

UNIVERSIDAD DIEGO PORTALES

# Informe Tarea II: Kafka

Sistemas Distribuidos

■ María José Erazo Gonzalez

■ Catalina Gómez Zapkovic

---

## 1. Problema y Solución

Se tiene que el gremio de sopaipilleros de Chile, crece a un ritmo agigantado por lo que requiere de una actualización que de sus plataformas informáticas para poder ser capaces de gestionar procesos de manera eficiente y escalable.

Para la solución a su problema, se realiza un sistema distribuido con Kafka de manera de poder gestionar los procesos internos del gremio, esto es levantar un servidor que pueda recibir peticiones, a su vez un broker de Kafka con tópicos y particiones con los cuales el servidor se pueda comunicar y realizar los procesos requeridos por el gremio de sopaipilleros. Aparte se realiza una base de datos que permitirá registrar los datos que son necesarios para este sistema distribuido.

## 2. Módulos de código

### 2.1. Docker Compose

En esta tarea se realizó, al igual que la primera, un Docker Compose con todos los servicios que se utilizarán para poder montar el sistema distribuido implementado. Este compose montará contenedores de manera separada para Zookeeper, Kafka, Api, Stock, Ubicación, Venta y DB (base de datos), por lo que todo lo realizado está “dockerizado”.

### 2.2. API

Se realizó una API en donde se conecta el servidor con Kafka levantándolo en el puerto 3000. En este se crea el broker y el producer de Kafka, este último es quien recibirá la información que se le envíe a cada topic. También se crearán todos los tópicos a utilizar a través de tres rutas distintas, estas son las siguientes.

- **Ruta 1: Registro de miembros**

En esta ruta se tiene una función asíncrona que estará recibiendo los datos que se envíen para registrar un nuevo miembro, la ruta es “/new”. Aquí también se crea el tópico de los nuevos miembros llamado “newMember”, este al crearse tiene como configuración predeterminada 2 particiones, como se pide que al ingresar un nuevo miembro que sea premium este vaya a una partición distinta, en esta función se realiza un if que verá cuando el miembro ingresado sea premium para poder enviar dicha información a esta partición.

- **Ruta 2: Registro de ventas**

En esta ruta, al igual que en la anterior, se recibe la información que llegue a “/nuevaVenta” y se crean dos tópicos, estos son “nuevaVenta” y “Ubicaciones”. En el tópico nuevaVenta se recibirá la información de la venta, esto es *Cliente*, *Cantidad de Sopaipillas* y *Stock restante* la cual se procesará en los otros módulos de código. En el tópico de ubicaciones se recibirá solo la *Ubicación*, *Patente* y *Hora*, información que también se procesará en los otros módulos de código explicados más adelante en este informe.

- **Ruta 3: Aviso de carrito prófugo**

Al igual que en las anteriores rutas, aquí llegará la información que se envíe a la ruta “/carritoPerdido”, esta también usará el tópico de “Ubicaciones” para recibir la información. Este se utilizará solo cuando una persona quiera denunciar a un carrito prófugo y deberá enviar las coordenadas de dicho carrito. Cabe destacar que estas serán enviadas a una partición distinta a la que se utiliza en la ruta anterior.

**■ Ruta 4: Ubicación**

Como cuarta ruta se agregó “**ubicacion**”. Como en la problemática se especifica que los carritos poseen sistemas inteligentes con internet y GPS, se asume que estos son capaces de enviar su ubicación cada cierto tiempo para poder tener el registro de la posición que tiene dicho carrito, es decir, sus coordenadas. Para esta ruta también se utilizará el tópico de “*Ubicaciones*” para recibir las entradas pero con una partición distinta a la que se utiliza en la ruta de “*carritoPerdido*”.

**2.3. Procesamiento de Venta**

En esta módulo de código

**2.4. Procesamiento de Stock****2.5. Procesamiento de Ubicación****3. Configuración de Kafka****4. Respuestas a las preguntas**

1. ¿Cómo Kafka puede escalar tanto vertical como horizontalmente? Relacione su respuesta con el problema asociado, dando un ejemplo para cada uno de los tipos de escalamiento.
2. ¿Qué características puede observar de Kafka como sistema distribuido? ¿Cómo se reflejan esas propiedades en la arquitectura de Kafka?

**5. Anexos****5.1. Enlace a Video**

<https://drive.google.com/drive/folders/1t01ZKLOTjSJBgPLkujt7UBCZnZolx7Yt?usp=sharing>

**5.2. Enlace a Github**

<https://github.com/majo-erazo/Tarea2-SD>