Prueba técnica t2ó

La tarea consiste en hacer una aplicación que lee datos de una API externa, los almacena y expone información sobre estos datos a través de una API REST.

En este caso vamos a leer los datos de la API de Blockchain.com cuya documentación está disponible aquí: <https://api.blockchain.com/v3/>

Concretamente leeremos los datos de la llamada GET /l3/{symbol}.

# Requisitos de la tarea

Ten en cuenta estos requisitos al hacer la tarea:

* Debe implementar todas las funcionalidades listadas a continuación.
* Debe estar hecha en Python.
* Se valorará positivamente el uso de librerías de manejo de datos como Pandas y/o NumPy.
* Se valorará positivamente el uso de tests.
* No es necesaria ninguna interfaz de usuario.
* Todas las llamadas REST devolverán los datos en formato JSON.

# Consejos antes de empezar

* No tienes porqué usar más de 5 ó 6 horas en la tarea, después de todo es tu tiempo libre.
* No te estreses si cumplir todos los requisitos te llevaría demasiado tiempo, nos interesa más un buen diseño y calidad del trabajo que una prueba completa.
* Puede que usar Pandas para la aplicación no parezca tener mucho sentido, recuerda que es una demostración de tus habilidades y conocimiento.
* En caso de que por falta de tiempo no te veas capaz de incluir alguna funcionalidad, mejora, o cambio de diseño que te habría gustado, cuéntanoslo en el README.
* Que te diviertas y te sea enriquecedor :)

# Funcionalidades

### 1. Carga de datos

Un usuario con la aplicación instalada debe poder ejecutar un comando indicando un símbolo compuesto de una criptomoneda y una moneda real (e.g. "BTC-USD") y el sistema debe cargar las órdenes L3 en su base de datos.

### 2. Estadísticas de compras

Como usuario con la aplicación ejecutada, debo poder visitar una URL y obtener las siguientes estadísticas de las órdenes de compra (bids) de un símbolo:

* El valor medio de las órdenes, donde el valor es la cantidad de la orden multiplicado por su precio.
* La orden de compra con mayor valor.
* La orden de compra con menor valor.
* El total de monedas en órdenes.
* El precio total de las órdenes.

Un ejemplo de la salida:

{

"bids": {

"average\_value": 10.25,

"greater\_value": {

"px": 31790.42,

"qty": 1.25824063,

"num": 16484191171,

"value": 40000.00

},

"lesser\_value": {

"px": 31777,

"qty": 0.05,

"num": 16484191502,

"value": 1588.85

},

"total\_qty": 40.12,

"total\_px": 154600

}

}

### 3. Estadísticas de ventas

Como usuario con la aplicación ejecutada, debo poder visitar una URL y obtener las mismas estadísticas que antes pero respecto a las órdenes de venta (asks) de un símbolo.

### 4. Estadísticas generales

Como usuario con la aplicación ejecutada, debo poder visitar una URL y obtener las siguientes estadísticas globales de todos los símbolos:

* Número de órdenes de compra.
* Número de órdenes de venta.
* Valor total de las órdenes de compra.
* Valor total de las órdenes de venta.
* El total de monedas de las órdenes de compra.
* El total de monedas de las órdenes de venta.

Un ejemplo de la salida:

{

"BTC-USD": {

"bids": {

"count": 500,

"qty": 45,

"value": 150000.00

},

"asks": {

"count": 500,

"qty": 500,

"value": 450000

}

}

}

# Presentación

El entregable de la prueba deberá contener:

* un archivo README indicando cómo usar el programa.
* un directorio con el proyecto.