Documentación del Programa

Dashboard de Finanzas en Python: Criptomonedas, Materias Primas y Acciones Infravaloradas

Autor:Jennifer Paola Coral Reina

Fecha: 16/12/2025

1. Resumen

Este programa implementa un flujo completo de trabajo típico en ciencia de datos aplicada a finanzas: obtención de datos (API y scraping), limpieza y tipificación (conversión a numérico), persistencia en archivos CSV, y visualización mediante gráficos interactivos (Plotly). El usuario puede operar el sistema de dos maneras: un menú por consola (CLI) y una interfaz gráfica (GUI) basada en ttkbootstrap.

2. Objetivos

2.1 Objetivo general

Construir una herramienta simple y reutilizable para consultar y analizar información financiera desde fuentes públicas, aplicando buenas prácticas básicas de Python (modularización, funciones, manejo de errores) y conceptos de ciencia de datos (DataFrames, limpieza y gráficos).

2.2 Objetivos específicos

- Consumir datos mediante requests HTTP (API CoinGecko) y scraping de tablas HTML (Yahoo Finance).
- Transformar la información en DataFrames de pandas para su análisis.
- Normalizar y convertir datos numéricos (porcentajes, volúmenes, capitalización, P/E).
- Guardar resultados en CSV dentro de la carpeta data/ para reutilización.
- Generar gráficos con Plotly para explorar resultados.
- Proveer una GUI que no se congele durante descargas (uso de threads).

3. Requisitos del entorno

3.1 Versión de Python recomendada

Python 3.10 o superior (recomendado).

3.2 Dependencias (requirements.txt)

Estas librerías son las mínimas esperadas para ejecutar el programa con scraping, API, DataFrames, gráficos y GUI:

```
pandas
numpy
requests
plotly
ttkbootstrap
lxml
beautifulsoup4
html5lib
```

3.3 Instalación con entorno virtual (Windows)

Se recomienda trabajar con un entorno virtual para aislar dependencias.

```
python -m venv .venv
.venv\Scripts\activate
python -m pip install -r requirements.txt
```

3.4 Verificación del intérprete activo

Si el programa 'no encuentra' módulos instalados, normalmente se está ejecutando con un Python diferente al del entorno virtual. Estos comandos ayudan a verificarlo:

```
where python
python -c "import sys; print(sys.executable)"
python -m pip -V
```

4. Estructura del proyecto

Una estructura típica de carpetas y archivos es la siguiente:

```
proyecto_finanzas/
├── app_gui.py
├── main.py
├── Funciones.py
├── criptos.py
├── materias_primas.py
├── acciones_infravaloradas.py
└── data/
    ├── criptos.csv
    ├── materias_primas.csv
    └── acciones_infravaloradas.csv
```

Nota: la carpeta data/ se crea automáticamente si no existe. Los CSV se guardan ahí para poder cargar datos sin volver a descargar.

5. Flujo general de funcionamiento

El flujo de trabajo es el mismo para los tres conjuntos de datos:

1. Actualizar: se descarga la información desde la fuente (API (los primeros 100) o página web (los primeros 100)).
2. Filtrar: se seleccionan solo columnas relevantes para el análisis.
3. Limpiar: se convierten columnas numéricas y se eliminan valores inválidos.
4. Persistir: se guarda un CSV en data/.
5. Visualizar: se muestran gráficos y/o se imprime una vista del DataFrame.

Diagrama (alto nivel):

```
Fuente (API/HTML) -> DataFrame -> Limpieza/Tipos -> CSV -> Gráficos/GUI/CLI
```

6. Descripción de módulos

6.1 Funciones.py

Contiene funciones comunes para reutilizar lógica de conexión y scraping. La función principal es conexion_web(url), que descarga la página y extrae tablas HTML con pandas.read_html, tambien contiene la función de limpieza de datos numéricos .

Función: conexion_web(url)

- Entrada: url (str) - dirección de la página a consultar.
- Salida: lista de DataFrames (cada DataFrame representa una tabla encontrada en la página).
- Manejo de errores: resp.raise_for_status() lanza excepción si el servidor responde con error (ej.: 403, 404, 500).

Función: modificacion_tipo_datos(x)

Ya que las pagina tiene sus propias equivalencias como por ejemplo el cambio % viene con el porcentaje en cada fila así que se elimina y cuando usa letras como “M” O “MM” se quita pero convirtiéndolo en el número real así no se manejan números cruzados

6.2 criptos.py (CoinGecko)

Este módulo consume la API pública de CoinGecko para obtener el Top 100 de criptomonedas por capitalización. El objetivo es trabajar con un subconjunto de columnas útiles y persistirlas como CSV.

Funciones principales

- actualizar_info_cripto(): descarga datos desde CoinGecko, filtra columnas, limpia tipos, guarda data/criptos.csv y devuelve el DataFrame.
- cargar_csv_cripto(): lee data/criptos.csv y devuelve el DataFrame para reutilización.
- grafico_nombre_vs_capi(df, top_n=20): gráfico de barras horizontal con Top N por capitalización.
- grafico_rango_atl_ath(df, top_n=20): visualiza rango ATL-ATH y precio actual para Top N por capitalización.

Nota: CoinGecko puede aplicar límites de uso. En caso de error, se recomienda esperar y reintentar más tarde.

6.3 materias_primas.py (Yahoo Finance)

Este módulo realiza scraping de la página de Yahoo Finance (materias primas) y extrae la tabla principal. Luego renombra columnas, convierte valores a numéricos y guarda el CSV para reutilización.

