Evidencia de aprendizaje 1: Proyecto integrado V – Línea de Énfasis (Entrega final)

Asignatura: Proyecto integrado V – Línea de Énfasis
Docente: Andrés Felipe Callejas

Estudiantes:
Elizabeth Alzate M.
Jimy Mora R.

PREICA2501B020128

Ingeniería Software y Datos

Institución: IU Digital de Antioquia
Fecha: 15 junio de 2025

ACTIVIDAD

Objetivo: Integrar los resultados finales del proyecto y comunicar su funcionamiento y hallazgos a través de un video explicativo, demostrando claridad técnica y síntesis.

Instrucciones:

1. Refinar indicadores y conclusiones en el dashboard.
2. Grabar video (5 máximo) que explique:
    - Estructura y funcionamiento del código (collector, modeller, workflows, logging).
    - Indicador económico seleccionado.
    - Modelo predictivo y su métrica.
    - Análisis de los KPI finales.
3. Colocar el video en docs/presentation.mp4 y, opcionalmente, guion en docs/.

DESARROLLO DE LA ACTIVIDAD

Resumen

El presente proyecto se enfoca en la extracción de datos históricos del precio del Ethereum (ETH) desde Yahoo Finance, usando técnicas de web scraping con la biblioteca Beautiful Soup. Se recopiló información de los últimos 5 años, la cual fue almacenada en formato CSV y en una base de datos SQLite, incluyendo los campos de fecha, precio de apertura, precio máximo, precio mínimo, precio de cierre y volumen. Para el registro de la ejecución del proceso, se implementó un sistema de logging con la biblioteca logging de Python. Adicionalmente, se configuró un flujo de trabajo en GitHub Actions para automatizar la ejecución del script, incluyendo la instalación de dependencias y la gestión de la versión de Python.

Introducción

El análisis de datos de criptomonedas, como Ethereum (ETH), es fundamental para comprender su comportamiento en el mercado y tomar decisiones informadas. Ethereum es una plataforma descentralizada de código abierto que establece una cadena de bloques con funcionalidad de contrato inteligente. Su criptomoneda nativa, Ether (ETH), es la segunda más grande del mercado por capitalización. Este proyecto se centra en la extracción automatizada de datos históricos del precio de ETH para su posterior análisis. La automatización se logra mediante un script en Python y la configuración de un flujo de trabajo en GitHub Actions. El objetivo principal es proporcionar una metodología para la obtención y el almacenamiento eficiente de datos de criptomonedas, así como la automatización del proceso de actualización de datos.

Metodología

La metodología de este proyecto se divide en las siguientes etapas:

1. Extracción de datos: Se utilizó la biblioteca Beautiful Soup para realizar web scraping en la fuente de datos del precio de Ethereum (ETH). Se extrajeron los datos históricos de los últimos 5 años, correspondientes a los campos de fecha, precio de apertura, precio máximo, precio mínimo, precio de cierre y volumen. La fuente de datos específica es la

página de Yahoo Finance para ETH-USD. Según la referencia de ethereum.org, Ethereum tiene una oferta actual de 120,731,250.41233969. El último precio conocido de Ethereum es de 2.337,76273072 USD, con una subida del 20,68 en las últimas 24 horas a la fecha y un volumen de negociación de 52 245 378 712,47 USD en 10284 mercados activos.

