

**Nombre:** Said Carrera

**Link de GitHub:** <https://github.com/SaidCarrera/evaluacion-practica-ecologistics.git>

## Estructura del Proyecto

```
saidcarrera@Said-MacBook-Pro ~ % tree ~/evaluacion-practica-ecologistics -L 2
/Users/saidcarrera/evaluacion-practica-ecologistics
├── README.md
├── camel-ecologistics
│   ├── data
│   ├── pom.xml
│   ├── src
│   └── target
├── envios.csv
├── openapi.yaml
├── postman_collection.json
├── src
│   └── main.py
└──
```

6 directories, 6 files  
saidcarrera@Said-MacBook-Pro ~ %

## Ejecución de Apache Camel

```
saidcarrera — java --enable-native-access=ALL-UNNAMED -classpath /opt/homebrew/Cellar/maven/3.9.11/libexec/boot/plexus-classworlds...

saidcarrera@Said-MacBook-Pro ~ % cd ~/camel-ecologistics
(mvn compile exec:java
WARNING: A terminally deprecated method in sun.misc.Unsafe has been called
WARNING: sun.misc.Unsafe::staticFieldBase has been called by com.google.inject.internal.aop.HiddenClassDefiner (file:/opt/homebrew/Cellar/mav
en/3.9.11/libexec/lib/guice-5.1.0-classes.jar)
WARNING: Please consider reporting this to the maintainers of class com.google.inject.internal.aop.HiddenClassDefiner
WARNING: sun.misc.Unsafe::staticFieldBase will be removed in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.ecologistics:camel-ecologistics >-----
[INFO] Building camel-ecologistics 1.0.0
[INFO] from pom.xml
[INFO]
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ camel-ecologistics ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /Users/saidcarrera/camel-ecologistics/src/main/resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ camel-ecologistics ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- exec:3.5.0:java (default-cli) @ camel-ecologistics ---
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
[INFO] Aplicación Camel iniciada. Esperando archivos CSV en data/in/
```

## Verificar archivo transformado

```
saidcarrera — zsh — 109x15

Last login: Wed Oct 29 20:42:55 on ttys008
[saidcarrera@SaiDs-MacBook-Pro ~ % cat ~/camel-ecologistics/data/out/envios.json
[{"id_envio":"001","cliente":"Juan Pérez","direccion":"Calle 12 #45","estado":"Entregado"}, {"id_envio":"002",
"cliente":"Maria Gómez","direccion":"Avenida 10 #33","estado":"En tránsito"}, {"id_envio":"003","cliente":"Lui
s Mora","direccion":"Carrera 8 #22saisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisai
saidcarrera@SaiDs-MacBook-Pro ~ % █
```

## Levantar el servidor FastAPI

```
saidcarrera — Python < uvicorn src.main:app --port 8090 --reload — 106x13

source .venv/bin/activate
[uvicorn src.main:app --port 8090 --reload
INFO: Will watch for changes in these directories: ['/Users/saidcarrera/evaluacion-practica-ecologisti
cs']
INFO: Uvicorn running on http://127.0.0.1:8090 (Press CTRL+C to quit)
INFO: Started reloader process [45115] using StatReload
INFO: Started server process [45117]
INFO: Waiting for application startup.
[INFO] Archivo cargado con 3 registros.
[INFO] Datos transformados a formato JSON.
[INFO] API iniciada en puerto 8080 (cuando uvicorn arranque).
INFO: Application startup complete.
█
```

## Probar la API con cURL

### Obtener todos los envíos

```
saidcarrera — zsh — 85x15

Last login: Wed Oct 29 20:47:24 on ttys006
[saidcarrera@SaiDs-MacBook-Pro ~ % curl http://127.0.0.1:8090/envios
[{"id":"001","cliente":"Juan Pérez","direccion":"Calle 12 #45","estado":"Entregado"},
{"id":"002","cliente":"María Gómez","direccion":"Avenida 10 #33","estado":"En tránsito"}, {"id":"003","cliente":"Lsaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisaisai
saidcarrera@SaiDs-MacBook-Pro ~ % █
```

### Obtener un envío específico

```
saidcarrera — -zsh — 120x30

Last login: Wed Oct 29 20:49:00 on ttys007
saidcarrera@SaiDs-MacBook-Pro ~ % curl http://127.0.0.1:8090/envios/001
{"id":"001","cliente":"Juan Pérez","direccion":"Calle 12 #45","estado":"Entregado"}%
saidcarrera@SaiDs-MacBook-Pro ~ %
```

## Crear un nuevo envío

```
saidcarrera — -zsh — 120x30

Last login: Wed Oct 29 20:49:13 on ttys007
saidcarrera@SaiDs-MacBook-Pro ~ % curl -X POST http://127.0.0.1:8090/envios \
-H "Content-Type: application/json" \
-d '{"id": "004", "cliente": "Ana Ruiz", "direccion": "Pasaje 5 #10", "estado": "Pendiente"}'
{"id":"004","cliente":"Ana Ruiz","direccion":"Pasaje 5 #10","estado":"Pendiente"}%
saidcarrera@SaiDs-MacBook-Pro ~ %
```

## REFLEXION:

Este examen práctico me ayudó a comprender mejor cómo integrar distintas tecnologías dentro de una arquitectura de software moderna.

El uso de Apache Camel me permitió experimentar con flujos de integración y transformación de datos, automatizando la lectura y conversión de archivos CSV a JSON.

Además, con FastAPI reforcé la creación de APIs REST eficientes y fáciles de mantener, logrando conectar los procesos de integración con una capa de servicio funcional.

Gracias a este trabajo entendí la importancia de diseñar soluciones modulares y escalables, donde cada componente cumple un rol claro dentro del sistema.

## PREGUNTAS:

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

Apliqué los patrones File Transfer y API REST.

El primero se refleja en el uso de Apache Camel, que lee el archivo envios.csv, transforma su contenido a formato JSON y genera un archivo de salida listo para consumo.

El segundo patrón se refleja en la API creada con FastAPI, que expone los datos mediante endpoints REST para consulta y registro de envíos.

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

Las APIs REST permiten una comunicación más rápida, segura y en tiempo real, eliminando la dependencia de archivos intermedios.

Además, facilitan la interoperabilidad entre sistemas y permiten escalar o modificar servicios sin afectar al resto del ecosistema.

Mientras que el File Transfer es útil para cargas por lotes, las APIs ofrecen mayor flexibilidad y automatización.

**¿Qué riesgos o limitaciones encontraste en tu enfoque?**

Una limitación es que la solución actual mantiene los datos en memoria, lo cual no es óptimo para grandes volúmenes.

También, si Camel o la API se detienen, el flujo podría interrumpirse.

Además, la configuración inicial de Camel puede ser compleja si se requieren transformaciones más avanzadas o múltiples rutas simultáneas.

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

Para escalar el sistema, implementaría una arquitectura basada en colas y mensajería (por ejemplo, RabbitMQ o Kafka) que permita integrar varios sistemas de forma asíncrona.

Cada sistema podría publicar o consumir datos mediante APIs estandarizadas y Camel gestionaría las transformaciones entre formatos.

De esta forma, se lograría una arquitectura modular, resiliente y fácilmente extensible.