

## Resolucion de Piezas del pastel

1. Teniendo en cuenta que la base de datos tiene un cuello de botella y no acepta solicitudes en paralelo se pueden implementar las siguientes soluciones:
  - a. En consultas de selección, pedir los datos necesarios, de forma de que no se devuelva toda la tabla. Realizar un estudio de los tiempos de todas las consultas y centrarse en optimizar aquellas que tarden más / sean más utilizadas.
  - b. Optimizar las consultas, especialmente aquellas que puedan realizarse de manera conjunta mediante uniones y comprobar que la consulta sea más rápida que realizarlas de forma independiente.
  - c. Utilizar un sistema distribuido asíncrono antes de la consulta como KAFKA, de forma que este administre y ponga en cola las consultas, para que posteriormente, se vayan procesando las mismas cuando la base de datos tenga los recursos.

De todas maneras las soluciones a estos problemas las realiza alguien especializado en base de datos de manera que se cubran todos los puntos esenciales, (organización, consultas, procedimientos almacenados, etc.).

2. Es el ej2.go en el repositorio
3. Es el ej3.go en el repositorio
4. El esquema mostrado emplea una arquitectura de microservicios para, en este caso, una aplicación web.
  - a. Client apps: son los usuarios que a través de la web (navegador) envían solicitudes a un servidor.
  - b. API Gateway: las solicitudes son procesadas por este punto de control centralizado, el cual establece un enrutamiento al microservicio correspondiente. Además puede manejar la autenticación, balance de carga, etc.
  - c. Catalog, Shopping Cart, Discount, Rendering: son los servicios consultados. Catalog puede contener la información de productos por ejemplo, Shopping Cart, maneja los productos antes de la compra final (carrito), etc. Cada uno de estos microservicios tiene su propia base de datos, por lo tanto son prácticamente independientes uno del otro, permitiendo escalabilidad y mantenimiento.
  - d. Message Broker: es un “servicio” dedicado a tareas asíncronas, por ejemplo envío de notificaciones, email, actualización de inventarios, etc. de manera que ante una gran demanda de solicitudes no se bloquee el flujo del proceso.

Dos mejoras básicas y sencillas son:

- Logs y monitoreo: logs para ver el funcionamiento de los microservicios y el monitoreo para entender la carga y predecir algún cuello de botella por ejemplo.
- Autenticación y autorización: cada microservicio puede requerir un token, como JWT, para mejorar la seguridad.

5. Para explicar de forma no técnica una tecnología que implica el uso de microservicios, y mejora la escalabilidad y reducción de tiempo de respuesta, voy a dar el siguiente ejemplo:

Imaginemos que una persona (cliente) necesita comprar un producto (orden), y para eso, va al kiosco donde hay un solo empleado (servicio).

Bajo esta idea, si solo hay un cliente, la compra es rápida, pero si tenemos 10.000 clientes, la cola sería interminable y el tiempo de espera, muy largo.

Luego si decidimos implementar una nueva tecnología como, convertir el kiosco en un supermercado, podemos mejorar la escalabilidad y tiempo de respuesta. Poner más empleados (cajas) cuando aumente la demanda (muchos clientes en cola), o sacar cajas, para ahorrar recursos.