

# Spot STOCK

Actividad 6: API Project



Alejandro González

# Índice

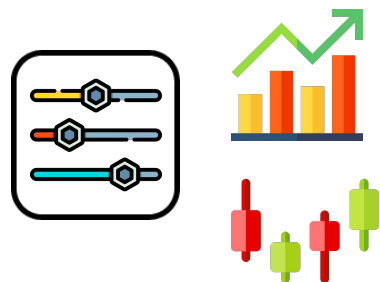
1. Objetivos
2. Estructura de la aplicación
3. Descripción de API
4. Aplicación
5. Ejemplo práctico

Anexo: (Project GitHub) [https://github.com/alexgg278/spot\\_stock](https://github.com/alexgg278/spot_stock)

Mas documentacion de la App en Readme de GitHub

# Objetivos

1. Utilizar un API financiera para obtener datos históricos de acciones dinámicamente
2. Realizar las consultas de manera interactiva y dinámica por parte del usuario
  - a. Consultar evolución de acciones de diferentes empresas sin necesidad de reprogramación (únicamente introduciendo el ticker/nombre de la empresa).
3. Construir una interfaz gráfica que permita al usuario visualizar y manipular distintos gráficos financieros de diferentes acciones



# Estructura de la APP (I)

- La carpeta “**app**” contiene el código principal de la aplicación.
- La carpeta “**other**” contiene un notebook con un ejemplo de llamada a la API y una aplicación para visualizar la respuesta de la API con matplotlib.

alexgg278 Adjust graph size to full screen	
..	
folder .idea	Adjust graph size to full screen
folder app	Adjust graph size to full screen
folder other	Adjust graph size to full screen
file README.md	Initial commit

- Inside the “**app**” folder we can find 5 different elements:
  - **assets**: The folder contains the .css format file for the web interface.
  - **app.py**: Contiene la 1º versión de la app
  - **app2.py**: Contiene la 2º versión de de la app
  - **functions.py**: Contiene las funciones python utilizadas en “**app.py**” y en “**app2.py**”.
  - **tickers.csv**: Es un dataset de dos columnas utilizado para asociar el nombre de la empresa al ticker en bolsa de la compañía.
  - Ej: GOOGL ➡ Alphabet Inc.

alexgg278 Adjust graph size to full screen	
..	
folder .ipynb_checkpoints	First MVP
folder __pycache__	Adjust graph size to full screen
folder assets	Second MVP
file app.py	Adjust graph size to full screen
file app2.py	Adjust graph size to full screen
file functions.py	Adjust graph size to full screen
file tickers.csv	Adjust graph size to full screen

# Estructura de la APP (II)

- **Requisitos para ejecutar:** Crear un python environment con las siguientes librerías.
  - *pandas, plotlib, dash, matplotlib, requests y csv*
- **Para ejecutar:**
  - Ejecutar *app.py* (versión con consulta por ticker) o *app2.py* (versión con consulta por nombre de empresa).
- **Se entregara:**
  - Un .zip con los archivos. Adicionalmente en el GitHub del proyecto se pueden encontrar todos los archivos y su último versión (es probable que continúe mejorando la aplicación).

# Descripción de la API

- La API utilizada es: [Alpha Vantage](#) (link)
  - Esta API permite obtener datos históricos del valor de acciones con frecuencia, diaria, semanal, etc.
  - Permite un máximo de 500 llamadas diarias.
- Lógica de la llamada
  - La url es: <https://www.alphavantage.co/query>
  - A la url se añade la frec. De datos (en nuestro caso diaria): *TIME\_SERIES\_DAILY*
  - Para recibir una respuesta completa añadimos: *full*
  - Por último el ticker de la empresa: *AMZN* (para obtener datos de amazon)
    - El ticker es el identificador de una empresa en la bolsa
- La idea es que el usuario a través de la interfaz introduzca el ticker o nombre de la empresa y la query a la API se construya de manera automática y así mostrar la gráfica sin tener que reprogramar la query en cada consulta.

# App (I)

- La App estaba basada en dash
  - Dash es una librería de python basada en plotly que permite visualizar datos de manera interactiva mediante diferentes tipos de gráficas en un formato web.
- La diferencia entre la app.py y la app2.py es la manera en la que el usuario introduce el ticker de la empresa a visualizar.

