

Para **The Open Group**, el **desacoplamiento** es un principio clave en la arquitectura empresarial que busca reducir la dependencia entre componentes o sistemas, permitiendo que evolucionen de manera independiente sin afectar la funcionalidad general.

Conceptos Claves del Desacoplamiento según The Open Group

1. **Modularidad:** Diseñar sistemas en bloques independientes que puedan ser modificados o reemplazados sin afectar a otros.
2. **Interoperabilidad:** Uso de estándares abiertos y APIs para permitir la comunicación entre sistemas sin una fuerte dependencia tecnológica.
3. **Flexibilidad y Escalabilidad:** Facilita la evolución y adaptación a nuevas tecnologías sin grandes reestructuraciones.
4. **Arquitecturas Basadas en Servicios:** Implementación de arquitecturas como **SOA (Service-Oriented Architecture)** o **Microservicios** para minimizar dependencias directas.
5. **Arquitecturas Basadas en Eventos (EDA - Event-Driven Architecture):** Promueve la comunicación asincrónica mediante eventos para evitar bloqueos o dependencias rígidas.
6. **Uso de Estándares:** Frameworks como **TOGAF, ArchiMate e IT4IT** ayudan a estructurar arquitecturas desacopladas, permitiendo separar capas de datos, aplicaciones e infraestructura.

Aplicación del Desacoplamiento en Arquitectura Empresarial

- **En Banca y Finanzas (BIAN + Open Group):** Se utilizan **Service Domains** desacoplados para mejorar la agilidad y la integración con otros sistemas.
- **En Gobierno de TI:** IT4IT ayuda a definir modelos de gestión de servicios que desacoplan las funciones de operación y desarrollo.
- **En Cloud y DevOps:** Se promueve el desacoplamiento mediante contenedores, APIs y CI/CD.

Caso: Desacoplamiento del Módulo de Autenticación en un Banco

Problema Inicial (Arquitectura Acoplada)

Un banco tiene un sistema monolítico donde la autenticación de clientes está embebida en el core bancario. Cada vez que se necesita cambiar el método de autenticación (por ejemplo, de usuario/contraseña a autenticación biométrica), se requiere modificar el core, lo que es costoso y riesgoso.

Solución con Desacoplamiento

1. **Separar el servicio de autenticación** en un **Service Domain de BIAN** llamado «**Authentication**», aislándolo del core bancario.
2. Exponer la autenticación como una **API RESTful** independiente, que pueda ser consumida por:
 - Aplicaciones móviles y web del banco
 - Cajeros automáticos (ATMs)
 - Canales de atención al cliente
3. Implementar una **arquitectura basada en eventos (EDA)** para manejar autenticaciones en tiempo real sin afectar el sistema central.
4. Adoptar una solución basada en **OAuth 2.0 y OpenID Connect**, facilitando la interoperabilidad con terceros sin comprometer la seguridad.

Beneficios del Desacoplamiento

- **Evolución independiente:** Se pueden agregar nuevos métodos de autenticación sin tocar el core bancario.
- **Mayor seguridad:** Aplicaciones externas solo interactúan con la API de autenticación sin acceder directamente al core.
- **Reducción de costos y riesgos:** Cualquier cambio se implementa en la API sin impactar otros módulos.
- **Facilita la omnicanalidad:** El mismo servicio de autenticación puede ser utilizado por múltiples canales sin duplicación de lógica.

Ejemplo 1: Desacoplamiento del Motor de Pagos

Problema Inicial (Arquitectura Acoplada)

Un banco tiene su motor de pagos embebido en su **core bancario**. Cada vez que se necesita integrar con un nuevo proveedor de pagos (por ejemplo, Visa, Mastercard, Apple Pay o PIX), se requieren cambios en el core, generando altos costos y tiempos de implementación prolongados.

Solución con Desacoplamiento

1. Extraer la funcionalidad de pagos en un **Service Domain de BIAN: «Payment Execution»**, convirtiéndolo en un microservicio independiente.
2. Implementar una **API de pagos unificada**, con capacidad de conectarse a múltiples proveedores sin modificar el core.
3. Usar una **arquitectura basada en eventos (EDA)** para procesar pagos de forma asíncrona y reducir la dependencia del core.

4. Utilizar un **orquestador de pagos** que seleccione dinámicamente el mejor proveedor según costos y SLA.

Beneficios

- **Menos impacto en el core bancario:** Se pueden integrar nuevos métodos de pago sin modificar el core.
- **Mayor flexibilidad:** Fácil integración con nuevos proveedores de pago sin grandes desarrollos.
- **Reducción de riesgos:** Se pueden realizar pruebas en un entorno desacoplado antes de tocar el core.

Ejemplo 2: Desacoplamiento de la Gestión de Saldos

Problema Inicial (Arquitectura Acoplada)

El módulo de consulta de saldos está dentro del **core bancario**. Cada vez que una aplicación móvil o web quiere consultar el saldo de un cliente, tiene que conectarse directamente al core, lo que genera una **alta carga** en el sistema y afecta el rendimiento en días de alta demanda.

Solución con Desacoplamiento

1. Crear un **Service Domain de BIAN: «Current Account Fulfillment»**, donde los saldos se manejen como un servicio separado.
2. Implementar una **capa de caché** (Redis o Memcached) para almacenar consultas de saldo recientes y evitar accesos innecesarios al core.
3. Exponer una **API de consulta de saldos** desacoplada del core, optimizada para respuestas rápidas y con soporte para múltiples canales.
4. Implementar una **arquitectura basada en eventos**, donde cualquier cambio en el saldo (por transacciones o movimientos) se refleje en tiempo real sin depender de consultas constantes al core.

Beneficios

- **Mejor rendimiento:** Se reduce la carga en el core bancario al evitar consultas directas innecesarias.
- **Omnicanalidad:** El mismo servicio de saldo puede ser utilizado por cajeros, aplicaciones móviles, banca web y otros canales.
- **Resiliencia y disponibilidad:** Si el core bancario falla, la API de saldos puede seguir funcionando con la información en caché.

Julio Pari (IT Architect BIAN)



Especialista BIAN Semantic API | Gobierno de Integración | IBM Integration CP4I | IBM API Connect 10 | IBM ACE | IBM DataPower | OpenShift | Azure | AWS. Cualquier consulta envíame un mensaje a: info@arquitecturabank.com o sino a través de LinkedIn: <https://www.linkedin.com/in/juliopari/>