Funciones principales

- actualizar_informacion_mp(): descarga tabla, filtra columnas, renombra, limpia valores y guarda data/materias_primas.csv.
- cargar_csv_mp(): lee data/materias_primas.csv y devuelve el DataFrame.
- graficos_top10(df): gráfico de barras con Top 10 por volumen.
- graficos_todas(df): gráfico general (histograma) con todas las materias primas.

6.4 acciones_infravaloradas.py (Yahoo Screener)

Este módulo hace scraping del screener de Yahoo Finance 'Undervalued Growth Stocks'. Se filtran columnas clave (símbolo, precio, variación, market cap, P/E) y se convierte a numérico para análisis.

Funciones principales

- actualizar_informacion_infra(): descarga tabla, filtra columnas, renombra, limpia valores y guarda data/acciones_infravaloradas.csv.
- cargar_csv_infra(): lee data/acciones_infravaloradas.csv y devuelve el DataFrame.
- ranking_pe_masbajas(df): gráfico de barras con las 20 acciones con P/E más bajo.
- pe_vs_cambio(df): gráfico de dispersión P/E vs % cambio, tamaño por capitalización.

Aclaración: si una columna numérica llega como texto (por ejemplo '1,2B' o 'N/A'), debe convertirse antes de usar nlargest/nsmallest.

6.5 main.py (menú por consola)

Permite operar el programa desde la terminal. El menú agrupa funciones por categoría (Criptos, Materias primas, Acciones infravaloradas) y ejecuta actualización, gráficos o impresión de datos.

Ejecución

```
python main.py
```

6.6 app_gui.py (interfaz gráfica)

La interfaz gráfica está construida con ttkbootstrap. El objetivo es ofrecer una experiencia simple: cada pestaña representa un conjunto de datos y contiene botones para actualizar CSV y ver gráficos.

Elementos principales

- Notebook (pestañas): Criptos, Materias primas, Acciones infravaloradas.
- Treeview: tabla para visualizar los DataFrames (hasta 200 filas para mantener fluidez).
- Botón Actualizar: descarga + guarda + muestra.
- Botones de gráficos: abren gráficos Plotly en una ventana/navegador externo (comportamiento normal de Plotly).
- Botón Salir: cierra la aplicación.

Carga automática al iniciar

Al iniciar, el programa intenta cargar los CSV existentes. Si están disponibles, muestra automáticamente los datos en cada pestaña. Si no existen, la pestaña queda vacía hasta que el usuario presione 'Actualizar'.

Por qué se usa threading

Las descargas web pueden tardar varios segundos. Si se ejecutaran directamente en el hilo principal de Tkinter, la ventana se congelaría. Por eso, la GUI ejecuta tareas lentas en un hilo secundario y luego actualiza la interfaz.

7. Guía de uso

7.1 Uso desde la GUI

6. Abrir la aplicación: python app_gui.py
7. Si no existen CSV o se quieren datos frescos, presionar 'Actualizar'.
8. Usar los botones de gráficos para explorar los datos.
9. Cerrar la aplicación con el botón 'Salir'.

7.2 Uso desde consola (CLI)

10. Ejecutar: python main.py
11. Elegir una sección (1 Criptos, 2 Materias primas, 3 Acciones infravaloradas).
12. Seleccionar la opción deseada: actualizar, gráfico o mostrar información.
13. Volver al menú principal o salir.

8. Manejo de errores y solución de problemas

8.1 El programa no reconoce librerías instaladas

Causa típica: se ejecuta con el Python global en vez del entorno virtual. Solución: activar el entorno o ejecutar usando el python del entorno.

```
.venv\Scripts\activate
python -m pip install -r requirements.txt
python app_gui_actualizado_con_salida_auto_csv.py
```

8.2 CSV no encontrado

Si se intenta 'Cargar CSV' y el archivo no existe, primero se debe presionar 'Actualizar' para generar el CSV inicial dentro de data/.

8.3 Yahoo no devuelve tablas

- Puede ocurrir por cambios de HTML, restricciones del sitio o conexión inestable.
- Reintentar más tarde o mejorar conexion_web agregando timeout y reintentos.
- Verificar que lxml/beautifulsoup4/html5lib estén instalados (pandas.read_html depende de ellos).

8.4 Errores por datos no numéricos

Funciones como nlargest/nsmallest requieren columnas numéricas. Si la fuente entrega valores con sufijos (K, M, B) o textos (N/A), es necesario convertirlos antes de ordenar o graficar.

9. Buenas prácticas y mejoras futuras

Buenas prácticas aplicadas

- Separación por módulos: cada fuente de datos en su archivo.
- Persistencia en CSV para reutilización y performance.
- Limpieza y tipificación antes de análisis/visualización.

- Gráficos como funciones reutilizables (reciben DataFrame por parámetro).
- GUI responsiva usando threading para tareas lentas.

Mejoras futuras posibles

- Guardar histórico por fecha (en vez de sobrescribir CSV).
- Agregar más métricas: variación 7d/30d, medias móviles, alertas.
- Incluir logging y niveles (INFO/WARNING/ERROR) en vez de print.
- Exportar reportes automáticos (PDF/Excel) para portfolio.
- Embed de gráficos dentro de la GUI (más avanzado).

10. Conclusión

El proyecto integra conceptos esenciales de programación y ciencia de datos de manera práctica. Permite adquirir datos financieros, procesarlos en DataFrames, persistirlos para reutilización y analizarlos con gráficos. La existencia de un modo CLI y un modo GUI facilita tanto pruebas técnicas como una presentación más amigable para usuarios finales.