camel direct:: el cliente envía id\_sensor, el servidor consulta SQLite y responde con la lectura más reciente.
- Justificación: emula decisión en línea (activar/desactivar riego) con respuesta inmediata.

Integrantes: Alex Caicedo Ramos

Fecha: 30 de octubre, 2025

## Logging centralizado

- Log a archivo `./logs/agrotech.log` con rotación diaria y por tamaño (Logback).

## Reflexión Individual:

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

- **File Transfer:** Detección de CSV, copia y transformación a JSON. Porque es simple y a la vez robusto, emula los procesos por lotes de SensData.
- **Shared Database:** Repositorio común para lecturas, FieldControl lee el último valor con baja latencia.
- **RPC simulado:** solicitud-respuesta inmediata para decisiones de riego en tiempo real.

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

Cuando muchos sistemas usan la misma base de datos un cambio puede afectar a todos si muchas personas la usan al mismo tiempo puede haber demoras o bloqueos también hay que cuidar la seguridad ver quién accede y proteger los datos si el sistema no está bien hecho puede fallar todo o ser difícil hacerlo crecer además hay que mantener los datos correctos y evitar errores cuando se guardan

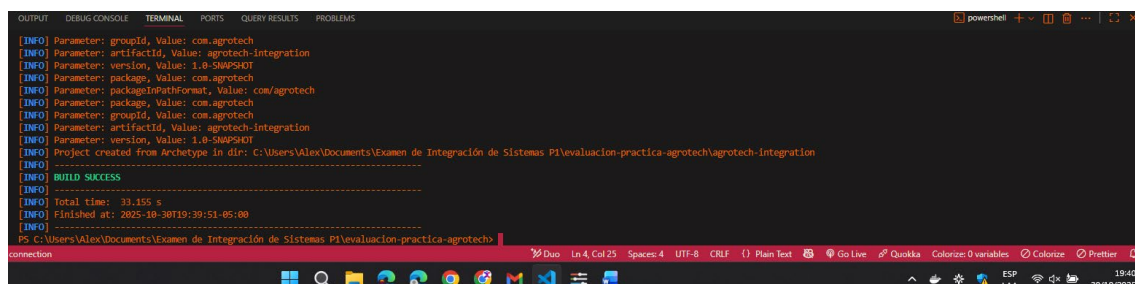
### ¿Cómo ayuda el RPC simulado a representar un flujo síncrono?

El cliente espera la respuesta antes de seguir se envían datos como el identificador del sensor y siempre se obtiene el mismo resultado además muestra la decisión que toma FieldControl en el momento

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

El envío de archivos puede ser más lento que el uso de eventos en tiempo real cuando todo depende de la misma base de datos hay más conexión entre sistemas pero si cada uno tiene su propio esquema es más fácil actualizarlos además las colas o flujos de mensajes como Kafka o RabbitMQ permiten crecer y resistir mejor los fallos por último es más difícil ver y mejorar cómo funcionan las cosas cuando no hay buenas trazas o contratos claros entre sistemas

