

Link del repositorio de github: <https://github.com/correafede/TPFinalPython>

## Informe de Funcionalidad y Diseño:

Este programa está diseñado para permitir a los usuarios consultar datos históricos de acciones, ingresar nuevos datos de acciones y generar resúmenes de los datos almacenados. Está compuesto por varias funciones que interactúan entre sí para lograr estas funcionalidades. A continuación se detalla el funcionamiento y diseño del programa:

### 1. Funcionalidad:

- **Consulta de datos:** Permite al usuario seleccionar entre consultar el último valor de una acción o revisar el historial de una acción en un rango de fechas específico.

```
# Preguntar al usuario si quiere hacer una consulta o ingresar datos
accion = input("¿Quiere hacer una consulta, ingresar datos o salir? (consulta/datos/resumen/salir): ")
```

- **Ingreso de datos:** Permite al usuario ingresar nuevos datos de acciones para un rango de fechas determinado.

```
# Solicitar al usuario el valor del ticker, una fecha de inicio y una fecha de fin
ticker = input("Por favor, ingresa el valor del ticker: ")
ticker = ticker.upper() # Convertir el ticker a mayúsculas
fecha_inicio = convertir_fecha(input("Por favor, ingresa la fecha de inicio (formato YYYYMMDD): "))
fecha_fin = convertir_fecha(input("Por favor, ingresa la fecha de fin (formato YYYYMMDD): "))
```

- **Resumen de datos:** Permite al usuario obtener un resumen de los datos almacenados, incluyendo el nombre de la acción, la fecha inicial y la fecha final de los datos disponibles.

```
# Hacer la consulta SQL para obtener el resumen
c.execute("""
    SELECT name, MIN(date), MAX(date)
    FROM stocks
    GROUP BY name
""")
# Imprimir el resumen
resumen = pd.DataFrame(c.fetchall(), columns=['Ticker', 'Fecha inicial', 'Fecha final'])
print(resumen)
```

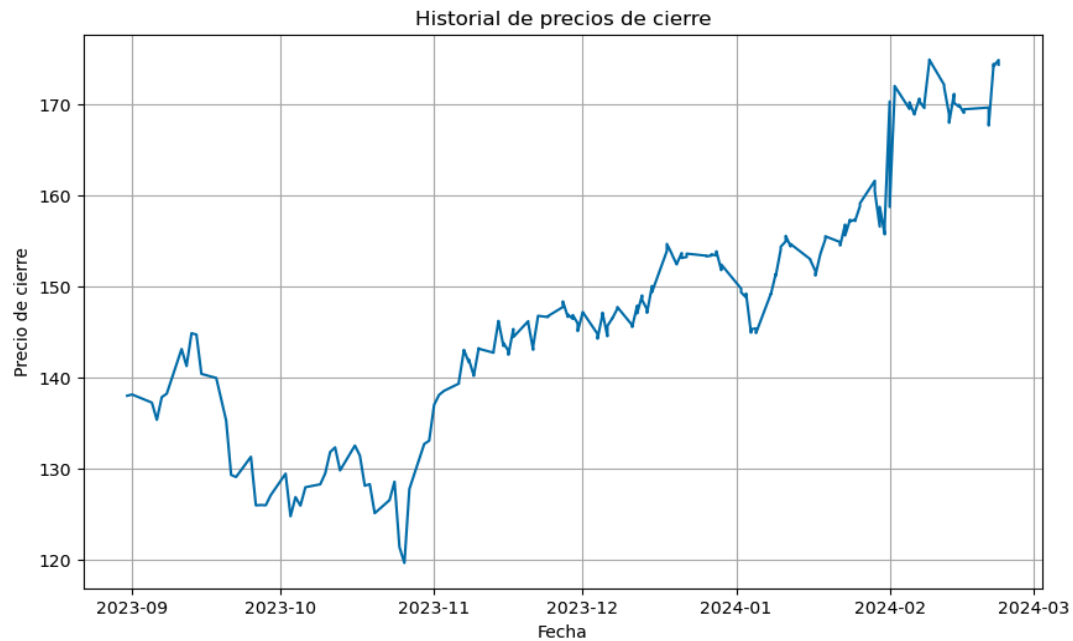
```

  Ticker Fecha inicial Fecha final
0    AAL    2022-01-10  2022-07-06
```

- **Interfaz de usuario:** Utiliza mensajes de texto para interactuar con el usuario, solicitando información como el tipo de consulta, el ticker de la acción, las fechas de inicio y fin, etc.

¿Quiere hacer una consulta, ingresar datos o salir? (consulta/datos/resumen/salir):

- **Gráficos:** Permite al usuario visualizar gráficos de los datos históricos de las acciones, incluyendo la posibilidad de comparar los datos de dos acciones diferentes en un mismo gráfico.



