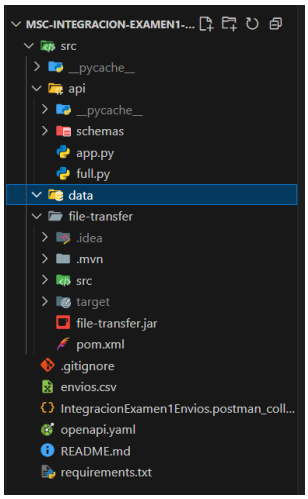
FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

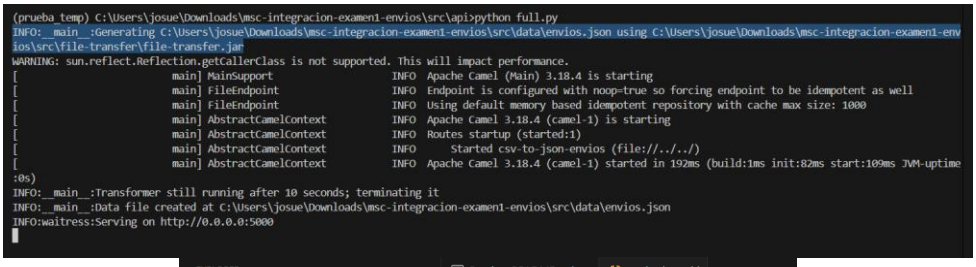
Examen Progreso I

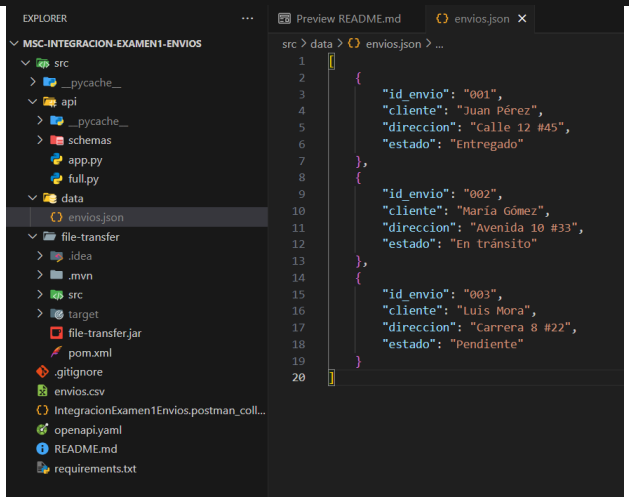
Autor (Quito, Josué)
Estudiantes del curso [ISWZ 3104] – [INTEGRACION DE SISTEMAS], Universidad de Las Américas, Quito-Ecuador

