

Los pilares del **desacoplamiento** en arquitectura de software buscan reducir la dependencia entre componentes para mejorar la flexibilidad, escalabilidad y mantenibilidad. A continuación, se detallan los tres pilares mencionados:

1. Diagrama de Dependencia

- Representa las relaciones entre los diferentes componentes del sistema.
- Permite identificar acoplamientos fuertes y debilidades en la arquitectura.
- Ayuda a visualizar qué partes del sistema dependen unas de otras y cómo reducir estas dependencias.
- Puede utilizarse para definir estrategias de separación, como la introducción de capas intermedias o patrones como *event-driven architecture*.

2. Modelo Ontológico

- Define una estructura semántica común para la interoperabilidad entre sistemas.
- Se basa en estándares y modelos conceptuales que facilitan la alineación de datos y procesos.
- Reduce la dependencia entre servicios al utilizar definiciones comunes y evitar acoplamientos específicos entre aplicaciones.
- En el contexto de BIAN, este modelo puede relacionarse con el *Business Object Model (BOM)* y otros estándares semánticos.

3. Contrato de Artefactos

- Establece acuerdos claros sobre cómo interactúan los componentes desacoplados.
- Incluye contratos de API, definiciones de mensajes, formatos de datos y reglas de interacción.
- Permite evolucionar los componentes sin afectar a los consumidores.
- Se puede implementar utilizando OpenAPI, AsyncAPI, GraphQL, o definiciones de eventos en sistemas event-driven.

Estos tres pilares trabajan en conjunto para garantizar que los sistemas sean más modulares, escalables y resistentes al cambio.

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/>