# Prueba práctica

## Parte 1: Java

Desarrollar dos microservicios usando Java 17 en donde se realice un consumo mediante el método POST:

* Un microservicio orquestador, que se encargará de recibir tu request, validar que el tipo de datos recibidos sean correctos y manejar excepciones.
* Un microservicio de dominio, que recibirá la información para almacenarla en una BD (la de tu preferencia).

La petición del cliente la recibe el microservicio orquestador, y despues de validarla la envía al microservicio de dominio para almacenarla en la BD.

### Consideraciones

* Definir el request a utilizar.
* Hacer el correcto uso de los códigos HTTP de respuesta (200, 400, 401 y 500), según sea el caso.
* Utilizar Spring Boot y hacer uso de sus herramientas como anotaciones.
* Hacer uso de Maven en el proyecto.
* Implementar pruebas unitarias (Lo pueden realizar de una sola clase, usando el plugin de jacoco).
* El consumo de su proyecto deberá realizarse desde Postman.
* En caso de error, el log nos debe dar el detalle adecuado para conocer la causa del problema.

### Entregables

* Código fuente de los microservicios
* Diagrama de flujo de los microservicios
* Archivo de Postman con los requests para probar el microservicio
* Reporte de resultados de las pruebas unitarias
* Evidencias (capturas de pantalla) de la ejecución de los microservicios y las pruebas realizadas con Postman

## Parte 2: Docker

Contenerizar los microservicios desarrollados en el punto anterior, bajo las siguientes consideraciones:

* Ocupar multistage build para compilar los microservicios y generar las imágenes Docker para desplegar las aplicacionesgithub
* Optimizar la JVM mediante el uso de parámetros al momento de ejecutar el jar dentro de la imagen
* **Puntos extra**: Utilizar herramientas como jlink y jdeps para la optimización de la imagen final

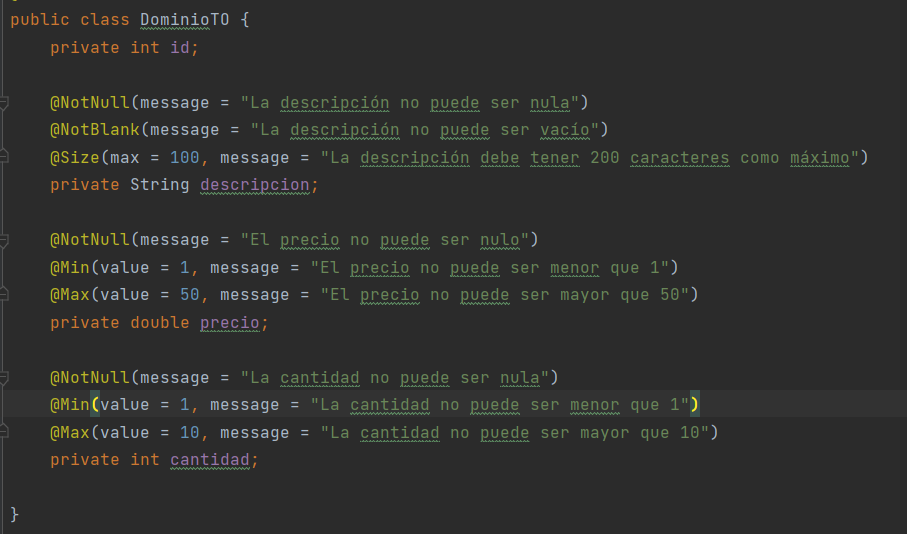
### Entregables

* Archivos Dockerfile generados
* Log del comando docker build (deshabilitar el uso de cache para generar un log completo en cada ejecución)
* Evidencias (capturas de pantalla) de la ejecución de los microservicios y las pruebas realizadas con Postman

