Informe Final

Sistema ETL de Web Scraping con arquitectura de lago de datos

Objetivo

Este proyecto implementa un sistema ETL (Extraer, Transformar, Cargar) completo que combina los requisitos de Tarea 1 y Tarea 2. El sistema extrae datos utilizando Scrapy, los transforma a través de canales de validación y los almacena en una arquitectura de lago de datos de tres niveles. Además, se creó un panel Streamlit para visualizar los datos procesados.
.

1. Decisiones de diseño

Se usaron las siguientes fuentes de la tarea 1 y se añadió una fuente de noticias adicional:

- https://www.npr.org/
- https://www.aljazeera.com
- https://quotes.toscrape.com/
- http://www.nytimes.com

Se definieron ítems personalizados con validaciones específicas (por ejemplo, “strip()” y campos obligatorios).

Se desarrolló un **“pipeline”** que limpia, valida y guarda simultáneamente en JSONL y PostgreSQL.

Se usó un archivo **“. jsonl”** en lugar de **“. json”** para evitar problemas de duplicación o estructura.

2. Extracción, limpieza y validación

Se desarrollaron 4 spiders para las fuentes mencionadas.

Se configuró USER_AGENT, DOWNLOAD_DELAY, y ROBOTSTXT_OBEY para scraping responsable y ético.

La ejecución se automatiza cada 2 días mediante un script .bat.

Se empleó XMLFeedSpider para parsear RSS correctamente.

La limpieza se implementó en el archivo pipelines.py, donde se validaron y transformaron los campos requeridos:

- **title:** Eliminación de espacios y caracteres especiales (`strip()`).
- **url:** Validación de formato URL y unicidad en BD.

- **date:** Transformación al formato estándar ISO YYYY-MM-DD.
- **source:** Asignación automática según el spider.
- **summary:** Limpieza de saltos de línea y truncamiento si es largo.

Inserción en PostgreSQL con control de duplicados usando “ON CONFLICT DO NOTHING”. Además, se implementó control de duplicados tanto en el JSONL como en la base de datos PostgreSQL, utilizando la URL como clave única.

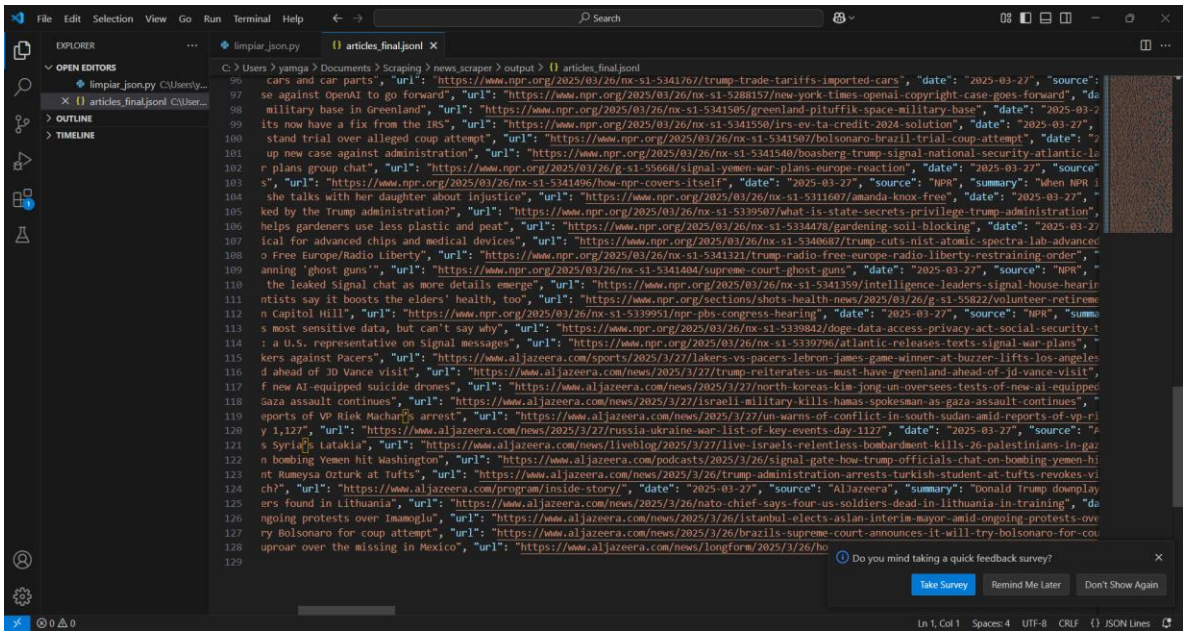
Codificación en UTF-8 para soportar caracteres especiales.

