1-SERVICIO WEB

Un servicio web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes y ejecutan sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet .

2-Diferencias entre rest y soap.

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)](https://www.redhat.com/es/topics/internet-of-things-570051), el desarrollo de aplicaciones móviles y la [informática sin servidor](https://www.redhat.com/es/topics/cloud-native-apps/what-is-serverless). 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.

Para aplicaciones donde la estructura de los mensajes y las operaciones no puede cambiar y deben mantenerse estables, se requiere de autenticación o sesiones y la comunicación no necesariamente vas a ocupar *Http*, *Soap* puede ser una opción.

Con *SOAP* ganas estabilidad a través de una estructura solida, En cambio, con REST ganas facilidad de cambio y adaptabilidad. Cual es mejor depende de la problemática que se pretende solucionar.

La seguridad es parte del diseño. Por si mismo usar un protocolo “seguro” es un esfuerzo estéril si no hay un diseño de tu sistema que busque ofrecer seguridad.

Ventajas de una API REST para el desarrollo de proyectos

Entre las principales ventajas que destacamos de API REST para tener una buena presencia en el desarrollo de proyectos:

– Un distanciamiento entre el cliente de API REST y el sistema de servidor, dando lugar a una gran ventaja permitiendo un mayor desarrollo de los proyectos y permitiendo un mayor crecimiento de manera individualizada.

– Una gran**individualidad por parte de API REST** hacia el lenguaje o la utilización de diferentes plataformas. Gracias a que AIR REST dispone de una gran facilidad para poder ajustarse a este tipo de herramientas, aportando una mayor autonomía en su procedimiento.

– Una **mayor flexibilidad** para nuestro proyecto, gracias a que API REST se adapta a las decisiones que tomemos. Como resultado permitiendo una mayor escalabilidad y poder ir evolucionando según lo que el cliente necesite en cada momento.

– El desarrollo de un proyecto de un sitio web basado en una API da lugar a un mayor resultado que la utilización de un sitio web basado en un sistema tradicional.

– La implantación de API REST da lugar a **solicitar menos requerimientos al servidor**. Debido a no disponer de memoria y poder solucionar un número más elevado de demandas. Al igual que da lugar a un menor proceso en el servidor al no ser necesario escribir el código HTML.

3-Principal razón para usar SOAP

En el futuro a corto y mediano plazo, es probable que SOAP se siga utilizando para servicios web de nivel empresarial que requieren transacciones complejas y de alta seguridad. Las API para servicios financieros, pasarelas de pago, software de CRM, gestión de identidades y servicios de telecomunicaciones son ejemplos de SOAP comúnmente utilizados.

Una de las API de SOAP más conocidas es la API pública de PayPal que le permite aceptar pagos con PayPal y con tarjeta de crédito, agregar un botón de PayPal a su sitio web, permitir que los usuarios inicien sesión con PayPal y realizar otras acciones relacionadas con PayPal.

La compatibilidad con sistemas heredados es otro argumento frecuente para usar SOAP. Los servicios web populares que existen desde hace un tiempo pueden tener muchos usuarios que aún se conectan a sus servicios a través de su API SOAP, que era el líder del mercado antes de que REST ganara popularidad. Salesforce, por ejemplo, proporciona una API SOAP y REST para que cada desarrollador pueda integrar Salesforce con su propia plataforma de la manera que más le convenga.

4) Limitaciones de REST-API

* El sistema de API REST, al poder disponer de más de un servidor, no sabes de forma concreta cuál atenderá tu demanda al sistema y como solucionará tu solicitud que previamente le haz confirmado.
* API REST no sustenta el estado de la aplicación, implicando que el cliente tenga que establecer una estructura propia para mantener el conjunto de la aplicación. Para ello será necesario enviarle al servidor un token para que pueda identificarnos e indicarle a la misma vez la función desempeñada en la aplicación.
* Para poder desarrollar una estructura para poder mantener el conjunto de la aplicación, tendrá que descuidar parte de su tiempo en el desarrollo de ello.
* Este desarrollo requiere una serie de procedimiento de larga duración y formación para poner en marcha estableciendo un nuevo lenguaje, protocolo HTML y bases de datos para lograr la creación de esta nueva estructura propia.
* Requiere un mayor esfuerzo, debido a que es conveniente la realización de pruebas para poder comprobar que todo funciona correctamente.

5-REST vs JSON

La arquitectura REST permite a los proveedores de API entregar datos en múltiples formatos, como texto sin formato, HTML, XML, YAML y JSON, que es una de sus características más queridas. Gracias a la creciente popularidad de REST, el formato JSON liviano y legible por humanos también ganó terreno rápidamente, ya que es muy adecuado para el intercambio rápido de datos.

