

# **ARQUITECTURA WEB: SOA VS. MICROSERVICES, APLICACIONES Y DIFERENCIAS**

**AUTOR: MAX JOSÉ BERMÚDEZ LEÓN  
DICIEMBRE: 2020**



**San Marcos**

## Introducción

SOA (Arquitectura Orientada a Servicios), es una arquitectura ideal para agilizar y completar de forma integral los sistemas de información y procesos de negocio de integración de los enfoques más tradicionales, esto ha hecho que se convierta en una tendencia de diseño actual y vigente. Por lo anterior capacitarse en SOA es una arquitectura actual que hay que conocer, por esto se ha diseñado este curso con el fin de no hacer un enfoque demasiado técnico en de las diferentes tecnologías, sino en proporcionar una visión general de las tecnologías y soluciones existentes, así mismo de cómo combinarlas dentro de una implementación SOA, permitiendo que quienes tomen el curso tengan los conocimientos adecuadas para el desarrollo de despliegues SOA desde su diseño y modelado.



## Tabla de contenido

Introducción.....	1
Arquitectura web: SOA vs. Microservices, Aplicaciones y diferencias .....	3
¿Qué es SOA?.....	3
Soa define las siguientes capas de software: .....	4
Servicios .....	4
Arquitectura SOA .....	5
Beneficios de SOA .....	6
Proceso de transición a la Arquitectura Orientada a Servicios (SOA).....	6
¿Qué es Microservices? .....	9
Características y aplicaciones de Microservices: .....	11
Diferencias entre SOA y Microservices .....	11
Conclusiones y recomendaciones.....	13
Referencias bibliográficas .....	14

**PARA COMPRENDER QUE SON LOS MICROSERVICIOS ES FUNDAMENTAL ENTENDER QUE SON LOS SERVICIOS (SOA), PUES ES LA BASE EN LA QUE SE SUSTENTAN LOS MICROSERVICIOS. TANTO SOA COMO MICROSERVICIOS SON “TIPOS DE ARQUITECTURA DE SOFTWARE” QUE SE BASAN EN SERVICIOS, SIN EMBARGO, SOA ES UNA ARQUITECTURA DE MÁS ALTO NIVEL, PUES NOS INVITA A CONSTRUIR APLICACIONES BASADA EN SERVICIOS, DONDE UN SERVICIO ES LA UNIDAD MÁS PEQUEÑA DE UN SOFTWARE, POR OTRO LADO, LOS MICROSERVICIOS TAMBIÉN NOS INVITA A CREAR SERVICIOS, LA DIFERENCIA RADICA EN QUE LOS MICROSERVICIOS ESTÁN DISEÑADOS EN LA CONSTRUCCIÓN DE PEQUEÑOS SERVICIOS O INCLUSO PROGRAMAS QUE CUMPLAN UNA FUNCIONALIDAD MUY PRECISA Y MUY PEQUEÑA, LA CUAL PUEDA SER DESACOPLADA DEL RESTO DE APLICACIONES Y FUNCIONAR DE UNA FORMA TOTALMENTE AUTÓNOMOS DEL RESTO DE APLICACIONES.**

## Arquitectura web: SOA vs. Microservices, Aplicaciones y diferencias

### ¿Qué es SOA?

La arquitectura orientada a servicios de cliente (en inglés Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

La integración de las distintas herramientas de la compañía, conlleva la definición clara de los servicios para dar soporte a las aplicaciones entre sí, brindando una forma clara de hacerlo.

SOA O LA ARQUITECTURA ORIENTADA A SERVICIOS NOS VA A PERMITIR LA CREACIÓN DE SISTEMAS DE INFORMACIÓN QUE SE APROXIMEN A LOS CRITERIOS DE LA EMPRESA:

- La arquitectura orientada a servicios de cliente (en inglés Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.
- Permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual

facilita la interacción entre diferentes sistemas propios o de terceros.

Soa define las siguientes capas de software:

- Aplicaciones básicas: Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.
- De exposición de funcionalidades: Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como servicios web).
- De integración de servicios: Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.
- De composición de procesos: Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio.
- De entrega: Donde los servicios son desplegados a los usuarios finales.

