Evidencias Proyecto final - Kafka + SQLite + Streamlit

Contenido

[Objetivo 2](#_Toc211418460)

[Estructura 2](#_Toc211418461)

[Configuración entorno inicial 2](#_Toc211418462)

[Instalación de WSL y Ubuntu 2](#_Toc211418463)

[Instalación podman dentro de Ubuntu 2](#_Toc211418464)

[Instalación de VS Code – extensiones 2](#_Toc211418465)

[Pasos a tener en cuenta para adelantar guía proyecto final 3](#_Toc211418466)

[Levantar Kafka 3](#_Toc211418467)

[Preparar entorno Python 3](#_Toc211418468)

[Crear BD 4](#_Toc211418469)

[Ejecutar panel 4](#_Toc211418470)

[Notas 7](#_Toc211418471)

[Troubleshooting 9](#_Toc211418472)

[Conclusión 9](#_Toc211418473)

## Objetivo

Leer datos desde SQLite.

Publicar cada fila como JSON en un tópico de Kafka.

Consumir esos mensajes desde una UI en Streamlit.

## Estructura

kafka\_proy\_final/

├── docker-compose.yml # (opcional, no usado en Podman)

├── podman-compose.yml # Servicios Kafka + Zookeeper + Kafdrop

├── requirements.txt # Dependencias de Python

├── README.md # Instrucciones del proyecto

├── src/

│ ├── app\_streamlit.py # Interfaz principal (DB → Kafka → Consumo)

│ ├── create\_db.py # Script para crear sample.db con 3 filas

│ └── sample.db # Base de datos SQLite generada

└── capturas/ # Evidencias del funcionamiento (archivo de Word con las evidencias)

## Configuración entorno inicial

### Instalación de WSL y Ubuntu

Desde powershell como administrador:  
***wsl --install***

***wsl --set-default-version 2***

***wsl --install -d Ubuntu-24.04***

### Instalación podman dentro de Ubuntu

***sudo apt install -y podman podman-compose***

Se usó **Podman** como alternativa ligera a Docker, lo que permitió levantar los servicios de **Kafka**, **Zookeeper** y **Kafdrop** definidos en podman-compose.yml

### Instalación de VS Code – extensiones

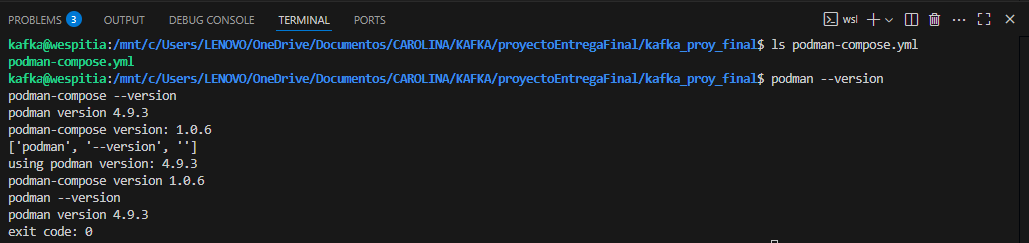
**Python** (para ejecución del entorno virtual y Streamlit)

También se atendió los pasos indicados de la guía de configuración de entorno – curso Apache Kafka en Windows