Manejo de errores mediante try/except.

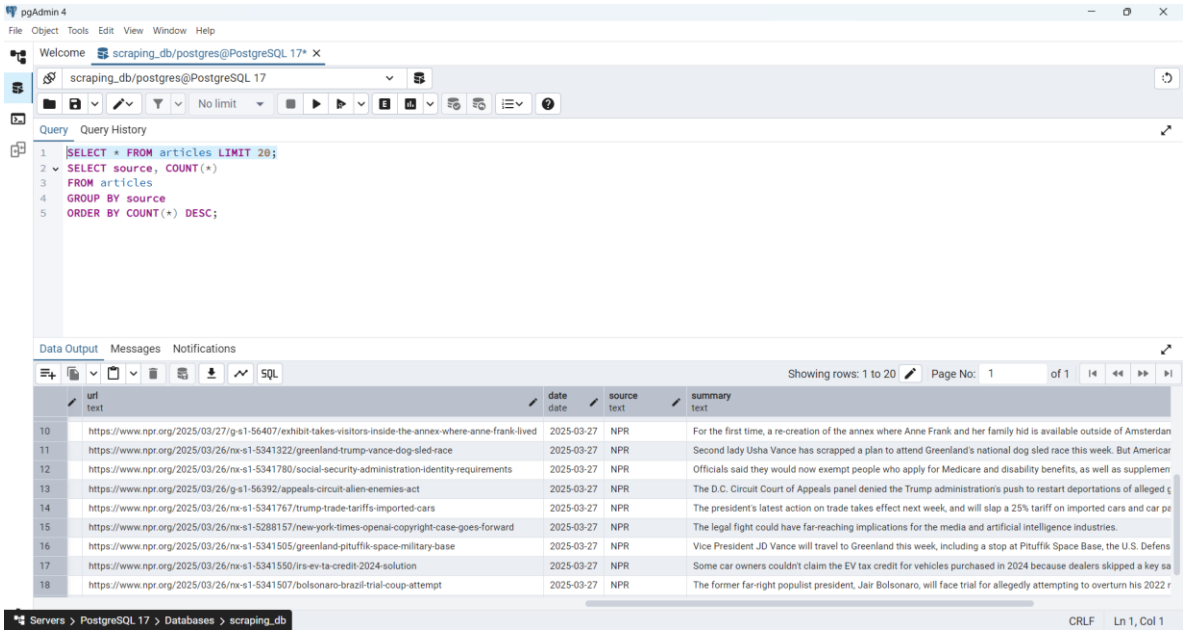
2.1.Almacenamiento de datos

Los datos procesados se almacenan en:

- **output/articles_final.jsonl:** Archivo acumulativo en formato JSONL (una línea por artículo), sin duplicados.



- **Base de datos PostgreSQL:** Conexión usando psycopg2, se creó la base de datos “scraping_db” y los datos se guardan en la tabla articles que contiene claves únicas basadas en la URL para evitar duplicados.



3. Flujo de trabajo

El flujo de trabajo del scraping es el siguiente:

1. Ejecución del spider (Scrapy)
2. Extracción de los artículos
3. Limpieza de los campos con validaciones en pipelines.py
4. Almacenamiento simultáneo en:
    - Archivo JSONL (output/articles_final.jsonl)
    - Tabla articles en PostgreSQL

4. Arquitectura del Datalake

- **Landing_Zone:** Contiene los datos crudos exportados directamente desde el spider.
- **Refined_Zone:** Contiene los datos ya limpiados, transformados y validados, extraídos desde PostgreSQL
- **Consumption_Zone**: Almacena vistas resumidas listas para análisis y visualización.

