



EVIDENCIA DE APRENDIZAJE 2

OPTIMIZANDO LA PRODUCTIVIDAD EN EL MUNDO DEL SOFTWARE

JULIANA MARÍA PEÑA SUAREZ

PROGRAMACIÓN PARA ANÁLISIS DE DATOS - PREICA2501B020065

ANDRÉS FELIPE CALLEJAS JARAMILLO

MEDELLIN-2025

INTRODUCCION

En un mundo cada vez más orientado al análisis de datos en tiempo real, la automatización de procesos y la integración de metodologías DevOps se han convertido en pilares fundamentales para la eficiencia y escalabilidad del desarrollo de software. Esta evidencia de aprendizaje retoma el trabajo iniciado en la Unidad 1, donde se desarrolló un scraper para extraer datos históricos del petróleo WTI desde Yahoo Finance. En esta segunda fase, el enfoque se amplía para incorporar prácticas de integración continua (CI) y control de versiones, alineadas con los principios de DevOps.

Se ha implementado un pipeline de GitHub Actions que permite ejecutar automáticamente el scraper cada vez que se actualiza el repositorio, generando y almacenando el archivo de resultados (data_web.csv) en formato estructurado. Este flujo simula un entorno de despliegue continuo donde los resultados se integran directamente al repositorio, asegurando trazabilidad, reproducibilidad y mantenimiento eficiente del código.

DESCRIPCION DE LA PAGINA Y ARTICULO A ANALIZAR

Yahoo Finance es una plataforma reconocida que proporciona información financiera en tiempo real, incluyendo precios de acciones, noticias económicas y datos históricos de diversos activos. La sección seleccionada para este proyecto es la de datos históricos del petróleo WTI:

<https://es.finance.yahoo.com/quote/CL=F/history/>

Esta página presenta una tabla con los precios diarios de apertura, cierre, máximos, mínimos y volumen de negociación del petróleo WTI, información crucial para Análisis económicos y financieros.

DESCRIPCION DEL TEMA DE INTERES

El petróleo WTI es un referente clave en los mercados energéticos globales. Su precio influye en diversas áreas, desde la inflación hasta las políticas monetarias. Analizar su comportamiento histórico permite identificar tendencias, evaluar riesgos y desarrollar modelos predictivos. Este proyecto se centra en automatizar la extracción de estos datos para facilitar su análisis posterior.

OBJETIVOS

Objetivo General:

Implementar un sistema automatizado de extracción y versionamiento de datos económicos utilizando GitHub como repositorio de control de versiones y GitHub Actions como herramienta de integración continua.

Objetivos Específicos:

Configurar un archivo .yml que automatice el proceso de scraping al detectar cambios en la rama principal del repositorio.

Asegurar que los datos extraídos se guarden automáticamente como archivo .csv dentro del control de versiones.

Aplicar buenas prácticas de ingeniería de software para garantizar modularidad y escalabilidad del código.

Documentar cada parte del proceso en un repositorio público con trazabilidad clara del flujo DevOps aplicado.

METODOLOGIA EMPLEADA EN EL SCRAPING

El proyecto fue desarrollado con un enfoque modular y orientado a buenas prácticas de software.

La lógica de scraping se encapsula en una clase Python (DataWeb), mientras que el punto de entrada se centraliza en main.py, lo cual facilita su integración con otros sistemas o pipelines.

Herramientas utilizadas:

- Git y GitHub: control de versiones, historial del código y colaboración.
- GitHub Actions: automatización de la ejecución del scraper mediante flujos CI/CD.
- Python 3.9: lenguaje principal de programación.
- BeautifulSoup y Requests: para el scraping del HTML.
- pandas: para estructurar, transformar y exportar los datos.
- YAML: para definir los pasos del pipeline en GitHub Actions.

Flujo del pipeline (.github/workflows/accionables.yml):

1. Se activa con cada push a la rama main.
2. Crea un entorno virtual y actualiza pip.
3. Instala dependencias del proyecto mediante `pip install -e ..`
4. Ejecuta el script main.py, que extrae los datos y genera el archivo .csv.
5. Realiza un commit automático con los cambios al repositorio.

Este flujo garantiza que el archivo de datos esté siempre actualizado y versionado.

RESULTADOS Y CONCLUSIONES

La herramienta desarrollada permitió automatizar exitosamente la extracción de datos históricos del petróleo WTI desde el portal Yahoo Finance, logrando almacenarlos en un formato estructurado (CSV) apto para análisis posteriores. Este resultado valida la efectividad del uso de bibliotecas del ecosistema Python, como requests, BeautifulSoup y pandas, en procesos de scraping orientados a la recolección de indicadores económicos públicos.

El proyecto, desarrollado en la ciudad de **Medellín, Antioquia, durante el año 2025**, se enmarca dentro de una práctica académica aplicada con proyección profesional. Su implementación fue concebida como una solución tecnológica adaptable a contextos reales, como el de la empresa **Energéticos del Norte S.A.S.**, la cual requiere integrar datos financieros actualizados en sus análisis de costos, tendencias de mercado y toma de decisiones estratégicas.

La arquitectura modular del código facilita su mantenimiento, escalabilidad y adaptación a otros dominios económicos. Además, constituye una base sólida para integrarse en sistemas de analítica avanzada, modelos predictivos y automatización de reportes periódicos, lo que potencia su valor práctico dentro de sectores que dependen de información financiera precisa y oportuna.

Se logró establecer exitosamente un entorno de integración continua que automatiza por completo el proceso de scraping y versionamiento de los datos históricos del petróleo WTI. Cada actualización al repositorio provoca la ejecución del pipeline, lo cual:

- Elimina pasos manuales repetitivos.
- Asegura la generación actualizada del archivo data_web.csv.
- Versiona automáticamente los resultados, permitiendo auditoría y trazabilidad.

- Mejora la calidad del proyecto al hacerlo más profesional y escalable.

Esta implementación demuestra cómo la combinación de scraping, control de versiones y DevOps puede integrarse en proyectos reales de análisis económico. Además, sienta las bases para incorporar validaciones automáticas, tests unitarios o incluso despliegues en servidores externos.

BIBLIOGRAFIA

- Scrapy. (2025). Scrapy Documentation. Recuperado de <https://docs.scrapy.org/>
- Yahoo Finance. (2025). CL=F - Crude Oil Futures Historical Data. Recuperado de <https://es.finance.yahoo.com/quote/CL=F/history/>
- Python Software Foundation. (2025). *Python (Versión 3.10) [Software]*. Recuperado de <https://www.python.org/>
- BeautifulSoup. (2025). *BeautifulSoup Documentation*. Recuperado de <https://www.crummy.com/software/BeautifulSoup/>
- Git. (2025). *Git Documentation*. Recuperado de <https://git-scm.com/doc>
- Microsoft. (2025). *Visual Studio Code Documentation*. Recuperado de <https://code.visualstudio.com/doc>
- McKinney, W. (2022). *Python for Data Analysis* (3ra ed.). O'Reilly Media.
- Cedeño, L. (2021). *Minería de datos aplicada con Python*. Alfaomega Grupo Editor.
- GitHub. (2025). *GitHub Actions Documentation*. Recuperado de <https://docs.github.com/en/actions>