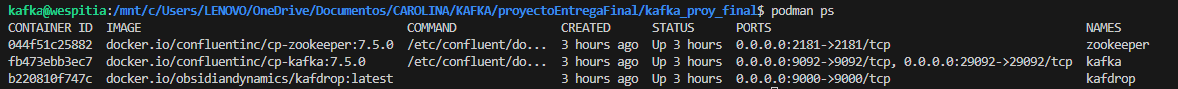


## Pasos a tener en cuenta para adelantar guía proyecto final

### Levantar Kafka

Previamente verificar que exista el ***podman compose*** en el proyecto 

***podman-compose up -d*listar servers:**

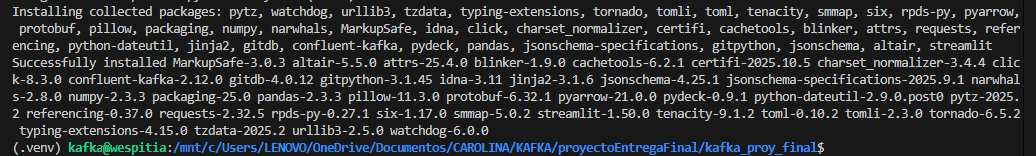
******

### Preparar entorno Python

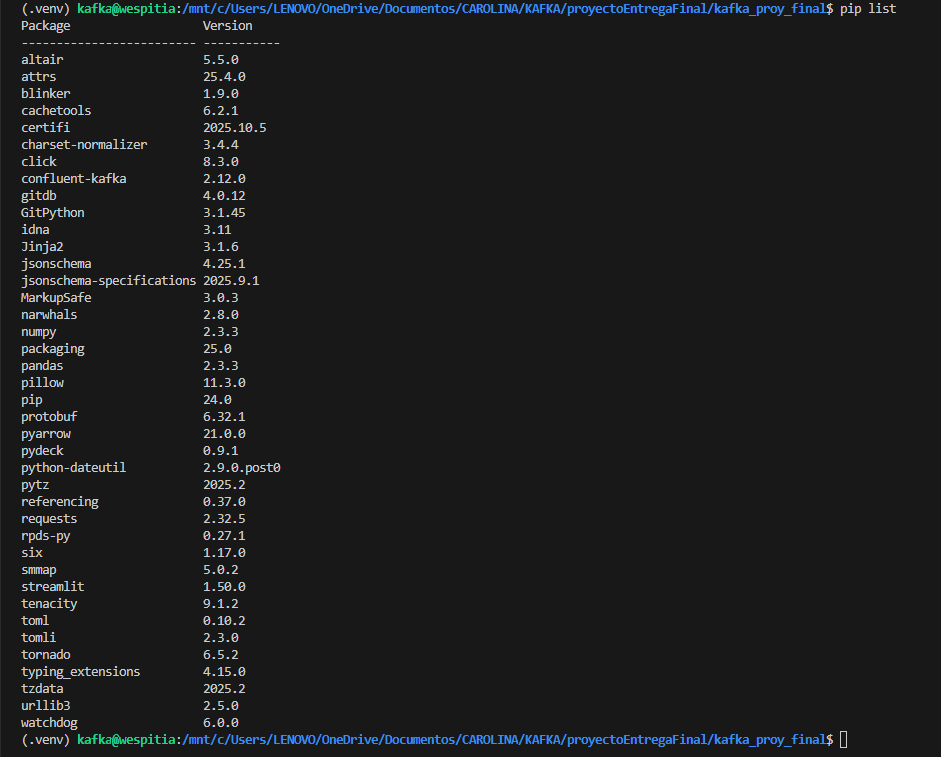
python3 -m venv .venv

source .venv/bin/activate

pip install -r requirements.txt



Comprobar entorno: ***pip list***

****

### Crear BD

****

### Ejecutar panel

Durante las pruebas iniciales, al ejecutar el comando:

***streamlit run app\_streamlit.py***

la aplicación se abría en la dirección por defecto:

***http://localhost:8501***

Sin embargo, Kafka estaba ejecutándose dentro de un contenedor en WSL2 (Ubuntu) usando Podman, lo cual implica que el contenedor no comparte directamente el localhost del sistema Windows.

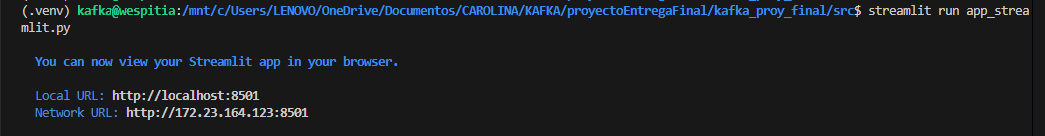
Por ello, aunque Streamlit funcionaba, la aplicación no podía conectarse con el bróker de Kafka.  
La Solución: obtener la IP real del entorno WSL con el comando: ***hostname -I***

Lo que arrojo como resultado la Ip 172.23.164.123, por ello previamente En el archivo podman-compose.yml se reemplazó en la linea:

***KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT://localhost:9092,PLAINTEXT\_INTERNAL://kafka:29092***

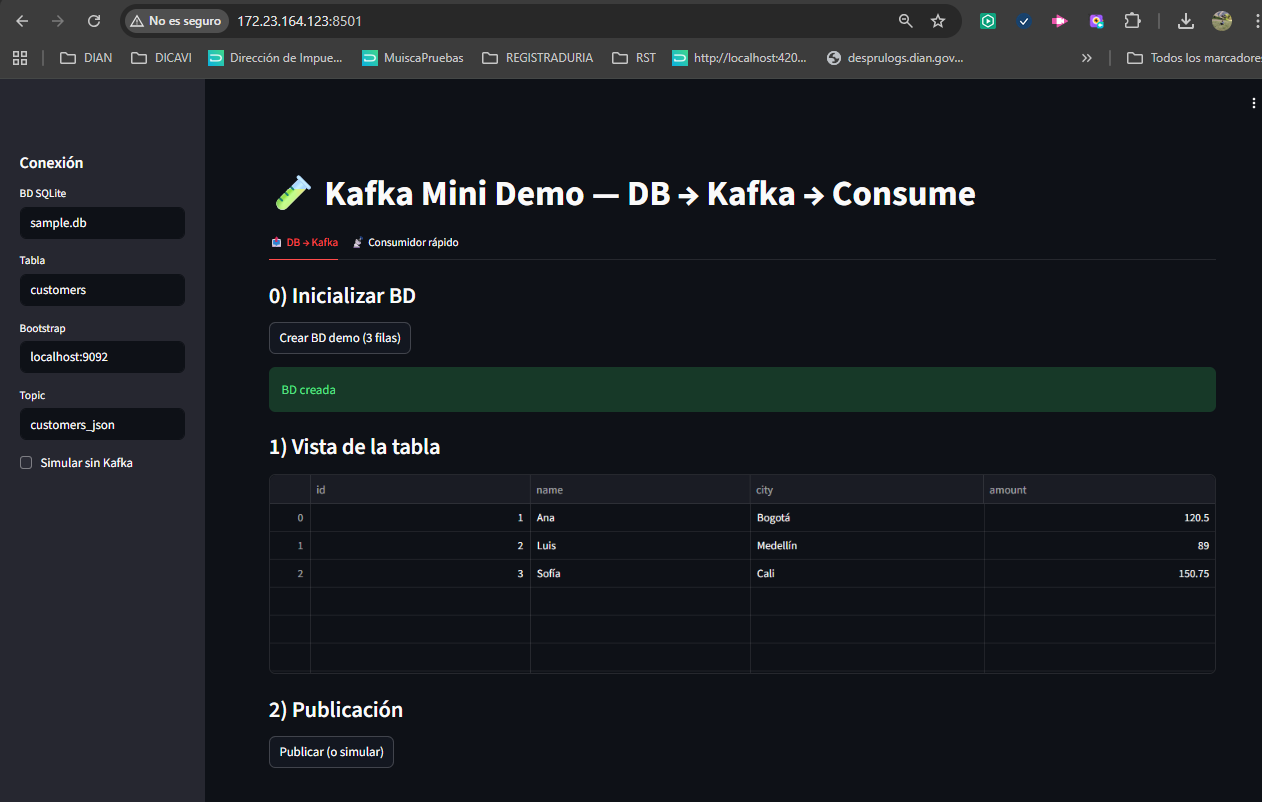
Por:

