

# **AUTOMATIZACIÓN Y RECOLECCIÓN DE DATOS HISTÓRICOS BTC-EUR**

LAURA ARBOLEDA GALLEGO  
GIORDAN JESE RICARDO PARRA

PROYECTO INTEGRADO V

ANDRÉS FELIPE CALLEJAS

INGENERÍA DE SOFTWARE Y DATOS

IU DIGITAL DE ANTIOQUIA

MEDELLÍN

2025

## Resumen

Este proyecto consiste en la creación de un sistema automatizado en la recolección y almacenamiento de datos históricos del precio del Bitcoin en Euros, desde su creación el 14 de septiembre del 2014 hasta que nuestros días. Estos datos son obtenidos de Yahoo Finanzas y para su estudio se utilizaron herramientas como Python y la implementación de técnicas como el *Web Scraping*. El proceso de datos se desarrolló mediante Pandas y la información fue almacenada en dos formatos diferentes: CSV y en una tabla de base de datos SQLite. Como resultado de esta práctica, se desarrolló una aplicación funcional capaz de consultar y almacenar datos financieros, todo esto integrado a un Main que es capaz de correr mediante flujos automatizado, escalando a entornos de producción y ejecución, como por ejemplo, en la plataforma de GitHub Actions.

## Introducción

En el presente trabajo, vamos a presentar un sistema automatizado de recopilación, almacenamiento y procesamiento de datos históricos del precio del Bitcoin (BTC) en euros (EUR), utilizando el lenguaje de programación Python. La motivación principal de este proyecto se centra en el análisis de datos financieros en tiempo real, recopilando la información para el desarrollo de balances que nos permitan tener un acercamiento con fines de obtener resultados que amplíen más la información, encontrando estrategias y obteniendo respuestas más acertadas en referencia al cómo se mueve este sistema financiero en estos días.

## Metodología

La metodología aplicada en esta primera parte del proyecto se basó en el desarrollo de un script automatizado que permite la extracción, transformación y almacenamiento de datos históricos de BTC-EUR. Este enfoque se dividió en las siguientes etapas:

- **Recolección de datos:** Se utilizó la biblioteca **request** para hacer una solicitud HTTP al sitio web de Yahoo Finanzas, simulando un navegador mediante un encabezado **User-**

**Agent.** Posteriormente, se usó **BeautifulSoup** para analizar los datos del HTML de la página y localizar la tabla con los datos históricos.

- **Procesamiento de datos:** Una vez los datos son extraídos, comienza su procesamiento y estructuración en un dataframe en **Pandas**, donde se renombran las columnas y se validan los registros, de manera tal que la tabla no tenga datos nulos, incompletos o no estructurados completamente.
- **Registro de eventos:** Se implementó un sistema de **logging** personalizado para monitorear la ejecución del programa, registrando eventos relevantes, como lo son el inicio de clases, los errores de conexión o la creación exitosa de los archivos en la carpeta data.
- **Almacenamiento:** Los datos procesados se guardaron en dos formatos. **CSV** y **SQLite**.
- **Estructura modular del código:** Se organizaron las funcionalidades del código mediante clases independientes, las cuales son **Logger**, **Collector** y **main**, permitiendo la modularidad que facilita la mantenibilidad, pruebas y futuras mejoras. El código en general se corre mediante el main, creando una carpeta de **Logs** donde se nos va a mostrar los datos de conexión, también se va a generar automáticamente la carpeta **static** y **data**, donde se van a guardar los archivos **CSV** y **SQLite**.