



# UNIVERSIDAD PRIVADA ANTENOR ORREGO

INGENIERÍA DE COMPUTACIÓN Y SISTEMAS  
**FACULTAD DE INGENIERÍA**

## ARQUITECTURA DE SOFTWARE

TEMA

### IMPLEMENTACION EN EL PERU DE SOA

DOCENTE

MENDOZA CORPUS, CARLOS ALFREDO;

#### INTEGRANTES

- ALFARO ASUNCIÓN JUAN CARLOS
- JACOBO ROMERO WENDY MITCHEL
- MIÑANO SICCHA BRAYAN DANIEL

TRUJILLO \_ 2024

# IMPLEMENTACION EN EL PERU DE SOA

La implementación de la **Arquitectura Orientada a Servicios** (SOA, por sus siglas en inglés) en el Perú ha sido un tema relevante en los últimos años. SOA es un enfoque de diseño de software que busca mejorar la calidad, eficiencia y agilidad de las empresas a través del trabajo colaborativo entre el área de tecnología y todas las demás áreas de la organización.

Desde la perspectiva de negocio, la implementación de SOA busca mejorar la calidad, eficiencia y agilidad de la empresa a través del trabajo colaborativo entre el área de tecnología y todas las demás áreas de la organización. Por lo tanto, las iniciativas de SOA son iniciativas de negocio y no solo de tecnología.

En Perú, se han realizado proyectos que implementan el SOA en diferentes sectores, incluida la justicia, donde se ha implementado un Sistema de Orientación al Adolescente para todos los adolescentes infractores, y en el sector de formación. También, se han realizado cursos de capacitación en Arquitectura Orientada a Servicios en el país. En estos cursos, se da una introducción a SOA, su base técnica en XML, WSD, y SOAP, y la implementación de SOA con diferentes bibliotecas de Java.

La implementación de SOA en el Perú ha sido abordada en diferentes sectores y se han ofrecido cursos de capacitación para promover su adopción en el país.

## 1-Definición de SOA.

- SOA es un enfoque arquitectónico que permite que diferentes servicios interactúen de forma independiente. Estos servicios son componentes autónomos que se combinan para ofrecer funcionalidad a gran escala en aplicaciones de software. Facilita la cooperación entre piezas de software en computadoras en red, permitiendo que los sistemas ejecuten múltiples servicios sin intervención humana.

En términos comerciales, SOA alinea los servicios de TI con los objetivos y procesos comerciales.

## 2- Elementos principales de SOA

**2.1- Impulsores SOA:** Aspectos como la competencia, la estrategia, las fuerzas regulatorias y del mercado impulsan la arquitectura empresarial y la gestión del rendimiento.

### 2.2- Habilitadores SOA.

**2.2.1- Modelo de negocio empresarial:** Es fundamental alinear los servicios con los objetivos de negocio.

**2.2.2- Optimización del rendimiento empresarial:** Evalúe el impacto de SOA y mida los procesos empresariales.

**2.2.3- Racionalización de cartera:** Permite la consolidación y simplificación de aplicaciones, datos e infraestructura.

**2.2.4- Definición de semántica empresarial:** Proporciona información comercial común relacionada con una empresa específica.

**2.2.5- Indicadores clave de rendimiento (KPI):** faciliten la medición del impacto de SOA.

### 3- Pasos para implementar SOA con éxito.

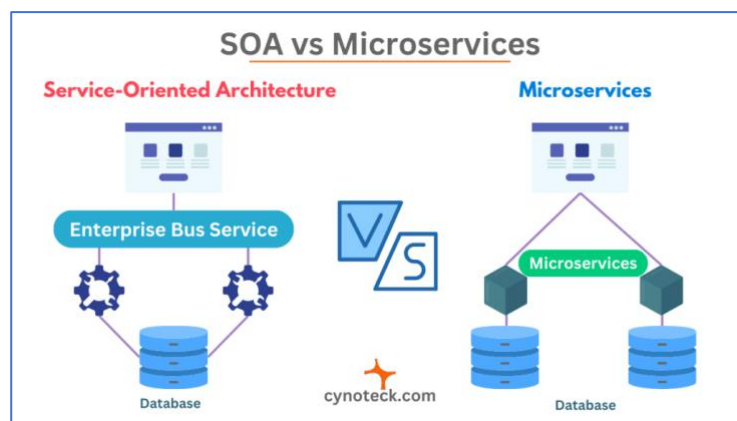
**3.1- Asegurar el apoyo de los directivos:** Comunicar los éxitos y fracasos de otras empresas que han adoptado SOA y explicar cómo se emularán prácticas probadas.