## app.py

El usuario introduce el ticker de la empresa en un text box. El ticker se añade a la query en la ejecución



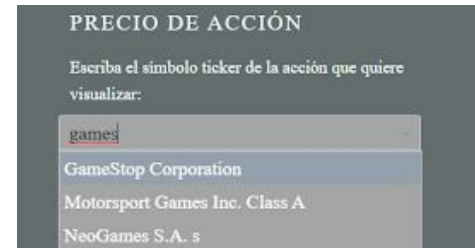
PRECIO DE ACCIÓN

Escriba el símbolo ticker de la acción que quiere visualizar:

GME

## app2.py

El usuario escribe el nombre de la empresa y lo selecciona de un desplegable. El programa asocia el nombre al ticker y realiza la query.



PRECIO DE ACCIÓN

Escriba el símbolo ticker de la acción que quiere visualizar:

games

GameStop Corporation

Motorsport Games Inc. Class A

NeoGames S.A. s

# App (II)

- **Tipo de grafico**

- Se han programado dos maneras de visualizar los datos: Time-series o Candlestick graph
- El modo de visualización puede ser elegido por el usuario mediante un selector

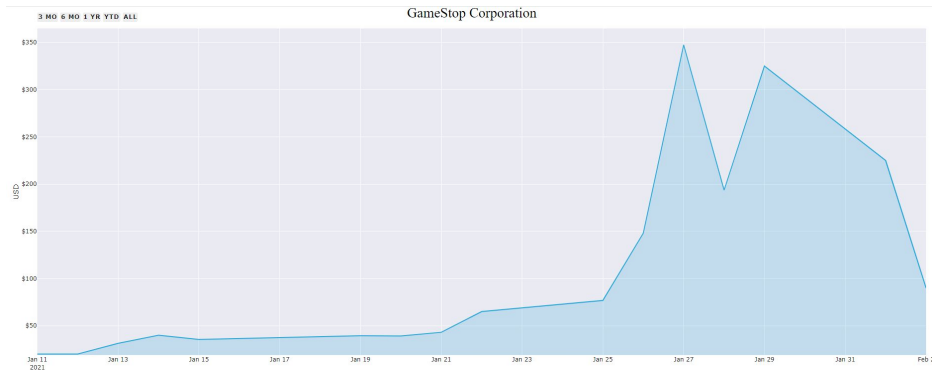
Escoge el tipo de gráfico:

☒ Time series

☐ Candlestick

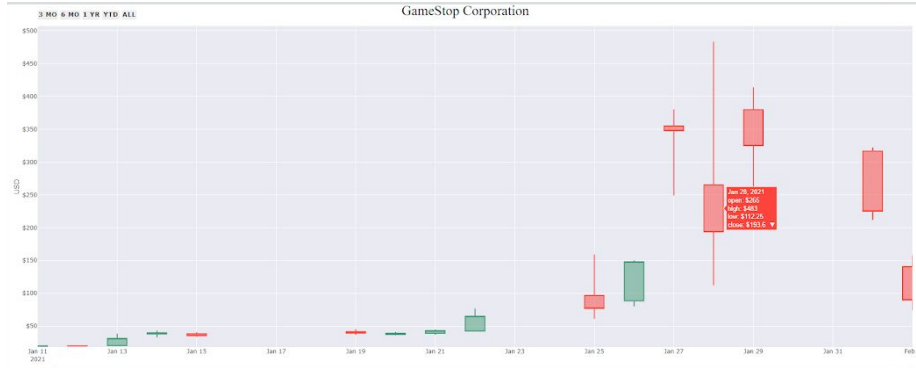
## Time-series

Muestra la tendencia temporal del valor de la acción



## Candlestick

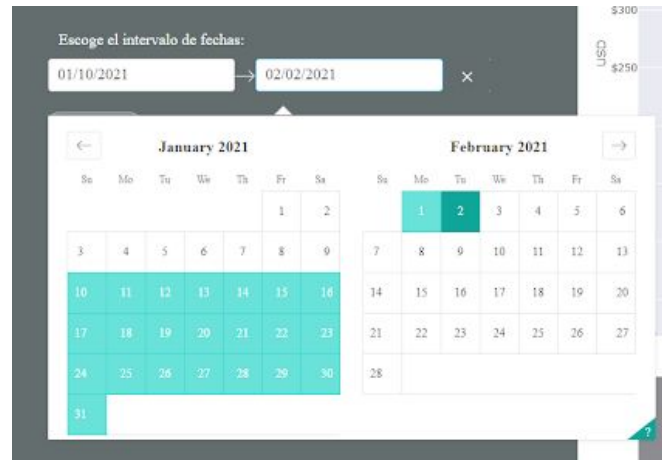
Muestra los valores de apertura, cierre, alto y bajo (diarios)



