



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
CARRERA DE INGENIERIA EN SOFTWARE

**Trabajo de:**

Aplicaciones Web Avanzadas

**Nombre:**

Christian Israel Fernández

**Tema:**

**SOA vs Microservicios**

**Que es SOAP**

Las siglas SOAP hacen referencia a Object Access Protocol, este protocolo fue creado por David Winer en 1998. SOAP es un formato de mensaje XML, el cual es utilizado en interacciones de servicios web. Mediante este protocolo se podía realizar peticiones RPC o remote procedure calls. Los mensajes SOAP se envían sobre HTTP habitualmente, sin embargo, esta no es una ley ya que se pueden utilizar otros protocolos, es decir, que un cliente o servidor podría realizar peticiones mediante http a un servidor web. Con el paso del tiempo el proyecto iniciado por David Winer interesó a importantes multinacionales entre las que se destacan IBM y Microsoft y de este interés por XML-RPC se desarrolló SOAP" [1].

En la actualidad existen dos versiones de SOAP en uso común: SOAP 1.1 y SOAP 1.2. Estas versiones están soportadas en IBM Integration Bus. SOAP está definido en los siguientes documentos emitidos por World Wide Web Consortium (W3C) [2]:

- Simple Object Access Protocol (SOAP) 1.1 (nota de W3C).
- SOAP Versión 1.2 Part 0: Primer (recomendación de W3C).
- SOAP Versión 1.2 Part 1: Messaging Framework (recomendación de W3C).
- SOAP Versión 1.2 Part 2: Adjuncts (recomendación de W3C).

SOAP proporciona un mecanismo estándar de empaquetar un mensaje. Un mensaje SOAP se compone de un sobre que contiene el cuerpo del mensaje y cualquier información de cabecera que se utiliza para describir el mensaje.

**Ventajas**

- No está asociado con ningún lenguaje
- No se encuentra fuertemente asociado a ningún protocolo de transporte



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
CARRERA DE INGENIERIA EN SOFTWARE

---

- No está atado a ninguna infraestructura de objeto distribuido
- Aprovecha los estándares existentes en la industria
- Permite la interoperabilidad entre múltiples entornos

#### Desventajas

- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear.
- Existe poca información de servicios web para algunos lenguajes de programación.
- Relativamente nuevo, algunos estándares no definidos.
- Dependencia de la disponibilidad de servidores y comunicaciones.
- No es la solución a todos los problemas.

#### Que son los microservicios

La arquitectura de microservicios o mejor conocido solo como microservicios, es un sistema particular de desarrollo de software, el cual ha aumentado en popularidad en los últimos años. Gracias a su sencilla escalabilidad, este método de arquitectura es considerado como apropiado cuando se busca procurar la compatibilidad con un amplio sector de diferentes plataformas o simplemente cuando no se tiene una certeza hacia qué tipo de dispositivos estamos orientando el trabajo.

Al momento de utilizar microservicios no existe una definición o estándar formal, sin embargo, existen ciertas características que ayudan a trabajar en esta arquitectura. El desarrollo de un proyecto basada en esta arquitectura básicamente es una aplicación o herramienta la cual trabaja mediante un conjunto de diversos servicios independientes que se despliegan según se vayan necesitando. Por tanto, tendremos una aplicación modular a base de “pequeñas piezas”, que podremos ir ampliando o reduciendo a medida que se requiera [3].

Pero para comprender la arquitectura basada en microservicios, lo mejor es considerar su opuesto, su némesis, la arquitectura monolítica. En ésta, la aplicación se desarrolla como una única unidad que no necesitará de ningún componente externo para funcionar. El problema de las aplicaciones monolíticas es que todos los “ciclos de cambio” están vinculados unos a otros, por lo que la más mínima modificación en una remota sección de la app conllevaría a la creación y despliegue de una versión completamente nueva, con el gasto de recursos correspondiente [4].

#### Ventajas

- Otorga a los desarrolladores libertad de desarrollar y desplegar servicios de forma independiente.
- Un microservicio se puede desarrollar con un equipo de trabajo mínimo.



**ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
CARRERA DE INGENIERIA EN SOFTWARE**

- 
- Se pueden usar diferentes lenguajes de programación en diferentes módulos.
  - Fácil integración y despliegue automático.
  - Fácil de entender y modificar, por lo que la integración de nuevos miembros al equipo de desarrollo será muy rápida.
  - El uso de contenedores hará el desarrollo y despliegue de la app mucho más rápido.
  - Funcionalidad modular, con lo que la modificación de un módulo no afectará al funcionamiento del resto.

