

A. Configuración completa de Kafka en el User Service

El objetivo es que cada vez que un usuario sea creado o actualizado, se envíe un evento a Kafka para que otros microservicios lo puedan consumir.

Actualización del código para Kafka:

Estructura actualizada del User Service:

```
user-service/  
  ??? src/  
    ? ??? main/  
      ? ? ??? java/  
        ? ? ? ??? com/  
          ? ? ? ??? example/  
            ? ? ? ??? userservice/  
              ? ? ? ??? UserServiceApplication.java  
                ? ? ? ??? controller/  
                  ? ? ? ? ??? UserController.java  
                    ? ? ? ??? model/  
                      ? ? ? ? ??? User.java  
                        ? ? ? ??? repository/  
                          ? ? ? ? ??? UserRepository.java  
                            ? ? ? ??? service/  
                              ? ? ? ? ??? UserService.java  
                                ? ? ? ??? producer/
```

```
? ? ?      ? ??? UserEventProducer.java
? ? ?      ??? config/
? ? ?      ??? KafkaProducerConfig.java
? ??? main/resources/
? ? ??? application.yml
??? pom.xml
```

// (Truncated for brevity)

B. Configuración de Docker para los Microservicios

Dockerfile para cada Microservicio:

```
# Usa la imagen base de OpenJDK
```

```
FROM openjdk:11-jdk-slim
```

```
# Establece el directorio de trabajo
```

```
WORKDIR /app
```

```
# Copia el archivo JAR al contenedor
```

```
COPY target/user-service-0.0.1-SNAPSHOT.jar user-service.jar
```

```
# Expone el puerto en el que corre el servicio
```

```
EXPOSE 8081
```

// (Truncated for brevity)

Docker Compose para orquestar múltiples contenedores:

version: '3.8'

services:

zookeeper:

image: wurstmeister/zookeeper:3.4.6

ports:

- "2181:2181"

kafka:

image: wurstmeister/kafka:2.12-2.3.0

ports:

- "9092:9092"

environment:

KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://localhost:9092

KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181

elasticsearch:

image: docker.elastic.co/elasticsearch/elasticsearch:7.9.3

ports:

- "9200:9200"

environment:

discovery.type: single-node

// (Truncated for brevity)