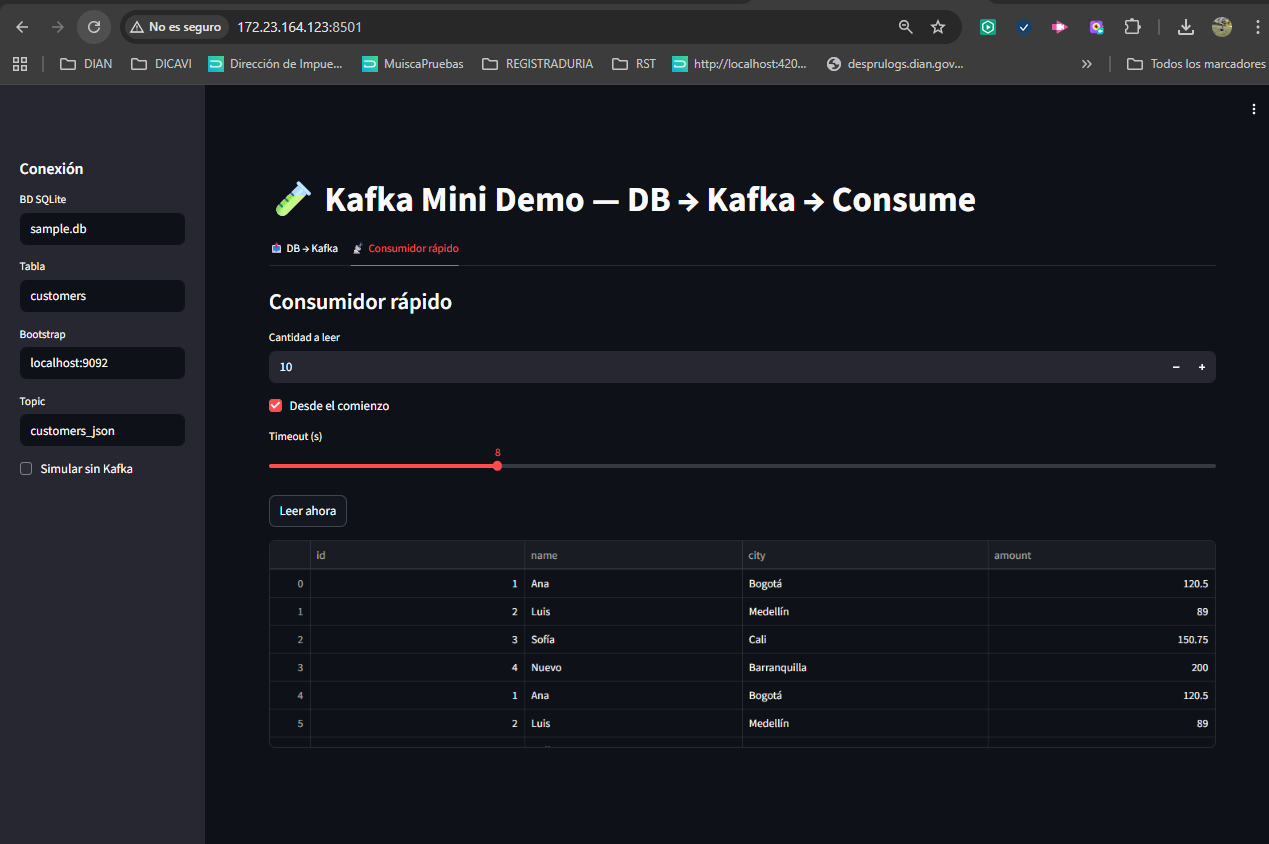
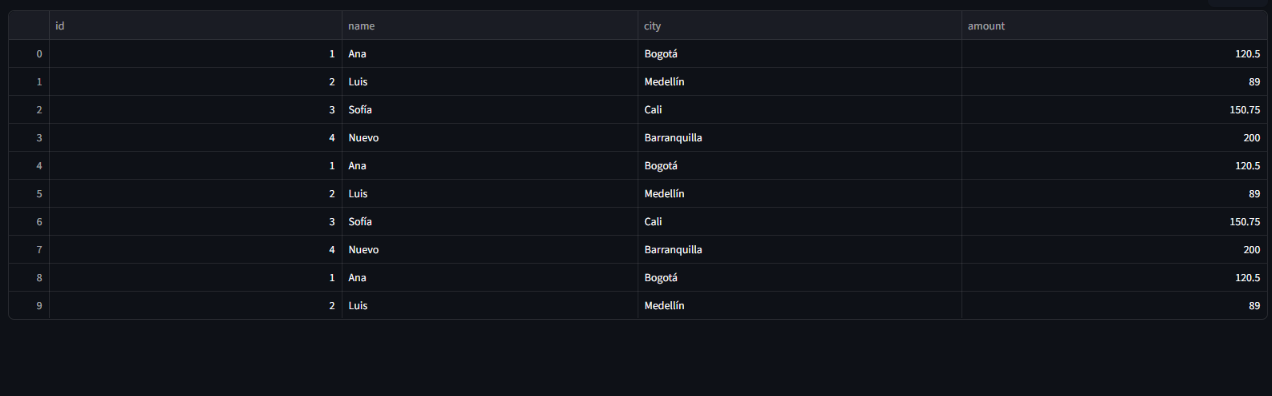
***KAFKA\_ADVERTISED\_LISTENERS: PLAINTEXT://172.23.164.123:9092,PLAINTEXT\_INTERNAL://kafka:29092***



Finalmente, Streamlit se accedió correctamente desde el navegador en:  
**http://172.23.164.123:8501**