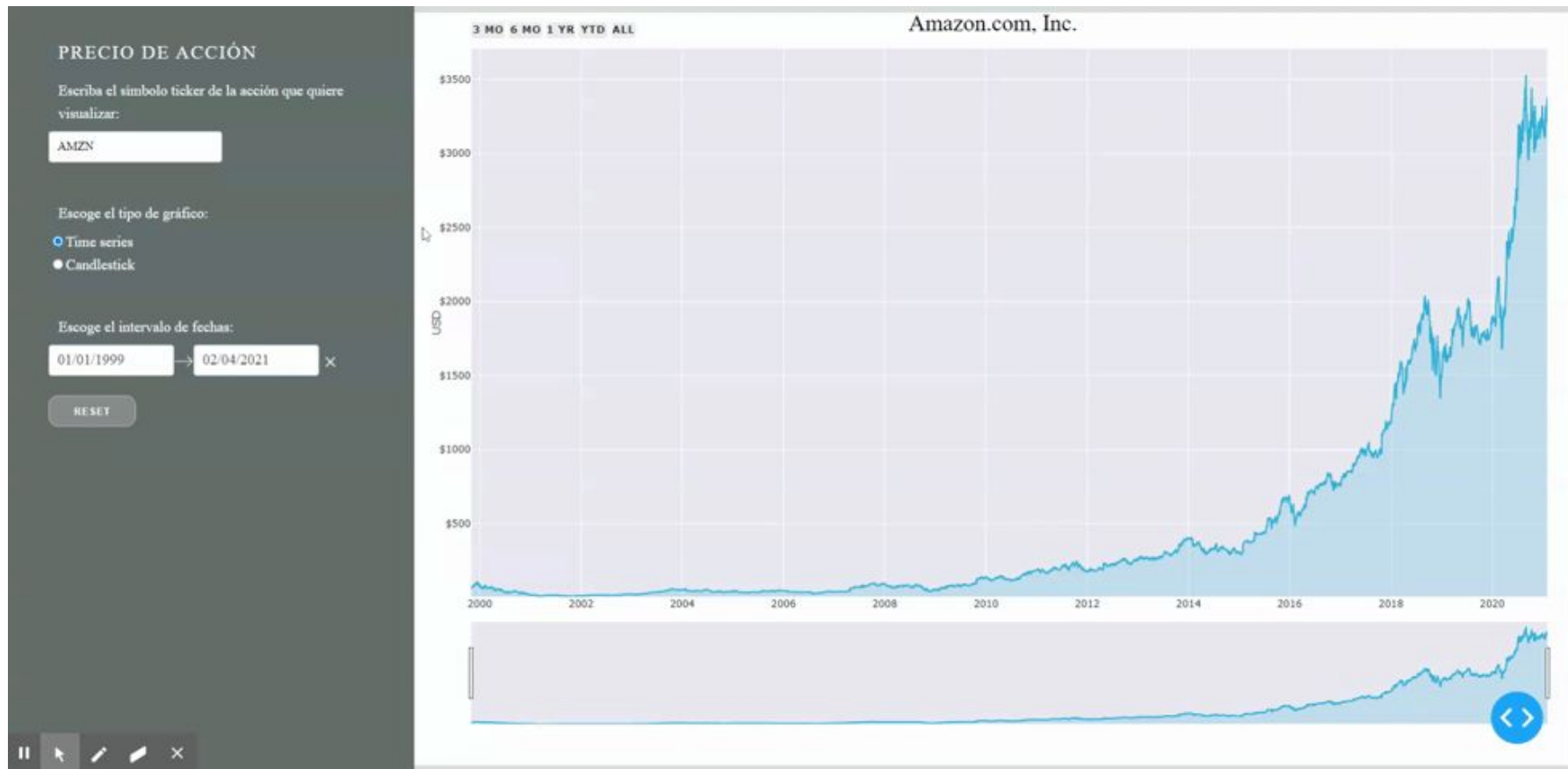


# App (III)

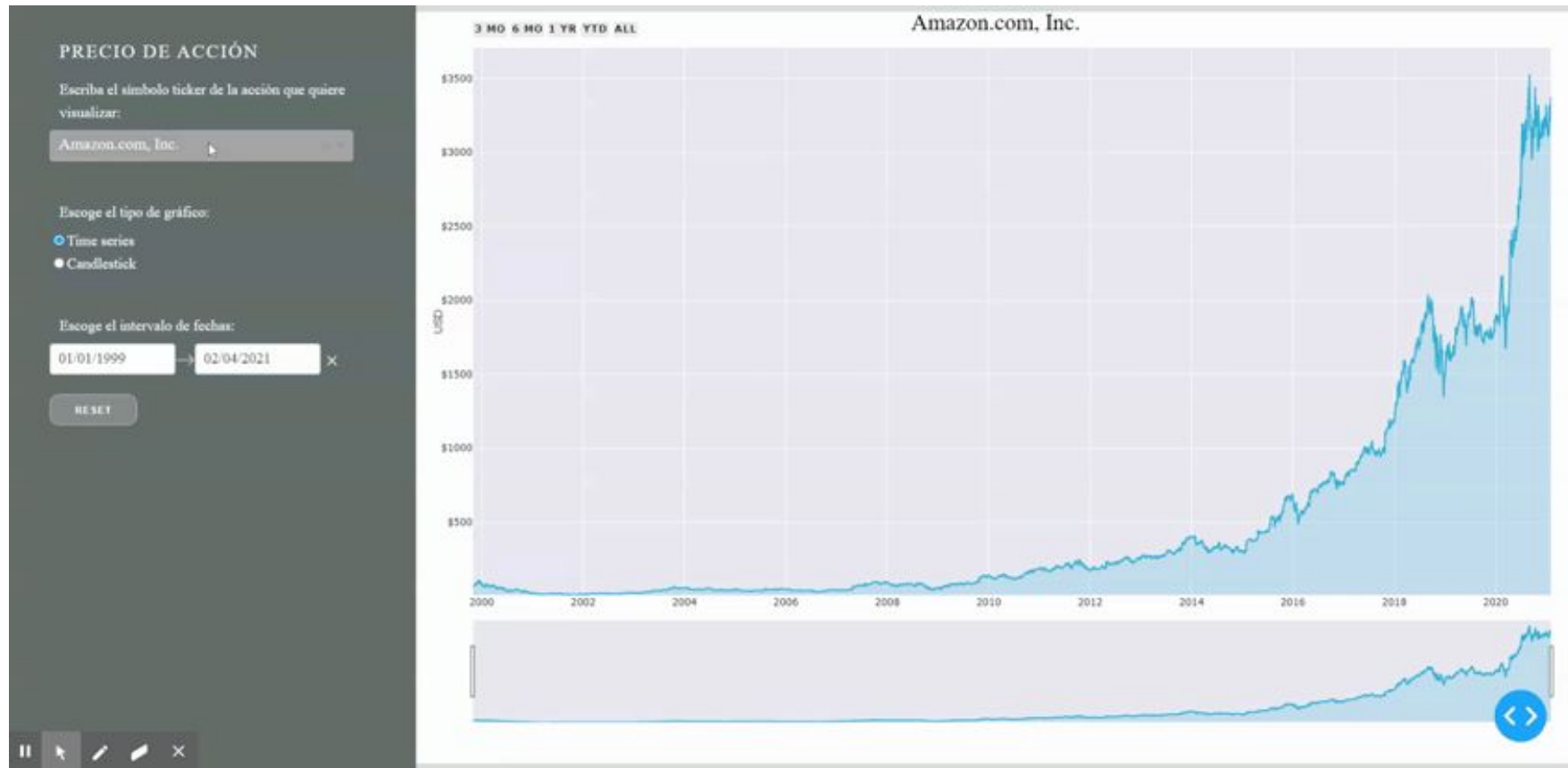
- Selector de fechas
  - Podemos seleccionar un intervalo de fechas sobre el que visualizar acciones.
  - Podemos seleccionar las fechas en el calendario desplegable o escribirlas.
  - La app filtra los datos para solo mostrar los incluidos en el intervalo temporal.
  - Se puede pulsar el botón de RESEt para volver al intervalo inicial.



# Ejemplo práctico app



# Ejemplo práctico app2



# GitHub

[https://github.com/alexgg278/spot\\_stock](https://github.com/alexgg278/spot_stock)

