

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

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

Estudiantes:
Elizabeth Alzate M.
Jimmy Mora R.

PREICA2501B020128

Ingeniería Software y Datos

Institución: IU Digital de Antioquia
Fecha: 11 mayo de 2025



ACTIVIDAD

Objetivo: Comprobar la habilidad para automatizar la recolección continua de datos históricos de un indicador económico, garantizando su persistencia y trazabilidad en un entorno de control de versiones y documentación inicial clara.

Instrucciones:

1. Seleccionar un indicador distinto de Yahoo Finanzas.
2. Descargar los datos históricos y almacenarlos en:
 - Base de datos SQLite (historical.db con tabla creada) o
 - CSV (historical.csv)
3. Configurar GitHub Actions en `.github/workflows/update_data.yml` para que en cada ejecución:
 - Descargue y actualice los datos sin perder el histórico previo.
4. Implementar en OOP (clases en `.py`) con logs de ejecución:
 - `collector.py`: clase para descarga y persistencia.
 - `logger.py`: configuración de logging.
5. Redactar en formato APA (Título, Resumen, Introducción, Metodología) y colocar el informe en `docs/report_entrega1.pdf`.

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 BeautifulSoup. 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 BeautifulSoup 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](https://etherscan.io), 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. Automatización: Se configuró un flujo de trabajo en GitHub Actions para automatizar la ejecución del script. El flujo de trabajo se define en el archivo `.github/workflows/update_data.yml` 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.