## Evidencias:



```

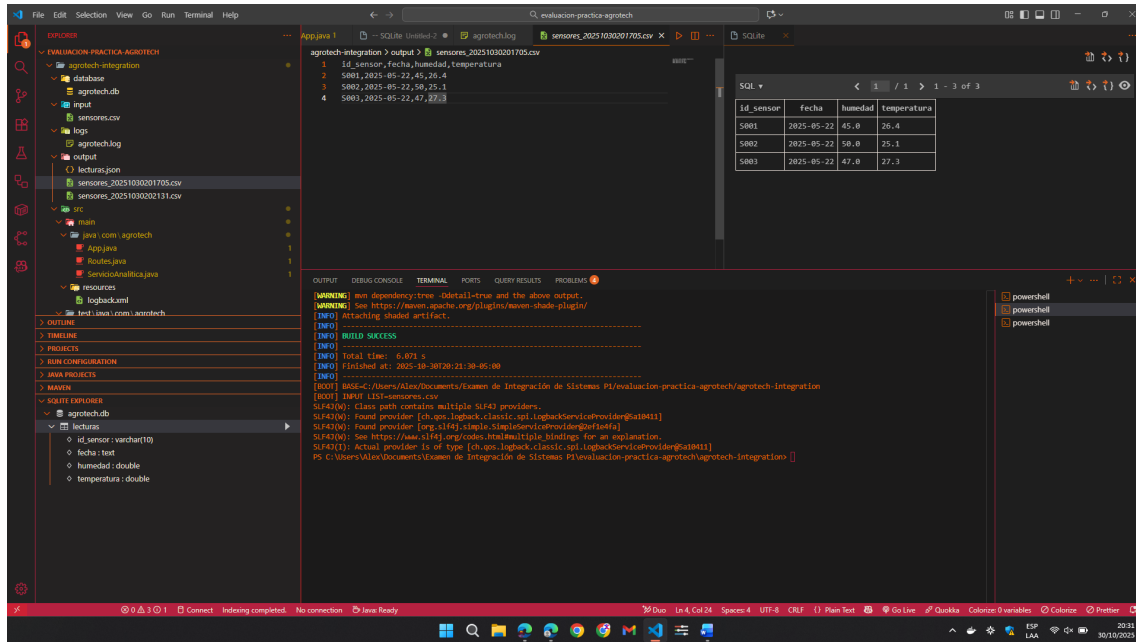
[INFO] Parameter: groupId, Value: com.agrotech
[INFO] Parameter: artifactId, Value: agrotech-integration
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.agrotech
[INFO] Parameter: packageInPathFormat, Value: com/agrotech
[INFO] Parameter: package, Value: com.agrotech
[INFO] Parameter: groupId, Value: com.agrotech
[INFO] Parameter: artifactId, Value: agrotech-integration
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\Alex\Documents\Examen de Integración de Sistemas PI\evaluacion-practica-agrotech\agrotech-integration
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 33.155 s
[INFO] Finished at: 2025-10-30T19:39:51-05:00
[INFO] -----
PS C:\Users\Alex\Documents\Examen de Integración de Sistemas PI\evaluacion-practica-agrotech>
```

# Integración de Sistemas

## Examen de Progreso 1

Integrantes: Alex Caicedo Ramos

Fecha: 30 de octubre, 2025



Output:

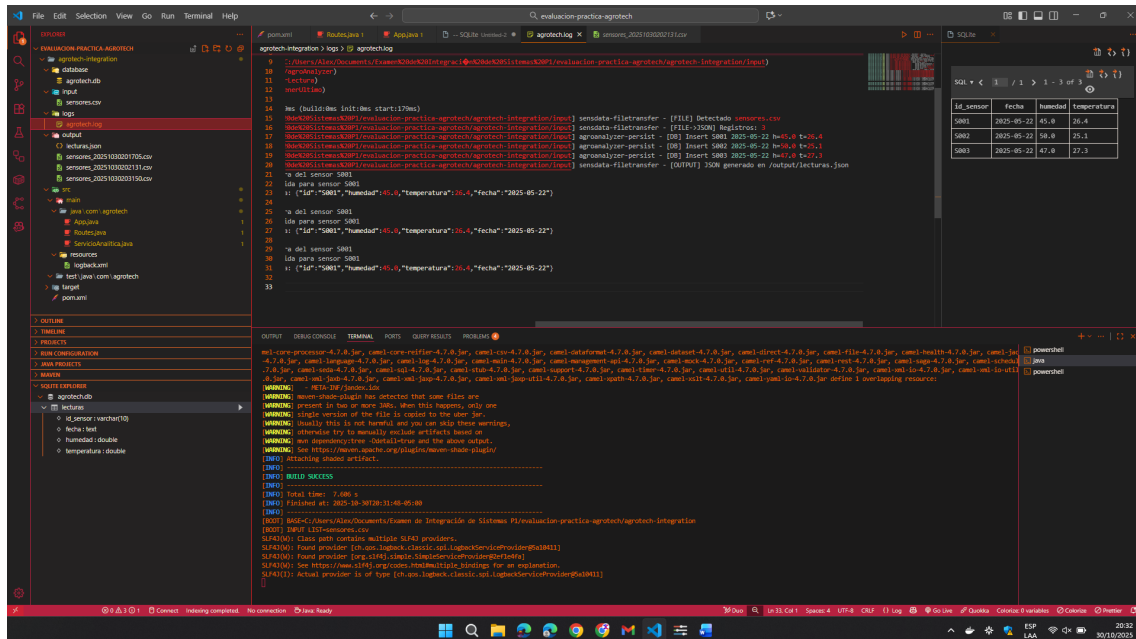
```
1 id_sensor: fecha, humedad, temperatura
2 5801, 2025-05-22, 45.8, 26.4
3 5802, 2025-05-22, 58.9, 25.1
4 5803, 2025-05-22, 47.8, 27.3
```

