Programación Backend

Consumo de APIs - Uso de Servicios  
// Práctica

Objetivo

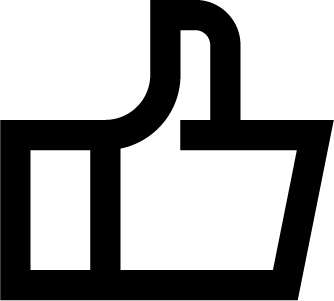
El objetivo de esta práctica es ejercitar en equipo cómo **consumir información de un microservicio** dado. El aprendizaje obtenido en esta instancia va a ser retomado luego en la clase 6 para adquirir nuevos conceptos.

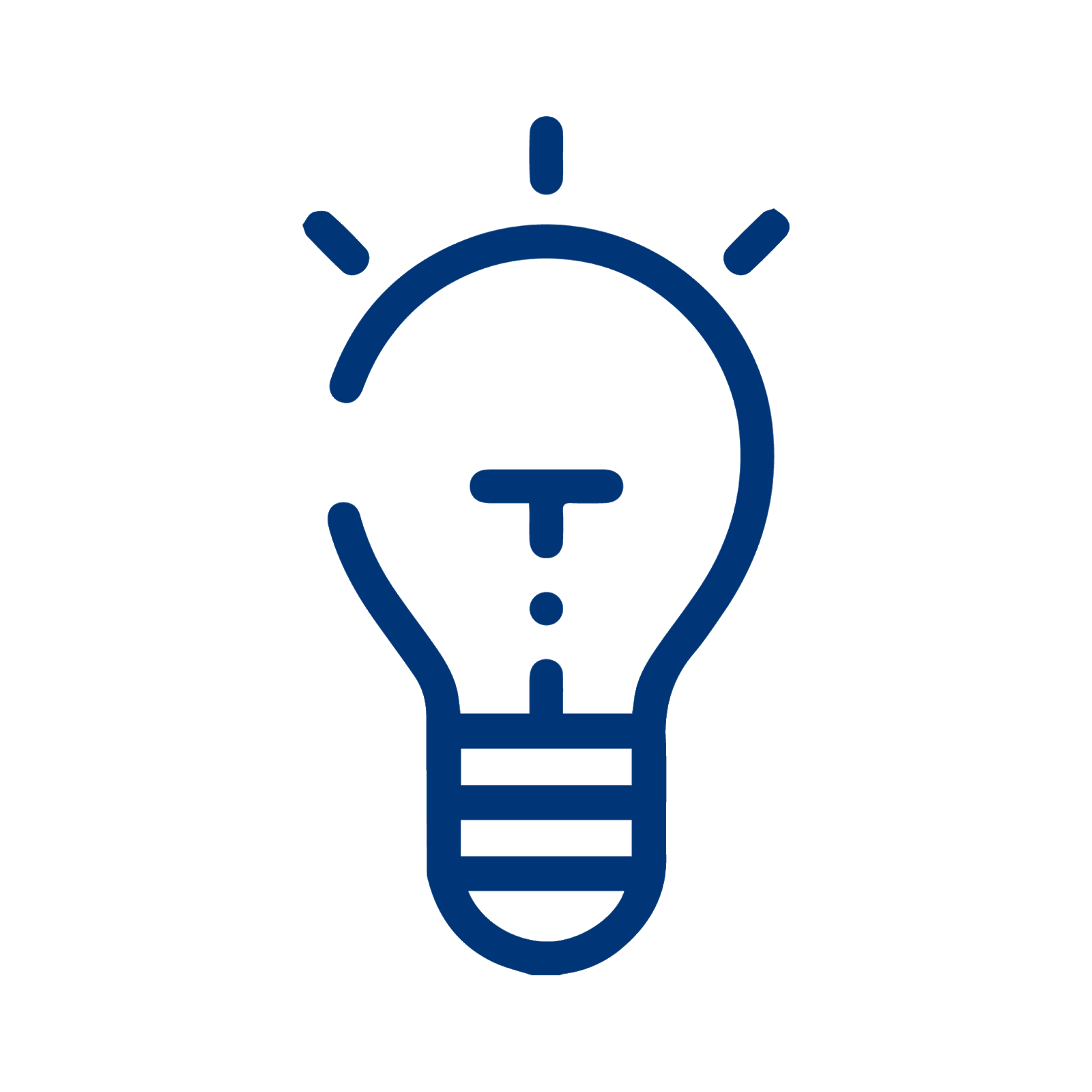
**Parte #1 Desafío:** Desarrollar una API que pueda obtener el precio de un ítem y convertirlo a precio en dólares.

