

**Evidencia de aprendizaje 1. Automatización de la Recolección de Datos Históricos
de un Indicador Económico**

Adriana María Aguilar Vilorio

Edwin Bernardo Villa Sánchez

Código de grupo:

PREICA2501B020128

Docente:

Andrés Felipe Callejas

Asignatura:

Proyecto integrado V

Programa Ingeniería de Software y Datos

Facultad Ingenierías y Ciencias Agropecuarias

Institución Universitaria Digital de Antioquia

2025

Tabla de Contenido

Resumen.....	3
Introducción	4
Objetivo.....	5
Metodología	6
Resultados	9
Bibliografía	10

Resumen

Este proyecto tiene como objetivo automatizar la recolección continua de datos históricos del precio del dólar australiano, utilizando herramientas de programación en Python y la integración con GitHub Actions. Mediante técnicas de web scraping aplicadas a Yahoo Finanzas, se asegura la persistencia, trazabilidad y disponibilidad de los datos históricos sin intervención manual. La solución está desarrollada bajo buenas prácticas de programación orientada a objetos (POO), cuenta con un sistema de logging configurable para auditoría y seguimiento, y se integra en un entorno de control de versiones utilizando Git y GitHub. El proyecto es fácilmente reproducible dentro de GitHub Codespaces, lo que favorece su uso colaborativo y remoto.

Introducción

La disponibilidad y confiabilidad de los datos financieros son elementos clave para la toma de decisiones en campos como economía, negocios, e ingeniería financiera. La automatización de estos procesos reduce la posibilidad de errores humanos, mejora la eficiencia operativa y facilita la actualización constante de la información.

Este informe describe el diseño e implementación de un sistema automatizado de recolección de datos históricos del tipo de cambio del dólar australiano (ticker: 6A=F), obtenido de la plataforma Yahoo Finanzas. El proyecto está implementado íntegramente en Python, con ejecución programada mediante GitHub Actions y desarrollo en un entorno completamente remoto utilizando GitHub Codespaces.

Objetivo

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.

Metodología

Selección del Indicador

Se eligió el indicador 6A=F, que representa el precio del futuro del dólar australiano, como fuente principal de datos. La URL base utilizada fue:

<https://es.finance.yahoo.com/quote/6A=F/>

Recolección de Datos

Se emplearon las bibliotecas requests y BeautifulSoup para extraer la tabla de datos históricos directamente del HTML de la página web. El contenido extraído incluye: fecha, apertura, máximo, mínimo, cierre, cierre ajustado y volumen.

Limpieza y Estructuración

Los datos se transforman en un DataFrame de pandas, normalizando los valores numéricos y realizando una conversión precisa de fechas al formato datetime. Además, se extraen las componentes de año, mes y día para facilitar futuros análisis.

Diseño Orientado a Objetos

La clase Collector encapsula el proceso de descarga, validación y guardado de datos. Esta clase es fácilmente reutilizable y extensible para futuros indicadores. Por otro lado, la clase Logger gestiona los registros de ejecución, errores y eventos del sistema, con distintos niveles de severidad (INFO, ERROR, DEBUG), y formato estandarizado. El archivo main.py sirve como punto de entrada y orquestación del flujo de datos.

Estructura del Repositorio

El repositorio sigue una estructura modular clara:

<PROYECTOINTEGRADOV_2025>/	
├── .github/	
│ ├── workflows/	
│ │ └── update_data.yml	# Automatización con GitHub Actions
├── src/	
│ └── <indicador_economico>/	
│ ├── static/	
│ │ ├── data/	
│ │ │ ├── dolar_data_enricher.csv	# Dataset enriquecido
│ │ │ ├── historical.csv	# Dataset generado
│ │ │ └── predicciones.csv	# Resultados de predicciones generadas
│ │ ├── models/	
│ │ │ └── model.pkl	# Modelo entrenado serializado
│ │ └── reports/	
│ │ ├── Dashboard.pdf	# Reporte PDF con capturas de gráficos
│ │ └── metricas.txt	# Justificación de métricas del modelo
│ ├── collector.py	# Clase principal para recolección de datos
│ ├── dashboard.py	# Dashboard con visualización de datos históricos
│ ├── dashboard_prediccion.py	# Dashboard con visualización de predicciones
│ ├── enricher.py	# Script para enriquecimiento y procesamiento de datos
│ ├── logger.py	# Configuración y manejo de logs
│ ├── main.py	# Punto de entrada del programa
│ └── modeller.py	# Entrenamiento y evaluación del modelo predictivo
├── docs/	
│ └── report_entrega1.pdf	# Documentación y reporte de entrega
├── logs/	
│ └── dolar_analysis_20250508.log	# Registros y logs de ejecución
├── setup.py	# Archivo de configuración e instalación
└── README.md	# Documentación principal del proyecto

Automatización con GitHub Actions

Se implementó un flujo de trabajo en `.github/workflows/update_data.yml` que se activa automáticamente cada vez que se realiza un push a la rama main. Este workflow configura el entorno, instala las dependencias necesarias y ejecuta el script principal (`main.py`), encargado de recolectar, transformar y guardar los datos históricos. En caso de detectar nuevos datos, el workflow realiza automáticamente un commit y push al repositorio, garantizando la trazabilidad y actualización continua del histórico sin necesidad de intervención manual.

Desarrollo en GitHub Codespaces

Todo el entorno de desarrollo se configura y ejecuta en Codespaces, asegurando portabilidad y facilidad de configuración. Se incluyen extensiones recomendadas como:

- Python
- Pylance
- Jupyter
- CSV Editor
- GitHub Copilot

Resultados

- Se construyó una herramienta funcional y automática para descargar, limpiar y almacenar los datos del dólar australiano.
- Se generó el archivo `historical.csv` con más de 6000 registros desde el año 2000 hasta la fecha actual.
- Se garantiza la persistencia del histórico, ya que las nuevas ejecuciones agregan solo datos no existentes, evitando duplicados.
- Los logs permiten rastrear todas las ejecuciones y errores ocurridos.
- El flujo automatizado de GitHub Actions verifica, ejecuta y versiona cada actualización de manera transparente.
- Todo el proyecto es portable y se ejecuta íntegramente en la nube sin configuración adicional local.

Bibliografía

- GitHub. (2024). *GitHub Actions documentation*. <https://docs.github.com/actions>
- Python Software Foundation. (2024). *Python language reference*. <https://docs.python.org/3/>
- Scikit-learn developers. (2024). *Scikit-learn: Machine learning in Python*. <https://scikit-learn.org/>
- Statsmodels developers. (2024). *Statsmodels: Statistical modeling in Python*. <https://www.statsmodels.org/stable/>
- Yahoo Finanzas. (2024). *Historial de cotizaciones de AUD/USD*. <https://es.finance.yahoo.com/quote/6A%3DF/history/>
- Git Development Community. (2024). *Git (versión 2.40) [Software]*. Recuperado de <https://git-scm.com/>