**3.2- Conseguir la confianza de todos los empleados:** La adopción exitosa de SOA requiere la colaboración de todos los miembros del equipo.

### **EJEMPLO DE EMPRESAS QUE UTILIZAN SOA Y MICROSERVICIOS**

Los microservicios son una arquitectura de desarrollo de software que ha cobrado popularidad en los últimos años. En contraste con las arquitecturas monolíticas convencionales, los microservicios dividen las aplicaciones en servicios independientes y distribuidos. A continuación, se presentan varios ejemplos de empresas que utilizan microservicios, así como sus **ventajas y desventajas**.

- **Netflix:** Esta plataforma de streaming adoptó los microservicios hace algunos años. La mayor parte de su arquitectura general se basa en esta metodología, lo que les permite mantener y escalar sus productos de manera más eficiente.
- **Amazon:** Otra empresa muy conocida que utiliza microservicios. Gracias a su modularidad y escalabilidad, pueden responder rápidamente a las necesidades del mercado.
- **Uber:** La aplicación de transporte que ha ganado popularidad en todo el mundo también ha adoptado los microservicios. Pueden agregar servicios según lo requieran y responder rápida y eficientemente a las demandas de sus usuarios.



### 1. SOA para empresas:

#### **Ventajas al usar SOA las Empresas**

- Ofrece servicios más versátiles en cuanto a su funcionalidad empresarial.
- Permite la coordinación con varios grupos para atender a las solicitudes de negocio.
- Facilita la integración de aplicaciones y sistemas legados.

### Desventajas al usar SOA las Empresas

- Requiere una mayor coordinación entre los diferentes grupos y servicios.
- Puede ser más complejo de implementar y mantener en comparación con los Microservicios.

Entre ellas tenemos:

#### a. Interbank:

##### ⇒ Ventajas:

- **Reusabilidad:** Interbank ha reutilizado componentes de software, reduciendo costos de desarrollo y tiempo de implementación.
- **Interoperabilidad:** La integración de sistemas bancarios internos y externos ha mejorado, permitiendo una mejor gestión de datos y servicios financieros.
- **Flexibilidad:** capacidad para adaptar y modificar servicios rápidamente sin afectar el funcionamiento general del sistema.

##### ⇒ Desventajas:

- **Complejidad:** La gestión de múltiples servicios puede ser compleja, requiriendo personal altamente capacitado y herramientas sofisticadas.
- **Costos iniciales:** La implementación de SOA puede ser costosa inicialmente debido a la necesidad de rediseñar sistemas y adquirir nuevas tecnologías.

#### b. Rimac Seguros:

##### ⇒ Ventajas:

- **Interoperabilidad:** integración efectiva de sistemas de seguros, mejorando la eficiencia operativa y la experiencia del cliente.
- **Reusabilidad:** Servicios de seguro comunes son reutilizados en diversas aplicaciones, reduciendo el tiempo y costo de desarrollo.
- **Escalabilidad:** Facilidad para escalar servicios individuales según sea necesario mejorando la capacidad de respuesta ante picos de demanda.

##### ⇒ Desventajas:

- **Rendimiento:** La comunicación entre servicios puede introducir latencia, afectando la rapidez en la respuesta del sistema.
- **Mantenimiento:** Requiere un esfuerzo continuo para mantener y actualizar los servicios, lo que puede ser costoso y laborioso.