Creación y publicación del tópico, desde el panel

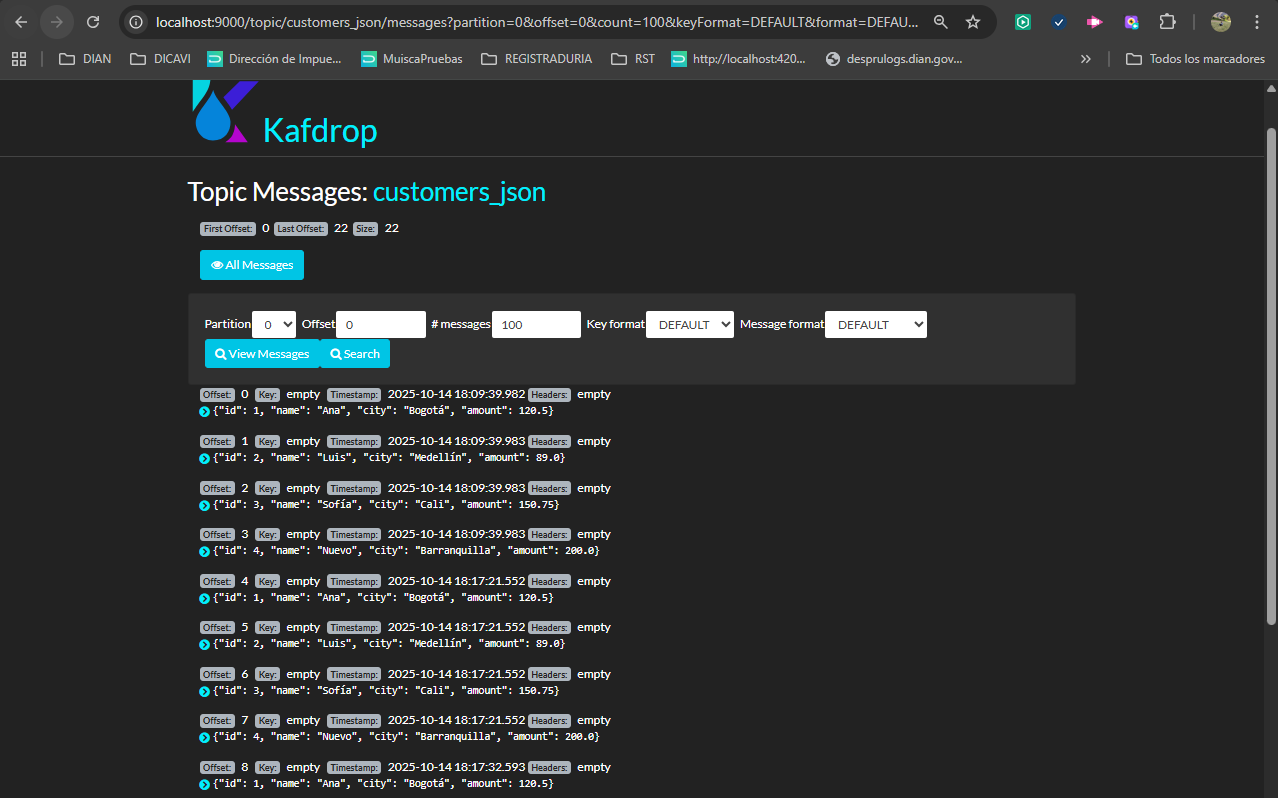


Lee mensajes del tópico - publicación 

### Notas

Para el ejercicio se cambió ruta de la bd, bootstrap y topic(desde el panel), por lo que se generó error porque no encuentra la base de datos en la ruta indicada “**unable to open database file**” (imagen)



Mensajes vistos desde kafdrop – tópico customers\_json  
 

### Troubleshooting

| **Problema** | **Causa** | **Solución** |
| --- | --- | --- |
| gio: Operation not supported | Streamlit intenta abrir navegador desde WSL | Abrir manualmente http://localhost:8501 |
| unable to open database file | Ruta errónea de la BD | Usar src/sample.db en lugar de /sample.db |
| use\_container\_width warning | Aviso de versión futura | Sin impacto funcional |
| No matching distribution confluent-kafka | Versión incompatible | Usar 2.9.0 |
| Venv en /mnt/... falla | WSL bloquea enlaces simbólicos | Crear entorno con python3 -m venv .venv --copies |

### Conclusión

Se logró implementar correctamente el flujo SQLite → Kafka → Streamlit, publicando y consumiendo mensajes JSON en un entorno híbrido Windows + WSL2 + Podman.

Todas las etapas (creación de BD, publicación y consumo) fueron verificadas visualmente en Streamlit y Kafdrop.

Nota: el proyecto se encuentra en git, de manera pública: <https://github.com/davileza23/TallerFinalKAFKA/tree/main/kafka_proy_final>