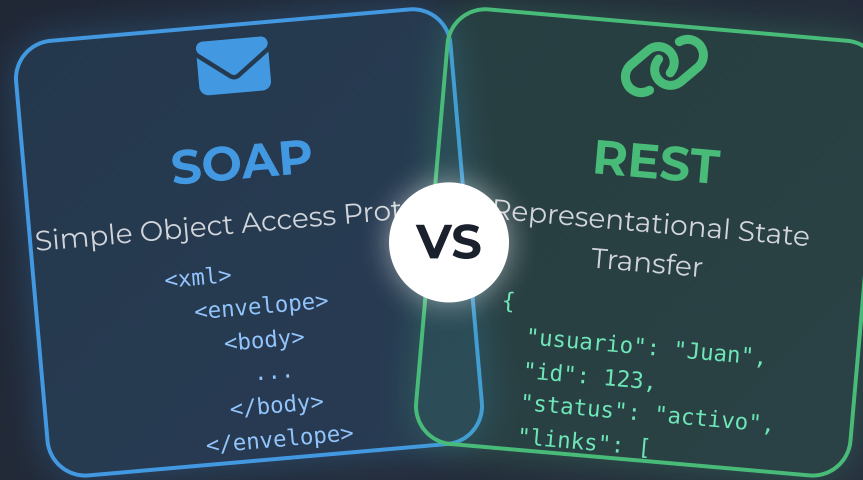


SOAP vs REST

Diferencias, Analogías y Aplicaciones



Dos arquitecturas para la comunicación entre sistemas en la web. **SOAP** es un protocolo estándar con reglas estrictas, mientras que **REST** es un estilo arquitectónico con mayor flexibilidad.

CONCEPTOS BÁSICOS

Entendiendo los fundamentos de SOAP y REST

SOAP

Definición

Protocolo estándar basado en XML que define una estructura estricta de mensajes para intercambiar información en implementaciones de servicios web.

✓ Protocolo

📄 XML

🛡️ Estricto

↔️ WS-*

Analogía

Como enviar una **carta formal** por correo certificado:

- Necesita un sobre específico (envelope)
- Tiene formato estandarizado
- Incluye instrucciones de manejo
- Contiene verificación de entrega



REST

Definición

Estilo arquitectónico que utiliza HTTP para acceder a recursos representados por URLs, transmitiendo datos típicamente en formato JSON.

🏗️ Arquitectura

🌐 HTTP

📄 JSON

🔗 Recursos

Analogía

Como **visitar habitaciones** de una casa:

- Cada habitación tiene una dirección (URL)
- Puedes mirar (GET) o modificar (POST, PUT)
- No necesitas instrucciones especiales
- Cada habitación contiene recursos específicos



Ambos permiten la comunicación entre aplicaciones, pero con enfoques fundamentalmente diferentes

ESTRUCTURA Y FORMATO

Comparación entre mensajes SOAP y REST

SOAP (XML)

Estructura del Mensaje

- 📁 Envelope (Sobre)
- 📄 Header (Cabecera) - Opcional
- 📄 Body (Cuerpo) - Obligatorio
- ⚠️ Fault (Error) - Opcional

Ejemplo de Mensaje SOAP

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

  <soap:Header>
    <auth>
      <username>usuario1</username>
      <password>*****</password>
    </auth>
  </soap:Header>

  <soap:Body>
    <getClientes>
      <id>12345</id>
    </getClientes>
  </soap:Body>
</soap:Envelope>
```

- Estructura rígida
- Validación estricta
- Compatible con WSDL

REST (JSON)

Estructura de Recursos

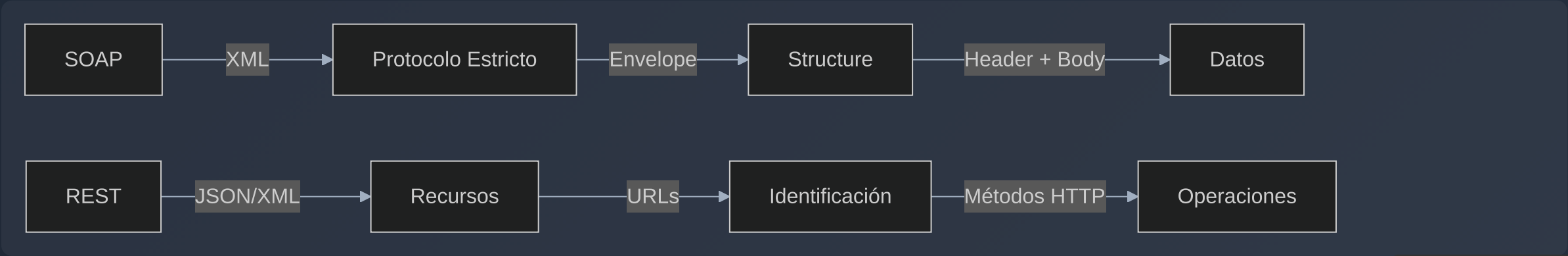
- 🌐 Endpoint (URL del recurso)
- ↔️ Métodos HTTP (GET, POST, PUT, DELETE)
- 📄 Cuerpo del mensaje (opcional, JSON/XML)
- 🏷️ Parámetros y headers HTTP

Ejemplo de Respuesta REST (JSON)

