

## ¿Dónde es útil SOAP?

Un diseño basado en SOAP es adecuado cuando:

- Se establece un contrato formal para la descripción de la interfaz que el servicio ofrece. El lenguaje de Descripción de Servicios Web (WSDL), como ya sabemos, permite describir con detalles el servicio Web.
- La arquitectura debe abordar requerimientos complejos no funcionales. Muchas especificaciones de servicios Web abordan tales requisitos y establecen un vocabulario común para ellos. Algunos ejemplos incluyen transacciones, seguridad, direccionamiento, ... La mayoría de aplicaciones del mundo real se comportan por encima de las operaciones CRUD y requieren mantener información contextual y el estado conversacional. Con la aproximación REST, abordar este tipo de arquitecturas resulta más complicado.
- La arquitectura necesita manejar procesado asíncrono e invocación. En estos casos, la infraestructura proporcionada por estándares como WSRM y APIs como JAX-WS junto con la asincronía por el lado del cliente nos permitirán el soporte de estas características.

## ¿Qué podemos concluir de todo este debate?

Amazon posee ambos estilos de uso de sus servicios Web. Pero el 85% de sus clientes prefieren la interfaz REST. A pesar de la promoción que las empresas han invertido para ensalzar a SOAP, parece que es evidente que los desarrolladores prefieren, en algunos casos, la aproximación más sencilla: REST.

Todo lo visto a lo largo del documento como en el párrafo anterior, parece que nos da pistas para el futuro éxito de REST. Aunque, todos sabemos que en el mundo de la tecnología no siempre acaba triunfando la tecnología mejor, recuérdese el caso de VHS vs BetaMax.

Lo que está claro es que falta una pieza en la Web. La comunicación Hombre – Máquina parece que funciona, pero la comunicación Máquina – Máquina sigue siendo un reto. Recientemente, se ha presentado una propuesta donde los principios de REST han sido aplicados a los estándar y guías de diseño asociadas con la nueva versión de SOAP, es decir, SOAP podría ser utilizado de tal manera que no violara los principios de REST. Esto parece prometedor. Pero en mi opinión, este tipo de propuestas (incluyendo a REST) no triunfarán si la industria no apuesta realmente por ellas (herramientas y frameworks).