### **Desventajas**

- Las pruebas o testeos pueden resultar complicados debido al despliegue distribuido.
- Un gran número de servicios puede dar lugar a grandes bloques de información que gestionar.
- Será labor de los desarrolladores lidiar con aspectos como la latencia de la red, tolerancia a fallos, balanceo de carga, cantidad de formatos admitidos, etc.
- Sistema distribuido puede llegar a significar doble trabajo.
- Si se cuenta con un gran número de servicios, integrarlos y gestionarlos puede resultar muy complejo.
- Esta tecnología suele incurrir en un alto consumo de memoria.
- Fragmentar una aplicación en diferentes microservicios puede llevar muchas horas de planificación (y casi podría considerarse un arte).

### **Caso Groupon**

Esta popular empresa de comercio electrónico conecta a los consumidores con los comerciantes locales. La empresa ofrece obsequios con descuento y cupones virtuales que permiten a los compradores probar nuevos servicios en su área. Groupon tiene casi 48,1 millones de clientes activos y ofrece servicios en 28 países y más de 500 ciudades en todo el mundo [5].

### **¿Por qué NodeJS?**

En 2012, la empresa se dio cuenta de que la pila de tecnología anterior (Ruby on Rails, Java y Php) se había vuelto obsoleta y era bastante complicado mantener la aplicación. La tarea más simple tomó meses para implementar y ese se convierte en el principal motivo de reemplazo. NodeJS fue una gran solución para su problema y tomó casi un año superar todos los problemas que enfrentaban en el proceso para finalmente terminar la transición [5].

### **Desafío**

- Con evidencia que muestra que los retrasos en el alcance impactan en las conversiones, Groupon debe garantizar la entrega de la campaña de millones de correos electrónicos diarios y notificaciones automáticas a los usuarios en más de 500 mercados en 28 países.



**ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
CARRERA DE INGENIERIA EN SOFTWARE**

---

- Mantenga una experiencia de usuario resiliente y con capacidad de respuesta, especialmente cuando se escale para manejar grandes picos de demanda (2-3 horas de aumento de tráfico de 7 a 10 veces es común) durante las vacaciones y eventos especiales (es decir, Black Friday, Cyber Monday, Navidad, Super Bowl ).
- La arquitectura de aplicación monolítica anterior era dolorosa y lenta de actualizar, difícil y costosa de escalar a la luz de la creciente demanda, e incapaz de manejar la concurrencia y el paralelismo sin encontrarse con errores de programación y vulnerabilidades.

### **Solución**

- Se reemplazó el monolito heredado con una arquitectura de microservicios reactivos basada en Akka y Play Framework.
- Compromiso personalizado predecible y oportuno garantizado con millones de clientes en el momento justo mediante la integración de varias fuentes de datos.
- La arquitectura de microservicios reactivos aumentó la productividad de los desarrolladores para las implementaciones.

### **Beneficios**

- Mayor rendimiento en varias máquinas virtuales implementadas en muy pocos hosts físicos que atienden más de 600 000 solicitudes por minuto.
- El flujo de trabajo de enriquecimiento de big data se redujo de 30 a 10 minutos, utilizando un 25 % menos de recursos informáticos, lo que permitió un compromiso más rápido con millones de clientes en el momento justo.
- Entrega de servicio estable capaz de escalar para los picos de demanda de 7 a 10 veces durante eventos especiales, sin enfrentar ningún tiempo de inactividad en más de 3 años.



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA DE SISTEMAS  
CARRERA DE INGENIERÍA EN SOFTWARE

---

## Referencias

- [1] "SOAP (Simple Object Access Protocol)." <https://desarrolloweb.com/articulos/1557.php> (accessed Feb. 26, 2022).
- [2] "¿Qué es SOAP? - Documentación de IBM." <https://www.ibm.com/docs/es/integration-bus/10.0?topic=ssmkhh-10-0-0-com-ibm-etools-mft-doc-ac55770--htm> (accessed Feb. 26, 2022).
- [3] "¿Qué son y para qué sirven los microservicios?" <https://www.redhat.com/es/topics/microservices> (accessed Mar. 06, 2022).
- [4] "► De una arquitectura tradicional a una arquitectura microservicios - Blog de Hiberus Tecnología." <https://www.hiberus.com/crecemos-contigo/de-una-arquitectura-tradicional-a-microservicios/> (accessed Mar. 06, 2022).
- [5] "Estudio de caso: cómo Groupon escala las ofertas personalizadas a 48 millones de clientes a tiempo - FastnForward Blog." <https://fastnforward.blog/case-study-how-groupon-scales-personalised-offers-to-48-million-customers-on-time/> (accessed Mar. 06, 2022).