

Extracción de datos desde una API

API es el acrónimo en inglés de application programming interface. La API es básicamente una interfaz que habla por nosotros con un programa. Cuando un programador decide poner a disposición pública algún conjunto de datos generados por sus programas, lo puede hacer a través de una interfaz de este tipo. Otros programadores extraen estos datos de la aplicación construyendo URLs o a través de clientes HTTP.

Una API está caracterizada por tres elementos:

- el **usuario**, que es la persona realizando la petición;
- el **cliente**, que es el ordenador que envía la petición al servidor;
- el **servidor**, que es el ordenador que responde a la petición.

Los datos se almacenan en un servidor y la documentación correspondiente se pone a disposición de los programadores para que puedan acceder.

Un usuario externo puede realizar peticiones de datos a este servidor. Ejemplos de estos sistemas son las redes sociales comunes como Twitter, Facebook, Google Direction API, YouTube, LinkedIn, etc. Incluso portales como la NASA o el New York Times pone a disposición sus datos a disposición para que los desarrolladores puedan acceder a ellos y manipularlos.

¿A qué tipos de datos podemos acceder? Por ejemplo, con Facebook podríamos obtener la lista de amigos entre un amigo nuestro y nosotros mismos, o el número de likes obtenidos en ciertas páginas. A Twitter le podríamos pedir la lista de temas y argumentos más tratados en una cierta área geográfica, por ejemplo en España. A través de la Google Direction API podemos obtener rutas para llegar de una dirección a otra. Por último, desde la Google Book API, podremos obtener la lista de libros de un autor, etc.

La mayoría de aplicaciones mencionadas requieren gestionar una autorización que se tiene que pedir, concretando en una contraseña o clave de la API. Las API públicas también tienen limitaciones en cuanto a cantidad de datos disponibles y número de peticiones por segundo posibles.

Los datos que obtenemos se representan a través de un formato específico conocido como JSON (Javascript Object Notation). JSON es un formato estándar que se usa para transmitir datos entre un servidor y una aplicación web como alternativa al XML.

Enunciado

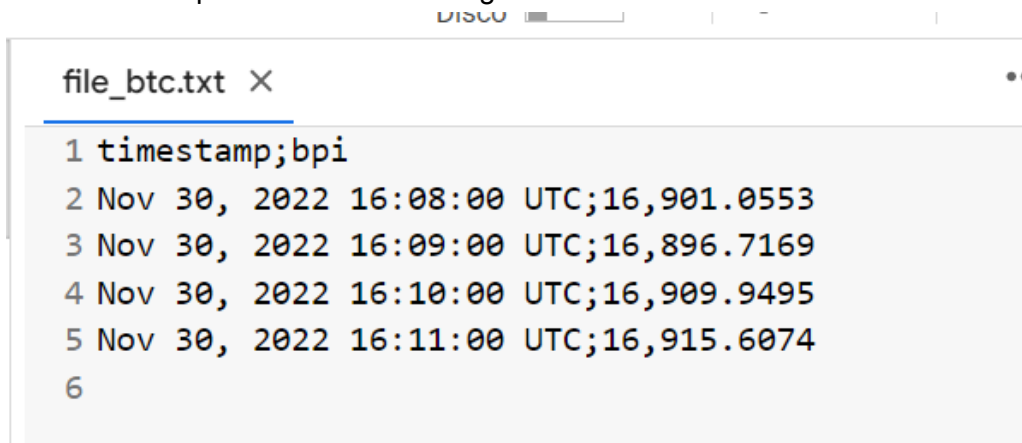
Vamos a utilizar ahora una API de CoinDesk que ofrece información sobre Bitcoin (BTC), una moneda digital descentralizada utilizada en todo el mundo. Un servicio que ofrece CoinDesk es el índice de precios de Bitcoin (BPI). Estos datos de precios de Bitcoin se calculan cada minuto y se publican en USD, EUR y GBP. Los datos de BPI son accesibles a través de una API REST.

URL de la API: **<https://api.coindesk.com/v1/bpi/currentprice.json>**

Dificultad Media

- 1) Las APIs normalmente las usan otros programas en lugar del usuario final directamente (tú, en este caso). Te proponemos que desarrolles una pequeña aplicación que haga una petición por minuto a la API y guarde el valor del BTC y timestamp de cada petición en un fichero de texto (puedes seleccionar una de las tres monedas, es indiferente).
 - a) Crea funciones y/o clases en jupyter
 - b) Crea el script en Python y ejecutas en Google Colab
 - c) Guarda el fichero en la misma carpeta

El resultado esperado es similar al siguiente:



```
1 timestamp;bpi
2 Nov 30, 2022 16:08:00 UTC;16,901.0553
3 Nov 30, 2022 16:09:00 UTC;16,896.7169
4 Nov 30, 2022 16:10:00 UTC;16,909.9495
5 Nov 30, 2022 16:11:00 UTC;16,915.6074
6
```

Dificultad Alta

- d) Crea una tabla en SQLite en tu propia máquina para guardar los datos en una tabla *BTCdiario* y creando columnas con su tipo de dato y almacenando los datos guardados.

