



ACTIVIDAD. INVESTIGACIÓN TECNOLOGÍAS QUE SE UTILIZAN PARA
DESARROLLAR SERVICIOS SOA

DOCENTE: JESÚS LEONARDO LÓPEZ HERNÁNDEZ

“ARQUITECTURA ORIENTADA A SERVICIOS”

ALUMNO:

MAURICIO PÉREZ SALAS 7A

MARCO LAGUNES MONTERO 7A

31/10/2025

Concepto general de SOAP

SOAP (Simple Object Access Protocol) es un protocolo de mensajería basado en XML que define reglas claras para el intercambio de información entre sistemas distribuidos. Su propósito es garantizar interoperabilidad entre distintas plataformas, contratos formales bien definidos y soporte para escenarios empresariales donde se requiere seguridad, confiabilidad y control.

Mensajes y contrato del servicio

Los mensajes SOAP se representan en XML y siguen una estructura estándar:

- Envelope: contenedor principal del mensaje.
- Header (opcional): incluye información adicional como seguridad, transacciones o metadatos.
- Body: contiene los datos de la operación (parámetros de entrada o resultados).
- Fault: sección destinada a reportar errores de forma estructurada.

Para asegurar que estos mensajes sean coherentes, se utilizan:

- XSD (XML Schema Definition): define la estructura, tipos de datos y restricciones de los mensajes XML.
- WSDL (Web Services Description Language): documento XML que describe el servicio web, indicando operaciones disponibles, mensajes de entrada y salida, tipos de datos utilizados, protocolo de transporte y dirección (endpoint). El WSDL funciona como un contrato formal entre el proveedor del servicio y los clientes que lo consumen.

Transporte de los mensajes

Los mensajes SOAP se transportan principalmente sobre HTTP o HTTPS, aprovechando la infraestructura web existente y permitiendo una integración sencilla con aplicaciones empresariales. Sin embargo, también pueden utilizar otros medios como SMTP (correo electrónico) o sistemas de mensajería (por ejemplo, JMS) cuando se requieren arquitecturas más complejas o comunicación asíncrona.

Extensiones WS-*

Alrededor de SOAP existe una familia de estándares conocida como WS-* que agrega funcionalidades avanzadas necesarias en entornos corporativos:

- WS-Security: proporciona mecanismos de autenticación, firma digital y cifrado a nivel de mensaje.
- WS-Addressing: incorpora información de direccionamiento dentro del mensaje, útil cuando intervienen múltiples nodos o rutas complejas.

- WS-ReliableMessaging: define reglas para asegurar la entrega confiable de mensajes (sin pérdidas, sin duplicados y en el orden correcto).
- WS-Policy: permite especificar políticas y requisitos del servicio, como niveles de seguridad o condiciones de uso.

Enfoques de desarrollo

Existen dos enfoques principales para construir servicios SOAP:

- Contract-first: primero se definen el WSDL y los XSD, y a partir de ese contrato se genera el código del servidor y de los clientes. Este enfoque es común cuando se busca un alto nivel de control y estandarización.
- Code-first: primero se implementa la lógica del servicio en el lenguaje elegido y luego se genera el WSDL automáticamente. Es más rápido de iniciar, pero requiere disciplina para mantener la compatibilidad del contrato.

Frameworks y plataformas de implementación

Las tecnologías concretas varían según la plataforma, pero todas respetan los mismos estándares (XML, SOAP, WSDL, WS-*), lo que permite la interoperabilidad:

- En Java: JAX-WS / Jakarta XML Web Services, Spring Web Services, Apache CXF, Axis2, con generación automática de stubs y clientes a partir del WSDL.
- En .NET: WCF (Windows Communication Foundation), con soporte integrado para SOAP y WS-*. Visual Studio permite generar proxies directamente desde el WSDL.
- En PHP: extensión SOAP nativa para publicar y consumir servicios definidos por WSDL.
- En Python: librerías como zeep o spyne que gestionan mensajes SOAP sin construir el XML manualmente.
- En otros lenguajes: librerías específicas que siempre trabajan a partir del contrato WSDL.

7. Herramientas de prueba y soporte

Para diseño, pruebas y validación de servicios SOAP se emplean herramientas como SoapUI y Postman, que permiten importar el WSDL, generar solicitudes, configurar encabezados de seguridad y verificar respuestas. Estas herramientas facilitan la comprobación de que el servicio cumple correctamente con el contrato definido y funcionan como apoyo fundamental en el ciclo de desarrollo.