

El desacoplamiento es un principio clave en la arquitectura empresarial y está alineado con los principios de **modularidad, interoperabilidad y reutilización** promovidos en TOGAF.

1. En la Arquitectura de Aplicaciones

- **Uso de microservicios y APIs:** Se recomienda diseñar aplicaciones desacopladas mediante microservicios y API Management, permitiendo flexibilidad y escalabilidad.
- **Separación de capas:** Se favorece la separación entre la capa de presentación, lógica de negocio y datos.
- **Interoperabilidad mediante estándares abiertos:** TOGAF fomenta el uso de estándares como REST, SOAP, GraphQL y eventos.

2. En la Arquitectura de Datos

- **Data Mesh y desacoplamiento de dominios:** Se busca evitar los silos de datos y promover el acceso federado a la información.
- **Eventos y CQRS:** Se pueden utilizar patrones como **Command Query Responsibility Segregation (CQRS)** y **Event Sourcing** para separar operaciones de lectura y escritura.

3. En la Arquitectura Tecnológica

- **Cloud y edge computing:** TOGAF favorece arquitecturas híbridas y multi-cloud, desacoplando cargas de trabajo según necesidades del negocio.
- **Infraestructura como código (IaC):** Automatización para reducir dependencia de infraestructura específica.

TOGAF y el Desacoplamiento en ADM

Dentro del **Architecture Development Method (ADM)** de TOGAF, el desacoplamiento se puede aplicar en diferentes fases:

- **Fase A (Visión de Arquitectura):** Se definen principios como modularidad y desacoplamiento.
- **Fase B (Arquitectura de Negocio):** Se identifican procesos y dominios que deben estar desacoplados para optimizar flujos de negocio.
- **Fase C (Arquitectura de Sistemas de Información):** Se diseña la interoperabilidad entre aplicaciones y datos de manera desacoplada.
- **Fase D (Arquitectura Tecnológica):** Se definen tecnologías y estrategias de desacoplamiento a nivel de infraestructura.

Conclusión

El desacoplamiento en TOGAF es clave para crear arquitecturas flexibles, escalables y resilientes. Se logra mediante el uso de **APIs, eventos, cloud, microservicios y arquitectura orientada a capacidades**.

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/>