I. EVIDENCIAS



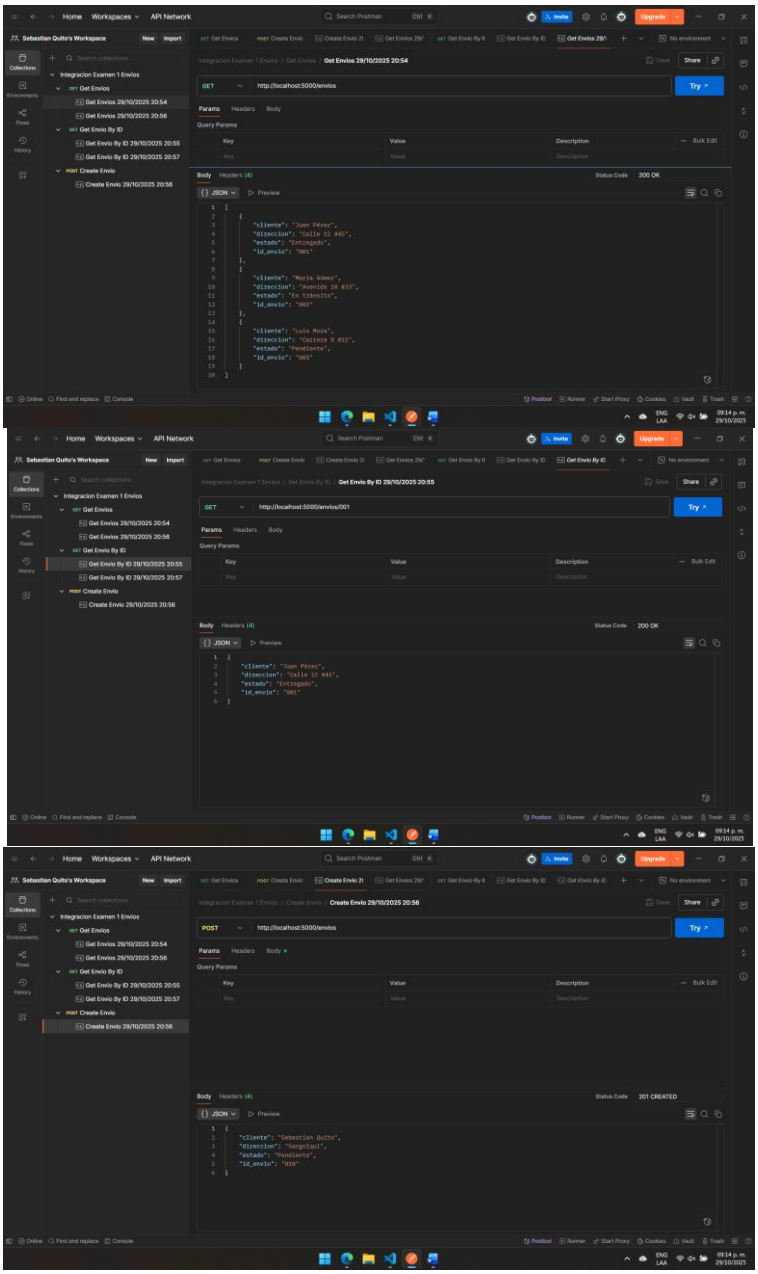
Carpeta data vacia ya que no se ha ejecutado Camel





Ejecución del proyecto, como se puede ver en los logs se indica que se creara el archivo por medio de file transfer, en este caso, Python ejecuta la versión compilada del proyecto de apache camel para crear el archivo .json.

FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS



Ejecución correcta de cada uno de los endpoints

II. PREGUNTAS DE REFLEXIÓN

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

   Se utilizaron los siguientes patrones de integración:
   - File Transfer con Apache Camel al leer un archivo CSV desde el filesystem y escribirlo en formato JSON en otra ruta dentro del disco.
   Data Format: Con la conversión CSV → POJO → JSON.

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

   Ventajas al pasar de File Transfer a APIs REST:

   - Menor latencia y consumo on-demand en tiempo real; no hay que esperar a lotes.

   - Validación y errores inmediatos (409, 404, 422) con contratos claros vía OpenAPI.

   - Observabilidad y trazabilidad mejores (logs por petición, status codes).

   - Seguridad y control: autenticación/autorización, rate limiting, versionado de API.

   - Evolución más ágil del contrato sin coordinar ventanas de drop de archivos

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

   Para este desarrollo rápido encontré los siguientes riesgos y limitaciones:

   - Acoplamiento al filesystem y rutas relativas: la ruta usa .. y file:../data; sensibles al directorio de trabajo.

   - La API lee y reescribe el archivo JSON completo en cada operación sin locking ni transacciones; acceso concurrente puede corromper o perder cambios.

   - Al usar un archivo JSON como persistencia ya que este no escala, no indexa, y no ofrece durabilidad/backup/consultas eficientes.

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

   Migrar para usar un bus de mensajes para que todos se comuniquen por eventos (publicar/suscribirse). Colocar un API Gateway para centralizar seguridad y límites. Definir un modelo de datos común para el esquema de envio. Si llegan archivos, guárdalos en almacenamiento de objetos y convertirlos en eventos. Finalmente, guardar la información en una base de datos no en archivos y despliegar en contenedores y agregar monitoreo y alertas.