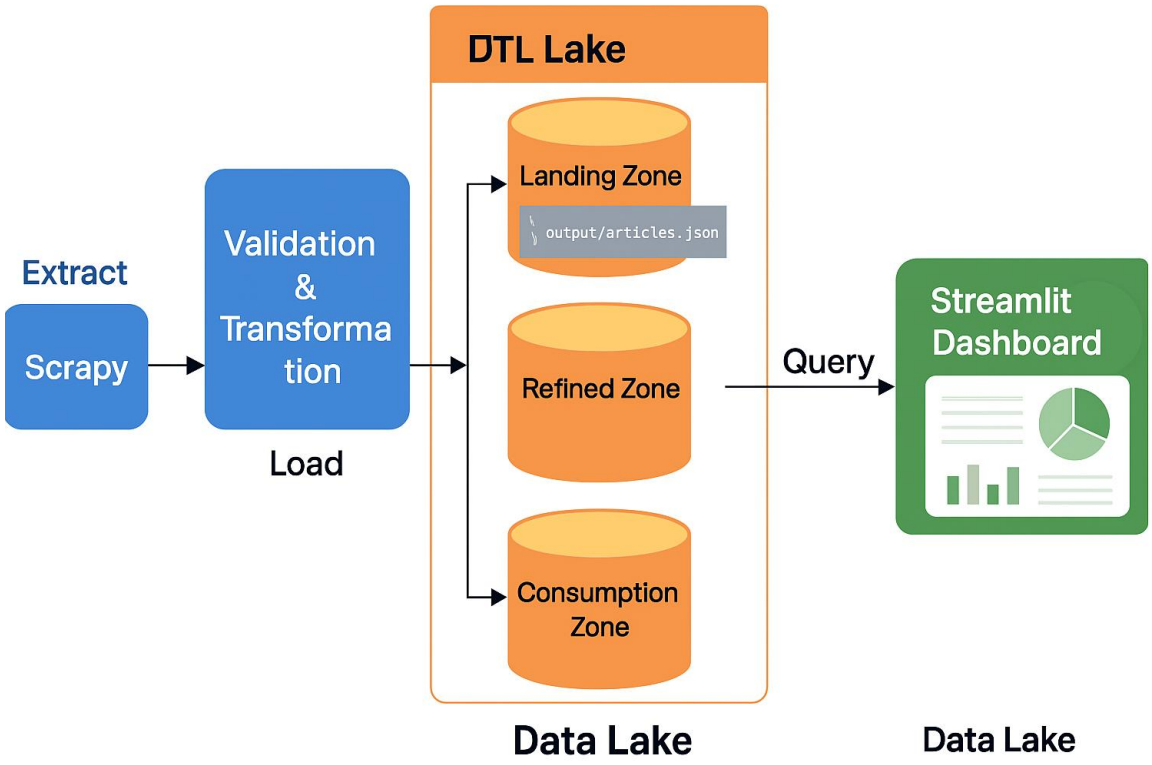
Con la siguiente estructura:

datalake/

├── LANDING_ZONE/articles_raw.jsonl

├── REFINED_ZONE/articles_postgres.csv

└── CONSUMPTION_ZONE/articles_summary.csv



5. **Dashboard en Streamlit**

Visualiza los datos de la Zona de Consumo conectando a PostgreSQL.

Permite filtrar por fecha usando st.date_input().

Incluye:

