

Prueba técnica Backend ViveLibre

Introducción

Esta prueba evalúa tu capacidad para el filtrado y procesamiento de estructuras de datos y desarrollar un microservicio que interactúe con un servicio externo. Tendrás que demostrar habilidades en procesamiento de datos, desarrollo de APIs y buenas prácticas de programación.

Ejercicio 1: Filtrado y Procesamiento de Libros

Se provee un archivo con una lista de libros (books.json) y se espera que realices las siguientes operaciones:

1. Filtra los libros con más de 400 páginas y aquellos cuyo título contenga "Harry".
2. Obtén los libros escritos por "J.K. Rowling".
3. Lista los títulos ordenados alfabéticamente y cuenta cuántos libros ha escrito cada autor.
4. Convierte `publicationTimestamp` a formato `AAAA-MM-DD`.
5. Calcula el promedio de páginas y encuentra el libro con más y menos páginas.
6. Añade un campo `wordCount` (250 palabras por página) y agrupa los libros por autor.
7. *(Opcional)* Verifica si hay autores duplicados y encuentra los libros sin `publicationTimestamp`.
8. *(Opcional)* Identifica los libros más recientes.
9. *(Opcional)* Genera un JSON con títulos y autores y exporta la lista a CSV (`id`, `title`, `author_name`, `pages`).

El código se implementará en **Java 17 o superior** en un proyecto **Maven**, además se espera un código claro y eficiente.

Cada punto detallado se implementará como un método independiente. Entrega tu solución en un repositorio en **GitHub**.

Ejercicio 2: Microservicio con Token

Se requiere implementar un microservicio que interactúe con un servicio externo que ya existe y devuelve un token. El flujo esperado es el siguiente:

1. **Consumo de Servicio Externo:** Implementa un cliente HTTP que obtenga un **token** de un servicio existente.
 - a. El servicio existente consta de las siguientes características:

Servicio de autenticación que escucha en el puerto 8080 y consta del siguiente *endpoint*:

cURL:

```
curl --location --request POST 'http://localhost:8080/token' \
--header 'Content-Type: application/json' \
--data-raw '{
  "username": "auth-vivelibre",
  "password": "password"
}'
```

Docker: `docker pull skeet15/auth-vivelibre`.

2. Procesamiento del Token:

- Extrae el **token** recibido.
- Agrega la **fecha y hora actual** en formato ISO 8601 (Ejemplo: `2025-02-27T14:00:00Z`).

3. Exposición de Microservicio:

- Implementa un microservicio que reciba una petición HTTP y devuelva una respuesta JSON con:

```
{
  "token": "TOKEN_RECIBIDO",
  "timestamp": "2025-02-27T14:00:00Z"
}
```

El microservicio deberá exponerse en el siguiente endpoint:

GET <http://localhost:8081/token>

4. Validaciones y Seguridad:

- Rechazar peticiones si el token es inválido o vacío.
- *(Opcional)* Agregar autenticación básica

5. Extras Opcionales:

- *(Opcional)* Logging y métricas (cantidad de peticiones procesadas).
- *(Opcional)* Persistencia de tokens en memoria o base de datos ligera.
- *(Opcional)* Expiración de tokens basada en tiempo.

El microservicio será desarrollado en **Java 17 o superior con Spring Boot**. Se valorará el uso de buenas prácticas y pruebas unitarias.

El proyecto deberá ser presentado en un repositorio en GitHub. Además, se deberá empaquetar el servicio en un docker. De manera que se proporcionarán **dos links**, uno al repositorio Git y otro al **DockerHub** donde se encuentre el contenedor de tu microservicio.

Adicionalmente, se puede entregar el **docker-compose.yml** correspondiente para comprobar que funciona correctamente.

Entrega

- Código fuente bien estructurado y documentado.
- Explicación breve de las decisiones técnicas tomadas.

Se valorará positivamente:

- Documentación (README, Swagger, etc.)
- Tests unitarios
- Configuración para múltiples entornos