**Conceptos a aplicar**

Te recomendamos que repases estas clases antes de comenzar y que tengas a mano el material para cualquier consulta durante el desarrollo del ejercicio:

| **Conceptos** |
| --- |
| Microservicios y Sistemas Distribuidos |
| Creación de aplicación en Fury |
| Scopes y estrategias de Deploy |

**¡Buena suerte!** ****



La API de ítems es una de las APIs más importantes del marketplace de MercadoLibre. Cada publicación que hace un vendedor de MeLi se obtiene desde esta API utilizando endpoints de la siguiente manera:

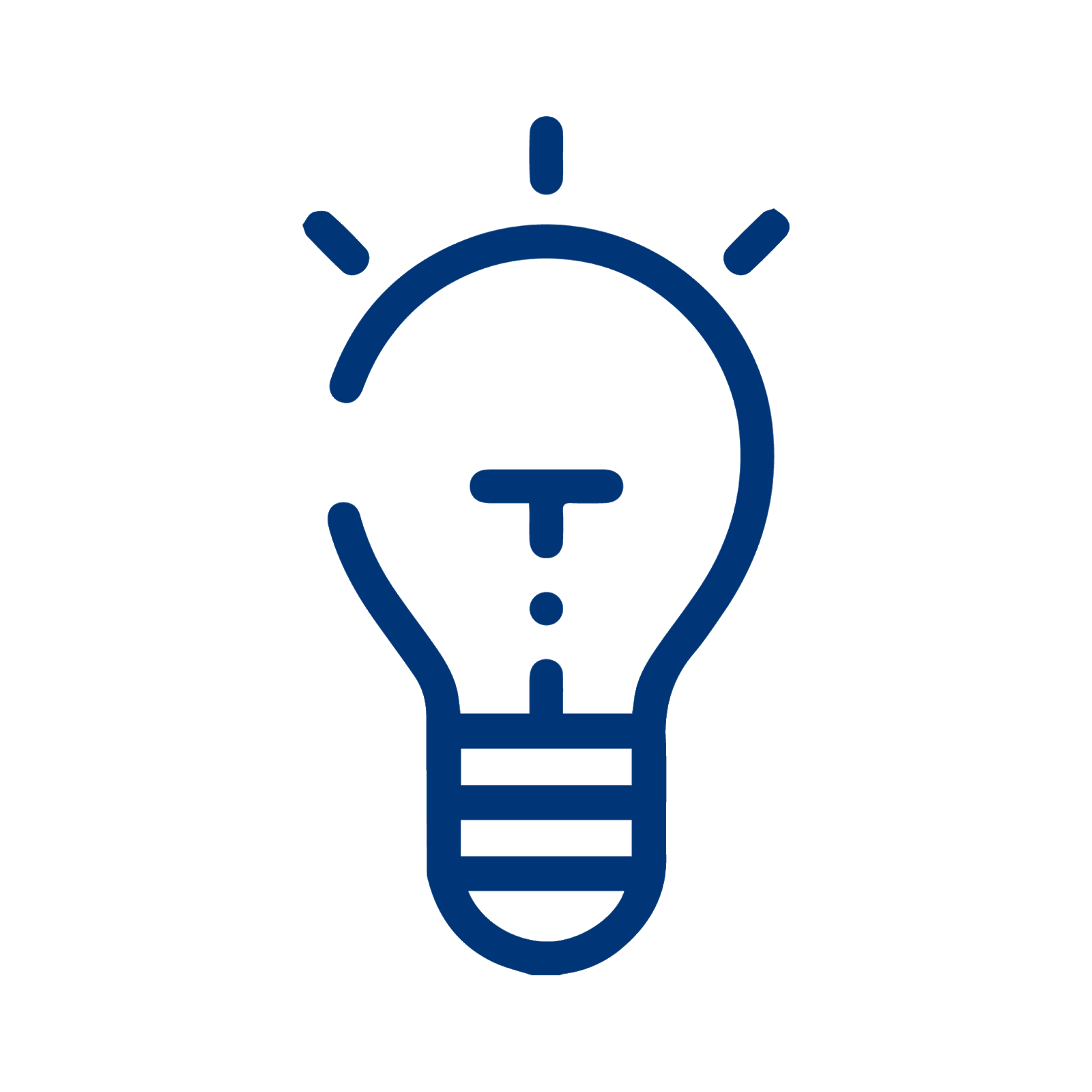
curl <https://api.mercadolibre.com/items/><item\_id>