SOA proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y puede dar soporte a las actividades de integración y consolidación.

Por lo tanto, si tenemos generada una arquitectura SOA, lo que nos plantea es la posibilidad de facilitar la integración entre las distintas aplicaciones para conseguir maximizar el beneficio obtenido con nuestra infraestructura.

## Servicios

Para poder llegar a la evolución completa, se requiere de un diseño estructurado que permita a pequeños servicios, ofertados por individuos, coordinarse para ofrecer un servicio conjunto y de mayor envergadura.

Así, en un trabajo de manufactura, muchas veces no interviene una única persona, sino que por ejemplo si hablamos de una silla de madera, entrará quien tale el árbol, quien extraiga la madera, quien la corte para prepararla, quien realice el encolado y quien la barnice. Todos ellos están realizando un servicio, pero al hacerlo de forma conjunta nos ofrecen un servicio mayor o más completo, como será la fabricación de una silla.

Para que todos estos pasos o servicios puedan dar lugar a un servicio o producto final se

requiere de ese diseño estructurado en el que cada uno sabe qué tarea ha de realizar y que parámetros de salida son necesarios y válidos.

Por lo tanto, el diseño de esa arquitectura orientada a servicios es uno de los elementos principales en cualquier estructura orientada a servicios.

### Arquitectura SOA

La computación orientada a servicios representa una nueva generación de sistemas que utilizan una plataforma de desarrollo que involucra múltiples componentes, pero que tiene sus propios elementos de diseño, con patrones de lenguajes, diseños de arquitectura, conceptos, tecnologías y entornos de relación de los sistemas.

SOA o arquitectura orientada a servicios establece un modelo de arquitectura que ayuda a mejorar la eficiencia, efectividad y productividad de una empresa, posicionando los servicios como lo primero que debemos de tener en cuenta cuando queremos presentar una lógica de solución de un problema, o para la consecución de los objetivos estratégicos de la compañía, asociado a la programación orientada a servicios.

Dentro de esta arquitectura por lo tanto podemos decir que SOA estará formada por la combinación de tecnologías, productos, API's (Interfaces relacionales de aplicaciones) extensiones de la tecnología de soporte y otras partes varias.

Sin embargo, muchas veces el SOA viene asociado a la introducción de nuevas herramientas o tecnologías dentro de la empresa que den soporte a la creación, ejecución, y evolución de las soluciones dentro de la misma.

Por lo tanto, una solución orientada a servicios dentro de una empresa dará como resultado una arquitectura orientada a servicios, basándose en unos principios de diseño orientado al servicio.

Cada servicio tendrá su propio conjunto de capacidades de acuerdo al entorno donde se



encuentre. Estas capacidades serán susceptibles de ser invocadas desde fuera (por otros servicios) para lo que se deberán de publicitar de manera expresa la manera de invocar estas capacidades, esto se puede hacer de manera equivalente a lo que sería una API.

## Beneficios de SOA

El SOA hace un especial hincapié en la reutilización de sus componentes, la idea que tenemos detrás de estas arquitecturas o tecnologías orientadas a servicios es que podamos reaprovechar los desarrollos de servicios que se realizan en un momento dado para la creación de otros servicios y por tanto tener mayores servicios.

Los servicios que creamos o desarrollamos deben seguir una serie de premisas, o estar desarrollado de una forma tal que seamos capaces de identificar quién y cómo podrá acceder a los mismos, para que así nuestro catálogo sea comprensible y reutilizable.

La consecuencia por lo tanto es que tendemos a alejarnos de las infraestructuras basadas en silos, en entornos únicos que funcionaban de manera completamente independiente y nos movemos a los sistemas que están formados por la suma de múltiples servicios que dan como resultado un servicio final que nos permita dar respuesta a las necesidades de la empresa en cuestión.

A medida que crezcamos sobre este método de creación e implantación de sistemas, los anteriores conceptos basados en aplicaciones y en entornos de silos que funcionan de manera independiente irán poco a poco desapareciendo para dar paso a arquitecturas basadas en servicios.

