

1. Flujo:

- a-Nuestra API recibirá un GET a un endpoint /items/prices/[item-id].
- b-La api llamará al servicio de moneda

```
curl -X GET -H 'Authorization: Bearer $ACCESS_TOKEN'
```

https://api.mercadolibre.com/currency_conversions/search?from=ARS&to=CLP

Con new Relic podemos saber el promedio en función a estadísticas el tema de las llamadas

Tomando en cuenta los re intentos a dicha api, se deberán hacer los cálculos correspondientes y devolver el item con su precio final.

Si llegan demasiadas request habría que limitar desde nuestra api la cantidad de requests que puede tomar y devolver el código correspondiente.

Si las cotizaciones cambian una vez al día se podría hacer un proceso que corra a determinada hora para que tome todas las cotizaciones posibles y las guarde en memoria o en una DB propia (dependiendo de la cantidad de información a guardar. Considero que no debería ser mucha y se podría manejar en memoria con redis por ejemplo). Se puede usar Cache.

Necesitamos algo que NO sea eventualmente consistente.
un KVS del tipo CACHE.

2. Del proy de https://github.com/mercadolibre/fury_bootcamp-demo-java-app

Podríamos tener pre cacheado la cotización, la cual por la anterior parte se asume que cambia una vez al día. Una alternativa sería guardarlo en un archivo json, o en mongodb. Genero una clase muy simple que solo contenga cotiz y su valor (un KVS).

Supongo que la implementación del Cache no es que corra un proceso, sino que la primera vez al obtener la información, si no la tiene la consulta. En nuestro caso deberíamos combinarlo con un Change Data Capture.

El error se podría dar si cuando tomamos la cotización haya una variación posterior antes de las 24 hrs.

Usaría el servicio que ofrece MELI para la obtención de las cotizaciones.

- Cómo harían para evitar ir “en vivo” a la API de items?

asegurándonos que el SCOPE está en local, o en no productivo. Esto se puede configurar desde la clase ScopeUtils.

- Qué pasa si cambia el precio de un item?

Asumiendo que esto se refiere a que esta info está pre cacheada, nuestros datos quedarían desactualizados.

- Cómo puedo enterarme inmediatamente cuando cambia el precio de un item? Qué harían al respecto?

Aplicando el patrón publish-subscribe. Asumiendo que la API posee un tópic, suscribirme al mismo para cuando el item cambie de precio.

- Cómo ejecuto el código que tengo en fury?

Asumiendo que se refiere a ejecutarlo de forma local:

```
fury get bootcamp-demo-java-app
```

```
fury run
```