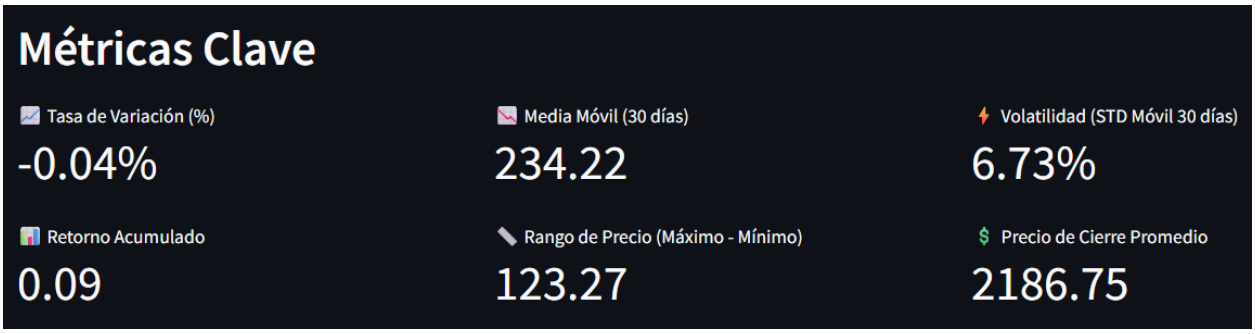
2. Almacenamiento de datos: Los datos extraídos se almacenaron en dos formatos:
    - CSV: Se generó un archivo CSV con los datos históricos.
    - SQLite: Se creó una base de datos SQLite para almacenar los datos, con una tabla que contiene los campos mencionados anteriormente. El archivo collector.py contiene la lógica para la extracción y el almacenamiento de los datos.
3. Logging: Se implementó un sistema de registro de eventos utilizando la biblioteca logging de Python. A través de la clase LoggerConfig en el archivo logger.py, se creó un FileHandler para guardar los registros en el archivo collector.log. Se registran mensajes informativos y críticos sobre la ejecución del script.
4. Enriquecimiento: Una vez recolectados los datos, este script procesa historical.db y historical.csv añadiéndoles características temporales adicionales (como año, mes, día de la semana, etc.) que son cruciales para el análisis posterior. Los datos enriquecidos se guardan en una nueva base de datos: enriched_historical.db

    python src/enricher.py
5. Modelo Arima: Posteriormente, con los datos enriquecidos, se procede a entrenar el modelo ARIMA de predicción y guarda el modelo entrenado como model.pkl.

    python src/modeller.py
6. App.py: Luego de obtener los resultados del modelo, creamos las visualizaciones, kpis y métricas asociadas a este, tales como: tasa de variación, media móvil, retorno acumulado, rango de precio y precio de cierre promedio. Estas métricas podemos encontrarlas en el dashboards, donde se evidencia el comportamiento diario, mensual o anual de las mismas

    https://kpis-ethereum.streamlit.app/

	date	open	high	low	close	volume	year	month	day	day_of_week	quarter	Price Change %	Moving Average 30
0	2025-06-15 00:00:00	2532.76	2538.76	2526.82	2537.62	0	2025	June	15	Sunday	2	None	None
1	2025-06-13 00:00:00	2651.92	2651.92	2443.96	2579.49	0	2025	June	13	Friday	2	1.65	None
2	2025-06-12 00:00:00	2773.6	2784.26	2619.97	2651.8	0	2025	June	12	Thursday	2	2.8033	None
3	2025-06-11 00:00:00	2813.74	2877.63	2746.46	2773.53	0	2025	June	11	Wednesday	2	4.5905	None
4	2025-06-10 00:00:00	2681.29	2824.81	2658.68	2813.52	0	2025	June	10	Tuesday	2	1.4418	None



7. Automatización: Luego de configurarse el orden de cada una de las tareas, se creó un flujo de trabajo en GitHub, usando un Git Actions que se ejecuta diariamente, con el fin de realizar todo el proceso de manera autónoma, realizando el recorrido por cada uno de los archivos en el orden descrito y así poder visualizar en tiempo real las gráficas con los cambios de la moneda y se encarga de:

   - Instalar las dependencias necesarias.
   - Configurar la versión de Python.
   - Ejecutar el script de extracción y almacenamiento de datos.

Esta configuración permite que los datos se actualicen de forma automática y periódica, sin perder el histórico de la información.

github/workflows/update_data.yml