

REST vs SOAP

Samuel Enrique Prieto Lozano

Instructor: Edwin Marín

Tecnólogo en Análisis y Desarrollo de Software

Centro de Electricidad, Electrónica y Telecomunicaciones

Servicio Nacional de Aprendizaje – SENA

01 de noviembre de 2024

Bogotá D.C

Contenido

Resumen	3
Abstract.....	3
Marco Teórico	4
Comparación de REST y SOAP	5
Análisis Comparativo y Casos de Uso	6
Conclusiones	7
Bibliografía	8

Resumen

Hoy en día, las APIs (interfaces de programación de aplicaciones) son esenciales para que distintas aplicaciones puedan comunicarse entre sí. Existen dos formas principales de estructurar una API: REST (Transferencia de Estado Representacional) y SOAP (Protocolo Simple de Acceso a Objetos). REST es una forma de comunicación sencilla y flexible, ideal para aplicaciones que buscan rapidez y buen rendimiento, como las aplicaciones móviles y servicios en la nube. SOAP, en cambio, es más formal y seguro, y se utiliza más en entornos corporativos y sistemas críticos. Este ensayo compara ambos enfoques, explicando sus características, ventajas, desventajas y los casos donde uno es más útil que el otro, con el fin de orientar a los desarrolladores en la elección del estilo más adecuado para sus proyectos.

Abstract

APIs have become essential for enabling communication between different applications. Two main approaches to structuring an API are REST (Representational State Transfer) and SOAP (Simple Object Access Protocol). REST is simple and flexible, making it ideal for applications needing speed and efficiency, like mobile apps and cloud services. SOAP, in contrast, is more structured and secure, typically used in corporate settings and critical systems. This essay compares both approaches, explaining their characteristics, advantages, disadvantages, and the scenarios where each is most useful, aiming to guide developers in choosing the best option for their projects.

Marco Teórico

Qué es una API

Una API es básicamente una serie de reglas que permiten que diferentes programas se comuniquen entre sí, intercambiando datos y servicios. En términos simples, actúa como un puente que permite a dos aplicaciones hablar entre sí, aunque hayan sido diseñadas de forma separada. Las APIs de tipo REST y SOAP son dos enfoques distintos que facilitan la integración de sistemas, cada uno con sus propias características y formas de operar.

API REST

REST, o Transferencia de Estado Representacional, es una forma de construir APIs introducida por Roy Fielding en el año 2000. REST se basa en una serie de principios que simplifican la comunicación, como el uso de HTTP (el mismo protocolo usado en la web) y de URLs (direcciones de internet) para identificar recursos o datos. Cada tipo de acción (como leer, actualizar o eliminar información) está asociado a un verbo HTTP: GET para obtener datos, POST para crearlos, PUT para actualizarlos y DELETE para eliminarlos.

La gran ventaja de REST es que permite trabajar con varios formatos de datos (como JSON o XML), aunque JSON es el más popular por su ligereza. Esto hace que REST sea rápido y fácil de usar en aplicaciones modernas, como apps móviles o servicios web. También facilita que se puedan hacer servicios "sin estado", lo que significa que cada solicitud es independiente y el servidor no tiene que recordar el historial de cada usuario, lo cual es útil para gestionar grandes volúmenes de datos de forma eficiente.

API SOAP

SOAP, el Protocolo Simple de Acceso a Objetos, fue creado por Microsoft y la W3C para establecer una forma más estructurada de comunicar aplicaciones. A diferencia de REST, SOAP utiliza siempre XML para estructurar los mensajes, lo cual da más rigidez y garantiza que la información sea muy precisa. SOAP incluye WSDL (Lenguaje de Descripción de Servicios Web), que define de forma detallada los servicios y los métodos de cada API.

Aunque SOAP es más complejo y requiere más recursos (es decir, más tiempo y poder de procesamiento), cuenta con funciones avanzadas como WS-Security, que protege los mensajes mediante cifrado y autenticación, algo crucial en aplicaciones

empresariales. También puede soportar transacciones y comunicación asíncrona (como cuando se necesita que varios sistemas se comuniquen sin fallos), algo importante en sistemas críticos como los bancarios o los de salud.