Sabemos a su vez que cada publicación tiene un **precio en una moneda específica**, y existe una API que nos resuelve la conversión entre monedas ([link](https://developers.mercadolibre.com.uy/es_ar/ubicacion-y-monedas) a la documentación pública).

Problema

Necesitamos entonces implementar una API que reciba un Item ID y que con ese Item ID, vaya a buscar el precio en moneda original para obtener la conversión con la moneda de origen (usando la api de conversión) y por último, devuelva el precio en dólares para ese ítem.

Partiremos de un código de ejemplo (puedes descargar la base desde Playground) que implementa un acceso a la API de ítems, dado un item\_id, y devuelve un precio.



¿Sabías que dada una publicación de Mercado Libre cualquiera se puede obtener el id del item publicado? ¡Te cuento cómo!

Si te fijás una URL de MeLi es de la forma: <https://articulo.mercadolibre.com.uy/MLU-477476349-mecha-para-encendedor-zippo-_JM#position=2&search_layout=stack&type=item&tracking_id=9ee62d18-a318-47f8-981f-61db5e06e9a3>

La forma de obtener el id del item es tomar la parte que comienza con ML, hasta el último número, sin el guión. En el ejemplo anterior sería **MLU477476349** que sale de:

[https://articulo.mercadolibre.com.uy/**MLU-477476349**-mecha-para-encendedor-zippo-\_JM#position=2&search\_layout=stack&type=item&tracking\_id=9ee62d18-a318-47f8-981f-61db5e06e9a3](https://articulo.mercadolibre.com.uy/MLU-477476349-mecha-para-encendedor-zippo-_JM#position=2&search_layout=stack&type=item&tracking_id=9ee62d18-a318-47f8-981f-61db5e06e9a3)

Entonces, si recibo un request al endpoint /prices/MLU477765491, devolverá: {"price":1950.0,"item\_id":"MLU477765491"}, y lo que tenemos que hacer es que nuestra API haga el mismo trabajo, pero para el precio en dólares.

Nuestra API recibirá un **GET** a un endpoint /items/prices\_usd/[item-id] y deberá devolver el precio en dólares, por ejemplo: {"item\_id":"MLU477765491","price\_usd":44.5205475}

Tenemos entonces que responder las siguientes **preguntas**:

* ¿Cuáles son las entidades que modelan esta solución? **Diseñar un diagrama de clases**.
* ¿Cómo sería el flujo de ejecución desde que nos llega un request? **Diseñar un diagrama que muestre dicho flujo.**
* ¿Qué ocurriría si nos llegan 1000 requests por minuto? ¿Y si llegaran 100.000? ¿Esta API tiene una implementación correcta? ¿Qué preguntas harías para pensar en optimizaciones a hacerle a la API?
* Una vez deployada en producción, ¿cómo te vas a asegurar de enterarte si hay algún problema con tu aplicación? Por ejemplo: si hay un alto grado de errores.
* Asumiendo que las cotizaciones cambian una vez al día, ¿qué opción podemos tomar para mejorar esta API?

Nota:

*No tomes estas soluciones como un “template de arquitectura que resuelve todos los problemas”, dado que tiene algunos problemas conocidos. Sin embargo, a modo de ejercitación, creemos que es super enriquecedor.*