Ejemplo de tabla generada en pandas:

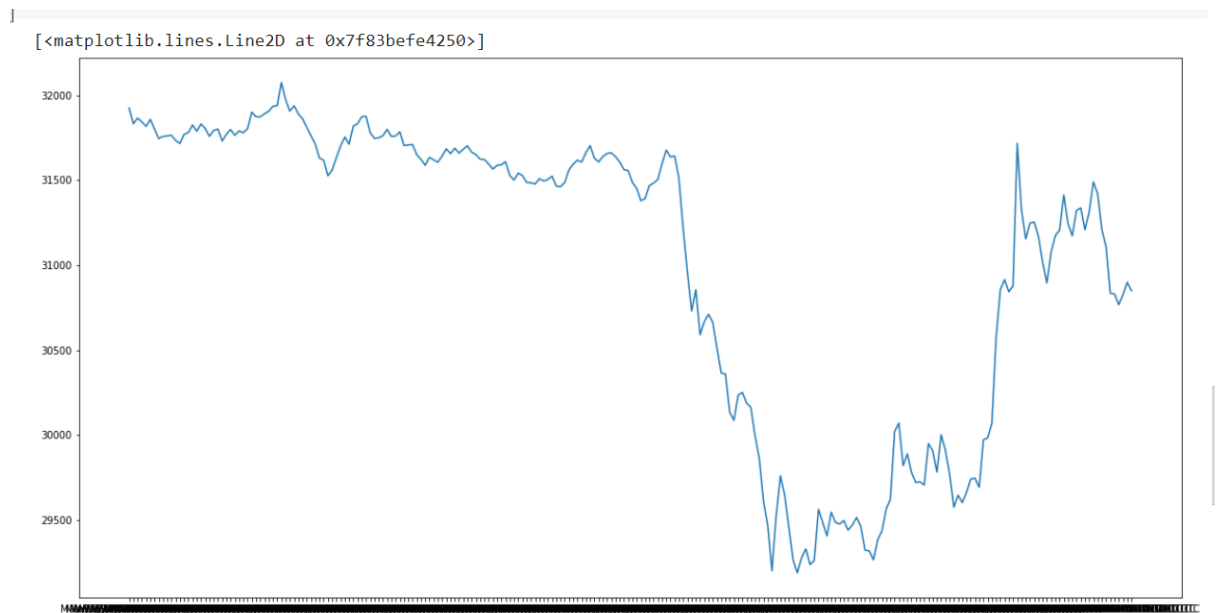
bpi	
timestamp	
May 11, 2022 10:19:00 UTC	31925.3154
May 11, 2022 10:20:00 UTC	31832.6575
May 11, 2022 10:21:00 UTC	31865.4802
May 11, 2022 10:22:00 UTC	31844.0942
May 11, 2022 10:23:00 UTC	31817.1770

- e) Realiza una prueba de conexión con SQL y muestra que se han cargado los datos en otra celda (si trabajas en local estará bien en un notebook diferente, en Google colab en el mismo).

- f) 1h límite - vía Google Colab de recogida datos

SE PREMIARÁ POR ALMACENAMIENTO DE MÁS DATOS

- 2) Finalmente procederemos a graficar **precio** vs. **tiempo** en la moneda que prefieras. Comenta el diseño de tu solución adjuntando un diagrama de la misma.



3) **Dificultad Alta:**

Para obtener más valores guardados, y no tener que esperar cada minuto, te retamos a que vaya creando un estimador de valores casero. Podemos guardar los primeros valores n veces, y realizar una pequeña ingesta de i.e. 10000 valores aleatorios, aplicando un criterio entre un mínimo y máximo de los últimos 90 días reales. (en Machine Learning utilizaremos serie temporales y regresión para comprobar si hemos sido capaces de realizar lo mismo).

El resultado tendrá que ser comentado y justificado en cada paso. Es preferible realizar el todo en Jupyter y finalmente en ficheros python ejecutables con funciones y clases.

Se hará uso de las funciones built-in de Python, Pandas, Numpy y librerías de visualización de Python.