JSON significa Notación de objetos de JavaScript. Es un formato de intercambio de datos ligero y fácil de analizar. A pesar de su nombre, JSON es completamente independiente del idioma, por lo que se puede usar con cualquier lenguaje de programación, no solo con JavaScript. Su sintaxis es un subconjunto del estándar ECMA-262 3.ª edición. Los archivos JSON consisten en colecciones de pares de nombre/valor y listas ordenadas de valores que son estructuras de datos universales utilizadas por la mayoría de los lenguajes de programación. Por lo tanto, JSON se puede integrar fácilmente con cualquier idioma.

JSON es un formato más liviano y menos detallado, y también es más fácil de leer y escribir. En la mayoría de los casos, es ideal para el intercambio de datos a través de Internet. Sin embargo, XML todavía tiene algunas ventajas. Por ejemplo, le permite colocar metadatos dentro de las etiquetas y también maneja mejor el contenido mixto, especialmente cuando las matrices de nodos mixtos requieren expresiones detalladas.

6- las razones por las que elegir API-REST son:

>Por su escalabilidad. Este protocolo destaca por su escalabilidad. Gracias a la separación entre el cliente y el servidor, el producto se puede escalar con un equipo de desarrollo sin que ello represente muchas dificultades.

>Por su flexibilidad y portabilidad. Con el requisito imprescindible de que los datos de cada una de las peticiones sean enviados de forma correcta, es posible realizar una migración de un servidor a otro o practicar cambios en la base de datos en todo momento. De está forma el front y el back se pueden alojar en servidores diferentes, lo que supone una enorme ventaja de manejo.

>Por su independencia. Debido a la separación entre el cliente y el servidor, el protocolo facilita que los desarrollos de las diferentes partes de un proyecto se pueden dar de manera independiente. Además de ello, la apirest se adapta en todo momento al tipo de sintaxis o plataformas de trabajo. Esto brinda la oportunidad de probar varios entornos dentro del desarrollo.

10- Dado que REST y SOAP son dos enfoques distintos para la transmisión de datos en linea cada una cuenta con ventajas y desventajas en su uso que puede hacer que te decantes por uno u otro.

Por ejemplo con SOAP se gana estabilidad a través de una estructura solida, en cambio con REST ganas facilidad de cambio y adaptabilidad, podemos llegar a decir que uno es mejor que el otro pero cada una es diferente.

7) Limitaciones de SOAP-API

* Debido al uso de XML, SOAP es considerablemente más lento que otras apis y que los datos binarios se codifican como texto.
* Tiene dependencia del WSDL (Web Service Description Language).
* Al contrario que Java, PHP y Python, ciertos lenguajes de programación no ofrecen un apoyo adecuado para su uso, ya sea a nivel de integración o de soporte de ideas.
* Al estar ampliamente estandarizado, es poco flexible y suele haber muchos errores a la hora de desarrollo si no se conocen dichos estándares. Al utilizar el protocolo TCP también tiene un peor rendimiento que otro tipo de Web Services.

8- Tabla de comparación SOAP vs. REST

| Soap | Rest |
| --- | --- |
|  |  |
| Es Un Protocolo | Es un estilo arquitectónico |
| Significa Simple Object Access Protocol | Significa Transferencia de estado representacional |
| No Puede Usar Rest | Puede Usar Soap de servicios web |
| Utiliza interfaces de servicios para exponer la lógica de negocios | Utiliza URI para exponer la lógica de negocio |
| Usa JAX-WS | Usa JAX-RS |
| Define las normas que deben seguirse estrictamente | **No** define tantos estándares como soap |
| Requiere más ancho de banda | Requiere menos ancho de banda |
| Define su propia seguridad | Hereda medidas de seguridad |

9- ¿Cómo decidir entre SOAP o REST?

**SOAP**

Una vez enviados los datos, se procesarán al igual que el tiempo de envío y recepción, dándonos unos 2-15 segundos para facturar.

En nuestro lenguaje de programación, cuando recibimos una respuesta del servicio web, tenemos que analizar la respuesta a través del documento DOM para obtener los datos importantes, lo que agrega bastante tiempo.

Cuando el Servicio Web se vuelva más popular, es probable que el servicio se sature cada vez más y esté se vuelva más lento.

**REST**

Cuando enviamos datos, solo enviaremos aquello que es importante, en cuestión de timbrado sería el XML, el cual se procesa de manera rápida (Hacer verificación de memoria caché para evitar un timbre duplicado), lo cual nos deja con tiempo de 100ms – 1 segundo para la emisión de una factura.

En nuestro lenguaje de programación, cuando obtenemos la respuesta, analizamos JSON (más rápido), lo que equivale a unos pocos milisegundos.

Cuando el Servicio Web se vuelva más popular, simplemente se añaden más instancias en la nube, así el trabajo se distribuye usando un balanceador de carga.