## 2. Microservicios para empresas:

### Ventajas de Microservicios en las Empresas

- Permite dividir los sistemas en partes individuales, lo que facilita el tratamiento y abordaje de problemas de manera independiente sin afectar al resto.
- Cada microservicio se desarrolla, implementa y actualiza de manera independiente, lo que brinda mayor flexibilidad y agilidad en el desarrollo de aplicaciones.
- Están enfocados en desarrollarse en función de cada negocio o cliente, lo que los hace más sencillos y menos condicionados por la tecnología.

### Desventajas de Microservicios en la Empresas

- Cada microservicio consume más memoria y CPU debido a que tienen sus propios recursos y bases de datos.
- Requiere una inversión de tiempo inicial mayor para fragmentar los distintos microservicios e implementar la comunicación entre ellos.

Entre ellas tenemos:

#### a) **BBVA Perú:**

##### ⇒ **Ventajas:**

- **Escalabilidad:** Capacidad para escalar componentes individuales de manera eficiente según las demandas del mercado.
- **Despliegue Independiente:** Posibilidad de implementar y actualizar microservicios sin interrumpir otros servicios, facilitando el desarrollo continuo.
- **Tecnología Independiente:** Permite utilizar las tecnologías mas adecuadas para cada microservicio, optimizando el rendimiento y la eficiencia.

##### ⇒ **Desventajas:**

- **Complejidad en la Gestion:** Adminisitrar numerosos microservicios puede ser un desafio significativo, especialmente en terminos de monitoreo y orquestación.
- **Latencia:** La comunicación entre microservicios puede agregar latencia, afectando el rendimiento general del sistema.
- **Consistencia de Datos:** Mantener la consistencia de datos en una arquitectura distribuida puede ser complicado, requiriendo estrategias avanzadas de manejo de datos.

