

Certificación Profesional en Python (ITBA)

Trabajo Práctico Final

Matías Huenul

Índice

1. Introducción	2
2. Diseño de la aplicación	2
2.1. Base de datos	2
2.2. Visualizaciones	2
2.3. Interacción con la API de Polygon	2
2.4. Utilidades	2
2.5. Flujo del programa	2
3. Conclusiones	2

1. Introducción

En este trabajo práctico se desarrolló un programa de línea de comandos en Python para obtener información de una API de finanzas, almacenarlas en una base de datos local SQLite3 y graficarlos usando Matplotlib.

2. Diseño de la aplicación

La aplicación se estructuró en módulos reutilizables, y un script principal que se encarga del flujo del programa y la interacción con el usuario.

2.1. Base de datos

El módulo **database** contiene las funciones necesarias para realizar consultas sobre la base de datos. En este caso, se utilizó el motor **SQLite3**, y se definió una única base de datos almacenada en el archivo **tickers.db**. La misma posee una tabla, **tickers**, la cual almacena los valores por ticker y fecha, y tiene la siguiente estructura.

- **symbol (string)**: Símbolo que identifica a un ticker.
- **name (string)**: Nombre del ticker.
- **value (real)**: Valor del ticker.
- **date (string)**: Fecha correspondiente al dato.

2.2. Visualizaciones

El módulo **plot** contiene una función para realizar gráficos de tipo **line plot** utilizando la biblioteca **Matplotlib**.

2.3. Interacción con la API de Polygon

El módulo **polygon** contiene las funciones para comunicarse con la Rest API de Polygon. Se define la función **get_tickers**, la cual realiza un request de tipo **GET** al endpoint **/v3/reference/tickers/{ticker}** y devuelve un diccionario con los datos de un ticker para una fecha dada. En particular, se devuelven los campos que serán almacenados en la base de datos, siendo calculado el valor del ticker como $value = \frac{market_cap}{weighted_shares_outstanding}$, de acuerdo a la documentación oficial de la API.

2.4. Utilidades

El módulo **utils** posee funciones útiles para el programa, como funciones de cálculo de fechas, escritura a archivos e interacción con el usuario.

2.5. Flujo del programa

La lógica principal del programa se encuentra en el script **main.py**. El mismo ejecuta un **wizard** que interactúa con el usuario y lo guía a través de las distintas operaciones disponibles, las cuales se pueden categorizar en tres grupos: **actualización**, **visualización** y **exportación** de datos. Si el usuario elige la primera, se debe ingresar el nombre del ticker a consultar, y las fechas de inicio y fin de la consulta.

3. Conclusiones

Referencias