```
// GET https://api.ejemplo.com/clientes/12345

{
  "id": 12345,
  "nombre": "Juan Pérez",
  "email": "juan@ejemplo.com",
  "activo": true,
  "fechaAlta": "2023-01-15",
  "direccion": {
    "calle": "Av. Principal 123",
    "ciudad": "Madrid",
    "codigoPostal": "28001"
  },
  "pedidos": [
    {
      "id": 9742,
      "fecha": "2023-02-01",
      "monto": 150.50
    },
    {
      "id": 9743,
      "fecha": "2023-02-05",
      "monto": 200.00
    }
  ]
}
```

- Estructura flexible
- Basado en recursos
- Hipermedia (HATEOAS)



VENTAJAS Y DESVENTAJAS

Comparación entre SOAP y REST

SOAP

+ VENTAJAS



Seguridad Robusta

WS-Security proporciona protección de punto a punto para transacciones sensibles.



Confiabilidad

Garantía de entrega de mensajes con WS-ReliableMessaging.



Transacciones Complejas

Ideal para operaciones que requieren ACID y transacciones distribuidas.



Estándares Definidos

Frameworks y reglas claras para implementación y operación.

- DESVENTAJAS



Sobrecarga

Mensajes XML extensos requieren más ancho de banda y procesamiento.

REST

+ VENTAJAS



Ligereza

Mensajes más pequeños que consumen menos ancho de banda.



Escalabilidad

Mejor rendimiento para aplicaciones web con muchos clientes.



Simplicidad

Fácil de entender e implementar con herramientas estándar.



Compatibilidad Web

Funciona con HTTP estándar y navegadores comunes.

- DESVENTAJAS



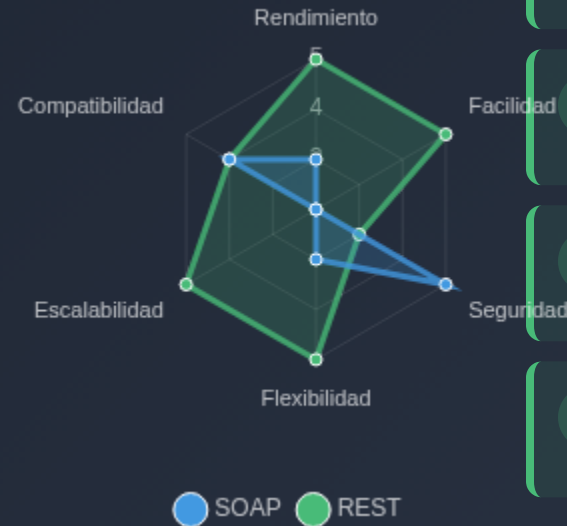
Seguridad Básica

Depende de la seguridad de la capa de transporte (HTTPS).



Falta de Estándares

SOAP VS REST



Comparación de atributos clave
(valores más altos = mejor)

CASOS DE USO

Cuando usar SOAP vs REST en aplicaciones reales

SOAP

✓ Ideal para: Operaciones críticas, transacciones complejas, alta seguridad



FINANZAS

Banca Online

Transferencias bancarias y sistemas de pagos que requieren alta seguridad y cumplimiento regulatorio.

- ✓ WS-Security para cifrado
- ✓ Transacciones ACID



TELECOMUNICACIONES

Provisión de Servicios

Activación de servicios telefónicos, gestión de red y sistemas de facturación integrados.

- ✓ Mensajes confiables
- ✓ Integración ERP



SALUD

Sistemas Médicos

Intercambio de historias clínicas y datos de pacientes bajo estrictas normas de privacidad.

- ✓ Cumplimiento HIPAA
- ✓ Registros auditables



ENTERPRISE

CRM y ERP

Integración entre sistemas corporativos de diferentes generaciones y plataformas.

- ✓ Interoperabilidad
- ✓ Sistemas legados