Título principal del panel

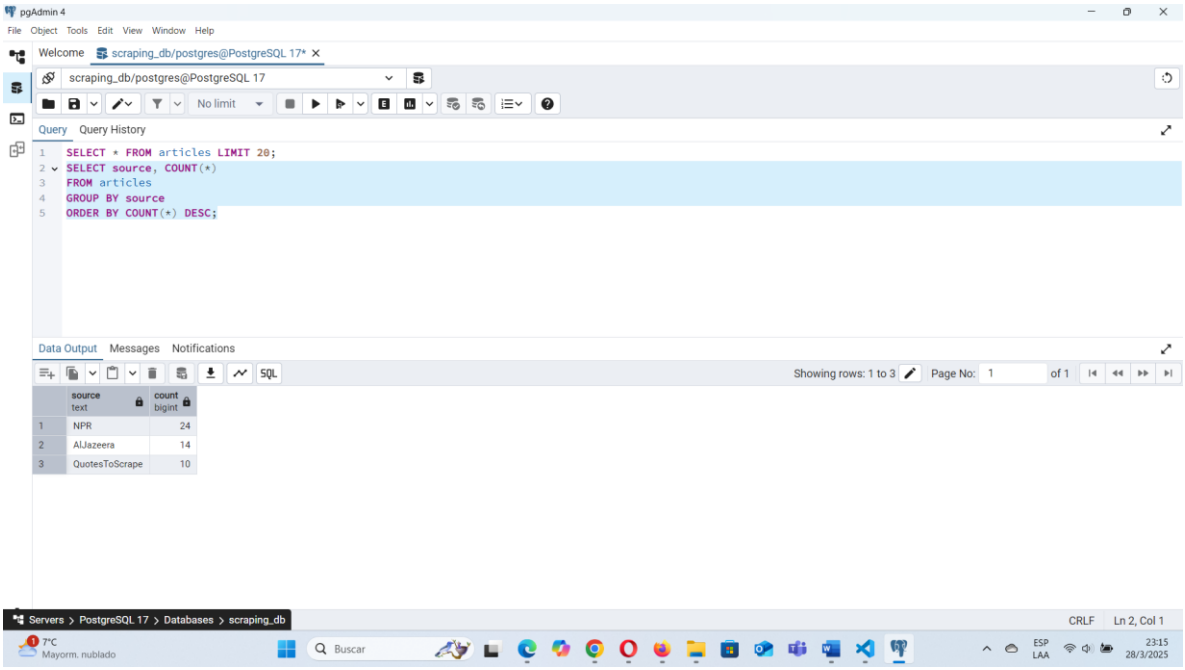
- Total, de artículos scrapeados (con st.metric())
- Tabla de artículos por fuente (st.dataframe())
- Gráfico de barras (st.bar_chart())

Se integra con la API externa **OpenWeatherMap**:

- El usuario ingresa una ciudad
- Se muestra el clima actual (temperatura, descripción, humedad)

6. **Conclusiones**

Se comprobó que los registros en la base de datos y JSONL no se duplican al re- ejecutar los spiders.



Se ejecutó un script de validación (output/ contar_articulos.py) para contar artículos por fuente y verificar unicidad de URLs.

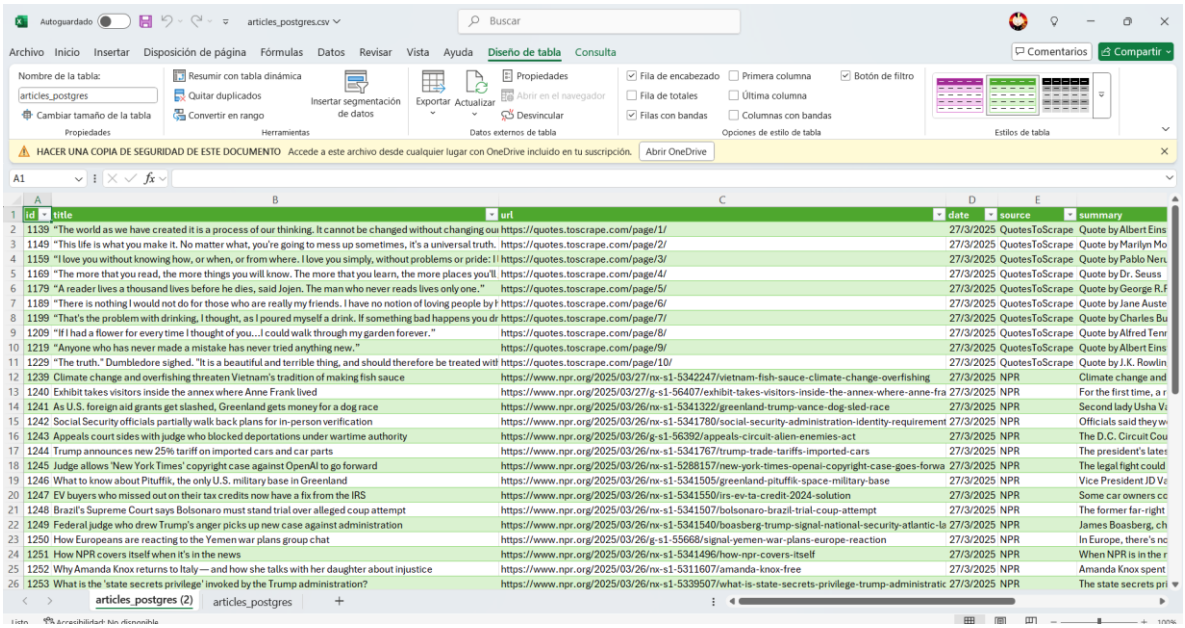
```
C:\Users\yamga\Documents\Scraping\news_scraper\output>python contar_articulos.py
📊 Total de URLs únicas: 48

🗂️ URLs únicas por fuente:
• QuotesToScrape: 10
• NPR: 24
• AlJazeera: 14

C:\Users\yamga\Documents\Scraping\news_scraper\output>
```

Y ambos arrojan los mismos resultados.