Comparación de REST y SOAP

Característica	REST	SOAP
Protocolo	Basado en HTTP, también puede usar HTTPS.	Funciona sobre varios protocolos como HTTP, SMTP y TCP.
Formato de datos	JSON, XML, YAML, texto; JSON es el más común.	Solo XML, que es más pesado y complejo.
Seguridad	HTTPS, soporta autenticación con OAuth.	WS-Security para cifrar mensajes y autenticar usuarios.
Escalabilidad	Alta, ideal para aplicaciones ligeras y que necesitan crecer.	Limitada, es más pesado para aplicaciones grandes.
Rendimiento	Rápido y eficiente, especialmente con JSON.	Más lento, XML requiere más procesamiento.
Transacciones	Sin soporte directo; se gestionan en la aplicación.	Tiene soporte para transacciones complejas.
Interoperabilidad	Alta entre sistemas que entienden HTTP y JSON o XML.	Muy alta; usa estándares que permiten la comunicación en varios entornos.
Flexibilidad	Muy flexible; se adapta a muchas necesidades.	Menos flexible por su estructura fija.
Estado	Sin estado, cada solicitud es independiente.	Puede ser con o sin estado; esto hace su gestión más compleja.
Facilidad de desarrollo	Fácil de implementar y mantener, con herramientas modernas.	Más complejo por el uso de XML, WSDL y estándares de SOAP.

Análisis Comparativo y Casos de Uso

Flexibilidad y Simplicidad

REST es popular por ser simple y flexible. Su capacidad para manejar varios formatos de datos y su arquitectura "sin estado" le permite adaptarse a muchos tipos de aplicaciones, desde apps móviles hasta microservicios. La simplicidad de REST hace que sea fácil de entender y rápido de implementar, lo cual ayuda a los desarrolladores a crear aplicaciones de manera eficiente.

Por otro lado, SOAP es un poco más rígido. Al depender de XML y WSDL, necesita más configuración y tiempo, pero esta estructura le da más precisión y seguridad. Esta característica es útil en aplicaciones donde es importante controlar y documentar bien cada operación, como en sistemas de salud o en bancos.

Seguridad y Transacciones

SOAP tiene un sistema de seguridad muy completo llamado WS-Security, que permite cifrar mensajes y verificar la identidad de los usuarios, lo cual es esencial en sistemas que manejan información confidencial. Además, SOAP soporta transacciones complejas, lo que significa que puede ejecutar varias acciones en distintos sistemas de manera coordinada, asegurando que los datos siempre estén correctos, algo esencial en bancos y otros entornos críticos.

REST, aunque no cuenta con un sistema de seguridad tan completo, usa HTTPS para proteger los datos y es compatible con protocolos de autenticación como OAuth, lo cual es suficiente para muchas aplicaciones que no necesitan tanta seguridad.

Rendimiento y Eficiencia

REST, al ser ligero y compatible con JSON, es más rápido en la mayoría de los casos. La estructura sin estado de REST permite que las aplicaciones sean más rápidas, ya que los servidores no necesitan recordar el historial de cada usuario. Esto es útil en aplicaciones que manejan grandes volúmenes de datos y requieren alta disponibilidad, como redes sociales o aplicaciones de mensajería.

SOAP, al usar XML y tener protocolos adicionales, consume más recursos. Esto puede ser una desventaja en cuanto a velocidad, pero es una compensación aceptable en sistemas que necesitan alta seguridad y transacciones fiables.

Casos de Uso de REST y SOAP

REST es ideal en:

- Aplicaciones móviles y web que necesitan rapidez y eficiencia.
- Servicios de microservicios y aplicaciones en la nube, que requieren ser escalables.
- APIs públicas, ya que son fáciles de usar y entender para los desarrolladores.

SOAP es ideal en:

- Aplicaciones empresariales con altos requisitos de seguridad.
- Integraciones complejas entre diferentes sistemas y plataformas.
- Sistemas que manejan información crítica, como los financieros y de salud.

Conclusiones

Decidir entre REST y SOAP depende de las necesidades de cada proyecto. REST es más flexible y ligero, lo que lo convierte en la mejor opción para aplicaciones que requieren rapidez y escalabilidad, como aplicaciones móviles y servicios web modernos. SOAP, en cambio, es un protocolo robusto y seguro, ideal para entornos corporativos que necesitan alta seguridad y control de la información, como bancos y sistemas de salud.

REST y SOAP son enfoques complementarios, y en algunos casos, es posible combinar ambos. Por ejemplo, una empresa podría utilizar REST para sus APIs públicas y SOAP para sus sistemas internos. Al considerar la seguridad, la escalabilidad y el rendimiento, los desarrolladores pueden elegir la opción que mejor se ajuste a las necesidades del proyecto.

Bibliografía

1. Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Dissertation, University of California, Irvine.
2. "SOAP Specifications." W3C. Recuperado de <https://www.w3.org/TR/soap/>
3. "REST API Design." Restfulapi.net. Recuperado de <https://restfulapi.net>
4. Pautasso, C., & Alonso, G. (2013). "Rest vs. SOAP: Understanding the Differences." *IEEE Internet Computing*, 17(3), 56-60.
5. Alam, M. (2019). *Comparative Study of REST and SOAP Web Services*. International Journal of Computer Applications, 178(6), 25-30.
6. Richardson, L., & Ruby, S. (2007). *RESTful Web Services*. O'Reilly Media.