💡 EJEMPLO REAL

Salesforce API: utiliza SOAP para operaciones complejas en su CRM, permitiendo transacciones seguras y confiables entre su plataforma y sistemas empresariales externos.

REST

✓ Ideal para: Aplicaciones web, móviles, microservicios, APIs públicas



SOCIAL

Redes Sociales

APIs públicas para integración de contenido, timeline y funciones sociales en aplicaciones de terceros.

- ✓ Alto tráfico
- ✓ Caché eficiente



MÓVIL

Apps Móviles

Comunicación eficiente con backends para aplicaciones móviles que requieren bajo consumo de datos.

- ✓ Bajo ancho de banda
- ✓ Respuestas ligeras



E-COMMERCE

Tiendas Online

Catálogos de productos, reseñas y sistemas de recomendación basados en recursos web.

- ✓ Escalabilidad
- ✓ Microservicios



IOT

Dispositivos IoT

Comunicación liviana entre dispositivos conectados con recursos limitados y servicios en la nube.

- ✓ Bajo procesamiento
- ✓ Orientado a recursos

💡 EJEMPLO REAL

Twitter API: utiliza REST para permitir que desarrolladores integren contenido de Twitter, publiquen tweets y accedan a datos de usuarios desde aplicaciones externas de manera simple y eficiente.

SOAP

Grandes empresas Sistemas críticos

TENDENCIA DE ADOPCIÓN



REST

Startups Aplicaciones modernas

ANALOGÍAS VISUALES

Entendiendo SOAP y REST de manera intuitiva



SOAP es como...

Correo Certificado

- Sobre formal y estandarizado
- Múltiples capas de protección
- Confirmación de entrega
- Proceso bien documentado
- Mayor peso pero más seguro

vs



REST es como...

Nota Adhesiva

- Mensaje directo y conciso
- Liviano y fácil de crear
- Se coloca en un lugar visible
- No requiere formalidades
- Eficiente para comunicación simple

💡 Como dejar un mensaje simple en el lugar exacto donde se necesita.



SOAP es como...

Llamada Formal

- Protocolo estricto de comunicación
- Introducción, cuerpo, despedida
- Se graba para mantener registros
- Canal dedicado y seguro
- Independiente del contexto exterior

💡 Como una conversación telefónica empresarial con agenda y formalidades.



REST es como...

Visitar una Casa

- Cada habitación es un recurso con URL
- Entrás y ves (GET) lo que hay
- Puedes añadir (POST) o cambiar (PUT)
- Estado visible en la habitación misma
- Flexibilidad para visitar cualquier sala

💡 Como navegar por una casa donde cada sala tiene una dirección específica.



SOAP es como...

Viajar en Avión

- Procesos de seguridad rigurosos
- Equipaje con normas estrictas
- Personal especializado