#### b) **Latam Airlines Perú:**

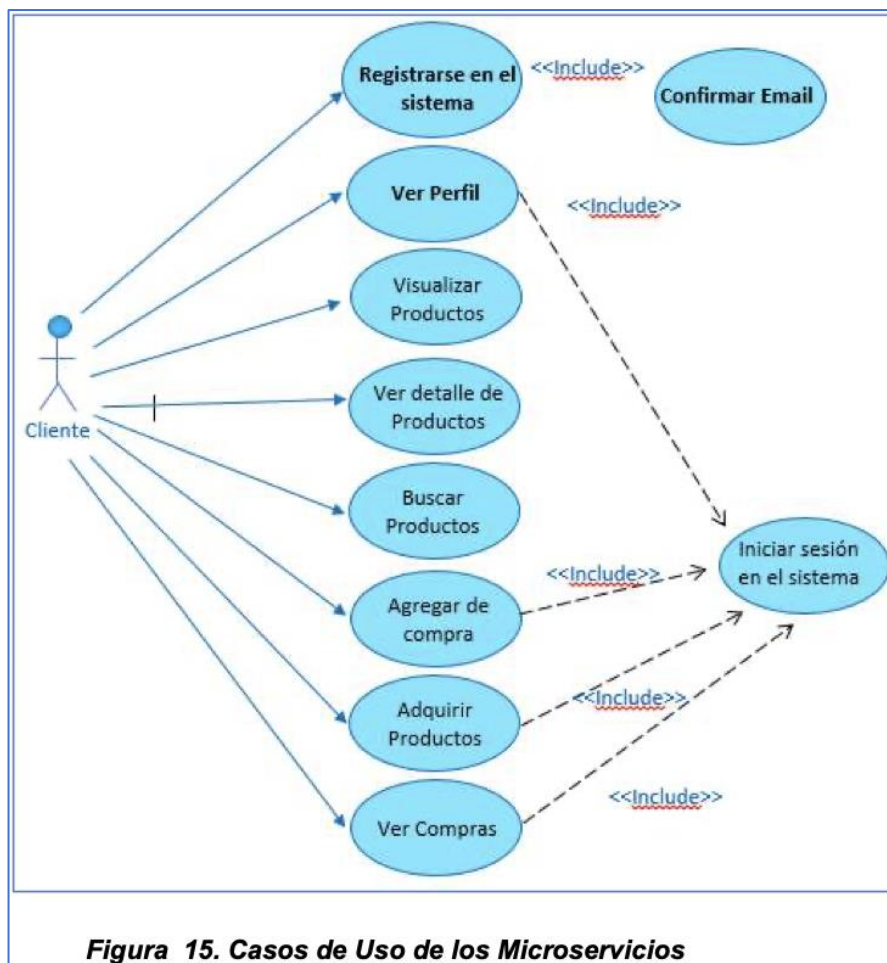
##### ⇒ **Ventajas:**

- **Agilidad:** Capacidad para responder rapidamente a cambios en el mercado y a nuevas oportunidades de negocio.
- **Resiliencia:** Los fallos en un microservicio no afectan a todo el sistema, mejorando la resiliencia general.

- **Desarrollo independiente:** Los equipos pueden trabajar en diferentes microservicios de manera independiente, acelerando el desarrollo.

⇒ **Desventajas:**

- **Desafíos en la Consistencia de Datos:** Asegurar que todos los microservicios tengan acceso a datos consistentes puede ser complejo.
- **Gestión de la Red:** La comunicación intensiva entre microservicios puede sobrecargar la red y requerir una gestión avanzada.
- **Seguridad:** Implementar y mantener la seguridad en cada microservicio puede ser complicado y propenso a errores.



## Diferencias entre la SOA y los microservicios

Función	SOA	Microservicios
Estilo arquitectónico	<ul style="list-style-type: none"> <li>De alto nivel (grano grueso), centralizado</li> </ul>	<ul style="list-style-type: none"> <li>Detallado (de grano fino), <a href="#">sistema distribuido</a></li> <li>Gestión de datos descentralizada</li> </ul>
Granularidad del servicio	<ul style="list-style-type: none"> <li>Servicios más grandes y amplios</li> </ul>	<ul style="list-style-type: none"> <li>Servicios más pequeños y selectivos</li> </ul>
Independencia	<ul style="list-style-type: none"> <li>Los servicios son interdependientes</li> <li>Puede compartir una base de datos para almacenar datos</li> </ul>	<ul style="list-style-type: none"> <li>Los servicios son muy independientes</li> <li>Desacoplado y autónomo</li> </ul>
Comunicación	<ul style="list-style-type: none"> <li>Sincrónico, a menudo orientado a los mensajes</li> <li>Utiliza datos compartidos</li> </ul>	<ul style="list-style-type: none"> <li>Asincrónico, a menudo basado en REST</li> <li>Evita el uso compartido de datos</li> </ul>
Almacenamiento de datos	<ul style="list-style-type: none"> <li>Gestión de datos centralizada</li> <li>Los servicios comparten la base de datos</li> </ul>	<ul style="list-style-type: none"> <li>Gestión de datos distribuida (descentralizada)</li> <li>Cada servicio es responsable de la gestión de sus propios datos</li> </ul>

Escalabilidad	<ul style="list-style-type: none"> <li>Escalado horizontal</li> <li>Escalar servicios específicos puede ser complicado cuando hay recursos compartidos y la comunicación está centralizada</li> </ul>	<ul style="list-style-type: none"> <li>Escalado horizontal y vertical</li> <li>Escalado más granular y selectivo, ya que los servicios funcionan de forma independiente</li> </ul>
Implementación	<ul style="list-style-type: none"> <li>Por lo general, implica la implementación de toda la aplicación como una sola unidad</li> </ul>	<ul style="list-style-type: none"> <li>Cada servicio se implementa y escala de forma independiente</li> <li>Facilita la <a href="#">entrega continua</a></li> </ul>
Acoplamiento	<ul style="list-style-type: none"> <li>Los servicios muestran cierto grado de acoplamiento por el uso compartido de recursos y la centralización de la comunicación</li> </ul>	<ul style="list-style-type: none"> <li>Acoplamiento flexible con dependencias mínimas entre servicios</li> </ul>



