

# Tipos de API - Diferencias entre REST y SOAP

REST y SOAP son dos enfoques distintos para la transmisión de datos en línea. Específicamente, ambas definen cómo diseñar [interfaces de programación de aplicaciones \(API\)](#), las cuales permiten la comunicación de datos entre aplicaciones web. La transferencia de estado representacional (REST) es un conjunto de principios arquitectónicos; mientras que el protocolo simple de acceso a objetos (SOAP) es un protocolo oficial, cuyo [mantenimiento está a cargo del Consorcio World Wide Web \(W3C\)](#). La principal diferencia es que SOAP es un protocolo, y REST no lo es. Por lo general, alguno de los dos regirá a las API, según el caso práctico y las preferencias del desarrollador.



## REST: transferencia de estado representacional

REST es un conjunto de principios arquitectónicos que se ajusta a las necesidades de las [aplicaciones móviles](#) y los [servicios web](#) ligeros. Dado que se trata de un conjunto de pautas, la implementación de las recomendaciones depende de los desarrolladores.

Cuando se envía una solicitud de datos a una API de REST, se suele hacer a través de un protocolo de transferencia de hipertexto, comúnmente denominado HTTP. Una vez que reciben la solicitud, las API diseñadas para REST (conocidas como API o servicios web de RESTful) pueden devolver mensajes en distintos formatos: HTML, XML, texto sin formato y JSON. El formato preferido para los mensajes es la notación de objetos JavaScript (JSON), ya que, a pesar de su nombre, puede leerlo cualquier lenguaje de programación, es ligero y lo comprenden tanto las personas como las máquinas. De esta forma, las API de RESTful son más flexibles y se pueden configurar con mayor facilidad.

Se considera que una aplicación es RESTful si cumple con seis pautas arquitectónicas. Una aplicación de RESTful debe tener lo siguiente:

1. Una arquitectura cliente-servidor compuesta por clientes, servidores y recursos.
2. Una comunicación cliente-servidor [sin estado](#). Esto significa que el contenido de los clientes no se almacena en el servidor entre las solicitudes, sino que la información sobre el estado de la sesión queda en el cliente.
3. Datos que pueden almacenarse en caché para eliminar la necesidad de algunas interacciones cliente-servidor.
4. Una interfaz uniforme entre elementos para que la información se transfiera de forma estandarizada, en lugar de ser específica para las necesidades de cierta aplicación. [Roy Fielding, el creador de REST, lo describe](#) como "la característica principal que distingue el estilo arquitectónico de REST de los demás estilos basados en la red".
5. Una restricción del sistema en capas, en el que las interacciones cliente-servidor pueden estar mediadas por capas jerárquicas.
6. Código según se solicite, lo que permite que los servidores amplíen las funciones de un cliente al transferir el código ejecutable (esto también reduce la visibilidad, así que es una pauta opcional).

# SOAP: protocolo simple de acceso a objetos

SOAP es un protocolo estándar que se creó originalmente para posibilitar la comunicación entre las aplicaciones que se diseñaban con diferentes lenguajes y en distintas plataformas. Como es un protocolo, impone reglas integradas que aumentan la complejidad y la sobrecarga, lo cual puede retrasar el tiempo que tardan las páginas en cargarse. Sin embargo, estos estándares también ofrecen normas integradas que pueden ser ideales para el sector empresarial. Los estándares de cumplimiento integrados incluyen la [seguridad](#), la atomicidad, la uniformidad, el aislamiento y la durabilidad (ACID), que forman un conjunto de propiedades que garantizan operaciones confiables de las bases de datos.

Las especificaciones comunes de los servicios web incluyen lo siguiente:

- **Seguridad de los servicios web (WS-Security):** estandariza la forma de proteger y transferir los mensajes usando identificadores únicos llamados tokens.
- **Mensajería segura de los servicios web (WS-ReliableMessaging):** estandariza el control de errores entre mensajes que se transfieren en infraestructuras de TI poco confiables.
- **Abordaje de los servicios web (WS-Addressing):** paquetes que enrutan la información como metadatos dentro de los encabezados SOAP, en lugar de mantener la información en un lugar más profundo de la red.
- **Lenguaje de descripción de los servicios web (WSDL):** describe qué hace un servicio web, así como dónde comienza y termina.

El envío de una solicitud de datos a una API de SOAP se puede administrar a través de cualquiera de los protocolos de la capa de la aplicación: HTTP (para los exploradores web), SMTP (para el correo electrónico), TCP, entre otros. Sin embargo, una vez que se recibe una solicitud, los mensajes SOAP de retorno deben ser documentos XML, que es un lenguaje de marcado que comprenden las personas y las máquinas. Una solicitud completa a una API de SOAP no se almacena en caché por un navegador, por lo que no se puede acceder a ella después sin reenviarla a la API.

## Diferencias entre SOAP y REST

Es posible que muchos sistemas heredados sigan rigiéndose por SOAP, aunque REST haya surgido más tarde y se considere una alternativa más rápida en los escenarios basados en la Web. REST es un conjunto de pautas que ofrece una implementación flexible, mientras que SOAP es un protocolo con requisitos específicos, como en el caso de la mensajería XML.

Las API de REST son ligeras, así que son ideales para los contextos más nuevos, como el [Internet de las cosas \(IoT\)](#), el desarrollo de aplicaciones móviles y la [informática sin servidor](#). Los servicios web de SOAP ofrecen seguridad y cumplimiento de las operaciones integrados que coinciden con muchas de las necesidades empresariales, pero que también los hacen más pesados. Asimismo, muchas API públicas, como la API de Google Maps, siguen las pautas de REST.