**SOLUCIÓN**

**PARTE 1**

**Modelo a Validar**

****

**DIAGRAMA DE FLUJO DE LOS MICROSERVICIOS**

**CONFIG SERVICE**

**EUREKA SERVICE**

**GATEWAY SERVICE**

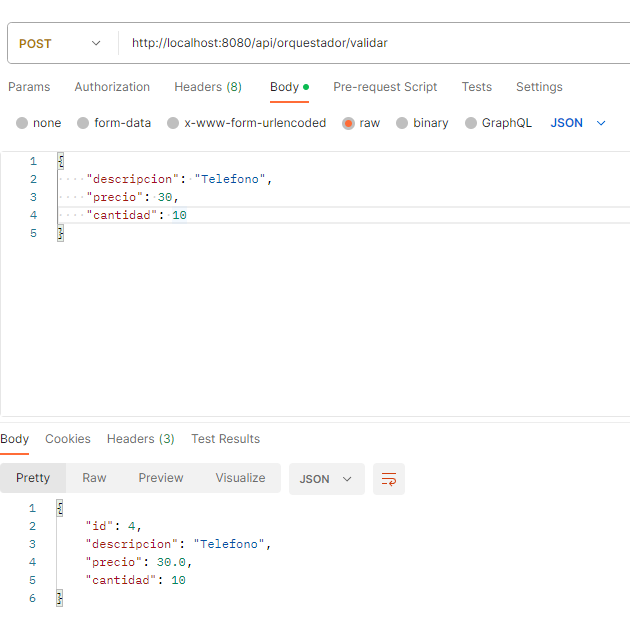
**ORQUESTADOR SERVICE**

**DOMINIO SERVICE**

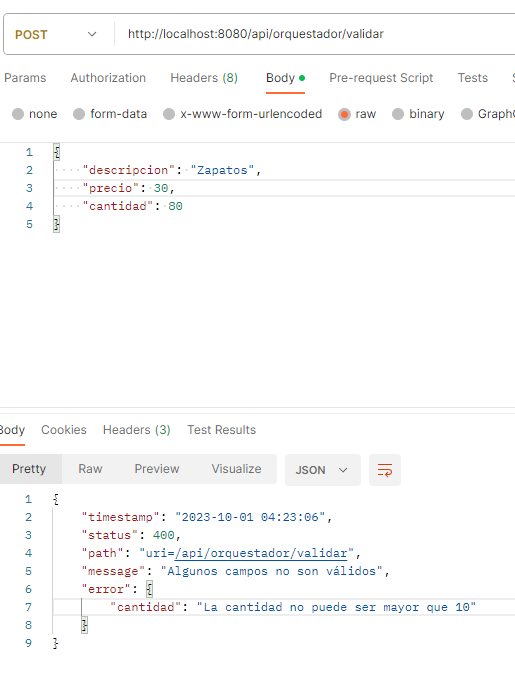
**DB**

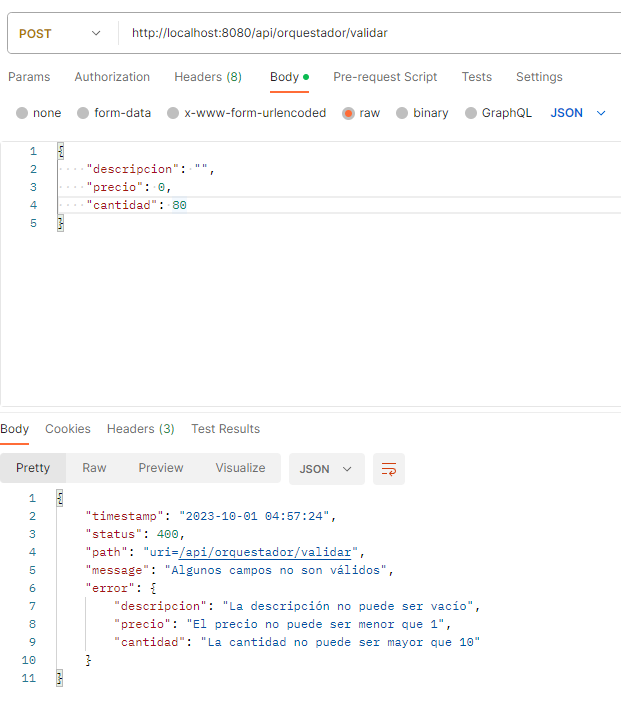
**EVIDENCIAS**

**Datos Correctos**

****

**Datos no válidos**

****

****

**PARTE 2**

## Docker

Ejecutar los siguientes comandos en la ruat principal de cada microservicio:

### config-service

1. Abrir Terminal en /prueba-ms-omni/config-service

2. docker build -t omni-config-service .

### eureka-service

1. Abrir Terminal en /prueba-ms-omni/eureka-service

2. docker build -t omni-eureka-service .

y los siguientes de igual manera

docker build -t omni-gateway-service .

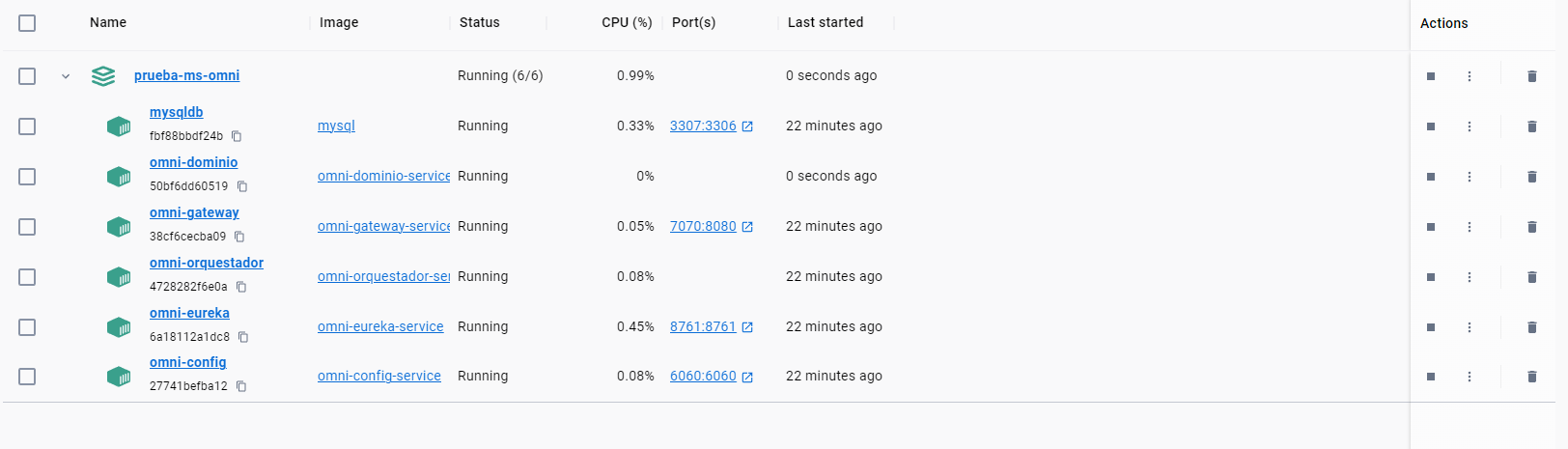
docker build -t omni-orquestador-service .

docker build -t omni-dominio-service .

### Finalmente Ejecutar docker-compose

1. Abrir terminal en /prueba-ms-omni

2. docker-compose up -d

****