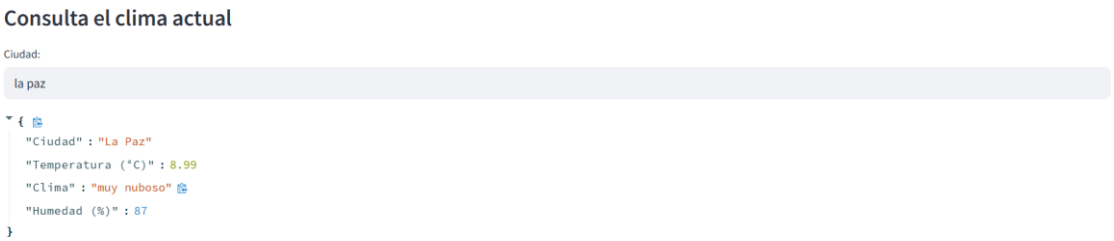
Se extrajo la base de datos desde el PostgreSQL en formato csv en (output/ articles_postgres.csv)



id	title	url	date	source	summary
1139	"The world as we have created it is a process of our thinking. It cannot be changed without changing ou	https://quotes.toscrape.com/page/1/	27/3/2025	QuotesToScrape	Quote by Albert Eins
1149	"This life is what you make it. No matter what, you're going to mess up sometimes, it's a universal truth.	https://quotes.toscrape.com/page/2/	27/3/2025	QuotesToScrape	Quote by Marilyn Mo
1159	"I love you without knowing how, or when, or from where. I love you simply, without problems or pride: I	https://quotes.toscrape.com/page/3/	27/3/2025	QuotesToScrape	Quote by Pablo Neru
1169	"The more that you read, the more things you will know. The more that you learn, the more places you'll	https://quotes.toscrape.com/page/4/	27/3/2025	QuotesToScrape	Quote by Dr. Seuss
1179	"A reader lives a thousand lives before he dies, said Jojen. The man who never reads lives only one."	https://quotes.toscrape.com/page/5/	27/3/2025	QuotesToScrape	Quote by George R.R
1189	"There is nothing I would not do for those who are really my friends. I have no notion of loving people by h	https://quotes.toscrape.com/page/6/	27/3/2025	QuotesToScrape	Quote by Jane Auste
1199	"That's the problem with drinking, I thought, as I poured myself a drink. If something bad happens you dr	https://quotes.toscrape.com/page/7/	27/3/2025	QuotesToScrape	Quote by Charles Bu
1209	"If I had a flower for every time I thought of you...I could walk through my garden forever."	https://quotes.toscrape.com/page/8/	27/3/2025	QuotesToScrape	Quote by Alfred Tenn
1219	"Anyone who has never made a mistake has never tried anything new."	https://quotes.toscrape.com/page/9/	27/3/2025	QuotesToScrape	Quote by Albert Eins
1229	"The truth." Dumbledore sighed. "It is a beautiful and terrible thing, and should therefore be treated with	https://quotes.toscrape.com/page/10/	27/3/2025	QuotesToScrape	Quote by J.K. Rowlin
1239	Climate change and overfishing threaten Vietnam's tradition of making fish sauce	https://www.npr.org/2025/03/27/nx-s1-5342247/vietnam-fish-sauce-climate-change-overfishing	27/3/2025	NPR	Climate change and
1240	Exhibit takes visitors inside the annex where Anne Frank lived	https://www.npr.org/2025/03/27/g-s1-56407/exhibit-takes-visitors-inside-the-annex-where-anne-fra	27/3/2025	NPR	For the first time, a r
1241	As U.S. foreign aid grants get slashed, Greenland gets money for a dog race	https://www.npr.org/2025/03/26/nx-s1-5341322/greenland-trump-vance-dog-sled-race	27/3/2025	NPR	Second lady Usha Va
1242	Social Security officials partially walk back plans for in-person verification	https://www.npr.org/2025/03/26/nx-s1-5341780/social-security-administration-identity-requirement	27/3/2025	NPR	Officials said they w
1243	Appeals court sides with judge who blocked deportations under wartime authority	https://www.npr.org/2025/03/26/g-s1-56392/appeals-circuit-alien-enemies-act	27/3/2025	NPR	The D.C. Circuit Cou
1244	Trump announces new 25% tariff on imported cars and car parts	https://www.npr.org/2025/03/26/nx-s1-5341767/trump-trade-tariffs-imported-cars	27/3/2025	NPR	The president's lates
1245	Judge allows 'New York Times' copyright case against OpenAI to go forward	https://www.npr.org/2025/03/26/nx-s1-5288157/new-york-times-openai-copyright-case-goes-forwa	27/3/2025	NPR	The legal fight could
1246	What to know about Pituffik, the only U.S. military base in Greenland	https://www.npr.org/2025/03/26/nx-s1-5341505/greenland-pituffik-space-military-base	27/3/2025	NPR	Vice President JD Va
1247	EV buyers who missed out on their tax credits now have a fix from the IRS	https://www.npr.org/2025/03/26/nx-s1-5341550/irs-ev-ta-credit-2024-solution	27/3/2025	NPR	Some car owners cc
1248	Brazil's Supreme Court says Bolsonaro must stand trial over alleged coup attempt	https://www.npr.org/2025/03/26/nx-s1-5341507/bolsonaro-brazil-trial-coup-attempt	27/3/2025	NPR	The former far-right
1249	Federal judge who drew Trump's anger picks up new case against administration	https://www.npr.org/2025/03/26/nx-s1-5341540/boasberg-trump-signal-national-security-atlantic-la	27/3/2025	NPR	James Boasberg, ch
1250	How Europeans are reacting to the Yemen war plans group chat	https://www.npr.org/2025/03/26/g-s1-55668/signal-yemen-war-plans-europe-reaction	27/3/2025	NPR	In Europe, there's nc
1251	How NPR covers itself when it's in the news	https://www.npr.org/2025/03/26/nx-s1-5341496/how-npr-covers-itself	27/3/2025	NPR	When NPR is in the r
1252	Why Amanda Knox returns to Italy — and how she talks with her daughter about injustice	https://www.npr.org/2025/03/26/nx-s1-5311607/amanda-knox-free	27/3/2025	NPR	Amanda Knox spent
1253	What is the 'state secrets privilege' invoked by the Trump administration?	https://www.npr.org/2025/03/26/nx-s1-5339507/what-is-state-secrets-privilege-trump-administratic	27/3/2025	NPR	The state secrets pri