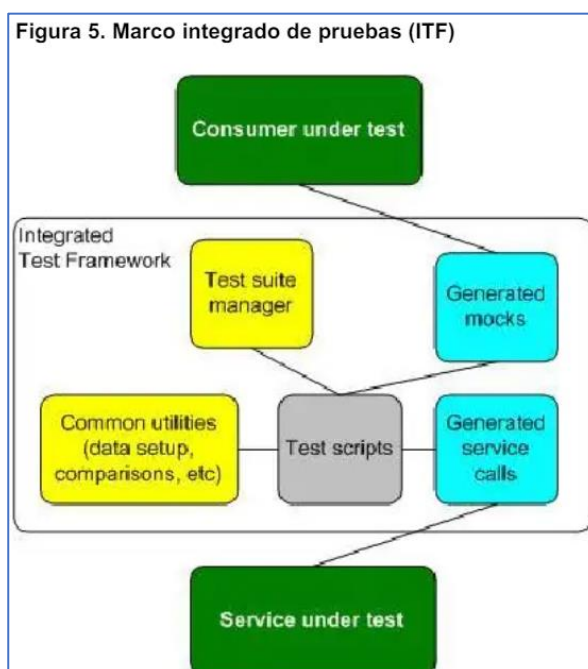
## **Propuesta de buenas prácticas para la implementación de SOA y Microservicios**

### **Adoptar un marco de pruebas de servicios**

A fin de automatizar y mantener suites de pruebas de integración de servicios, existen ciertas capacidades comunes que se deben desarrollar y reutilizar. Entre ellas se encuentran:

1. La capacidad de producir agentes de prueba en ausencia de la interfaz de usuario de una aplicación.
2. Generación de mensajes de prueba, basados en la descripción del servicio (WSDL).
3. Variación de datos de entrada, usando una tabla de datos.
4. Scripts de desmontaje y configuración de datos.
5. Datos de salida de informes de pruebas.
6. Definición de resultados esperados.
7. Ejecución de pruebas en cada nivel integrado de la pila (por lo general, a través de entorno de prueba unitaria).
8. Emulación de servicios externos (falsos).
9. Inspección y validación de mensajes de servicio de aplicaciones consumidoras.
10. Envío de múltiples mensajes de prueba a través de subprocessos separados.

Estas capacidades vienen empaquetadas en un marco integrado de pruebas (ITF, por su sigla en inglés). Por lo general, el marco está compuesto de herramientas comerciales o de código abierto combinadas con personalizaciones para satisfacer las necesidades específicas del entorno. En lugar de implementar estas capacidades para cada uno de los servicios, es aconsejable mantener el ITF como un activo individual reutilizable que contenga las utilidades, herramientas y scripts más comunes. Se recomienda que el marco esté basado en una herramienta de pruebas funcionales.





## REFERENCIAS BIBLIOGRAFICAS ISO 690-1020

- ¿Qué es la arquitectura orientada a los servicios (SOA)? (2024, 10 de mayo). RED HAT. <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture>
- SOA frente a microservicios: qué es lo mejor para tu empresa. ATlassian [en línea]. 10 de marzo de 2024. Disponible en: <https://www.atlassian.com/es/microservices/microservices-architecture/soa-vs-microservices>
- 9 buenas prácticas para utilizar los Microservices en tus proyectos. HOSTGATOR [en línea]. 8 de abril de 2024. Disponible en: <https://www.hostgator.mx/blog/microservices-en-tus-proyectos/>
- VALENCIA, ALEX. Buenas Practicas SOA. SCRIBD [en línea]. 25 de mayo de 2023. Disponible en: <https://es.scribd.com/document/229038149/Buenas-Practicas-SOA>

Alojar su producto software en GitHub, adjuntar enlace en informe.

⇒ [https://github.com/WendyJacobo/ARQUITECTURA\\_SOFT.git](https://github.com/WendyJacobo/ARQUITECTURA_SOFT.git)

