

Diseño de Arquitectura de Microservicios

Actualmente la aplicación monolítica consta de las siguientes funcionalidades:

- **Gestión de usuarios:** administra los perfiles de los usuarios, la autenticación y la autorización. Almacena información personal, administra las sesiones de inicio de sesión y gestiona las preferencias de los usuarios.
- **Catálogo de productos:** mantiene una lista completa de productos, incluidas descripciones, precios, imágenes y niveles de inventario. Admite funciones de búsqueda y categorización de productos.
- **Procesamiento de pedidos:** gestiona todos los aspectos del proceso de pedido, desde la gestión del carrito hasta la realización del pedido, el procesamiento de pagos y el seguimiento del historial de pedidos.
- **Atención al cliente:** gestiona consultas, devoluciones, quejas y comentarios de los clientes a través de un sistema basado en tickets integrado con las bases de datos de usuarios y pedidos.

Se propone dividir la aplicación en microservicios basados en el dominio del negocio.

Microservicio de Usuarios: gestionará la autenticación, autorización y perfiles de usuarios.

Base de datos dedicada: almacenamiento de datos de los usuarios

Comunicación: APIs REST para solicitudes CRUD de usuarios.

Microservicio de Catálogo de Productos: administra la lista de productos, sus categorías, inventarios, precios y descripciones etc.

- **Base de datos dedicada:** almacena información de productos e inventarios.
- **Comunicación:** APIs REST para consultas de productos, agregado nuevos productos y actualización.

Microservicio de Procesamiento de Pedidos: nos permite realizar pedidos, desde el carrito hasta el pago y el seguimiento del estado del pedido.

- **Base de datos dedicada:** almacena carritos, pedidos e historial de compras.
- **Comunicación:** APIs REST para procesar pedidos.

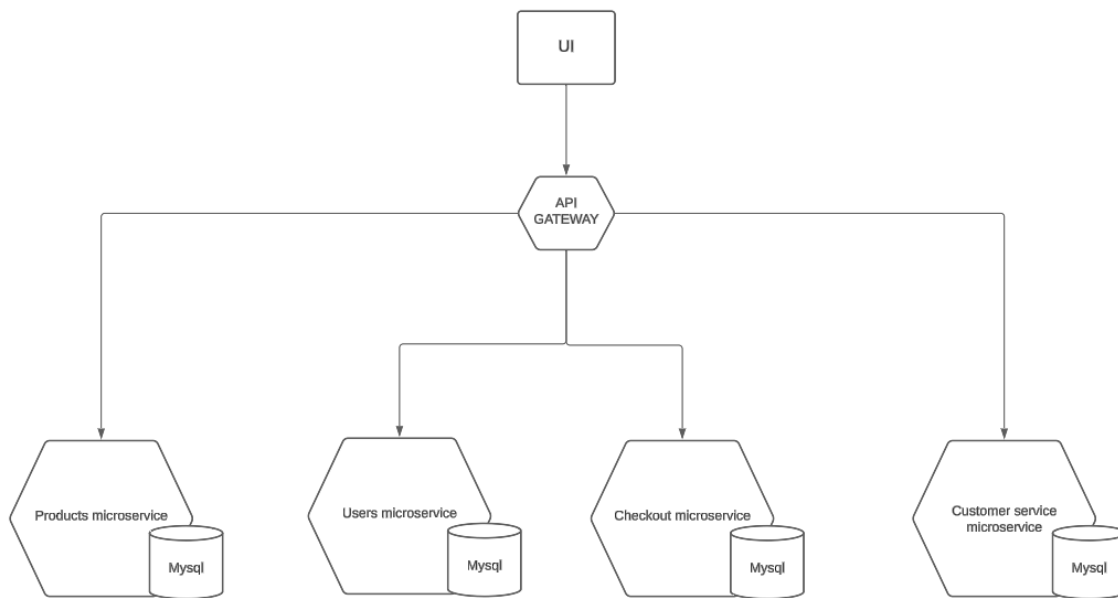
Microservicio de Atención al Cliente: gestiona el sistema de tickets, consultas y devoluciones.

- **Base de datos dedicada:** almacena tickets, estado y datos de interacción con clientes.
- **Comunicación:** APIs REST para gestionar tickets y mensajes a otros microservicios para obtener datos específicos de pedidos o usuarios.

Interacciones y Comunicación entre Microservicios

La comunicación entre microservicios será principalmente a través de:

- **APIs RESTful:** comunicación directa y síncrona/asíncrona.
- **Mensajería asíncrona:** utilizar un sistema de mensajería como Rabbit o Kafka útil para notificaciones de actualización de inventarios, pagos completados o generación de reportes.



Hoja de Ruta para la Migración a Microservicios

Fase de Migración de Servicios:

1. **Migración del Servicio de Usuarios:** Desacoplar y migrar los componentes relacionados con la autenticación, autorización y gestión de perfiles de usuario hacia un microservicio autónomo.
2. **Migración del Servicio de Catálogo de Productos:** Crear un microservicio dedicado para gestionar el inventario, descripciones, precios y categorización de productos.
3. **Migración del Servicio de Procesamiento de Pedidos:** Establecer un microservicio específico para gestionar el ciclo de vida de los pedidos, desde el carrito hasta el procesamiento de pagos y la actualización del historial de compras.
4. **Migración del Servicio de Atención al Cliente:** Transferir las funcionalidades de soporte al cliente y el sistema de tickets a un microservicio independiente.

Fase de Despliegue y Pruebas:

1. **Pruebas de Integración en Entornos de QA:** Evaluar la interacción y comunicación entre los microservicios en entornos de prueba controlados antes de promoverlos al entorno de producción.
2. **Implementación de Pruebas de Estrés:** Ejecutar pruebas de carga para asegurar que cada microservicio puede escalar de manera eficiente bajo condiciones de alta demanda.
3. **Monitoreo Activo:** Establecer un sistema de monitoreo constante para rastrear errores, latencias y métricas de desempeño de cada microservicio, permitiendo la detección temprana de problemas.
4. **Pruebas de Regresión:** Realizar pruebas de regresión para validar que cada microservicio mantenga la funcionalidad esperada y que no se introduzcan errores tras la migración.