

Universidad de Las Américas

Facultad de Ingenierías y Ciencias Agropecuarias

*Ingeniería de Software*

Informe de laboratorio

1. DATOS DEL ALUMNO:



José Ernesto Pérez Tapia


2. TEMA DE LA PRÁCTICA:

Progreso 1

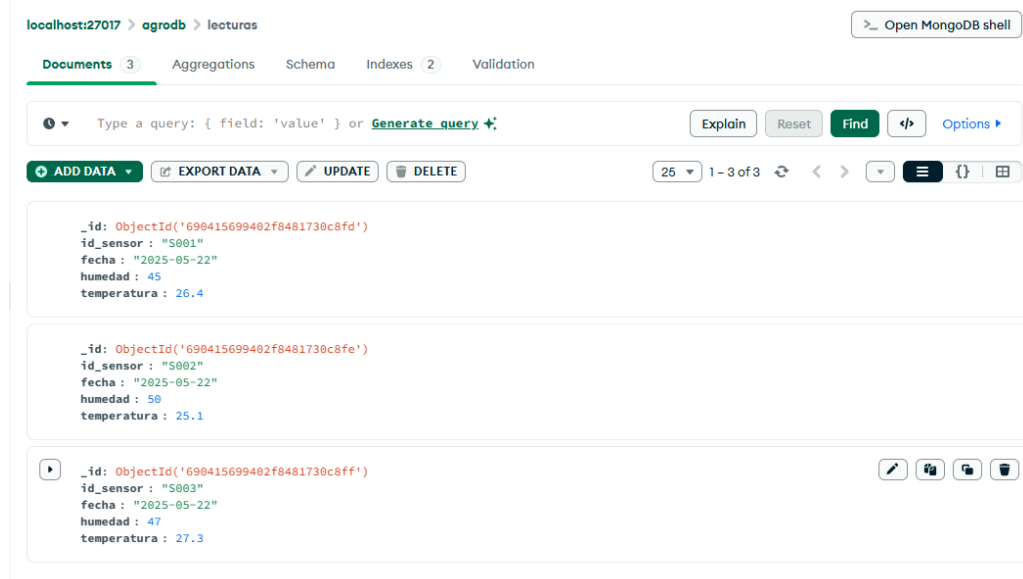
3. DESARROLLO DE LA PRÁCTICA Y RESULTADOS

Primero realizamos un File transfer protocol en el cual toma el archivo de una carpeta, la cambia el formato de csv a JSON y lo envía a la carpeta de salida como lo podemos observar aqui:

```
ensData >  sensores.csv >  data  
1 id_sensor,fecha,humedad,temperatura  
2 S001,2025-05-22,45,26.4  
3 S002,2025-05-22,50,25.1  
4 S003,2025-05-22,47,27.3
```

```
AgroAnalyzer >  sensores.json > ...  
1 [  
2   {  
3     "fecha": "2025-05-22",  
4     "id_sensor": "S001",  
5     "humedad": "45",  
6     "temperatura": "26.4"  
7   },  
8   {  
9     "fecha": "2025-05-22",  
10    "id_sensor": "S002",  
11    "humedad": "50",  
12    "temperatura": "25.1"  
13  },  
14  {  
15    "fecha": "2025-05-22",  
16    "id_sensor": "S003",  
17    "humedad": "47",  
18    "temperatura": "27.3"  
19  }  
20 ]
```

AgroAnalyzer debe guardar los datos procesados en una base de datos



AgroAnalyzer guarda las lecturas procesadas en una base de datos compartida: en nuestro caso la colección MongoDB agrodb.lecturas (equivalente a la tabla SQL lecturas). FieldControl lee directamente esa base y devuelve la lectura más reciente por sensor.

Para la tercer instrucción simulamos una llamada para obtener los datos del sensor 3 y nos queda de esta manera

```
[RPC] Servicio Analítica - Buscando datos del sensor: S003
[RPC] Leyendo archivo: AgroAnalyzer/sensores.json
[RPC] Total de sensores en archivo: 3
[RPC] Sensor encontrado -> {"id":"S003","humedad":47,"temperatura":27.3,"fecha":"2025-05-22"}
```

```
>> Realizando llamada RPC para sensor: S003 <<
[com.agrotech.routes.FieldControl.main()] INFO route1 - [RPC] Cliente - Enviando solicitud
para sensor: S003

[RPC] Servicio Analítica - Buscando datos del sensor: S003
[RPC] Leyendo archivo: AgroAnalyzer/sensores.json
[RPC] Total de sensores en archivo: 3
[RPC] Sensor encontrado -> {"id":"S003","humedad":47,"temperatura":27.3,"fecha":"2025-05-22"}
[com.agrotech.routes.FieldControl.main()] INFO route1 - [RPC] Cliente - Respuesta recibida
: {"id":"S003","humedad":47,"temperatura":27.3,"fecha":"2025-05-22"}
```

[https://github.com/JosecoLoco/Progreso\\_1\\_Jose\\_Perez](https://github.com/JosecoLoco/Progreso_1_Jose_Perez)

1. ¿Qué patrón aplicaste en cada fase del flujo y por qué?  
File Transfer Pattern para el CSV a JSON Remote Procedure Call (RPC) para consultas sincrónicas Shared Database para almacenamiento compartido Elegidos por su simplicidad y capacidad de integración directa.
2. ¿Qué riesgos observas al usar una base de datos compartida?  
Acoplamiento fuerte entre aplicaciones Problemas de concurrencia Dificultad en cambios de esquema Riesgo de inconsistencia de datos
3. ¿Cómo ayuda el RPC simulado a representar un flujo síncrono?  
Garantiza respuesta inmediata Simula llamadas locales Facilita depuración Mantiene consistencia en las operaciones
4. ¿Qué limitaciones tienen los patrones clásicos frente a arquitecturas modernas?  
Menor escalabilidad Acoplamiento más fuerte Menos tolerancia a fallos Dificultad para cambios dinámicos Menos flexibilidad que microservicio