

Lab9: Microservicios

Proceso de migración

Para definir la prioridad de los componentes a migrar analicemos primero las dependencias que existen entre ellos.

- **Manejo de usuarios:** es una herramienta de la cual dependen el resto de los elementos.
- **Catálogo:** es requerido por el proceso de orden y por el proceso de atención al usuario (sí, por ejemplo, se requieren detalles de un producto).
- **Órdenes:** es requerido solo por el proceso de atención al usuario si el usuario requiere detalles de una orden.
- **Atención al usuario:** no es requerido por ninguno de los otros elementos.

Este mapa de dependencias ya deja ver un orden en el que deberíamos migrar los componentes que forman el monolito.

Podemos definir un conjunto de servicios de alta prioridad y uno de baja prioridad. El de alta prioridad debería ser migrado en este orden:

Servicios de alta prioridad

1. **Manejo de usuarios:** Gracias a esta herramienta se hacen procesos de autenticación y autorización, por lo que es la puerta de entrada a la aplicación, el servicio más usado, y un pre-requisito para el resto de servicios. Por eso deberá ser migrado primero.
2. **Catálogo:** Este servicio también podemos imaginar que recibe muchas solicitudes, pues sería un pre-requisito para realizar órdenes o consultar detalles de un producto o un listado filtrado – lo cual permite al usuario interactuar con la aplicación.
3. **Órdenes:** Es de suponer que la realización y consulta del estatus de una orden son dos de las principales actividades de un usuario en la aplicación. Es también un servicio de alta prioridad que podrá migrarse una vez se hayan migrado los dos componentes anteriores.

Servicio de baja prioridad

1. **Atención al cliente:** Es un servicio de menos uso que los tres anteriores. Puede ser el último componente a migrar, ya que tampoco hay otros componentes que dependan de él.

Migración de datos

Actualmente la aplicación utiliza una única base de datos relacional. En el nuevo modelo de microservicios cada uno de los cuatro componentes manejará una base de datos propia: Se crearán Tablas y Esquemas nuevos, uno por microservicio. Mientras ocurre la migración se debe mantener acceso tanto a la base de datos preexistente como a las nuevas que se generen para cada microservicio. Posteriormente, cuando todos los componentes hayan sido migrados, se mantendrá acceso de lectura a la vieja base de datos. Para garantizar consistencia entre las distintas BDs adoptaremos un modelo de consistencia eventual – el principal momento en que se debe garantizar la consistencia es cuando se realizan órdenes y se actualiza el inventario con el catálogo.

Descripción de la arquitectura

Se generarán cinco servicios:

- **Servicio de usuarios:** Maneja los perfiles de usuario, la autenticación y autorización. Cuenta con su propia base de datos para usuarios.
- **Servicio de catálogo:** Encargado de manejar información detallada de cada producto, así como niveles de inventario. Permite consultar listados de productos y detalles de uno. Se incluirá caché para agilizar los procesos de obtención de listado o de búsqueda en el catálogo.
- **Órdenes (Carrito):** El usuario interactuará con este servicio mientras el usuario selecciona los productos que agrega a su carrito. Naturalmente, este servicio interactúa con el servicio de catálogo para realizar búsquedas o consultar detalle de un producto.
- **Órdenes (Check-out):** Este servicio crea la orden una vez que el usuario la finaliza. El proceso de pago puede ser manejado por un tercero.
- **Atención al usuario:** Maneja consultas del usuario, devoluciones, quejas.

La comunicación con servicios lucirá así:

- Tendremos un **API-Gateway** que funcione como único punto de acceso para frontend y redireccione las solicitudes al servicio adecuado.
- Para la comunicación entre los servicios internos (los cinco que listamos en la sección anterior) tendremos una **mall de servicios**.

Pasos de la migración:

1. Implementar API Gateway
 - a. Se inicializará un Gateway para centralizar las solicitudes externas.
2. Implementar servicio de usuarios:
 - a. Se creará una BD de usuarios separada y las funcionalidades relacionadas con el manejo de usuarios se migrarán a este servicio.
 - b. Las solicitudes de autenticación y autorización se redireccionarán a este servicio luego de que haya sido implementado.
3. Implementar servicio de catálogo:
 - a. También se crea una base de datos para el catálogo y las funcionalidades relacionadas a catálogo se migrarán a este servicio.
 - b. Las solicitudes de catálogo se redireccionarán a este servicio.
4. Implementar servicio de órdenes (carrito) :
 - a. Se creará una base de datos separada para las órdenes en carrito.
 - b. Se implementa la comunicación asíncrona con el servicio de catálogo.
 - c. Se redireccionan solicitudes relacionadas con el carrito a este servicio.
5. Implementar servicio de checkout.
 - a. Se creará una base de datos separada para las órdenes completadas.
 - b. Se redireccionan las solicitudes relacionadas con checkout a este servicio.
6. Implementar servicio para atención a clientes:
7. (Este paso se hace gradualmente) Deshabilitar las funcionalidades del monolito una vez que tengan un componente análogo en los microservicios.