

1. Introducción a los microservicios: la aplicación se divide en pequeños servicios autónomos.

- * Cada servicio es una pequeña aplicación con responsabilidad única, con propio proceso y base de datos.
- * Se comunican mediante APIs o mensajería asincrónica, evitando acoplamientos fuertes
- * Se puede actualizar una pieza sin tumbar todo el sistema

2. Anatomía de una arquitectura distribuida

- **API Gateway:** es la puerta de entrada única, maneja la autenticación, el enrutamiento y el balanceo de carga.
- **Persistencia Descentralizada:** Se rompe el dogma de "una base de datos para todo"
- **Resiliencia:** se asume que los fallos ocurrirán por eso se implementan **circuit breakers** y **retries** para evitar que un fallo haga colapsar la aplicación

3. Diseño y Domain-Driven Design (DDD)

- * cada microservicio corresponde a un contexto específico
- * Las APIs deben ser estables y versionadas para que los equipos no rompan la compatibilidad

4. Empaquetado (Docker): el contenedor es la unidad estándar de despliegue.

5. Orquestación local con Docker Compose: un archivo YAML define todos los servicios, redes y volúmenes necesarios