



Ingeniería de Software

Integración de Sistemas

Liseth Abad

Fecha de entrega:
29 de octubre de 2025

Examen Progreso 1

<https://github.com/lisa-vsqz/FileTransferAPI.git>

1. Reflexión individual

1. ¿Qué patrón de integración aplicaste y cómo se refleja en tu solución?

Se utilizó principalmente el patrón de File Transfer y API REST. En la solución, se reciben los datos de envíos desde un archivo CSV, que luego se transforma a JSON y se guardan en la carpeta output. Apache Camel me sirvió como el intermediario ya que utiliza el archivo envios.csv de una carpeta local, transforma los datos y después los deja disponibles tanto en envios.json como en memoria para su posterior uso a través de una API REST. De esta manera, el patrón garantiza que los datos puedan ser consumidos por otros sistemas sin depender de la ubicación física del archivo original.

2. ¿Qué ventajas identificas al pasar de File Transfer a APIs REST?

Primero, los datos se vuelven accesibles en tiempo real, lo que evita la necesidad de esperar a que se generen y transfieran archivos completos desde envíos.csv a envíos.json. Además, la API permite consultas específicas, como obtener un envío por su ID o agregar nuevos registros, lo que incrementa la flexibilidad. Finalmente, facilita la documentación con OpenAPI, lo que mejora la comprensión y el consumo de los servicios por parte de sistemas externos.

3. ¿Qué riesgos o limitaciones encontraste en tu enfoque?

Durante el desarrollo resalto el hecho de la dependencia de mantener la lista de envíos en memoria. Es decir que, si el sistema se reinicia, los datos persisten a menos que se almacenen en disco o en una base de datos. Además, REST facilita el acceso a los datos, pero el manejo de múltiples usuarios simultáneos podría generar conflictos o inconsistencias si no hay mecanismos de control adecuados. Finalmente, tuve la limitación de que la lectura de archivos CSV depende de que el formato sea correcto; ya que cualquier error en la estructura podría interrumpir la transformación a JSON y el programa tendría errores en la ejecución.

4. ¿Cómo escalarías esta integración si EcoLogistics tuviera 50 sistemas distintos?

Si deseara escalar el EcoLogistics implementaría un servicio centralizado de integración basado en APIs. Aunque este enfoque requiere mayor control y mejor gobernanza, permitiría que cada uno de los 50 sistemas exponga sus propios endpoints REST de forma estandarizada. Además, consideraría usar un API Gateway para centralizar la seguridad, el enrutamiento y el monitoreo, junto con un broker de mensajería como Kafka para manejar el volumen de datos y garantizar la entrega confiable entre sistemas.