Se ejecuto el dashboard con los datos que agarran los spiders.

Y se añadió una API para conocer los datos del tiempo según la ciudad.



6.1.Scripts auxiliares incluidos

- **contar_articulos.py:** Script para contar artículos totales y por fuente.
- **limpiar_postgres.py:** Script opcional para borrar todos los datos de la tabla articles.
- **limpiar_json.py:** Script para limpiar el archivo JSONL si se requiere.
- **exportar_postgres_csv.py:** Script usa csv.writer con el parámetro quotechar='"' y quoting=csv.QUOTE_ALL para que cada campo quede entre comillas y exporte desde PostgreSQL respetando la estructura de las columnas.
- **exportar_resumen.py:** Hace la conexión a la base de datos, y realiza la consulta para exportar el summary.

automatización

ejecutar_scrapers.bat para ejecutar spiders cada 2 días periódicamente.

6.2.Problemas Resueltos

- **Duplicación de artículos:** Se resolvió con una restricción `UNIQUE` en el campo "url".
- **Inconsistencias en codificación:** Se forzó `utf-8` y manejo de errores con try/except.
- **Error exportación desde PostgreSQL:** Se codifico un script que exporta correctamente los datos almacenados respetando los delimitadores de columnas.

7. Conclusiones

El sistema es funcional, automatizado, modular y escalable. La estructura de almacenamiento en (JSON + PostgreSQL) garantiza que sea más fácil para el análisis. La implementación de control de duplicados asegura la calidad del dataset evitando errores de duplicados o datos sobrescritos.

Se aplicó correctamente el patrón ETL con tres zonas del lago de datos.

El uso de Scrapy y Streamlit permitió integrar extracción, transformación y visualización de forma eficiente.

La integración con una API externa enriquece la experiencia del usuario y permite combinar fuentes estructuradas con servicios en tiempo real.