Las aplicaciones en los entornos de SOA carecen de sentido individual, ya que o bien es un conjunto de servicios, o bien es en sí mismo un servicio para otro sistema mayor.

## Proceso de transición a la Arquitectura Orientada a Servicios (SOA)

Para abordar un proyecto SOA se debe partir de una concepción estratégica del negocio, alinear todos los elementos instrumentales –sistemas, procesos, información, aplicaciones

hacia el núcleo de negocio de la Administración, que son los servicios a la ciudadanía, requiere un enfoque global.

Si se desea llevar a cabo el proceso de transición a SOA, los administradores y los desarrolladores deben tener en cuenta los siguientes aspectos:

La Arquitectura Orientada a Servicios no es nueva, por lo que será vital adquirir conocimiento suficiente sobre la SOA y los Web Services antes de estructurar el plan.

SOA es mucho más que un software de despliegue. Se requiere de un análisis de las técnicas de diseño y desarrollo para avanzar con garantías de éxito.

El proceso de transición a SOA debe realizarse de forma gradual y teniendo en cuenta que implica un gran cambio en la forma de trabajar para todos los empleados.

Las entidades que ya trabajen con SOA pero busquen optimizar sus resultados con Data Services, deberán tener en cuenta que se debe:

- Ser exigente con el servicio escogido, evitando extremos y persiguiendo la coherencia.
- Entender los servicios como algo limitado y no como una aplicación completa.
- Aplicar la máxima simplicidad a la hora de diseñar.
- Garantizar la alta disponibilidad y escalabilidad de los servicios.
- Esta optimización es la vía más indicada para superar las limitaciones que adolecen a un proyecto SOA, a través de la visualización de datos que ayuda a evitar:
- Falta de disponibilidad del servicio dependiente: que se da cuando estos servicios aún no están implementados y resulta en tiempos de inactividad o en la construcción de componentes redundantes.
- Falta de disponibilidad de recursos: puede suceder cuando los recursos se tienen que compartir entre distintos equipos de desarrollo.
- Restricciones de tiempo: la variable indefectiblemente asociada a todo proyecto y que marca una de las limitaciones más importantes.
- Cambio de comportamiento del servicio dependiente: que, no sólo invalida los flujos

de trabajo presentes, sino que también incide en la consistencia de los datos.

Ejemplo:

Si llegamos a una compañía que ha ido creciendo de forma clásica a través tanto del crecimiento orgánico de la misma como por la adquisición de diferentes compañías, en multitud de ocasiones nos podemos encontrar entornos complejos de manejar.

Así, si vemos una estructura en la cual priman aplicaciones propietarias, diseñadas en un momento dado para dar respuesta a los problemas de la compañía en ese momento, sumado a múltiples cambios o adquisiciones de diferentes compañías a lo largo de la historia (podemos pensar en un gran banco, que crece de manera propia y al mismo tiempo por la adquisición de otros bancos en otros países).

Si planteamos su situación en un momento dado, tendremos un conjunto de aplicaciones y sistemas, que, si bien dan servicio puntual a cada unidad en el momento de su creación, en el momento de pasar por la integración con otros sistemas, perdemos esta facilidad.

En estos casos, si no disponemos de una arquitectura orientada a servicio, cada vez que se quiera ofertar un nuevo servicio, se ha de desarrollar una aplicación a medida para cada servicio, con los costes que ello tiene, no solo desde el punto de vista económico para el desarrollo, sino desde el punto de vista estratégico por el tiempo necesario para llevar a cabo este desarrollo.

Si por el otro lado, nos encontramos en una empresa con una estructura y arquitectura orientada a servicios, donde los desarrollos de cada servicio, sistema o aplicación se ha realizado pensando que este se debe de integrar con otros, por ejemplo, siguiendo en el entorno bancario, con las aplicaciones preexistentes, con herramientas de uso, con los diferentes departamentos, etc.

