

Proyecto Integrado V - línea de Énfasis (Entrega 1)

Katherine Rojas Bello

Hugo Carvajal González

ingeniería en Desarrollo de software y datos

Proyecto integrado 5

Junio de 2025

introducción

A continuación, presentamos un proyecto que se basa en recolectar datos en este caso en el precio del oro, ya que es uno de los indicadores más observados a nivel global por analistas financieros, inversores y gobiernos. La recolección manual de esta información suele ser tediosa y propensa a errores. Este proyecto se centra en la construcción de un recolector de datos automatizado que garantice la disponibilidad continua de dicha información, donde podemos asegurar su persistencia e integridad.

Resumen

En este proyecto buscamos automatizar la recolección de los precios históricos del oro, permitiendo su almacenamiento tanto en archivos CSV como en bases de datos, mediante un sistema que se programa en Python. La solución implementa una arquitectura modular con enfoque en la trazabilidad, utilizando registros de logs y control de versiones en GitHub. Asimismo, se incorpora automatización continua mediante GitHub Actions, asegurando la actualización constante de los datos.

Objetivos

- Automatizar la descarga del precio del oro desde una fuente confiable, en este caso de <https://finance.yahoo.com/>
- Guardar los datos recolectados en formato CSV y en una base de datos SQLite.
 - Registrar la trazabilidad del proceso mediante logs.
 - Ayudarnos con el desarrollo con Python usando Git y GitHub.
- Automatizar la ejecución periódica del proceso usando GitHub Actions.

Metodología

Clase Collector Esta clase utiliza requests, pandas y BeautifulSoup para hacer scraping de la página de Yahoo Finanzas, extrayendo la tabla de precios históricos del oro. Los datos obtenidos son formateados y retornados como un DataFrame.

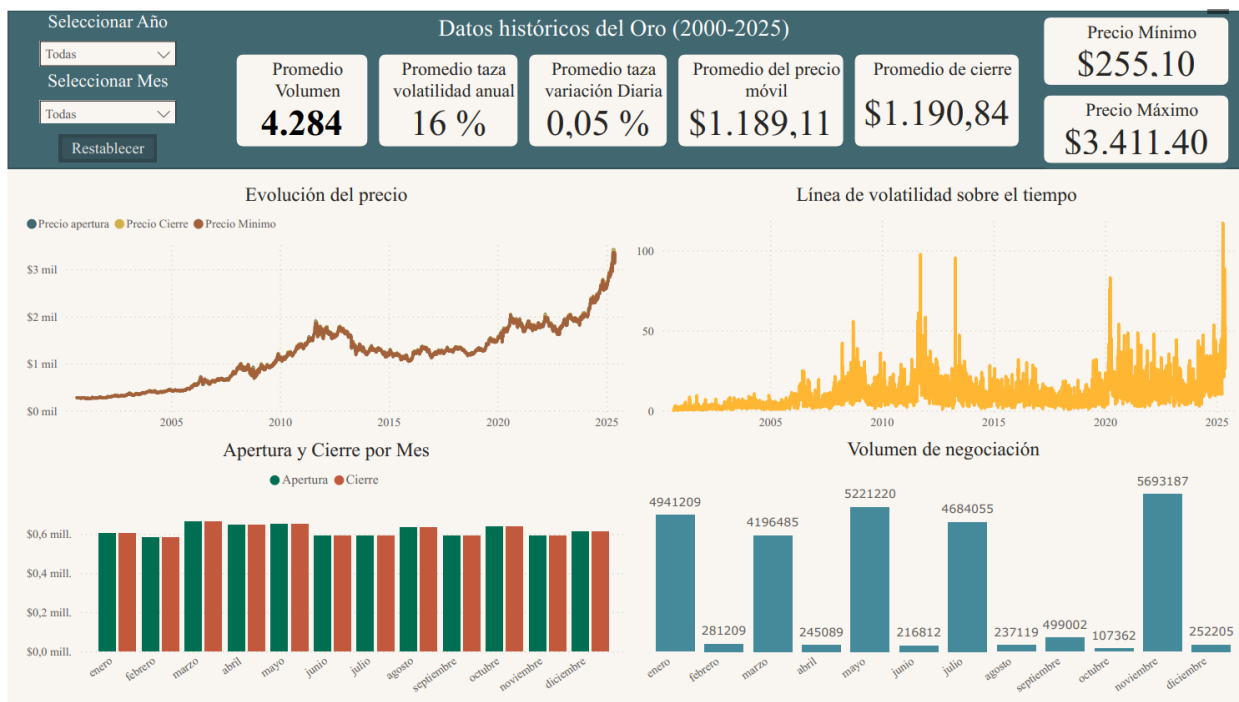
Clase Logger La clase Logger gestiona el registro de logs. Crea un archivo log con nombre dinámico basado en la fecha y hora de ejecución. Usa la librería logging para registrar mensajes de info, warning y error, permitiendo auditar cada ejecución.

Script execution.py Contiene la función main() que instancia las clases Logger y Collector, ejecuta el proceso de recolección de datos y guarda los resultados en un archivo CSV dentro del repositorio.

GitHub Actions El flujo de trabajo definido en .github/workflows/update_data.yml automatiza la ejecución del script de recolección, permitiendo mantener los datos actualizados sin intervención manual.

Resultados Se obtuvo un archivo Oro_datosOriginales.csv que almacena los datos históricos del precio del oro. El sistema de logs permite seguir la pista de posibles errores o fallos en la recolección. La automatización con GitHub Actions asegura la continuidad del proceso sin pérdida del histórico.

Dashboard



Video Explicativo

<https://drive.google.com/file/d/1zDBivSAyTvwUS96BudHcss8bpbQ-W5xM/view?usp=sharing>

Conclusión

La implementación de este sistema automatizado demuestra de una manera eficiente y equilibrada las soluciones modernas para la recolección de datos financieros. El uso como el registro de logs, el modularidad del código y la automatización en la nube permiten crear una herramienta confiable y de fácil mantenimiento.

Referencias

- Python Software Foundation. (2023). *Python 3 Documentation*. <https://docs.python.org/3/>
- Yahoo Finanzas: <https://finance.yahoo.com/>
- BeautifulSoup Documentation. <https://www.crummy.com/software/BeautifulSoup/>
- GitHub Actions Documentation. <https://docs.github.com/en/actions>