SQL:

id_sensor	fecha	humedad	temperatura
5801	2025-05-22	45.8	26.4
5802	2025-05-22	58.9	25.1
5803	2025-05-22	47.8	27.3

Output:

```
WARNING: non dependency: true - Detail: true and the above output.
WARNING: See https://www.apache.org/plugins/maven-shade-plugin/
[INFO] Attaching shaded artifact.
[INFO] BUILD SUCCESS
[INFO] Total time: 6.071 s
[INFO] Finished at: 2025-10-30T20:21:38-05:00
[INFO]
[ERROR] BASE-C:\Users\Alex\Documents\Examen de Integración de Sistemas PI\evaluacion-practica-agrotech\agrotech-integration
[ERROR] INPUT LIST-sensors.csv
SLF4J(W): Class path contains multiple SLF4J providers.
SLF4J(W): Found provider [ch.qos.logback.classic.spi.LogbackServiceProvider@8a1811]
SLF4J(W): Found provider [org.slf4j.simple.SimpleServiceProvider@610afa]
SLF4J(W): See https://www.slf4j.org/faq.html#multiple_bindings for an explanation.
SLF4J(I): Actual provider is of type [ch.qos.logback.classic.spi.LogbackServiceProvider@8a1811]
PS C:\Users\Alex\Documents\Examen de Integración de Sistemas PI\evaluacion-practica-agrotech\agrotech-integration>
```



Output:

```
1 id_sensor: fecha, humedad, temperatura
2 5801, 2025-05-22, 45.8, 26.4
3 5802, 2025-05-22, 58.9, 25.1
4 5803, 2025-05-22, 47.8, 27.3
```

SQL:

id_sensor	fecha	humedad	temperatura
5801	2025-05-22	45.8	26.4
5802	2025-05-22	58.9	25.1
5803	2025-05-22	47.8	27.3

Output:

```
WARNING: non dependency: true - Detail: true and the above output.
WARNING: See https://www.apache.org/plugins/maven-shade-plugin/
[INFO] Attaching shaded artifact.
[INFO] BUILD SUCCESS
[INFO] Total time: 7.099 s
[INFO] Finished at: 2025-10-30T20:31:40-05:00
[INFO]
[ERROR] BASE-C:\Users\Alex\Documents\Examen de Integración de Sistemas PI\evaluacion-practica-agrotech\agrotech-integration
[ERROR] INPUT LIST-sensors.csv
SLF4J(W): Class path contains multiple SLF4J providers.
SLF4J(W): Found provider [ch.qos.logback.classic.spi.LogbackServiceProvider@8a1811]
SLF4J(W): Found provider [org.slf4j.simple.SimpleServiceProvider@610afa]
SLF4J(W): See https://www.slf4j.org/faq.html#multiple_bindings for an explanation.
SLF4J(I): Actual provider is of type [ch.qos.logback.classic.spi.LogbackServiceProvider@8a1811]
PS C:\Users\Alex\Documents\Examen de Integración de Sistemas PI\evaluacion-practica-agrotech\agrotech-integration>
```