Tendremos distintos módulos que podremos reutilizar para lanzar el nuevo servicio, pues en muchos aspectos tendrá elementos comunes con los servicios ya existentes dentro del sistema.

Si lo que hablamos es de integrar otra entidad recién adquirida, en el caso de la primera adquisición, posiblemente deberemos desarrollar una aplicación a medida para poder acceder a los datos del sistema, pero según vamos reutilizando la entrada de datos a nuestros entornos será siempre igual, por lo que lo único que deberíamos rehacer será la conexión con los datos antiguos.

Esto cada vez nos dará más servicios a reutilizar y por lo tanto menos entornos a desarrollar, por lo que podremos necesitar menos tiempo en la integración o creación de nuevos servicios y tendremos mayores beneficios.

Estos cambios estructurales pueden ayudar nuevamente a cambiar la sensación que existe en muchas organizaciones de que los departamentos de TI son frenos para las compañías y muchas veces un gasto necesario, para ser ayuda en transformar el departamento en motores de innovación y de desarrollo dentro de la compañía al ser capaces de realizar y ofrecer servicios cada vez con mayor rapidez.

### ¿Qué es Microservices?

Por su parte, Microservices (microservicios) entiende el desarrollo de una aplicación como la implementación de un conjunto de pequeños servicios. Cada uno de estos pequeños servicios trabajan al mismo tiempo, se ejecutan en su propio proceso y se mantienen en constante comunicación entre sí, utilizando mecanismos más ligeros respecto a SOA.

Podríamos decir que los Microservices se representan, en cierta forma, como un paso más allá de SOA, por tanto, pueden entenderse como una evolución de la arquitectura orientada a servicios. Con Microservices podemos desarrollar e implementar aplicaciones de software, aplicaciones web o móviles en base a una arquitectura que se basa en conjuntos de servicios

independientes. De hecho, los Microservices pueden ser calificados como tal porque cada servicio se empaqueta y se despliega de forma totalmente independiente. Están diseñados como servicios altamente autónomos que no dependen de una comunicación sincronizada y centralizada.

Los microservicios se ejecutan de forma independiente y cumplen una única función específica como pueden ser: administrar a los diferentes usuarios de una web; permitir que el carrito de compra en un comercio online funcione, así como el motor de búsqueda o la función de iniciar sesión en una red social asociada al desarrollo, entre muchas otras.

Al ser servicios que se desarrollan de forma totalmente independiente, pueden ser escritos en lenguajes de programación diferentes, así como utilizar distintas bases de datos. Los microservicios utilizan API de HTTP, REST o Thrift para comunicarse entre sí y la administración del sistema está prácticamente descentralizada en su totalidad. Estos microservicios están soportados por una maquinaria totalmente automatizada y se construyen con independencia de despliegue.

Los Microservices son aptos para ser utilizados por empresas de diferentes tamaños, ya que hablamos de soluciones escalables. Cada uno de estos Microservicios poseen una firma asociada limitada de un módulo lo que permite, como adelantábamos, que diferentes servicios puedan ser escritos en distintos lenguajes de programación, así como ser administrados por equipos de trabajo diferentes.

Un poco de historia: pero, ¿de donde proviene el término de Microservicios? La palabra se acuñó en un taller de arquitectos de software celebrado en 2011 en Venecia. Se empleó por primera vez para identificar un estilo de arquitectura de software que los asistentes al taller habían venido explorando y utilizando recientemente. Pero, aunque el término se acuñase en 2011, gigantes tecnológicos como Microsoft o Amazon ya venían utilizando los microservicios, también denominados “SOA de grano fino”, antes de que este término se generalizase.



### Características y aplicaciones de Microservices:

- Trabajan de forma autónoma e independiente.
- No es necesario que participen en transacciones distribuidas.
- Presentan un alcance limitado.
- No dependen de la sincronía con otros servicios.
- No han sido creados para formar parte de composiciones complejas de servicios.
- Se gobiernan de forma descentralizada y pueden emplear lenguajes de programación diferentes.
- Utilizan almacenamiento de datos propios.
- Actúan con mecanismos de comunicación basados en mensajes.
- La arquitectura Microservices está vinculada al movimiento y filosofía DevOps.
- Emplea sistemas de mensajería menos elaborados y simples respecto a SOA.
- Emplea protocolos ligeros como HTTP, REST o API de Thrift.
- Cada microservicio puede tener un almacenamiento de datos independiente.
- Promueve la colaboración de equipos.