- Reglas uniformes en todas las aerolíneas
- Más costoso pero más capacidad

vs



REST es como...

Andar en Bicicleta

- Simple y directo
- Ligero y eficiente para distancias cortas
- Flexible para cambiar de ruta
- Fácil de aprender y usar
- Límites de carga pero muy ágil

💡 Como un medio simple y eficiente para llegar a destinos cercanos.



SOAP es como...

Restaurante Formal

- Protocolo estricto de servicio
- Menú extenso con formalidades
- Órdenes específicas y detalladas
- Procesos estandarizados para cada plato
- Formato consistente en cada interacción

💡 Como un restaurante donde cada solicitud sigue un protocolo formal.



REST es como...

Food Court

- Cada restaurante es un recurso diferente
- URLs claras (letreros) de cada local
- Pides directamente lo que quieres
- Puedes elegir entre múltiples formatos
- Interacciones simples y bajo demanda

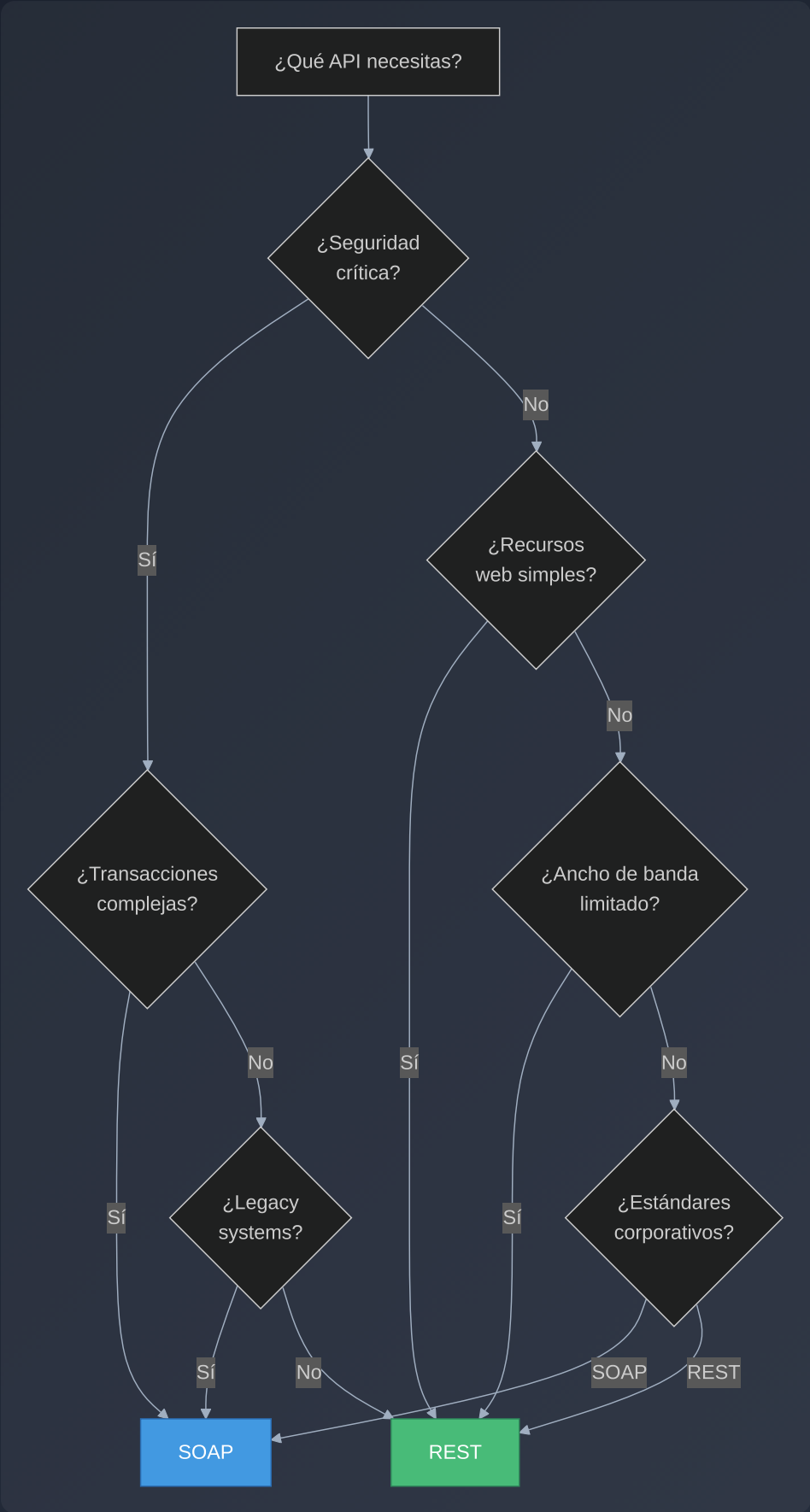
💡 Como visitar locales independientes donde cada uno tiene su propia oferta.

SOAP es como sistemas con **protocolos formales y estructurados** mientras que REST es como **interacciones simples basadas en recursos**

CONCLUSIONES Y RECOMENDACIONES

Eligiendo la arquitectura adecuada para tu proyecto

Guía de Decisión



Recomendaciones Finales

Usa SOAP cuando:

- Necesites garantías formales de seguridad (WS-Security)
- Trabajes con transacciones distribuidas complejas
- Requieras procesamiento asíncrono confiable
- Debas integrarte con sistemas legacy
- Tu arquitectura demande contratos formales (WSDL)
- La confiabilidad sea más importante que el rendimiento

- Finanzas
- Salud
- Enterprise

Usa REST cuando:

- Necesites alto rendimiento y escalabilidad
- Desarrolles para aplicaciones móviles o web
- Trabajes con ancho de banda limitado
- Construyas APIs públicas
- Implementes arquitecturas de microservicios
- La simplicidad y velocidad sean prioritarias

- Móvil
- E-commerce
- SaaS

Enfoque Híbrido

Muchas organizaciones implementan ambas arquitecturas para diferentes propósitos. Es común ver REST para APIs públicas y móviles, mientras que SOAP se utiliza internamente para servicios críticos que requieren alta seguridad y transacciones complejas.

Recursos Adicionales



Especificación SOAP
W3C SOAP Specification



Principios REST
RESTful API Design Guidelines



Herramientas de Desarrollo
Postman (REST) / SoapUI (SOAP)

SOAP vs REST: Diferencias, Analogías y Aplicaciones

© 2023 - Todos los derechos reservados

