

**Entrega 1:**

**Profesor**

Andrés Felipe Callejas Jaramillo

**Estudiantes**

Davinson Stiven Rincón Campos

Estefanía Jiménez Tabares



**Institución Universitaria Digital De Antioquia.**

**Proyecto Integrado V - Línea de Énfasis**

**Medellín-Ant.**

**11/05/2025**

# **Automatización de la Recolección de Datos Históricos del Indicador TSLA con Persistencia Local y Control de Versiones**

## **Resumen**

Este informe documenta la automatización de la recolección continua de datos históricos del indicador bursátil TSLA desde Yahoo Finanzas, utilizando técnicas de scraping web con BeautifulSoup. La información extraída se almacena tanto en un archivo CSV como en una base de datos SQLite. El sistema se implementa en Python bajo el paradigma de programación orientada a objetos, con registros de ejecución mediante logs. Además, se configura una acción automática en GitHub Actions que permite ejecutar el proceso en la nube cada vez que se actualiza el repositorio. La solución garantiza trazabilidad, persistencia y control de versiones del dataset actualizado.

## **Introducción**

La recolección automatizada de indicadores financieros es fundamental para la toma de decisiones basada en datos. Este proyecto tiene como objetivo demostrar la capacidad de integrar la extracción de datos históricos de Yahoo Finanzas para el indicador TSLA (Tesla Inc.), garantizando su persistencia local y trazabilidad en un entorno versionado. Para ello, se utiliza Python, técnicas de scraping, estructuras de almacenamiento en CSV y SQLite, y se configura un entorno de integración continua con GitHub Actions.

## **Metodología**

La metodología utilizada se divide en cuatro componentes principales:

1. ***Extracción de Datos:*** Se desarrolla un script que utiliza la biblioteca requests para consultar Yahoo Finanzas y BeautifulSoup para analizar y extraer la tabla de precios históricos de TSLA. La estructura HTML es interpretada dinámicamente para recolectar fechas, precios de apertura, máximos, mínimos, cierre, ajuste y volumen diario.
2. ***Procesamiento y Almacenamiento:*** Los datos extraídos se almacenan en dos formatos:
  - CSV para accesibilidad y manipulación rápida.
  - SQLite para consultas estructuradas y futuras integraciones con análisis avanzados.
3. ***Programación Orientada a Objetos (OOP):*** Se diseñan dos clases principales:
  - Logger para manejar trazabilidad mediante archivos de logs.
  - Collector para encapsular la lógica de recolección y almacenamiento de datos.
4. ***Automatización con GitHub Actions:*** Se configura un flujo de trabajo (update\_data.yml) que activa la ejecución automática del script al detectar cambios en el repositorio, garantizando una integración continua del histórico financiero actualizado.

## **Conclusiones**

El proyecto demuestra cómo es posible automatizar de manera eficiente y trazable la recolección de datos financieros con herramientas modernas de software. La implementación propuesta permite mantener actualizado un dataset de alta calidad, integrándose con herramientas de control de versiones y ejecución automática. Este enfoque es aplicable a múltiples indicadores económicos y puede ser extendido a sistemas de análisis o visualización.