### Diferencias entre SOA y Microservices

Ambas arquitecturas presentan más similitudes que diferencias. De hecho, Martin Fowler, ingeniero de software y pionero en el mundo de los microservicios, asegura que deberíamos pensar en SOA como un súper-conjunto de microservicios. Por tanto, podríamos decir que ambas arquitecturas se complementan y pueden entenderse como una evolución.

No hay una arquitectura mejor que otra, la elección depende del objetivo y propósito del proyecto a desarrollar. En términos generales, SOA suele ser más recomendable para organizaciones más grandes que trabajan en base a la integración de aplicaciones. Los Microservices se caracterizan por ser sistemas más pequeños y fraccionados que permiten un mayor control al desarrollador de software.

Una diferenciación importante entre ambas arquitecturas es el tamaño y alcance de cada



una de ellas. En el sistema de microservicios, cada servicio presenta una única funcionalidad que ejecuta a la perfección. Por su parte, SOA incluye más funcionalidades y es capaz de desarrollar sistemas completos.

Con Microservices los desarrolladores tienen más flexibilidad al no depender del sistema global; con SOA los equipos están más “encorsetados” ya que deben conocer el mecanismo de comunicación común para todos los servicios, lo que no ocurre con Microservices.

Con Microservices cada uno de los servicios pueden ser implementados de forma independiente, al contrario de lo que ocurre con SOA que trabaja en base a la integración de servicios.

En cuanto a los fallos en el sistema, si se producen en SOA afectarían a todo el sistema, al contrario que con Microservices, arquitectura con la que el fallo afectaría a un único punto y no a la totalidad del sistema.

Por último, si nos fijamos en la operatividad, SOA utiliza múltiples protocolos heterogéneos por lo que permite integrar en una misma aplicación varios sistemas que emplean diferentes protocolos. Microservices simplifica el patrón de arquitectura y reduce el número de opciones de integración.

En definitiva, ambas arquitecturas se emplean en la actualidad y otorgan buenos resultados, la elección entre una u otra depende del proyecto que queramos desarrollar.

## Conclusiones y recomendaciones

Algunos de los aspectos aprendidos y que de gran peso es la base de datos su definición, requerimiento, ventajas y características donde podemos decir que la base de datos: Es una colección de datos o información usados para dar servicios a muchas aplicaciones al mismo tiempo.

En cuanto al requerimiento podemos decir que cumple las mismas tareas de análisis que del software y tiene como característica relacionar la información como vía organización y asociación donde la base de datos tiene una ventaja que es utilizar la plataforma para el desarrollo del sistema de aplicación en las organizaciones.

Otro aspecto importante sería el diseño y creación de la base de datos, donde existen distintos modos de organizar la información y representar las relaciones entre por datos los tres modelos lógicos principales dentro de una base de datos son el jerárquico, de redes y el relacional, los cuales tiene ciertas ventajas de procesamiento y de negocios.

## Referencias bibliográficas

- ECITYCLIC (2020). ¿QUÉ ES SOA O ARQUITECTURA ORIENTADA A SERVICIOS? Sitio de descarga de la información: <https://www.ecitycllic.com/es/noticias/que-es-soa-o-arquitectura-orientada-a-servicios>
- El Blog Ceupe (2020).QUE ES SOA. Sitio de descarga de la información: <https://www.ceupe.com/blog/que-es-soa.html>
- Juan Carlos Cantó (BILIB). (2018). Arquitectura web. Sitio de descarga de la información: <https://www.bilib.es/actualidad/blog/noticia/articulo/arquitectura-web-soa-vs-microservices-aplicaciones-y-diferencias/>



[www.usanmarcos.ac.cr](http://www.usanmarcos.ac.cr)

San José, Costa Rica