

Punto 2: Arquitectura Orientada a Servicios (SOA)

Daniel Muñetón Piedrahíta

Cédula: 1214721642

¿Qué es SOA?

La Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés) es un modelo de diseño de software que organiza las aplicaciones en servicios independientes, autónomos y reutilizables, que pueden integrarse fácilmente en distintos sistemas.

Estos servicios se comunican entre sí a través de protocolos y estándares abiertos (XML, SOAP, WSDL, REST, JSON, UDDI), lo que permite la interoperabilidad entre plataformas, lenguajes de programación y tecnologías diversas.

SOA no es un producto específico ni una tecnología en sí misma, sino un estilo arquitectónico basado en principios como:

- **Bajo acoplamiento:** los servicios dependen mínimamente entre sí.
- **Alta cohesión:** cada servicio cumple una única función clara.
- **Reutilización:** los servicios pueden emplearse en distintos contextos.
- **Autonomía:** cada servicio se ejecuta de manera independiente.
- **Descubrimiento dinámico:** los servicios pueden registrarse y ser encontrados en repositorios.
- **Interoperabilidad:** pueden interactuar aunque estén contruidos en diferentes tecnologías.

Elementos básicos de SOA

Consumidores

Aplicaciones o usuarios que utilizan los servicios.

Ejemplo: una app móvil que consulta disponibilidad de vuelos.

Bus de Servicios (ESB - Enterprise Service Bus)

Canal de comunicación que conecta los servicios.

Se encarga de enrutar mensajes, transformar datos y garantizar la interoperabilidad.

Repositorio de Servicios

Catálogo donde se registran los servicios disponibles, sus descripciones y cómo usarlos.

Servidores

Infraestructura donde se alojan y ejecutan los servicios, asegurando rendimiento y disponibilidad.

Ventajas de SOA

Para la organización

- Mejor toma de decisiones: acceso a información integrada y actualizada.
- Mayor productividad: unificación de procesos y sistemas.
- Mejora en relaciones con clientes y proveedores: integración más sencilla con terceros.
- Capacidad de respuesta ágil: facilita la creación de portales unificados de servicios.

Para TI

- **Reutilización de servicios:** aprovecha sistemas existentes, incluso los más antiguos.
- **Desarrollo más rápido y económico:** servicios reutilizables reducen tiempos y costos.
- **Mayor seguridad y gestión:** autenticación y autorización robusta en los servicios.
- **Flexibilidad:** permite escalar e integrar nuevas funcionalidades sin rehacer todo el sistema.

Desventajas o retos de SOA

- Complejidad en la administración del Bus de Servicios (ESB).
- Sobrecarga en rendimiento debido al uso intensivo de XML/SOAP.
- Mayor esfuerzo inicial en la gobernanza de servicios (definir estándares, seguridad y control).
- Riesgo de duplicidad de servicios si no existe un repositorio bien gestionado.

Gobernanza en SOA

La gobernanza es clave para que una implementación SOA sea exitosa. Consiste en establecer políticas, procesos y normas que regulen:

- El ciclo de vida de los servicios.
- La seguridad (autenticación, autorización y cifrado).
- La documentación y estándares de integración.
- El monitoreo y la calidad del servicio (QoS).
- Relación de SOA con otras arquitecturas modernas

SOA sirvió como base para arquitecturas más recientes como los microservicios.

SOA: usa un ESB como mediador central y puede manejar servicios más grandes.

Microservicios: dividen el sistema en componentes más pequeños y livianos, comunicados generalmente por APIs REST.

Actualmente, muchas empresas migran de SOA a microservicios en la nube para obtener mayor escalabilidad y agilidad, aunque SOA sigue siendo válido en entornos empresariales complejos.

Ejemplos prácticos de SOA

Aerolínea

Servicios: reservas, pagos, check-in, programa de fidelización.

Los consumidores (apps móviles, agencias de viaje, página web) acceden a los servicios sin importar la plataforma.

El ESB conecta reservas con pagos y check-in.

El repositorio describe cómo usar cada servicio.

Los servidores garantizan disponibilidad y rendimiento.

Banca

Servicios: consulta de saldo, transferencias, pagos en línea.

Integración con apps móviles, cajeros automáticos y portales web.

Salud

Servicios: historia clínica, gestión de citas, facturación, farmacia.

Permite la interoperabilidad entre hospitales, EPS y pacientes.

Conclusión

La Arquitectura Orientada a Servicios (SOA) es un modelo flexible y robusto que permite integrar y reutilizar aplicaciones en distintos entornos tecnológicos. Sus beneficios en productividad, interoperabilidad y reutilización la hacen una estrategia efectiva para empresas grandes y con sistemas heterogéneos.

No obstante, requiere una gestión sólida y gobernanza para evitar problemas de complejidad y redundancia. Aunque hoy los microservicios dominan, SOA sigue siendo relevante en entornos donde la integración de sistemas legados es prioritaria.

Referencias

- Espinal Martín, Y. (s. f.). Arquitectura de software. Arquitectura orientada a servicios. Serie Científica de la Universidad de las Ciencias Informáticas, 5(1). Universidad de las Ciencias Informáticas (Cuba). Recuperado de <https://doaj.org/article/9c2b1d1c00834ae4966d67805ee92b9b>
- Salinas, E., Cerpa, N., & Rojas, P. (2011). A service oriented architecture for the implementation of the personal software process. Ingeniare. Revista chilena de ingeniería, 19(1), 40–52.
<https://doi.org/10.4067/S0718-33052011000100005>