## 2. Diseño:

- **Base de datos SQLite:** Utiliza una base de datos SQLite para almacenar los datos de las acciones. La tabla `stocks` tiene tres columnas: `name` para el nombre de la acción, `date` para la fecha y `close` para el precio de cierre.

```
# Crear la tabla si no existe
c.execute('''
    CREATE TABLE IF NOT EXISTS stocks
    (name text, date text, close real)
''')
```

- **API de Financial Modeling Prep:** Utiliza la API de Financial Modeling Prep para obtener datos históricos de las acciones en formato JSON.

```
# Hacer la solicitud a la API
url = f"https://financialmodelingprep.com/api/v3/historical-chart/1min/{ticker}?from={hoy}&to={hoy}&apikey=DdNI6qCLlZsXjjmgCDR6qEohCeAzqmiW"
response = requests.get(url)
data = response.json()
```

- **Bucle principal:** Utiliza un bucle `while True` para controlar la ejecución del programa, permitiendo al usuario realizar múltiples acciones antes de salir del programa.

```
while True:
    # Preguntar al usuario si quiere hacer una consulta
    accion = input("¿Quieres hacer una consulta, o salir? ")
    if accion.lower() == 'salir':
        break
    elif accion.lower() == 'consulta':
```

- **Funciones auxiliares:**

- **convertir\_fecha:** Convierte fechas en formato 'YYYYMMDD' al formato 'YYYY-MM-DD' necesario para las consultas a la base de datos. Esto permite una mejor experiencia al usuario.
- **Interacción con el usuario:** Utiliza mensajes de texto para solicitar información al usuario y mostrar resultados o mensajes de error.

```
else:
    print("Opción inválida. Por favor, ingresa 'ultimo' o 'historial'.")
```

- **Manipulación de datos:** Utiliza la biblioteca Pandas para manipular y analizar los datos obtenidos de la API y de la base de datos.
- **Gráficos:** Utiliza la biblioteca Matplotlib para generar gráficos de los datos históricos de las acciones.

```
grafico = input("¿Quiere visualizar el gráfico? (s/n): ")
if grafico.lower() == 's':
    # Convertir la columna 'date' a formato de fecha
    df['date'] = pd.to_datetime(df['date'], format="%Y-%m-%d")
    # Crear el gráfico
    plt.figure(figsize=(10, 6))
    plt.plot(df['date'], df['close'])
    plt.title('Historial de precios de cierre')
    plt.xlabel('Fecha')
    plt.ylabel('Precio de cierre')
    plt.grid(True)
    plt.show()
```

### 3. Consideraciones de Diseño:

- **Modularidad:** El programa está estructurado en funciones y bloques de código separados para facilitar la comprensión y mantenimiento del código.
- **Manejo de errores:** Se incluyen validaciones para manejar errores de entrada del usuario, como fechas inválidas o acciones incorrectas.
- **Optimización de consultas:** Se utilizan consultas SQL optimizadas para acceder y manipular los datos de la base de datos de manera eficiente.
- **Presentación de datos:** Se presenta la información de manera clara y legible para el usuario, incluyendo gráficos para visualizar los datos históricos de las acciones.
- **Seguridad:** Se utiliza una base de datos local SQLite y no se realizan operaciones que puedan comprometer la seguridad de los datos del usuario.

En resumen, el programa proporciona una interfaz interactiva y amigable para que los usuarios puedan consultar, ingresar y analizar datos históricos de acciones de manera eficiente y segura.

¿Quiere hacer una consulta, ingresar datos o salir? (consulta/datos/resumen/salir): datos  
 Por favor, ingresa el valor del ticker: AAPL  
 Por favor, ingresa la fecha de inicio (formato YYYYMMDD): 20210101  
 Por favor, ingresa la fecha de fin (formato YYYYMMDD): 20220101  
 Se guardaron los datos de la acción AAPL desde 2021-01-01 hasta 2022-01-01.  
 La base de datos se ha actualizado correctamente.  
 ¿Desea volver al menú de inicio? (si/no): si  
 ¿Quiere hacer una consulta, ingresar datos o salir? (consulta/datos/resumen/salir): datos  
 Por favor, ingresa el valor del ticker: AAL  
 Por favor, ingresa la fecha de inicio (formato YYYYMMDD): 20210101  
 Por favor, ingresa la fecha de fin (formato YYYYMMDD): 20220101  
 Se guardaron los datos de la acción AAL desde 2021-01-01 hasta 2022-01-01.  
 La base de datos se ha actualizado correctamente.  
 ¿Desea volver al menú de inicio? (si/no): si  
 ¿Quiere hacer una consulta, ingresar datos o salir? (consulta/datos/resumen/salir): resumen  
 Ticker Fecha inicial Fecha final  
 0 AAL 2021-07-06 2022-07-06  
 1 AAPL 2021-07-06 2021-12-31  
 ¿Desea volver al menú de inicio? (si/no): si  
 ¿Quiere hacer una consulta, ingresar datos o salir? (consulta/datos/resumen/salir): consulta  
 ¿Quiere consultar el último valor de una acción o revisar el historial de una acción? (ultimo/historial): historial  
 Por favor, ingresa el valor del ticker: AAL  
 Por favor, ingresa la fecha de inicio (formato YYYYMMDD): 20210101  
 Por favor, ingresa la fecha de fin (formato YYYYMMDD): 20220101

	name	date	close
0	AAL	2021-07-06	21.000
1	AAL	2021-07-07	20.310
2	AAL	2021-07-08	20.360
3	AAL	2021-07-09	20.880
4	AAL	2021-07-12	20.850

```

..    ...    ...    ...
159  AAL  2021-12-29  18.050
160  AAL  2021-12-30  18.080
161  AAL  2021-12-30  18.220
162  AAL  2021-12-31  17.950
163  AAL  2021-12-31  18.065

```

[164 rows x 3 columns]

¿Quiere visualizar el gráfico? (s/n): s

